IPP Project Report

Recreation of $t\bar{t}$ Analyses Using The Coffea Framework

Performed in collaboration with the IPP Summer Fellowship, CERN Summer Student Program and UVIC ATLAS team

By

Benjamin P. Scheuer

benjamin.philip.scheuer@cern.ch

University of Victoria

Supervisor

Joseph E. Lambert

September 5, 2023

Abstract

Data generation and processing is a major project that every analysis group at ATLAS must overcome. To help set a standard, the ATLAS software tutorial is a mandatory on-boarding task for any new analysis member. The tutorial teaches how to interact with LXPLUS, Athena, the Grid and various other in-house tools. The aim of this project is to improve upon the suite of analysis tools presented to ATLAS contributors by the tutorial. Specifically, has been made to allow for the processing of ATLAS data into reconstructed and filtered events, then into histograms relevant to analysis.

Contents

1. Introduction

This project was completed in conjunction with the 2023 Canadian IPP Summer Student Fellowship and the CERN Summer Student Program. The work was completed under the guidance of the UVIC ATLAS team. During my stay, two major tasks were completed: following the analysis tutorial[1] for the purpose of learning and auditing, and replicating a published ATLAS $t\bar{t}$ analysis [2] using the coffea framework [3] within python. With the constant development of tools in data analysis, it is important to not only keep active methods up to date, but also investigate new methods. This is the purpose of this project.

1.1 The ATLAS Software Tutorial

The tutorial is designed to be completed over the course of a week long lecture series, where the tutorial itself is meant to serve as both reference and practice material. It introduces the essential pieces of software that would allow an ATLAS contributor to generate and handle data on a large scale. Following along the tutorial was largely meant for the training of myself and any results of this portion of the project were primarily clerical, so a simple summary of the software discussed in the tutorial will be the only account of this work.

1.2 Developing an Analysis Using Coffea

Coffea is a modern python framework designed for large scale data analysis. Starting with ROOT files, Coffea can be used to store, filter and process data (into histograms or other finished products) very simply and efficiently. Using this framework, a template script was built that can easily be modified to fit the specific needs of any analysis. To verify the efficacy of the data processing script, it was then modelled after a published $t\bar{t}$ analysis and the output was compared to that of the paper.

2. Analysis Tools

2.1 LXPLUS

LXPLUS (Linux Public Login User Service) is a computing service made accessible to all users at CERN. Its primary purpose is to provide a common workspace and cluster computing capabilities to the analysis staff at CERN. The process of setting up current software releases has been streamlined for each team. This allows a user to easily establish the desired versions of each software, so that their work is compatible with their colleagues'. This is an essential feature when working in a large collaboration. Moreover, to satisfy the demands of large scale data analysis, LXPLUS offers access to high capacity cloud computing via the Grid system. Various tools have been implemented to improve quality of life, such as BigPanDA[4] which tracks jobs globally and gives users access to job information.

2.2 ATLAS Software

To understand and predict the various phenomenon occurring in the ATLAS detector, Monte Carlo simulations and event reconstructions are necessary. The primary tool for event generation is known as AthGeneration. It is a tool which can be run on a simple configuration file, which defines the desired parameters for the simulated events. These choices are then sent to the MadGraph software, which computes the cross sections of particular events given the chosen characteristics. These statistics are then used to simulate real events in the ATLAS detector using a Monte Carlo approach. The resulting output is a data file that can be used as a proof of concept for proposed projects. Once a data file has been obtained, real or Monte Carlo, to make use of its contents, the physics objects must be reconstructed. This includes things such as leptons, jets and missing transverse energy. Although there is more than one way to accomplish this, the best tool available for most applications is Athena. After defining a configuration file, an Athena job will be ran and the resulting data will be packaged into a convenient ROOT file, ready for analysis.

