Available on CMS information server

CMS DP -2023/009

The Compact Muon Solenoid Experiment

CMS Performance Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

22 March 2023

Performance of the track selection DNN in Run 3

CMS Collaboration

Abstract

The DP-Note describes the DNN used for track selection in Run3 and shows the physics performance in Run3 simulation. The tracking efficiency, fake rate and duplicate rate are shown as a function of p_T, η, pileup for $t\bar{t}$ events and as a function of the track displacement for stop-antistop events.

Performance of the track selection DNN in Run 3

Detector Performance Note

The CMS Collaboration, March 2023

cms-pog-conveners-tracking@cern.ch

In memoriam

We dedicate this DP-NOTE to our colleague Minxi Yang, who contributed to the development and optimization of the track selection DNN

Introduction

Iterative Tracking at CMS

- The CMS track reconstruction follows an iterative approach [1]: the reconstruction algorithm is run several times starting starting from relatively easier tracks (higher pT, low displacement, 4 pixel hits in the seeds) and moving to more complex tracks in the later iterations
- Each iteration has 4 main steps:
  - Seeding
  - Pattern recognition
  - Track fit
  - Track selection

Track selection

- After the pattern recognition and the fit, based on Kalman Filter techniques, high purity tracks are selected and the hits belonging to those tracks are not used in the following iterations, thus keeping the complexity of the pattern recognition under control for later iterations.
- The track selection was gradually improved: starting with a parametric selection in Run 1 [1], moving to a BDT in Run 2, and to a DNN in Run 3.
- The track selection DNN is presented in this DP-NOTE

Disclaimer

Several developments happened in parallel between Run 2 and Run 3

- Part of the tracking iteration were switched from the legacy CKF algorithm [1] to the mkFit algorithm [2,3] for Run 3
  - In the current default tracking the InitialStepPreSplitting, Initial, HighPtTriplet, DetachedQuad, DetachedTriplet iterations use mkFit. The rest, including PixelLess, uses CKF as in Run 2.
- The track selection was updated from a BDT to a DNN, while switching to mkFit
  - The DNN was developed initially for pure CKF reconstruction, but later for both mkFit and CKF tracks, which have slight differences, most notably in the total number of hits
  - In the current default tracking a DNN trained on mkFit tracks is used for the mkFit iterations, while a DNN trained on the CKF tracks is used for the CKF iterations
  - The BDT was trained on CKF only, but the conditions were different at the beginning of Run 2 (lower pileup and center of mass energy)

In the following slides, the Run 3 DNN and the improvement compared to BDT are highlighted. The DNN performance is shown in the current default tracking (mkFit +CKF Run 3) and compared to the result of the Run 2 BDT on the same tracks.

- The same comparison could be more coherent using pure CKF, but the conditions were also different and no training has been done on the same set of tracks, so the default tracking is preferred

DNN Inputs and target

Input Features

- the track p_T, η, ϕ, and their respective uncertainties δp_T, $\delta\eta$, $\delta\phi$
- p_x, p_y, p_z, p_T for the innermost and outermost state of the track
- the transverse and longitudinal impact parameters, d_0, d_z, computed both from the beamspot and from the closest primary vertex, and their respective uncertainties δd_0, δd_z
- the track χ^2 and number of degrees of freedom
- number of Pixel hits, number of Strip hits
- number of missing hits inside the innermost hit and outside the outermost hit
- number of inactive layers crossed inside the innermost hit and outside the outermost hit
- number of layers without hits overall
- the iteration flag (integer)

