GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN IN PUBLICA COMMODA
SEIT 1737

Fakultät für
Physik

Bachelor's Thesis

Auswirkungen simulierter Detektordefekte auf die Tracking-Performance des ATLAS-ITk

Effects of Simulated Detector Defects on the ATLAS ITk Tracking Performance

prepared by

Yinge Chen

born in Luohe, Henan, China

at the II. Physikalischen Institut

Thesis number:	II.Physik-UniGö-BSc-2025/08
Thesis period:	17th June 2025 until 23rd September 2025
First referee:	Prof. Dr. Stan Lai
Second referee:	PD Dr. Benjamin Schwenker

Abstract

This study aims to simulate the tracking performance of the ATLAS Inner Tracker (ITk), which is designed to replace the current Inner Detector (ID) for the High-Luminosity LHC (HL-LHC) era. The goal is to evaluate how different track reconstruction chains and detector deactivation scenarios impact the quality of reconstructed tracks.

Two track reconstruction chains are studied: the default reconstruction chain, which is currently used in Run 3 and supports both ID and ITk geometries, and the fast track reconstruction chain, which is specifically developed for the HL-LHC to reduce computational load and increase reconstruction speed. Tracking performance is compared between these two chains using a $t\bar{t}$ Monte Carlo sample of 250 events, under both ideal detector conditions and various scenarios with partial detector deactivation. In these masking-only scenarios, deactivation is implemented by injecting a defect map during the early stages of reconstruction, which marks specific regions of the ITk geometry as non-functional. Reconstruction results from the masked configurations are compared against those from the ideal detector to assess the relative performance, this setup allows for a systematic evaluation of the robustness of each reconstruction chain under imperfect detector conditions.

Abstract

Diese Studie untersucht die Leistungsfähigkeit der Spurrekonstruktion des ATLAS Inner Trackers (ITk), der im Rahmen des High-Luminosity LHC (HL-LHC) den derzeitigen Inner Detector (ID) ersetzen soll. Ziel ist es, zu evaluieren, wie sich verschiedene Spurrekonstruktionsketten und Szenarien mit teilweiser Detektordeaktivierung auf die Qualität der rekonstruierten Spuren auswirken.

Es werden zwei Spurrekonstruktionsketten betrachtet: die Standardrekonstruktionskette, die aktuell in Run 3 verwendet wird und sowohl ID- als auch ITk-Geometrien unterstützt, sowie die schnelle Spurrekonstruktionskette, die speziell für den HL-LHC entwickelt wurde, um die Rechenlast zu reduzieren und die Rekonstruktionsgeschwindigkeit zu erhöhen. Die Leistungsfähigkeit der beiden Ketten wird anhand eines $t\bar{t}$-Monte-Carlo-Datensatzes mit 250 Ereignissen verglichen, sowohl unter idealen Detektorbedingungen als auch unter verschiedenen Szenarien mit partieller Detektordeaktivierung. In diesen reinen Maskierungsszenarien wird die Deaktivierung durch die Anwendung einer Defektkarte in den frühen Phasen der Rekonstruktion realisiert, wodurch bestimmte Bereiche der ITk-Geometrie als inaktiv markiert werden. Die Rekonstruktionsergebnisse dieser maskierten Konfigurationen werden mit denen des idealen Detektors verglichen, um die relative Leistungsfähigkeit zu beurteilen. Diese Konfiguration ermöglicht eine systematische Bewertung der Robustheit der jeweiligen Spurrekonstruktionskette unter nicht-idealen Detektorbedingungen.

Contents

1. Introduction

The Standard Model of particle physics provides a well-tested description of the known fundamental particles and their interactions [1–4]. Its predictions have been tested and verified by many decades of experimental results, including the discovery of the Higgs boson at the Large Hadron Collider (LHC) in 2012 [5, 6]. Nonetheless, the SM does not account for phenomena such as dark matter, neutrino masses, or the matter–antimatter asymmetry [7–9]. Progress in addressing these questions requires both precise SM measurements and searches for new phenomena, which in turn rely on data collected in high-energy particle collisions with well-instrumented detectors.

One experimental method for studying particle physics is the proton–proton collider. Proton–proton collisions at high center-of-mass energy enable the study of short-distance (hard) parton interactions. Although the hadronic initial state is not fixed event by event, it is well constrained by universal parton distribution functions, yielding a wide variety of final states that are sensitive to rare processes. The LHC is, at present, the only proton–proton collider in operation at the multi-TeV scale, hosting several interaction points where large general-purpose experiments such as ATLAS and CMS record collision data [10]. The accelerator complex is being upgraded to the High-Luminosity LHC (HL-LHC). The high pile-up and radiation environment expected in that era require ATLAS to replace its current Inner Detector with an all-silicon Inner Tracker (ITk) that includes pixel and strip subdetectors [11].

In practice, an ideal detector does not exist. Inefficiencies already arise during construction, as the production of complex silicon modules inevitably leads to a fraction of units that fail to meet operational specifications. During operation, various effects, including but not limited to radiation damage, can cause parts of the detector to become non-functional. In fact, concerns about limited yield in the production of pixel modules for the ITk have prompted considerations of a staged installation approach, where data-taking could begin with only a subset of the detector installed and operational [12].

In this study, a simplified "mask-only" approach is used to obtain a first estimate of which spatial regions of the ITk are most critical for maintaining tracking performance, from a purely geometrical standpoint. For each Monte Carlo sample and defect configuration, two different track reconstruction chains are applied. The primary simulated dataset that consists of $t\bar{t}$ events at $\sqrt{s} = 14$ TeV with 250 events, and an independent 500-event sample is used for cross-checks.

Within the ATHENA framework [13], detector defect maps are defined at various granularities (pixel layers, pixel modules, strip layers, strip modules, and front-end core-columns). Clusters falling into defective regions are removed during reconstruction. This method does not model detailed analog effects of radiation damage, such as charge collection loss, increased noise, or timing shifts, and instead provides a preliminary estimate of geometrical robustness.

Two distinct tracking workflows are available in ATHENA. The first is the default reconstruction chain, referred to as `newStrategy` in the Athena track reconstruction framework [14]. Unless explicitly overridden, this is the standard tracking workflow compatible with both the Inner Detector (ID) and the ITk geometries. It follows the traditional reconstruction approach, including a seeding stage followed by ambiguity solving; The second chain is the fast track reconstruction [15], referred to internally as `itkFastTrackingStrategy`. This is a still-developing offline reconstruction workflow designed specifically for the HL-LHC ITk geometry. Its motivation is to reduce computational cost by removing the conventional ambiguity solving stage entirely. Instead, it relies on more aggressive shared cluster handling during seeding and tighter η-dependent cuts to suppress the fake rate.

The structure of this thesis is as follows. Chapter 2 introduces the ATLAS experiment, the current ID, the design and assembly of the ITk, and the overall context of the HL-LHC upgrade. Chapter 3 describes the two tracking reconstruction chains used throughout this study, along with their key characteristics and implementation in the simulation.

Chapter 4 presents the main results. First, a baseline comparison between the two reconstruction chains is performed on an ideal, defect-free detector. Second, random pixel cell masking is applied at varying levels (from 2% to 20%) to evaluate the robustness of both chains. Third, per-layer pixel module masking is studied to address the practical question: if approximately 30% of the modules were to fail, which placement of defects would minimize the impact on tracking performance under the default track reconstruction chain. Fourth, strip detector masking results are presented, including some counter-intuitive cases that prompted a full re-run of the masking configurations under the fast track reconstruction chain. Conclusions and outlook are summarized in Chapter 5.

For reasons of length and to maintain a smooth reading flow, certain technical but important details, such as the internal ATHENA representation and indexing of the detector geometry, as well as specific code implementations and supplementary datasets, are collected in the Appendix. Among these, Appendix A and Appendix B provide a detailed description of the technical aspects of the two track reconstruction chains studied in this work.

2. The ATLAS Detector and the HL-LHC Upgrade

2.1. The ATLAS Detector in Run 3

The LHC at CERN [10], located near Geneva, is currently the world's only operating proton-proton collider at the multi-TeV energy scale. Protons are accelerated through a sequence of machines: Linac4, the Proton Synchrotron Booster, the Proton Synchrotron, and the Super Proton Synchrotron before being injected into the LHC ring at 450 GeV. In Run 3, they are further accelerated to reach a centre-of-mass energy of up to 13.6 TeV [16, P. 9]. Along the 27-kilometre ring, nine experiments are positioned at various interaction points or along the beamline, each with its own scientific focus. ATLAS (A Toroidal LHC ApparatuS), situated at Interaction Point 1, is one of the two large general-purpose detectors [16]. It is designed to explore a broad physics program, ranging from precision measurements within the Standard Model to searches for new phenomena beyond it.

The ATLAS detector is arranged in concentric layers around the collision point, forming a nested structure. Each layer has a distinct role in recording and identifying the diverse particles produced in high-energy proton-proton collisions. The innermost region is devoted to tracking charged particles. Surrounding the tracker are calorimeters, which measure particle energies, and the outermost system is dedicated to muon detection. A solenoidal magnetic field of 2 T encloses the tracking system, bending the paths of charged particles to allow momentum measurement, while large air-core toroidal magnets provide bending power for the muon spectrometer.

In Run 3, the tracking system, known as the Inner Detector (ID), consists of three subdetectors (Figure 2.1). Closest to the beam pipe is the Pixel Detector, which is augmented by the Insertable B-Layer (IBL) added between Run 1 and Run 2 to improve the resolution of track impact parameters, thereby enhancing vertex reconstruction and flavour-tagging performance [16, P. 39]. These layers are built from silicon pixel sensors that provide precise spatial measurements near the interaction point. With approximately 80 million readout channels, the Pixel Detector plays a central role in reconstructing primary vertices and identifying secondary vertices from heavy-flavour decays within the beam pipe. Each track typically traverses three layers segmented in $r\phi$ and z, with pixel sizes of 50 μm $\times$ 400 μm and intrinsic spatial resolutions of 10 μm in $r\phi$ and 115 μm in z in the barrel region. Similar performance is achieved in the endcaps.

Surrounding the Pixel Detector is the Semiconductor Tracker (SCT), composed of silicon microstrip sensors arranged in four barrel layers and nine endcap disks. Each track crosses eight strip layers (providing four space points in total). The SCT extends the tracking coverage to larger radii, linking hits from the inner pixel layers to those in the outer tracker. In the barrel region, each module consists of two daisy-chained 6.4 cm sensors with a strip pitch of 80 μm, featuring small-angle (40 mrad) stereo strips to enable measurements in both $r\phi$ and z. The endcaps adopt a similar geometry with radial and stereo strip orientations. The intrinsic spatial resolutions per module are approximately 17 μm in $r\phi$ and 580 μm in z (barrel) or r (disks). The SCT comprises approximately 6.3 million readout channels [16, P. 14].

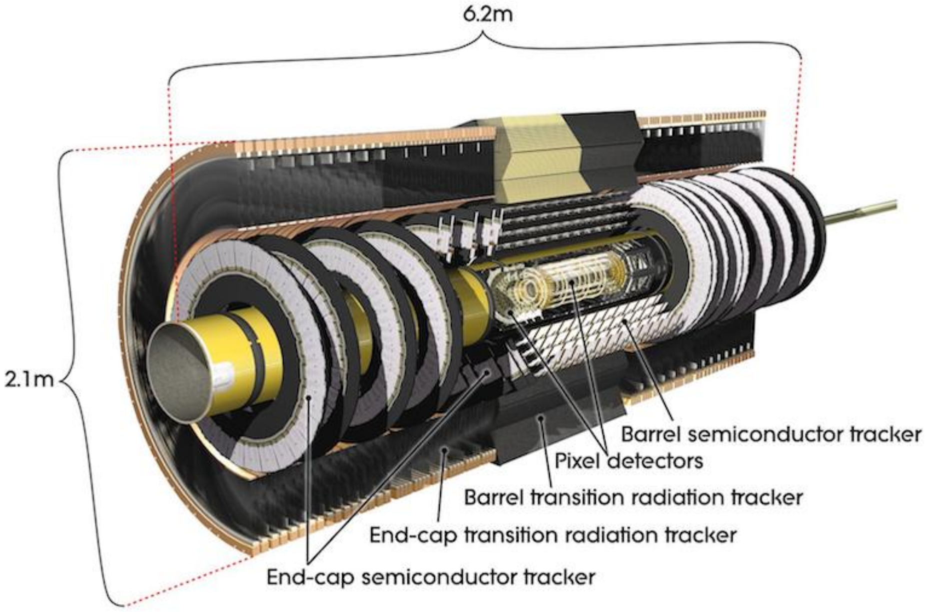


Figure 2.1.: Overview of the ATLAS Inner Detector layout. The pixel, SCT, and TRT subsystems are shown in both barrel and endcap regions. This system will be replaced by the ITk during the HL-LHC upgrade. Figure adapted from [17, P. 6].

The outermost part of the ID in Run 3 is the Transition Radiation Tracker (TRT), the only subsystem in the ID that is not silicon-based. It consists of thin cylindrical gas-filled straw tubes that register hits when charged particles ionise the gas as they pass through. The freed electrons drift towards a central anode wire, producing an amplified signal.

Because of their large number and fine segmentation, the TRT straws contribute many position measurements per track [17], improving pattern recognition and momentum determination. In addition to tracking, the TRT can identify electrons through transition radiation: high-energy electrons emit X-ray photons when crossing materials of differing dielectric properties, which in turn produce higher-amplitude signals in the straws. This capability provided complementary particle identification in Run 3, although gaseous detectors like the TRT are less suitable for the high-radiation, high-occupancy environment expected at the HL-LHC.

Geometrically, the ID is divided into a central barrel and two endcaps. In the barrel region, the Pixel, SCT, and TRT are arranged in concentric cylinders aligned with the beam axis. The endcaps consist of disk-shaped modules (Pixel and SCT) and radial straw "wheels" (TRT) oriented perpendicular to the beam, extending coverage to particles emitted at small polar angles. This layout ensures that most particles within the acceptance cross several detector layers at near-normal incidence, which is optimal for hit resolution and efficiency. The Run 3 ID provides tracking coverage up to pseudorapidity $|\eta| = 2.5$ [16], as shown in Figure 2.1.

As charged particles traverse the silicon tracking detectors, they deposit charge that is collected on readout channels (pixels or strips) and digitized if it exceeds the electronics threshold. Neighbouring above-threshold channels are then grouped by a clustering algorithm to form clusters, which provide a position estimate and an associated uncertainty. In strip detectors, two stereo clusters are combined to form a two-dimensional space point. Pattern-recognition and track fitting use these cluster-level measurements (clusters or space points) as inputs. By linking hits across successive layers, trajectories are reconstructed and particle momenta are obtained from the curvature in the magnetic field, as shown in Figure 2.2. Extrapolating the fitted tracks back to the interaction region allows reconstruction of primary vertices and the identification of displaced vertices from long-lived decays.

The calorimeter system measures the energy deposited by particles as they interact with matter and is one of the key subsystems for particle identification, energy measurement, and event reconstruction, such as jets and missing transverse energy. Calorimeters are generally classified functionally into electromagnetic and hadronic types: electromagnetic calorimeters measure the energy of electrons and photons, while hadronic calorimeters sample the energy of hadrons, such as protons and neutrons, through nuclear interactions. In ATLAS, however, the system is organized based on detection technology into the Liquid Argon (LAr) Calorimeter and the Tile Hadronic Calorimeter. The LAr Calorimeter includes both electromagnetic components, such as the electromagnetic barrel (EMB), end-

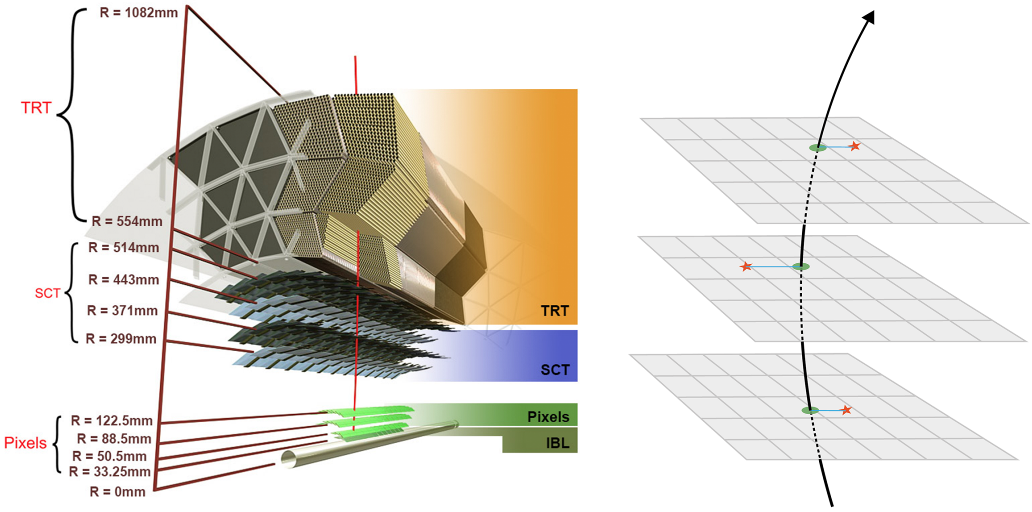


Figure 2.2.: Left: layout of the ATLAS Run 3 Inner Detector showing the pixel system (including the IBL), the silicon microstrip SCT, and the Transition Radiation Tracker (TRT) with representative radii. Right: concept of tracking: a charged particle traverses successive layers, producing cluster-level measurements (hits/space points) that are linked by pattern recognition into a reconstructed track in the solenoidal field. Image courtesy of @CERN.

cap (EMEC), and forward calorimeter (FCal), and hadronic components like the hadronic endcap (HEC). The Tile Calorimeter serves as the hadronic barrel and is divided into the Tile Barrel and Tile Extended Barrel. This technological classification maps closely onto the functional division, with different components optimized for measuring different types of particle showers [18].

Calorimeters are capable of stopping most known particles, with the exception of muons and neutrinos. Neutrinos are electrically neutral and interact only weakly, making them invisible to the detector; their presence is inferred indirectly through missing transverse energy. Muons, on the other hand, are minimally ionizing and typically traverse the calorimeters with little energy loss. To identify muons and measure their momenta, the outermost layer of the ATLAS detector is a dedicated tracking system known as the Muon Spectrometer. It surrounds the calorimeters and provides both precision momentum measurements and prompt muon triggering over the pseudorapidity range $|\eta| < 2.7$. It determines the transverse momentum p_T of muons from the curvature of their tracks in a large air-core toroidal magnetic field, generated by superconducting barrel and endcap coils. The system consists of precision detectors: monitored drift tubes (MDTs) and, in the forward region, cathode strip chambers (CSCs) combined with fast-timing detectors: resistive plate chambers (RPCs) in the barrel and thin gap chambers (TGCs) in the endcaps [19].

First-level muon triggers are implemented in hardware and combine information from both the muon spectrometer and the calorimeters to select events at a maximum rate of 100 kHz. These are followed by more refined software-based trigger stages that reduce the event rate to approximately 1 kHz on average [19].

The design of the Run 3 Inner Detector reflects a balance between high spatial resolution from silicon sensors and the additional functionality provided by the gaseous TRT. In the forthcoming HL-LHC era, the Inner Detector will be entirely replaced by the all-silicon Inner Tracker (ITk), which extends coverage to higher pseudorapidity and provides increased granularity and radiation hardness. In preparation for the HL-LHC, not only will the tracking detector be replaced, but the calorimeter and muon spectrometer systems will also undergo targeted upgrades to meet the demands of higher luminosity and radiation levels. Further details can be found in Ref.[11].

2.2. The HL-LHC Upgrade

At fixed $\sqrt{s}$, the event rate for a process with cross section σ scales linearly with the instantaneous luminosity $\mathcal{L}$:

$$\frac{dN}{dt} \equiv R = \mathcal{L}\,\sigma\,, \qquad N = \mathcal{L}_{\text{int}}\,\sigma\,,$$

with $[\mathcal{L}] = \text{cm}^{-2}\text{s}^{-1}$ and $[\sigma] = \text{cm}^2$ [20, P. 11]. Here, the integrated luminosity $\mathcal{L}_{\text{int}}$ is defined as the time integral of the instantaneous luminosity,

$$\mathcal{L}_{\text{int}} = \int \mathcal{L}(t)\,dt\,.$$

For observed yields, one may include acceptance and efficiency as $N_{\text{obs}} = \mathcal{L}_{\text{int}}\,\sigma\,A\,\varepsilon$.

High luminosity therefore serves two roles: it increases the expected yield for small-cross-section processes (via larger $\mathcal{L}$ and $\mathcal{L}_{\text{int}}$), and it reduces statistical uncertainties in precision measurements by providing larger datasets and richer control samples. The HL-LHC is a luminosity upgrade: it keeps $\sqrt{s}$ essentially fixed, while increasing the collision rate and the accumulated dataset.

The HL-LHC is a major upgrade of the LHC designed to increase the collider's data output by roughly an order of magnitude beyond Run 3. It will boost the instantaneous luminosity to a peak of about 7×10^{34} $\text{cm}^{-2}\text{s}^{-1}$, compared to approximately 2×10^{34} $\text{cm}^{-2}\text{s}^{-1}$ in Run 3 [21]. The average number of interactions per bunch crossing (pile-up) is expected to reach around 200 at the HL-LHC, in contrast to up to 60 during Run 3 [22]. Over its lifetime (Run 4 to Run 6, expected to run from around 2030 to 2040), the HL-LHC aims to collect a total integrated luminosity of about 3 000 fb^{-1}, roughly ten times the ~ 300 fb^{-1} expected by the end of Run 3. The HL-LHC is scheduled to begin operation after the third long shutdown (LS3, 2026–2029), with Run 4 starting around 2030 and continuing futher, as shown in Figure 2.3.

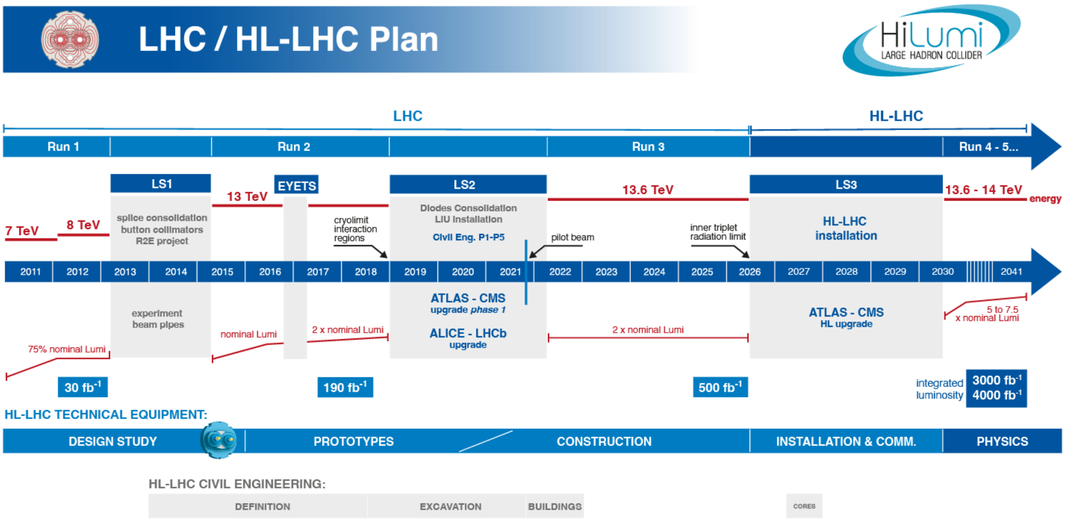


Figure 2.3.: Timeline of the LHC and HL-LHC programme from Run 1 to Run 5. The figure shows centre-of-mass energies, major shutdowns (LS1–LS3) with associated machine and detector upgrades, and the progressive increase in instantaneous luminosity from initial running to the HL-LHC goal of 5–7.5 times the nominal design value. The HL-LHC era (Run 4 and beyond) aims to deliver a total integrated luminosity of about 3 000–4 000 fb^{-1} by the late 2030s. Image © CERN.

The primary physics goal of the HL-LHC is to make full use of the large increase in collisions to improve the understanding of the Higgs boson and to search for rare processes and new phenomena.

Achieving these physics goals requires significant upgrades to the accelerator and experiments. The HL-LHC will run with much higher instantaneous and integrated luminosity, leading to many more proton–proton interactions per second. In Run 3, the LHC operates with an average pile-up μ of around 60 at peak. In contrast, the HL-LHC will run with average pile-up levels of about 140, and possibly up to 200 in the highest luminosity conditions. Such dense collision environments make event reconstruction more challenging: tracking detectors must separate hundreds of overlapping particle tracks and multiple interaction vertices within the same event.

To cope with this, the experiments are replacing or upgrading major detector systems during LS3. The inner tracking systems will be completely rebuilt using radiation-hard silicon detectors with higher granularity, to withstand the HL-LHC radiation dose and to resolve tracks in crowded events. A dedicated High-Granularity Timing Detector (HGTD) will be installed in front of the LAr end-cap calorimeters, covering $2.4 < |\eta| < 4.0$. Based on LGAD technology, it will provide per-track time resolution of $\mathcal{O}(30 \text{ ps})$ [23]. Trigger and data acquisition systems are being upgraded to handle much higher rates, with hardware and software able to process events at MHz-level input [24].

2.3. The Inner Tracker (ITk): Design and Components

The HL-LHC will present far harsher conditions than Run 3, such conditions would exceed the mechanical, electronic, and radiation-tolerance limits of the Run 3 ID. In particular, the limited pseudorapidity coverage ($|\eta| < 2.5$), the coarser granularity of the silicon sensors, the presence of the TRT which is unsuited for HL-LHC radiation levels, would suffer unmanageable occupancy at high pile-up and adds significantly to the material budget, further motivating an all-silicon solution.

Figure 2.4 shows a longitudinal (R-z) cutaway view of the ITk geometry. From top to bottom, the three concentric volumes correspond to the outer strip detector, the central pixel detector, and the inner strip detector. Along the z-axis, the central cylindrical section constitutes the central barrel region, while the outer ring-like structures on both sides are referred to as the endcaps.

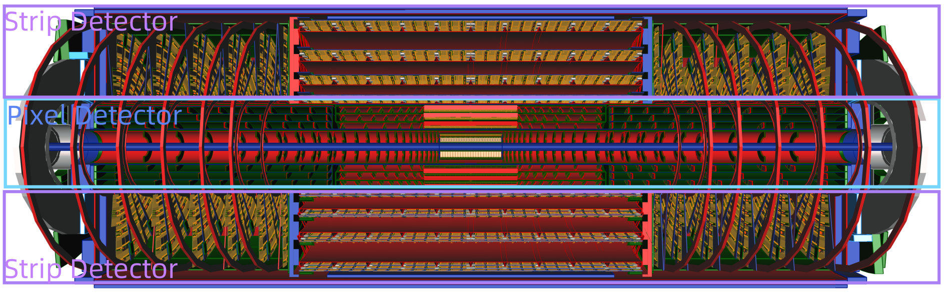
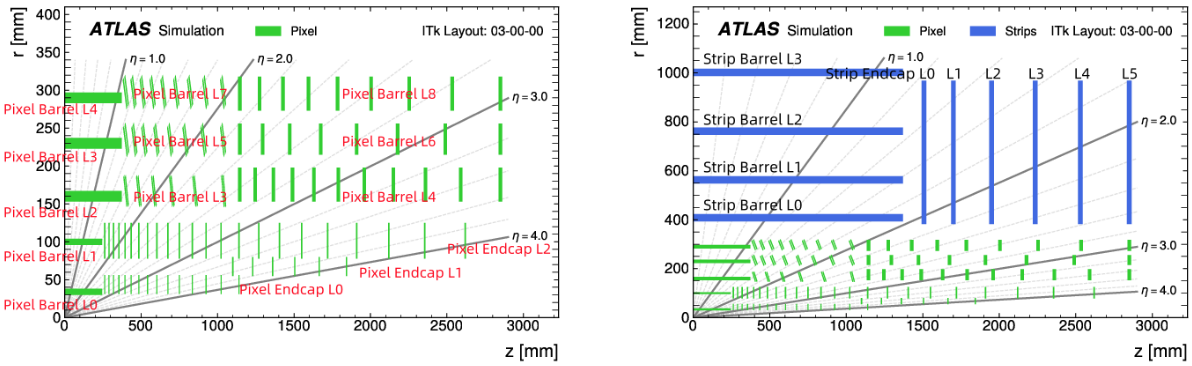


Figure 2.4.: Longitudinal-section of the ITk Showing the pixel and strip subdetectors. Figure adapted from Ref. [25].

The new all-silicon ITk overcomes these shortcomings with an expanded geometry, finer segmentation, and sensors and electronics qualified for HL-LHC radiation and occupancy. Its design targets include high granularity, low material budget, and extended tracking coverage to $|\eta| \approx 4.0$. After multiple design iterations and optimizations, the current ITk geometry has converged to the 03-00-00 layout. It is composed of an inner pixel detector and an outer strip detector. In the transverse direction, the central region adopts a barrel design, while the two ends employ endcap structures, arranged such that the incident particle trajectories are as perpendicular as possible to the sensor surfaces. The detailed layout and the corresponding indices used in the ATHENA geometry description are shown in Figure 2.5. All simulated samples presented in this work are produced using the 03-00-00 layout configuration.



(a) Pixel detector layout with internal ATHENA layer indices.

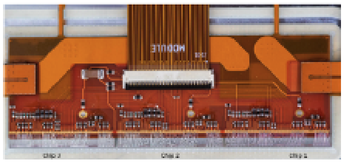
(b) Full ITk layout including both pixel and strip detectors, with internal ATHENA layer indices of strip detector.

Figure 2.5.: Longitudinal-sectional view of the ITk layout. Note that the layer indexing used here follows ATHENA's internal conventions, which differ from those in most public technical documents. Figures adapted from Ref. [26].

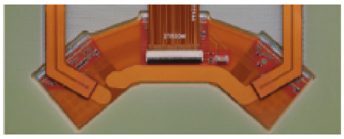
Compared to the Run 3 ID, the ITk pixel subsystem increases the number of barrel layers from four (including the IBL) to five, and replaces the small endcap disks with a set of forward pixel rings, extending precision tracking into the high-$|\eta|$ region. The five barrel pixel layers are implemented with a combination of flat and inclined module staves to maximize acceptance, and the new forward pixel rings provide coverage up to $|\eta| \sim 4$. The innermost pixel layers (barrel layer 0, endcap layer 0 and endcap layer 1) use 3D silicon pixel sensors for their superior radiation hardness [21, P. 6], while the remaining layers employ thin planar silicon sensors [27, p. 100]. Furthermore, the pixel cell size is reduced to $50 \times 50\ \mu\text{m}^2$ (with $25 \times 100\ \mu\text{m}^2$ in the innermost layer) from $50 \times 400\ \mu\text{m}^2$ in Run 3 [21, P. 5], greatly improving spatial resolution and reducing per-pixel occupancy at high pile-up.

All ITk pixel front-ends are read out by the RD53C chip, a pixel readout ASIC developed for the HL-LHC upgrades of both ATLAS and CMS using 65 nm CMOS technology. The ATLAS-specific instance of RD53C (designated ITkPix V2) is a system-on-chip consisting of a two-dimensional matrix of pixels and a digital control/readout region (the "chip bottom"). The pixel matrix is built from identical 8 × 8 pixel core blocks, each containing 64 pixel front-end channels with local digital logic for timing, configuration, and data processing [28, P. 4].

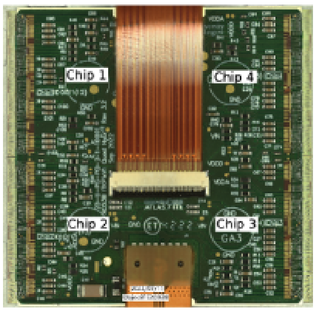
In the ITk pixel detector, a module refers to the fundamental assembly of sensor and electronics: a silicon sensor (either planar or 3D) bump-bonded to one or more RD53C ASICs and wire-bonded to a flexible PCB, forming a fully independent readout unit. The outer pixel layers use *quad modules*, where a single large planar sensor is read out by four RD53C chips arranged in a 2 × 2 array. Figures 2.6c and 2.6d show prototype quad modules intended for different regions of the pixel detector. The innermost pixel layers instead use triplet modules, each composed of three small 3D sensors grouped to provide complete coverage. A prototype triplet module used in Pixel Barrel Layer 0 is shown in Figure 2.6a, while Figure 2.6b shows the version used in Pixel Endcap Layers 0 and 1. All these modules are mounted onto lightweight carbon-fiber supports (staves in the barrel and rings in the endcap) that integrate cooling and data/power buses [21]. An innovative serial powering scheme is employed to deliver low-voltage power through chains of modules, significantly reducing the mass of cables. Efficient two-phase CO_2 evaporative cooling is used to remove heat and keep the sensors at around −35°C during the entire HL-LHC operation [21].



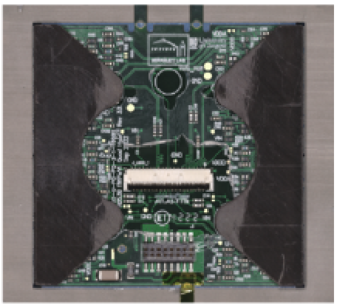
a) 3D triplet for the Inner System barrel stave



b) 3D triplet for the Inner System endcap ring



c) Inner System and Outer Endcaps quad module with data and power pigtails connected



d) Outer Barrel quad module with carbon fibre wire bond protection

Figure 2.6.: Different module types are used in the ITk pixel detector depending on the layer and sensor position: triplet modules are employed in Pixel Barrel Layer 0, Pixel Endcap Layer 0, and Layer 1, while all other layers use quad modules. Figure adapted from Ref. [29]

The ITk strip subsystem retains four barrel layers, but with optimized radii compared to the Run 3 design. In the endcaps, the SCT's nine disks per side are replaced by six larger disk wheels, simplifying the layout and reducing the number of support structures.

The barrel region of the ITk strip tracker employs two types of modules. The *short-strip* (SS) modules are segmented into four sensor regions, each with a strip length of approximately 24.2 mm. The *long-strip* (LS) modules have two sensor segments with strip lengths of 48.4 mm. Both SS and LS sensors share a common strip pitch of 75.5 μm. In the endcaps, six different trapezoidal sensor designs are used, with strip pitches ranging from 70 to 80 μm and strip lengths varying between 15.1 mm and 60.2 mm [30].

Each strip module is double-sided, consisting of two sensors glued back-to-back with a small stereo angle to enable three-dimensional space-point reconstruction. The modules are read out by custom ABCStar ASICs, with data aggregation and control performed by hybrid controller chips. All modules are mounted onto lightweight carbon-fiber support structures—staves in the barrel and wedge-shaped petals in the endcaps. These structures integrate cooling channels and service buses for power and data, thereby minimizing external cabling. Additionally, a neutron moderator surrounds the strip tracking volume to reduce radiation damage from secondary neutrons back-scattered from the calorimeters [26].

Mechanically, the ITk employs a unified support concept, combining staves, rings, longerons, and petals, to provide the necessary rigidity, thermal stability, and alignment precision with minimal mass. In the barrel, each layer's stave (carrying pixel or strip modules) is attached to lightweight yet stiff support rods called longerons that run along the detector length. In the endcaps, the pixel detector uses support rings to mount its forward modules, while the strip detector's endcaps are built as large wheels composed of multiple petal sectors, as shown in Figure 2.7. This integrated design leads to continuous overlap between barrel and endcap elements, improving detector hermeticity compared to the more separated layout of the Run 3 ID. All the support structures incorporate power, readout, and cooling services internally, avoiding bulky external cables and material. Thanks to the increased coverage and granularity, a typical track in the central region will traverse at least nine silicon hits (five pixel layers and four strip layers), and even tracks at high $|\eta|$ are measured with multiple high-precision hits.

In summary, the ITk replaces the mixed-technology Run 3 ID with an all-silicon, high-granularity tracker, expands acceptance into the forward region, improves spatial resolution, and incorporates radiation-hard sensors and electronics. These changes are driven by the need to maintain high tracking efficiency, low fake rates, and precise momentum and vertex measurements under the extreme occupancy, radiation, and data throughput of the HL-LHC.

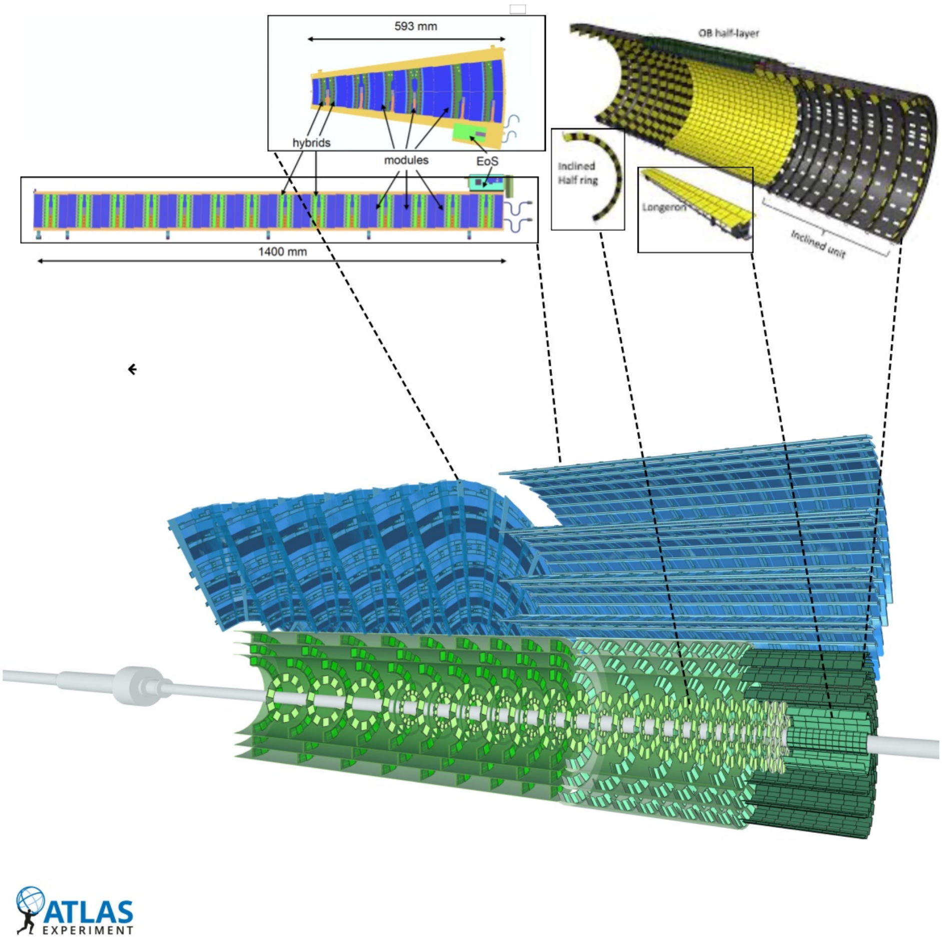


Figure 2.7.: Pixel and strip modules are mounted on mechanical support structures before being assembled into the full detector. In the ITk barrel region, the pixel inner system and strip detector use staves as local supports, while the pixel outer barrel employs longerons. The pixel endcap consists of concentric rings, and the strip endcap modules are supported by petal-shaped structures. Figures adapted from Ref. [30–32].

3. Simulation Setup and Tracking Reconstruction Chain

The motivation for modeling detector defects in this study arises from several practical considerations. First, the high pile-up environment expected at the HL-LHC will subject the detectors to significantly elevated radiation levels, leading to gradual degradation of detector components; this degradation is driven by effects such as increased leakage current and shifts in effective doping concentration. To account for such issues, the simulation tools employed in this study support the disabling of individual pixel unit. Second, the production yield of detector modules is inherently imperfect. For instance, during the ITk Pixel module production, certain readout chips that pass wafer-level testing have been observed to exhibit failures after assembly, such as spontaneous readout freezes or persistent output of erroneous data [33]. In some cases, these issues can be mitigated by disabling one or more internal core columns, each accounting for approximately 2% of the chip's pixels. Notably, this already exceeds the quality control requirement, which mandates that the total fraction of dead pixels per module must remain below 1%.

Additionally, as discussed in Ref.[12], the tight construction schedule of the ITk Pixel detector raises concerns about completing the full installation in time for LS3. To address this risk, the ATLAS Collaboration considered staged installation scenarios, in which only part of the pixel system would be installed during LS3 and the remaining modules added during LS4. The study investigates the impact of such partial installation on tracking performance, physics analyses, and computing resource requirements in Run-4. Two scenarios are explored: one where the entire Pixel outer endcaps are missing, and another where Pixel Layer 4 (barrel and endcap) is absent [12, P. 5]. To further evaluate robustness, I model configurations in which up to 30% of the pixel modules are disabled and examine how these defects can be arranged to minimize performance degradation.

This work aims to answer the following central question:

How does tracking performance change when parts of the Pixel or Strip detectors are disabled at various levels of granularity?

3.1. Modeling Detector Defects

The investigation is performed using the standard ATLAS simulation and reconstruction framework. Events are generated using tools such as PYTHIA, producing particle-level collision records (`EVNT`). These particles are then propagated through the full detector geometry using GEANT4, which simulates energy depositions and produces idealized detector responses (`HITS`). The digitization stage converts these responses into simulated raw readouts (`RDO`) by modeling the behavior of the detector electronics.

Subsequently, the ATHENA reconstruction framework processes the `RDO` files to recover tracks, clusters, and other high-level physics objects, resulting in datasets such as the Event Summary Data (`ESD`) and the Analysis Object Data (`AOD`). All reconstruction steps are executed via `Reco_tf.py`, the general-purpose ATLAS reconstruction driver, which orchestrates each stage of the offline processing chain based on the specified input and output format. The overall processing workflow is illustrated in Figure 3.1.

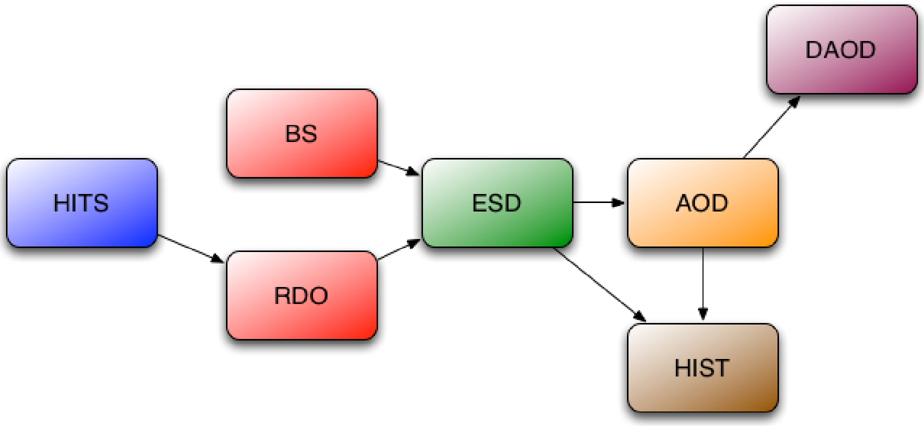


Figure 3.1.: Overview of the Reco_tf workflow, figure adapted from Ref. [34].

Within this framework, a simplified **masking-only simulation** method is implemented. A binary defect map is generated during reconstruction to label selected detector elements as either fully functional or entirely inactive. For the pixel detector, masking can be applied at the level of individual pixels, front-end core columns, or full modules. For the strip detector, masking is applied to individual strips or entire modules. The defect map is injected at the `RDO` stage by removing hits associated with masked elements, effectively simulating dead regions without altering the physics of charge deposition in the remaining active regions.

Although detector degradation due to radiation may seem like a natural motivation for masking-based simulations, the masking-only approach, while suitable for modeling binary production failures, does not reproduce the complex reality of radiation damage. In practice, sensor degradation is gradual rather than absolute. Radiation damage reduces the total collected charge in a cluster by introducing trapping centers in the silicon; it shrinks cluster size and alters its shape as weak peripheral signals fall below threshold; it degrades position resolution due to increased noise and reduced charge sharing; and it lowers hit efficiency even in nominally active modules. None of these effects are captured in the masking-only model, which assumes that all unmasked regions retain ideal, pre-irradiation performance.

Given these limitations, the goal of this study is not to replicate the physics of radiation damage. Instead, the masking-based strategy is better suited for modeling complete module-level failure, such as delayed installation scenarios where only a subset of modules is initially present [12], or for use as a stress test of the tracking reconstruction algorithms under pessimistic conditions. Framed this way, the study enables a systematic investigation into whether tracking performance degrades smoothly or undergoes a sharp transition as the fraction of disabled channels increases past a critical threshold. It also facilitates direct comparisons between two different reconstruction chains in the presence of localized inefficiencies. Ultimately, the work serves as a test of tracking robustness under highly adverse detector configurations.

3.2. Track Reconstruction Chains Used in Simulation

ATLAS track reconstruction has traditionally relied on a modular chain of algorithms developed and tuned for the Run 2 and Run 3 ID configurations. In simulation frameworks based on ATHENA, this reconstruction workflow is implemented through the default reconstruction chain. which combines silicon-based seeding, combinatorial Kalman filtering, and ambiguity resolution [15]. This reconstruction chain automatically adapts to both ID and ITk geometries.

To address the increased computational demands of the HL-LHC, a streamlined alternative known as the fast track reconstruction chain, has been developed specifically for the ITk geometry [14]. This strategy avoids ambiguity resolution entirely and uses only pixel-based seeds, while applying tighter selection criteria during track finding and a faster Kalman fit with simplified material treatment [15]. These modifications significantly reduce CPU usage with minimal loss in tracking performance.

The following subsections describe the default and fast tracking strategies in detail, highlighting their algorithmic differences and implications for high pile-up conditions at the HL-LHC.

3.2.1. The Default Track Reconstruction Chain: `newStrategy`

The default reconstruction chain begins with the formation of seeds from three space-points in the pixel or strip detectors that are roughly consistent with a charged-particle trajectory. These seeds define an initial track hypothesis, which is then extended using a combinatorial Kalman filter (CKF). The CKF incrementally adds compatible hits to the seed, following the estimated trajectory direction and curvature, to construct full track candidates. Only those candidates that pass a set of loose selection criteria are retained. Table 3.1 summarizes the minimum hit requirements used in this study; values in parentheses indicate the corresponding requirements in the default reconstruction chain. These requirements are independent of pseudorapidity.

Table 3.1.: Changes in tracking cuts used in Silicon Track Finding for fast and default (in brackets) track reconstruction, depending on the pseudorapidity interval. Here z_0 is defined with respect to the mean position of the beam spot. Table is adapted from [15, P. 6]

Requirement	**Pseudorapidity interval**		
	$\|\eta\| < 2.0$	$2.0 < \|\eta\| < 2.6$	$2.6 < \|\eta\| < 4.0$
Pixel+Strip hits	≥ 9 (7)	≥ 8 (7)	≥ 7 (7)
unique hits	≥ 7 (1)	≥ 6 (1)	≥ 5 (1)
shared hits	≤ 2 (no cut)	≤ 2 (no cut)	≤ 2 (no cut)
p_T [MeV]	> 1000 (900)	> 400 (400)	> 400 (400)
$\|z_0\|$ [cm]	≤ 15 (20)	≤ 15 (20)	≤ 15 (20)

The default track reconstruction chain follows a traditional, multi-stage approach designed to efficiently reconstruct charged-particle trajectories. It consists of two main phases: seeding and the subsequent ambiguity resolution, which includes both scoring and solving stages. A simplified overview of this process is shown in Figure 3.2. The process begins with clustering and space point formation, where raw electronic signals from the detector are converted into spatial hit positions. These space points serve as the basis for the subsequent formation of track seeds, constructed from hit triplets that are geometrically consistent with a single particle trajectory. These seeds are then extended into full track candidates by iteratively incorporating additional compatible hits. A preliminary selection is applied to discard unlikely candidates at an early stage. The remaining tracks are then scored based on their quality and internal consistency, and low-quality or duplicate candidates are removed. During this stage, the algorithm also computes the cluster-splitting probabilities for pixel clusters, which are later used to guide ambiguity resolution.

In the final phase, any remaining ambiguities, such as overlapping tracks that share the

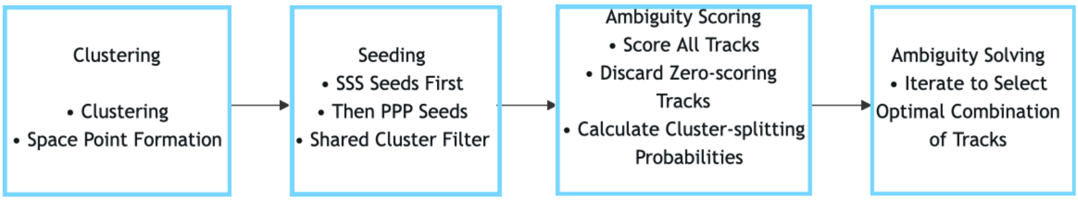


Figure 3.2.: Simplified overview of the default track reconstruction chain, highlighting the key stages from hit clustering to ambiguity resolution.

same detector hits, are resolved through an optimization process that selects the most consistent combination of tracks, taking into account potential hit-sharing and cluster-splitting scenarios. The reconstruction chain adopts a relatively permissive strategy during the early seeding stage, allowing multiple track candidates to originate from the same seed and postponing the resolution of shared clusters. These conflicts are systematically addressed during the dedicated ambiguity resolution stage. Technical details and module-level implementations of each step are provided in Appendix A and B.

To quantify the computational cost of the default reconstruction chain, Table 3.2 shows the CPU time required for each processing step in $t\bar{t}$ Monte Carlo events, comparing the ITk layout at $\langle\mu\rangle = 200$ with the current Run 2 Inner Detector at $\langle\mu\rangle = 20$. The timing was measured on an Intel Xeon 7210 processor (1.3 GHz, 64 cores, 116 GB RAM).

To ensure that the performance results are independent of the specific hardware used, the CPU time is normalized using the HEP-SPEC06 (HS06) benchmark [35]. HS06 is the standard metric for processor performance in the High Energy Physics community, derived from the industry-standard SPEC CPU2006 suite. Accordingly, the resource consumption is expressed in units of HS06·s, calculated by multiplying the raw execution time by a factor of 3.5, which corresponds to the measured HS06 performance score of a single core of the Xeon 7210 processor.

