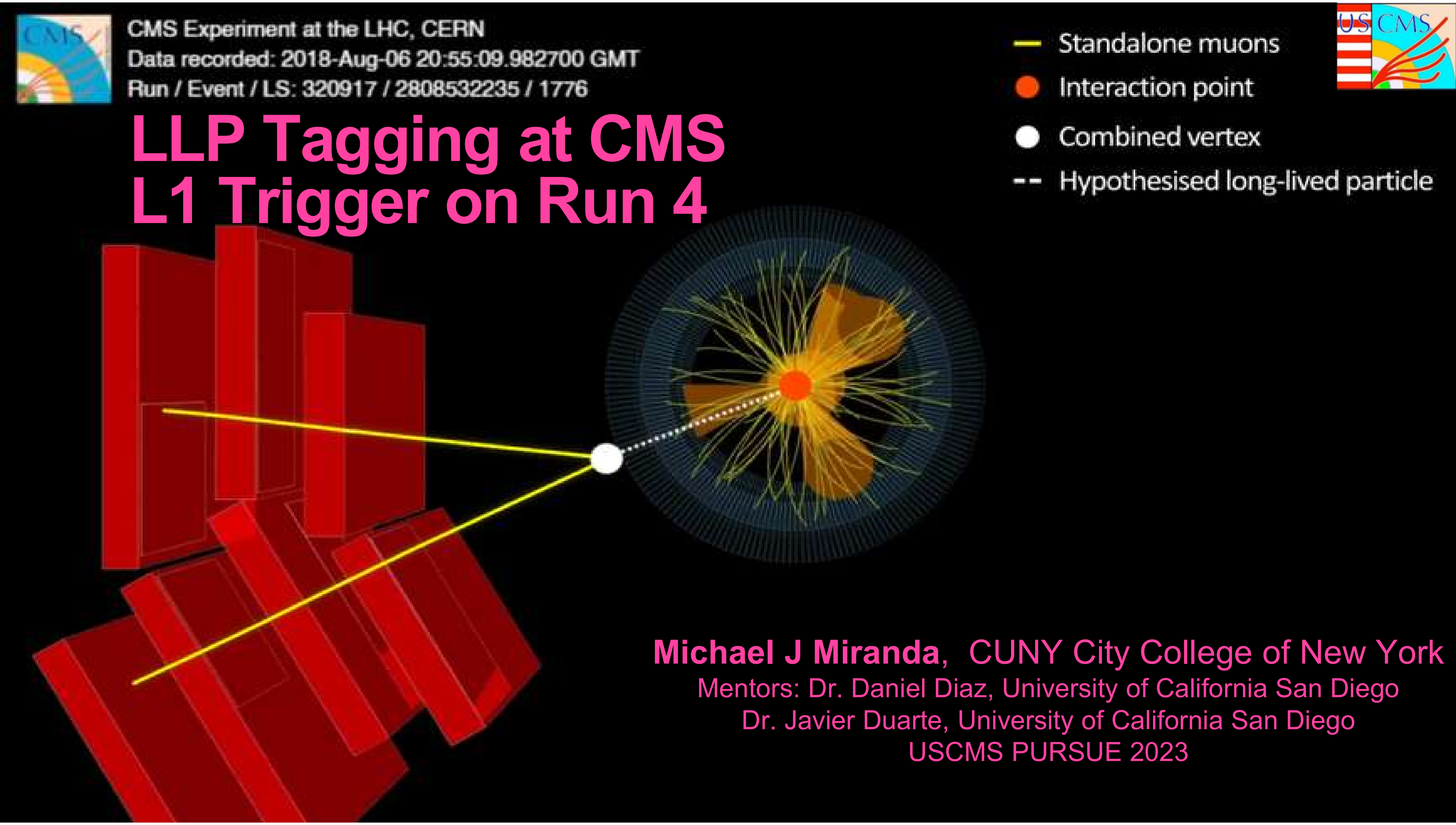


LLP Tagging at CMS L1 Trigger on Run 4

Michael J Miranda, CUNY City College of New York
Mentors: Dr. Daniel Diaz, University of California San Diego
Dr. Javier Duarte, University of California San Diego
USCMS PURSUE 2023