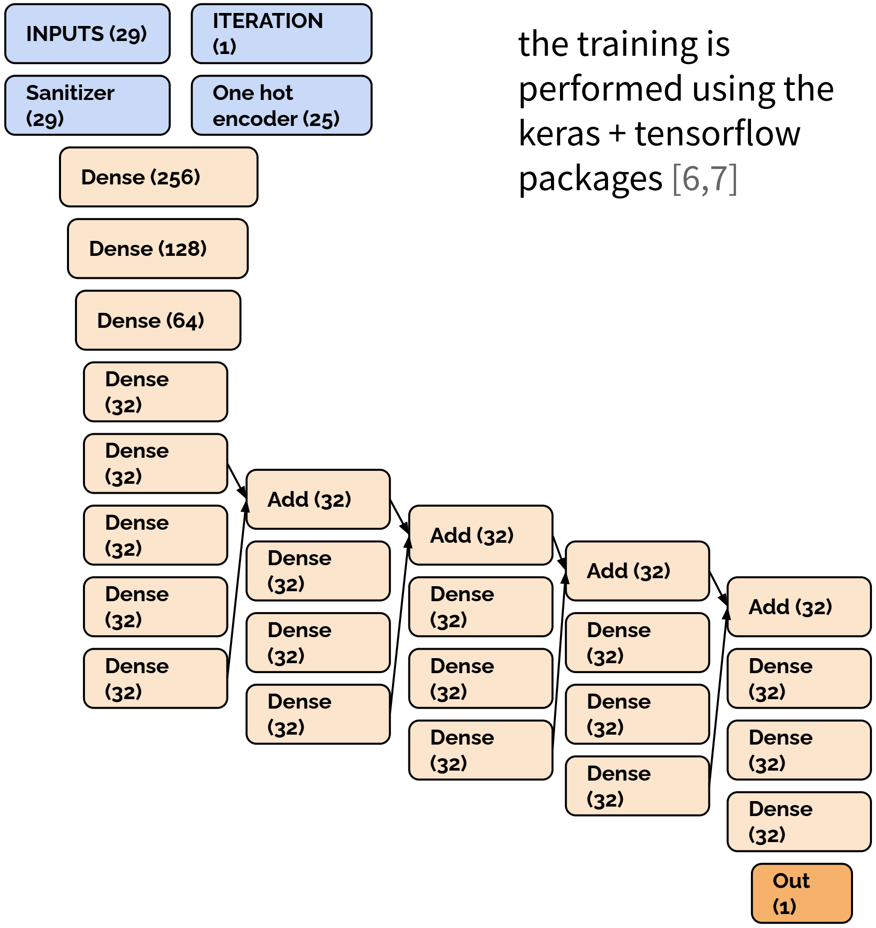
Target

- true/false flag: a true track must have more than 75% of its hits matched to a simulated track.

The track selection DNN

<u>DNN architecture</u>

- Relatively simple feed-forward network, with 5 iteration of “skip connection” and sum of the layer outputs in the downstream layers
- The “sanitizer” layer applies log/absolute value transformations to some of the inputs, while the “one hot encoder” converts the iteration flag into a boolean vector by category
- Activations:
  - ELU [4] in hidden layers
  - sigmoid for output
- Loss function: binary cross-entropy



the training is performed using the keras + tensorflow packages [6,7]

Training procedure and working points

Training procedure

- Training performed on tracks, including tracks from pileup vertices, from several simulated samples all generated at a center-of-mass energy of 14 TeV with pileup sampled from minimum bias events with Poisson mean distributed flat from 20 to 70
  - QCD multijet production - generated with a flat hard-scattering p_T from 15 GeV to 7 TeV
  - $t\bar{t}$ production
  - Drell-Yan with Z decaying into electrons
  - Stop-antistop ($\tilde{t}\bar{\tilde{t}}$) production in RPV SUSY, with stop masses $m(\tilde{t})$ of 1 TeV and 1.8 TeV and stop decay lengths $c\tau_0(\tilde{t})$ of 10 or 100 cm. These samples are used to increase the amount of displaced tracks
- The track selection is not applied to tracks used in training
  - All the tracks are labeled as “high purity” and the hit masking for the later iterations uses from the previous iterations
  - This is a reasonable approximation and it allows to train for all the tracking iteration in a single step
- Batch size 512, Adam optimizer [5]
- 5 training epochs over 1.3B tracks

Choice of the working points

- The working point are chosen iteration by iteration in a validation sample similar to the training one
  - The efficiency is chosen to roughly match the BDT efficiency
- The choice of the working point is validated in tracking with the hit masking applied

Tracking efficiency, fake rate & duplicate rate

- A reconstructed track is considered associated to a simulated particle if more than 75% of its hits have been originated from this simulated particle. If this is not the case, the reconstructed track is considered as a random combination of hits and marked as a misidentified (fake) track.
- Simulated tracks coming from the signal (hard scattering) vertex are used in the efficiency computation. The tracking efficiency is defined as the fraction of simulated tracks associated to at least one reconstructed track
- All simulated tracks coming from any vertex (including pileup vertices) are used in the fake rate and duplicate rate computation. The tracking fake rate is defined as the fraction of misidentified reconstructed tracks; the tracking duplicate rate is defined as the fraction of reconstructed tracks associated multiple times to the same simulated track.

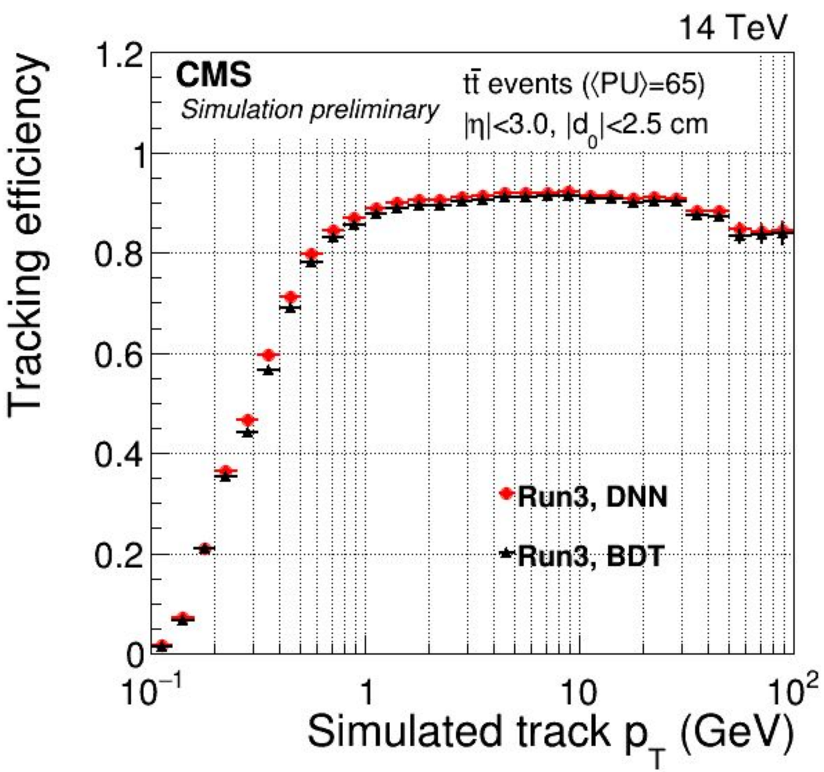
- *The performance has been measured in both a simulated $t\bar{t}$ sample and in a sample with stop-antistop production in RPV SUSY, similar to the ones used in training, where the stops have a significant decay length and produce displaced tracks, i.e:*