2.3 Ntuple Analysis

Top-level analysis focuses around a few specific tasks. The first of which is data loading and comprehension. If you are working directly with ROOT this process has been streamlined. However, in any other application, the data needs to be read in and processed into an accessible form. This can be done in a number of ways, but to maximize legibility, the tutorial uses python notebooks and the interactive platform SWAN. SWAN works directly from the /eos directory in LXPLUS, so it is well integrated into the general workflow. Once the data is accessed, it is necessary to filter events according to the interests of the study. This is usually a set of restrictions on the physics objects discussed earlier. Their reconstructions will produce kinematic variables such as transverse momentum and charge, which can be used to distinguish between events. If the analysis is done using python, these variables can naturally be represented within Awkward arrays[5]. Data in this form is compatible with the Boolean masking that is native to python, which can be used to filter events. Lastly, the data needs to be filled into histograms. The primary challenge represented here is memory allocation. When using sufficiently large data files, not all of the data can be loaded in at one time, so it must be processed in chunks and the results must be accumulated. To deal with this, there exist frameworks such as Coffea. These frameworks take large tasks and reduce them into pieces that will not overdraft the computing resources. The resulting output can then be passed to any plotting software, which will generate completed histograms.

3. Coffea Analysis Template

During the course of the tutorial, the user is introduced to the use of the Coffea framework to preform a simple analysis. Although it is a good tool to illustrate the use of Coffea, it lacks certain qualities that would make it useful on a broader scale. To complement this, two alternative scripts have been created, one to verbosely and precisely handle the data and another to basic bulk processing autonomously. Both

of these scripts, after some further modification will be made available in the tutorial. This should allow new users a simple and immediately effective means of analysis as they are gaining proficiency in the ATLAS system. Although the notebook environment provides an ideal test bench, it is recommended to move away from python notebooks when performing large scale analyses. The template can be very easily moved to a standard python file, so the usefulness of the template is not limited in this sense.

3.1 Data Loading

The Coffea framework is designed with high energy physics applications in mind, so it has built-in handling of ROOT files. Once a ROOT has been passed into the framework, it will be processed into a dictionary, which allows the user to select their desired branches and only process data based on that selection. The two templates approach this selection in two ways: specific well ordered packaging(a) and pushing everything to the processor(b), seen in figure 1.

```
el_pt     = branch_forms["el_pt_NOSYS"]
el_eta    = branch_forms["el_eta"]
el_phi    = branch_forms["el_phi"]
el_charge = branch_forms["el_charge"]
el_content = {"pt": el_pt, "eta": el_eta, "phi": el_phi, "charge": el_charge}
output["Electron"] = zip_forms(el_content, "Electron", 'PtEtaPhiMLorentzVector')

mu_pt     = branch_forms["mu_pt_NOSYS"]
mu_eta    = branch_forms["mu_eta"]
mu_phi    = branch_forms["mu_phi"]
mu_charge = branch_forms["mu_charge"]
mu_content = {"pt": mu_pt, "eta": mu_eta, "phi": mu_phi, "charge": mu_charge}
output["Muon"] = zip_forms(mu_content, "Muon", 'PtEtaPhiMLorentzVector')

jet_pt     = branch_forms["jet_pt_NOSYS"]
jet_eta    = branch_forms["jet_eta"]
jet_phi    = branch_forms["jet_phi"]
jet_ftag_legacy_select = branch_forms["jet_ftag_legacy_select"]
jet_content = {"pt": jet_pt, "eta": jet_eta, "phi": jet_phi, "ftag": jet_ftag_legacy_select}
output["Jet"] = zip_forms(jet_content, "Jet", 'PtEtaPhiB')

met_met_NOSYS     = branch_forms["met_met_NOSYS"]
met_content = {"met_met_NOSYS":met_met_NOSYS}
output["Met"] = zip_forms(met_content, "Met", 'MetData')
```

(a) Comprehensive Selection.

```
output["Events"] = zip_forms(branch_forms, "Events", 'PtEtaPhiMLorentzVector')


return output
```

(b) Complete Loading.

