Boosting CPU Efficiency in ATLAS Inner Detector Reconstruction with Track Overlay

Fang-Ying Tsai[1], Dominik Duda[2], Stephen Jiggins[3], Martina Javurkova[4], William Leight[4], and John Derek Chapman[5], on behalf of the ATLAS Computing Activity

[1]Stony Brook University, NY, USA
[2]University of Edinburgh, Scotland, UK
[3]DESY, Hamburg, Germany
[4]University of Massachusetts Amherst, MA, USA
[5]University of Cambridge, Cambridge, UK

E-mail: fang-ying.tsai@cern.ch

Abstract. In response to the rising CPU consumption and storage demands of the High-Luminosity Large Hadron Collider (HL-LHC), efforts are ongoing to enhance the CPU processing efficiency of reconstruction within the ATLAS inner detector (ID). The track overlay approach involves pre-reconstructing pile-up (PU) tracks and subsequently running reconstruction exclusively on hard-scatter (HS) tracks. This approach conserves valuable CPU resources by concentrating on events of interest. A key component of the workflow is the incorporation of machine learning (ML)-based decision processes. ML decisions guide the selection of events suitable for track overlay, with events in denser environments continuing to use the standard overlay. This strategy ensures an efficient use of resources, balancing efficiency and precision in ID reconstruction. This presentation focuses on constructing the ML model and verifying the workflow with ML decisions. The track overlay approach is demonstrated to improve CPU usage and reduce the size of standard data format files in the Run 3 detector setup.

1 Introduction

This article builds upon previous work presented in the CHEP2023 proceedings [1], which provided an extensive overview of the track overlay methodology. Traditionally, the ATLAS [2] reconstruction chain utilizes a Monte Carlo (MC) overlay, where pile-up events are overlaid on hard-scatter ones after the digitization stage (see Figure 1). While this approach accurately describes the data, it can be computationally intensive and time-consuming. To mitigate these issues, we propose the track overlay method (see Figure 2). The main ideas behind track overlay include:

- Using previously reconstructed pile-up tracks to reduce reconstruction time.
- Maintaining the reconstruction efficiency for hard scatter events at a similar level to MC overlay, while simultaneously enhancing overall computational efficiency by reducing CPU usage.

To manage scenarios where track overlay may not be suitable, such as events with many pile-up vertices or high p_{T} jets containing multiple tracks in dense environments with extremely collimated particles, we have developed a hybrid strategy. This approach integrates machine learning (ML) algorithms to dynamically identify and prioritize events that benefit most from the efficiency gains of track overlay. Conversely, when conditions imply that track overlay would result in poor accuracy, the ATLAS reconstruction chain can revert to traditional MC overlay methods.

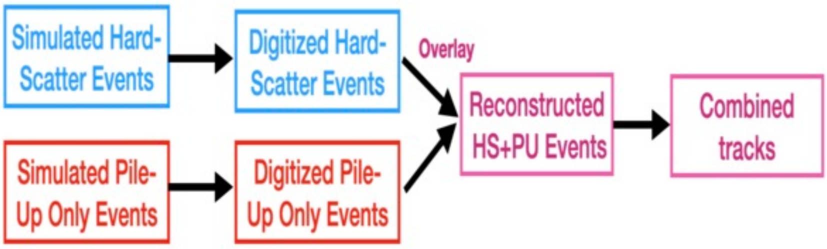

Simulated Hard-Scatter Events
Digitized Hard-Scatter Events
Overlay
Reconstructed HS+PU Events
Combined tracks
Simulated Pile-Up Only Events
Digitized Pile-Up Only Events

Figure 1: Diagram of the MC overlay process, where simulated pile-up collisions are digitized separately and then combined, followed by the reconstruction algorithm.

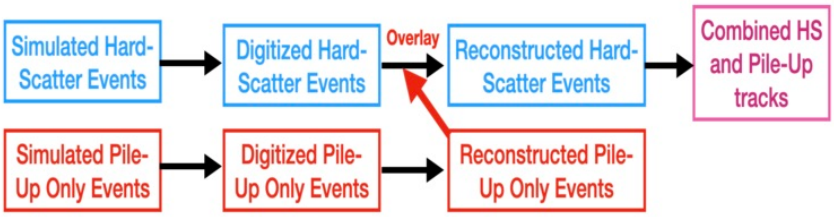

Simulated Hard-Scatter Events
Digitized Hard-Scatter Events
Overlay
Reconstructed Hard-Scatter Events
Combined HS and Pile-Up tracks
Simulated Pile-Up Only Events
Digitized Pile-Up Only Events
Reconstructed Pile-Up Only Events

Figure 2: In the track overlay method, pre-reconstructed pile-up tracks are overlaid onto HS digitized events, followed by the reconstruction process.