$$pp \rightarrow \tilde{t}\,\bar{\tilde{t}} \qquad \textit{with} \qquad \tilde{t} \rightarrow bl\ (l=e, \mu, \tau)\ , \qquad m(\tilde{t})=800\ \mathrm{GeV}\ , \qquad c\tau_0(\tilde{t})=50\ \mathrm{cm}$$

with superimposed pileup events. The number of pileup events is sampled from minimum bias events with Poisson mean distributed flat from from 55 to 75. The detector conditions match the most recent Run 3 simulation. The efficiency, fake rate, duplicate rate are shown as a function of p_T, η, pileup for the $t\bar{t}$ sample, while the same quantities are shown as a function of the track displacement for the $\tilde{t}\,\bar{\tilde{t}}$ sample.

- *The physics results are shown after applying the high purity BDT or DNN selection to each iteration and after merging all the tracks from the iterations into one collection*

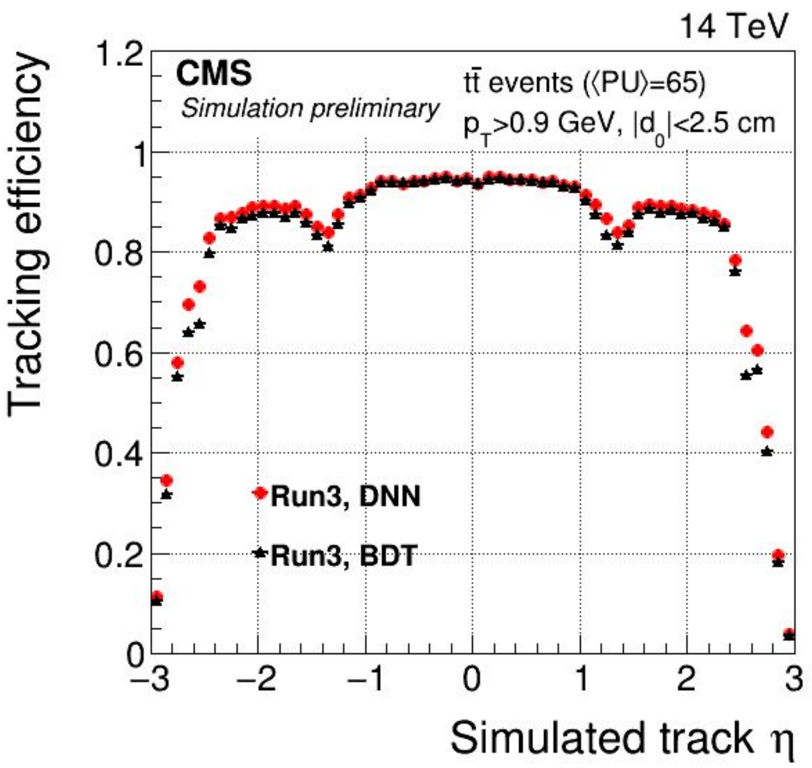
Tracking efficiency vs p_T



- The tracking efficiency is shown as a function of the simulated track p_T for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for simulated tracks with $|\eta|<3.0$ and $|d_0|<2.5$ cm

- The tracking efficiency when the DNN is used is consistent or slightly higher than the one obtained using the BDT across the entire p_T range. The efficiency improves the most at low p_T, up to 5%