Figure 1: Approaches to Data Loading in Coffea.

Naturally, the comprehensive selection will always be preferable from a computational standpoint, however if the analysis is interested in each of the branches of the input ROOT file, then the two methods become equivalent.

3.2 Event Filtering

In any given data file, not all events will be interesting for a given analysis. As such, two layers of filtering need to be performed: the filtering of events and the filtering of objects within the events. For example, a given analysis might require that for each event, there is exactly one lepton with energy greater than 27 GeV. This can be seen as two filters, one to remove low energy leptons from each event and another to remove events with too many or too few of those leptons (figure 2.a). The filters defined using Boolean masks can then be freely combined and applied to the events (figure 2.b).

```
    #1 Lepton Cut
####################


        el_lep_kinematic_mask = ( (elec.pt >= 27*GeV)  )
        mu_lep_kinematic_mask = ( (muon.pt >= 27*GeV)  )

        lep_cut = (
                        ak.num(elec.pt[el_lep_kinematic_mask], axis=1)        #one lepton of pt >= 27 GeV
                        +ak.num(muon.pt[mu_lep_kinematic_mask], axis=1)
                        == 1
                          )
```

(a) Simple Event Filter.

```
#Object masks
        jet_object_mask = (
            np.logical_not(bjet_mask) &
            dijet_kinematic_mask
                          )

        bjet_object_mask = (
            bjet_mask
            & dijet_kinematic_mask
                          )
#Event filters
        analysis_filter = (
            dilep_cut
            & met_cut
            & dijet_cut
            & bjet_cut
            )
#Filter Data
#################### #################### ####################
        elec_filtered = elec[analysis_filter]
        muon_filtered = muon[analysis_filter]
        jet_filtered = jet[jet_object_mask][analysis_filter]
        bjet_filtered = jet[bjet_object_mask][analysis_filter]
        sf_filtered = sf[analysis_filter]
```

(b) Filter Application.

Figure 2: Filtering Data using Boolean Masks.

Using the Boolean masks in this way allows for the modification, creation and deletion of filters with very little work. The use of for-loops is also avoided entirely, which are infamously slow in python. The only major difficulty with this setup is dealing with the data structure of the events after filtering. Some physics objects, such as Jets, can have multiple instances in a single event. This means that the data array will have a dimension where each entry is dedicated to an instance of the object. Unless forced by the filter, the length of this dimension will depend on the event, thus the data cannot be casually indexed or cast into a numpy array. Awkward[5] is a convenient python package for dealing with this. Awkward is utilized in the template, so that if

a user tries to access a physics object that does not exist, rather than giving an index error, the script will continue to histogram filling and ignore the event in question.

3.3 Histogram Filling

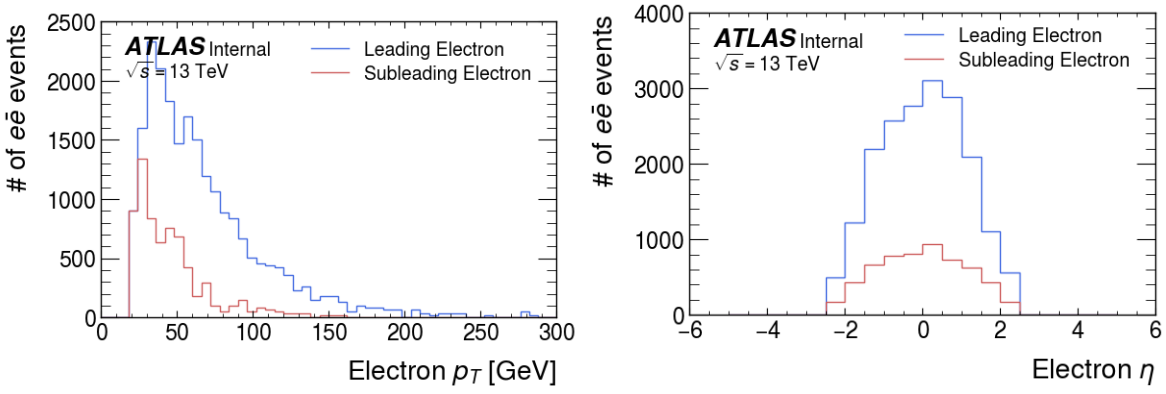
Within Coffea, there are three steps to creating histograms: axis creation, histogram definition and filling. These processes are handled by the hist python package and are trivialized to one line commands for each step. To correctly interface with Coffea, a processor class must be created. The class must have a constructor or "__init__", this is where the axis and histogram definitions must be made. There must also be accumulator and setupNPArray properties, which will be used by Coffea. Lastly, there must be a process property, this is the code that will be run for each piece of the data set. This is where the event filtering and histogram filling must go.