Table 3.2.: Estimated CPU time per event (in HS06 × seconds) for $t\bar{t}$ Monte Carlo events using the default reconstruction chain. Results are shown for the ITk layout at $\langle\mu\rangle = 200$ and the current Run 2 Inner Detector at $\langle\mu\rangle = 20$. The timing is measured on an Intel Xeon 7210 and scaled to HS06 units. Table is adapted from Ref. [15].

Detector	$\langle\mu\rangle$	**Cluster Finding**	**Space Points**	**SiTrack Finding**	**Ambiguity Resolution**	**TRT+Back Tracking**	**Other**	**Total**
ITk	200	22.0	6.5	78.0	97.0	—	15.5	219.0
Run-2	20	1.5	0.7	23.0	15.0	19.0	24.5	64.0

Although the `newStrategy` reconstruction chain is compatible with the ITk geometry, applying it under high pile-up conditions leads to a significant increase in CPU requirements. As shown in Table 3.2, the per-event tracking time at $\langle\mu\rangle = 200$ exceeds 200 HS06×s, which is substantially higher than the corresponding value for Run 2 conditions. This motivates the development of faster reconstruction strategies for HL-LHC.

3.2.2. Fast ITk Track Reconstruction: `itkFastTrackingStrategy`

Although the default reconstruction chain described above has been specifically optimized for the ITk geometry and HL-LHC conditions, and maintains excellent tracking performance while remaining compatible with the upgraded detector layout, a dedicated reconstruction chain, `itkFastTrackingStrategy`, has been developed to further improve computational efficiency under high pile-up. Compared to the strategies used in Run 2 and Run 3 with the ID, this configuration has been tailored to optimize tracking for events with an average pile-up ($\langle\mu\rangle$) of 140 to 200 interactions.

As shown in Table 3.2, the most CPU-intensive step of the default reconstruction chain under ITk geometry is Ambiguity Resolution. In the fast track reconstruction chain, this stage is entirely removed from the execution pipeline. Instead, its functionality is shifted upstream into the combinatorial track finding stage. The cuts on the number of hits and non-shared hits, which are typically applied during Ambiguity Resolution, are now enforced earlier during track finding. This design helps suppress ambiguous candidates from the outset, while also tightening the selection criteria applied to tracks, as shown in Table 3.1.

Another significant contributor to the CPU time is the seeding stage. In Run 2, seeding accounted for roughly 20% of the total tracking time, while road building and combinatorial filtering took the remaining 80%. Under ITk conditions with $\langle\mu\rangle = 200$, seeding alone consumes around 50% of the total CPU. The complexity of seeding grows substantially with pile-up, making it a major bottleneck. To reduce this load, the fast track reconstruction chain changes the seed formation strategy: only PPP-seeds are used, skipping the SSS-seed iteration entirely. Consequently, there is no need to run space-point formation on strip clusters, which further reduces CPU consumption. Reconstruction logs from this study confirm a substantial reduction in the number of seeds used by the fast track reconstruction chain, greatly lowering the combinatorial complexity.

For track parameter estimation, the fast Kalman filter is used directly within the combinatorial track finding algorithm. While it now incorporates precise cluster calibrations, the fast fit uses an approximate material model and simplified cluster corrections. As a result, the final fit is expected to yield a slight degradation in resolution compared to the full offline track fit used in the default reconstruction.

Table 3.3 presents the CPU performance of the fast ITk track reconstruction chain compared against the default reconstruction chain. The evaluation is based on $t\bar{t}$ Monte Carlo samples with pile-up conditions of $\langle\mu\rangle = 140$ and $\langle\mu\rangle = 200$. To ensure a fair comparison, all tests including those for the default ITk reconstruction were performed on the same machine.

Table 3.3.: The CPU required in HS06 × seconds to reconstruct $t\bar{t}$ Monte Carlo events with $\langle\mu\rangle = 140$ and 200 in the ITk. Listed are the results for the different reconstruction steps using the default and the fast ITk track reconstruction. An Intel Xeon E5-2620v2 was used with 2.1 GHz and six physical cores per CPU. The CPU time is multiplied with the HS06 factor of 17.8 for single thread running. Table is adapeted from Ref. [15, P. 7].

$\langle\mu\rangle$	Tracking	Byte Stream Decoding	Cluster Finding	Space Points	Si Track Finding	Ambiguity Resolution	Total ITk
140	default	$1.2^{(*)}$	17.1	6.0	41.1	58.2	123.6
	fast	$1.2^{(*)}$	4.5	0.9	12.4	-	19.0
200	default	$1.6^{(*)}$	26.3	8.6	85.8	92.0	214.3
	fast	$1.6^{(*)}$	6.3	1.2	22.6	-	31.7

($^{(*)}$) Scaled from Run-2.

The results indicate that the fast reconstruction chain is approximately 6.5 times faster than the default configuration at $\langle\mu\rangle = 140$, and 6.8 times faster at $\langle\mu\rangle = 200$. This significant difference is primarily due to the computational cost of the Silicon Track Finding and Ambiguity Resolution stages. In the default configuration, these two stages require approximately 99.3 HS06×seconds at $\langle\mu\rangle = 140$, and 177.8 HS06×seconds at $\langle\mu\rangle = 200$. In the fast reconstruction, the corresponding logic is merged and simplified, resulting in a reduced time of only about 12.4 HS06×seconds and 22.6 HS06×seconds for $\langle\mu\rangle = 140$ and 200, respectively.

This corresponds to an approximate 8-fold speed-up in the track finding stage alone. Moreover, the fast reconstruction is about 1.8 times faster at $\langle\mu\rangle = 140$ than at $\langle\mu\rangle = 200$, indicating that pile-up still has a measurable effect on reconstruction performance. Nonetheless, the overall improvement in computational efficiency remains substantial under both pile-up scenarios.

As discussed in Ref. [15], performance tests were carried out using two single muon samples ($p_T = 2\,\text{GeV}$ and $p_T = 100\,\text{GeV}$) and two $t\bar{t}$ samples with average pile-up values of $\langle\mu\rangle = 140$ and 200, all simulated under ideal detector geometry. The results show that the tracking efficiency of the fast reconstruction chain is slightly lower than that of the default chain across all tested samples. This difference is particularly visible in the forward region and in the barrel–endcap transition zone around $|\eta| \in [1, 2]$ [15, p. 9]. The original study attributes this to the preliminary nature of the tuning of the fast reconstruction software. As will be shown in Section 4.1, little has changed in this respect even after six years. On the other hand, the resolution widths of d_0 (σ_{d_0}) and z_0 (σ_{z_0}) remain comparable between the two reconstruction chains over the full η range in the single muon samples.

Overall, the fast reconstruction chain adopts the same classical tracking approach used in ATLAS during LHC Run 1 and Run 2, which is based on a combinatorial Kalman filter. However, it eliminates the most computationally expensive step: the ambiguity solving stage, and significantly reduces the cost of seeding. In exchange for a small loss in tracking efficiency, the reconstruction time is reduced by a factor of seven compared to the default ITk chain. Continued optimization of the fast reconstruction chain is expected, with the long-term goal of replacing the current default strategy for all data and simulation processing at the HL-LHC [36, P. 4].

Evaluation of Two Reconstruction Chains under Varying ITk Configuration

In this study, all samples were reconstructed twice under the same detector geometry and input configuration, each time using one of the two available track reconstruction chain: the default reconstruction chain (`newStrategy`) and the fast track reconstruction chain (`itkFastTrackingStrategy`). In the case of random masking, the same approach was applied; however, due to the stochastic nature of the module deactivation, the exact defect map cannot be reproduced or recorded. As a result, the two reconstruction runs under random masking correspond to different realizations of the underlying defect configuration.

Default Setting in Athena

Unless explicitly overridden by the flag:

```
--preExec ’flags.Tracking.doITkFastTracking=True’
```

`Reco_tf` will default to using `newStrategy`. To mitigate potential strategy-specific artifacts and allow for systematic comparison under identical defect conditions, the author ensured that all samples were reconstructed independently using both strategies.

In addition, the following η region definitions are adopted from the reconstruction log, and are used consistently throughout this thesis:

- Barrel: $|\eta| \in [0.0,\ 0.8)$

- Transition: $|\eta| \in [0.8,\ 1.6)$
- Endcap: $|\eta| \in [1.6,\ 2.5)$
- Forward: $|\eta| \in [2.5,\ 4.0)$

4. Tracking Performance Evaluation

This chapter presents tracking reconstruction results for the 250-event $t\bar{t}$ MC sample, comparing a baseline configuration (defined here as reconstruction on a perfect detector geometry without any injected defect map at the RDO or AOD stage) to scenarios where parts of the detector are disabled. The performance metrics include the number and distribution of reconstructed tracks, reconstructed parameters, tracking efficiency, fake rate, and resolution.

Section 4.1 compares baseline performance between two seeding strategies. Section 4.2 studies the impact of randomized pixel detector failures at various granularities as a robustness test. Section 4.3 investigates how to optimally distribute a significant number of non-functional modules in order to exploit the ITk detector geometry and redundancy, with the goal of minimizing overall tracking performance degradation under such adverse conditions. Both track reconstruction chains were tested in this study. Section 4.4 applies a similar approach to strip layers, identifying layers whose failure leads to unacceptable degradation. Anomalous behavior observed with ATHENA's default `newStrategy` prompted a re-simulation of all samples using `itkFastTrackingStrategy`.

The author also performed a cross-check using another 500-event $t\bar{t}$ MC sample under release 04-00-00 with the same defect configurations; for clarity and brevity, these results are presented in the Appendix H.

4.1. Baseline Tracking Performance with Different Track Reconstruction Chains

As described in Section 3.2, two track reconstruction chains are currently compatible with the ITk geometry. The first, referred to as `newStrategy` in the seeding stage code, is the default configuration in Athena and is compatible with both the ID and ITk geometries. The second, `itkFastTrackingStrategy`, is designed specifically for the ITk at the HL-LHC, it achieves a significant reduction in computational cost at the expense of a small degradation in tracking performance.

Due to the differing design priorities of these two reconstruction chains, their tracking performance exhibits noticeable variations across several metrics. Since all subsequent comparisons between defect configurations and the baseline are performed separately under both reconstruction chains, this chapter begins by comparing their respective baseline performances.

Two of the most important metrics for evaluating detector tracking performance are the tracking efficiency and the fake rate.

To assess whether a reconstructed track corresponds to a real particle, a key quantity is the truth match probability (TMP), which quantifies how closely a reconstructed track matches a specific truth particle. It is defined as [27]:

$$\mathrm{TMP} = \frac{2N_{\mathrm{pix}}^{\mathrm{common}} + N_{\mathrm{strip}}^{\mathrm{common}}}{2N_{\mathrm{pix}}^{\mathrm{track}} + N_{\mathrm{strip}}^{\mathrm{track}}}$$

where:

- $N_{\mathrm{pix}}^{\mathrm{common}}$ and $N_{\mathrm{strip}}^{\mathrm{common}}$ are the number of pixel/strip clusters shared between the reconstructed track and a given truth particle;

- $N_{\mathrm{pix}}^{\mathrm{track}}$ and $N_{\mathrm{strip}}^{\mathrm{track}}$ are the total number of clusters assigned to the track;

the relative weight of 2 for pixel hits accounts for the fact that pixel layers provide 2D measurements, whereas double-sided strip layers provide two 1D measurements.

A value of TMP $= 1.0$ indicates that all clusters on the track originate from the same truth particle. A value of 0.5 implies that only half the clusters are associated with the matched truth particle, while the rest originate from other particles. In this study, tracks with TMP< 0.5 are considered fake tracks.

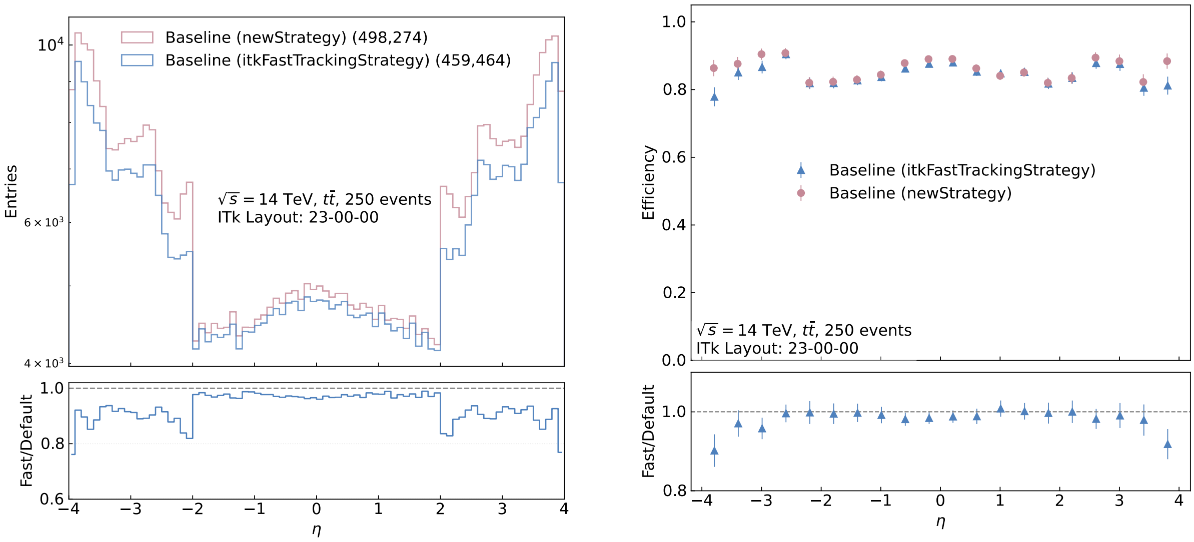
The tracking efficiency, $\varepsilon_{\mathrm{track}}$, is defined as the number of selected reconstructed tracks that are matched to a selected truth particle with TMP≥ 0.5, divided by the number of selected truth particles [27]:

$$\varepsilon_{\mathrm{track}} = \frac{N_{\mathrm{reco}}^{(\mathrm{selected,\, matched})}}{N_{\mathrm{truth}}^{(\mathrm{selected})}}$$

In other words, a truth particle is considered successfully reconstructed if there exists a reconstructed track with TMP≥ 0.5 and the track passes the defined quality cuts.

The fake rate, on the other hand, is not simply the complement of tracking efficiency. It is defined as the fraction of reconstructed tracks that do not correspond to any truth particle [27]:

$$\text{Fake Rate} = \frac{N_{\text{reco}}^{(\text{not matched})}}{N_{\text{reco}}^{(\text{total})}}$$



(a) Reconstructed track η distribution.

(b) Tracking efficiency vs. track η.

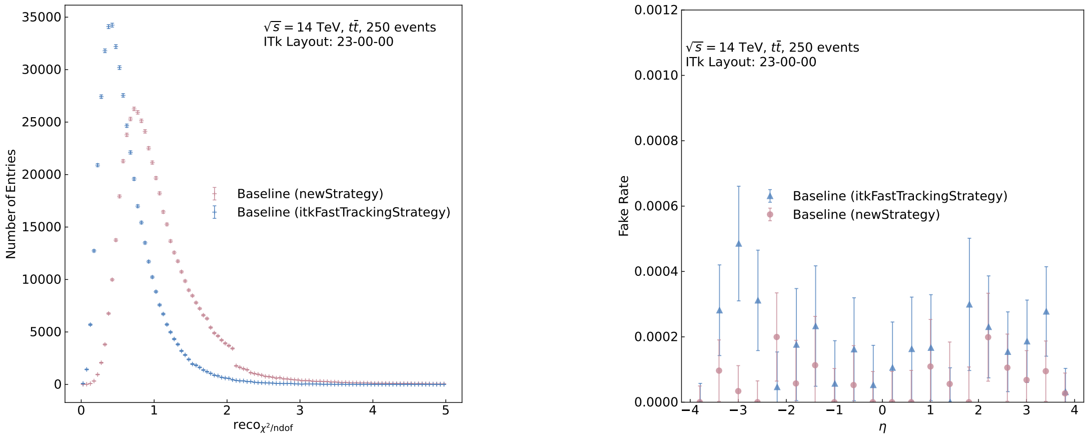
Figure 4.1.: Comparison of two seeding strategies in the baseline configuration: (a) reconstructed track η distribution, and (b) tracking efficiency as a function of η.

Figure 4.1a shows the η distribution of the number of reconstructed tracks for the same Monte Carlo sample under these two different seeding strategies. The bottom panel presents the bin-by-bin ratio of reconstructed tracks from the fast track reconstruction chain to those from the default reconstruction chain. Although the overall shape of the η distribution is similar, the default reconstruction chain yields a higher total number of reconstructed tracks (497,274 vs. 459,464), with the largest discrepancy observed in the forward region.

A similar trend is seen in the tracking efficiency vs. η distribution shown in Figure 4.1b. Overall, the default reconstruction chain provides slightly better tracking efficiency than the fast track reconstruction chain. However, this advantage is marginal in the region $\eta \in [0, 1]$, and nearly disappears in $\eta \in [1, 2]$. The most pronounced difference is observed in the forward region, in agreement with simulation results reported in Ref. [15].

Figure 4.2a compares the χ^2/N_{dof} distributions of baseline tracks reconstructed with the two strategies. The difference in peak positions reflects the use of different fitters. The mean and standard deviation of the distributions are 1.08 ± 0.56 for the default reconstruction chain and 0.64 ± 0.39 for the fast track reconstruction chain.

As shown in Figure 4.2b, both strategies achieve low fake rates (below 0.06%). The fake rate difference in the central barrel region is negligible, but the fast reconstruction chain tends to produce slightly higher fake rates in the forward region.



(a) Track χ^2/N_{dof} distribution.

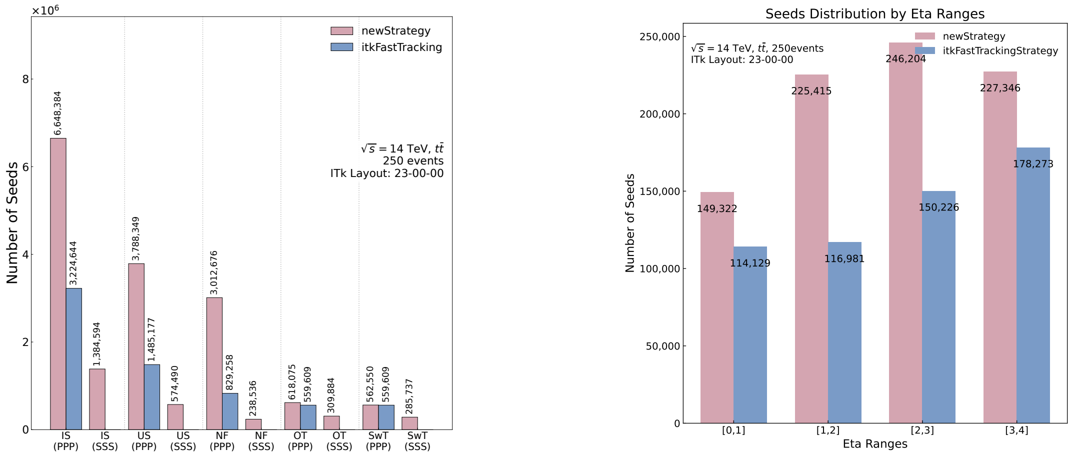
(b) Fake rate vs. η.

Figure 4.2.: Comparison of two seeding strategies in the baseline configuration: (a) track fit quality via χ^2/N_{dof}, and (b) fake rate as a function of η.

Additional insights into the seeding statistics can be drawn from the track reconstruction logs, visualized in Figure 4.3 and detailed in Appendix F. According to Table 4.1 and Figure 4.3a, the default reconstruction chain utilized approximately 2.49 times more input seeds than the fast track reconstruction chain, with the number of PPP seeds alone being 2.06 times higher. More significantly, the number of used seeds was 2.94 times higher. However, a large fraction of the default reconstruction chain's seeds did not lead to valid tracks, in fact, the output track count from it was 618,045, while the number of seeds associated with tracks was 562,550, indicating 55,525 extra tracks. This phenomenon is attributed to the one-to-many mapping between triplet seeds and output tracks in the default reconstruction chain, whereas the fast reconstruction chain's more aggressive cluster sharing suppression avoids this.

Table 4.1.: Comparison of seed input and usage between the default reconstruction chain and the fast track reconstruction chain, including total and PPP-only categories.

Algorithm	**Input Seeds**		**Used Seeds**		**Cannot Find Track**		**Seeds with Track**	
	Total	**PPP**	**Total**	**PPP**	**Total**	**PPP**	**Total**	**PPP**
The default reconstruction chain	8,032,978	6,648,384	4,362,839	3,788,349	3,251,212	3,012,676	848,287	562,550
The fast reconstruction chain	3,224,644	3,224,644	1,485,177	1,485,177	829,258	829,258	559,609	559,609
Ratio (default/fast)	**2.49**	**2.06**	**2.94**	**2.55**	**3.92**	**3.63**	**1.52**	**1.01**



(a) Seeds with tracks vs. η for both strategies. ***(b)*** Seed usage breakdown at different stages.

Figure 4.3.: Comparison of seeding behavior between the default reconstruction chain and the fast track reconstruction chain. (a) Summary of seed processing statistics for PPP and SSS pixel seeds. The labels correspond to Input Seeds (IS), Used Seeds (US), Can not find Track(NF), Output Tracks (OT), and Seeds with Track (SwT). The fast track reconstruction chain enforces a one-to-one seed-to-track mapping, resulting in zero extra tracks. (b) The η distribution of seeds that result in tracks.

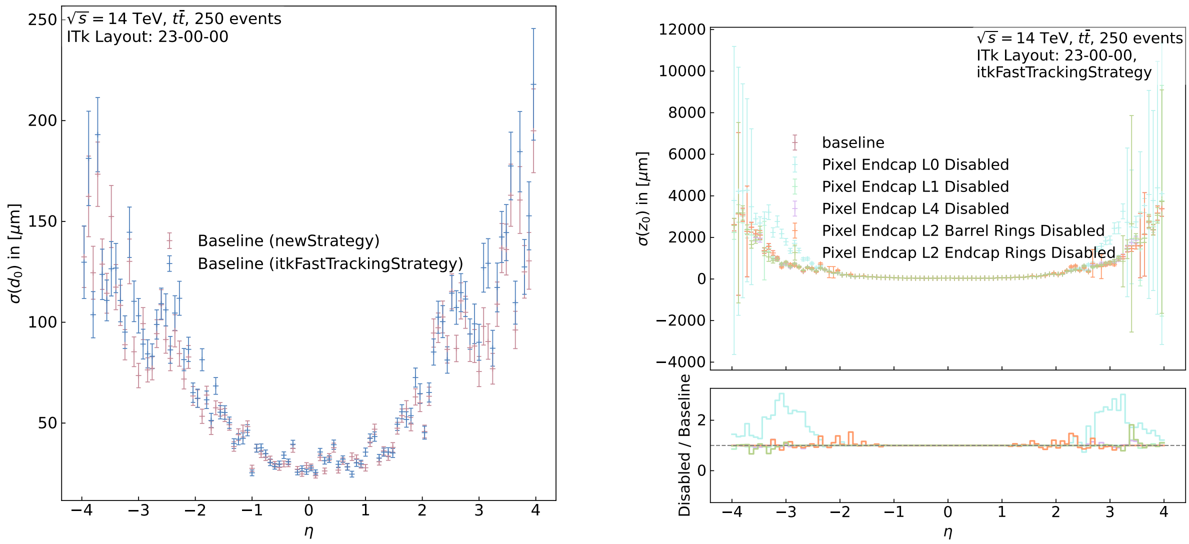
Figure 4.3b presents the binned counts of seeds with tracks across four $|\eta|$ intervals. While the default reconstruction chain shows the highest concentration in the range $\eta \in [2, 3]$, the fast track reconstruction chain's distribution peaks in the forward region. Despite having 1.93 times more seeds with tracks in the range $\eta \in [1, 2]$, the default reconstruction chain only slightly outperforms in the number of reconstructed tracks, as shown in Figure 4.1a, and the tracking efficiency in this region is nearly identical, as shown in Figure 4.1b.

It is also worth noting that even in the forward region, where no strip seeds are used, and thus both reconstruction chains rely exclusively on PPP seeds, the default reconstruction chain still produces more seeds associated with tracks than the fast reconstruction chain. This η region does not exhibit the largest difference in seed counts between the two reconstruction chains compared to the others, the largest seed count difference appears in the region $|\eta| \in [1, 2]$, which is also where the final difference in reconstructed tracks is the smallest.

One possible explanation lies in the cluster sharing logic used to resolve ambiguities. The fast track reconstruction chain employs the `filterSharedTracksFast` algorithm, which iterates through candidate tracks sorted by quality. Crucially, it marks the clusters associated with a candidate track as 'used' in the global exclusion set before the track actually passes the final quality cuts. If the candidate track subsequently fails these cuts and is discarded, its clusters are not removed from the exclusion set. Consequently, these clusters remain unavailable to subsequent, lower-quality track candidates, even though the track that originally claimed them was rejected.

This effect may be particularly pronounced in the forward region, where the number of available layers is reduced, hit sharing is more common, and the consequences of cluster overclaiming are more severe. Despite the aggressive filtering, the fake rate of the fast track reconstruction chain remains slightly higher than that of the default reconstruction chain, although still well controlled overall.

The width of resolution for d_0 and z_0 show no significant difference between the two track reconstruction chains across the full η range, as shown in Figure 4.4, consistent with the simulation results reported in Ref. [15, P. 10].



(a) Resolution width of d_0 vs η.

(b) Resolution width of z_0 vs η.

Figure 4.4.: Track parameter resolution widths for d_0 and z_0 vs η. The resolution is extracted from the width of the core Gaussian component of the residual distributions.

In summary, based on this sample and purely from a reconstruction quality perspective, the default chain appears to offer superior performance. However, this comes at a significantly higher computational cost, particularly during the seeding phase in the $\eta \in [1, 2]$ region. In contrast, the fast track reconstruction chain achieves a track-to-used-seed ratio that is over five times higher, indicating much more efficient seed usage. If tracking performance is the highest priority, the default reconstruction chain may be preferred. On the other hand, considering the high event rates and data throughput constraints at the

HL-LHC, the fast reconstruction chain delivers excellent efficiency in the central region and could benefit from targeted optimizations in the forward region.

An alternative approach worth considering is region-based reconstruction strategy. In the forward region, where track density is higher, the aggressive shared-cluster filtering strategy of the fast tracking chain may lead to performance degradation. The default reconstruction chain exhibits relatively better seeding efficiency in this region compared to its performance in other η intervals. This suggests that selectively employing the default reconstruction chain in the forward region, where computational overhead may be acceptable, could offer a viable compromise between reconstruction performance and resource constraints. Alternatively, the conservative cluster-handling strategy used in the default chain may itself offer valuable insights for improving forward-region reconstruction.

4.2. Tracking Performance with Randomized Pixel Detector Failures

The baseline tracking performance for both reconstruction chains discussed above is evaluated on an ideal, defect-free detector. However, in reality, non-negligible module failure rates have been observed during the ITk pixel production process. For instance, bump-bonding issues at the pixel cell level can lead to disconnected channels, affecting detector performance [33].

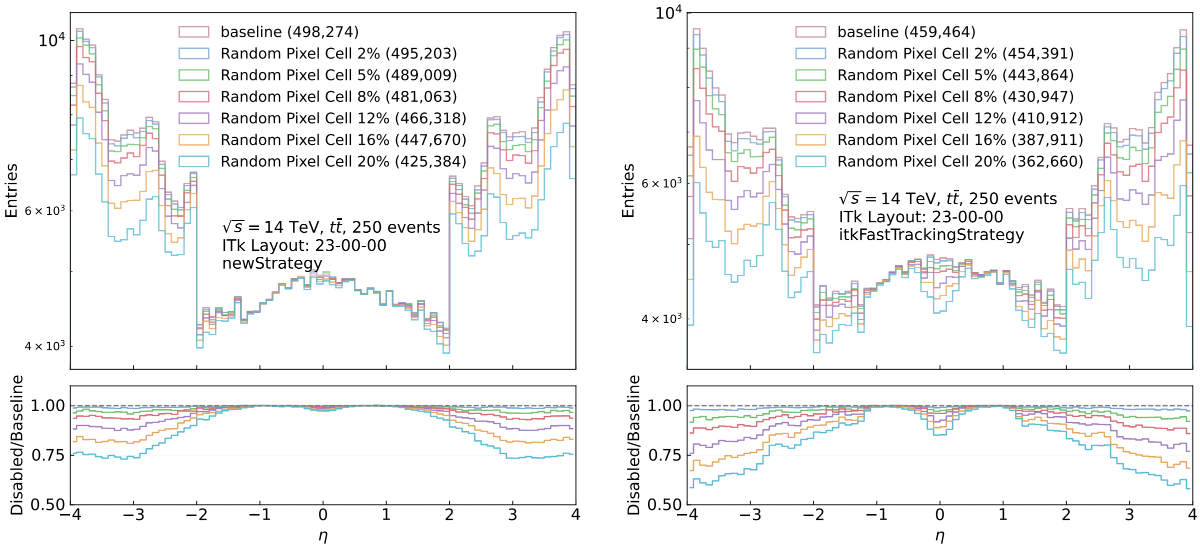
As a first step toward understanding the impact of such imperfections, a series of randomized pixel cell masking tests were performed. In this study, random masking fractions of 2%, 5%, 8%, 12%, 16%, and 20% were applied uniformly across the detector, and the resulting tracking performance was evaluated under both reconstruction chains.

While such high levels of random pixel cell failures are unlikely to occur in realistic operating conditions, these tests provide a useful benchmark for assessing the robustness of the reconstruction algorithms under pessimistic failure scenarios.

4.2.1. Random Masking at the Pixel Cell Level

As expected, the default reconstruction chain is significantly more robust than the fast reconstruction chain under random pixel cell masking, due to fundamental differences in their design. Using the default reconstruction chain, even when 20% of the pixel cells are randomly disabled, the number of reconstructed tracks remains largely stable within $|\eta| < 2$. Most of the degradation is concentrated in the forward region, where there is no redundancy from the strip detector, as shown in Figure 4.5a, in this region, the total number of reconstructed tracks decreases to about 75% of the baseline. Notably, the ratio between the masked and baseline configurations remains nearly constant as a function of η in the forward region.

In the fast track reconstruction chain, the performance progressively deteriorates across the region $|\eta| \in [1, 4]$, with the number of reconstructed tracks dropping to around 60% of the baseline in the highest η bins when 20% of the pixel cells are masked. By contrast, when 20% of the pixel cells are randomly disabled, the track count in the central region ($|\eta| \approx 0$) remains nearly unchanged With the default reconstruction chain, but drops to about 80% of the baseline under the fast reconstruction chain, as shown in Figure 4.5b.



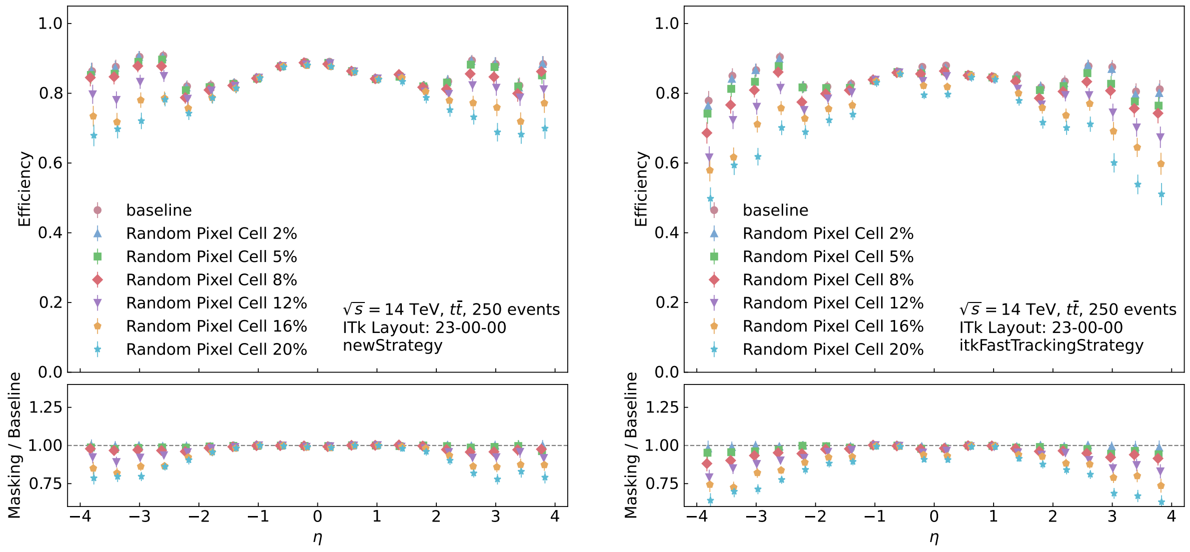
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.5.: η distributions of reconstructed tracks under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

In terms of tracking efficiency, the forward region suffers the most under both reconstruction chains. When 20% of the pixel cells are randomly disabled, the efficiency drops to approximately 75% of the baseline for the default chain, and to about 60% for the fast track reconstruction chain, as illustrated in Figures 4.6a and 4.6b, respectively.

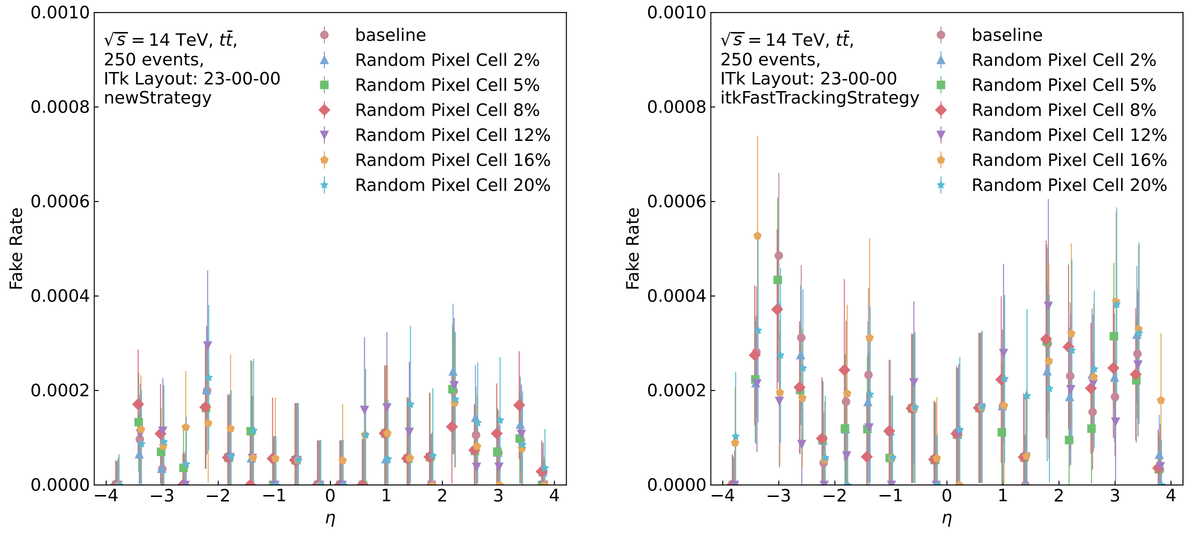
Despite these losses, the fake rate remains relatively stable. With the default reconstruction chain, the overall fake rate is lower and remains below 0.04% even with 20% of pixel cells randomly disabled, as shown in Figure 4.7a. For the fast track reconstruction chain, the fake rate in the forward region stays below 0.06% under the same defect level, as illustrated in Figure 4.7b.

For the default reconstruction chain, the peak of the χ^2/N_{dof} distribution remains rel-



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

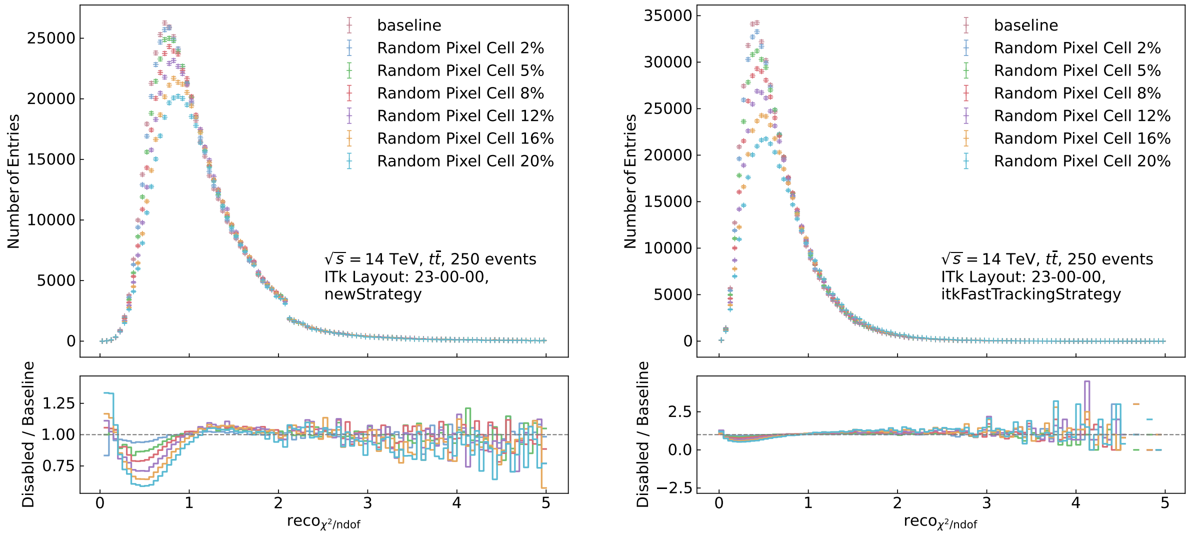
Figure 4.6.: Tracking efficiency in η bins under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.7.: Fake rate in η bins under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

atively stable, shifting only slightly from 1.079 to 1.153, a 6.9% increase under 20% pixel module masking, as shown in Figure 4.8a. In contrast, the fast reconstruction chain exhibits a more pronounced shift, with the peak moving from 0.6408 to 0.7435, corresponding to a 16.0% increase, as illustrated in Figure 4.8b.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

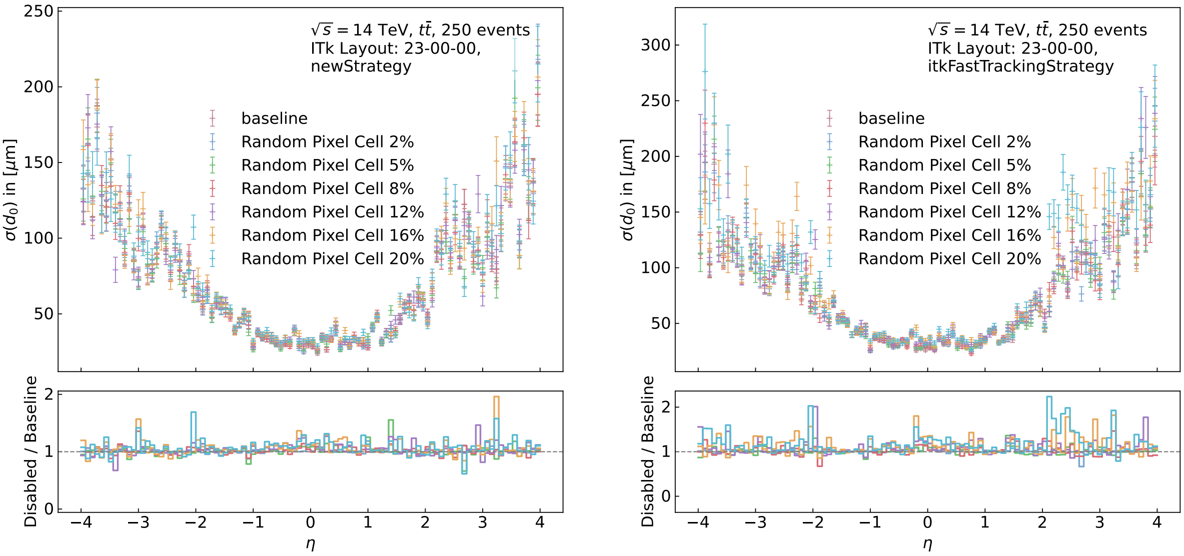
Figure 4.8.: Distribution of χ^2/N_{dof} under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

The width of the d_0 and z_0 resolution distributions increases with pixel masking in both reconstruction chains. The relative changes compared to baseline are of similar magnitude in each chain, indicating that both algorithms respond comparably in terms of spatial resolution when facing increasing levels of pixel degradation, as shown in Figure 4.9 and 4.10.

Table 4.2 summarizes the seeding stage behavior with the default reconstruction chain, comparing the baseline with configurations in which different fractions of pixel cells are randomly disabled. Random masking leads to a slight reduction in the number of input seeds, used seeds, and seeds associated with tracks.

In the default reconstruction chain, the seeding algorithm allows a single triplet seed to produce multiple track candidates. However, since each seed is ultimately expected to correspond to at most one physical track, only one candidate can be retained, while the others are recorded as "extra tracks" in the reconstruction summary. These are later resolved during the dedicated ambiguity solving stage.

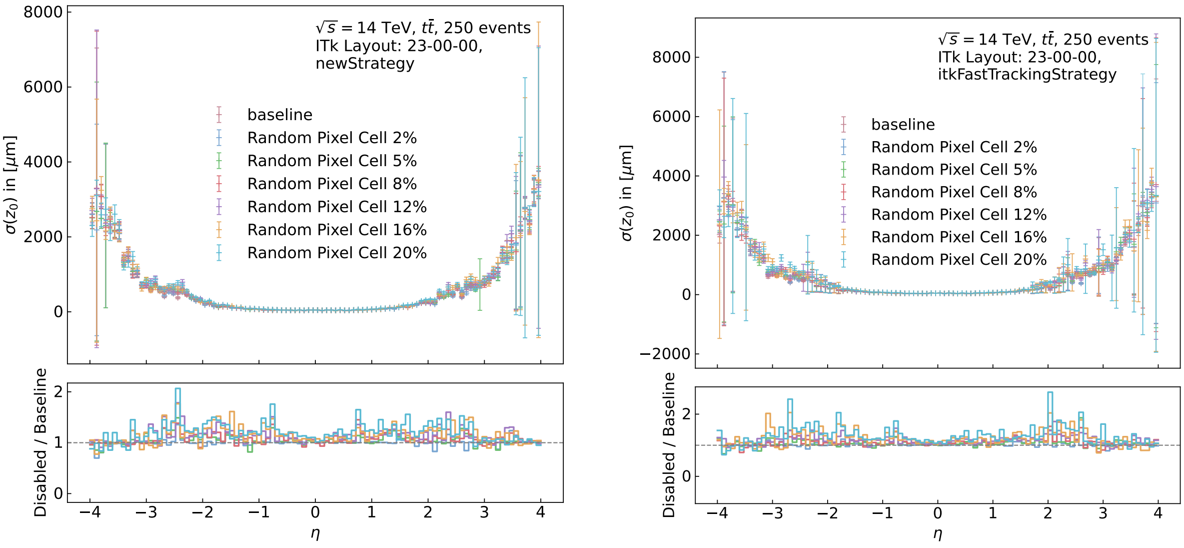
As the pixel cell masking fraction increases, the number of input seeds decreases from



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

Figure 4.9.: Distribution of d_0 resolution width under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

Figure 4.10.: Distribution of z_0 resolution width under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

6,648,384 in the baseline configuration to 6,108,547 under 20% masking, corresponding to a reduction of approximately 8.1%. Meanwhile, the number of extra tracks grows significantly: when 20% of the pixel modules are disabled, the number of extra tracks from PPP seeds increases by a factor of approximately 2.22 (from 55,525 to 123,272), while for SSS seeds the increase is even more pronounced, reaching a factor of 5.25 (from 24,147 to 126,880).

Despite this growth, the majority of these additional candidates are rejected during the ambiguity solving stage and do not appear in the final reconstructed track collection.

Table 4.2.: Summary of seed usage under the default chain with different fractions of randomly disabled pixel cells. Values that increase compared to the baseline are highlighted in magenta, values that decrease are highlighted in blue, and values identical to the baseline are shown in dark sepia..

	Baseline	**2% disabled**	**5% disabled**	**8% disabled**	**12% disabled**	**16% disabled**	**20% disabled**
Input seeds (PPP)	6,648,384	6,626,407	6,579,177	6,519,131	6,415,262	6,276,558	6,108,547
Input seeds (SSS)	1,384,594	1,384,594	1,384,594	1,384,594	1,384,594	1,384,594	1,384,594
Used seeds (PPP)	3,788,349	3,798,890	3,806,778	3,808,593	3,799,132	3,770,036	3,728,108
Used seeds (SSS)	574,490	574,445	574,504	574,489	574,589	574,445	574,610
Seeds with tracks (PPP)	562,550	560,798	557,201	553,787	546,232	536,428	525,362
Seeds with tracks (SSS)	285,737	285,704	285,697	285,537	285,440	285,462	285,335
Output tracks (PPP)	618,075	625,285	634,501	642,156	648,707	650,306	648,634
Output tracks (SSS)	309,884	323,425	342,062	359,177	379,942	397,578	412,215
Extra tracks (PPP)	55,525	64,487	77,300	88,369	102,475	113,878	123,272
Extra tracks (SSS)	24,147	37,721	56,365	73,640	94,502	112,116	126,880

Table 4.3 summarizes the seed usage under different pixel cell-level masking configurations for the fast track reconstruction chain. As this reconstruction strategy uses only PPP seeds, and its internal seed ambiguity resolution enforces a one-to-one correspondence between each triplet seed and a single track candidate, the number of extra tracks is always zero. Consequently, the number of output tracks is equal to the number of seeds with tracks.

As in the case of the default reconstruction chain, the number of input seeds decreases as the masking fraction increases. With 20% of the pixel cells randomly disabled, the number of input seeds drops by approximately 12.5% compared to the baseline (from 3,224,644 to 2,821,455). Interestingly, the number of used PPP seeds does not decrease; instead, it slightly increases for several configurations. However, because the fast track reconstruction chain does not include a dedicated ambiguity solving stage, and applies relatively strict selection cuts early in the process, the number of final output tracks remains below the baseline across all masking levels. This behavior differs from that of the default reconstruction chain.

When comparing the output tracks from the two reconstruction chains, there seems to be a clear difference: with the default reconstruction chain, the number of output tracks

Table 4.3.: Summary of PPP seed usage with the fast track reconstruction chain with different fractions of randomly disabled pixel cells. Values that decrease compared to the baseline are highlighted in blue, and values that increase are highlighted in magenta.

	Baseline	**2% disabled**	**5% disabled**	**8% disabled**	**12% disabled**	**16% disabled**	**20% disabled**
Input seeds (PPP)	3,224,644	3,197,850	3,153,612	3,098,995	3,022,076	2,928,911	2,821,455
Used seeds (PPP)	1,485,177	1,506,433	1,544,758	1,584,843	1,641,090	1,686,305	1,717,017
Seeds with tracks (PPP)	559,609	556,464	548,270	537,696	519,648	497,406	471,569

increases relative to the baseline as more pixel cells are randomly masked (up to 20%), whereas with the fast track reconstruction chain, the number of output tracks decreases. However, this discrepancy is largely due to differences in how track candidates are handled.

In the fast reconstruction chain, each seed can produce at most one track, so the number of output tracks directly follows the number of seeds with tracks. In contrast, the default chain allows multiple track candidates per seed before ambiguity resolution, which can inflate the output track count even when fewer seeds succeed. The apparent increase in tracks is thus not a sign of improved reconstruction, but a byproduct of extra tracks being temporarily retained.

4.2.2. Random Masking at the Pixel Module Level

A pixel module can be regarded as a collection of pixel cells. For example, the ITkPix V2 FE chip features a matrix of 384 × 400 pixel cells, and a quad module consists of four such chips assembled together [27].

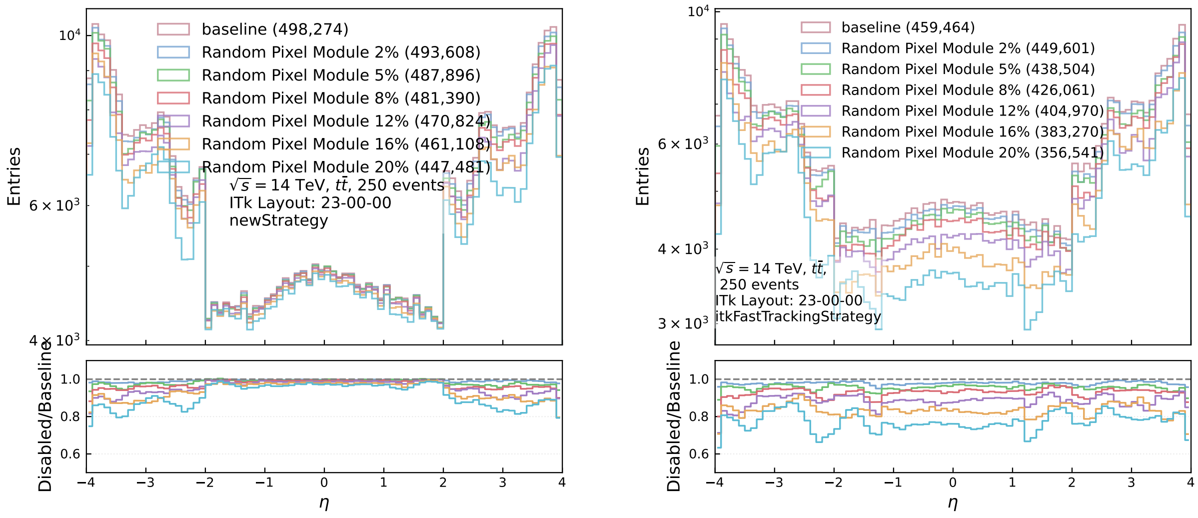
In a large-scale pixel detector, the total area affected by randomly disabling 20% of pixel cells is, in principle, similar to that affected by disabling 20% of pixel modules. However, module-level masking should introduce stronger spatial correlations, as entire regions are removed, potentially leading to localized inefficiencies.

This section investigates the impact of randomly disabling 2%, 5%, 8%, 12%, 16%, and 20% of pixel modules on tracking efficiency, under both reconstruction chains. It also compares the results to those obtained from pixel cell-level masking, to assess whether spatial correlation at the module level causes a significantly different degradation in tracking performance when the masked area is approximately the same.

The overall trend of the module-level number of reconstructed tracks under random defect configurations indicates that the default reconstruction chain remains largely unaffected in the barrel and transition regions ($|\eta| < 2$), where sufficient SSS seeds are still available. As shown in Figure 4.11a, the forward region suffers the most significant impact, as SSS seeding is no longer available in that regime.