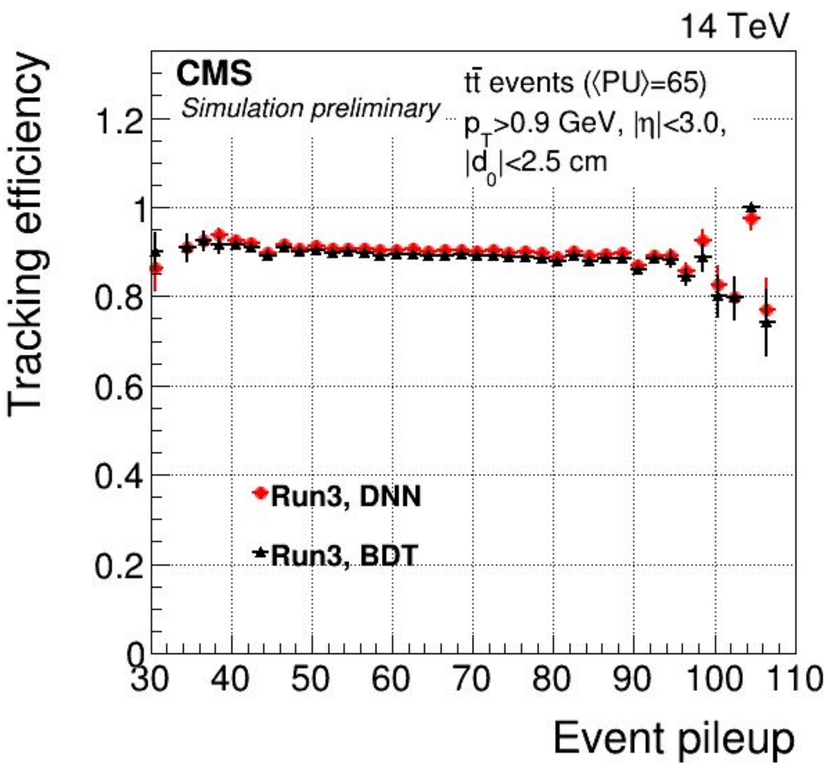
Tracking efficiency vs η



- The tracking efficiency is shown a function of the simulated track pseudorapidity η for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for simulated tracks with p_T>0.9 GeV and $|d_0|$<2.5 cm

- The tracking efficiency when the DNN is used is consistent or slightly higher than the one obtained using the BDT in all the η regions. The improvement is at most 2%

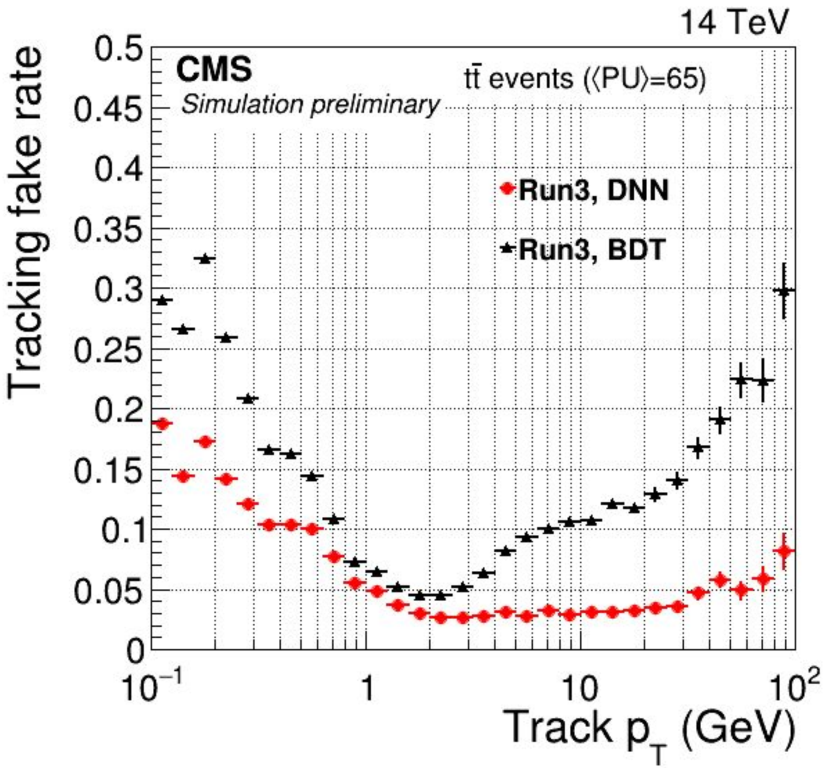
Tracking efficiency vs PU



- The tracking efficiency is shown as a function of the event pileup (PU) for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for simulated tracks with p_T>0.9 GeV, |η|<3.0 and $|d_0|$<2.5 cm