3.4 Plotting

The output data set is a dictionary containing the defined histograms and arrays of data, which can be filled at the same stage as histograms. The histograms are still python hist objects, which are compatible with matplotlib, so plotting is a very simple process(a) and has all the capabilities inherited from matplotlib(b)(c).

```
# Now let's plot the output of some of the arrays

fig, ax = plt.subplots(1, figsize=(6, 4), dpi=100)

out["histo_el_pt_1"].plot1d(ax=ax, color="royalblue", label="Leading Electron")

out["histo_el_pt_2"].plot1d(ax=ax, color="indianred", label="Subleading Electron")

hep.atlas.label(data=True, label="Internal")
ax.legend(bbox_to_anchor=(0.4, 1.03), loc="upper left")



ax.set_xlabel('Electron $p_T$ [GeV]')
ax.set_ylabel('# of $e\\bar{e}$ events')
plt.show()
```

(a)



(b) (c)

Figure 3: Example Plotting

4. Recreating a $t\bar{t}$ Analysis

The best way to determine if the template framework is working as intended and has all the necessary features is to recreate a published analysis. The analysis of choice[2] is a paper produced by the ATLAS top group, whom measured the cross-sections within the lepton+jets channel. The paper contains extensive information regarding the data parameters, filter setting and choices of reconstruction tools. The analysis will have been considered successfully reproduced if using the same settings, the same pass fractions are found for the filters and the histogram distributions match within a tolerance given by statistical deviation.

4.1 Analysis and Reconstruction Settings

There are three categories of settings that make up an analyses' configuration. These arise in the different stages of reconstruction and filtering. A detector-level object is a reconstruction based on the energy deposits on the EM calorimeter. The settings relating to these objects correspond to choice of algorithms and restrictions on the kinematic properties of the reconstructed particles. The particle-level objects are defined within simulated events and are constrained to only long lived particles (mean lifetime > 30 ps). The settings are similarly a choice of algorithm and kinematic constraints, with additional requirements for the number and type of particles. Lastly is the event-level selection. Selection is based on a set of six filters. Each filter can be characterized by two qualities, which lepton is present, electron or muon, and how many jets are present, of which, how many have been identified as originating from b decays. The events passed by these filters are mutually exclusive and are referred to as channels. Tables of selection requirements have been included in appendix A.

To perform the recreated analysis, a template had to be chosen. In this case, the more simple of the two template files has been selected. This reduces the influence of the new software as it is being implemented. The work flow for processing the data mimics that of a real analysis: raw detector(or simulated) data is processed into n-tuples. At this stage is when the particle and detector level selections take place. From there, the n-tuples are filtered based on the event level selection. Then, the remaining events are processed by Coffea into histograms and compact data, which can be used for physics.

4.2 Yields and Distributions

A yield represents the number of events that pass event-level selection. When compared with total events, this is a measure of the frequency that events pass selection. For scientific purposes, this is useful for measurements. However, in our analysis, it is also a measure of how closely our selection parameters match the known analysis. It is important to note that reproducing exactly the same yield is not possible, due to statistical fluctuation and software changes. However, physics demands that a very similar result should be possible. The target yields are shown in appendix B, the yields of interest pertain to the $t\bar{t}$ row of figure (a) and (b).