Figure 4.11b shows that, for the fast reconstruction chain, both the low-$|\eta|$ region and the forward region experience a reduction in the number of reconstructed tracks. Several η bins exhibit noticeable fluctuations, likely corresponding to geometric transition zones in the detector,such as the interface between the outermost pixel barrel layer and the first endcap ring, or the connection between the pixel inner system and the outer endcap.

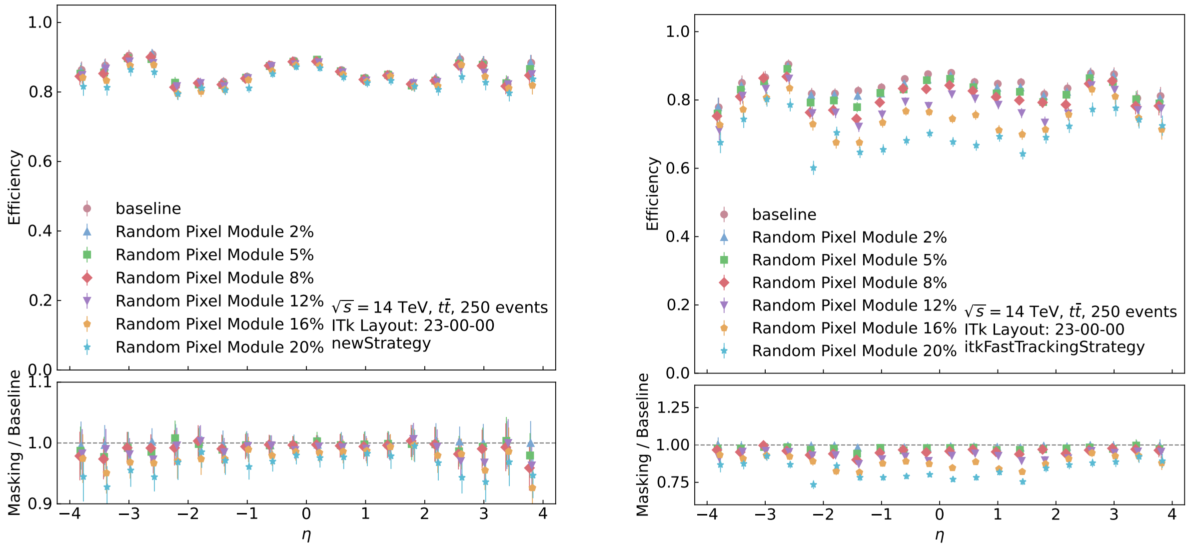
Overall, with 20% of modules randomly disabled, the default chain sees at most a 5% reduction in the number of reconstructed tracks in the barrel and transition regions, and a 15–25% reduction in the forward region. By comparison, the fast chain fluctuates around 70–80% of the baseline across the entire η range.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.11.: η distributions of reconstructed tracks under increasing levels of random pixel module masking: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

For the default reconstruction chain, the behavior of the tracking efficiency closely follows that of the number of reconstructed tracks: the loss is smallest in the low-η region. When 20% of the modules are randomly disabled, the tracking efficiency remains above 90% of the baseline even in the forward region, as shown in Figure 4.12a. In contrast, for the fast track reconstruction chain, the efficiency decreases more rapidly in the low-η region than in the forward region. As shown in Figure 4.12b, the tracking efficiency in the low-η region drops to about 75% of the baseline under 20% module masking. This difference can be attributed to the reconstruction thresholds: in the default chain, the combined pixel+strip hit requirement is uniformly set to 7 across all η, whereas in the fast track reconstruction chain, the corresponding cut is more stringent in the low-η region, as summarized in Table 3.1.



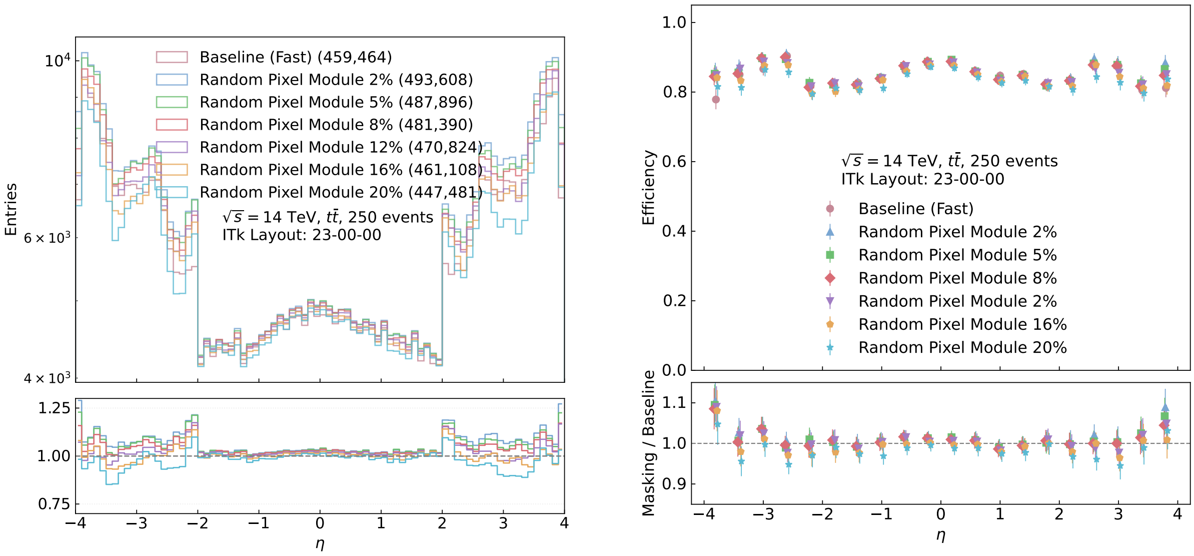
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.12.: Tracking efficiency in η bins under increasing levels of random pixel module masking: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

It should be noted that the baseline performance of the two reconstruction chains already differs in both the number and quality of reconstructed tracks. Figure 4.13 illustrates this difference by directly comparing the results obtained when modules are randomly disabled with the default reconstruction chain with baseline of the fast track reconstruction chain. As shown, even with 12% of the modules randomly disabled, the number of reconstructed tracks in the default chain remains higher than the baseline of the Fast Track chain (470,824 vs. 459,464). This advantage is visible across the entire η range. In the region $|\eta| < 2$, the number of reconstructed tracks with the default reconstruction chain remains comparable to the baseline performance of the fast reconstruction chain even when 20% of the pixel modules are randomly disabled. In the forward region, the default chain still outperforms the fast reconstruction baseline even with 12% of the pixel modules disabled.

For tracking efficiency, the same pattern is observed. In the very forward region, the default reconstruction chain maintains an efficiency above the fast track baseline even when 20% of the modules are randomly disabled.

Overall, the fake rate is well controlled for both reconstruction chains. At baseline and under low masking probabilities, the default reconstruction chain yields a slightly lower fake rate compared to the fast track reconstruction chain. However, when 20% of the pixel modules are randomly masked, the fake rate with the default reconstruction chain increases, becoming comparable to that of the fast chain. Even with 20% of the modules

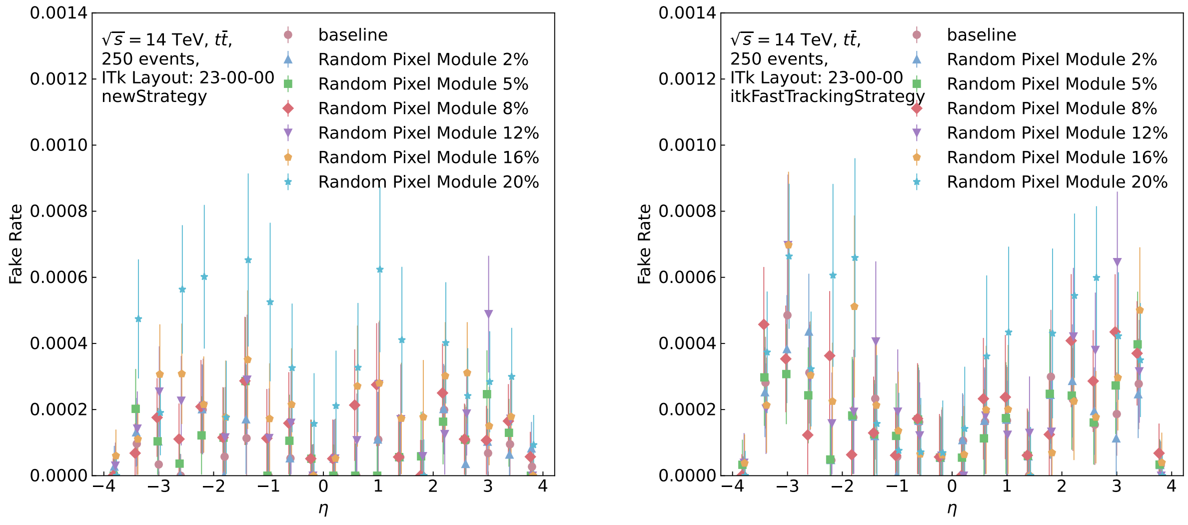


(a) Number of reconstructed tracks. ***(b)*** Tracking efficiency.

Figure 4.13.: Comparison of (a) the number of reconstructed tracks and (b) the tracking efficiency between the baseline of the fast track reconstruction chain and the default reconstruction chain with randomly disabled pixel modules.

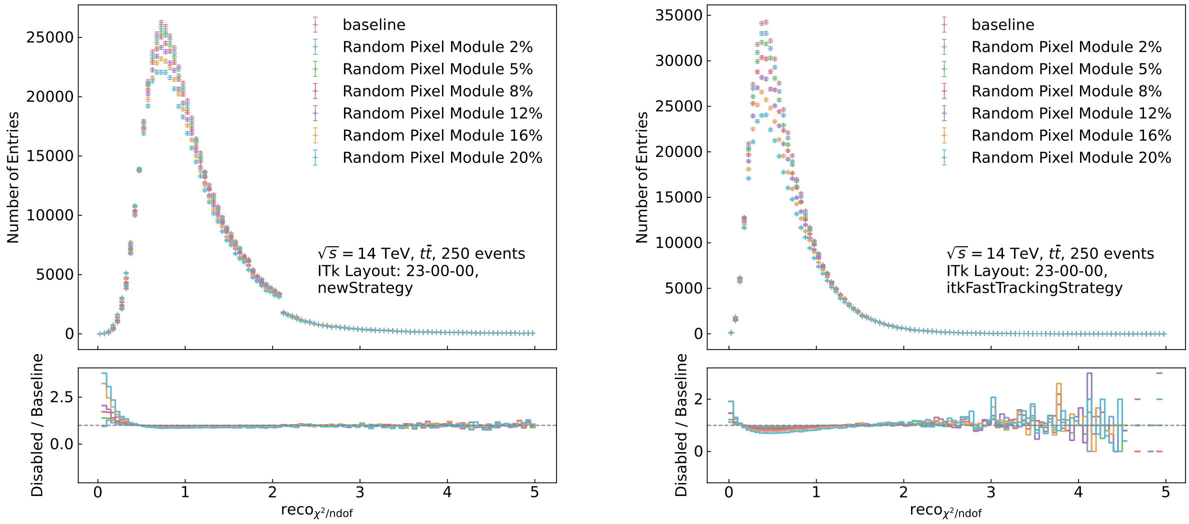
disabled, no sharp increase is observed; instead, the fake rate remains below 0.08%, as shown in Figure 4.14.

The default reconstruction chain demonstrates robustness against module loss. As illustrated in Figure 4.15a, the χ^2/N_{dof} distribution remains largely unchanged when 20% of the pixel modules are randomly disabled, with the mean decreasing slightly from 1.079 to 1.070. In contrast, the fast track reconstruction chain exhibits a slight degradation under the same conditions. As shown in Figure 4.15b, the peak of the distribution shifts rightward from a baseline of 0.641 to 0.667, representing an increase of approximately 4%.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

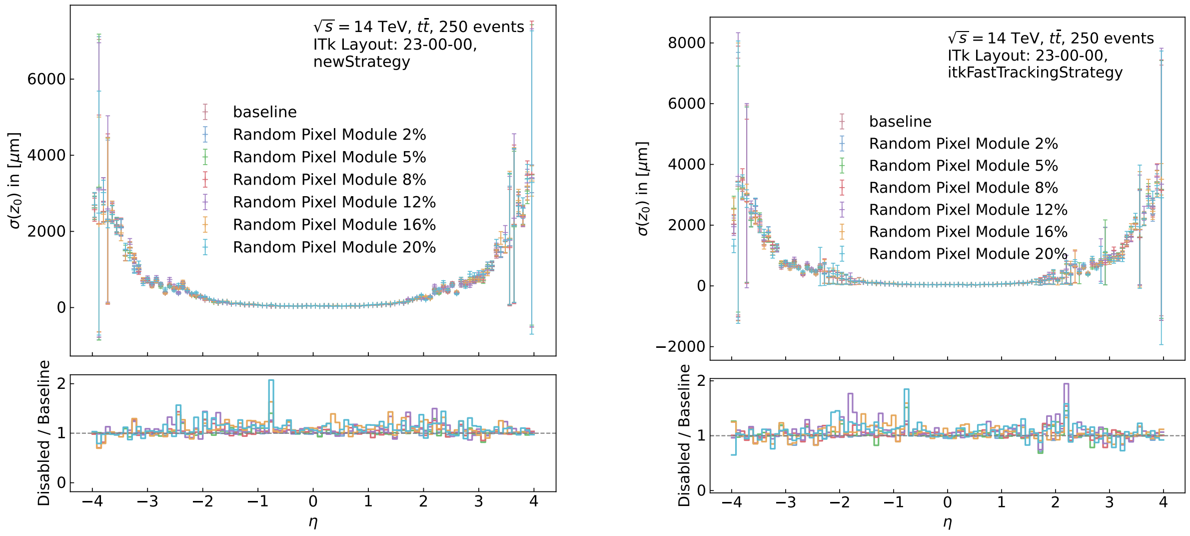
Figure 4.14.: Fake rate in η bins under increasing levels of random pixel module masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

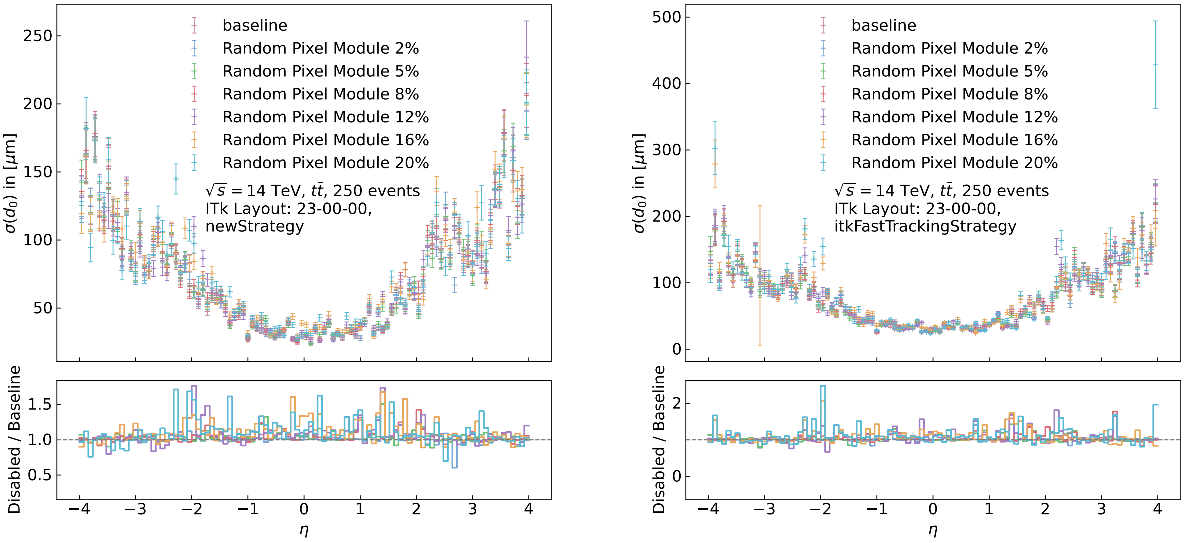
Figure 4.15.: Distribution of χ^2/N_{dof} under increasing levels of random pixel module masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

Figures 4.16 and 4.17 show that up to a masking fraction of 20%, no significant impact is observed on the width of the d_0 and z_0 resolution distributions.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.16.: Distribution of z_0 resolution width under increasing levels of random pixel module masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.17.: Distribution of d_0 resolution width under increasing levels of random pixel module masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%.

Table 4.4 summarizes the seed usage statistics with the default reconstruction chain for various pixel module-level masking fractions. A clear trend is observed: as the fraction of randomly disabled pixel modules increases, the number of PPP seeds consistently decreases across all stages: input seeds, used seeds, seeds with tracks, output tracks, and extra tracks; the seed usage of SSS seeds remains largely stable across all configurations, as expected.

Quantitatively, the number of input PPP seeds drops by 44.16%, from 6,648,384 at baseline to 3,712,143 with 20% of the pixel modules randomly disabled. The number of used seeds decreases even more substantially, by 49.71%, from 3,788,349 to 1,905,124. In comparison, the number of output tracks associated with PPP seeds shows a more moderate reduction of 24.29%, decreasing from 618,075 to 467,918. This suggests that despite a nearly 50% reduction in seed availability, the track reconstruction efficiency per used seed is partially maintained under masking conditions.

Table 4.4.: Summary of seed usage under the default chain with different fractions of randomly disabled pixel modules. For each masking configuration, values that increase compared to the baseline are highlighted in magenta, and values that decrease are highlighted in blue.

	Baseline	**2% disabled**	**5% disabled**	**8% disabled**	**12% disabled**	**16% disabled**	**20% disabled**
Input seeds (PPP)	6,648,384	6,252,676	5,811,677	5,386,233	4,799,373	4,262,445	3,712,143
Input seeds (SSS)	1,384,594	1,384,594	1,384,594	1,384,594	1,384,594	1,384,594	1,384,594
Used seeds (PPP)	3,788,349	3,517,064	3,219,548	2,950,736	2,578,778	2,239,059	1,905,124
Used seeds (SSS)	574,490	574,573	574,471	574,458	574,352	574,080	574,091
Seeds with tracks (PPP)	562,550	546,614	528,892	509,279	481,294	454,625	422,103
Seeds with tracks (SSS)	285,737	286,569	287,307	288,421	289,425	291,767	293,585
Output tracks (PPP)	618,075	601,031	582,010	561,445	531,972	503,412	467,918
Output tracks (SSS)	309,884	310,557	311,472	312,820	313,983	317,396	319,150
Extra tracks (PPP)	55,525	54,417	53,118	52,166	50,678	48,787	45,815
Extra tracks (SSS)	24,147	23,988	24,165	24,399	24,558	25,629	25,565

Table 4.5 summarizes the seed usage behavior with the fast track reconstruction chain. Due to the PPP seed-only strategy employed in this chain, the number of available seeds decreases significantly as more pixel modules are disabled. Between the baseline and the 20% disabled configuration, the number of input seeds drops by approximately 44% (from 3,224,644 to 1,807,461), while the number of used seeds decreases by about 43% (from 1,485,177 to 847,313).

However, the number of seeds with tracks shows a smaller relative reduction of only 26% (from 559,609 to 415,431), suggesting that a large fraction of useful seeds are still preserved, even under substantial detector degradation. This trend is consistent with what is observed with the default reconstruction chain: the number of pixel-based seeds drops sharply with increasing masking, while the seed-to-track efficiency improves, indicating a higher utilization of available seeds in the presence of masking.

Table 4.5.: Summary of PPP seed usage with the fast track reconstruction chain with different fractions of randomly disabled pixel modules. Values that decrease compared to the baseline are highlighted in blue.

Masking Fraction	**Baseline**	**2%**	**5%**	**8%**	**12%**	**16%**	**20%**
Input seeds (PPP)	3,224,644	3,042,764	2,836,781	2,648,735	2,353,405	2,091,885	1,807,461
Used seeds (PPP)	1,485,177	1,401,039	1,307,209	1,227,682	1,091,450	978,622	847,313
Seeds with tracks (PPP)	559,609	544,850	527,965	510,171	481,023	450,671	415,431

Comparison of Masking Configuration: Pixel Cell-Level vs. Pixel Module-Level The analysis revisits the question at the beginning of this section: if pixel modules are considered as collections of a large number of pixel cells, then when the base count is sufficiently high, disabling the same proportion of pixel cells and pixel modules may not result in a significant difference in the total masked detector area. However, the spatial pattern of the masked regions is quite different, which can impact tracking performance and seed usage behavior.

Table 4.6.: Number of reconstructed tracks under two reconstruction chains, comparing pixel cell-level and pixel module-level random masking.

Masking Level	**Pixel Cell-Level Masking**		**Pixel Module-Level Masking**	
	Default	**Fast Track**	**Default**	**Fast Track**
Baseline	498,274	459,464	498,274	459,464
2%	495,203	454,391	493,608	449,601
5%	489,009	443,864	487,896	438,504
8%	481,063	430,097	481,390	426,061
12%	466,318	410,912	470,824	404,970
16%	447,670	387,911	461,108	383,270
20%	425,384	362,660	447,481	356,541

Table 4.6 compares the number of reconstructed tracks under two reconstruction chains, given the same random masking fraction but applied at different levels of the pixel detector (cell-level vs. module-level).

We begin with the default reconstruction chain. In terms of the total number of reconstructed tracks, cell-level masking results in slightly more tracks than module-level masking when 2% and 5% of the pixel detector is disabled. However, from 8% to 20% masking, the trend reverses, and module-level masking preserves more tracks. In the $|\eta| \in [0, 2]$ region, cell-level masking shows smaller deviation from the baseline, but the impact becomes more significant in the forward region.

Regarding tracking efficiency, when 20% of pixel cells are randomly disabled, the efficiency in the forward region drops to 75% relative to the baseline. In contrast, with 20% pixel module masking, the efficiency in the same region drops only to 90%. As for the fake rate, module-level masking leads to only a slightly higher increase compared to cell-level masking. The mean shift in χ^2/N_{dof} becomes more pronounced as the cell-level masking fraction increases, while module-level masking shows a milder shift. For the resolution width of d_0 and z_0, there is no significant difference between the two masking strategies, neither demonstrates a clear advantage.

With the fast track reconstruction chain, cell-level masking consistently yields a slightly higher number of reconstructed tracks than module-level masking across the entire range from 2% to 20% random masking. Cell-level masking tends to preserve more tracks in the barrel region, but causes larger degradation in the forward region. In contrast, module-level masking leads to a more uniform reduction in reconstructed tracks across all η regions. For tracking efficiency, module-level masking degrades more rapidly in the central barrel as the masking fraction increases, but performs better in the forward region. This trend is opposite to what is observed with the default reconstruction chain. For both χ^2/N_{dof} and the resolution width of d_0 and z_0, the two masking strategies behave similarly, with no significant differences observed.

In terms of seed usage, the difference between the two masking strategies with the default reconstruction chain is striking. With pixel cell-level masking, the number of input PPP seeds only decreases by 8.1% when 20% of the pixel cells are disabled. In contrast, under module-level masking, the number of input PPP seeds drops by 44.16%. Similarly, the number of used PPP seeds remains relatively stable under cell-level masking but drops by nearly 50% when 20% of the modules are masked. The number of output tracks from PPP seeds also decreases with increasing module-level masking, while the number of output tracks from SSS seeds increases slightly as a compensation effect. Under cell-level masking, however, both PPP and SSS output tracks increase with higher masking fractions. This means that for the same masking fraction, cell-level masking produces more output tracks than module-level masking. Nevertheless, in terms of final reconstructed tracks and tracking efficiency, pixel cell-level masking performs worse. A likely explanation is that cell-level masking tends to distort clusters, while module-level masking directly removes a large portion of clusters.

A similar trend is observed in the fast track reconstruction chain. Compared to pixel cell-level masking, module-level masking results in a more significant reduction in both input and used seeds. Consequently, the number of seeds with tracks is consistently higher under pixel cell-level masking for the same masking fraction.

4.3. Tracking Performance with Selectively Configured Pixel Detector Defects

To identify pixel layers most critical to tracking performance, a preliminary study was performed by disabling individual layers, aiming to highlight regions sensitive to local defects.

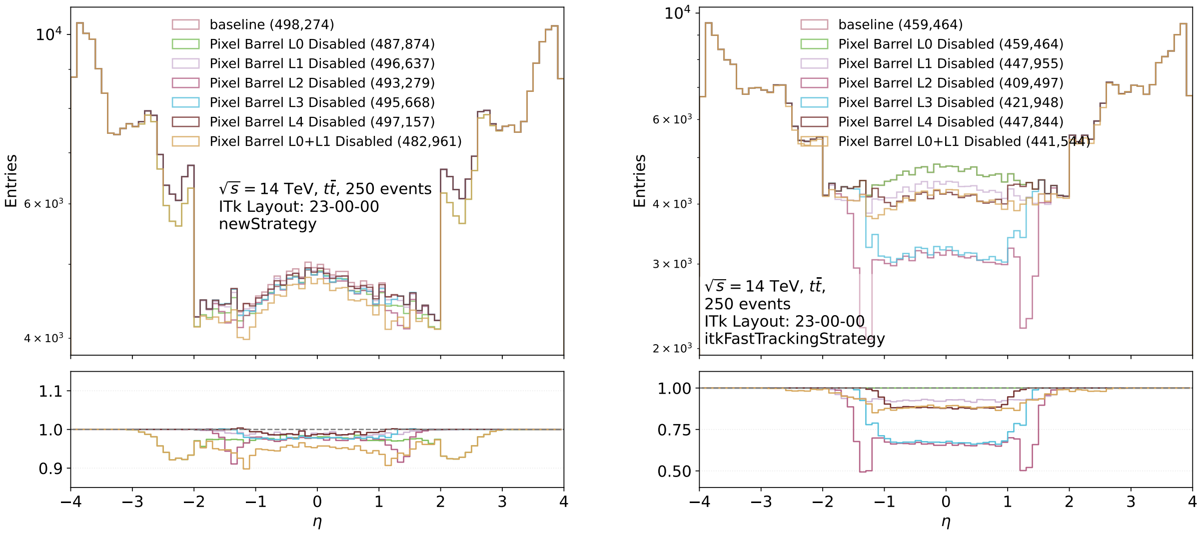
Due to the detector geometry, some regions are inherently more robust than others. In the barrel region, the presence of overlapping strip layers provides additional redundancy, especially for high-p_T tracks; the forward region lacks such coverage and receives no compensation from the strip detector, making it more susceptible to local defects.

In this study, the layer labeled as pixel endcap layer 2 in the internal ATHENA ITk geometry was further divided into two regions due to its length, as shown in Figure 2.5: the inner rings (`eta module` indices 0–13), which are more barrel-like and cover the range $|\eta| \in [0, 3)$, and the outer rings (indices 14–22), corresponding to the forward region spanning $|\eta| \in [3, 4]$. The impact of selectively disabling these regions was evaluated using the number of reconstructed tracks, tracking efficiency, fake rate, and reconstructed parameters.

4.3.1. Pixel Barrel Layers

Figure 4.18 shows the η distribution of the number of reconstructed tracks under different pixel barrel defect configurations, compared to the full detector baseline. The configurations include disabling individual pixel barrel layers, as well as a combined failure scenario where both layer 0 and layer 1 are removed. The motivation for this combined configuration is based on their intended design for replacement once a certain integrated luminosity is reached.

The comparison includes both reconstruction chains. The fast chain is more sensitive to pixel layer removal, as expected from its dependence on PPP seeds. Because the default reconstruction chain prioritizes SSS seeds, disabling any single pixel barrel layer among L0, L1, L3, or L4 only reduces the number of reconstructed tracks in the barrel and transition regions to about 98% of the baseline. Disabling L2 has the most significant impact among the individual layers, while it does not lead to a notable reduction in the central barrel, the number of tracks drops to as low as 92% of the baseline around $|\eta| \approx 1$. The most pronounced degradation occurs when both L0 and L1 are disabled simultaneously. In contrast, under the fast reconstruction chain, disabling pixel barrel L2 or L3 leads to a substantially larger decrease in the number of reconstructed tracks within the central region ($|\eta| \in [0,1]$) than what is observed when both L0 and L1 are disabled.

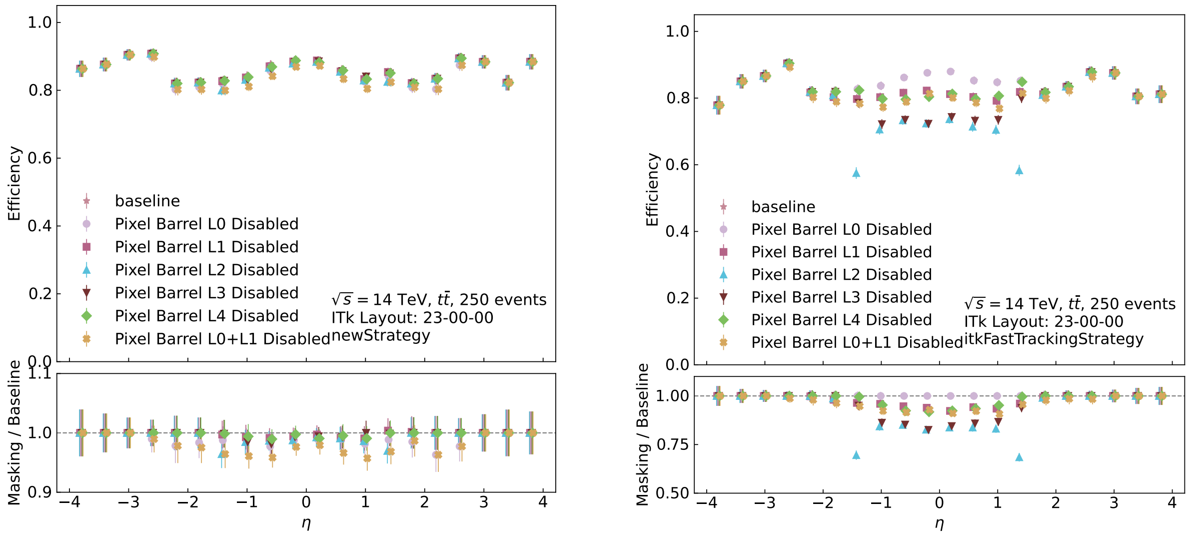


(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.18.: η distributions of reconstructed tracks under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain. Disabling Layer 2 leads to the largest loss in both reconstruction chains, followed by Layer 3.

With the default reconstruction chain, the impact of pixel barrel defects on tracking efficiency is modest, as shown in Figure 4.19a, owing to the redundancy provided by strip detectors in the barrel region, the algorithm is resilient to localized pixel losses. Even disabling both pixel barrel layers 0 and 1 reduces efficiency to only about 95% of the baseline. Among single-layer defects, disabling layer 2 leads to the largest drop, though still less severe than the combined L0+L1 scenario.

As shown in Figure 4.19b, the fast reconstruction chain is sensitive to pixel defects. Disabling any outer barrel layer causes a larger efficiency loss compared to the default reconstruction chain, with layer 2 having the most significant impact. Notably, removing layer 2 alone results in a greater loss than removing both layers 0 and 1. Disabling layer 2 or 3 yields nearly identical degradation in tracking efficiency and the number of reconstructed tracks within the barrel region, but only layer 2 removal causes a clear efficiency drop in the transition region.



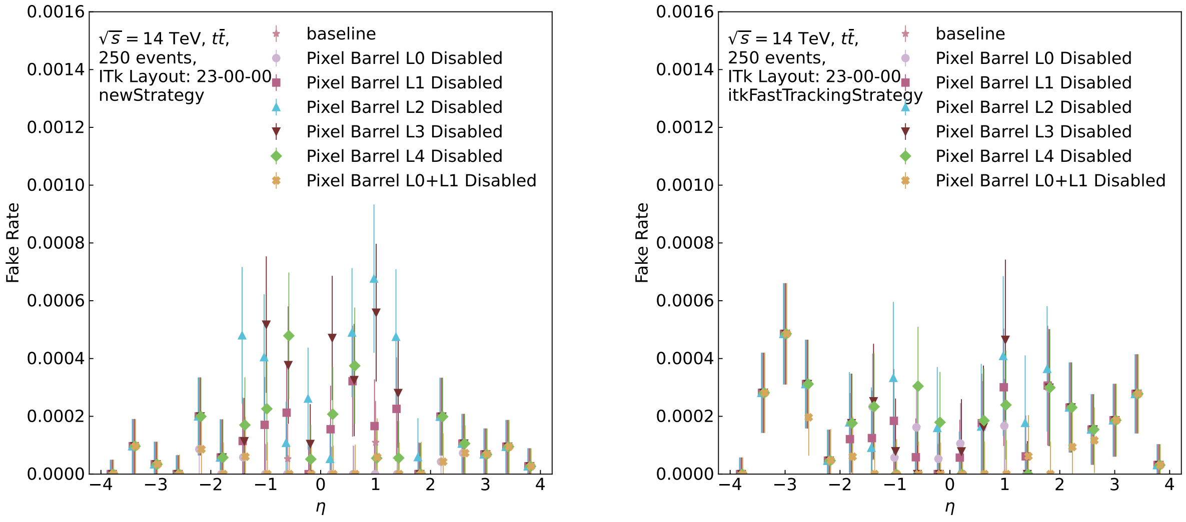
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.19.: Tracking efficiency in η bins under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain.

Figure 4.20 illustrates that for both reconstruction chains, disabling any single pixel layer or the combination of layers 0 and 1 does not lead to a significant increase in fake rate. This holds even for the fast reconstruction chain under pixel losses. The impact remains confined to the barrel region, with the largest effects observed when disabling pixel barrel layers 2 or 3, consistent with the trends seen in the number of reconstructed tracks and tracking efficiency.

Disabling any single pixel barrel layer, or even the entire inner system of the pixel barrel, does not result in significant shifts in the peak of the χ^2/N_{dof} distribution under either reconstruction chain, as shown in Figure 4.21. Figure 4.22a and 4.23a show that with the default reconstruction chain, masking Pixel Barrel L0 or L1 individually leads to a measurable broadening in the resolution width of both d_0 and z_0. This indicates a moderate degradation in vertex and impact parameter resolution when early layers are removed.

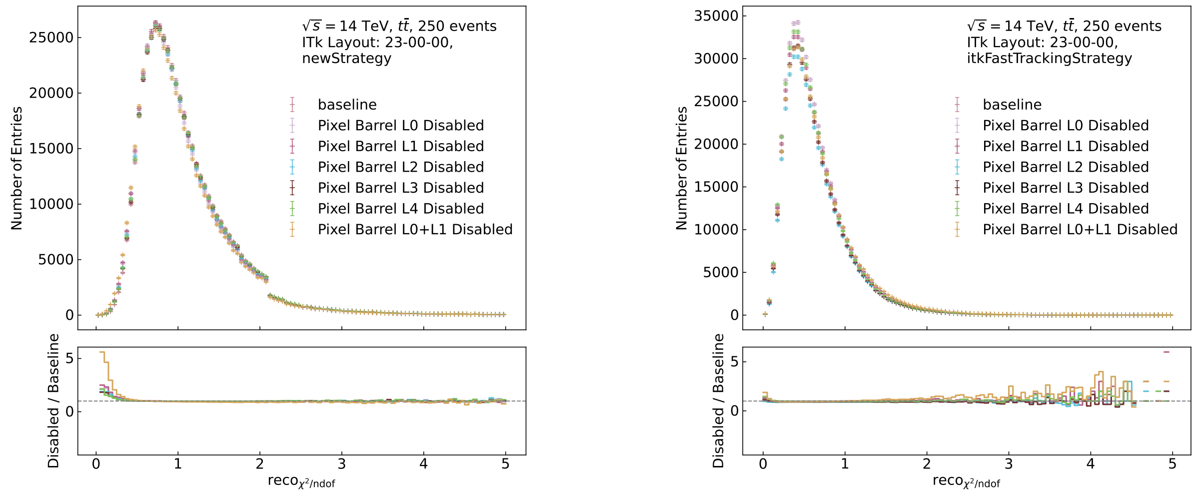
The fast track reconstruction chain shows greater tolerance to individual pixel barrel layer defects in terms of resolution. The resolution widths remain largely unaffected when only one pixel barrel layer is disabled. Noticeable degradation comparable to that observed with the default reconstruction chain only emerges when both pixel barrel L0 and L1, as the inner system, are simultaneously disabled, as shown in Figure 4.22b and 4.23b.



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

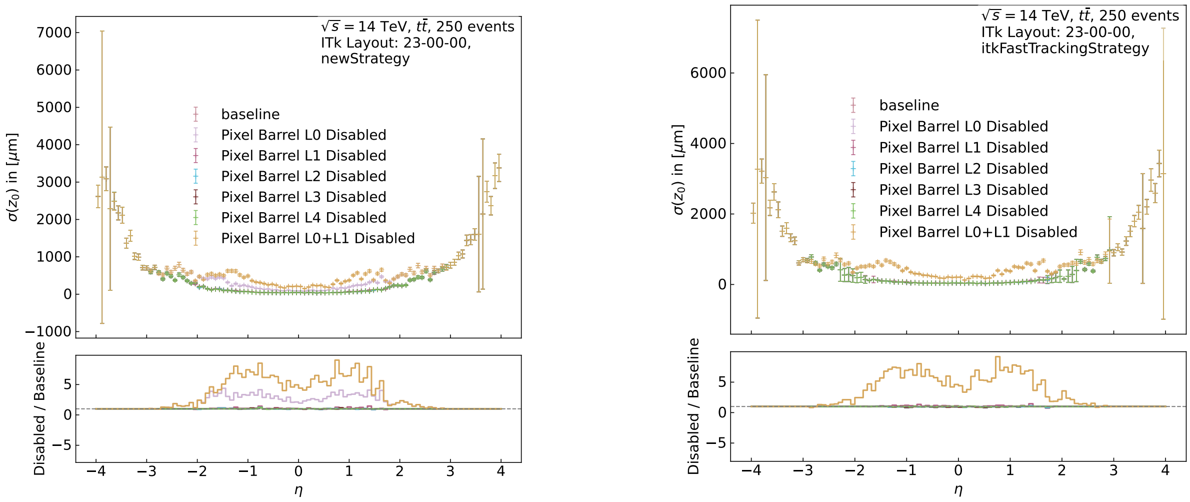
Figure 4.20.: Fake rate in η bins under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain.



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

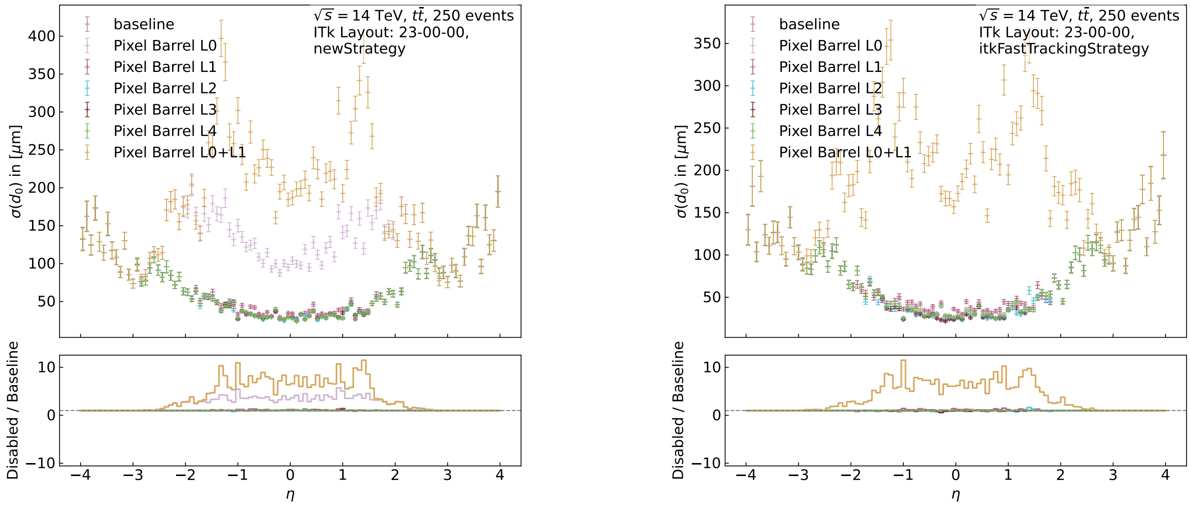
Figure 4.21.: Distribution of χ^2/N_{dof} under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain.



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

Figure 4.22.: Distribution of z_0 resolution width under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain.



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

Figure 4.23.: Distribution of d_0 resolution width under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain.

Overall, for individual pixel barrel layer masking with the default reconstruction chain, the most severe degradation occurs when L0 and L1 are simultaneously disabled, followed by L2. In contrast, with the fast track reconstruction chain, disabling L2 or L3 individually has a greater negative impact than simultaneously disabling L0 and L1. Disabling only pixel barrel L0 leads to a noticeable degradation in $\sigma(d_0)$ and $\sigma(z_0)$ under the default chain, while for the fast chain, a significant impact on resolution is observed only when both replaceable inner system layers (pixel barrel L0 and L1) are disabled.

4.3.2. Pixel Endcap Layers

Author's note: In ATHENA, the forward region's endcap layer 2 consists of 23 η-modules. Disabling the entire layer would result in the most severe loss of tracking performance. To avoid this, pixel endcap layer 2 was divided into two parts: the portion overlapping with the outer barrel in the z direction is referred to in this work as the **endcap l2 barrel rings**, while the remaining portion, overlapping with the outer endcap region, is referred to as the **endcap l2 endcap rings**. Although both names contain "endcap", they correspond to different spatial regions. Please refer to the Figure 4.24 below for clarification.

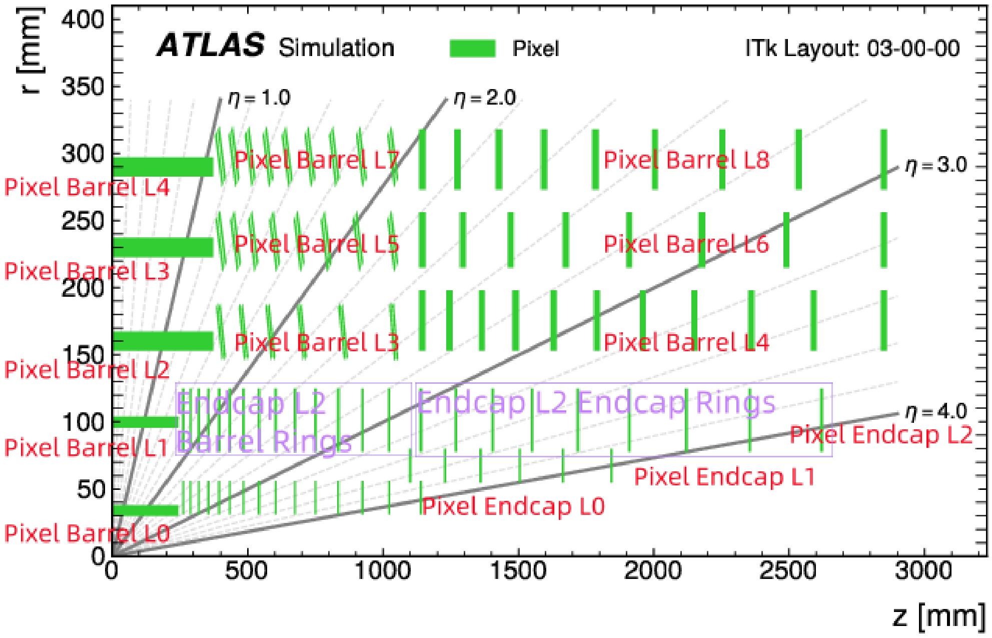
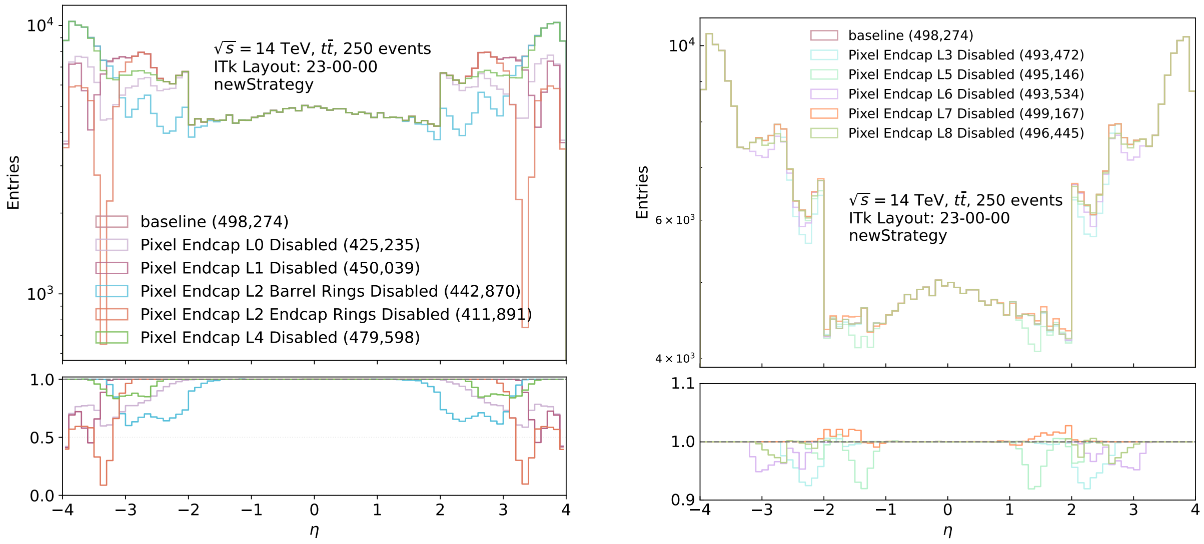


Figure 4.24.: Schematic view of the ITk pixel detector layout in the r-z plane. In this study, endcap layer 2 is further divided into two spatially distinct parts. Figure adapted from Ref.[26].

The pixel endcap layers can be roughly divided into two groups based on their spatial redundancy. The first group consists of endcap layers L0, L1, and L2. These inner layers are not covered by any surrounding strip detector layers. As a result, disabling any one of them leads to a significant loss in tracking performance in the forward region, even with the default reconstruction chain. This is evident in Figure 4.25a, where the impact of disabling endcap L0, L1, or L2 is clearly visible. (The configuration with endcap L4 disabled is also included in this figure due to layout constraints.)

The second group includes endcap layers L3 to L8 (i.e., L3, L4, L5, L6, L7, L8). These layers are located in regions where strip detectors provide additional spatial coverage. Therefore, disabling any single layer in this group causes only limited performance degradation in the default reconstruction chain, as shown in Figure 4.25b.



(a) Pixel Endcap L0, L1, L2, L4 disabled with the default reconstruction chain.

(b) Pixel Endcap L3, L5, L6, L7, L8 disabled with the default reconstruction chain.

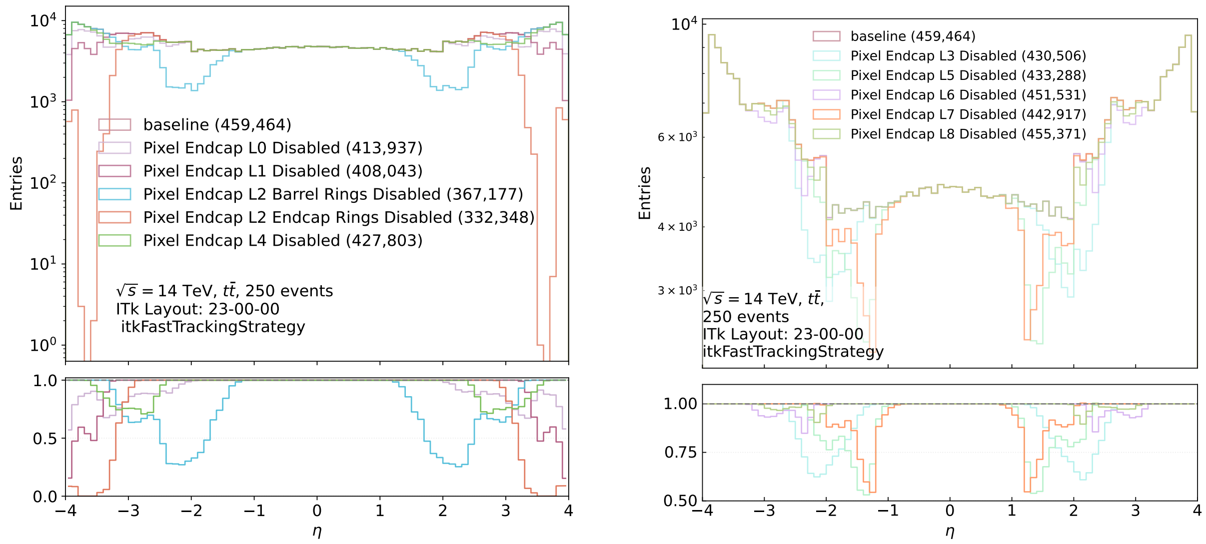
Figure 4.25.: η distributions of reconstructed tracks under individual pixel endcap layer deactivations, including separate masking of Endcap L2 barrel rings and endcap rings, for default reconstruction chain.

However, the notion of "limited impact" applies primarily to the default reconstruction chain, which benefits from an SSS seed-first strategy. The fast track reconstruction chain, which relies exclusively on PPP seeds, is significantly more sensitive to the loss of pixel endcap layers. Disabling any endcap layer results in a substantial drop in the number of reconstructed tracks in the corresponding η region.

The reduction is most pronounced when the outer endcap layers are disabled, but even for layers L6 and L8 which show relatively smaller degradation, the number of reconstructed tracks in the forward and endcap regions can drop to as low as 75% of the baseline. This behavior is illustrated in Figure 4.26.

For the fast reconstruction chain, the impact of disabling strip endcap layers L4, L6, and

L8 is relatively smaller (though only in comparison to other layers), and these three layers correspond to the "outer endcap" region of the detector. This configuration aligns with one of the staged installation scenarios evaluated in Ref. [12], in which the installation of the outermost strip endcap layers was deferred. However, as the performance results indicate, this approach leads to a substantial degradation in tracking performance, and the note itself acknowledges that the outcome is clearly unsatisfactory.