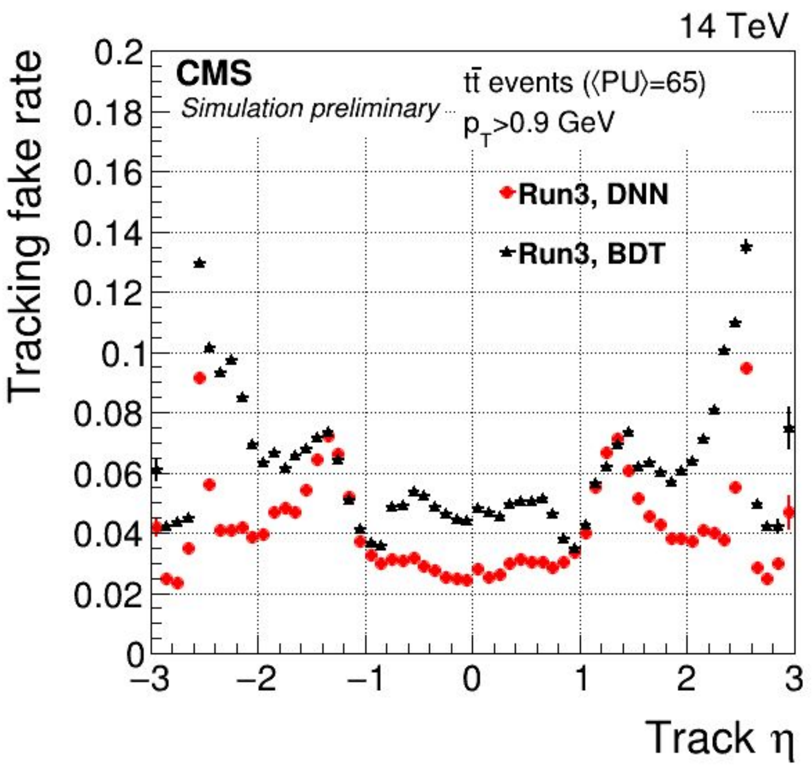
- The tracking efficiency when the DNN is used is consistent or slightly higher than the one obtained using the BDT independently of the PU. Overall the efficiency is increased by 1%

Tracking fake rate vs p_T



- The tracking fake rate is shown as a function of the reconstructed track p_T for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black)
- The tracking fake rate when the DNN is used is notably lower than the one obtained using the BDT, especially for very low and very high p_T values. Overall the fake rate is reduced by about 40%

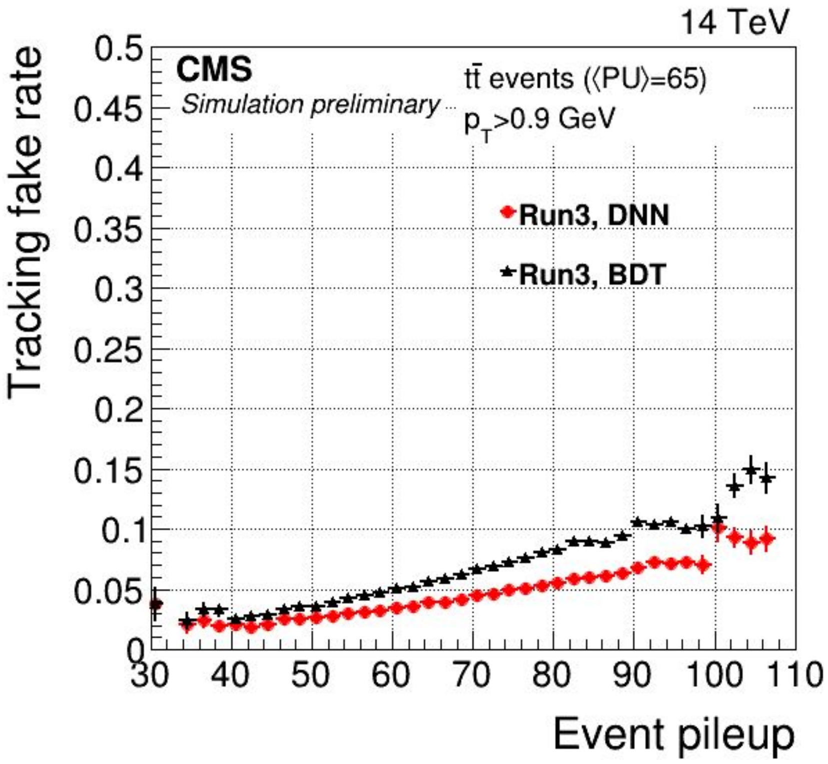
Tracking fake rate vs η



- The tracking fake rate is shown as a function of the reconstructed track η for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for tracks with p_T>0.9 GeV

- The tracking fake rate when the DNN is used is lower or consistent with the one obtained using the BDT. The largest fake rate reductions are in the tracker endcaps (|η|>2) and in the barrel (|η|<1). The discontinuities follow the tracker regions