The distribution of kinematic variables is a reflection of both selection parameters and physics. As such, it is quite important our findings here are closely linked to the known analysis. The target analysis has concerns itself with a number of distributions, from them Fourth Jet p_T and Lepton p_T have been chosen for comparison. This is only as a means of preliminary results, all of the distributions will be considered before the project is concluded. The distributions mentioned before are shown in figure (a) of appendix C.

5. Results/Discussion

The project is currently a work in-progress. Accordingly, all stated outcomes are a reflection of preliminary results, which will be used to modify and improve the project at a future time.

5.1 Yields

The observed fractional yields are presented within figure (c) in appendix B. A correction factor for luminosity has been incorporated into the scaling of events, so we should expect the yields to match within the stated uncertainty. After some direct comparison, this is not the case, so the results are inconsistent. Notably, the number of predicted events in most channels is 50% to 100% greater than the expectation. The likely causes of such an error are the scaling and selection methods. Once this issue has been addressed, a closer observation into the specific yields can be conducted.

5.2 Distributions

The observed distributions are presented withing figure (b) in appendix C. The comparison of the two energy plots is promising, however it is still too early to say whether or not they match properly. This is because the plots in figure (b) are only run over a small sample and are still subject to significant statistical deviation. This effect can be seen in the irregularities of the tails in the distributions. Moving forward, these distributions will be useful when addressing the issues observed in the yields.

6. Conclusion

This project has been an excellent opportunity for me to exercise the tools used by the ATLAS collaboration. With some additional work, the Coffea framework will be shown to be effectual and the project will be ready to be implemented into the ATLAS software tutorial. From there, it will be an additional tool available to new or experienced contributors that wish to utilize Coffea for data analysis. I would like to issue an acknowledgment to the IPP for making this project possible and to Joseph Lambert for teaching and aiding me throughout this entire process; I really enjoyed my time at CERN, so thank you.

A. Appendix: Analysis Settings

Detector Level Selection Requirements

Electrons	$\|\eta$clust$\|$ < 2.47. Not 1.37 < $\|\eta$clust$\|$ < 1.52
Muons	$\|\eta\|$ < 2.5
Jets	anti-kt jet algorithm R = 0.4 implemented in FastJet Cuts applied for jets with pT below 120 GeV
bJets	MV2c10 tagging average efficiency of 60% for b-quark-initiated jets in ttbar identified as b-jets via ghost matching to weakly decaying b-hadrons

Particle Level Selection Requirements

Electrons	cone of size ΔR = 0.1 pT > 25 GeV and $\|\eta\|$ < 2.5
Muons	cone of size ΔR = 0.1 pT > 25 GeV and $\|\eta\|$ < 2.5
Jets	anti-kt algorithm radius parameter of R = 0.4 pT > 25 GeV and $\|\eta\|$ < 2.5
Event selection	nElectron + nMuon == 1 nJet >= 4 nbJet == 1 | nbJet == 2

Event Level Selection Requirements

Lepton pT cut	pT > 25 GeV for 2015 pT > 27 GeV for 2016 pT > 28 GeV for 2017 / 2018
Electron+jet channel	E miss T > 30 GeV and mT(W) > 30 GeV
Muon+jets channel	E miss T + mT(W) > 60 GeV
SR1	≥ 4 jets with 1 b-tagged jet
SR2	4 jets with 2 bjets
SR3	>=5 jets with 2 bjets