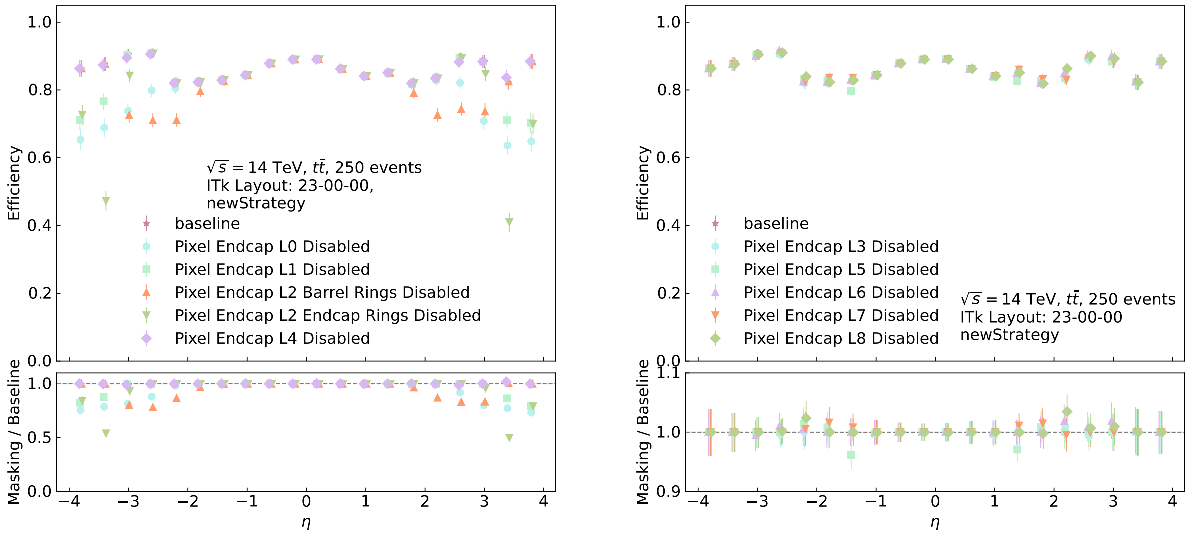
(a) Pixel Endcap L0, L1, L2, L4 disabled with the fast track reconstruction chain.

(b) Pixel Endcap L3, L5, L6, L7, L8 disabled with the fast track reconstruction chain.

Figure 4.26.: η distributions of reconstructed tracks under individual pixel endcap layer deactivations, including separate masking of Endcap L2 barrel rings and endcap rings, for fast track reconstruction chain.

The tracking efficiency of the default reconstruction chain remains relatively stable under single-layer pixel endcap defects. For configurations in which endcap layers L3, L4, L5, L7, or L8 are individually disabled, the overall efficiency stays above 98% across most η regions. In some cases, slight increases above the baseline are observed, possibly due to statistical fluctuations or specific algorithmic effects, though such increases are not necessarily indicative of improved performance.

The most significant losses occur when inner endcap layers are disabled, particularly endcap L2 endcap rings, the tracking efficiency in the forward region can drop to approximately 50% of the baseline. Other inner layers such as L0 and L1 also lead to noticeable efficiency degradation, with values in the forward region reaching as low as 75% of the baseline. These effects are shown in Figure 4.27.

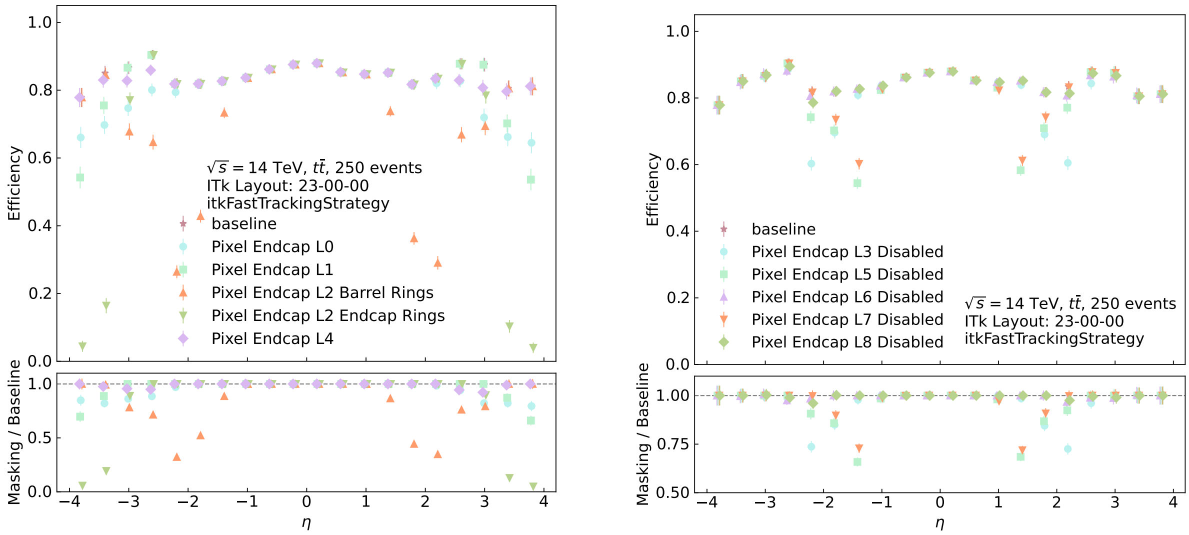


(a) Pixel Endcap L0, L1, L2, L4 disabled with the default reconstruction chain.

(b) Pixel Endcap L3, L5, L6, L7, L8 disabled with the default reconstruction chain.

Figure 4.27.: Tracking efficiency in η bins under individual pixel endcap layer deactivations, including separate masking of Endcap L2 barrel rings and endcap rings, for default reconstruction chains.

When individual forward pixel layers are disabled, the tracking efficiency with the fast track reconstruction chain drops extremely sharply in the affected η regions. In extreme cases, such as when endcap layer L2 barrel rings or endcap rings are masked, the efficiency approaches zero, because very few tracks can be reconstructed in those regions. Disabling outer endcap layers such as L3, L5, L6, L7, or L8 results in comparatively smaller losses with the fast track reconstruction chain. Nevertheless, masking endcap L5 or L7 still reduces the tracking efficiency in the endcap region to approximately 85% of the baseline. These effects are illustrated in Figure 4.28.



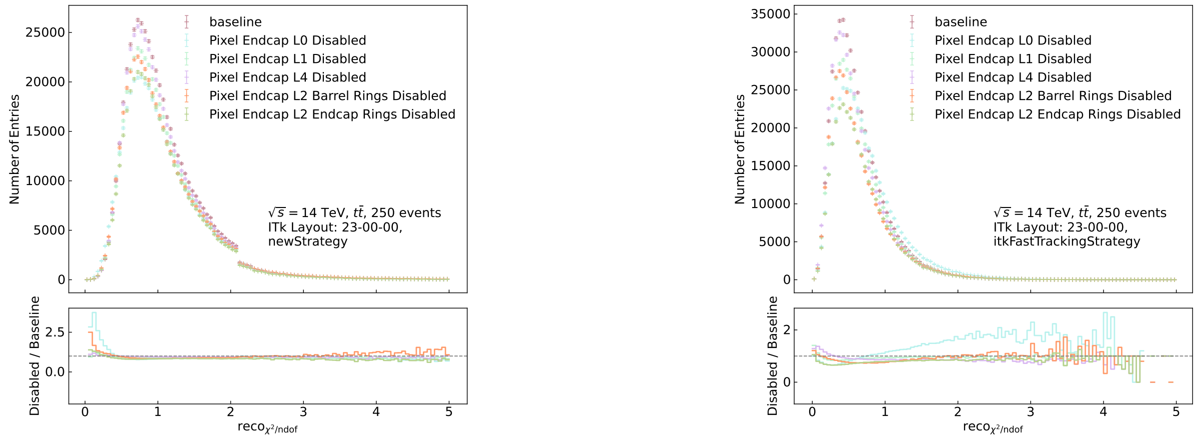
(a) Pixel Endcap L0, L1, L2, L4 disabled with the fast track reconstruction chain.

(b) Pixel Endcap L3, L5, L6, L7, L8 disabled with the fast track reconstruction chain.

Figure 4.28.: Tracking efficiency in η bins under individual pixel endcap layer deactivations, including separate masking of Endcap L2 barrel rings and endcap rings, for fast track reconstruction chains.

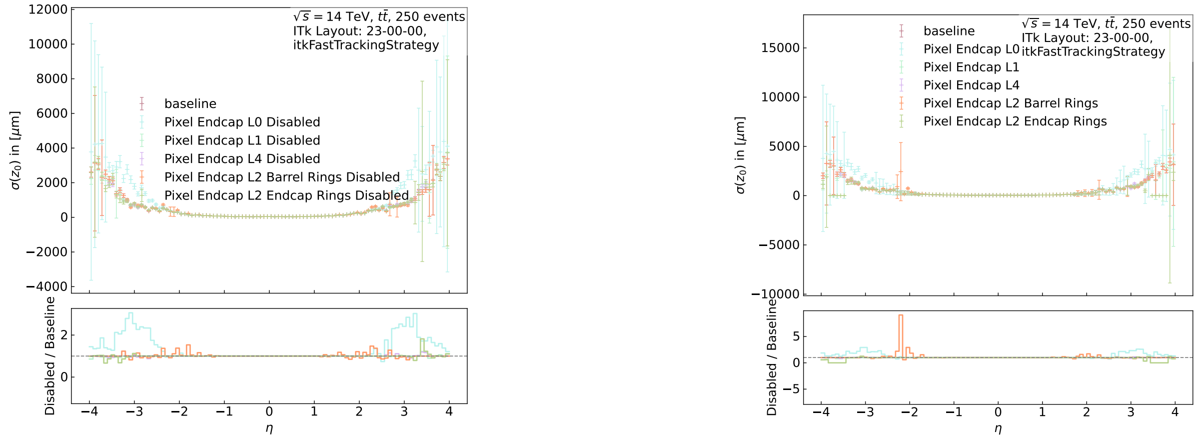
With the default reconstruction chain, disabling any single layer among the Pixel Barrel L0, L1, L2 barrel rings, the L2 endcap rings, or the L4 layer does not lead to a noticeable shift in the peak or a significant change in the tail of the χ^2/N_{dof} distribution. This behavior is shown in Figure 4.29a. With the fast track reconstruction chain, disabling any one of endcap layers L0, L1, L2, or L4 leads to a shift in the mean of the χ^2/N_{dof} distribution, with L0 causing the largest deviation (0.7371 vs. 0.6408), as shown in Figure 4.29b. This is expected, since Pixel Endcap L0 is the innermost endcap layer and closest to the $|\eta| \approx 4$ region. Therefore, masking endcap L0 has a relatively strong impact on the resolution of d_0 and z_0 in the forward region, regardless of the reconstruction chain. This effect is explicitly illustrated in the resolution comparisons shown in Figures 4.29c, 4.29d, 4.29e, and 4.29f.

Disabling any individual layer among Pixel Endcap L3, L5, L6, L7, or L8 has only a limited impact on the overall χ^2/N_{dof} distribution as well as on the width of the z_0 and d_0 resolution. This observation holds for both reconstruction chains and is illustrated in Figure 4.30.



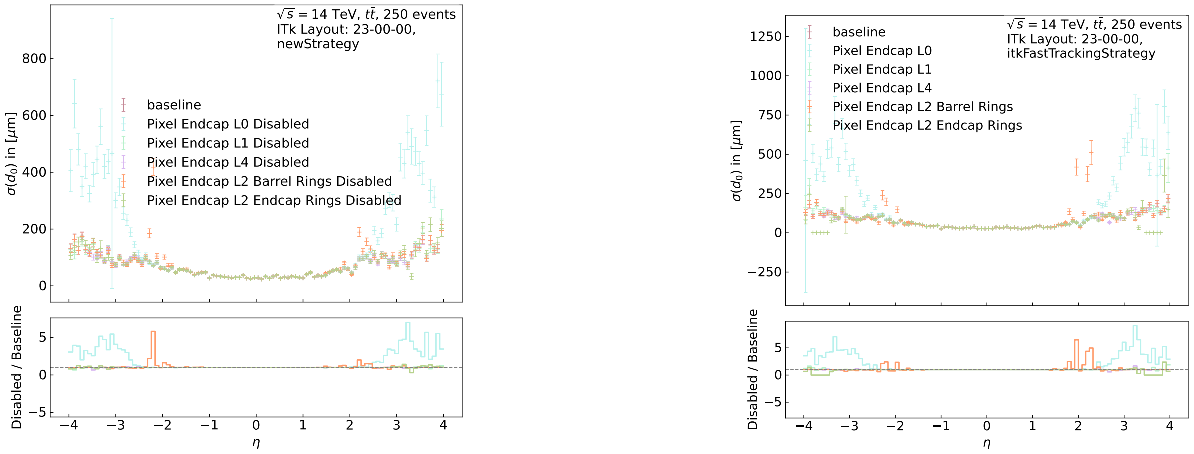
(a) χ^2/N_{dof} with the default reconstruction chain.

(b) χ^2/N_{dof} with the fast track reconstruction chain.



(c) Width of resolution for z_0 with the default reconstruction chain.

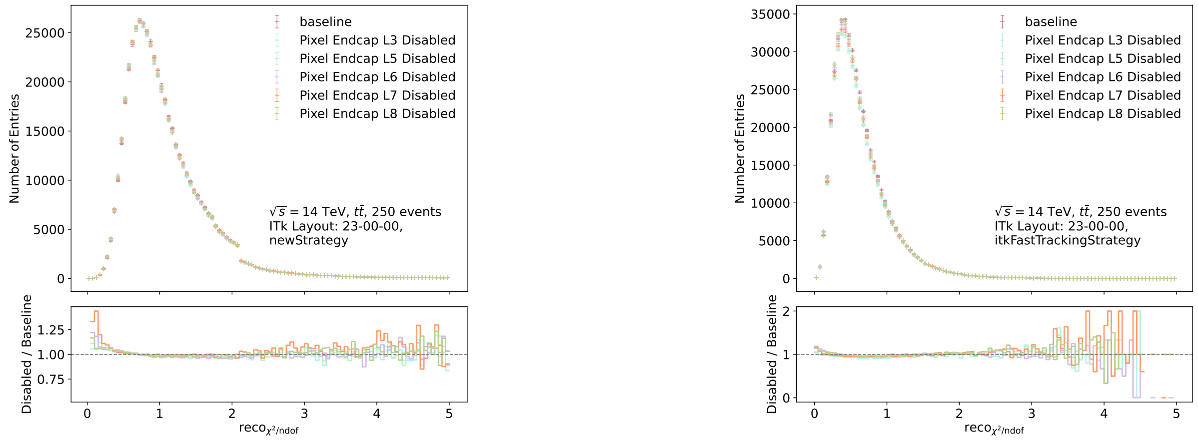
(d) Width of resolution for z_0 with the fast track reconstruction chain.



(e) Width of resolution for d_0 with the default reconstruction chain..

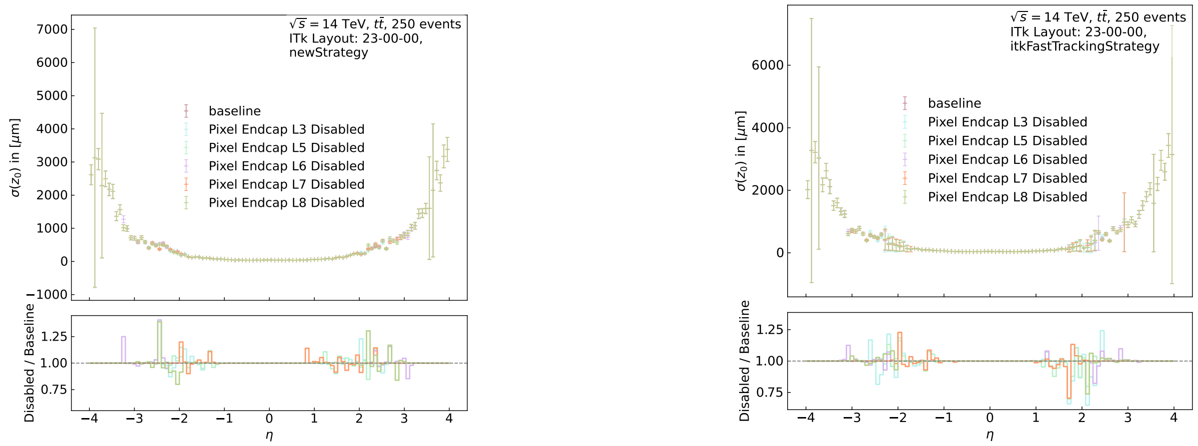
(f) Width of resolution for d_0 with the fast track reconstruction chain.

Figure 4.29.: Summary of χ^2/N_{dof} and track parameter resolution widths under pixel endcap layer masking (L0, L1, L2, and L4) for both reconstruction chains. Top: χ^2/N_{dof} distributions. Middle: width of z_0 residual distributions. Bottom: width of d_0 residual distributions.



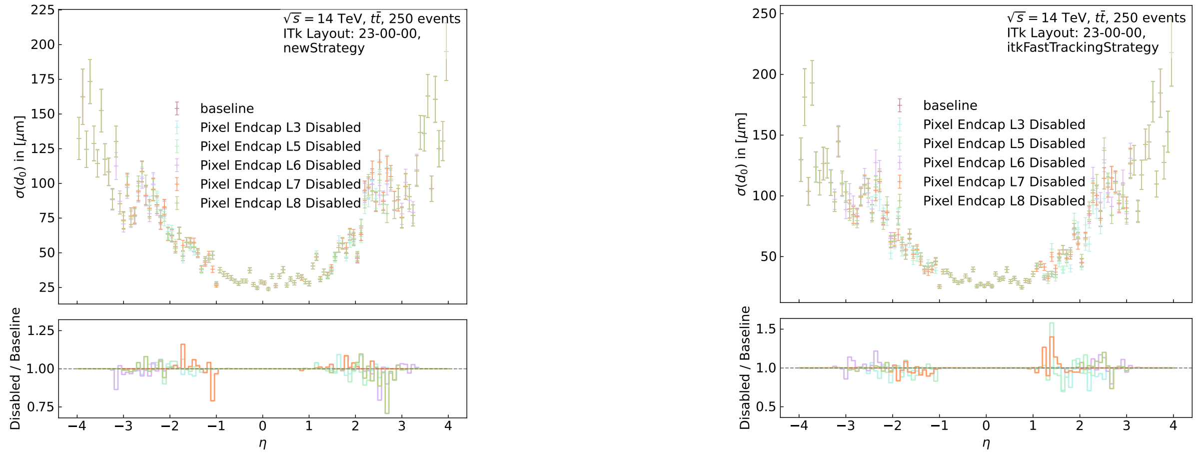
(a) χ^2/N_dof with the default reconstruction chain.

(b) χ^2/N_dof with the fast track reconstruction chain.



(c) Width of resolution for z_0 with the default reconstruction chain.

(d) Width of resolution for z_0 with the fast track reconstruction chain.



(e) Width of resolution for d_0 with the default reconstruction chain.

(f) Width of resolution for d_0 with the fast track reconstruction chain.

Figure 4.30.: Summary of χ^2/N_dof and track parameter resolution widths under masking of pixel endcap layers L3, L5, L6, L7, and L8 for both reconstruction chains. Top: χ^2/N_dof distributions. Middle: width of z_0 residual distributions. Bottom: width of d_0 residual distributions.

4.3.3. Under Selective Pixel Detector Degradation

As discussed in Ref. [12], one idea considered in the context of potential delays in pixel detector delivery was to split the installation into two stages-installing part of the detector after LS3 and deferring the rest to LS4. This note investigated two such scenarios: one where the pixel outer endcap (corresponding to layers Endcap L4, L6, and L8 in ATHENA, comprising 2344 quad modules) would be installed later, and another where Layer 4 (corresponding to Endcap L7 and L8, comprising 1944 quad modules) would be deferred. The study concluded that, with the fast track reconstruction chain, such partial configurations are not viable, and even the default reconstruction chain is unable to recover satisfactory tracking performance.

This line of investigation was originally intended as a pessimistic stress test, aiming to evaluate how well tracking performance could be maintained under severe pixel detector degradation. At the time, the default reconstruction chain served as the practical reference, and various masking configurations were tested to probe the limits of geometrical redundancy.

However, after the fast reconstruction chain became the baseline, it was found that tracking performance degraded noticeably even under relatively modest levels of pixel damage. This limited fault tolerance arises from the strict PPP-seed-only strategy and the absence of a dedicated ambiguity solving stage, both design choices made to reduce computational cost. As a result, such levels of damage were found to be beyond recovery within the fast reconstruction chain. Nevertheless, several test simulations were still performed with the fast track reconstruction chain to document the failure modes. The results are included in Appendix E for completeness.

A specific configuration was identified in which a total of 2528 modules, including both quad and triplet modules were masked, in combination with 5% random pixel cell masking and 10% FECC masking. Due to the geometric layout of the detector, the forward region exhibits effectively no fault tolerance under either reconstruction chain. Based on the results presented in Sections 4.3.1 and 4.3.2, the most critical layers with the default reconstruction chain are Pixel Barrel L0, L2, and L3. For the fast track reconstruction chain, the most significant degradation is observed when Pixel Barrel L2 or L3 is disabled.

In both reconstruction chains, endcap layers L0, L1, L2, and L4 reside in the high-$|\eta|$ region and therefore exhibit no redundancy. Their removal leads to substantial losses in tracking performance. The key distinction lies in the ability of the default reconstruction chain to partially compensate for such losses using SSS seeds, particularly when outer endcap layers (L3, L5, L6, L7, L8) are affected. This makes such configurations relatively more tolerable under the default chain. In contrast, the fast reconstruction chain lacks this redundancy and is significantly more sensitive to pixel detector losses across all regions.

With the default reconstruction chain, if one assumes a pessimistic scenario in which 30% of the quad modules and 10% of the triplet modules fail, this corresponds to approximately 2,393 non-functional quad modules and 119 non-functional triplet modules.

It is clear that modules located in the forward region ($|\eta| > 3$) must be prioritized for full functionality, as this region exhibits minimal fault tolerance. When choosing between relatively robust layers such as Pixel Barrel L1 and L4 for allocating damaged quad modules, it is preferable to assign failures to L4. This is because L1 plays a more critical role in determining impact parameter resolution, and a module in L1 covers a larger solid angle due to its proximity to the beam line.

Although masking Pixel Barrel L0 is known to degrade the d_0 and z_0 resolution with the default reconstruction chain, it is important to note that only Barrel L0, Endcap L0, and Endcap L1 are built using triplet modules. Given the high sensitivity of the forward region and its lack of redundancy, all 119 non-functional triplet modules are conservatively assigned to the Pixel Barrel. This choice prioritizes preserving performance in the endcap and forward regions, even at the expense of some degradation in the barrel.

Three configurations were considered by the author, motivated by the desire to mitigate the impact of module failures in sensitive regions by concentrating masking on geometrically redundant areas as much as possible.

Configuration 1: Concentrated Failures on Specific Layers

- Triplet module failures: all 119 assigned to **central Barrel Layer 0**
- Quad module failures: approximated by masking **central Barrel Layers 1 and 4, Endcap Layers 3 and 8**, totaling approximately 2679 failed modules

Configuration 2: Uniform Quad Failures in $|\eta| < 3$

- Triplet module failures: all 119 assigned to **central Barrel Layer 0**
- Quad module failures: uniformly distributed across all quad modules with $|\eta| < 3$, totaling approximately 2531 failed modules

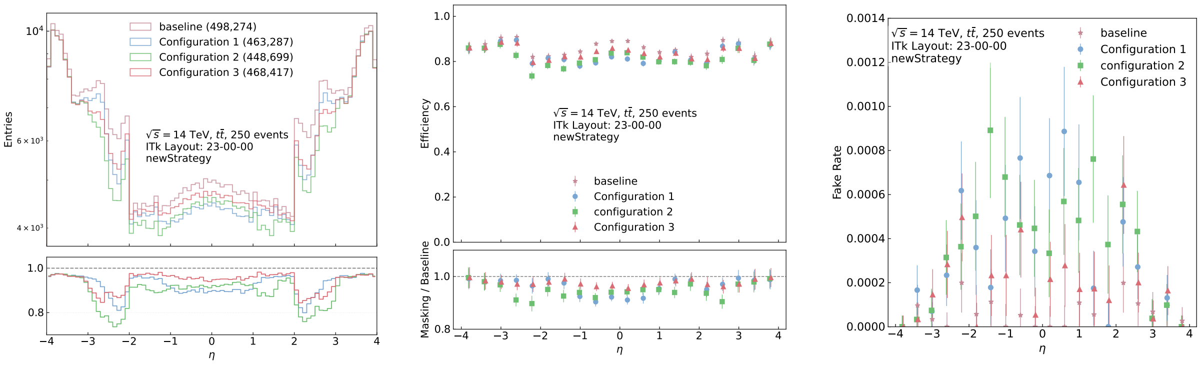
Configuration 3: Hybrid Strategy

- Triplet module failures: all 119 assigned to **central Barrel Layer 0**
- Quad module failures: a combination of full masking of **central Barrel Layer 4** and **Endcap Layer 8**, with the remaining failures uniformly distributed across modules in the $|\eta| < 3$ region, totaling approximately 2528 failed modules

It should be noted that the simulation tools used in this study model module degradation as a per-module failure probability, rather than allowing selection of which specific modules are disabled. As a result, the total number of failed modules in each configuration may vary slightly from the target due to statistical fluctuations.

With the default reconstruction chain, the best-performing configuration among those tested is Configuration 3. It results in a modest reduction of approximately 6% in the total number of reconstructed tracks (from 498,274 to 468,417, as shown in Figure 4.31a). Across the full η range, the track count remains above 80% of the baseline, and in the central region ($|\eta| < 2$), the redundancy provided by the strip detector allows the number of reconstructed tracks to reach more than 95% of the baseline within each η bin. The majority of the loss is concentrated in the pixel endcap layers.

Figure 4.31b)shows that in terms of tracking efficiency, Configuration 3 maintains at least 95% of the baseline performance across the full η range. No significant increase in the fake rate is observed, as shown in Figure 4.31c.



(a) Number of reconstructed tracks vs. η

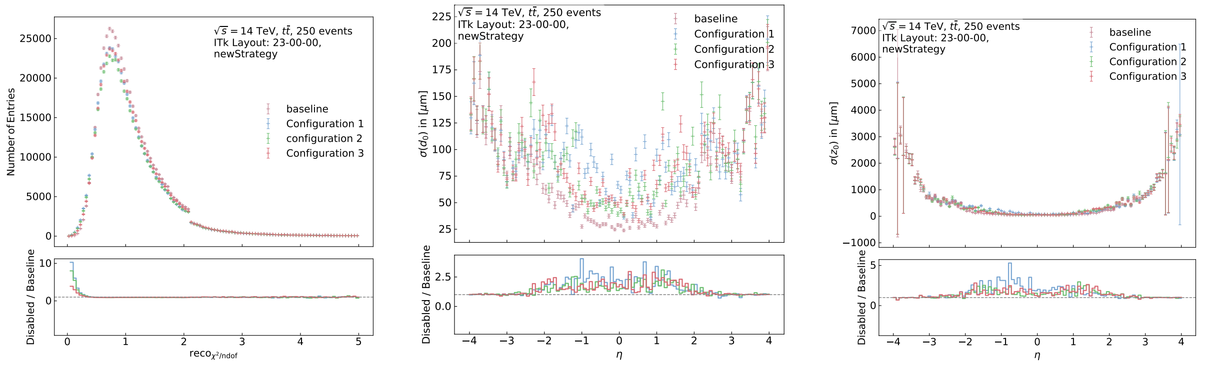
(b) Tracking efficiency vs. η

(c) Fake rate vs. η

Figure 4.31.: Tracking performance of the three selected configurations with the default reconstruction chain compared to the baseline.

Figure 4.32a shows that across the three selected defect configurations, the χ^2/N_{dof} distribution remains largely stable, with no significant shifts observed in either the peak value or the tail.

However, the width of the d_0 resolution increases for all three configurations. This degradation is primarily attributed to the intentional masking of Pixel Barrel Layer 0, where 119 out of 288 triplet modules were set as completely non-functional. The impact is more pronounced in d_0 than z_0, due to L0's proximity to the collision point and its critical role in determining the transverse impact parameter (d_0). The longitudinal impact parameter (z_0) is less affected by this specific masking configuration. These resolutions are shown in Figure 4.32b and Figure 4.32c, respectively.



(a) χ^2/N_{dof} ***(b)*** Width of resolution for d_0 ***(c)*** Width of resolution for z_0

Figure 4.32.: Track quality metrics for the three selected defect configurations with the default reconstruction chain compared to baseline.

Overall, Configuration 3 represents a relatively balanced compromise with the default reconstruction chain. While approximately one third of the pixel modules are non-functional, the tracking performance remains moderately stable across key metrics, and the most sensitive regions are partially protected by the detector's inherent redundancy.

The corresponding results under the fast reconstruction chain were found to be significantly worse and are therefore not discussed in detail in the main text. For completeness, the author includes them in Appendix E for reference.

Table 4.7 summarizes the seed usage statistics for three different configurations, along with the corresponding percentage changes relative to the baseline. Although the overall performance gap between the baseline and Configuration 3 is not that substantial, the underlying seed statistics reveal notable differences. In Configuration 3, approximately one-third of the pixel modules were disabled, resulting in a 40.3% reduction in the number of PPP input seeds and a 50.5% drop in used PPP seeds, relative to the baseline.

Despite this, the number of PPP seeds that led to reconstructed tracks decreased by only 19.3%. This suggests that the reduced number of seeds was partially offset by a higher utilization rate per seed, as the surviving seeds had less competition in the ambiguity resolution stage.

At the same time, the number of extra tracks originating from SSS seeds nearly doubled. One possible explanation is that, during ambiguity resolution, track candidates are accepted in order of descending score. The scoring algorithm favors candidates with more pixel and silicon clusters and penalizes those with holes or poor fit quality. When more pixel modules are available, PPP-based candidates typically receive higher scores and suppress overlapping SSS-based candidates. When many pixel modules are masked, PPP candidates become rarer, allowing lower-scoring SSS candidates to be promoted and accepted. As a result, more tracks from SSS seeds appear in the final output, even though the total number of SSS seeds remains unchanged.

Table 4.7.: Statistical Comparison with Seeds Usage of Configuration 1-3.

Configuration	**Input Seeds**		**Used Seeds**		**Seeds with Track**		**Output Extra Tracks**	
	PPP	**SSS**	**PPP**	**SSS**	**PPP**	**SSS**	**PPP**	**SSS**
Baseline	6 648 384	1 384 594	3 788 349	574 490	562 550	285 737	55 525	24 147
Configuration 1	3 602 132	1 384 594	1 655 370	574 096	438 545	298 359	51 899	47 226
	−45.8%	–	−56.3%	−0.07%	−22.0%	+4.4%	−6.5%	+95.6%
Configuration 2	3 262 653	1 384 594	1 460 222	573 704	410 374	299 617	43 561	47 930
	−50.9%	–	−61.4%	−0.14%	−27.0%	+4.9%	−21.5%	+98.6%
Configuration 3	3 970 465	1 384 594	1 875 334	574 400	454 246	293 614	52 241	45 159
	−40.3%	–	−50.5%	−0.02%	−19.3%	+2.8%	−5.9%	+87.0%

4.4. Tracking Performance with Individual Strip Layer Failures

This section investigates the impact of disabling individual layers of the strip detector, which consists of four barrel layers and six endcap layers, on overall tracking performance under two track reconstruction chains. The initial motivation was to identify which detector layers, when disabled, lead to more significant degradation in reconstruction quality, based on their geometric location and contribution to seeding and tracking.

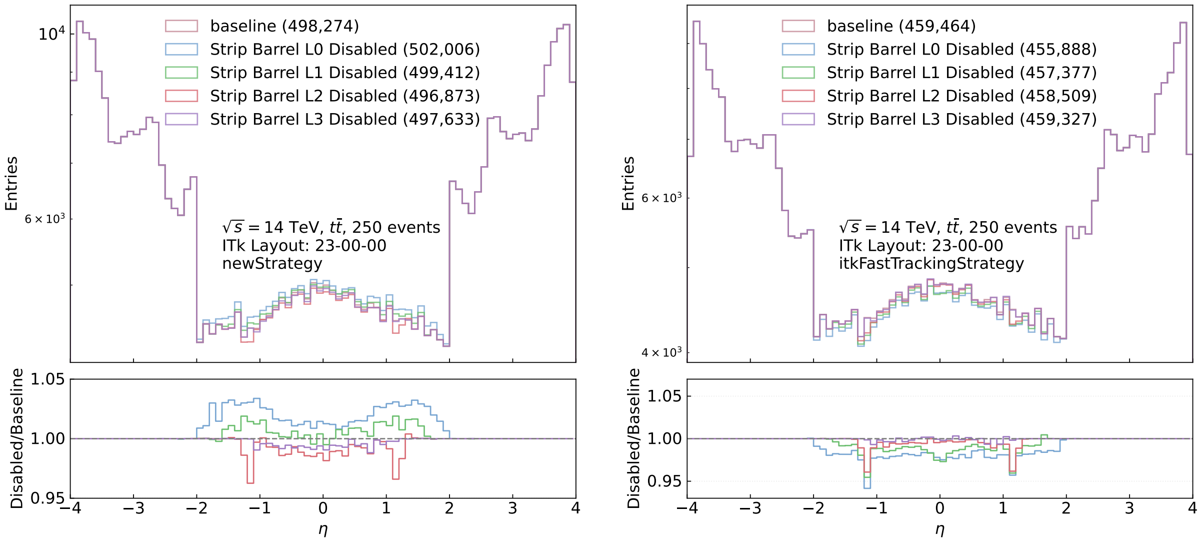
However, with the default reconstruction chain, some unexpected behaviors were observed: in certain cases, disabling specific strip detector layers resulted in a slight increase in the number of reconstructed tracks and tracking efficiency compared to the baseline. To better understand this counterintuitive outcome, this section also includes an analysis of seed usage based on the reconstruction logs.

4.4.1. Strip Barrel Layers

To evaluate the relative importance of each strip barrel layer, a layer-by-layer disabling approach was applied. Figure 4.33 compares the number of reconstructed tracks as a function of η when individual strip barrel layers are disabled, under both track reconstruction chains.

In Figure 4.33a, under the default track reconstruction chain, a somewhat counterintuitive behavior is observed: disabling Strip Barrel Layer 0 or Layer 1 results in a slightly higher number of reconstructed tracks compared to the baseline with the same strategy. This phenomenon was also seen in an independent $t\bar{t}$ sample with 500 events, suggesting a potentially reproducible effect.

In contrast, disabling any individual strip barrel layer always leads to a decrease in the number of reconstructed tracks with the fast track reconstruction chain, as shown in Figure 4.33b. The largest reductions occur when Layer 0 or Layer 1 is disabled, followed by Layer 2. For both track reconstruction chains, disabling Layer 3 produces the smallest deviation from the baseline. Additionally, removing Layer 2 consistently causes a dip in reconstructed track counts in the transition region (likely near the barrel-to-endcap overlap), with reductions reaching as low as 98% of the baseline in that η-bin.



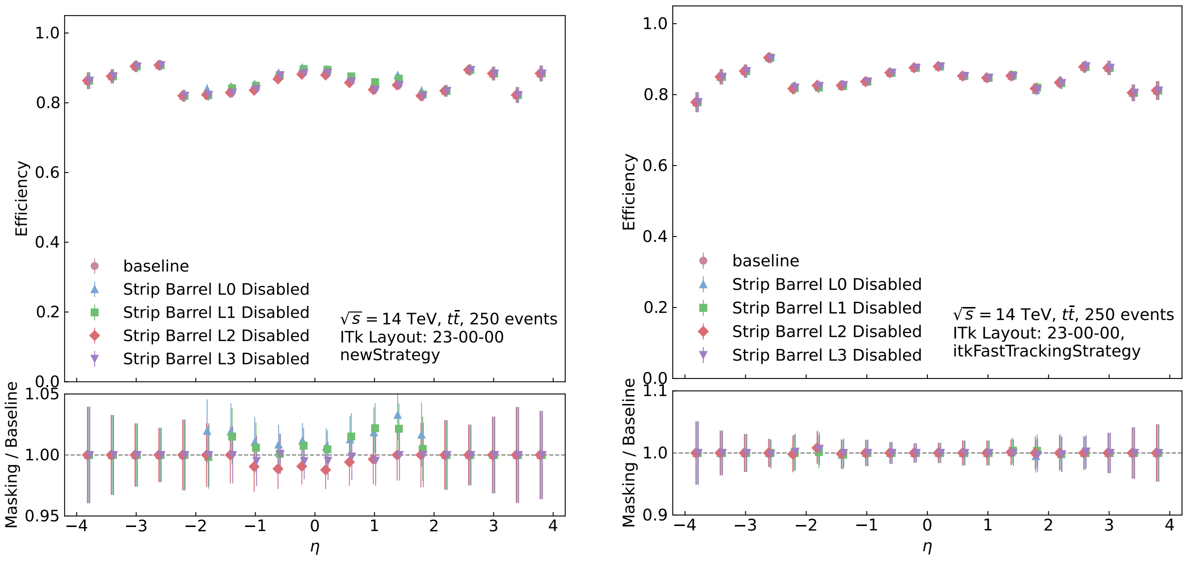
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.33.: η distributions of reconstructed tracks under individual strip barrel layer deactivations, from L0 to L3, for both reconstruction chains.

Figure 4.34 shows the η-binned histograms of tracking efficiency under different strip barrel layer disabled configurations, for both track recosntruction chain. For the default reconstruction chain, the trends resemble those observed in the number of reconstructed tracks: disabling Strip Barrel Layer 0 or Layer 1 leads to a slight, but counter-intuitive, increase in tracking efficiency in the barrel and transition regions, compared to the baseline; disabling Layer 2 or Layer 3 causes a minor decrease in efficiency, mostly within the barrel. For the fast track reconstruction chain, the tracking efficiency remains nearly unchanged regardless of which strip barrel layer is disabled.

Figure 4.35 presents the η-binned histograms of fake rate under different strip barrel layer disabled configurations.

Under the default chain, disabling Strip Barrel Layer 0 or Layer 1 leads to an increase

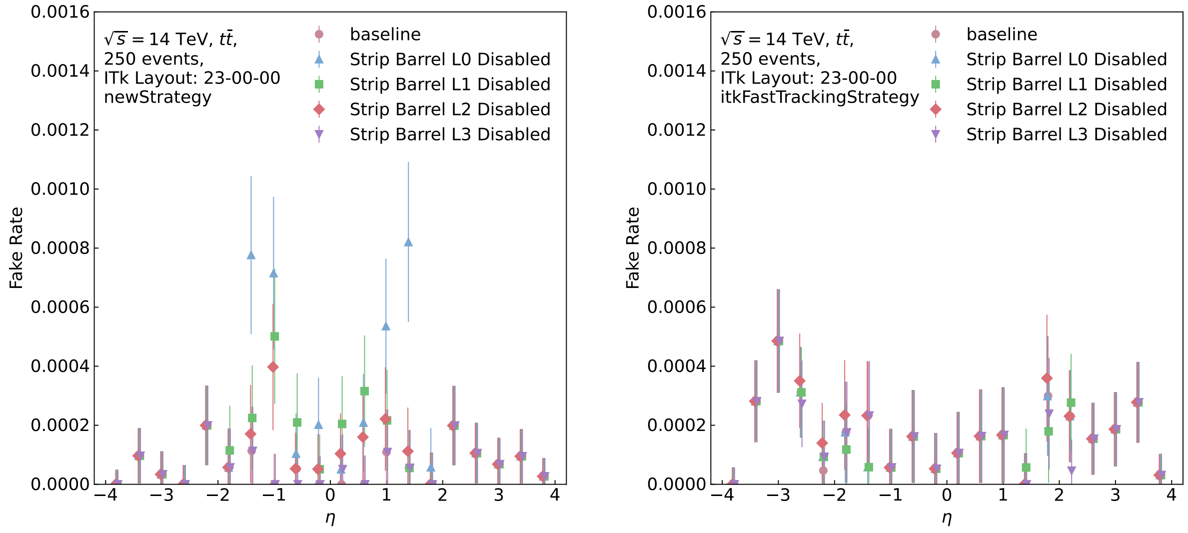


(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.34.: Tracking efficiency in η bins under individual strip barrel layer deactivations, from L0 to L3, for both reconstruction chains.

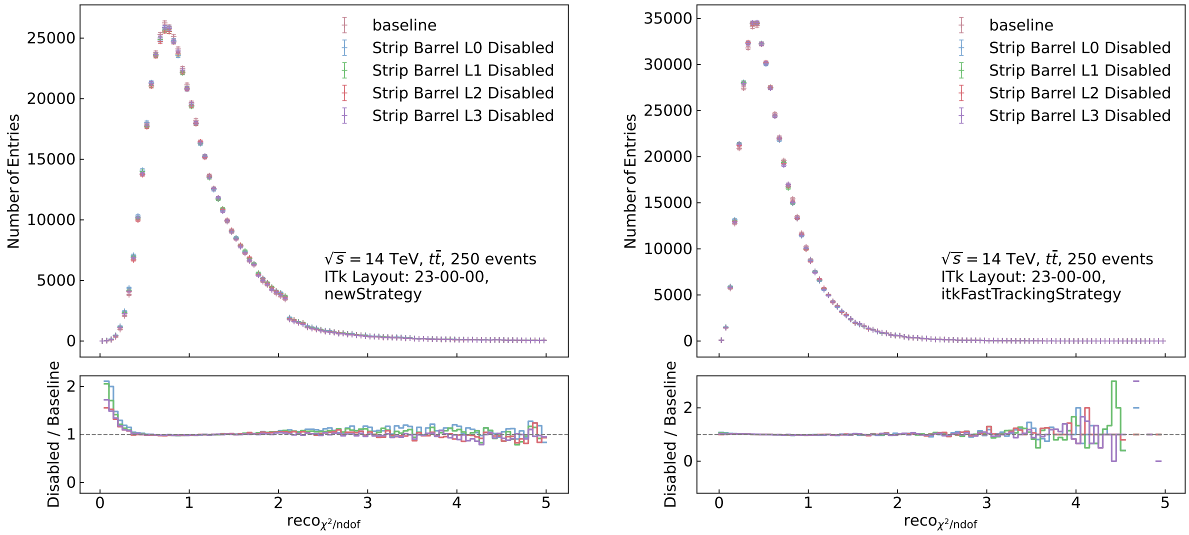
in fake rate in the barrel and transition regions, while the endcap and forward regions remain largely unaffected. Under the fast track reconstruction, disabling any individual strip barrel layer results in only a slight increase in fake rate, and this effect is confined to the endcap region. The barrel, transition, and forward regions show virtually no change. For both track reconstruction chains, the fake rate under all configurations remains within acceptable bounds.

The distributions of χ^2/N_{dof} remain stable when any individual strip barrel layer is disabled, with no systematic deviations observed relative to the baseline in either track reconstruction chain, as shown in Figures 4.36.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

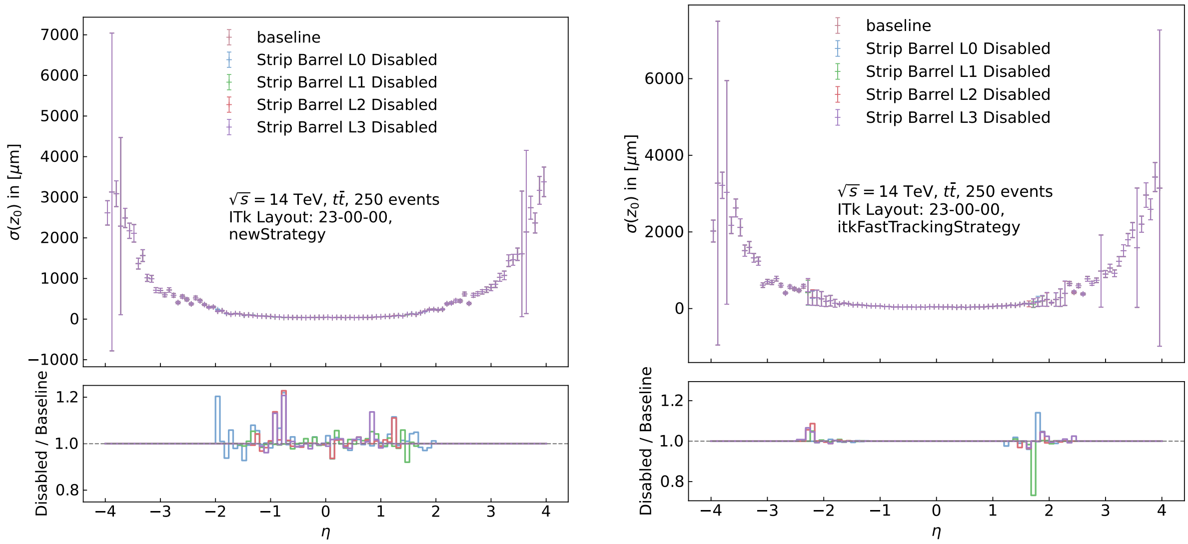
Figure 4.35.: Fake rate in η bins under individual strip barrel layer deactivations, from L0 to L3, for both reconstruction chains.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.36.: Distribution of χ^2/N_{dof} under individual strip barrel layer deactivations, from L0 to L3, for both reconstruction chains.

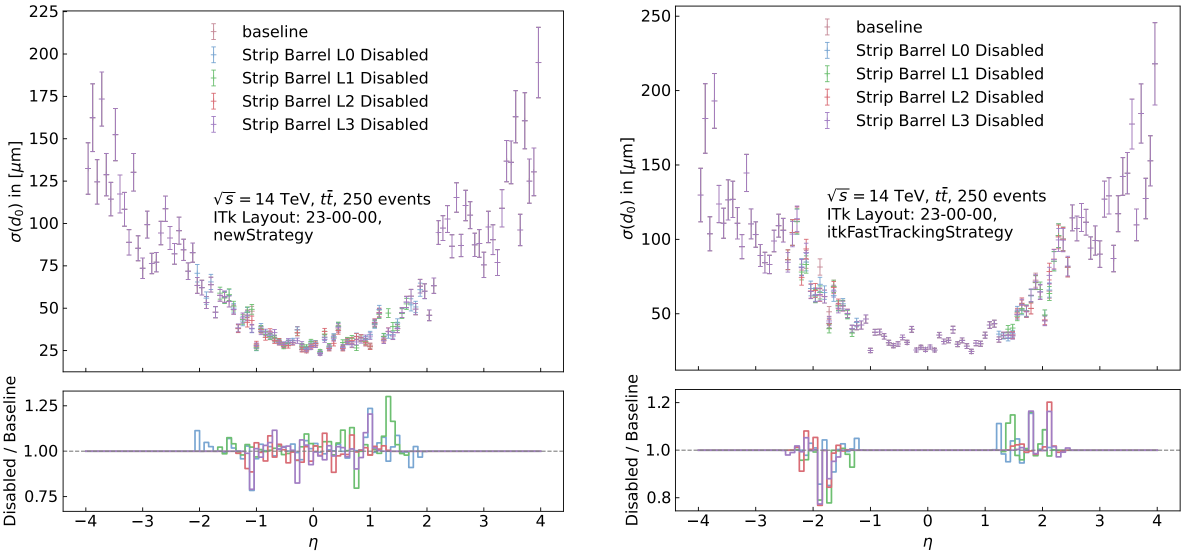
The $\sigma(z_0)$ also appears highly robust under all layer-level defect scenarios. This stability is especially pronounced with the fast track reconstruction chain, where almost no variation is observed across different configurations, as shown in Figure 4.37.



(a) The default reconstruction chain. ***(b)*** The fast track reconstruction chain.

Figure 4.37.: Distribution of z_0 resolution width under individual strip barrel layer deactivations, from L0 to L3, for both reconstruction chains.

The robustness of $\sigma(d_0)$ in both track reconstruction chains is slightly worse than that of $\sigma(z_0)$, as shown in Figures 4.38a and 4.38b. In both cases, deviations are observed in specific η regions: for the default track reconstruction chain, fluctuations are mainly concentrated within $|\eta| \in [0, 2]$, while for the fast track reconstruction chain, they are more pronounced in the forward region, $|\eta| \in [1, 2.5]$.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.38.: Distribution of d_0 resolution width under individual strip barrel layer deactivations, from L0 to L3, for both reconstruction chains.

For all key performance metrics, the impact of disabling any individual strip barrel layer is mainly limited to the region $\eta \in [0, 2]$ with the default reconstruction chain. This matches the geometrical coverage of the strip barrel. The fast track reconstruction chain also exhibits changes in the number of reconstructed tracks in this region, but tracking efficiency, fake rate, and the resolutions of d_0 and z_0 remain almost unchanged in the barrel region ($|\eta| \in [0, 1]$). Fluctuations, if any, are restricted to $|\eta| \in [1, 2.5]$.

Table 4.8 lists the seed usage and number of reconstructed tracks with the fast track reconstruction chain for the baseline and configurations where individual strip barrel layers are disabled. Table 4.9 presents the η-binned counts of seeds with tracks for the fast track reconstruction chain, showing the effect of disabling each strip barrel layer across four η intervals: [0,1), [1,2), [2,3), and [3,4].

Disabling any single strip barrel layer has no impact on the number of input PPP seeds in the fast track reconstruction chain and causes only minor fluctuations (less than 0.03%) in the number of used seeds. Furthermore, removing an individual strip barrel layer affects only the number of seeds with tracks within its η coverage region, while other η regions remain nearly unaffected. This behavior is listed in Table 4.9, where values that remain exactly the same as the baseline are marked in dark sepia.

Overall, disabling individual strip barrel layers has only a minor impact on the number of reconstructed tracks when using only PPP seeds in the fast track reconstruction chain.

The only slightly more noticeable case is when Strip Barrel Layer 0 is disabled, which results in a 0.78% reduction in the number of reconstructed tracks compared to the baseline.

Table 4.8.: Summary of PPP seed usage and number of reconstructed tracks with the fast track reconstruction chain for the baseline and each strip barrel layer disabled configuration.

Configuration	**Input Seeds**	**Used Seeds**	**Seeds with Track**	**Reconstructed Tracks**
Baseline	3,224,644	1,485,177	559,609	459,464
Strip Barrel L0 Disabled	3,224,644	1,485,407	558,999	455,888
Strip Barrel L1 Disabled	3,224,644	1,484,910	559,222	457,377
Strip Barrel L2 Disabled	3,224,644	1,484,801	558,535	458,599
Strip Barrel L3 Disabled	3,224,644	1,484,889	558,433	459,327

Table 4.9.: Number of PPP seeds with tracks in each $|\eta|$ range with the fast track reconstruction chain, for the baseline and each strip barrel layer disabled configuration. Although the overall variation is small compared to the baseline, values that remain exactly the same are marked in dark sepia. for clarity.