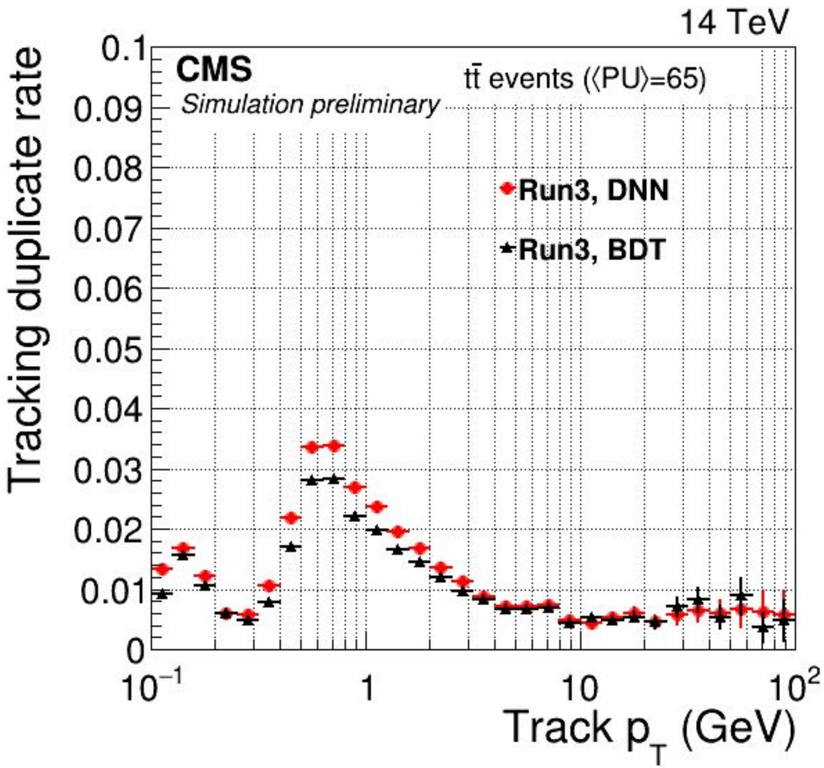
Tracking fake rate vs PU



- The tracking fake rate is shown as a function of the event pileup (PU) for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for tracks with p_T>0.9 GeV

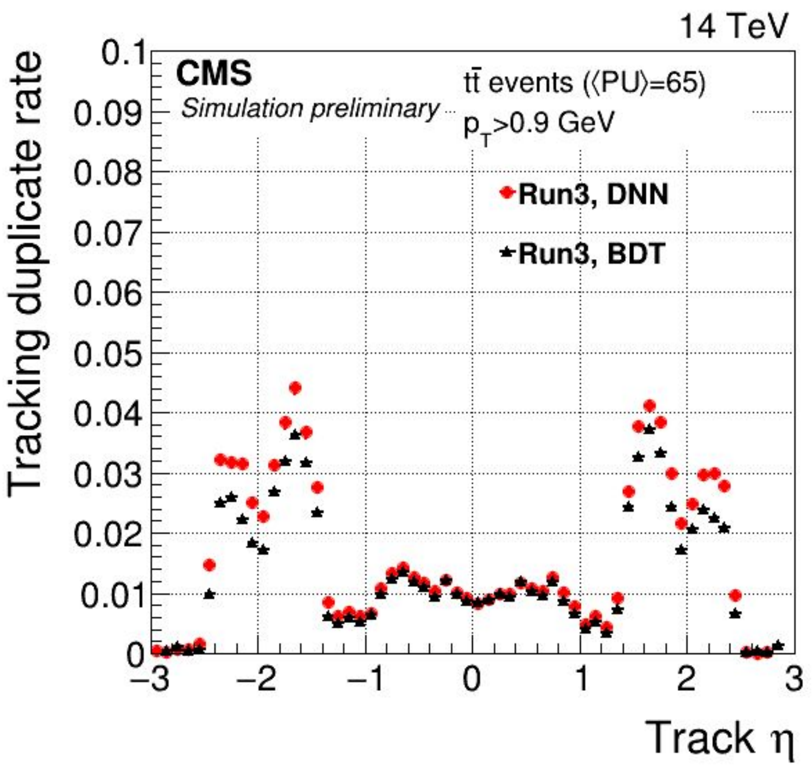
- The tracking fake rate when the DNN is used is lower than the one obtained using the BDT across the full PU range, with a reduction up to about 30% for higher PU values

Tracking duplicate rate vs p_T



- The tracking duplicate rate is shown as a function of the reconstructed track p_T for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black)
- The tracking duplicate rate is slightly increased when the DNN is used instead of the BDT. The overall increase is around 20%
- A slightly higher duplicate rate is expected due to merging of mkFit and CKF tracks selected by different DNNs