Figure 4: $t\bar{t}$ Analysis Recreation Settings

B. Appendix: $t\bar{t}$ Yields

	e+jets Yields - Event Selection Cuts		
	SR1	SR2	SR3
$t\bar{t}$	1684661.9 ± 84233.1	461020.4 ± 23051.0	457329.1 ± 22866.5
Single Top	117625.6 ± 17643.8	23993.9 ± 3599.1	17317.2 ± 2597.6
W+jets	151598.7 ± 72767.4	10226.9 ± 4908.9	7533.1 ± 3615.9
Z+jets	36180.3 ± 17366.5	3180.3 ± 1526.6	2408.2 ± 1155.9
Diboson	6813.6 ± 3270.5	587.5 ± 282.0	558.2 ± 267.9
$t\bar{t}V$	5867.4 ± 586.7	731.9 ± 73.2	2265.9 ± 226.6
$t\bar{t}H$	1640.3 ± 164.0	280.6 ± 28.1	1104.7 ± 110.5
Multijet	143858.3 ± 71929.2	16889.9 ± 8445.0	14270.9 ± 7135.5
Total pred.	2148246.2 ± 134863.2	516911.6 ± 25340.4	502787.3 ± 24394.3
Data	2104763	508387	510606

(a) Target Yields for Electron+Jet Channels

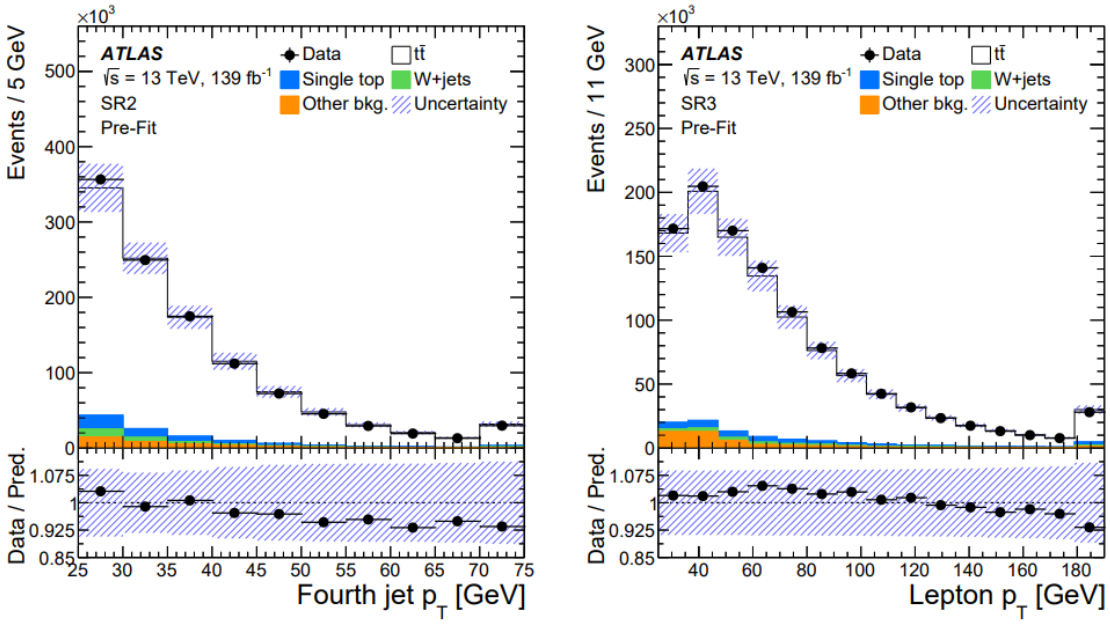
	μ+jets Yields - Event Selection Cuts		
	SR1	SR2	SR3
$t\bar{t}$	1940391.4 ± 97019.6	531132.4 ± 26556.6	526181.4 ± 26309.1
Single Top	137070.1 ± 20560.5	27943.7 ± 4191.5	19811.2 ± 2971.7
W+jets	198064.9 ± 95071.2	13547.1 ± 6502.6	9898.7 ± 4751.4
Z+jets	32679.4 ± 15686.1	3113.3 ± 1494.4	2043.7 ± 981.0
Diboson	8089.6 ± 3883.0	722.4 ± 346.8	649.3 ± 311.7
$t\bar{t}V$	6335.7 ± 633.6	798.1 ± 79.8	2483.0 ± 248.3
$t\bar{t}H$	1733.5 ± 173.3	291.5 ± 29.1	1179.1 ± 117.9
Multijet	69492.4 ± 34746.2	10245.3 ± 5122.6	7958.1 ± 3979.0
Total pred.	2393857.0 ± 142628.6	587793.6 ± 28172.8	570204.4 ± 27212.9
Data	2436093	592165	592644