$|\eta|$ **Range**	**Baseline**	**L0 Disabled**	**L1 Disabled**	**L2 Disabled**	**L3 Disabled**
$[0.0, 1.0]$	114,129	113,664	114,059	113,620	113,274
$[1.0, 2.0]$	116,981	116,837	116,664	116,416	116,660
$[2.0, 3.0]$	150,226	150,225	150,226	150,226	150,226
$[3.0, 4.0]$	178,273	178,273	178,273	178,273	178,273

Table 4.10 lists the seeds usage and tracking results with the default reconstruction chain for baseline and each masking configuration, disabling any single strip barrel layer results in a clear reduction in the number of input SSS seeds: by 45.9% for L0, 38.7% for L1, 30.6% for L2, and 14.0% for L3, relative to the baseline. In particular, the impact is substantial when disabling the innermost layers (L0 and L1). To compensate, the reconstruction algorithm makes increased use of PPP seeds. The number of used PPP seeds increases by 2.83% for L1, 2.51% for L2, and 0.74% for L0. More notably, the number of PPP seeds associated with reconstructed tracks increases by 21.8% for L1, 18.1% for L2, 7.3% for L0, and 0.8% for L3.

This behavior reflects the design of the seeding strategy in the default reconstruction chain, where SSS seeds are prioritized during track finding, and PPP seeds are only used afterward. Since there is a global limit on the total number of seeds used, and both PPP and SSS seeds share a common counter, a decrease in SSS seed usage naturally leads to more PPP seeds being used.

Despite the fact that some layer-disabled configurations may ultimately yield more reconstructed tracks than the baseline, this is a result of the ambiguity solving stage compensating for initial losses. At the seeding stage, all disabled configurations produce fewer seeds with tracks than the baseline.

Table 4.10.: Summary of PPP and SSS seed usage and tracking results with the default reconstruction chain, for the baseline and each strip barrel layer disabled configuration. Values that increase compared to the baseline are highlighted in magenta, and values that decrease are highlighted in blue.

Config.	**Input PPP**	**Input SSS**	**Used PPP**	**Used SSS**	**SwTs PPP**	**SwTs SSS**	**Extra Trk PPP**	**Extra Trk SSS**	**Rec. Tracks**
Baseline	6,648,384	1,384,594	3,788,349	574,490	562,550	285,737	55,525	24,147	498,274
L0 Disabled	6,648,384	749,716	3,816,360	351,010	603,416	214,762	69,747	16,907	502,006
L1 Disabled	6,648,384	848,902	3,895,557	327,208	685,261	119,256	84,024	7,960	499,412
L2 Disabled	6,648,384	960,890	3,883,458	368,001	664,540	142,462	76,748	11,679	496,873
L3 Disabled	6,648,384	1,191,185	3,794,678	541,029	566,935	268,202	56,481	23,079	497,633

Note: All configurations refer to Strip Barrel layers. "Rec. Tracks" stands for Reconstructed Tracks.

It can be seen from comparing Table 4.11 and Table 4.9 that, with the fast track reconstruction chain, disabling any single strip barrel layer affects the number of seeds with tracks almost exclusively within the $|\eta| \in [0, 1]$ region. The impact on the $|\eta| \in [1, 2]$ region is already minimal, with the largest relative drop observed when Strip Barrel Layer 2 is disabled, resulting in a 0.48% reduction compared to the baseline.

In contrast, with the default reconstruction chain, disabling Strip Barrel Layer 0 leads to a much more significant reduction in the $|\eta| \in [1, 2]$ region, with the number of seeds with tracks decreasing by 7.81% compared to the baseline; even in the $|\eta| \in [2, 3]$ region, a measurable impact remains, with a 0.27% decrease. Meanwhile, in the fast track reconstruction chain, disabling any strip barrel layer has no impact on the number of seeds with tracks in the $|\eta| \in [3, 4]$ region.

Table 4.11.: Number of PPP seeds with track in each $|\eta|$ range with the fast track reconstruction chain, for the baseline and each strip barrel layer disabled configuration. Values that increase compared to the baseline are highlighted in magenta, and values that decrease are highlighted in blue. Values that show no change or negligible change compared to the baseline are highlighted in dark sepia.

$\lvert\eta\rvert$ Range	Baseline	L0 Disabled	L1 Disabled	L2 Disabled	L3 Disabled
$[0.0, 1.0]$	149,322	137,482	122,095	119,965	138,809
$[1.0, 2.0]$	225,415	207,821	208,851	213,473	222,771
$[2.0, 3.0]$	246,204	245,529	246,225	246,218	246,211
$[3.0, 4.0]$	227,346	227,346	227,346	227,346	227,346

Disabling any strip barrel layer suppresses the number of extra tracks originating from SSS seeds, while increasing those from PPP seeds, as shown Table 4.12.

Table 4.12.: Number of extra tracks produced with the default reconstruction chain, separated by PPP and SSS seed origin. Values that increase compared to the baseline are highlighted in magenta, and values that decrease are highlighted in blue.

Configuration	Extra Tracks from PPP Seeds	Extra Tracks from SSS Seeds	Total Extra Tracks
Baseline	55,525	24,147	79,672
Strip Barrel L0 Disabled	69,747	16,907	86,654
Strip Barrel L1 Disabled	84,024	7,960	91,984
Strip Barrel L2 Disabled	76,748	11,679	88,427
Strip Barrel L3 Disabled	56,481	23,079	79,560

Table 4.13 presents the ratio of the number of output tracks to the number of used seeds in η bins of width 0.5 with the fast track reconstruction chain. To make changes more visible, all values with no or negligible difference compared to the baseline are highlighted in dark sepia.

From this, it is clear that with the fast track reconstruction chain, disabling Strip Barrel Layer 2 or Layer 3 affects the ratio in the $|\eta| \in [0, 1.5]$ region. Considering that Strip Barrel Layer 0 and Layer 1 cover a larger solid angle, their influence extends up to $|\eta| \in [0, 2]$. However, even where differences exist, the deviations from the baseline are minimal, the largest relative drop is only about 0.85%.

Table 4.13.: Track / Used ratio of PPP seeds in each $|\eta|$ bin with the fast track reconstruction chain, for the baseline and each strip barrel layer disabled configuration. Values that show no change or negligible change compared to the baseline are highlighted in dark sepia..

Configuration	$[0.0, 0.5]$	$[0.5, 1.0]$	$[1.0, 1.5]$	$[1.5, 2.0]$	$[2.0, 2.5]$	$[2.5, 3.0]$	$[3.0, 3.5]$	$[3.5, 4.0]$
Baseline	0.3073	0.2341	0.2563	0.5211	0.4513	0.3269	0.4201	0.8587
L0 Disabled	0.3047	0.2336	0.2565	0.5190	0.4512	0.3269	0.4201	0.8587
L1 Disabled	0.3071	0.2343	0.2552	0.5199	0.4513	0.3269	0.4201	0.8587
L2 Disabled	0.3064	0.2329	0.2539	0.5211	0.4513	0.3269	0.4201	0.8587
L3 Disabled	0.3056	0.2321	0.2549	0.5211	0.4513	0.3269	0.4201	0.8587

However, for the default reconstruction chain, this ratio is significantly affected in the $|\eta| \in [0, 2]$ region, especially when strip barrel L1 or L2 is disabled: in these two cases, the algorithm effectively stops using SSS seeds in the $|\eta| \in [0, 1]$ range and instead relies entirely on PPP seeds. The relevant data are highlighted in red in Table 4.14. This behavior has also been independently confirmed in a separate MC sample of 500 $t\bar{t}$ events.

Disabling Strip Barrel L0 has a broader impact due to its larger solid angle. In particular, it is the only layer which removal slightly affects the $|\eta| \in [2.0, 2.5]$ bin, it is also the only configuration that leads to an overall increase in the track-to-used-seed ratio for both PPP and SSS seeds across the barrel and transition regions.

Table 4.14.: Track-to-used ratio of PPP and SSS seeds in each $|\eta|$ bin with the default reconstruction chain. Increases compared to the baseline are shown in magenta, decreases in blue, and values identical to the baseline are shown in dark sepia.. Configurations where SSS seeds are entirely unused (i.e., when Layer 1 or 2 is disabled) are marked with red background.

Seed Type	**Configuration**	$[0.0, 0.5]$	$[0.5, 1.0]$	$[1.0, 1.5]$	$[1.5, 2.0]$	$[2.0, 2.5]$	$[2.5, 3.0]$	$[3.0, 3.5]$	$[3.5, 4.0]$
PPP	Baseline	0.08758	0.06870	0.05348	0.06634	0.14910	0.20560	0.30920	0.65370
	L0 Disabled	0.10140	0.08061	0.07790	0.08228	0.14920	0.20560	0.30920	0.65370
	L1 Disabled	0.24940	0.17780	0.09754	0.07189	0.14910	0.20560	0.30920	0.65370
	L2 Disabled	0.24700	0.17300	0.08070	0.06638	0.14910	0.20560	0.30920	0.65370
	L3 Disabled	0.09907	0.07322	0.05360	0.06637	0.14910	0.20560	0.30920	0.65370
SSS	Baseline	0.67220	0.69950	0.68580	0.32140	0.47310	0.49120	-	-
	L0 Disabled	0.85790	0.87150	0.83680	0.44040	0.47670	0.49120	-	-
	L1 Disabled	0.00000	0.00000	0.58570	0.31010	0.47310	0.49120	-	-
	L2 Disabled	0.00000	0.00000	0.62440	0.32140	0.47310	0.49120	-	-
	L3 Disabled	0.69990	0.72900	0.69490	0.32230	0.47310	0.49120	-	-

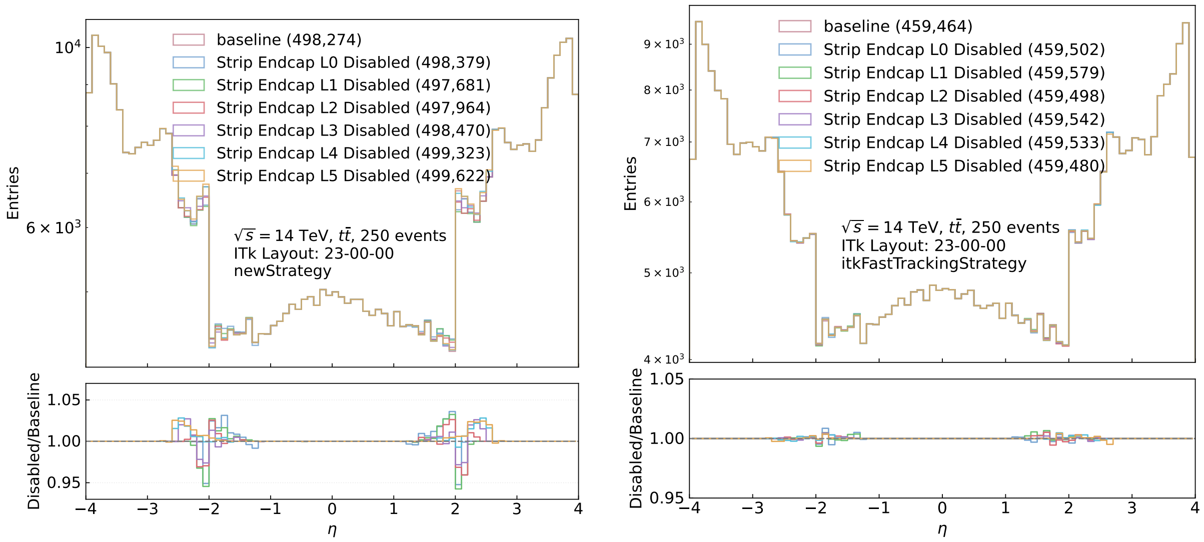
Based on the results above, it can be seen that the default reconstruction chain is inherently more sensitive to the removal of individual strip barrel layers. Disabling Strip Barrel Layer 1 or Layer 2 causes the algorithm to completely abandon the use of SSS seeds in the $|\eta| \in [0, 1]$ region and instead rely on PPP seeds, leading to a significant increase in extra tracks from PPP seeding. A similar but less severe compensation effect is also observed when Strip Barrel Layer 0 is disabled.

In contrast, the fast reconstruction chain shows limited sensitivity to the removal of any single strip barrel layer. The most notable impact occurs when Strip Barrel Layer 0 or Layer 1 is disabled.

4.4.2. Strip Endcap Disks

A similar layer-by-layer masking study was also performed for the strip endcap detector. Figure 4.39 shows the distribution of the number of reconstructed tracks vs η under the baseline configuration and with individual strip endcap layers disabled, for both reconstruction chains. Under both reconstruction chains, the impact of disabling a single strip endcap layer is localized to the region $|\eta| \in [1, 3]$.

For the default reconstruction chain, disabling any strip endcap layer leads to an increase in the number of reconstructed tracks relative to the baseline within $|\eta| \in [1, 2]$, which roughly corresponds to the transition region between the barrel and endcap. A localized dip is observed near $|\eta| \approx 2$, after which the track count rises again in the range $|\eta| \in [2, 3]$ and remains above the baseline. Meanwhile, the fast reconstruction chain exhibits much smaller variations: the number of reconstructed tracks fluctuates within approximately $\pm 1\%$ of the baseline across the entire $|\eta| \in [1, 3]$ region, indicating that this reconstruction chain is less sensitive to the removal of individual strip endcap layers, just as it is with the masking of individual strip barrel layers.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

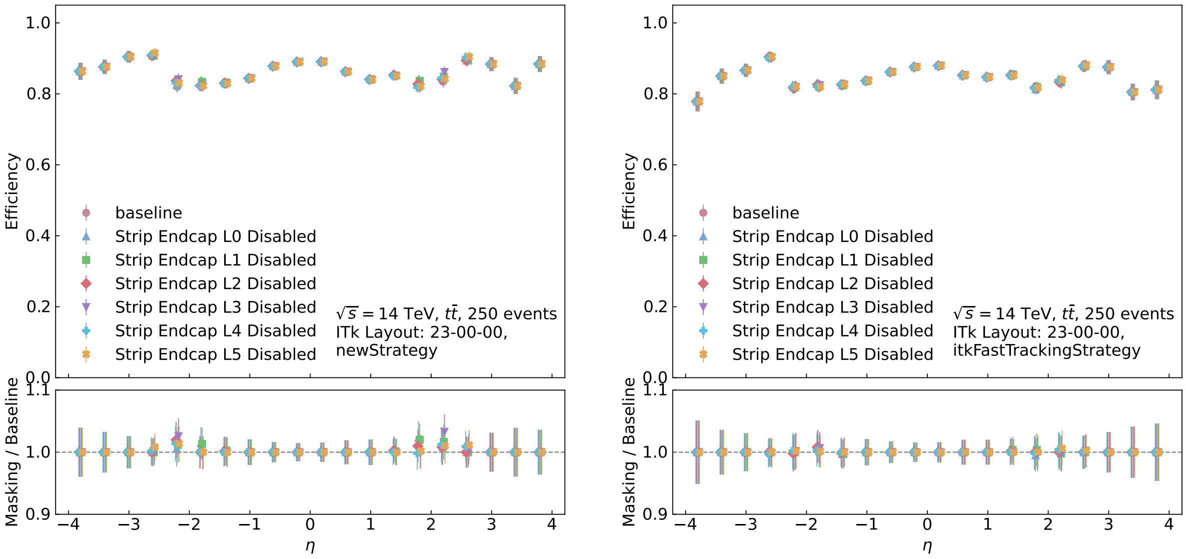
Figure 4.39.: η distributions of reconstructed tracks under individual strip endcap layer deactivations, from L0 to L5, for both reconstruction chains.

Figure 4.40 shows the distribution of tracking efficiency versus η under the baseline configuration and with individual strip endcap layers disabled, for both reconstruction chains.

For tracking efficiency, the impact of disabling any single strip endcap layer is minimal under both reconstruction chains. Across the barrel, transition, and forward regions, the efficiency remains nearly unchanged compared to the baseline. However, a very slight increase is observed near $|\eta| \approx 2$ across all defect configurations, particularly with the default reconstruction chain.

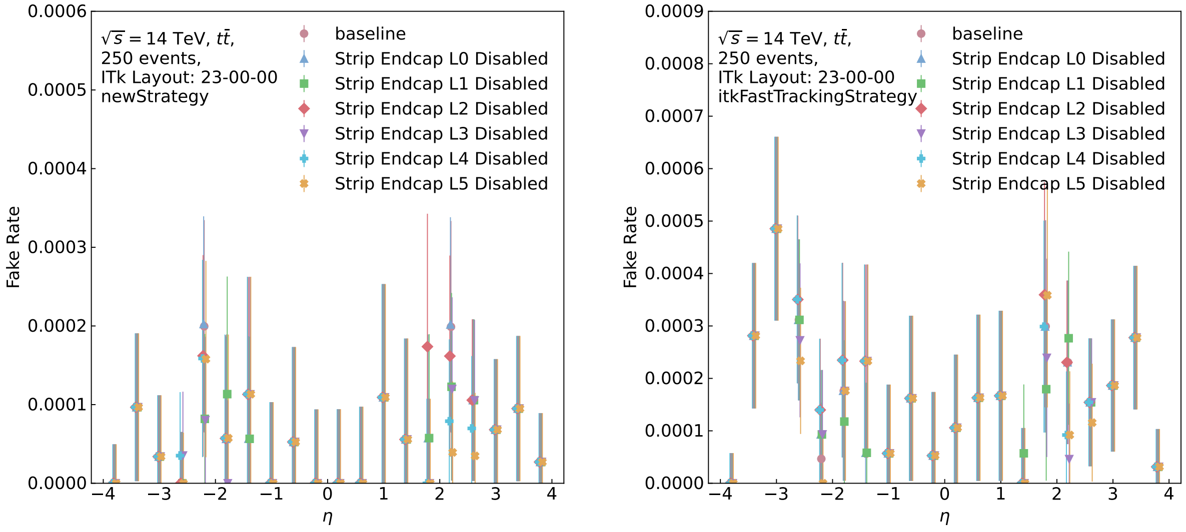
Figure 4.41 shows histograms of the fake rate as a function of η under various masking configurations, for both track reconstruction chains. Disabling any individual strip endcap layer leads to only minor changes under both track reconstruction chains. These changes remain small in magnitude and are confined to the endcap region, consistent with the η coverage of the corresponding strip endcap detectors, that is, the region where part of the detector has been intentionally disabled.

No significant or systematic shift is observed in the χ^2/N_{dof} distributions when disabling any single strip endcap layer under either track reconstruction chains, as shown in Figure 4.42.



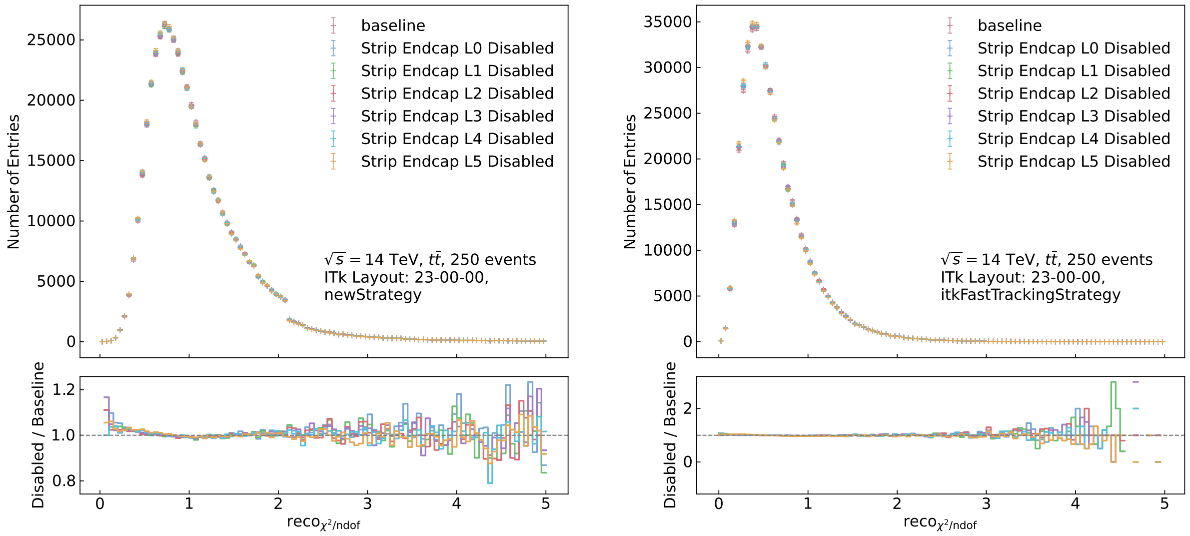
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.40.: Tracking efficiency in η bins under individual strip barrel layer deactivations, from L0 to L5, for both reconstruction chains.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

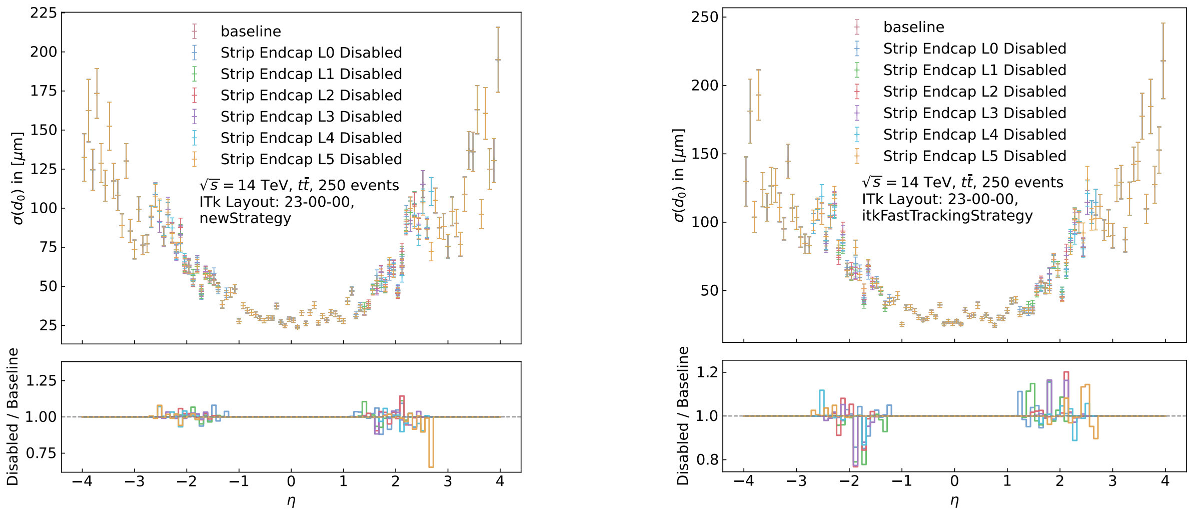
Figure 4.41.: Fake rate in η bins under individual strip barrel layer deactivations, from L0 to L5, for both reconstruction chains.



(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure 4.42.: Distribution of χ^2/N_{dof} under individual strip barrel layer deactivations, from L0 to L5, for both reconstruction chains.

For the $\sigma(d_0)$ resolution, small fluctuations are observed in the region $|\eta| \in [1,3]$, which corresponds to the coverage of the strip endcap detectors. These fluctuations occur in both reconstruction chains, with variations up to approximately ±20%. As shown in Figure 4.43, the deviations with the fast track reconstruction chain tend to be slightly larger than those with the default reconstruction chain, but the overall scale remains limited.

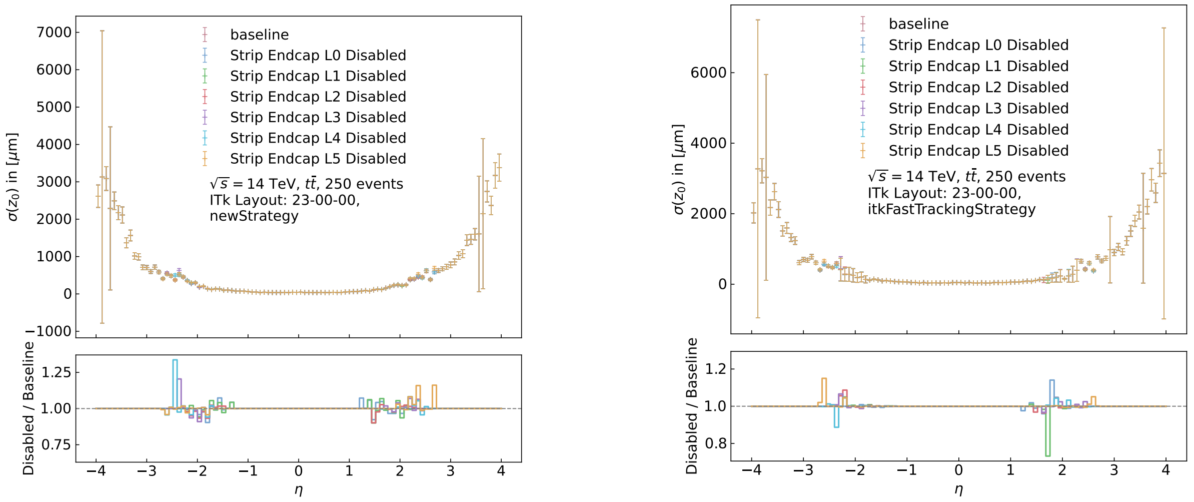


(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

Figure 4.43.: Distribution of d_0 resolution width under individual strip endcap layer deactivations, from L0 to L5, for both reconstruction chains.

Similarly, the $\sigma(z_0)$ resolution shows fluctuations within the same η range for both chains, again within roughly ±20%, as seen in Figure 4.44. These variations are expected due to the intentional disabling of strip endcap layers in this region.



(a) With the default reconstruction chain.

(b) With the fast track reconstruction chain.

Figure 4.44.: Distribution of z_0 resolution width under individual strip endcap layer deactivations, from L0 to L5, for both reconstruction chains.

Table 4.15 summarizes the seeding usage with the default reconstruction chain when individual strip endcap layers are disabled. Disabling any single strip endcap layer consistently leads to a reduction in the number of input SSS seeds. Similar to the behavior observed when disabling strip barrel layers, this reduction triggers a compensatory increase in the use of PPP seeds, due to the insufficiency of viable SSS seeds. The effect is particularly notable when inner endcap layers (L0, L1, or L2) are disabled, as they contribute more significantly to SSS seed formation.

Regardless of which strip endcap layer is masked, the total number of track candidates formed during the seeding stage is always lower than the baseline. However, due to the influence of the ambiguity solving stage in the default reconstruction chain, the final number of reconstructed tracks slightly exceeds the baseline when Endcap Layers L0, L3, L4, or L5 are disabled.

Table 4.15.: Summary of PPP and SSS seed usage and tracking results with the default reconstruction chain, for the baseline and each strip endcap layer disabled configuration. Values that increase compared to the baseline are highlighted in magenta, values that decrease are highlighted in blue, and values identical to the baseline are shown in dark sepia.

Configuration	**Input PPP**	**Input SSS**	**Used PPP**	**Used SSS**	**SwTs PPP**	**SwTs SSS**	**Output PPP**	**Output SSS**	**Reconstructed Tracks**
Baseline	6,648,384	1,384,594	3,788,349	574,490	562,550	285,737	618,075	309,884	498,274
Strip Endcap L0	6,648,384	957,402	3,802,245	433,517	580,539	243,072	642,743	263,939	498,379
Strip Endcap L1	6,648,384	916,733	3,800,409	425,971	578,075	238,786	639,706	260,273	497,681
Strip Endcap L2	6,648,384	931,536	3,797,398	431,008	573,220	244,961	632,514	267,340	497,964
Strip Endcap L3	6,648,384	1,016,567	3,794,260	458,710	568,270	253,869	626,044	276,952	498,470
Strip Endcap L4	6,648,384	1,122,823	3,790,690	484,360	563,609	262,492	619,791	286,181	499,323
Strip Endcap L5	6,648,384	1,126,383	3,788,574	482,956	559,601	264,053	615,014	288,039	499,622

For the fast reconstruction chain, disabling any individual strip barrel layer results in a very slight increase at most 0.025% in the number of reconstructed tracks, as summarized in Table 4.16. In some cases, layers whose removal leads to fewer used seeds and seeds with tracks still end up producing more tracks compared to the baseline. This may suggest that the conversion efficiency from used seeds to reconstructed tracks can marginally improve under certain defective configurations even in the fast reconstruction chain. However, given the small magnitude of these differences, they are likely within the range of statistical fluctuations and should not be overinterpreted.

Table 4.16.: Summary of PPP seed usage and number of reconstructed tracks with the fast track reconstruction chain for the baseline and each strip endcap layer disabled configuration. Values that increase compared to the baseline are highlighted in magenta, values that decrease are highlighted in blue, and values identical to the baseline are shown in dark sepia.

Configuration	Input Seeds	Used Seeds	Seeds with Track	Reconstructed Tracks
Baseline	3,224,644	1,485,177	559,609	459,464
Strip Endcap L0	3,224,644	1,485,213	559,644	459,502
Strip Endcap L1	3,224,644	1,485,182	559,641	459,579
Strip Endcap L2	3,224,644	1,484,912	559,628	459,498
Strip Endcap L3	3,224,644	1,484,734	559,520	459,542
Strip Endcap L4	3,224,644	1,484,236	559,514	459,533
Strip Endcap L5	3,224,644	1,483,708	558,790	459,480

Table 4.17 shows that for the default reconstruction chain, disabling any individual strip endcap layer except for L5 leads to an increase in the number of extra tracks originating from PPP seeds. The magnitude of this increase appears to correlate with the degree to which PPP seeds are used to compensate for the reduction in SSS seeds. At the same time, disabling any layer consistently reduces the number of extra tracks originating from SSS seeds. Overall, all configurations except for the one with strip endcap L5 disabled result in a higher total number of extra tracks. Considering that each seed is expected to correspond to at most one physical track, these extra tracks are subject to further filtering in the ambiguity solving stage.

Table 4.17.: Number of extra tracks produced with the default reconstruction chain, separated by PPP and SSS seed origin.Values that increase compared to the baseline are highlighted in magenta, and values that decrease are highlighted in blue.

Configuration	Extra Tracks from PPP Seeds	Extra Tracks from SSS Seeds	Total Extra Tracks
Baseline	55,525	24,147	79,672
Strip Endcap L0	62,204	20,867	83,071
Strip Endcap L1	61,631	21,487	83,118
Strip Endcap L2	59,294	22,379	81,673
Strip Endcap L3	57,774	23,083	80,857
Strip Endcap L4	56,182	23,689	79,871
Strip Endcap L5	55,413	23,986	79,399

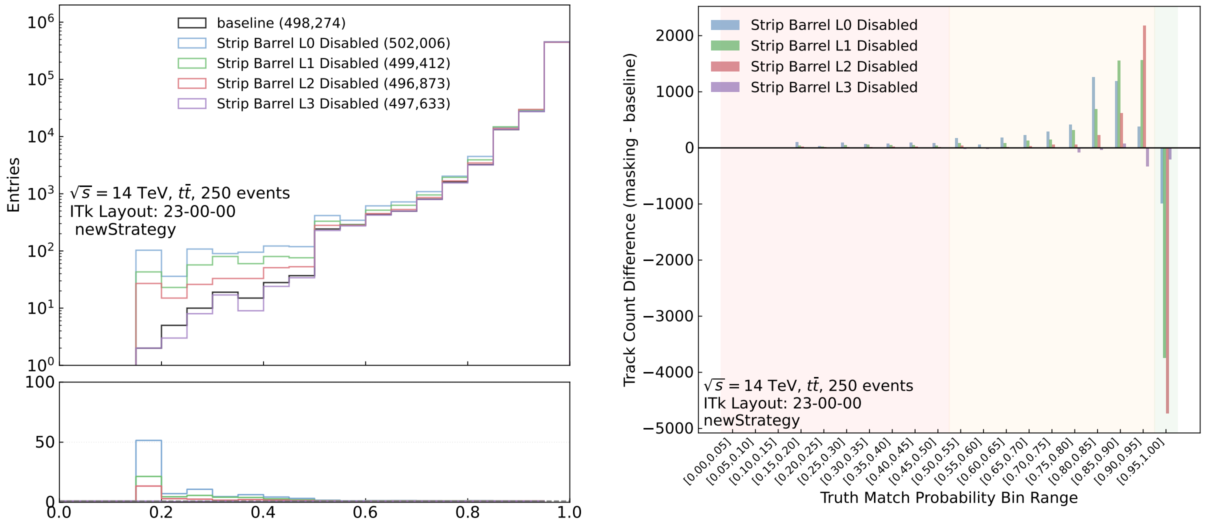
As shown in Table 4.18, disabling any of the inner strip endcap layers (L0, L1, or L2) with the default reconstruction chain leads to a simultaneous increase in the PPP and SSS track-to-used ratios within the corresponding $|\eta|$ regions. When any of the outer endcap layers (L3, L4, or L5), which cover the more forward regions are disabled, in these regions, the reconstruction chain tends to entirely abandon the use of SSS seeds. This behavior is highlighted in red background.

Table 4.18.: Track / Used ratio of PPP and SSS seeds in each $|\eta|$ bin with the default reconstruction chain, for the baseline and each strip endcap layer disabled configuration. Increases compared to the baseline are highlighted in magenta, decreases in blue, unchanged values are shown in dark sepia, and zero values are marked with red background.

Seed Type	**Configuration**	$[0.0, 0.5]$	$[0.5, 1.0]$	$[1.0, 1.5]$	$[1.5, 2.0]$	$[2.0, 2.5]$	$[2.5, 3.0]$	$[3.0, 3.5]$	$[3.5, 4.0]$
PPP	Baseline	0.08758	0.06870	0.05348	0.06634	0.14910	0.20560	0.30920	0.65370
	L0 Disabled	0.08760	0.06873	0.06007	0.07685	0.15370	0.20560	0.30920	0.65370
	L1 Disabled	0.08760	0.06872	0.05466	0.07510	0.15880	0.20560	0.30920	0.65370
	L2 Disabled	0.08758	0.06871	0.05327	0.07058	0.15950	0.20560	0.30920	0.65370
	L3 Disabled	0.08759	0.06870	0.05346	0.06793	0.15540	0.20560	0.30920	0.65370
	L4 Disabled	0.08758	0.06870	0.05348	0.06606	0.15140	0.20500	0.30920	0.65370
	L5 Disabled	0.08757	0.06870	0.05348	0.06591	0.14770	0.20240	0.30920	0.65370
SSS	Baseline	0.67220	0.69950	0.68580	0.32140	0.47310	0.49120	-	-
	L0 Disabled	0.67220	0.69950	0.72240	0.37750	0.52240	0.49120	-	-
	L1 Disabled	0.67220	0.69950	0.71560	0.37270	0.51770	0.49120	-	-
	L2 Disabled	0.67220	0.69950	0.69860	0.40820	0.48590	0.49120	-	-
	L3 Disabled	0.67220	0.69950	0.68610	0.38940	0.48770	0.00000	-	-
	L4 Disabled	0.67220	0.69950	0.68580	0.37640	0.49130	0.00000	-	-
	L5 Disabled	0.67220	0.69950	0.68580	0.36480	0.58530	0.00000	-	-

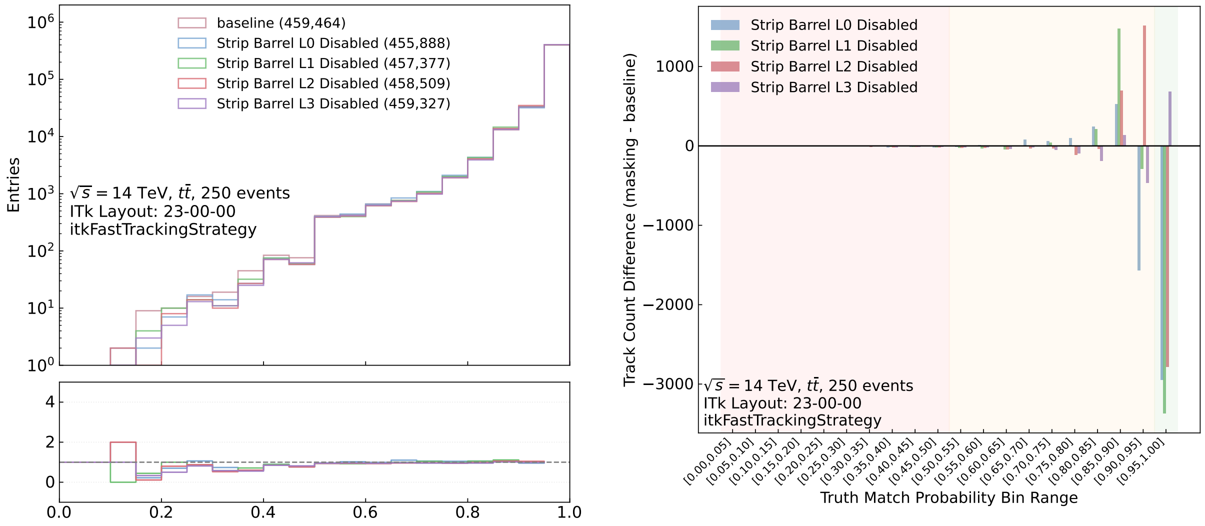
For the fast reconstruction chain, the track-to-used ratio remains largely stable across all $|\eta|$ bins, regardless of which strip endcap layer is disabled. Only minor fluctuations are observed, and they are confined to the $|\eta|$ intervals covered by the specific disabled layer, as shown in Table 4.19.

The TMP distributions under single-layer masking of the strip barrel are shown in Figure 4.45. Both reconstruction chains exhibit a consistent trend: when strip barrel layers L0 and L1 are disabled, the number of tracks with TMP in the range $[0.95, 1]$ decreases compared to the baseline, while the number of tracks with lower TMP values increases. The key difference is that, under the default reconstruction chain, more tracks fall into the low-TMP region, particularly in the range $[0.15, 0.5]$, these tracks are more likely to be classified as fake, as shown in Figure 4.45b. In contrast, this trend is not observed with the fast track reconstruction chain, as illustrated in Figure 4.45d.



(a) TMP count distribution with 0.5 bin width (with the default reconstruction chain).

(b) Difference in TMP counts per 0.5 bin: masking configuration − baseline (with the default reconstruction chain).



(c) TMP count distribution with 0.5 bin width (with fast track reconstruction chain).

(d) Difference in TMP counts per 0.5 bin: masking configuration − baseline (with fast track reconstruction chain).

Figure 4.45.: Comparison of TMP-related distributions under single-layer masking of the strip barrel detector. The top row shows results with the default reconstruction chain, while the bottom row corresponds to the fast reconstruction chain. Left: overall TMP count distribution in bins of width 0.5. Right: bin-wise difference in TMP counts between the masking configuration and the baseline.

Table 4.19.: Track-to-used ratio of PPP seeds in each $|\eta|$ bin with the fast track reconstruction chain, for the baseline and for each configuration in which a strip endcap layer is disabled. Values that increase compared to the baseline are highlighted in magenta, those that decrease are highlighted in blue, and values identical to the baseline are shown in dark sepia.. The observed variations are very small and entirely localized in the endcap region.

Configuration	$[0.0, 0.5]$	$[0.5, 1.0]$	$[1.0, 1.5]$	$[1.5, 2.0]$	$[2.0, 2.5]$	$[2.5, 3.0]$	$[3.0, 3.5]$	$[3.5, 4.0]$
Baseline	0.3073	0.2341	0.2563	0.5211	0.4513	0.3269	0.4201	0.8587
L0 Disabled	0.3073	0.2341	0.2559	0.5218	0.4511	0.3269	0.4201	0.8587
L1 Disabled	0.3073	0.2341	0.2564	0.5207	0.4512	0.3269	0.4201	0.8587
L2 Disabled	0.3073	0.2341	0.2562	0.5215	0.4519	0.3269	0.4201	0.8587
L3 Disabled	0.3073	0.2341	0.2563	0.5225	0.4510	0.3269	0.4201	0.8587
L4 Disabled	0.3073	0.2341	0.2563	0.5228	0.4516	0.3272	0.4201	0.8587
L5 Disabled	0.3073	0.2341	0.2563	0.5218	0.4504	0.3262	0.4201	0.8587

Overall, the impact of disabling a single strip endcap layer on track reconstruction is more complex than a simple degradation in performance. Due to the algorithmic design of the reconstruction chain, such defects do not necessarily lead to a reduction in the number of reconstructed tracks or the tracking efficiency.

With the default reconstruction chain, disabling any individual strip layer consistently reduces the number of available SSS seeds. As a result, the algorithm compensates by relying more heavily on PPP seeds. In some cases, the affected $|\eta|$ region becomes entirely dependent on PPP seeds. Interestingly, although certain defect configurations yield more tracks at the seeding stage than the baseline, the subsequent ambiguity solving stage can further promote some of these tracks, leading to an overall increase in the final number of reconstructed tracks and tracking efficiency.

However, ATHENA's default fake rejection criterion is based on a TMP threshold of 0.5. This raises questions about whether a track with TMP just above 0.5 is necessarily a genuine one, and whether a track with TMP just below 0.5 is necessarily fake. Analysis of the TMP distribution shows that disabling any single strip layer (except endcap L5) tends to shift tracks from the highest TMP bin ($[0.95, 1.0]$) into the next-highest bin, which is still above the 0.5 threshold. In some cases, the number of tracks in this next-highest bin exceeds the number lost from the top bin, resulting in a net increase in accepted tracks compared to the baseline.

This observation does not necessarily indicate a real improvement in performance, especially considering that most defect configurations cause an increase in extra tracks from PPP seeds. These extra tracks may include fake or duplicate tracks. The most prominent anomalous behavior is observed when disabling strip barrel L0 or L1, where both tracking efficiency and fake rate increase simultaneously within the region $|\eta| \in [0, 2]$. This behavior is difficult to interpret and suggests that these two innermost barrel layers, which cover a large solid angle, should be treated with particular caution with the default reconstruction chain.

A particularly interesting case is strip endcap L5. It is the only configuration with the default reconstruction chain where disabling the layer results in a decrease in the number of tracks with TMP in $[0.5, 0.95]$, accompanied by an increase in tracks with TMP in $[0.95, 1.0]$. Furthermore, disabling L5 reduces the number of extra tracks from both PPP and SSS seeds, which is not observed in any other configuration.

In contrast, the fast reconstruction chain is significantly more robust to strip layer defects. Since it exclusively relies on PPP seeds, the track-to-used ratio remains stable across all $|\eta|$ bins. Only very minor fluctuations are observed, and these are confined to the region covered by the disabled layer. In the strip barrel region, where redundancy is high, the results follow intuitive expectations. In the strip endcap region, although the number of reconstructed tracks can occasionally exceed the baseline in this sample, the behavior is more consistent with statistical fluctuations than with any systematic trend.

Regarding resolution-related metrics, such as χ^2/ndof and the widths of the z_0 and d_0 resolution distributions, no significant degradation is observed in either reconstruction chain after disabling any individual strip layer. Notably, under the fast reconstruction chain, $\sigma(z_0)$ and $\sigma(d_0)$ remain almost unchanged in the $|\eta| \in [0, 1]$ region when strip barrel layers are disabled. Since this region is mainly covered by the pixel barrel, this result suggests that resolution performance is primarily driven by the pixel detector.

The author is uncertain about how effective the ambiguity solving stage in the default reconstruction chain is at suppressing duplicate tracks. This uncertainty arises from the fact that the seeding stage allows a single seed to correspond to multiple track candidates. In principle, the corresponding analysis files contain a branch of type `vector<ElementLink<DataVector<xAOD::TruthParticle_v1> > >`, which records the number of reconstructed tracks associated with each truth particle. However, due to the complexity of this branch structure and the limited time available for this thesis project, the author was unable to fully unpack and analyze it.

In contrast, the algorithmic design of the fast reconstruction chain inherently avoids such cases, as its track finding and ambiguity resolution procedures are more restrictive by construction.

5. Summary and Conclusion

This study explores two complementary dimensions of tracking robustness for the ATLAS ITk detector at the HL-LHC. The first focuses on comparing two reconstruction chains: the default and the fast reconstruction chains under identical detector defect conditions. The second examines how different defect configurations, such as randomly disabling individual pixel cells, entire pixel modules, or specific detector regions, impact tracking performance.

The observed differences between the two reconstruction chains stem from their distinct algorithmic designs. The default chain allows a certain degree of ambiguity during the seeding stage, permitting multiple track candidates per seed ("extra tracks"), and resolves conflicts through a dedicated ambiguity solving stage. This design enables it to consistently reconstruct more tracks, achieve higher tracking efficiency, and maintain a lower fake rate across all tested configurations, including both ideal and partially degraded detectors: random masking of pixel cells and modules at 2%, 5%, 8%, 12%, 16%, and 20%, applied per layer across the full pixel and strip detector system. Remarkably, even with 12% of pixel modules randomly disabled, the default chain outperforms the fast chain running on an ideal detector in terms of both track count and efficiency, as seen in Figure 4.13.

However, a limitation of the default chain is its strong dependence on the innermost pixel barrel layer (pixel barrel L0) for accurate d_0 and z_0 reconstruction, as seen in Figures 4.22a and 4.23a. Since pixel barrel L0 is located in the highest-radiation zone, this dependence may pose a risk. Nevertheless, the planned LS replacement of pixel barrel L0 and L1 during the HL-LHC phase may mitigate this concern.

While the default chain performs better than the fast chain in a $t\bar{t}$ sample of 250 events, this alone does not imply that it is simply the superior algorithm. The fast reconstruction chain was designed with speed and computational efficiency as its primary goals, rather than maximum reconstruction performance. As discussed in Section 3.2.2, a head-to-head comparison using ideal geometry shows that although the fast chain reconstructs slightly fewer tracks, it dramatically reduces computational cost. This trade-off is particularly important in the high-luminosity, high-pileup conditions expected at the HL-LHC, where fast and efficient tracking is essential. Runtime logs from this study confirm that the number of input seeds in the fast chain is often less than 40% of that in the default chain, highlighting its resource efficiency.

The overall conclusion is therefore balanced. The default reconstruction chain offers high precision and strong robustness against pixel degradation, albeit with significantly higher computational cost. Thanks to ITk's design redundancy, it can still reconstruct 94% of the baseline track count even with 30% of quad modules and 10% of triplet modules disabled, with only moderate degradation in tracking efficiency and fake rate, as discussed in Section 4.3.3. Of course, this level of degradation would likely be unacceptable in practice.

In contrast, the fast reconstruction chain trades off some reconstruction quality in favor of lower latency and computational demand. Ref. [12] explored whether a partially installed pixel detector could be used for early data taking in the case of delayed module delivery or poor production yield. The conclusion was negative, and this study supports that finding: fast reconstruction is highly sensitive to pixel detector failures, particularly shown in Sections 4.2.1, 4.2.2, and 4.3.1.

Ultimately, there is no perfect solution, only a trade-off between hardware reliability and algorithmic complexity. Either the detector must be produced and operated to the highest standards, or the reconstruction algorithms, especially the fast chain must be improved to better tolerate non-ideal conditions. Accepting higher computational cost may also be a necessary compromise.

As the author is not familiar with the technical aspects of detector fabrication, the discussion focuses exclusively on reconstruction algorithms.

The two reconstruction chains differ significantly in two stages: seeding and post-seeding processing. At the seeding stage, two key differences emerge: the order of seed usage across detector regions (e.g., SSS vs. PPP), and how shared clusters among track candidates are handled. It is intuitive that partial pixel damage impacts the fast chain more severely, since it relies solely on PPP seeds, while the default chain, which prioritizes SSS seeds, is less affected.

Interestingly, in the forward region, i.e. $|\eta| \in [2.5, 4.0)$, where no strip detectors are available and both chains must rely solely on PPP seeds, the default chain still outperforms the fast chain under baseline conditions. This may be due to the default chain's more flexible handling of shared clusters, which becomes particularly advantageous in high-occupancy regions where track overlap is common.

The efficiency of resource usage also varies across η regions. Some regions are more sensitive to resource constraints, while others show diminishing returns from additional effort. For instance, in the range $|\eta| \in [1, 2]$ with an ideal detector, both chains reconstruct a similar number of tracks, but the default chain uses nearly twice as many seeds with tracks. In contrast, in the forward region, even when the number of seeds with tracks is similar, the final reconstruction outcomes differ significantly (Figure 4.3a).

Assuming no major changes to the fast reconstruction algorithm, a region-dependent strategy could be considered. For example, in regions like $|\eta| \in [1, 2]$, where the benefit of extra reconstruction effort is limited, the fast chain may be sufficient. In contrast, in the forward region or areas with concentrated damage, selectively enabling the default chain could improve performance. If the Detector Control System can provide real-time detector status, a damage-aware reconstruction strategy may also be feasible.

This study also includes a comparison between two masking strategies that result in similar total masked detector area: randomly disabling individual pixel cells vs. disabling entire pixel modules. The goal is to evaluate whether the spatial pattern of defects, even under the same overall masking fraction, leads to different reconstruction outcomes.

For the fast track reconstruction chain, cell-level masking generally leads to better performance than module-level masking at the same defect fraction. In the region $|\eta| \in [0.5, 1]$, the number of reconstructed tracks remains largely unchanged even when 20% of pixel cells are randomly disabled. In contrast, module-level masking causes a much sharper degradation in the barrel region. In the forward region, however, cell-level masking shows slightly worse performance than module-level masking. Additionally, at 20% masking, the fake rate under module-level masking becomes slightly higher than that under cell-level masking.

For the default reconstruction chain, cell-level masking yields more reconstructed tracks than module-level masking at low defect fractions (below 5%). However, as the masking fraction increases from 5% to 20%, module-level masking begins to outperform cell-level masking, resulting in higher tracking efficiency and lower fake rate at the same defect levels.

An important observation is that the number of input seeds is much more resilient under cell-level masking. When pixel cells are randomly disabled, the number of input seeds decreases only slightly for both reconstruction chains, as shown in Tables 4.2 and 4.3. In contrast, module-level masking leads to a significant drop in seed availability, as seen in Tables 4.4 and 4.5.

These results suggest that, if a small fraction of the pixel detector is expected to be non-functional, distributing the defects at the pixel-cell level may be more favorable than disabling full modules. In other words, having all modules operational but partially degraded may yield better tracking performance than having some modules entirely nonoperational.