2 Hybrid overlay validation

The hybrid overlay method incorporates a Deep Neural Network (DNN) to dynamically determine whether to use track overlay or MC overlay on an event-by-event basis. This decision is guided by a comprehensive set of factors to ensure optimal efficiency and accuracy in the reconstruction process. When training the DNN, the labeling criteria for the ML model involve assigning a 'Diff' label to each track based on its reconstruction status. A 'Diff' label is assigned if a track is matched[1] to a generator-level particle in the track overlay but not in the MC overlay. The DNN model is trained using a combination of 14 input features at the truth (generator) level, which captures various kinematic properties, event topology, and pile-up information. These features are combined into a single structure. The following variables are considered, where ΔR represents the angular distance between two objects:

- p_x, p_y, p_z, e, and p_T of the track.
- Density within a $\Delta R = 0.2$ and $\Delta R = 0.5$ region.
- Sum of distances within a $\Delta R = 0.2$ and $\Delta R = 0.5$ region.
- Sum of p_T of the track within a $\Delta R = 0.2$ and $\Delta R = 0.5$ region.
- Number of pile-up tracks.
- Number of tracks in the event.
- Transverse momentum of the event.

The learning rate is adjusted during training using a decay function. Initially, the learning rate remains constant for the first two epochs, after which it decreases exponentially with the epoch number to ensure stable convergence. The model is trained using a sample of dijet events, where the leading jet p_T lies in the range 1.8–2.5 TeV, and is evaluated accordingly. The DNN model assigns discriminant scores to each track, indicating the likelihood of it being impacted by complex environments. This includes scenarios with high track density, such as in the presence of pile-up overlapping tracks inside jets, where hits are

[1] The determination of a truth match is based on the probability that a reconstructed hit matches with a simulated hit. If the probability exceeds 0.5, the hits are considered matched.

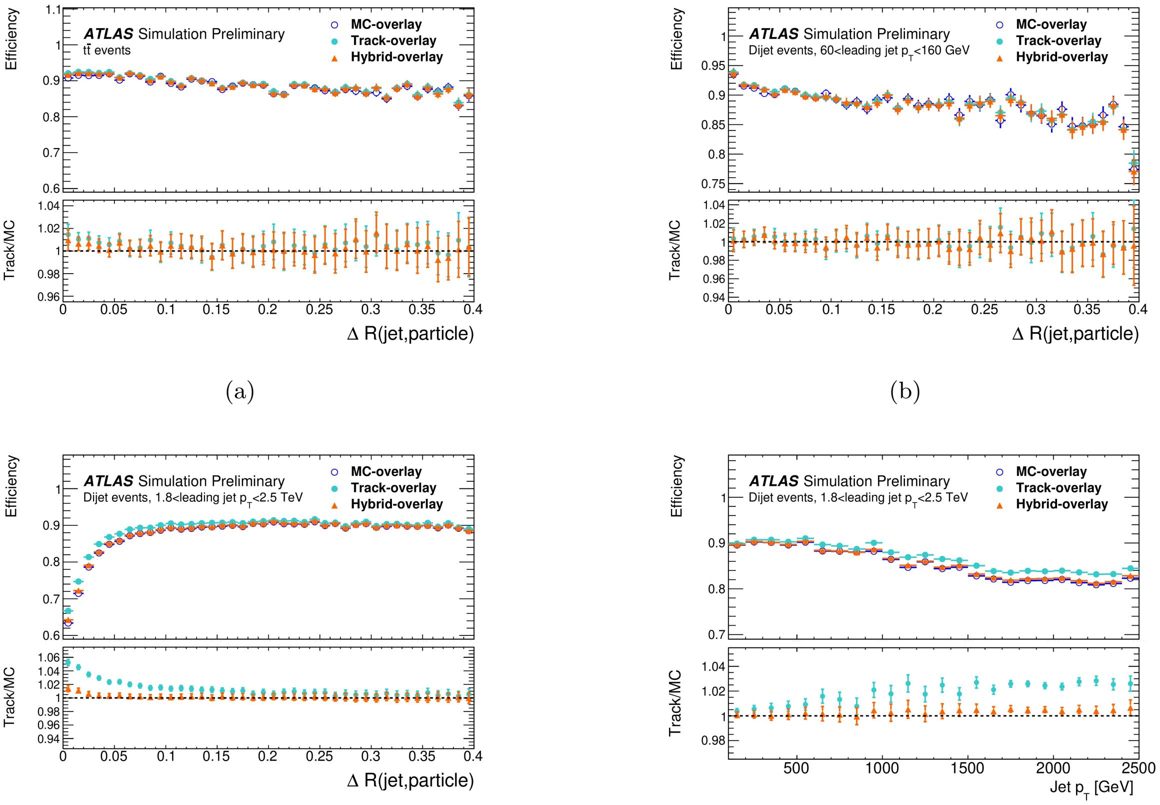
often shared or misinterpreted, complicating the pattern recognition and overall reconstruction process. Tracks with ML scores above a certain threshold are classified as unsuitable for track overlay ('bad') and are likely influenced by dense environments. The ML threshold was optimized by comparing ML-based selections to a random selection strategy, where events were randomly assigned to either track overlay or MC overlay while keeping the same proportion of events in each category. To evaluate performance, a fraction was calculated that measures how well the selection method improves efficiency. This fraction compares the efficiency of MC overlay and hybrid overlay relative to track overlay. Since hybrid overlay efficiency depends on the selection method, this fraction is calculated for both random and ML-based decisions. Given the same proportion of events sent back to MC overlay, a lower fraction value indicates better agreement between hybrid overlay and MC overlay, demonstrating the advantage of ML-based selection over random selection. This ensures that as many events as possible are processed by track overlay while maintaining high reconstruction accuracy, particularly in high-p_{T} environments.