(b) Target Yields for Muon+Jet Channels

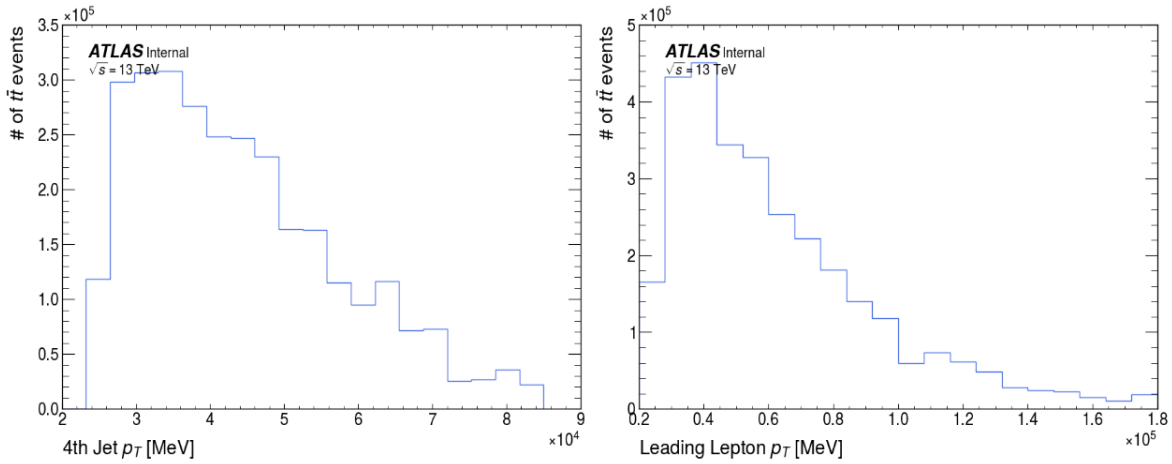
SR1_EJchannel	3076793.58 +/- 69740.54 events
SR2_EJchannel	418197.31 +/- 25802.80 events
SR3_EJchannel	1116411.68 +/- 42319.21 events
SR1_MJchannel	3076634.72 +/- 70286.75 events
SR2_MJchannel	868672.89 +/- 37237.90 events
SR3_MJchannel	875194.85 +/- 37604.66 events

(c) Produced Yields for All Channels

C. Appendix: $t\bar{t}$ Distributions



(a) Target Distributions



(b) In-Progress Produced Distributions

D. Appendix: References

[1]ATLAS Collaboration . (n.d.). *Analysis software tutorial.* ATLAS Software Documentation. https://atlassoftwaredocs.web.cern.ch/ABtutorial/

[2] ATLAS Collaboration . (2020). *Measurement of the $t\bar{t}$ production cross-section in the lepton+jet channel at $\sqrt{s} = 13$ TeV with the ATLAS experiment.* Phys Lett. B. 10.1016/j.physletb.2020.135797

[3]*Columnar Object Framework for effective analysis.* coffea. (n.d.). https://coffeateam.github.io/coffea/

[4]ATLAS Collaboration. (n.d.). *ATLAS PanDA.* https://bigpanda.cern.ch/

[5]User guide — Awkward Array 2.3.3 documentation. (n.d.). Awkward-Array.org. from https://awkward-array.org/doc/main/user-guide/index.html