An unintuitive result was observed in the layer-by-layer strip detector masking tests: under the default reconstruction chain, disabling strip barrel layer L0 or L1 led to a slight increase in the number of reconstructed tracks and tracking efficiency in the barrel region. This behavior was not observed in the fast track reconstruction chain.

The underlying cause lies in the seed usage priority of the default chain. The total number of seeds is constrained by a fixed computational budget, and SSS seeds are given priority during seed generation. When SSS seed formation is reduced, such as when key strip layers are disabled, the unused quota is effectively allocated to PPP seeds, which become more abundant as a result. Although many of these additional tracks satisfy the TMP threshold of 0.5, a closer inspection reveals a decrease in the number of high-quality matches with TMP in the range $[0.95, 1]$.

As such, the apparent improvement in tracking metrics does not necessarily imply a genuine gain in reconstruction performance. Instead, it reflects a change in seed composition and an artifact of how computational resources are redistributed between seed types under degraded input conditions. This suggests that TMP distributions can be a useful diagnostic in simulation studies for understanding how reconstruction performance shifts under non-ideal conditions. Such insights may help guide future algorithmic improvements, especially in seed prioritization and ambiguity handling.

In summary, the fast reconstruction chain indeed improves computational speed and significantly reduces the number of seeds used. However, it also places higher demands on the pixel detector's efficiency and stability. Given that its performance still lags behind the default reconstruction chain even under ideal detector conditions, there remains considerable room for further improvement.

If the long-term goal includes searches for rare processes with very small cross-sections, then greater caution is warranted in the design and optimization of the reconstruction chain itself. Ultimately, the choice depends on whether improving detector yield and stability is more feasible than enhancing reconstruction algorithms or accepting higher computational costs. After all, collider physics has never been solely a physics problem, it is inherently a systems-level challenge that requires careful trade-offs across hardware, software, and operational constraints.

A. Technical Details of the Default Reconstruction Chain

This section describes technical details of the default reconstruction chain, as inferred from the reconstruction logs. The chain involves three key components: `SiSPSeededTrackFinder`, `ITkAmbiguityScore.ITkAmbiguityScoreProcessor`, and `ITkAmbiguitySolver.ITkAmbiguityProcessor`.

The `SiSPSeededTrackFinder` module performs track seeding, Kalman-based track extension, and an initial round of shared cluster conflict mitigation, as shown in Figure A.1. The seeding phase runs in two passes: first using strip-based seeds (SSS), followed by pixel-based seeds (PPP). As a result, reconstruction logs (`log.RAWtoALL`) typically show that PPP seeds are less efficient at producing reconstructed tracks. However, this does not imply intrinsic inferiority of PPP seeds, but merely a consequence of their lower priority in the reconstruction sequence. The default reconstruction chain is compatible with both

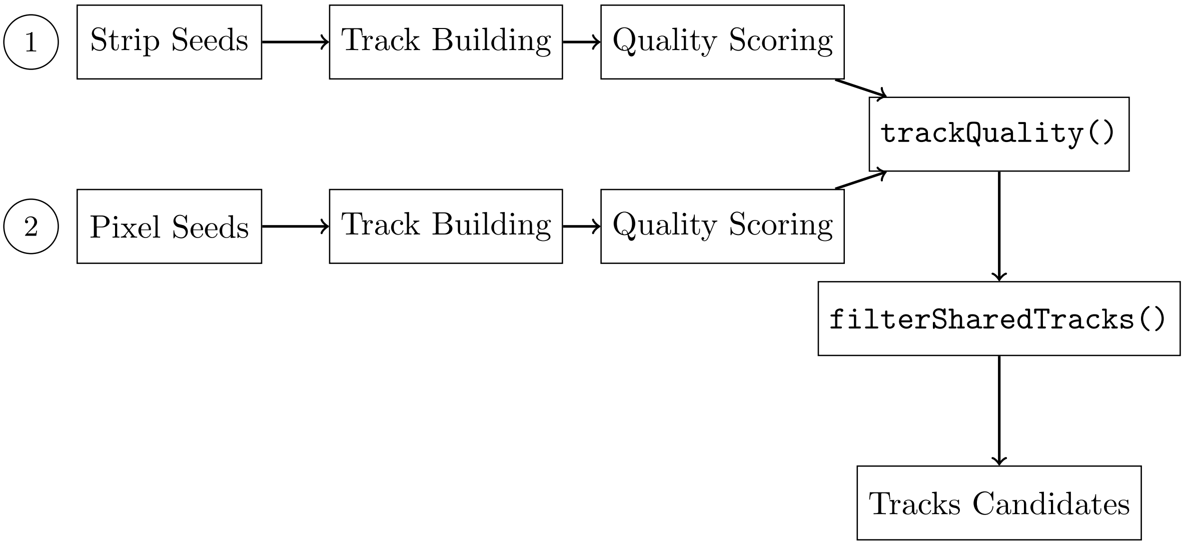


Figure A.1.: Overall workflow of the `SiSPSeededTrackFinder` module. The process involves strip-first then pixel-based seeding, Kalman extension, scoring of track candidates, and a first round of shared-cluster conflict resolution.

ID and ITk geometries, though it handles them differently. In the ID geometry, a z-vertex constraint is estimated using pixel triplets to guide seed formation. In contrast, under ITk geometry, PPP seeds are generated without any z-vertex constraint, using hits from the entire tracker volume. Prioritizing strip-based seeds in the outer layers helps reduce combinatorial complexity by quickly assigning clusters in the sparser detector regions.

During track formation, a list of candidate tracks is maintained and continuously sorted by quality, this list is called `qualitySortedTrackCandidates`. For every triplet seed (regardless of type), the algorithm may generate multiple track candidates, each of which is scored using the `trackQuality()` function and inserted into the sorted list. The scoring procedure loops over all hits in a track, skips invalid measurements, and accumulates a score based on valid hits. If the track is identified as originating from bremsstrahlung, a penalty factor of 0.7 is applied to the total score. The details are illustrated in Figure A.2. Once initial track candidates are built, conflicts may arise due to multiple tracks sharing the same hits (clusters). Since a real hit should belong to only one particle trajectory, the `filterSharedTracks()` step aims to select the most trustworthy candidates from groups of conflicting tracks, while minimizing the reuse of clusters (PRDs). The selection is guided by the quality score and follows a greedy approach, as shown in Figure A.3. Starting from the highest-scoring track, the algorithm loops over its hits and collects those not yet used by previously accepted tracks into a list called `freeClusters`. A track is accepted if it satisfies either of the following criteria:

- The number of `freeClusters` is greater than or equal to a predefined threshold `m_nfreeCut`, indicating the track contains enough independent information.
- All clusters used by the track are free (i.e., no overlap at all).

If either condition is satisfied, the track is accepted, and its `freeClusters` are marked as used. Otherwise, the track is rejected, and the temporarily reserved clusters are released back for future candidates.

Importantly, the inclusion of the `m_nfreeCut` threshold means that not all conflicts can be resolved at this stage. This step takes a greedy approach: it locks in high-scoring tracks early and permits retaining other candidates with sufficiently many free clusters. This allows real tracks to survive despite partial overlap—due to realistic effects such as cluster splitting, sensor overlaps, scattering, or dead regions. However, this also means that residual conflicts are expected and not entirely removed. Final disambiguation is deferred to the next stage: ambiguity solving.

The ambiguity resolution stage is handled by two modules: `ITkAmbiguityScoreProcessor` and `ITkAmbiguityProcessor`.

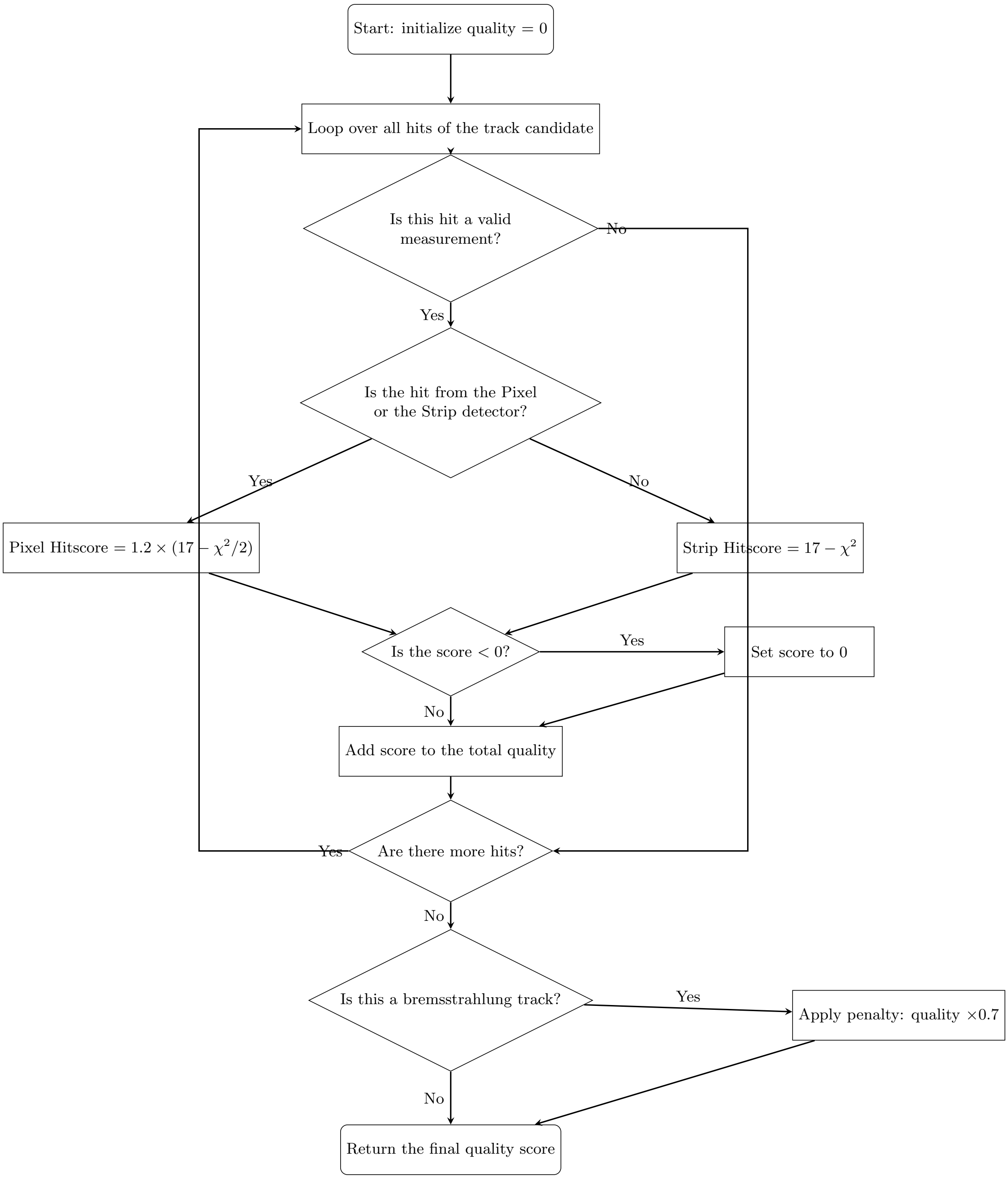


Figure A.2.: Flowchart of the `trackQuality()` scoring function used in `SiSPSeededTrackFinder`. The function evaluates each track by summing hit-based scores, optionally applying a penalty for bremsstrahlung candidates.

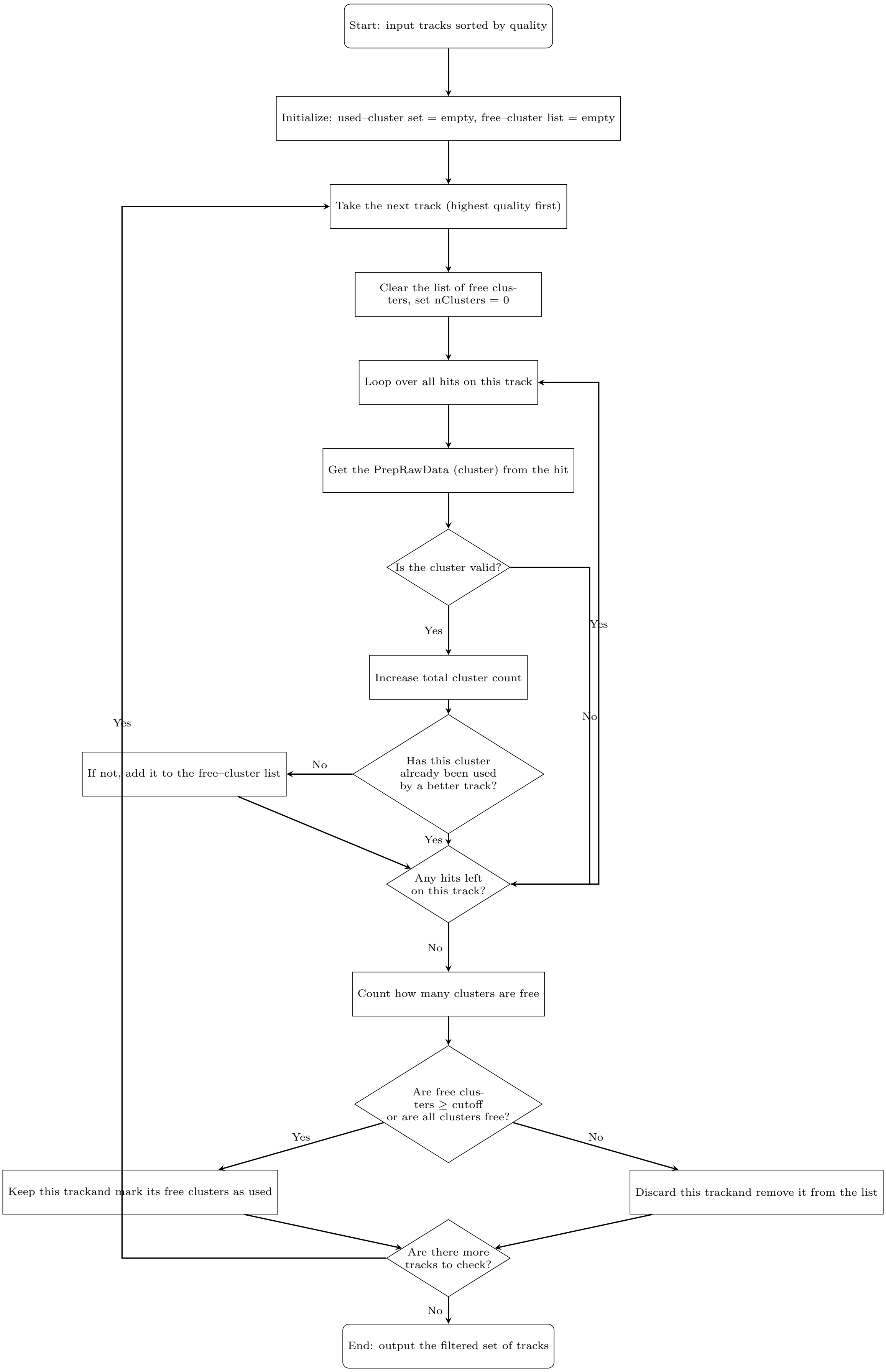


Figure A.3.: Workflow of the `filterSharedTracks()` function in `SiSPSeededTrackFinder`. This procedure selects the best subset of track candidates by iterating over a quality-sorted list and filtering out tracks that excessively share clusters with higher-scoring candidates.

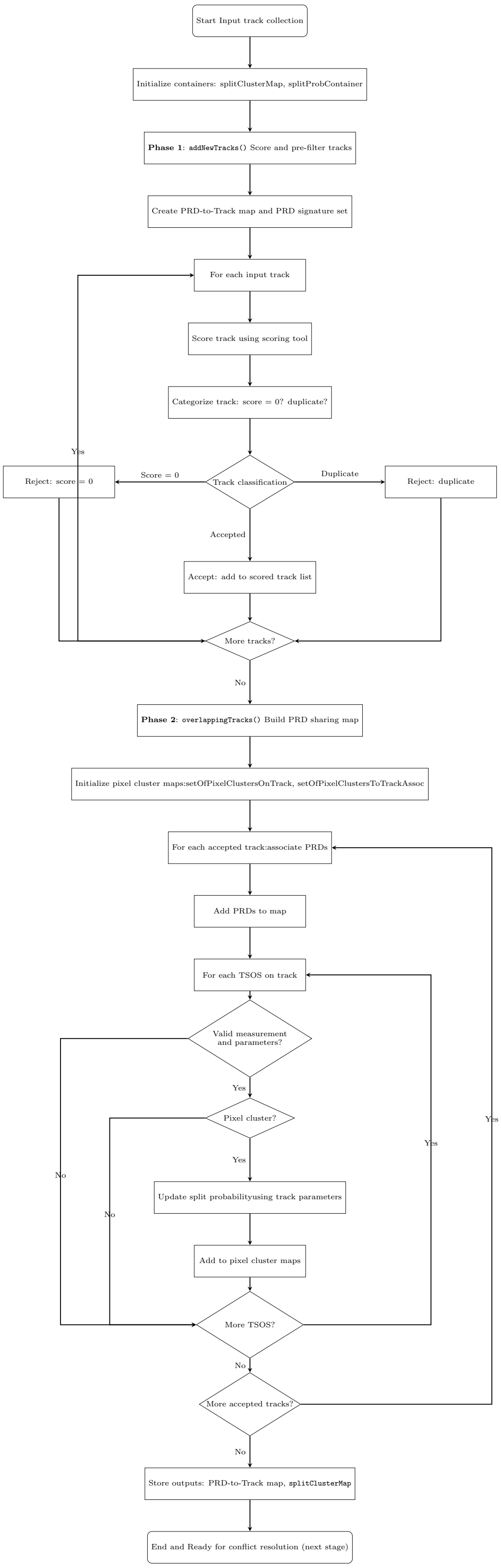


Figure A.4.: Workflow of the `ITkAmbiguityScoreProcessor` module. This stage filters and classifies seeded tracks based on quality scores, and constructs PRD-to-track and pixel-split probability mappings in preparation for further ambiguity resolution.

The `ITkAmbiguityScoreProcessor` has two main responsibilities, the simplified workflow is shown in Figure A.4:

1. **Track Categorization and Scoring**: It uses a scoring tool to assign each track from the seeding stage a score and categorizes them into:
    - `kNcandScoreZero`: tracks with zero score, to be discarded.
    - `kNcandDouble`: duplicated tracks, to be discarded.
    - `kNaccept`: valid tracks, to be kept for further processing.

    Accepted tracks are sorted by descending score and stored in a structure called `trackScoreTrackMap`.

2. **Preparing for Ambiguity Solving**: The `overlappingTracks()` function constructs a mapping from PRDs to tracks (`PRDtoTrackMap`) to identify which PRDs are shared among which tracks. This step performs no decision-making—it merely records the usage relationships. Then, for each cluster and track, the module uses a `splitProbTool` to compute a split probability, indicating whether the cluster is shared or split. The results are stored in a `splitClusterMap` and a `splitProbContainer` for use in the next stage.

The sorted list `trackScoreTrackMap` is passed into the `ITkAmbiguityProcessor`, which performs the final ambiguity resolution. The goal is to iterate through this list, selecting the best track each time, and clean or remove any conflicting candidates, including performing refits when necessary. The full workflow is shown in Figure A.5.

Similar to the earlier greedy filtering, the processor loops through tracks in descending order of score and applies three possible actions depending on the result of hit conflict resolution:

1. **Keep as-is**: If the track has minimal conflict and is already fitted, it is accepted directly.

2. **Refit required**: If the track is acceptable but not yet fitted, it is refitted before being added back to the candidate list.

3. **Subtrack creation**: If the original track has too many conflicts, but a cleaned subtrack can be formed by removing shared hits, this subtrack is rescored and reentered into the candidate pool.

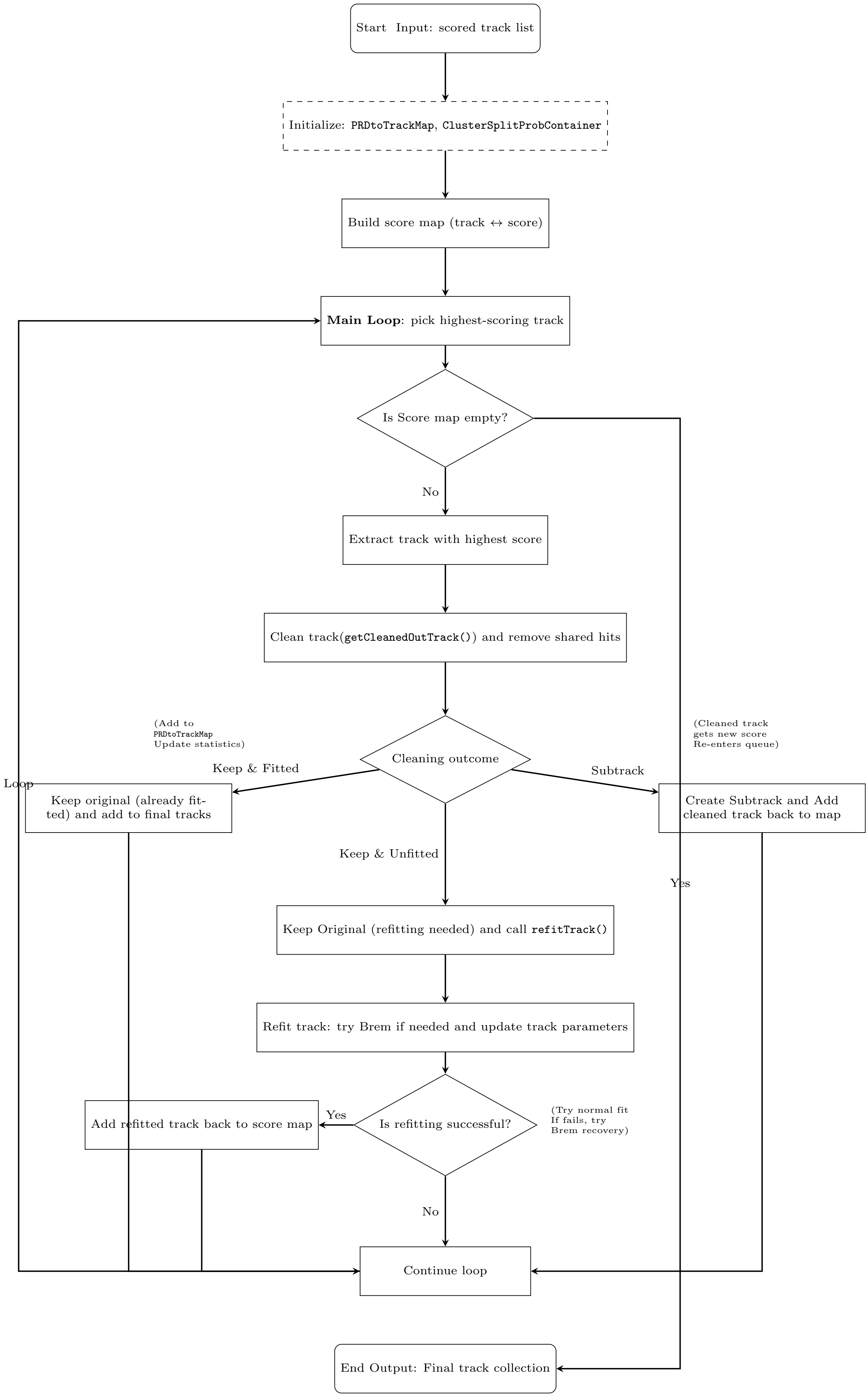


Figure A.5.: Flowchart of the `ITkAmbiguityProcessor`, the final ambiguity resolution module. Tracks are processed in descending score order, and candidates are either accepted, refitted, or cleaned into subtracks, depending on their overlap and split cluster status.

The cluster-splitting probabilities calculated earlier guide decisions in the cleaning process. During refitting, the algorithm dynamically chooses a fitting strategy: if the track is suspected to be from bremsstrahlung, a dedicated electron hypothesis is used; otherwise, it first tries a general particle hypothesis and only resorts to the electron fit if the initial refit fails.

The loop continues until all candidates are processed, and the result is a final, non-overlapping, high-quality track collection, which is what we refer to as the reconstructed tracks in the main body of this thesis.

B. Technical Details of the Fast Track Reconstruction Chain

This section describes the technical details of the fast track reconstruction chain. As mentioned in the main text, unlike the default reconstruction chain, the fast chain does not include a dedicated ambiguity solving stage, specifically, it does not invoke `ITkAmbiguityScore.ITkAmbiguityScoreProcessor` or `ITkAmbiguitySolver.ITkAmbiguityProcessor`.

Overall, the fast reconstruction chain shares a similar structure with the default chain in its seeding stage: it uses `m_seedsmaker` to generate track seeds and `m_trackmaker` to perform combinatorial Kalman tracking. Track candidates are scored, sorted by quality, and then passed to a dedicated function for resolving shared-cluster conflicts. The overall procedure is illustrated in Figure B.1. A key difference is that the fast reconstruction chain performs **only one pass of PPP seeding**, whereas the default chain runs both an initial SSS seeding pass followed by a PPP one. Additionally, the function used to handle shared clusters is different: the fast chain uses `filterSharedTracksFast()` instead of the default `filterSharedTracks()`.

The differences between `filterSharedTracksFast()` and `filterSharedTracks()` lie in three main aspects: the iteration targets, the selection logic, and the cluster locking strategy. These are detailed in Figure B.2. First, the default version iterates over each track's *measurements* (i.e., the list of `Trk::MeasurementBase*` hits), whereas the fast version iterates over full `TrackStateOnSurface*` (TSOS) objects, which contain not only the hit but also associated measurement and fitting information [37]. The cluster-locking strategy also differs: in the default chain, a track only locks its clusters if it passes the selection criteria (i.e., either having enough free clusters or all clusters being free). In contrast, in the fast track reconstruction chain, clusters touched by any higher-scoring track are immediately marked as used, regardless of whether the track itself is accepted. This more aggressive locking approach favors speed over completeness, it avoids costly checks and helps suppress ambiguous solutions, even at the cost of discarding potentially good tracks.

Finally, the selection criteria used in the fast chain are η-dependent, applied via the

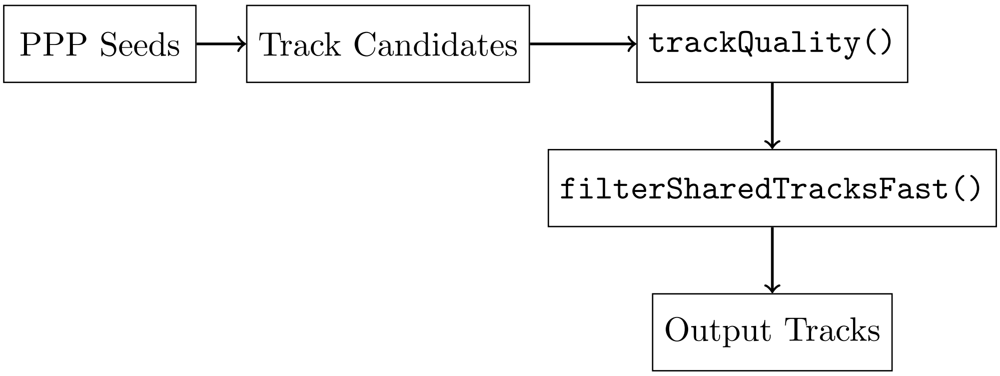


Figure B.1.: The `SiSPSeededTrackFinder` module in the fast track reconstruction chain: PPP seeds only and differences from the default chain in shared cluster handling and quality selection.

`passEtaDepCuts(track, nClusters, nFreeClusters, nPixels)` function. The logic of this function is shown in Figure B.3. It applies a set of quality requirements based on the candidate's pseudorapidity (η) to ensure that only well-measured, physically plausible tracks are accepted. The function first retrieves the track parameters from the first TSOS. If they are not available, the candidate is immediately rejected. Then, it calculates $|\eta|$ and applies the following cuts, each with thresholds that vary depending on η:

- Minimum number of total silicon hits;
- Minimum number of unshared (i.e., not reused) clusters;
- Maximum number of shared clusters;
- Minimum number of pixel hits;
- Minimum transverse momentum (p_T).

If the candidate has a valid perigee state (i.e., a track state at the point of closest approach to the beamline), it must also satisfy a maximum constraint on the transverse impact parameter $|d_0|$.

The function returns `true` only if all applicable conditions are met. Otherwise, the track is discarded. This mechanism serves as an efficient filter to retain only the most trustworthy candidates, with criteria tailored to the detector geometry and expected track quality as a function of η.

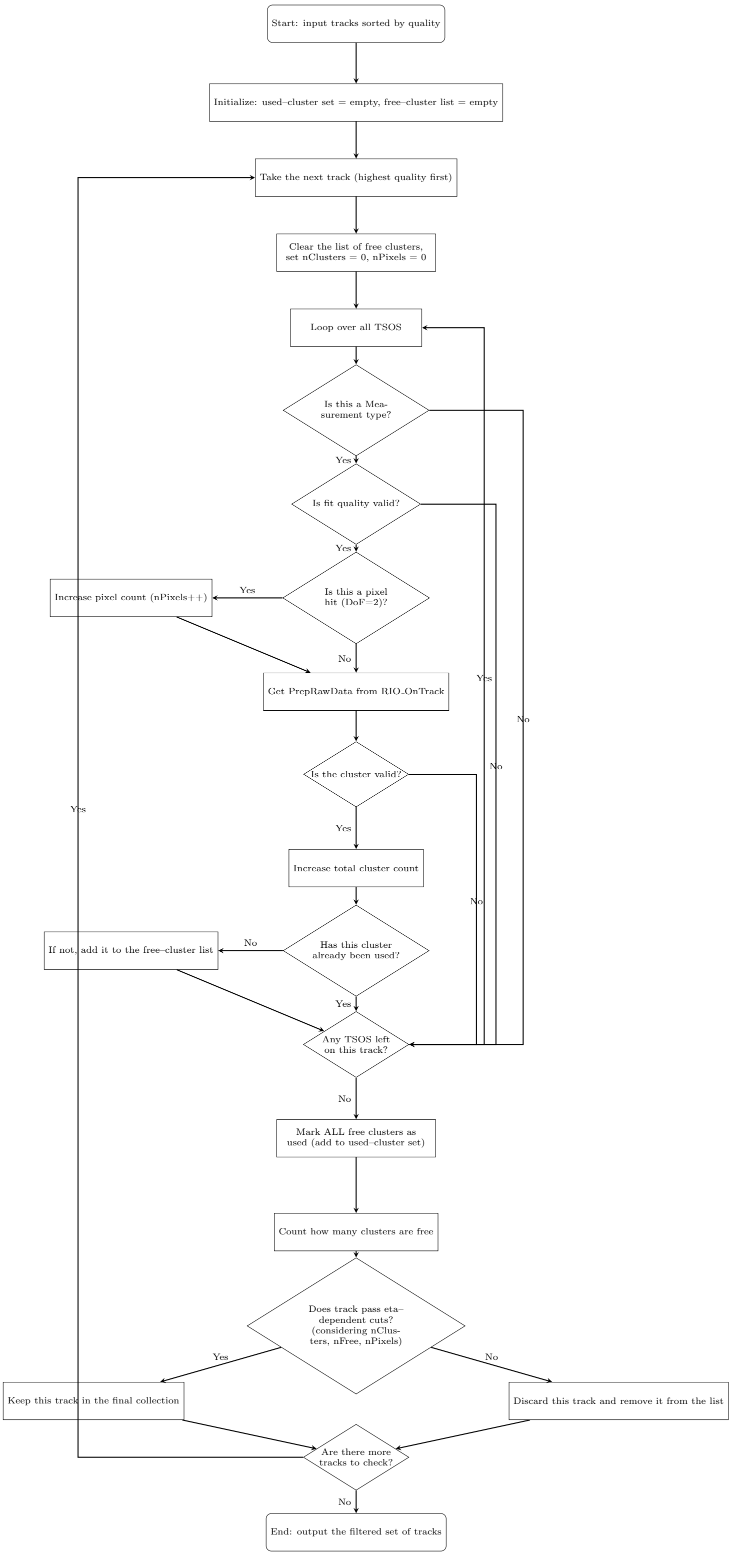


Figure B.2.: Workflow of the `sharedFilterTracksFast` in the fast track reconstruction chain: differences from the default chain in TSOS-based traversal and cluster reuse policy.

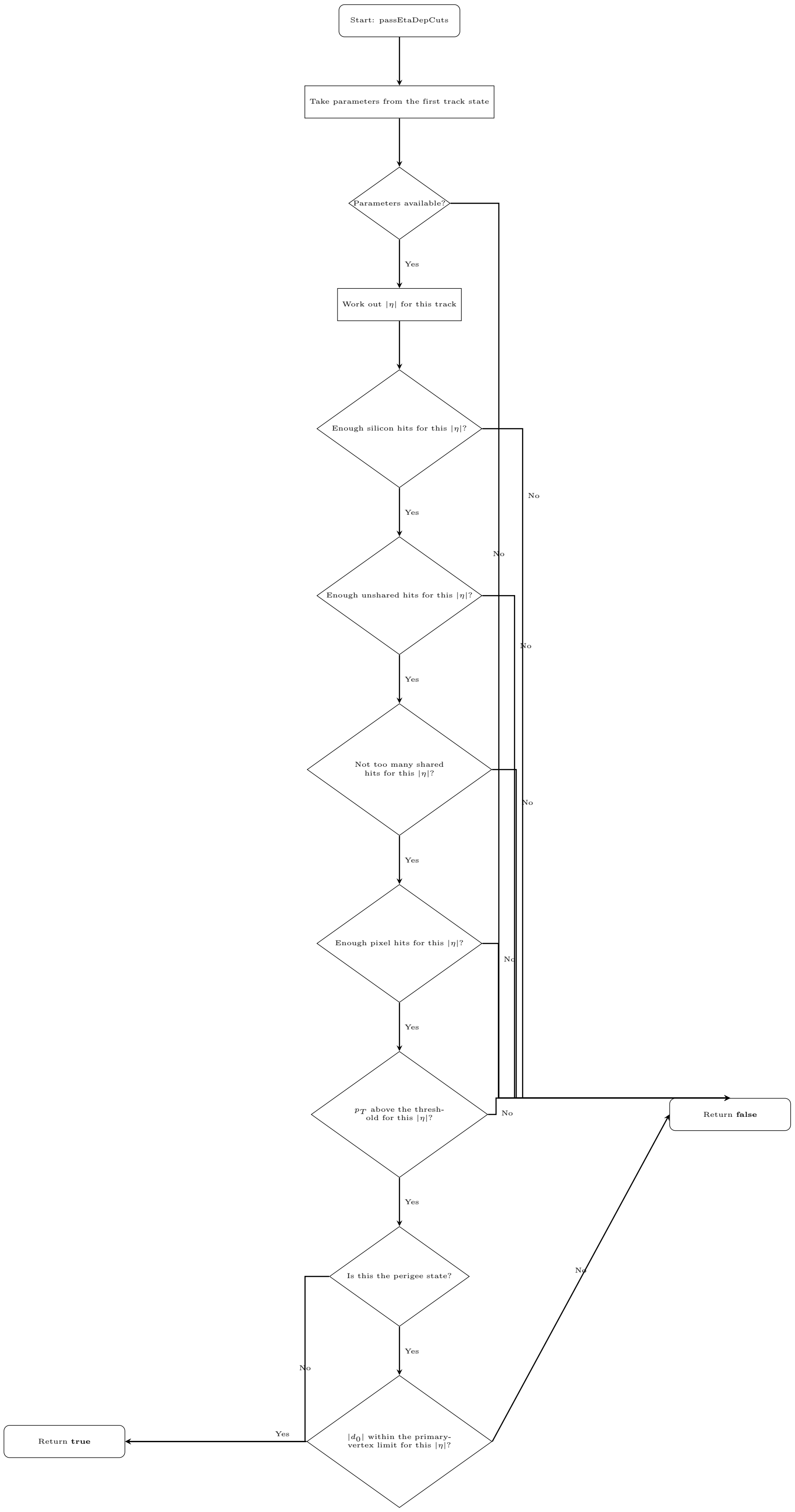


Figure B.3.: Workflow of the `passEtaDepCuts()` function: a sequence of decision steps applying η-dependent quality criteria to track candidates.

C. ITk Pixel Masking Simulation Configuration in `InDetDefectsEmulation`

The masking simulation used in this thesis is implemented at the RDO level by discarding pixel hits from the RDO container. The original RDO container is first remapped (e.g. `PixelRDOs` → `PixelRDOs_ORIG`), and the `PixelDefectsEmulatorAlg` algorithm then removes the corresponding pixel hits according to the simulated defect information.

1. **RDO Remapping (renaming the original container)**
    - Functions: `PixelRDORemappingCfg` / `ITkPixelRDORemappingCfg`.
    - Purpose: Rename `PixelRDOs` (or `ITkPixelRDOs`) to `PixelRDOs_ORIG` (or `ITkPixelRDOs_ORIG`) so that the subsequent simulation algorithm can read the original data from `*_ORIG` and write the modified data back to the original key, effectively replacing the original collection.
2. **Defect Conditions Generation (conditions algorithm)**
    - Functions: `PixelDefectsEmulatorCondAlgCfg` / `ITkPixelDefectsEmulatorCondAlgCfg`.
    - Output (written to the conditions store): default `WriteKey` is `PixelEmulatedDefects` / `ITkPixelEmulatedDefects`.
    - This step does not modify the RDO directly. Instead, it generates a "defect map" listing the defective modules, pixels, columns, or circuits, according to the configured patterns and probabilities. The event algorithm uses this map to remove hits.
3. **Event Algorithm Application (hit removal based on the defect map)**
    - Functions: `PixelDefectsEmulatorAlgCfg` / `ITkPixelDefectsEmulatorAlgCfg`.
    - Input: `InputKey` = `*_ORIG`, `EmulatedDefectsKey` = `PixelEmulatedDefects` (or ITk version).

- Output: `OutputKey` = `PixelRDOs` (or `ITkPixelRDOs`).
- Action: remove (or mark invalid, depending on the C++ implementation) the RDO entries corresponding to the defective modules, pixels, columns, or circuits defined in the defect map. In practice, the configuration name indicates that hits are discarded.

4. **(Optional) Converting defects to `DetectorElementStatus`**
   - Function: `ITkPixelDefectsEmulatorToDetectorElementStatusCondAlgCfg`.
   - Converts `ITkPixelEmulatedDefects` into `ITkPixelDetectorElementStatusFromEmulatedDefects` for downstream components that depend on `DetectorElementStatus` (e.g. logic that uses bad-module status).

Configurable Parameters ("Injection Controls")

All these are set in `*DefectsEmulatorCondAlgCfg` via `kwargs` (with slightly different defaults for ITk and non-ITk):

1. **ModulePatterns** (default: `[[-2,2,0,99,-99,99,-99,99,0,9999,0,0,0]]`)

   Each row defines a selection region. The fields (in order) are:
   - Barrel/endcap: $-2..2$ (all), where -2 = endcap A, 0 = barrel, 2 = endcap C.
   - Layer/disk: 0..99 (all layers).
   - η: $-99..99$ (all η rings/segments).
   - ϕ: $-99..99$ (all ϕ segments).
   - Column count: 0..9999 (all column counts).
   - Side: 0..0 (Pixel has only one side).
   - Final flag: for strips only (connected-row auto-match); ignored for Pixel.

   This pattern defines *where* defects are injected. To mask only L0 barrel, or only a specific η/ϕ sector, narrow the corresponding ranges. Multiple patterns can be given to apply different strategies to different regions.

2. **DefectProbabilities** (default ITk: `[[0., 1e-2, 1e-1, 0.]]`)

   Four elements per pattern: `[module, pixel, core-column, circuit]`.
   - **module**: probability that an entire module is defective.

- **pixel**: probability that an individual pixel is defective.
- **core-column**: probability that an entire column is defective.
- **circuit**: probability that a readout circuit (ROC/FE/sub-circuit) is defective.

  Each `ModulePattern` can have its own `DefectProbabilities`.

3. **NDefectFractionsPerPattern** (default: `[[1., -1, 1.]]`)

   Controls the multiplicity distribution for multi-entity defects (columns, circuits), i.e. how many defective columns/circuits per defective module. The special value `-1` means "use internal logic" (e.g. Poisson) instead of manually specifying fractions. If `emulateITkPixelDefectsPoisson(flags, acc)` is called, it overrides this with a Poisson distribution.

4. **HistogramGroupName / DefectsHistSvcCfg**

   By default, histogramming is disabled (`""`). To enable, set a prefix (e.g. `"/PixelDefects/EmulatedDefects/"`) and configure an output ROOT file via `DefectsHistSvcCfg` or `ITkDefectsHistSvcCfg`.

The pattern `[[-2,2,0,99,-99,99,-99,99,0,9999,0,1,0]]` is a "sentinel" pattern matching *all* Pixel detector modules. By changing the ranges, one can restrict the masking region (for the detailed range definitions, see Tables C.1 and C.2). For example:

- $[-2, 2]$: `bec` (barrel/endcap). $[-2, 2]$ = all, $[-2, 0]$ = endcap A + barrel, $[0, 2]$ = barrel + endcap C. Selecting only endcap A and C requires two patterns combined via `combineModuleDefects`.
- $[0, 99]$: Pixel layer index. Barrel: $[0, 4]$ valid, Endcap: $[0, 8]$ valid (see Fig. 2.5).
- ϕ range $[-99, 99]$: all ϕ. For barrel, this corresponds to the stave count (e.g. L0 has 12 staves $\Rightarrow [0, 11]$). For endcap, this corresponds to "sensors per ring".
- η range $[-99, 99]$: For barrel, corresponds to "flat sensors per row" along the stave; for endcap, the ring index within a layer (0 for the innermost ring, increasing outward).
- `columns_or_strips`: distinguishes module types: quad ($[800, 800]$), barrel triplet ($[200, 200]$), endcap ring triplet ($[384, 384]$). The correct type must be used to mask intended modules.
- Side: $[0, 0]$ for Pixel.

Layer	Barrel/Endcap	Phi Module Range	Eta Module Values	Index Range (ϕ, η)
0	barrel	0–11	$-12 \sim -1$, $1 \sim 12$	0–767, 0–200
1	barrel	0–19	$-6 \sim -1$, $1 \sim 6$	0–772, 0–804
2	barrel	0–31	$-9 \sim -1$, $1 \sim 9$	0–772, 0–804
3	barrel	0–43	$-9 \sim -1$, $1 \sim 9$	0–772, 0–804
4	barrel	0–55	$-9 \sim -1$, $1 \sim 9$	0–772, 0–804

Table C.1.: Range and index of the central barrel layers of the ITk Pixel (from `InDetIdDict.xml`).

Disk	Barrel/Endcap	Eta Modules	Phi Modules	Index Range (ϕ, η)
0	endcap	0–14	0–17	0–400, 0–383
1	endcap	0–5	0–29	0–772, 0–804
2	endcap	0–22	0–19	0–772, 0–804
3	endcap	0–5	0–31	0–772, 0–804
4	endcap	0–10	0–31	0–772, 0–804
5	endcap	0–7	0–43	0–772, 0–804
6	endcap	0–7	0–43	0–772, 0–804
7	endcap	0–8	0–55	0–772, 0–804
8	endcap	0–8	0–51	0–772, 0–804

Table C.2.: Range and index of the endcap rings in the ITk Pixel (from `InDetIdDict.xml`).

Example: Masking the entire L7 layer of the Pixel endcap

```
# Import masking configuration helpers
from InDetDefectsEmulation.PixelDefectsEmulatorPostInclude import
    emulateITkPixelDefects
from InDetDefectsEmulation.StripDefectsEmulatorConfig import moduleDefect,
    combineModuleDefects

# Define masking pattern for Endcap A, Layer 7 (entire layer)
PixelEndcapA = moduleDefect(
    bec=[-2,-2], # Barrel/Endcap code: -2 = Endcap A
    layer=[7,7], # Target layer: L7 only
```

```
    phi_range=[-99,99], # All phi modules
    eta_range=[0,99], # All eta modules
    columns_or_strips=[800,800], # Quad modules
    side_range=[0,0], # Pixel side index
    all_rows=False, # Disable connected-rows logic (for strips)
    probability=[1.0, 0.0, 0.0, 0.0], # 100\% module defect probability
    fractionsOfNDefects=[[1.], [1.]], # Exactly one defective entity
    noiseProbability=None, noiseShape=[], # No noise injection
    cornerDefectParam=[], cornerDefectNCornerFractions=[] # No corner defects
)

# Define masking pattern for Endcap C, Layer 7 (entire layer)
PixelEndcapC = moduleDefect(
    bec=[2,2], # Barrel/Endcap code: 2 = Endcap C
    layer=[7,7], # Target layer: L7 only
    phi_range=[-99,99], # All phi modules
    eta_range=[0,99], # All eta modules
    columns_or_strips=[800,800], # Quad modules
    side_range=[0,0], # Pixel side index
    all_rows=False, # Disable connected-rows logic (for strips)
    probability=[1.0, 0.0, 0.0, 0.0], # 100\% module defect probability
    fractionsOfNDefects=[[1.], [1.]], # Exactly one defective entity
    noiseProbability=None, noiseShape=[], # No noise injection
    cornerDefectParam=[], cornerDefectNCornerFractions=[] # No corner defects
)

# Merge both Endcap A and Endcap C patterns into a single configuration
pat, prob, nfrac, nnoise, nshape, cpar, cfrac = combineModuleDefects([
    PixelEndcapA, PixelEndcapC])

# Apply the masking configuration
emulateITkPixelDefects(flags, cfg,
    ModulePatterns=pat, # Combined module patterns
    DefectProbabilities=prob, # Defect probabilities
    NDefectFractionsPerPattern=nfrac, # Defect multiplicity fractions
    NoiseProbability=nnoise, NoiseShape=nshape, # Noise settings
    CornerDefectParamsPerPattern=cpar, # Corner defect settings
    NCornerDefectFractionsPerPattern=cfrac, # Corner defect multiplicity
    HistogramFileName='killPixelEndcapL7.root', # Output histogram file
```

```
    PropagateDefectsToStatus=True # Update DetectorElementStatus)
```

D. ITk Strip Masking Simulation Configuration in `InDetDefectsEmulation`

The masking simulation for the ITk strip detector follows the same principle as for the pixel detector: first, a *conditions algorithm* generates a persistent “defect map” (conditions data, event-independent), and then an *event algorithm* removes the RDO hits overlapping with the defective elements on an event-by-event basis.

For strips, two levels of defects are supported:

- **ModuleDefectProb**: module-level defects, in which the entire module is marked as defective.
- **StripDefectProb**: strip-level defects, in which individual strips within a module are marked as defective.

In the Athena framework, strips are still referred to as `SCT` in some naming conventions. In the endcap part, the log output uses the term *wheel* for each endcap disk, and *ring* for the concentric circular layers within a wheel. The η-module index corresponds to the radial ordering, analogous to the pixel detector.

The mapping of the strip geometry (from the detector numerology) is as follows:

```
Barrel (index 0):
  Number of barrel layers = 4
  layer = 0: 112 etaModules × 28 phiModules
  layer = 1: 112 etaModules × 40 phiModules
  layer = 2: 56 etaModules × 56 phiModules
  layer = 3: 56 etaModules × 72 phiModules
```

For endcaps:

- Endcap A corresponds to `bec=[-2,-2]`.
- Endcap C corresponds to `bec=[2,2]`.

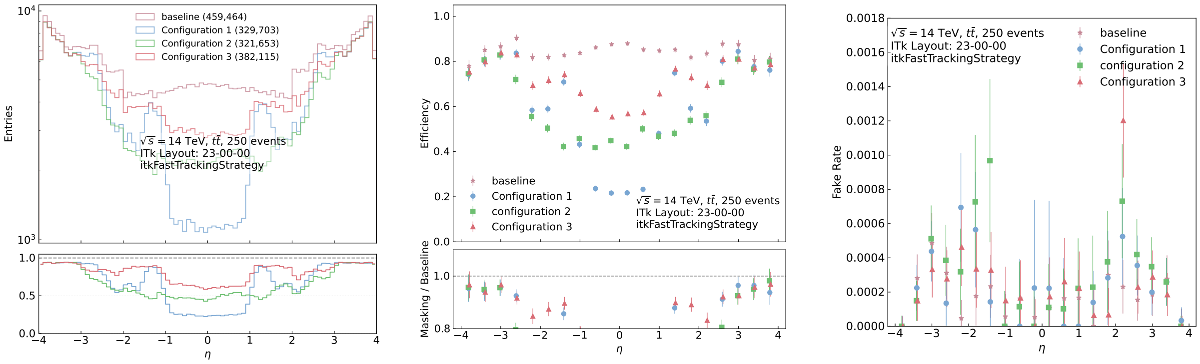
For the barrel, the `bec` setting is `[-1,1]`; however, it has been observed that `[0,0]` in practice also selects the strip barrel.