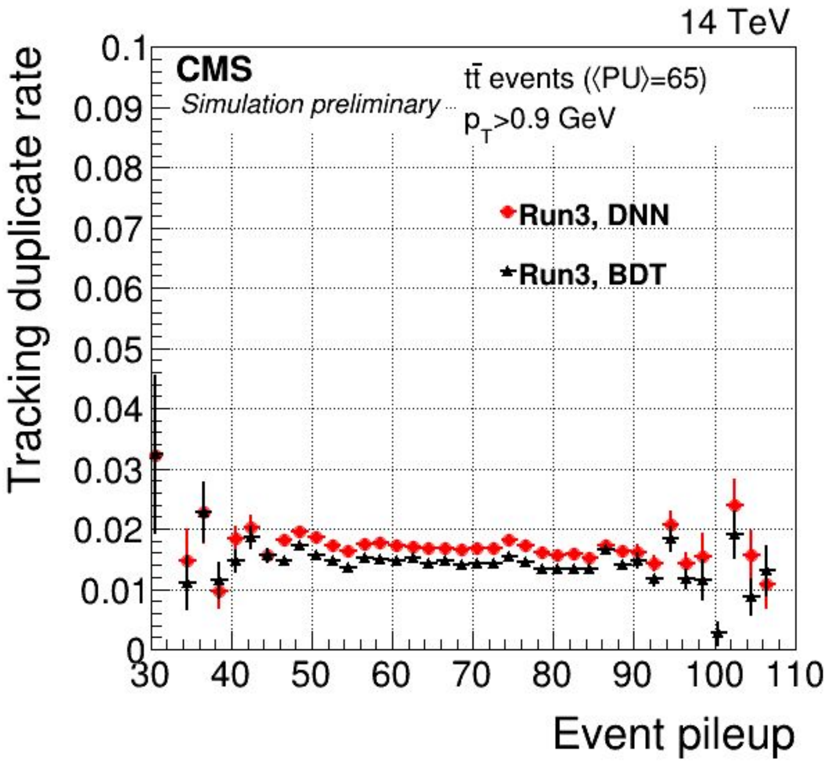
Tracking duplicate rate vs η



- The tracking duplicate rate is shown as a function of the reconstructed track η for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for tracks with p_T>0.9 GeV

- The tracking duplicate rate when the DNN is used is higher or consistent with the one obtained using the BDT. In particular the increase is visible in the endcap and transition regions (|η|>1). The increase is up to 20%

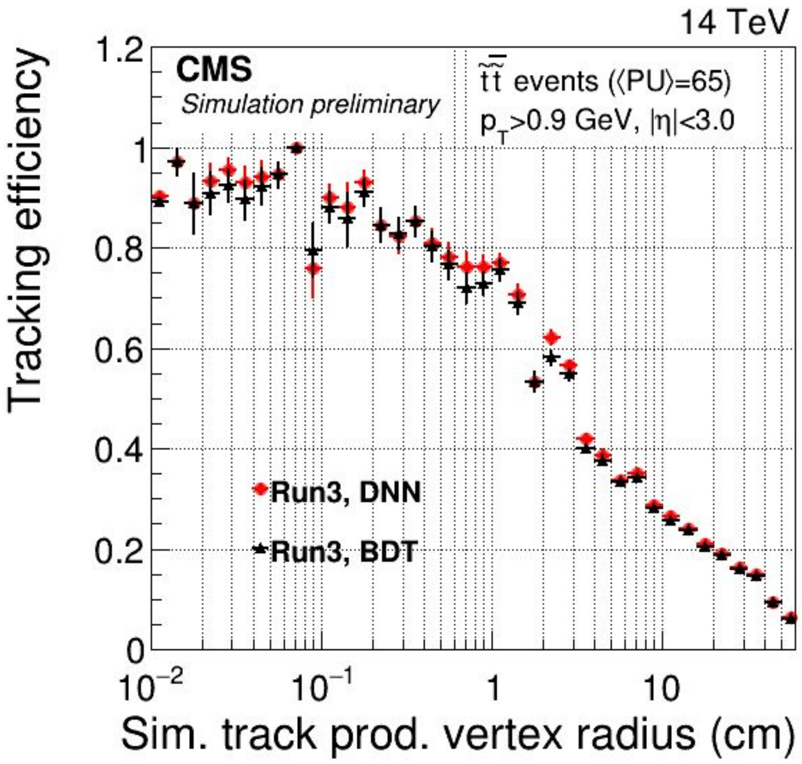
Tracking duplicate rate vs PU



- The tracking duplicate rate is shown as a function of the event pileup (PU) for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for tracks with p_T>0.9 GeV

- The tracking duplicate rate when the DNN is used is higher than the one obtained using the BDT. The increase of about 20% is consistent across the entire PU range

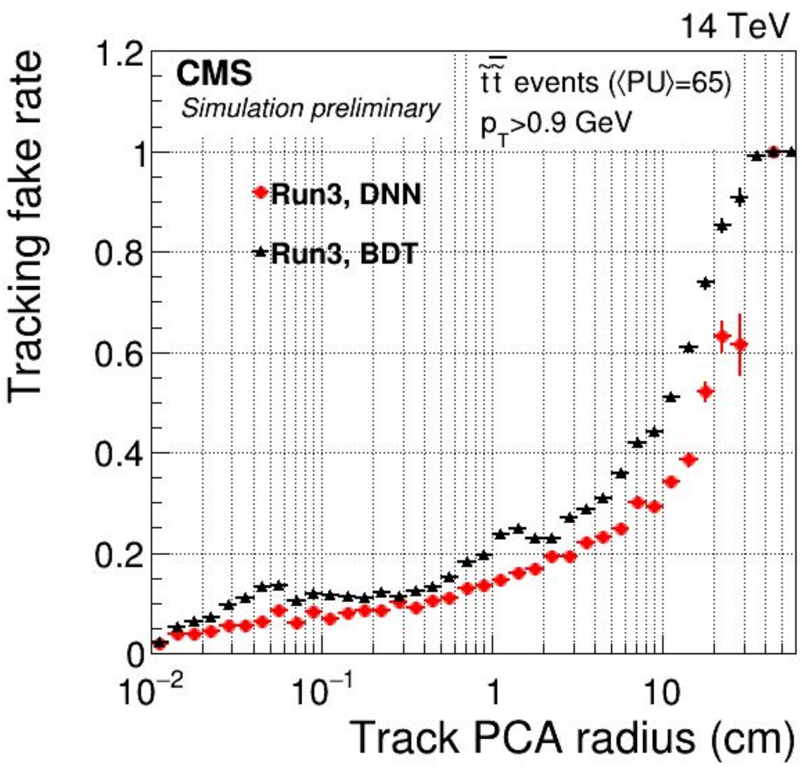
Tracking efficiency vs radius in $pp \rightarrow \tilde{t}\,\bar{\tilde{t}}$