The fractions used to evaluate the ML model are computed at the event level. However, in practice, within Athena [3], we cannot directly evaluate the model in this way. Instead, the ML model assigns a score to each track individually. If an event contains even a single 'bad' track, it is redirected to MC overlay. The current implementation achieves a 35.3% selection rate for high p_{T} events in multi-jets production (where the leading jet p_{T} ranges between 1.8 TeV and 2.5 TeV), with approximately 35% of events using track overlay and 65% using MC overlay. For the production of top quark pair events, the model is more effective, with 86.4% of events being processed using track overlay. The hybrid approach optimizes event reconstruction by balancing computational efficiency and accuracy, ensuring that complex events are handled appropriately.

The performance of track overlay has been validated across numerous physics processes. Notable differences were observed between pure track overlay and MC overlay in certain regions of phase space, particularly in dense environments. Despite these differences, the distributions in the hybrid overlay show good agreement across varied conditions, with minor deviations within 2%, which are well within the acceptable range, as shown in Figure 3. The ML model ensures that events with higher HS reconstruction efficiency in track overlay are redirected to MC overlay, and the HS reconstruction efficiency is matched between the workflows. However, there is an indication that the model's performance could benefit from being trained on a wider jet p_{T} sample. Currently, the model is trained on a specific high p_{T} jet sample. Local evaluations, such as the fake rate and the number of holes, yield good results within this high-p_{T} range. However, the model's performance in the lower p_{T} regions may not be as optimal, as it has not been trained on these ranges. Expanding the training range may allow the ML model to capture more characteristics, improving the accuracy of shared hits, the number of holes, and the fake rate, particularly in lower p_{T} samples, such as $t\bar{t}$ events. Although higher-level observables like m($\gamma\gamma$) have been observed to show good agreement between hybrid overlay and MC overlay, they have not yet been fully quantified. Further observables and detailed studies are needed in the future.