Example: Masking both sides of endcap disk L5

```
from InDetDefectsEmulation.StripDefectsEmulatorConfig import (
    ITkStripDefectsEmulatorCondAlgCfg,
    ITkStripDefectsEmulatorAlgCfg,
    ITkStripDefectsEmulatorToDetectorElementStatusCondAlgCfg,
    DefectsHistSvcCfg,
    moduleDefect,
    combineModuleDefects
)

# Endcap A, disk L5
endcap_a_shutdown = moduleDefect(
    bec=[-2, -2], # Endcap A
    layer=[5, 5], # Disk index 5
    phi_range=[-99, 99], # All phi modules
    eta_range=[-99, 99], # All eta modules
    side_range=[0, 0, 1, 1], # Both sides
    all_rows=True, # Apply to all rows of physical sensor
    probability=[1.0, 0.0] # 100% module defects, no strip defects
)

# Endcap C, disk L5
endcap_c_shutdown = moduleDefect(
    bec=[2, 2], # Endcap C
    layer=[5, 5],
    phi_range=[-99, 99],
    eta_range=[-99, 99],
    side_range=[0, 0, 1, 1],
    all_rows=True,
    probability=[1.0, 0.0]
)

# Combine patterns
pat, prob, frac, nnoise, nshape, cpar, cfrac = combineModuleDefects([
    endcap_a_shutdown,
    endcap_c_shutdown
```

```
])

# Enable histogramming
cfg.merge(DefectsHistSvcCfg(flags,
    HistogramGroup="ITkStripDefects",
    FileName="killStripEndcapDisk5.root"
))

# Schedule condition algorithm
cfg.merge(ITkStripDefectsEmulatorCondAlgCfg(flags,
    ModulePatterns=pat,
    DefectProbabilities=prob,
    CornerDefectParamsPerPattern=cpar,
    NCornerDefectFractionsPerPattern=cfrac,
    HistogramGroupName="/ITkStripDefects/StripEmulatedDefects/",
    WriteKey="ITkStripEmulatedDefects"
))

# Schedule event algorithm
cfg.merge(ITkStripDefectsEmulatorAlgCfg(flags,
    EmulatedDefectsKey="ITkStripEmulatedDefects",
    HistogramGroupName="/ITkStripDefects/StripRejectedRDOs/"
))

# Propagate defects to DetectorElementStatus
cfg.merge(ITkStripDefectsEmulatorToDetectorElementStatusCondAlgCfg(flags,
    EmulatedDefectsKey="ITkStripEmulatedDefects",
    WriteKey="ITkStripDetectorElementStatusFromEmulatedDefects"
))
strip_det_el_status_cond_alg = cfg.getCondAlgo(
    "ITkStripDetectorElementStatusCondAlgNoByteStreamErrors"
)
strip_det_el_status_cond_alg.ConditionsSummaryTool.\
    SCTDetElStatusCondDataBaseKey = "
        ITkStripDetectorElementStatusFromEmulatedDefects"
```

E. Results for Failure Modes under the Fast Reconstruction Chain

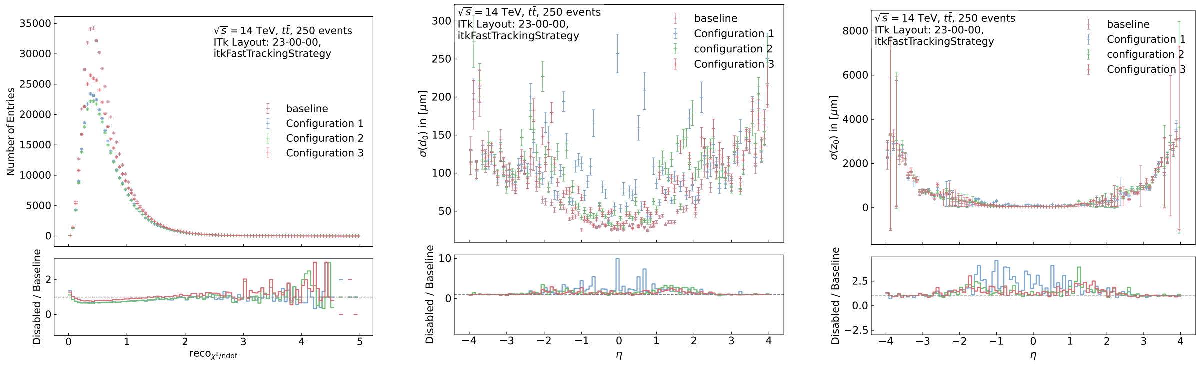


(a) Number of reconstructed tracks vs. η

(b) Tracking efficiency vs. η

(c) Fake rate vs. η

Figure E.1.: Tracking performance of the three selected configurations under the `fast reconstruction chain`, compared to the baseline. While all configurations result in substantial degradation, Configuration 3 demonstrates relatively better performance.



(a) χ^2/N_{dof} ***(b)*** Width of resolution for d_0 ***(c)*** Width of resolution for z_0

Figure E.2.: Track quality metrics for the three selected defect configurations under the fast reconstruction chain. A noticeable leftward shift in the peak of the χ^2/N_{dof} distribution is observed in the masked configurations. Among them, Configuration 3 shows the least distortion.

F. Detailed Comparison of Baseline Seeding Usage in two Reconstruction Chains from Log Outputs

Table F.1.: newStrategy Results are from 250 $t\bar{t}$ events at $\sqrt{s} = 14$ TeV using ITk Layout 23-00-00.

Metric	**PPP**	**PPS**	**PSS**	**SSS**	**ALL**
Seed Processing Statistics					
Input seeds	6,648,384	0	0	1,384,594	8,032,978
No track parameters	53,404	0	0	317,498	370,902
Used seeds	3,788,349	0	0	574,490	4,362,839
Used seeds brem	0	0	0	0	0
Det elements in road	75.1	0	0	71.8	74.7
Two clusters on DE	0	0	0	0	0
Wrong DE road	3	0	0	738	741
Wrong initialization	0	0	0	2	2
Can not find track	3,012,676	0	0	238,536	3,251,212
It is not new track	192,194	0	0	40,081	232,275
Attempts brem model	0	0	0	0	0
Output tracks	618,075	0	0	309,884	927,959
Output extra tracks	55,525	0	0	24,147	79,672
Output tracks brem	0	0	0	0	0
Seeds with Track Summary					
Seeds with track	562,550	0	0	285,737	848,287
Tracking Efficiency by η Region					
Track/Used ratio					**Seeds**
$0.0 < \|\eta\| \leq 0.5$	0.088	0	0	0.672	73,485
$0.5 < \|\eta\| \leq 1.0$	0.069	0	0	0.700	75,837
$1.0 < \|\eta\| \leq 1.5$	0.053	0	0	0.686	89,355
$1.5 < \|\eta\| \leq 2.0$	0.066	0	0	0.321	136,060
$2.0 < \|\eta\| \leq 2.5$	0.149	0	0	0.473	131,743
$2.5 < \|\eta\| \leq 3.0$	0.206	0	0	0.491	114,461
$3.0 < \|\eta\| \leq 3.5$	0.309	0	0	0	110,003
$3.5 < \|\eta\| \leq 4.0$	0.654	0	0	0	117,343

Table F.2.: itkFastTrackingStrategy Results are from 250 $t\bar{t}$ events at $\sqrt{s} = 14$ TeV using ITk Layout 23-00-00.

Metric	**PPP**	**PPS**	**PSS**	**SSS**	**ALL**
Seed Processing Statistics					
Input seeds	3,224,644	0	0	0	3,224,644
No track parameters	18,747	0	0	0	18,747
Used seeds	1,485,177	0	0	0	1,485,177
Used seeds brem	0	0	0	0	0
Det elements in road	73.0	0	0	0	73.0
Two clusters on DE	0	0	0	0	0
Wrong DE road	0	0	0	0	0
Wrong initialization	0	0	0	0	0
Can not find track	829,258	0	0	0	829,258
It is not new track	81,525	0	0	0	81,525
Attempts brem model	0	0	0	0	0
Output tracks	559,609	0	0	0	559,609
Output extra tracks	0	0	0	0	0
Output tracks brem	0	0	0	0	0
Seeds with Track Summary					
Seeds with track	559,609	0	0	0	559,609
Tracking Efficiency by η Region					
Track/Used ratio					**Seeds**
$0.0 < \|\eta\| \leq 0.5$	0.307	0	0	0	56,882
$0.5 < \|\eta\| \leq 1.0$	0.234	0	0	0	57,247
$1.0 < \|\eta\| \leq 1.5$	0.256	0	0	0	57,030
$1.5 < \|\eta\| \leq 2.0$	0.521	0	0	0	59,951
$2.0 < \|\eta\| \leq 2.5$	0.451	0	0	0	66,053
$2.5 < \|\eta\| \leq 3.0$	0.327	0	0	0	84,173
$3.0 < \|\eta\| \leq 3.5$	0.420	0	0	0	87,558
$3.5 < \|\eta\| \leq 4.0$	0.859	0	0	0	90,715

G. Variation of the Track-to-Used Ratio with Pixel Masking Fraction in Two Reconstruction Chains

Table G.1.: Track-to-used seed ratio in different $|\eta|$ bins under the `newStrategy`, with varying fractions of randomly disabled pixel modules. Results are shown separately for PPP and SSS seeds.

PPP seeds

$\|\eta\|$ bin	Baseline	2% disabled	5% disabled	8% disabled	12% disabled	20% disabled
0.0–0.5	0.08758	0.08695	0.08580	0.08516	0.08399	0.08230
0.5–1.0	0.06870	0.06838	0.06795	0.06832	0.06805	0.06921
1.0–1.5	0.05348	0.05328	0.05333	0.05319	0.05334	0.05395
1.5–2.0	0.06634	0.06595	0.06538	0.06508	0.06472	0.06386
2.0–2.5	0.14910	0.14970	0.15090	0.15230	0.15430	0.15860
2.5–3.0	0.20560	0.20680	0.20880	0.21110	0.21290	0.21750
3.0–3.5	0.30920	0.31160	0.31390	0.31750	0.31970	0.32340
3.5–4.0	0.65370	0.65740	0.66220	0.66440	0.66230	0.63740

SSS seeds

$\|\eta\|$ bin	Baseline	2% disabled	5% disabled	8% disabled	12% disabled	20% disabled
0.0–0.5	0.6722	0.6720	0.6726	0.6723	0.6722	0.6718
0.5–1.0	0.6995	0.6998	0.6994	0.6994	0.6993	0.6997
1.0–1.5	0.6858	0.6851	0.6855	0.6847	0.6847	0.6850
1.5–2.0	0.3214	0.3214	0.3213	0.3213	0.3212	0.3210
2.0–2.5	0.4731	0.4733	0.4727	0.4721	0.4710	0.4702
2.5–3.0	0.4912	0.4912	0.4737	0.4737	0.4737	0.4386
3.0–3.5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3.5–4.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table G.2.: Track-to-used seed ratio in different $|\eta|$ bins for the Fast Track reconstruction chain, under varying pixel module masking fractions.

$\lvert\eta\rvert$ **Bin**	**Baseline**	**2%**	**5%**	**8%**	**12%**	**16%**	**20%**
0.0–0.5	0.3073	0.3199	0.3294	0.3445	0.3642	0.3856	0.4048
0.5–1.0	0.2341	0.2447	0.2546	0.2666	0.2871	0.3001	0.3212
1.0–1.5	0.2563	0.2669	0.2826	0.2868	0.3059	0.3241	0.3460
1.5–2.0	0.5211	0.5343	0.5451	0.5694	0.5865	0.6120	0.6369
2.0–2.5	0.4513	0.4660	0.4813	0.4995	0.5158	0.5482	0.5745
2.5–3.0	0.3269	0.3357	0.3545	0.3634	0.3927	0.4153	0.4454
3.0–3.5	0.4201	0.4291	0.4423	0.4537	0.4883	0.4970	0.5351
3.5–4.0	0.8587	0.8645	0.8735	0.8772	0.8891	0.8894	0.8969

H. Cross-Check with a 500 events $t\bar{t}$ Sample at 14 TeV

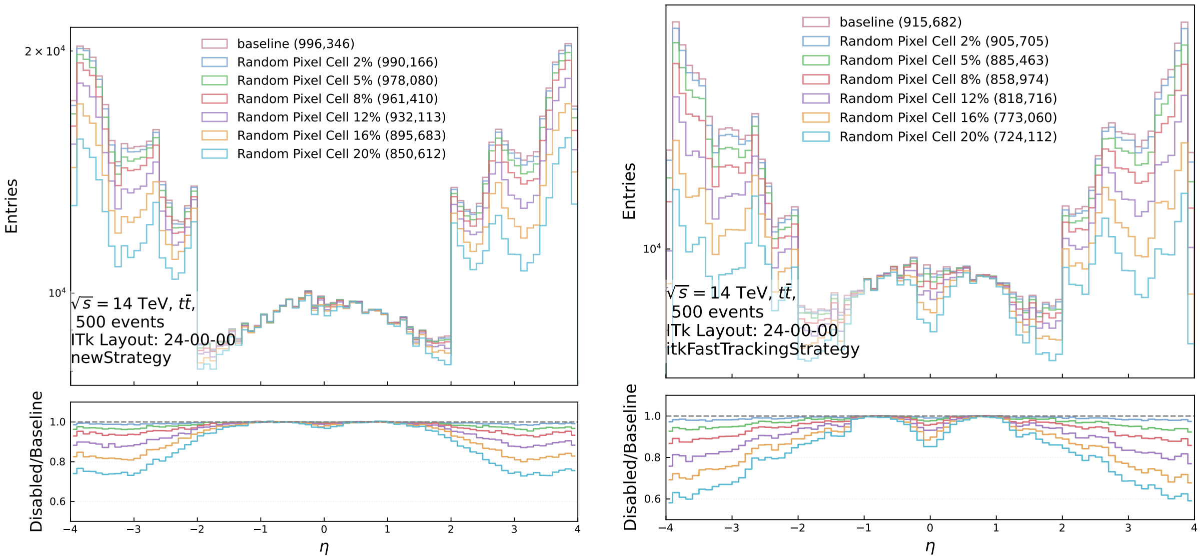
Two 14 TeV $t\bar{t}$ Monte Carlo samples were used in this study:

1. The primary sample used in the main analysis, consisting of 250 events, as described in the main body of the text:

   `mc21_14TeV.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.recon.RDO.e8481_s4149_r15238/*`

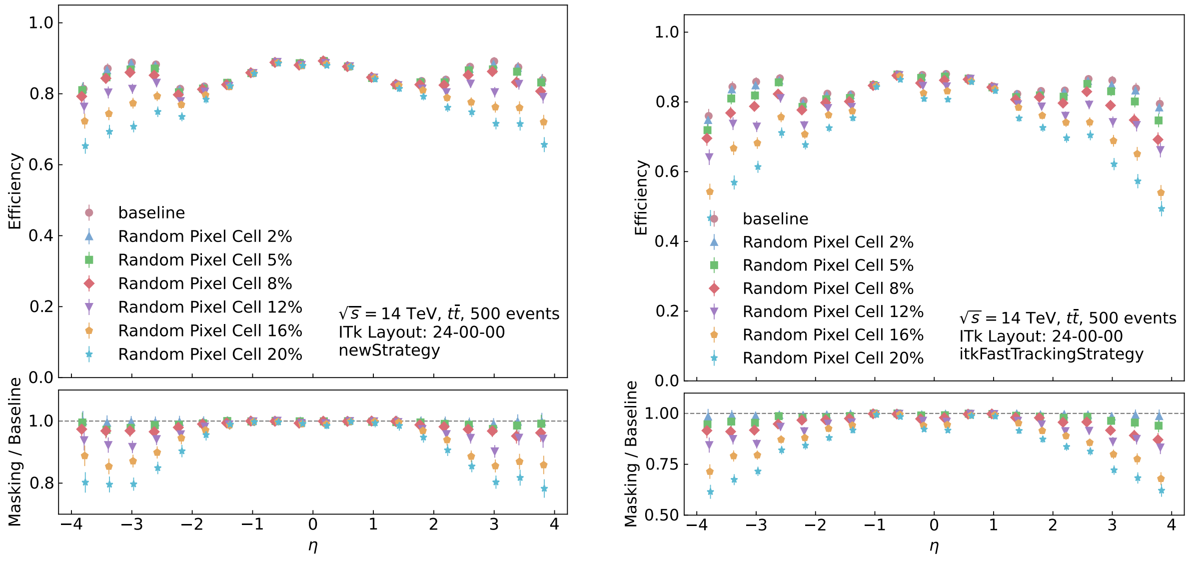
2. An additional sample used in this section for cross-check purposes:
   `mc21_14TeV.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.recon.RDO.e8481_s4494_r16535/*`

Due to limited time for analysis during the writing of this thesis, only 500 events were used from the second sample. It should be noted that this sample contains partially reduced information, most notably, the `truthOrigin` branch indicates that the majority of tracks are unlinked. As a result, the evaluation of the fake rate is not included in this section.



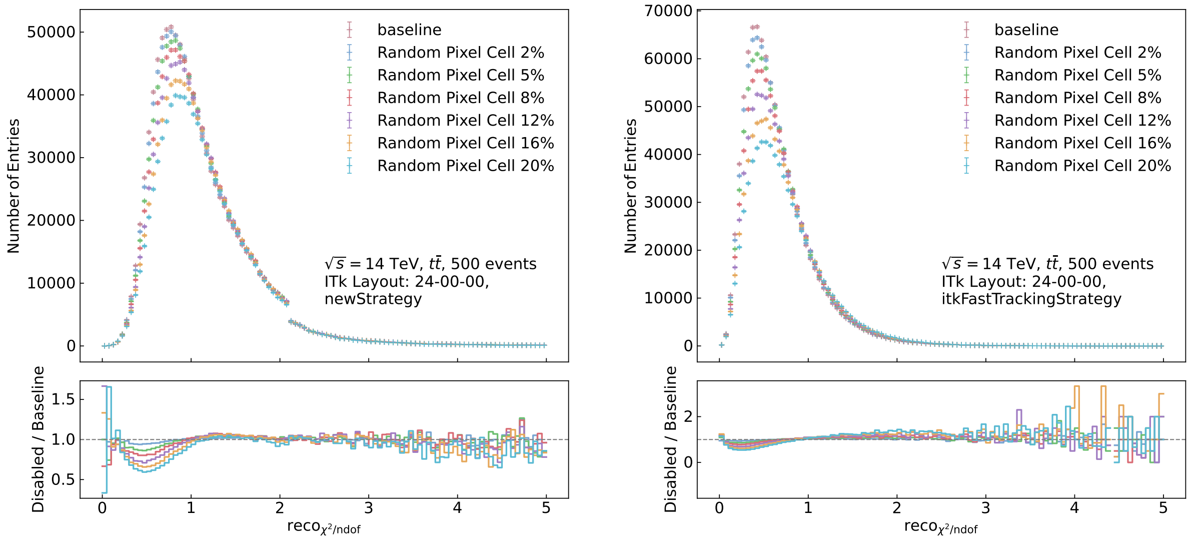
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.1.: Distributions of reconstructed track η under increasing levels of random pixel cell masking (0% to 20%) for both reconstruction chains. Results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



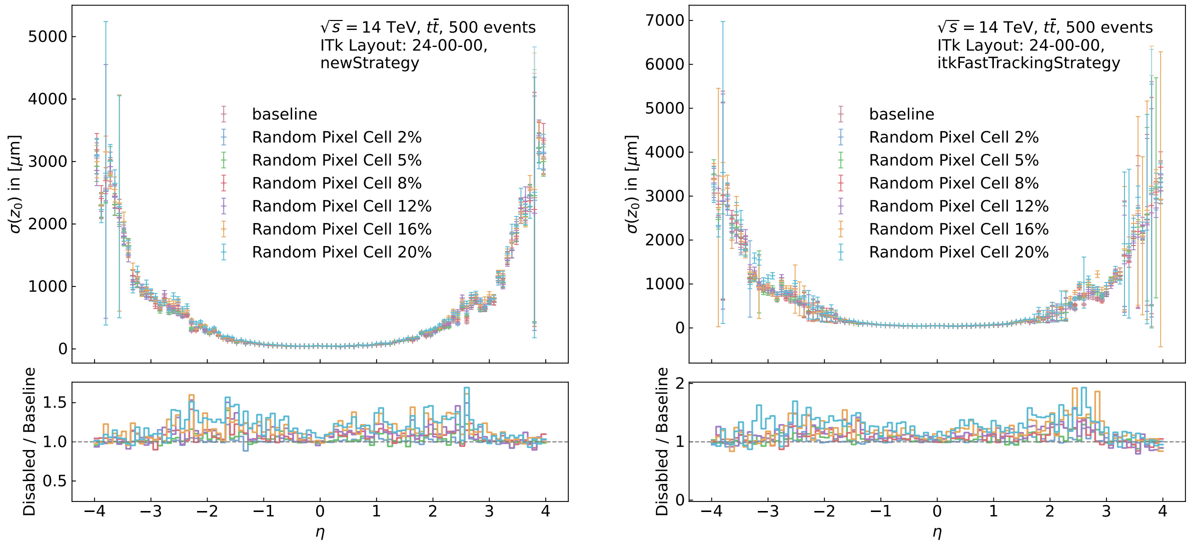
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.2.: Tracking efficiency in η bins under increasing levels of random pixel cell masking (0% to 20%) for both reconstruction chains. Results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



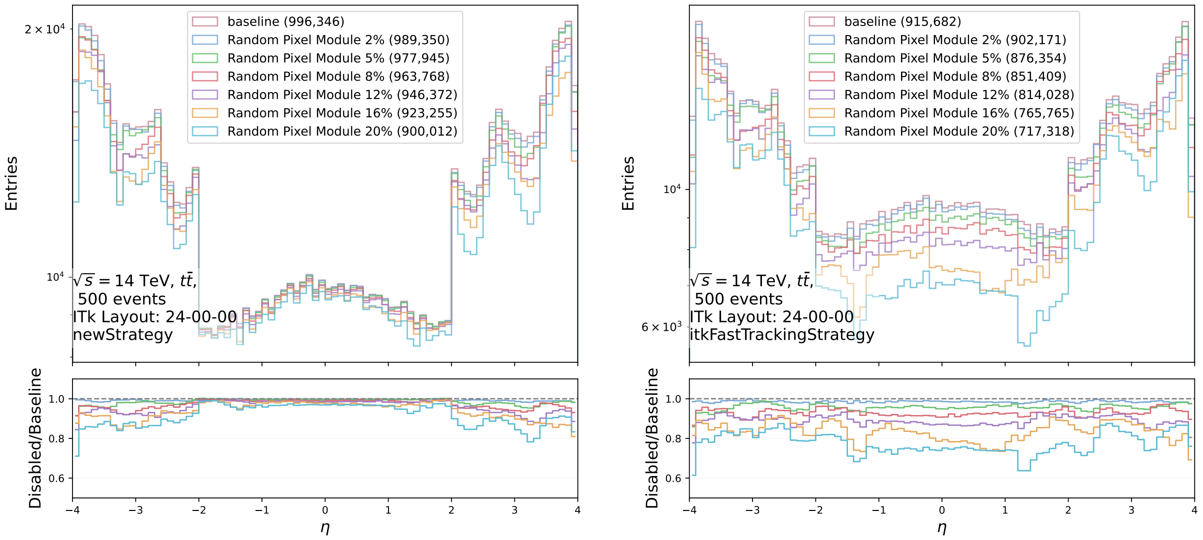
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.3.: Distribution of χ^2/N_{dof} for reconstructed tracks under increasing levels of random pixel cell masking (0% to 20%) for both reconstruction chains. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



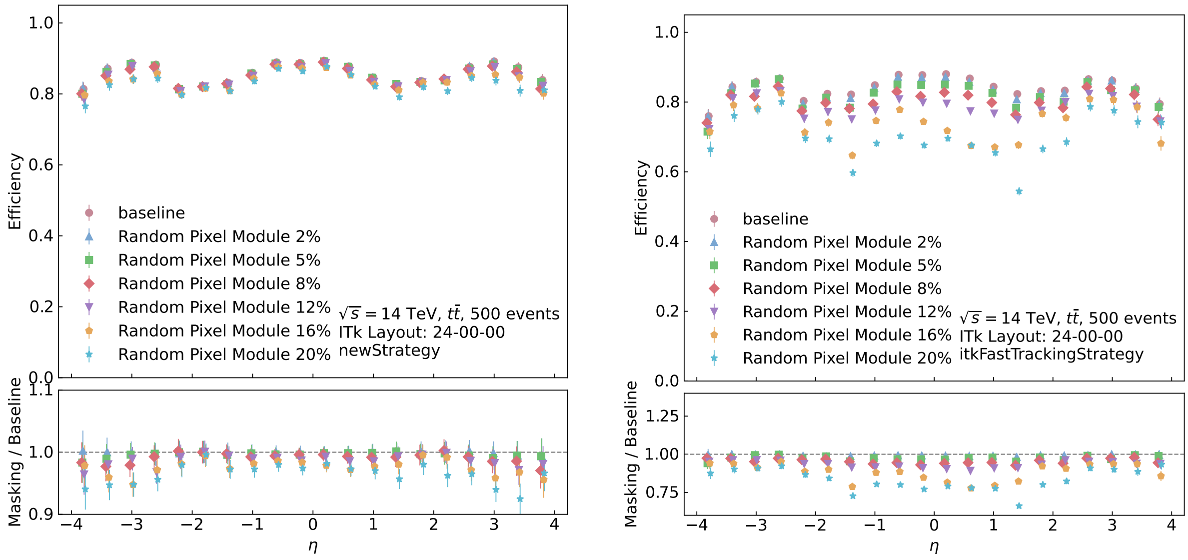
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.4.: Distribution of z_0 resolution width under increasing levels of random pixel cell masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



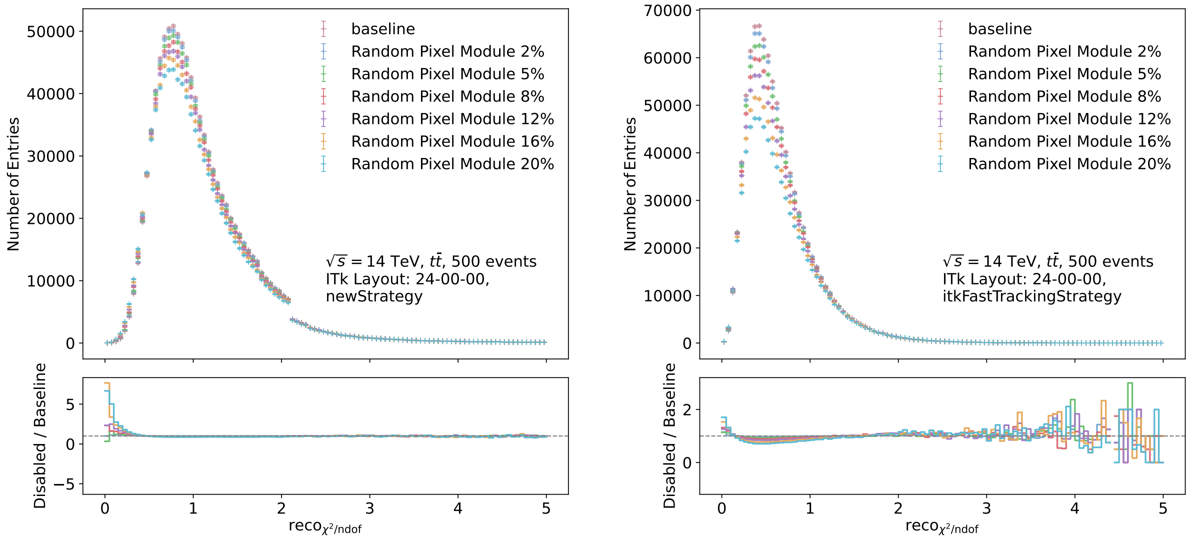
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.5.: Distributions of reconstructed track η under increasing levels of random pixel module masking (0% to 20%) for both reconstruction chains. Results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



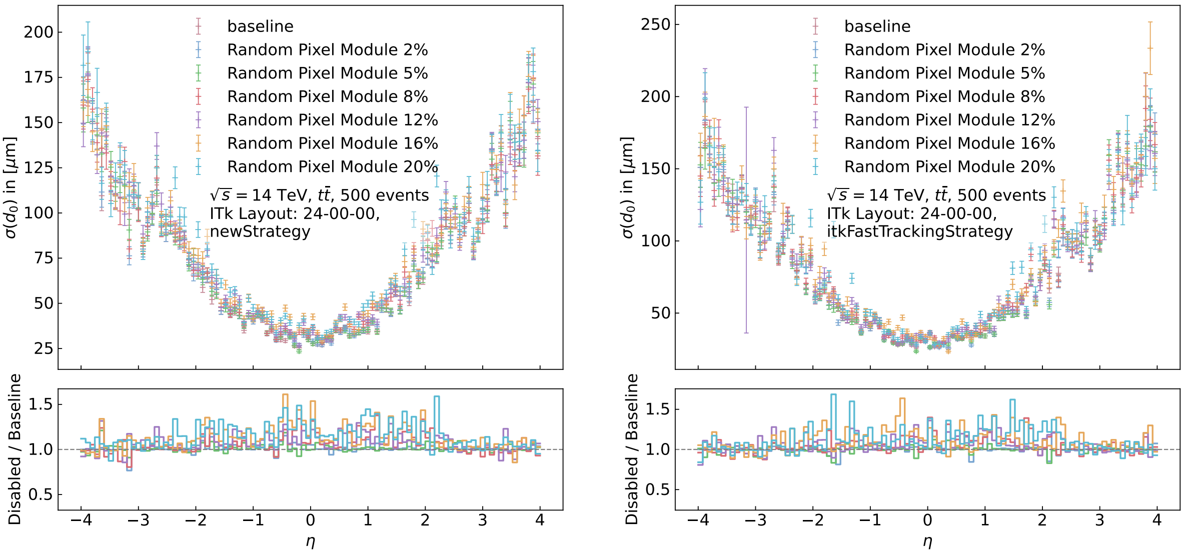
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.6.: Tracking efficiency in η bins under increasing levels of random pixel module masking (0% to 20%) for both reconstruction chains. Results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



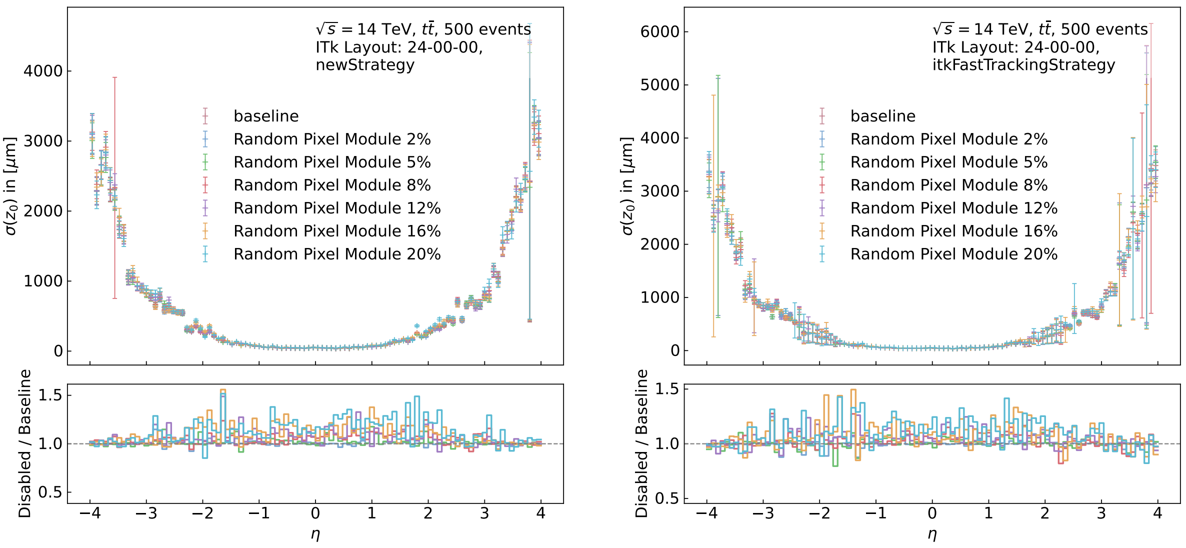
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.7.: Distribution of χ^2/N_{dof} for reconstructed tracks under increasing levels of random pixel module masking (0% to 20%) for both reconstruction chains. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



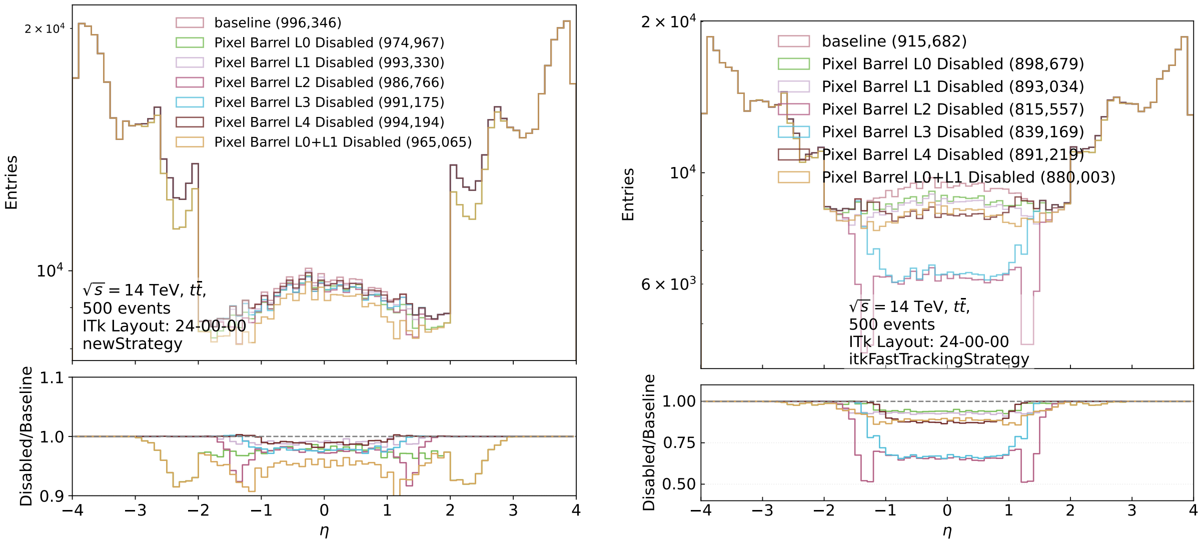
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.8.: Distribution of d_0 resolution width under increasing levels of random pixel module masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



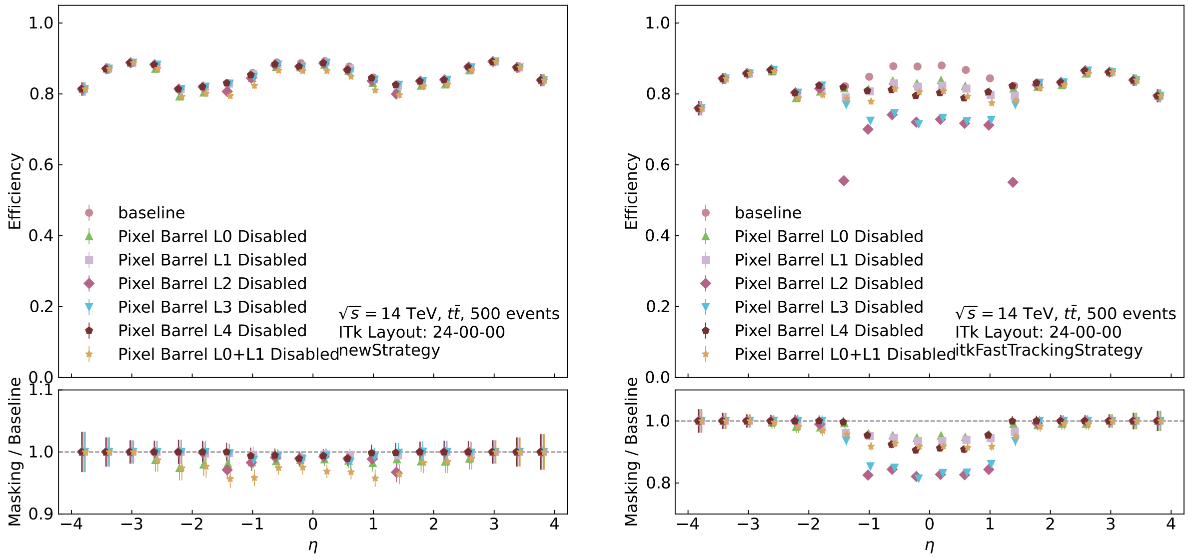
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.9.: Distribution of z_0 resolution width under increasing levels of random pixel module masking for both reconstruction chains: 0% (baseline), 2%, 5%, 8%, 12%, 16%, and 20%. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



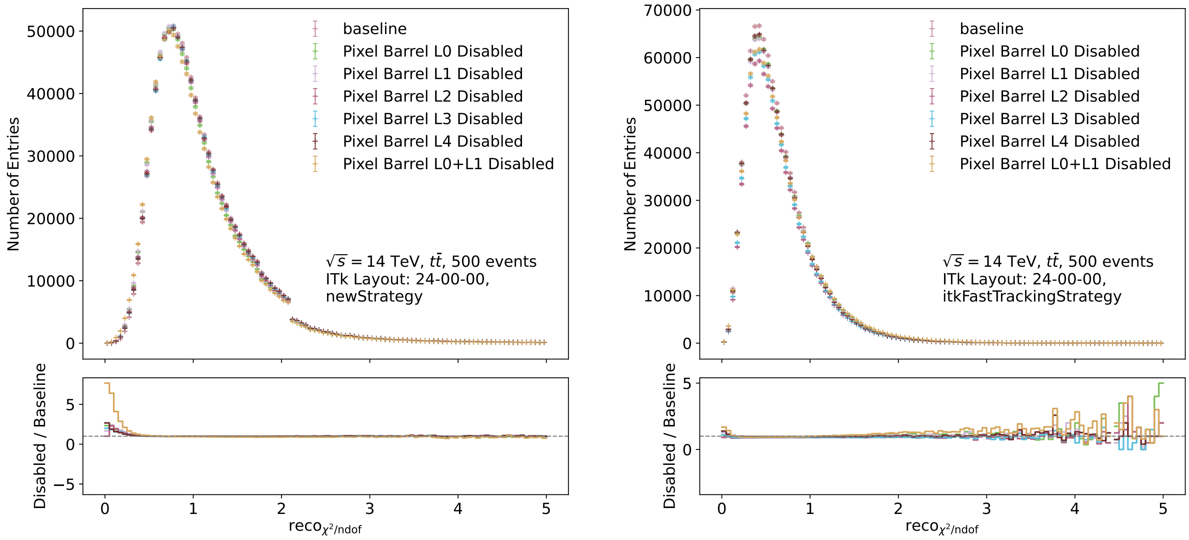
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.10.: η distributions of reconstructed tracks under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain. Results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



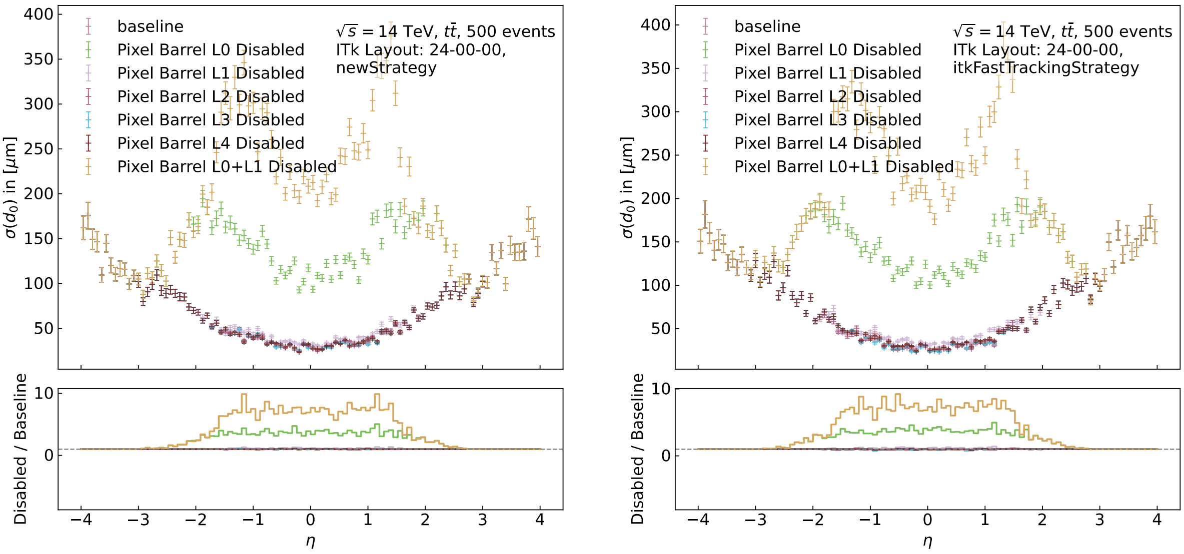
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.11.: Tracking efficiency in η bins under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain. Results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



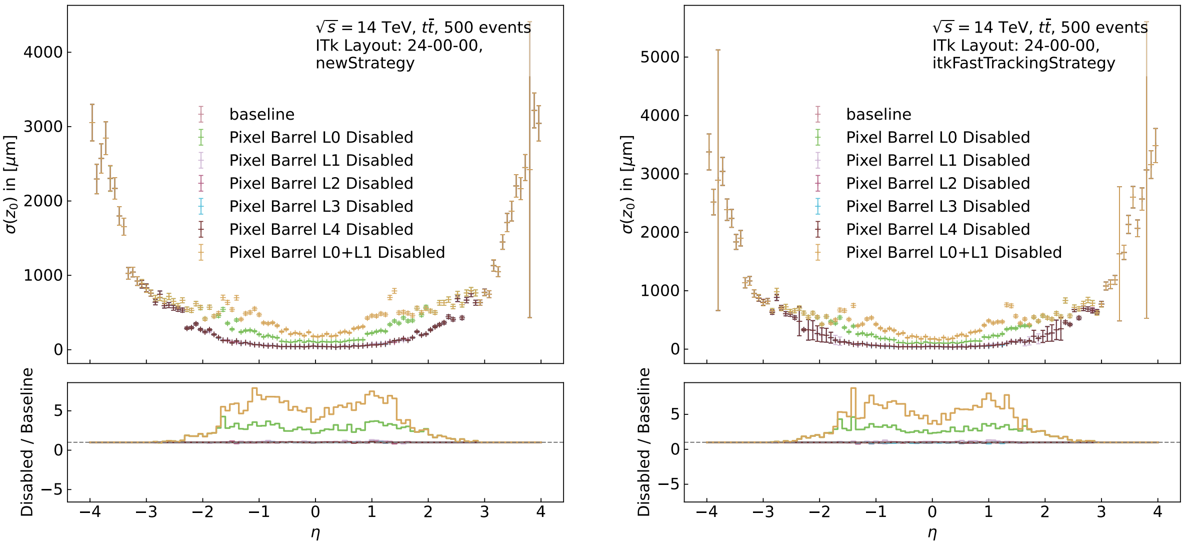
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.12.: Distribution of χ^2/N_{dof} for reconstructed tracks under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



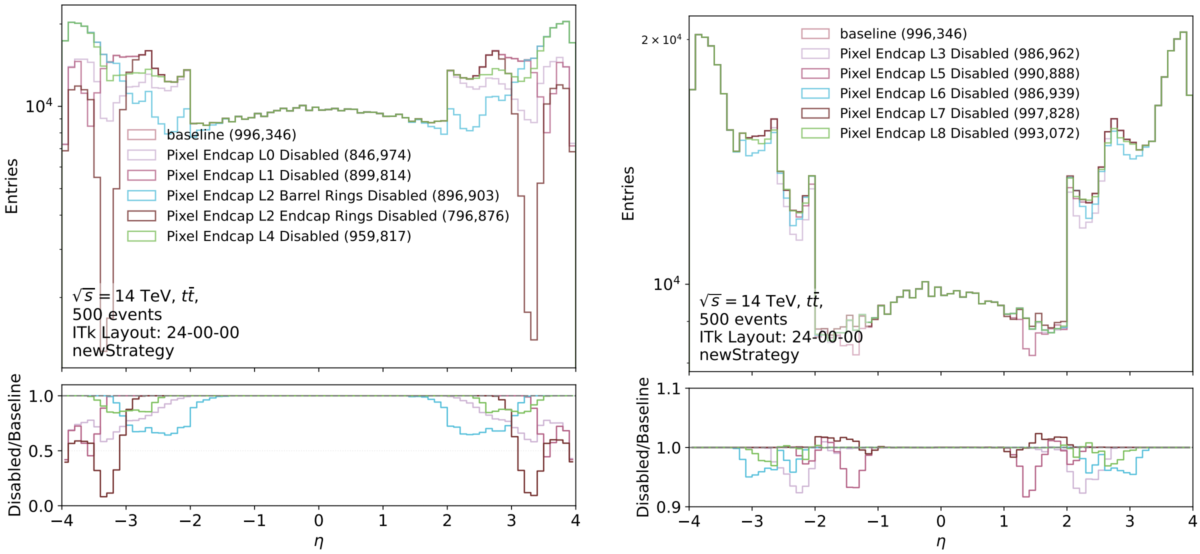
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.13.: Distribution of d_0 resolution width under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



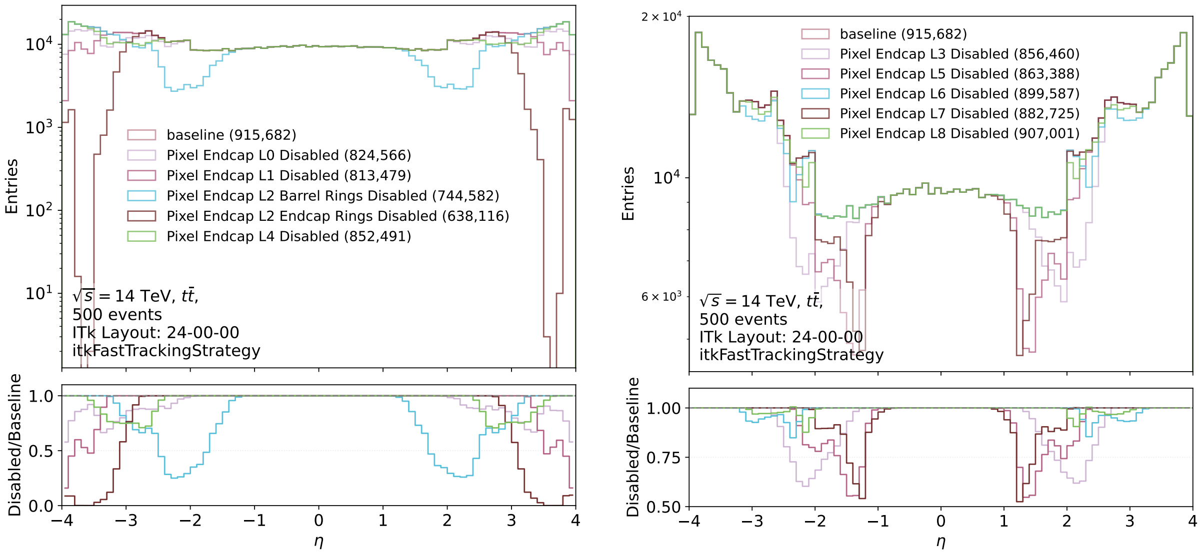
(a) With the default reconstruction chain. ***(b)*** With the fast track reconstruction chain.

Figure H.14.: Distribution of z_0 resolution width under individual pixel barrel layer deactivations (L0–L4) and the combined L0+L1 case for both reconstruction chain. Results are obtained using an independent 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



(a) Default reconstruction chain: deactivating layers L0, L1, L2, and L4.

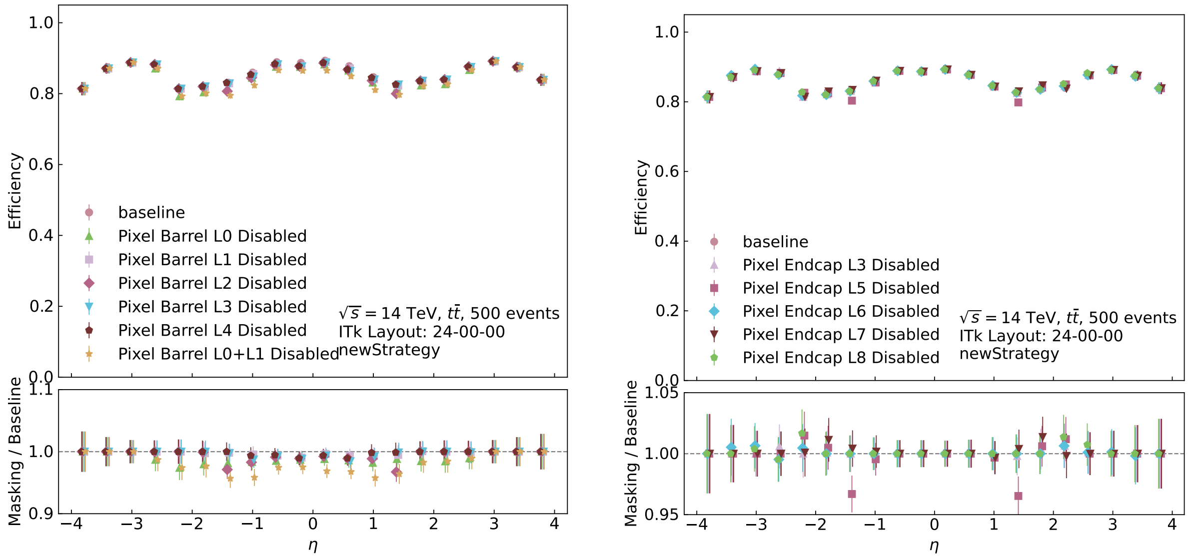
(b) Default reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.



(c) Fast reconstruction chain: deactivating layers L0, L1, L2, and L4.

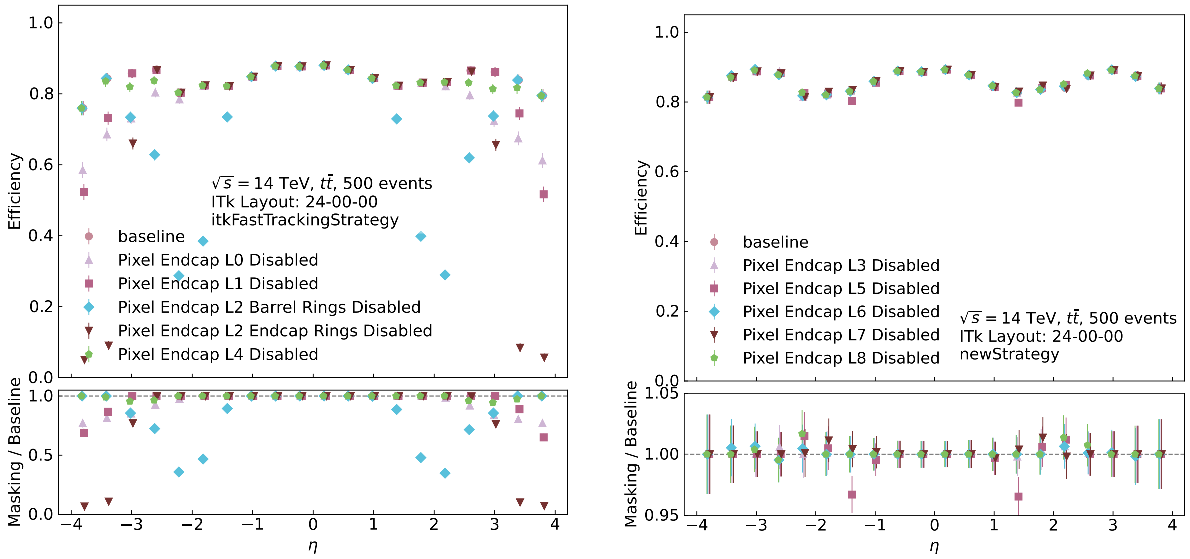
(d) Fast reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.