- The tracking efficiency is shown as a function of the simulated track production radius for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for simulated tracks with p_T >0.9 GeV, $|\eta|$<3.0

- The tracking efficiency when the DNN is used is consistent or slightly higher than the one obtained using the BDT at all radii. Notice the higher statistics for high radius values (>1cm) in the stop-antistop production with long stop decay lengths, as evident in decreasing statistical uncertainties for the corresponding bins

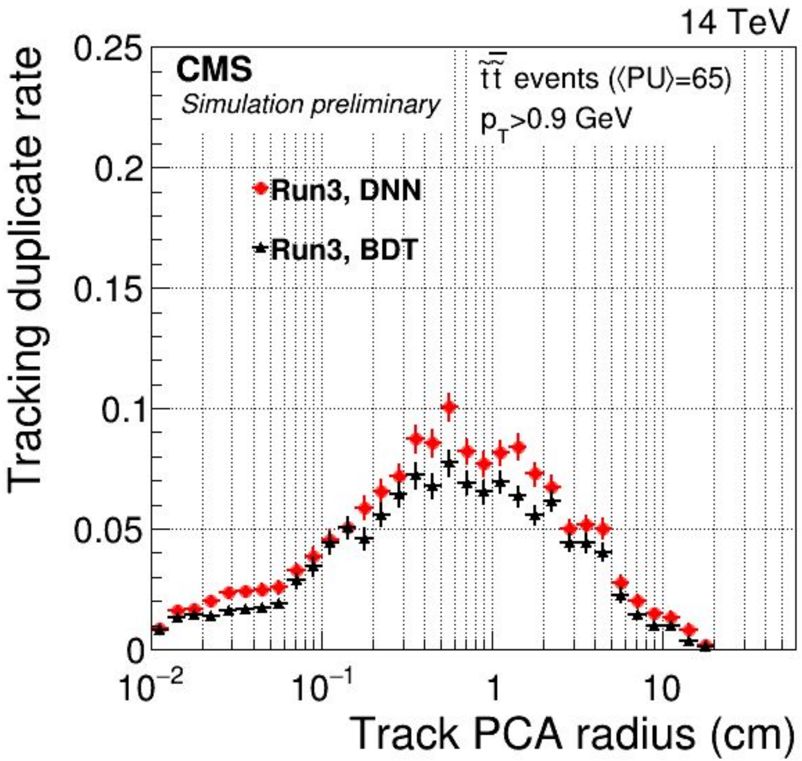
Tracking fake rate vs radius in $pp \to \tilde{t}\,\bar{\tilde{t}}$



- The tracking fake rate is shown as a function of the radius of the track point of closest approach to the beamline (or d_0) for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for tracks with p_T>0.9 GeV

- The tracking fake rate when the DNN is used is lower than the one obtained using the BDT across all the radii values, with a reduction of about 30%

Tracking duplicate rate vs radius in pp $\rightarrow \tilde{t}\,\bar{\tilde{t}}$



- The tracking duplicate rate is shown as a function of radius of the track point of closest approach to the beamline (or d_0) for the high purity tracks selected by the DNN (red) and for the high purity tracks selected by the BDT (black), for tracks with p_T>0.9 GeV

- The tracking duplicate rate when the DNN is used is higher than the one obtained using the BDT for all the radii, by about 20%

References

[1] Description and performance of track and primary-vertex reconstruction with the CMS tracker, CMS Collaboration, e-Print: 1405.6569 [physics.ins-det], DOI: 10.1088/1748-0221/9/10/P10009, Published in: JINST 9 (2014) 10, P10009

[2] Speeding up particle track reconstruction using a parallel Kalman filter algorithm, Steven Lantz (Cornell U.), Kevin McDermott (Cornell U.), Michael Reid (Cornell U.), Daniel Riley (Cornell U.), Peter Wittich (Cornell U.) et al., e-Print: 2006.00071 [physics.ins-det], DOI: 10.1088/1748-0221/15/09/P09030, Published in: JINST 15 (2020) 09, P09030

[3] Performance of Run 3 track reconstruction with the mkFit algorithm, CMS-DP-22/018, CMS Collaboration

[4] Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUS), Djork-Arne Clevert, Thomas Unterthiner & Sepp Hochreiter, e-Print: 1511.07289 [cs.LG]

[5] Adam: A Method for Stochastic Optimization, Diederik P. Kingma, Jimmy Lei Ba, e-Print: 1412.6980 [cs.LG]

[6] https://keras.io. F. Chollet et al., Keras, Software available from https://github.com/keras-team/keras

[7] TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems M. Abadi et al., Software available from tensorflow.org, e-Print: 1603.04467 [cs.DC]