3 CPU benchmarking

The hybrid overlay method aims not only to maintain reconstruction accuracy but also to optimize computational efficiency. To evaluate the performance of different overlay methods, CPU benchmarking was conducted using the HEP-SPEC06 metric [5]. The benchmarking involved measuring the CPU usage during the overlay and reconstruction steps for MC overlay, track overlay, and hybrid overlay methods. The preliminary results, summarized in Table 1, show that the hybrid overlay approach offers a significant reduction in resource use. Specifically, there is a reduction of around 44% in reconstruction time compared to the MC overlay, with no observable loss in performance due to the inclusion of the ML decision-making process. This evaluation was conducted under conditions representative of Run 3 of the LHC, with approximately 50–60 pile-up vertices per event, demonstrating the robustness and efficiency of the hybrid overlay method under realistic operational scenarios. Further analysis reveals that approximately 60% (Run 3) of the CPU time for MC overlay is dedicated to tracking [6], which means that the tracking process takes 2.91 seconds on average. Non-tracking processes, which remain consistent between MC overlay and track overlay, account for 1.95 seconds. In the track overlay configuration, tracking time is reduced to 0.77 seconds. This represents a significant reduction in tracking time, with an approximate change of 74%. In the current setup, however, processing outside of the track reconstruction implies that the MC overlay demonstrates faster performance at the overlay step compared to the track overlay. In track overlay, the need to store and manage large collections of tracks and their associated hits contributes to the slower performance. A significant factor is the heavy computation load associated with LZMA compression, which accounts for about 40% of the total CPU usage. Additionally, approximately 10% of the CPU time is spent on decompression and another 30% on compression tasks. One viable approach is to relax the compression settings, however, this adjustment would result in larger file sizes.



(a)

(b)

(c)

(d)

Figure 3: HS reconstruction efficiency for different event types and regions, comparing MC overlay, track overlay, and hybrid overlay validation. (a) Efficiency in bins of ΔR(jet, particle) for $t\bar{t}$ events. (b) Efficiency in bins of ΔR(jet, particle) for low $p_{\rm T}$ jets events. (c) Efficiency in bins of ΔR(jet, particle) for high $p_{\rm T}$ jets events. (d) Efficiency in bins of jet $p_{\rm T}$ for high $p_{\rm T}$ jets events [4].

If these are intermediate files, the increased file size might be acceptable, but a thorough assessment of the implications is necessary to ensure that the benefit outweighs the costs. Further work is needed to improve the performance of these other aspects.

Table 1: CPU benchmarking results for different overlay configurations. The configurations include MC overlay, track overlay, and various setups of the hybrid overlay method. These setups involve forcing all events to be processed using MC overlay, forcing all events to be processed using track overlay, and dynamic selection on an event-by-event basis using ML decision-making.

Configuration	**Overlay**	**Reconstruction**
MC Overlay	2.34s	4.86s
Track Overlay	3.26s	2.72s
Hybrid Overlay (All to MC Overlay)	3.33s	4.93s
Hybrid Overlay (All to Track Overlay)	3.25s	2.65s
Hybrid Overlay (ML)	3.30s	2.71s

4 Summary

ATLAS has advanced its methodology for accelerating the production of MC simulation by implementing a track overlay approach. This method, particularly effective for events with high p_{T} jets, utilizes a DNN to make real-time decisions on whether to use track overlay or MC overlay for each event. Performance evaluations of the hybrid overlay method demonstrate that it maintains high accuracy across various physics processes, with significant improvements in CPU efficiency. Benchmarking results indicate a substantial reduction in reconstruction time, achieving around 44% savings compared to the traditional MC overlay method. However, as we approach the HL-LHC, the anticipated increase in pile-up collisions is expected to diminish these CPU savings, likely reducing from the current 44% to approximately 30% of the total reconstruction CPU time. Despite this expected reduction, the hybrid overlay method represents a significant advancement in optimizing both the accuracy and efficiency of event reconstruction in the ATLAS experiment. Validation of the track overlay for official production is planned for this year, further solidifying its role in the experiment's future.

References

[1] W. Leight et al., Faster simulated track reconstruction in the ATLAS Fast Chain, EPJ Web of Conf. 295, 03014 (2024), https://doi.org/10.1051/epjconf/202429503014

[2] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider. JINST 3, S08003(2008)

[3] ATLAS Collaboration, Athena: Software framework for ATLAS experiment, Zenodo, 2019, https://doi.org/10.5281/zenodo.2641997

[4] ATLAS Collaboration, track overlay validation for ACAT 2024, https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2024-001/

[5] J. L. Henning, SPEC CPU2006 benchmark descriptions, SIGARCH Comput. Archit. News. 34 (2006)

[6] ATLAS collaboration, Software Performance of the ATLAS Track Reconstruction for LHC Run 3, ATL-PHYS-PUB-2021-012 (2021),https://cds.cern.ch/record/2766886