Figure H.15.: η distribution of reconstructed tracks under individual pixel endcap layer deactivations, grouped into early layers (L0, L1, L2, L4) and later layers (L3, L5, L6, L7, L8). Results are shown for both the default and fast track reconstruction chains. All results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



(a) Default reconstruction chain: deactivating layers L0, L1, L2, and L4.

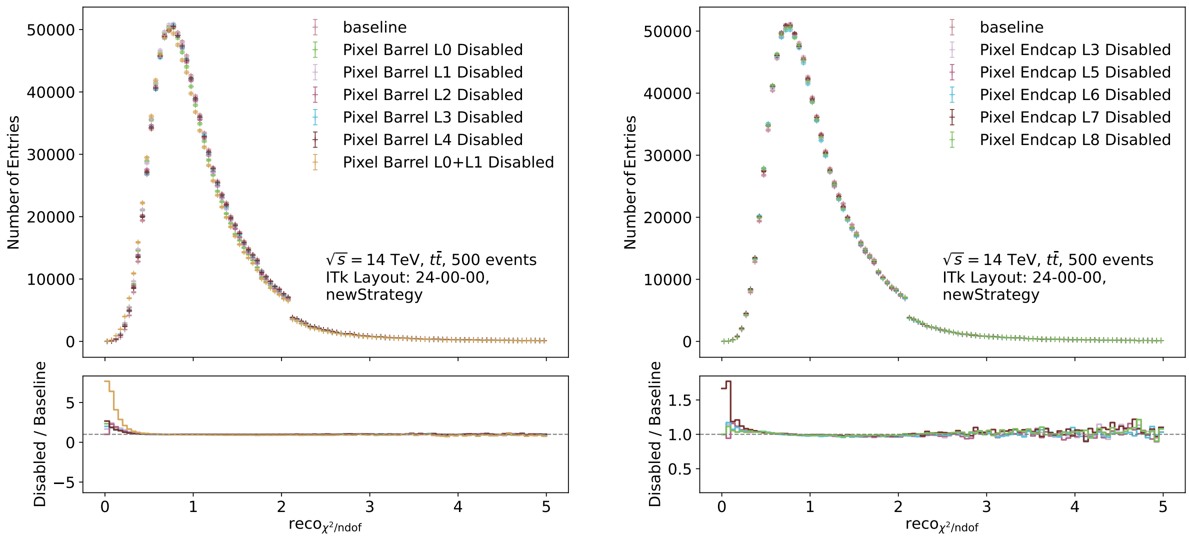
(b) Default reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.



(c) Fast reconstruction chain: deactivating layers L0, L1, L2, and L4.

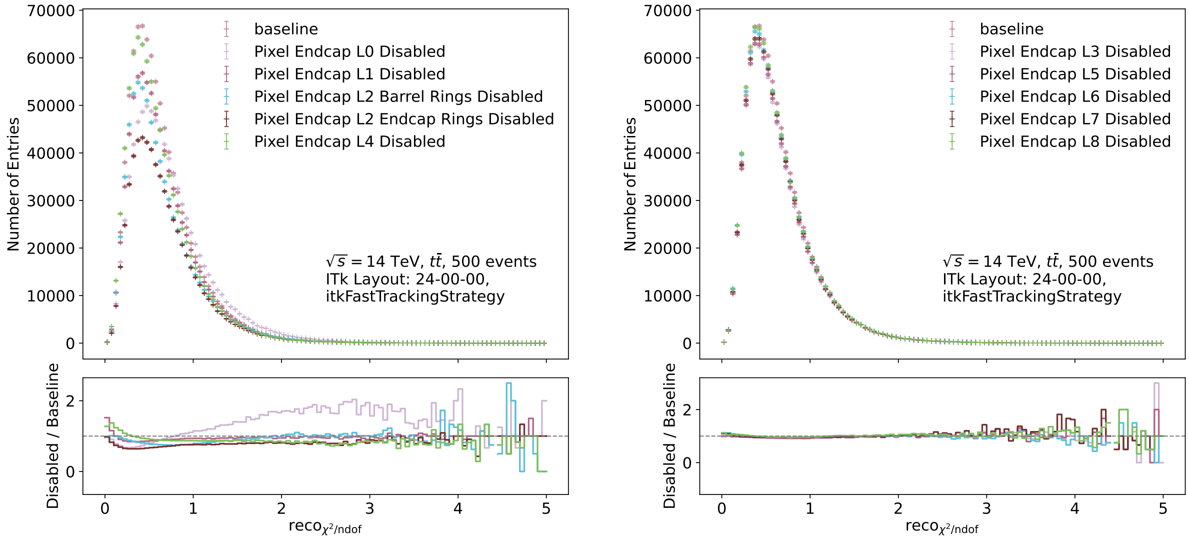
(d) Fast reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.

Figure H.16.: Tracking efficiency as a function of η under individual pixel endcap layer deactivations, grouped into early layers (L0, L1, L2, L4) and later layers (L3, L5, L6, L7, L8). Results are shown for both the default and fast track reconstruction chains. All results are based on a 500-event $t\bar{t}$ MC sample simulated with the ITk layout `24-00-00`.



(a) Default reconstruction chain: deactivating layers L0, L1, L2, and L4.

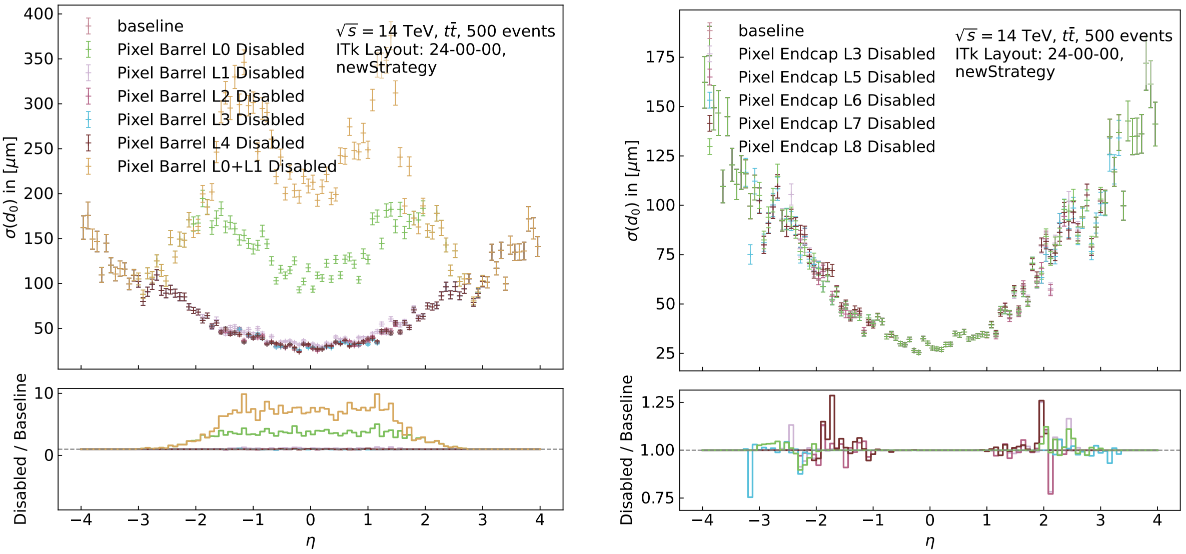
(b) Default reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.



(c) Fast reconstruction chain: deactivating layers L0, L1, L2, and L4.

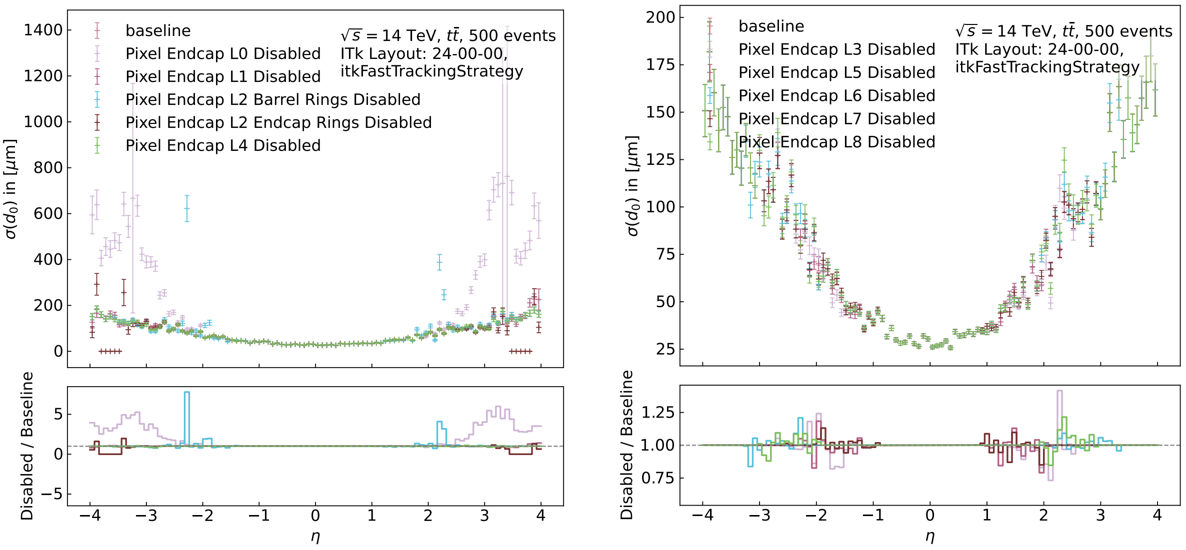
(d) Fast reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.

Figure H.17.: Distributions of χ^2/N_{dof} for reconstructed tracks under pixel barrel layer deactivation, grouped into early layers (L0, L1, L2, L4) and later layers (L3, L5, L6, L7, L8). Results are shown for both the default and fast track reconstruction chains, using a 500-event $t\bar{t}$ MC sample with the ITk layout `24-00-00`.



(a) Default reconstruction chain: deactivating layers L0, L1, L2, and L4.

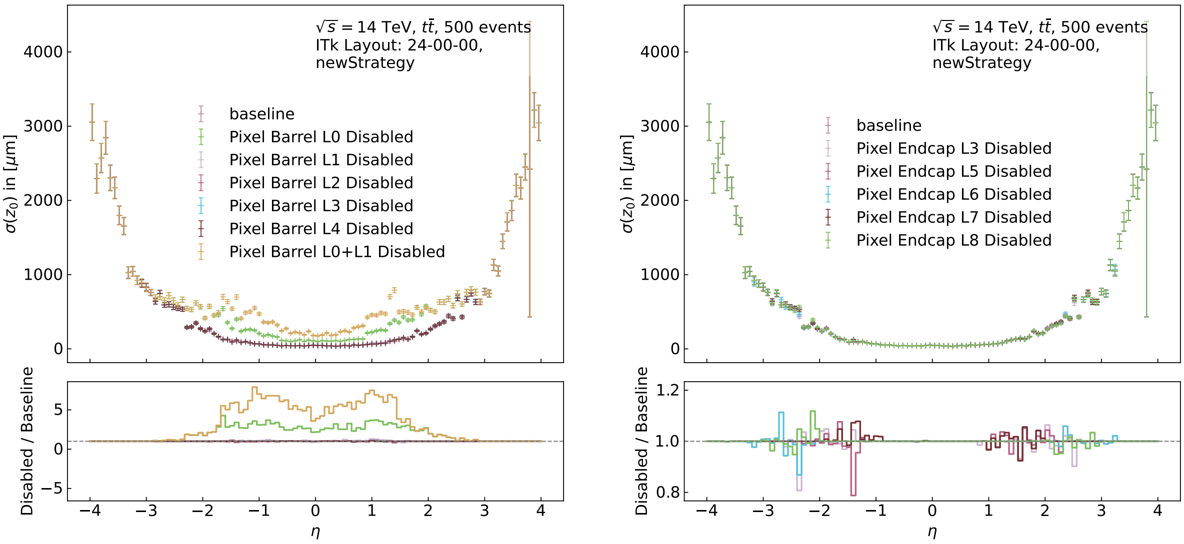
(b) Default reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.



(c) Fast reconstruction chain: deactivating layers L0, L1, L2, and L4.

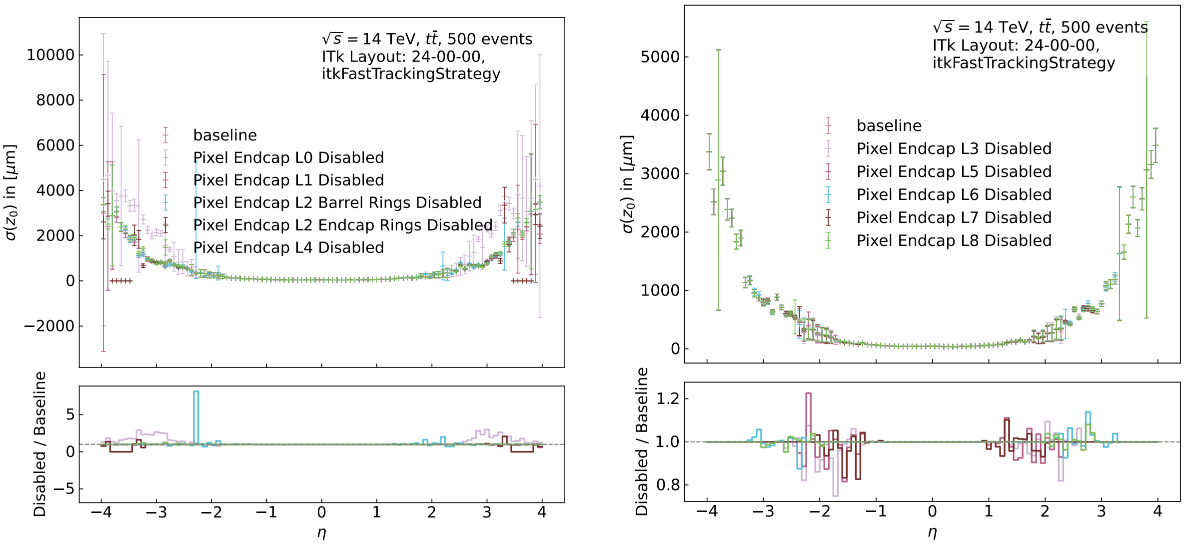
(d) Fast reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.

Figure H.18.: Distributions of d_0 resolution width under pixel barrel layer deactivation, grouped into early layers (L0, L1, L2, L4) and later layers (L3, L5, L6, L7, L8). Results are shown for both the default and fast track reconstruction chains, based on a 500-event $t\bar{t}$ MC sample with ITk layout `24-00-00`.



(a) Default reconstruction chain: deactivating layers L0, L1, L2, and L4.

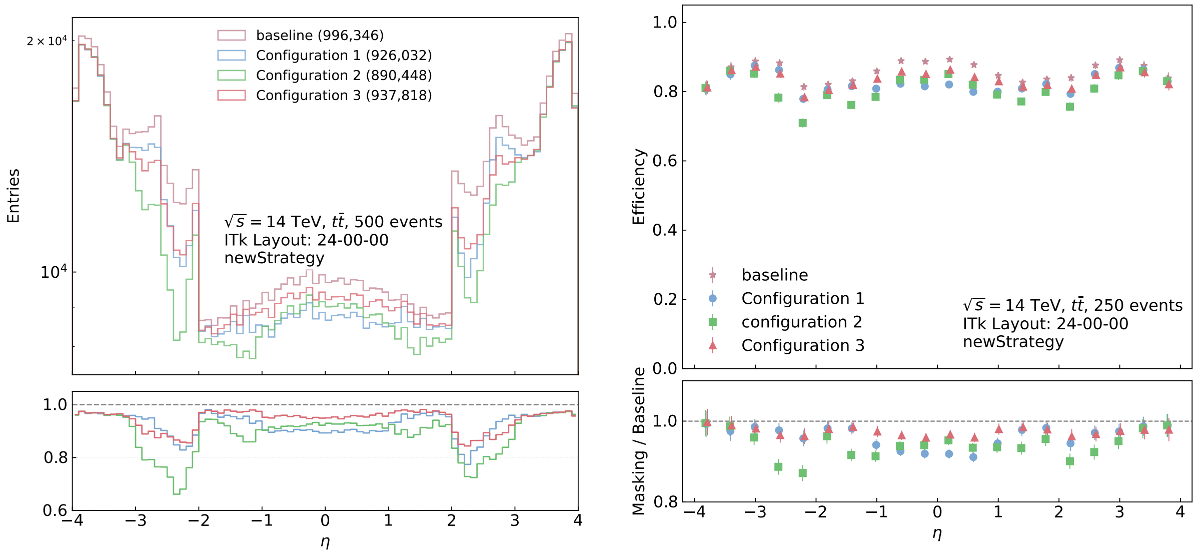
(b) Default reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.



(c) Fast reconstruction chain: deactivating layers L0, L1, L2, and L4.

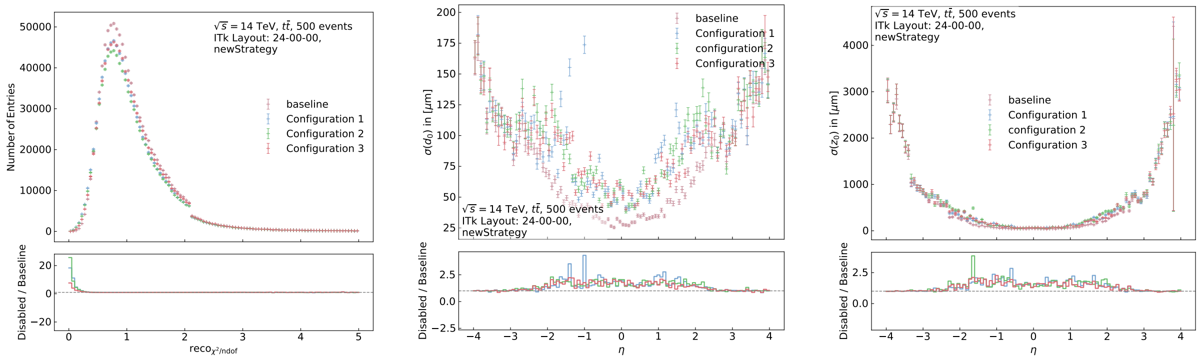
(d) Fast reconstruction chain: deactivating layers L3, L5, L6, L7, and L8.

Figure H.19.: Distributions of z_0 resolution width under pixel barrel layer deactivation, grouped into early layers (L0, L1, L2, L4) and later layers (L3, L5, L6, L7, L8). Results are shown for both the default and fast track reconstruction chains, based on a 500-event $t\bar{t}$ MC sample with ITk layout `24-00-00`.



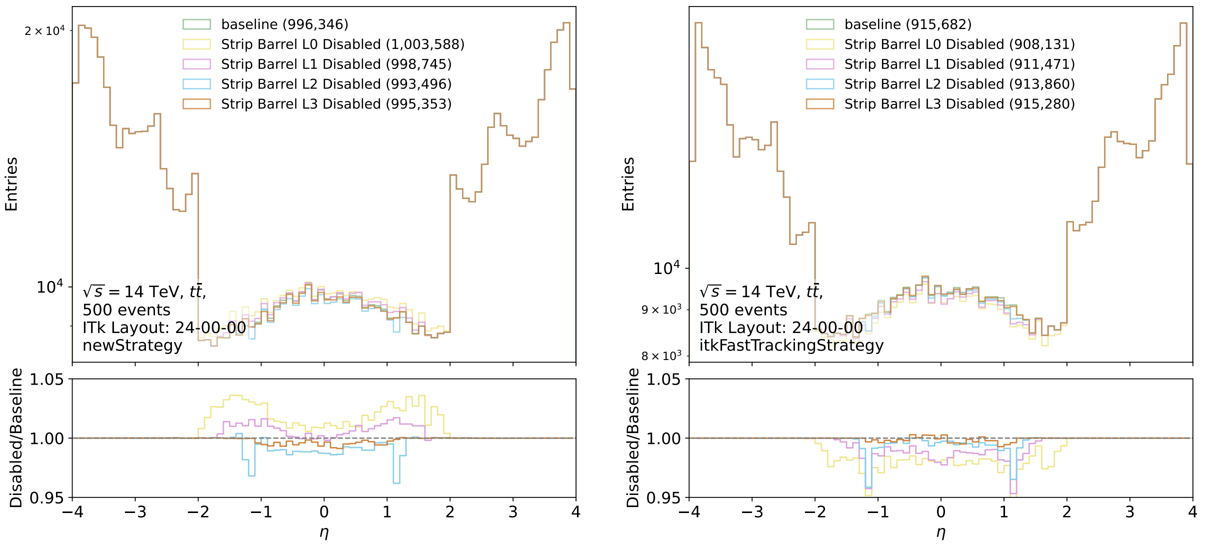
(a) Number of reconstructed tracks vs. η ***(b)*** Tracking efficiency vs. η

Figure H.20.: Tracking performance of the three selected configurations with the default reconstruction chain compared to the baseline, based on a 500-event $t\bar{t}$ MC sample with ITk layout 24-00-00.



(a) χ^2/N_{dof} ***(b)*** Width of resolution for d_0 ***(c)*** Width of resolution for z_0

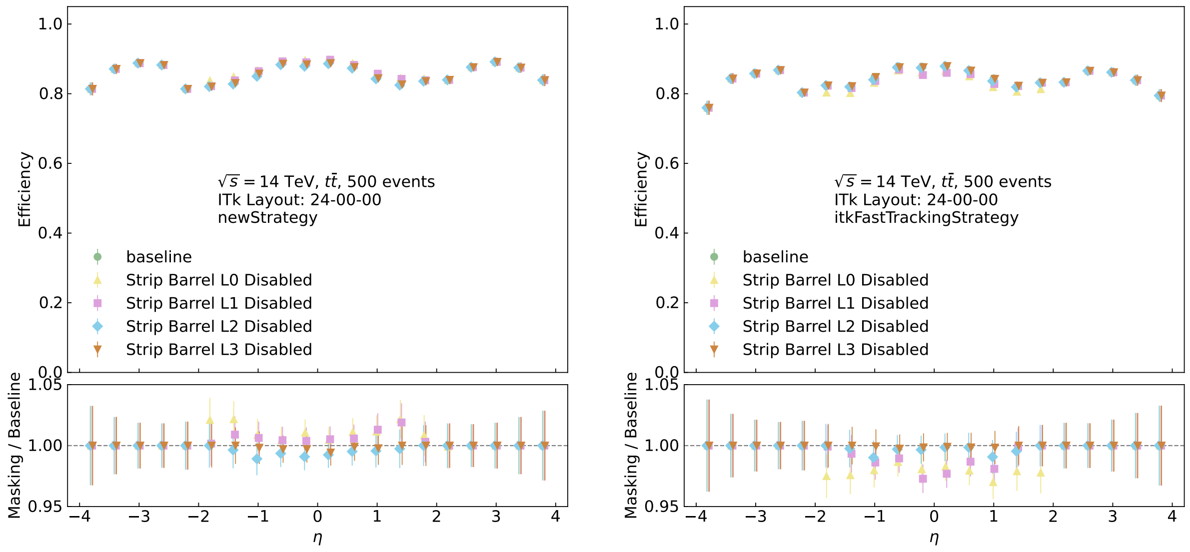
Figure H.21.: Track quality metrics for the three selected defect configurations with the default reconstruction chain compared to baseline, based on a 500-event $t\bar{t}$ MC sample with ITk layout 24-00-00.



(a) Default reconstruction chain.

(b) Fast reconstruction chain.

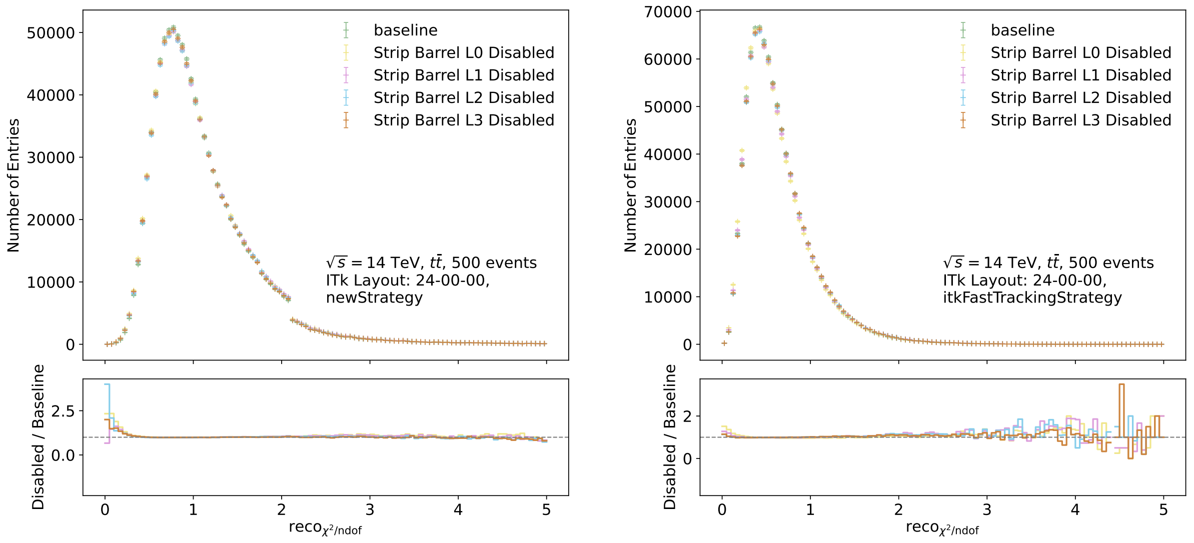
Figure H.22.: Distributions of reconstructed track η under strip barrel layer deactivation, shown for both reconstruction chains. Results are based on a 500-event $t\bar{t}$ MC sample with the ITk layout 24-00-00.



(a) Default reconstruction chain.

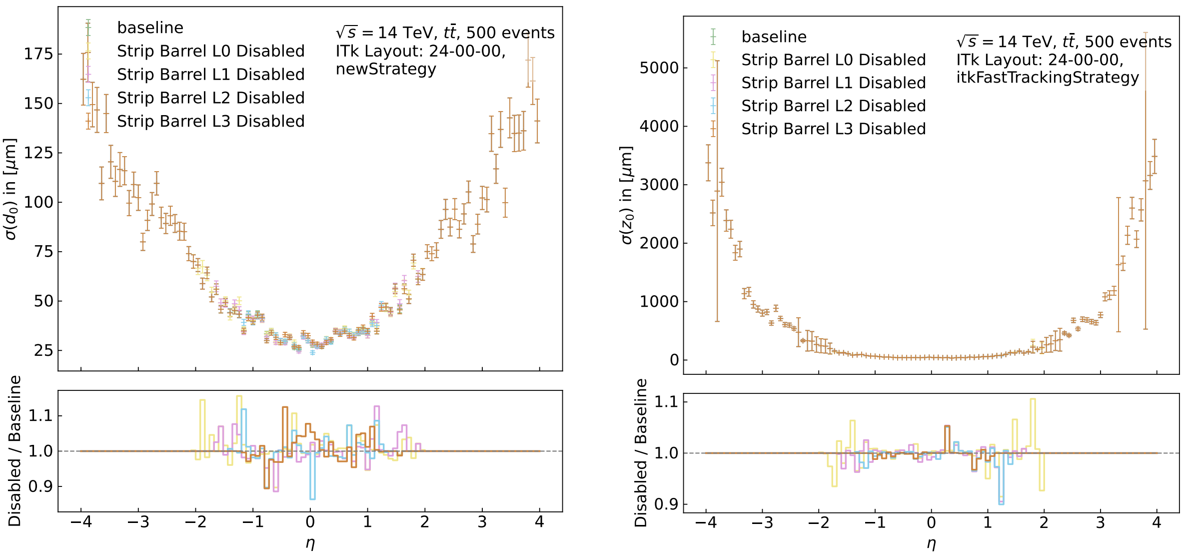
(b) Fast reconstruction chain.

Figure H.23.: Tracking efficiency as a function of η under strip barrel layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout 24-00-00.



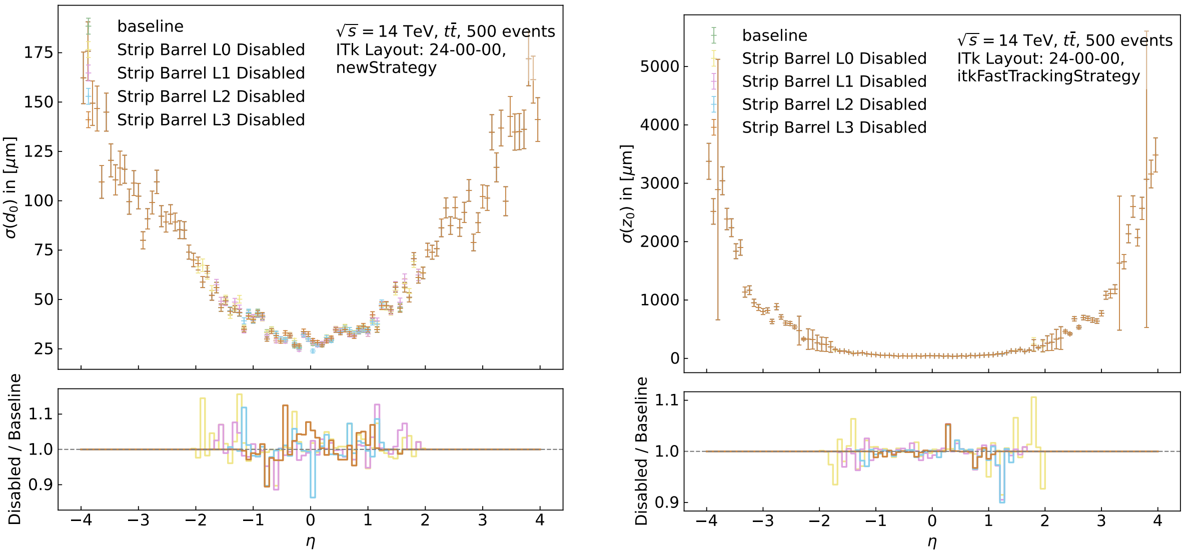
(a) Default reconstruction chain. ***(b)*** Fast reconstruction chain.

Figure H.24.: Distributions of χ^2/N_{dof} for reconstructed tracks under strip barrel layer deactivation. Results are shown for both reconstruction chains, using a 500-event $t\bar{t}$ MC sample with the ITk layout `24-00-00`.



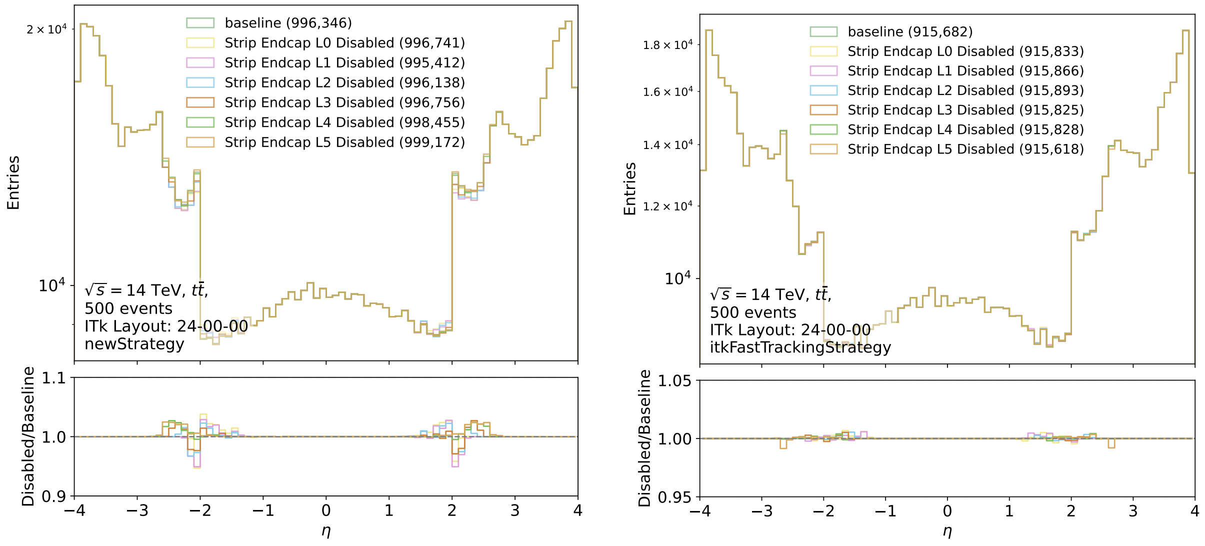
(a) Default reconstruction chain. ***(b)*** Fast reconstruction chain.

Figure H.25.: Distributions of d_0 resolution width under strip barrel layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout `24-00-00`.



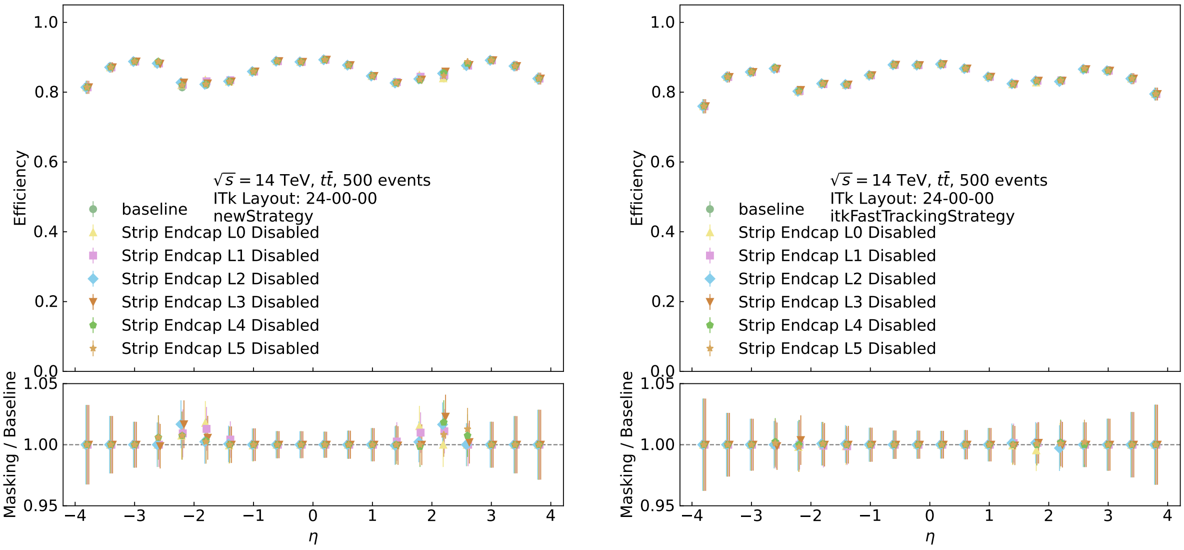
(a) Default reconstruction chain. *(b)* Fast reconstruction chain.

Figure H.26.: Distributions of z_0 resolution width under strip barrel layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout 24-00-00.



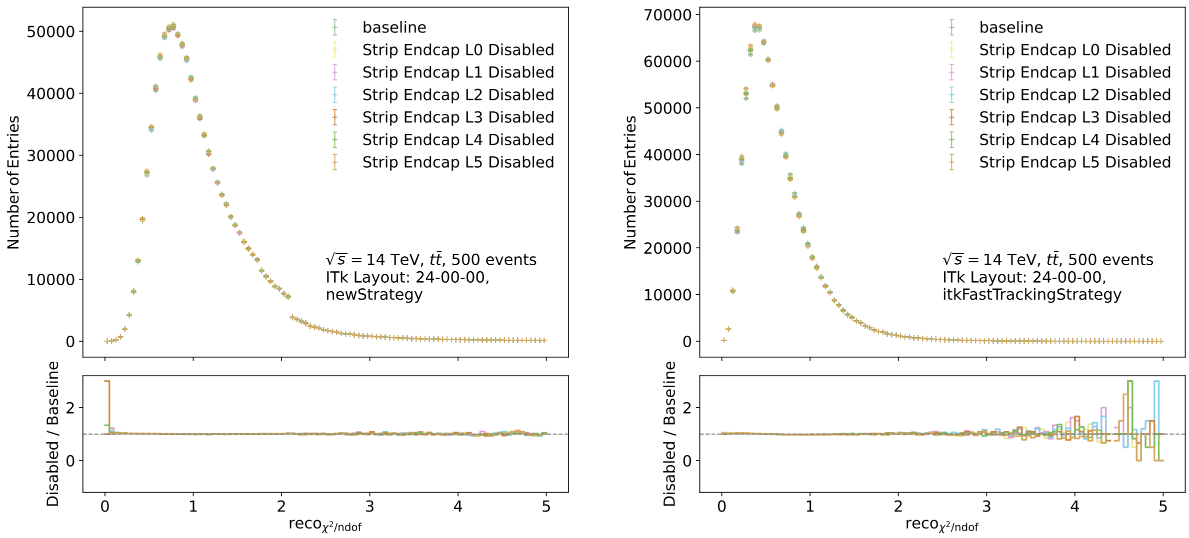
(a) Default reconstruction chain. *(b)* Fast reconstruction chain.

Figure H.27.: Distributions of reconstructed track η under strip endcap layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout 24-00-00.



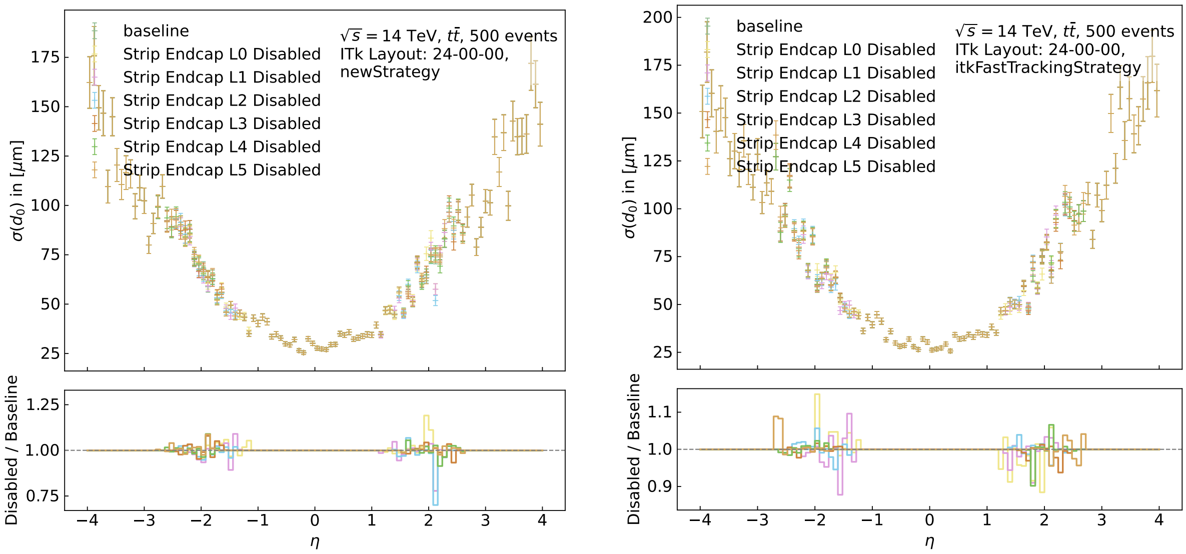
(a) Default reconstruction chain. ***(b)*** Fast reconstruction chain.

Figure H.28.: Tracking efficiency in η bins under strip endcap layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout `24-00-00`.



(a) Default reconstruction chain. ***(b)*** Fast reconstruction chain.

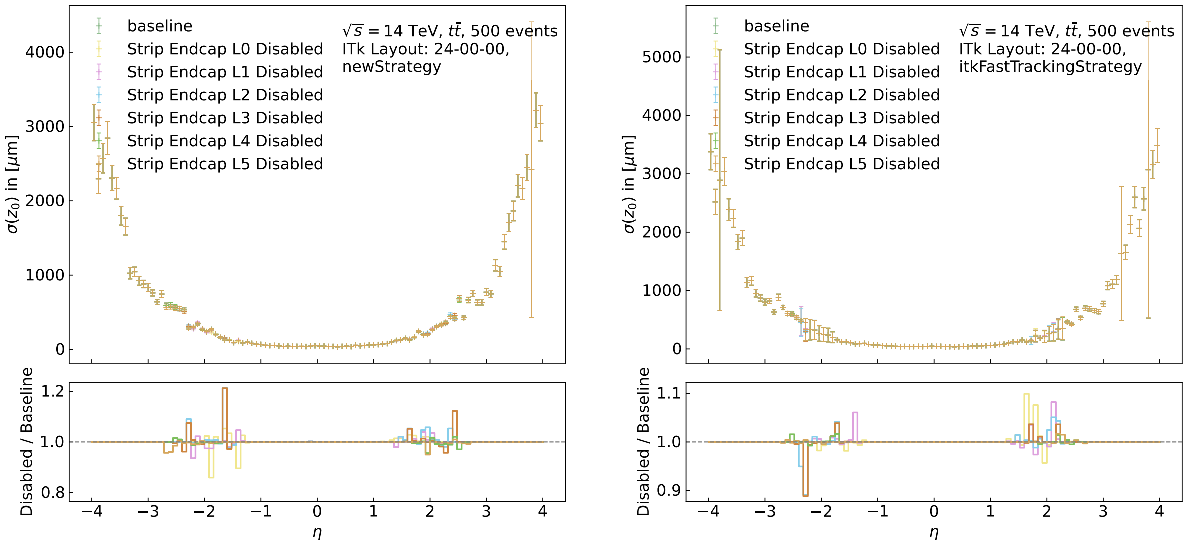
Figure H.29.: Distributions of χ^2/N_{dof} for reconstructed tracks under strip endcap layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout `24-00-00`.



(a) Default reconstruction chain.

(b) Fast reconstruction chain.

Figure H.30.: Distributions of d_0 resolution width under progressive strip endcap layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout 24-00-00.



(a) Default reconstruction chain.

(b) Fast reconstruction chain.

Figure H.31.: Distributions of z_0 resolution width under progressive strip endcap layer deactivation, shown for both reconstruction chains. Based on a 500-event $t\bar{t}$ MC sample with the ITk layout 24-00-00.

Bibliography

[1] S. L. Glashow. “Partial-symmetries of weak interactions”. In: *Nuclear Physics* 22.4 (1961), pp. 579–588. ISSN: 0029-5582.

[2] A. Salam. “Weak and Electromagnetic Interactions”. In: *Conf. Proc. C* 680519 (1968), pp. 367–377.

[3] S. Weinberg. “A Model of Leptons”. In: *Phys. Rev. Lett.* 19 (21 Nov. 1967), pp. 1264–1266.

[4] M. Gell-Mann. “A Schematic Model of Baryons and Mesons”. In: *Phys. Lett.* 8 (1964), pp. 214–215.

[5] CMS Collaboration. “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 30–61. arXiv: `1207.7235 [hep-ex]`.

[6] ATLAS Collaboration. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 1–29. arXiv: `1207.7214 [hep-ex]`.

[7] G. Bertone, D. Hooper, and J. Silk. “Particle dark matter: Evidence, candidates and constraints”. In: *Phys. Rept.* 405 (2005), pp. 279–390. arXiv: `hep-ph/0404175`.

[8] Super-Kamiokande Collaboration. “Evidence for oscillation of atmospheric neutrinos”. In: *Phys. Rev. Lett.* 81 (1998), pp. 1562–1567. arXiv: `hep-ex/9807003`.

[9] A. D. Sakharov. “Violation of CP Invariance, C asymmetry, and baryon asymmetry of the universe”. In: *Pisma Zh. Eksp. Teor. Fiz.* 5 (1967), pp. 32–35.

[10] L. R. Evans. “The Large Hadron Collider Project”. In: (1997).

[11] ATLAS Collaboration. *Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment.* Tech. rep. CERN, 2012.

[12] ATLAS Collaboration. *Study of tracking and software performance of the ATLAS Inner Tracker in staged-installation scenarios at the High-Luminosity LHC*. Tech. rep. All figures including auxiliary figures are available at `https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2021-040`. CERN, 2021.

[13] ATLAS Collaboration. “The ATLAS Simulation Infrastructure”. In: *Eur. Phys. J. C* 70 (2010), pp. 823–874. arXiv: `1005.4568 [physics.ins-det]`.

[14] ATLAS Collaboration. *Athena-SiSPSeededTrackFinder.cxx (Athena GitLab source)*. `https://gitlab.cern.ch/atlas/athena/-/blob/main/InnerDetector/InDetRecAlgs/SiSPSeededTrackFinder/src/SiSPSeededTrackFinder.cxx`. Accessed: 30 Aug 2025; commit hash: `074ec84859f21d207aedce87d1d82c7bc7b953dc`.

[15] ATLAS Collaboration. *Fast Track Reconstruction for HL-LHC*. Tech. rep. All figures including auxiliary figures are available at `https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2019-041`. CERN, 2019.

[16] ATLAS Collaboration. “The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run 3”. In: *JINST* 19.05 (2024), P05063. arXiv: `2305.16623 [physics.ins-det]`.

[17] ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008). Also published by CERN Geneva in 2010, S08003.

[18] ATLAS Collaboration. *The simulation principle and performance of the ATLAS fast calorimeter simulation FastCaloSim*. Tech. rep. ATL-PHYS-PUB-2010-013. CERN, 2010.

[19] ATLAS Collaboration. “Resolution of the ATLAS muon spectrometer monitored drift tubes in LHC Run 2”. In: *JINST* 14 (2019).

[20] K. Hanagaki et al. *Experimental Techniques in Modern High-Energy Physics: A Beginner’s Guide*. 1st ed. Vol. 1040. Lecture Notes in Physics. Springer Tokyo, 2023. ISBN: 978-4-431-56929-9.

[21] ATLAS Collaboration. *The ATLAS Upgrade for the HL-LHC*. Tech. rep. ATL-UPGRADE-PUB-2025-001. CERN, 2025.

[22] ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider: A Description of the Detector Configuration for Run 3”. In: *JINST* 19 (2024), P05063.

[23] M. Wu. “A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade”. In: *JPS Conference Proceedings*. Vol. 42. Presented on behalf of the ATLAS Collaboration at VERTEX 2022, Tateyama, Japan, 24–28 Oct 2022. ATL-HGTD-PROC-2023-002. 2024, p. 011026.

[24] N. Dos Santos. “ATLAS Trigger and Data Acquisition Upgrades for the High-Luminosity LHC”. In: *PoS LHCP2023*. Vol. LHCP2023. Presented on behalf of the ATLAS Collaboration at the 11th Large Hadron Collider Physics Conference , Belgrade, Serbia, 22–26 May 2023. ATL-DAQ-PROC-2024-002. 2024, p. 251.

[25] ATLAS Collaboration. *Expected tracking and related performance with the updated ATLAS Inner Tracker layout at the HL-LHC*. Tech. rep. ATL-PHYS-PUB-2021-024. CERN, 2021.

[26] ATLAS Collaboration. “Expected tracking performance of the ATLAS Inner Tracker at the High-Luminosity LHC”. In: *JINST* 20 (2025), P02018. arXiv: `2412.15090 [hep-ex]`.

[27] ATLAS Collaboration. *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*. Tech. rep. CERN-LHCC-2017-021, ATLAS-TDR-030. CERN, 2017.

[28] M. Garcia-Sciveres. *RD53C Chip Manual*. RD53 Public Document CERN-RD53-PUB-24-001. Published: 27 Feb 2024. RD53 Collaboration, 2024.

[29] F. M. Sanchez. “ATLAS ITk Pixel Detector Overview”. In: *Proceedings of Science (ICHEP2024)*. ATL-ITK-PROC-2024-038, presented at ICHEP 2024, Prague, Czech Republic, 18–24 Jul 2024. 2025, p. 878.

[30] ATLAS Collaboration. *Technical Design Report for the ATLAS Inner Tracker Strip Detector*. Tech. rep. CERN-LHCC-2017-005, ATLAS-TDR-025. (cit. on p. 10). CERN, 2017.

[31] F. Munoz Sanchez and A. Collaboration. “Carbon Based Local Supports for the ATLAS ITk-Pixel Detector”. In: *Proceedings of the 10th International Workshop on Semiconductor Pixel Detectors for Tracking and Imaging (PIXEL 2022)*. Vol. Pixel2022. Presented on behalf of the ATLAS Collaboration in Santa Fe, USA, 11–16 Dec 2022. 2023, p. 077.

[32] ATLAS Collaboration. “ATLAS Inner Tracker (ITk) Schematics”. General Photo. 2024.

[33] L. Meng et al. *Pixel column issue in the ATLAS Inner Tracker modules*. Prepared for submission to JINST, 20th Anniversary Trento Workshop on Advanced Silicon Radiation Detectors, 4–6 February 2025, FBK, Trento, Italy. 2025.

[34] ATLAS Collaboration. *ATLAS Software Documentation: Reco$_t$fintroduction.* `https://atlassoftwaredocs.web.cern.ch/internal-links/tracking-tutorial/recotf/`. Last Update: 23 Aug 2024.

[35] D. Giordano and E. Santorinaiou. “Next Generation of HEP CPU Benchmarks”. In: *Journal of Physics: Conference Series*. Vol. 1525. 19th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2019), 11â15 March 2019, Saas-Fee, Switzerland. 2020, p. 012073.

[36] ATLAS Collaboration. *ATLAS HL-LHC Computing Conceptual Design Report*. Tech. rep. CERN, 2020.

[37] ATLAS Collaboration. *TrackStateOnSurface.h Athena / Tracking TrkEvent TrkTrack*. 2025.

Acknowledgements

Somehow, this thesis is finally done.

I would first like to express my sincere gratitude to my supervisor, Stan, for giving me the opportunity to work on this thesis at a time when I knew almost nothing about the ITk detector. Without his trust and willingness to take me on, I might not have been able to work on any thesis at all. I'm truly grateful, this has been something genuinely good for me.

I would also like to thank Kira, who directly supervised my work during this project. Her feedback and patience were important throughout the whole project and the writing of this thesis. I am also aware that this project may have taken up some of your holiday time, and I truly appreciate your support nonetheless.

I would also like to thank my second referee, Dr. Benjamin Schwenker for taking the time to read through this work. I know it turned out to be quite lengthy, a direct result of the nature of the work, which could make it a demanding read, I sincerely thank you for your time. My thanks also go to the Upgrade Tracking group for attending my presentation and for providing a key suggestion that led me to rethink and restructure this thesis.

I thank my family for their years of support.

As this may be the only thesis I ever write, I dedicate it to my memory, time, and self-deception.

Erklärung nach §13(9) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen:
Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den 3. Dezember 2025

(Yinge Chen)