Abstract

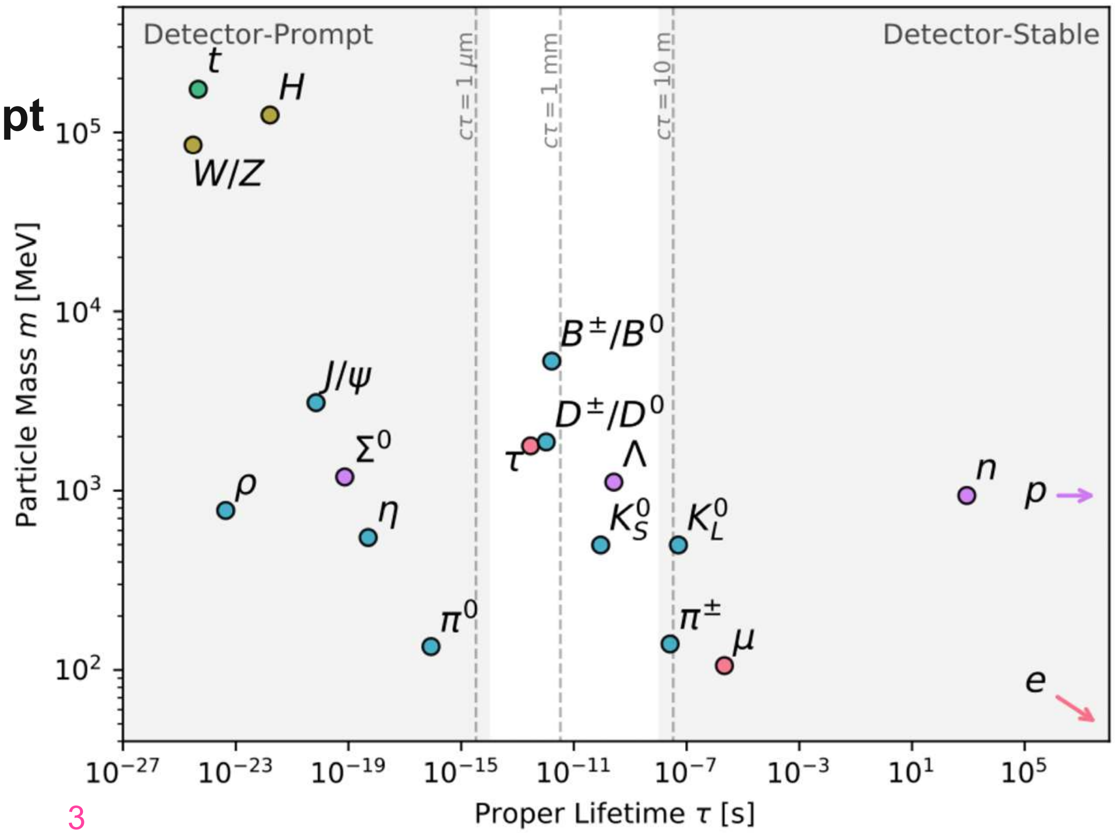
Abstract: Trigger efficiency and rate studies are presented. The selection of events containing long-lived particles (LLPs) at the compact muon solenoid (CMS) detector has been historically difficult due to a lack of LLP aware triggers. To counter this issue, we are developing a machine learning based trigger to select events containing jets produced by LLPs, to be used in the CMS Level-1 trigger (L1). LLPs appear in a number of extensions to the standard model (SM), so-called beyond the standard model (BSM) theories and yet they have been less studied at the large hadron collider due to the inability to explicitly select the events containing these particles. Trigger performances can be evaluated by expected signal efficiency and event rate calculations using simulated events generated by Monte Carlo which are then processed through an L1 emulator. For signal we are using a BSM sample of long-lived stop squarks decaying to bottom quarks, while we use SM QCD as our background. Jets were constructed using a seeded clustering algorithm based on the definition of a cone with radius R = 0.4, where R is defined as the difference in pseudorapidity (φ) and azimuthal angle (η). The quality of these clustered jets were assessed with plots of the transverse momentum (pT), η , φ , and mass of the jets in terms of all jets, the jet with the largest value of pT; described here as the lead jet, and the jet with the second largest value of pT; described as the sublead jet. Based on a recent technical design report discussing the phase-2 upgrade of the CMS L1 trigger, requirements for hadronic seeds using PUPPI jets as well as missing transverse energy seeds were defined. In addition to the calculations of signal efficiency and event rate for these triggers, an efficiency curve for a Single PuppiJet trigger graphed against generator-level LLP pT is created.

What is a Long-Lived Particle (LLP)?

- Most particles produced at CMS are called **prompt** in that they **decay immediately** and are **not directly observed** by the detectors

- An LLP is simply a particle that is **not prompt**
  - That is, they live long enough to travel a measurable distance within the detector

- And while the LLPs shown here are all standard model particles, the objects of interest in this project are **beyond the standard model LLPs**



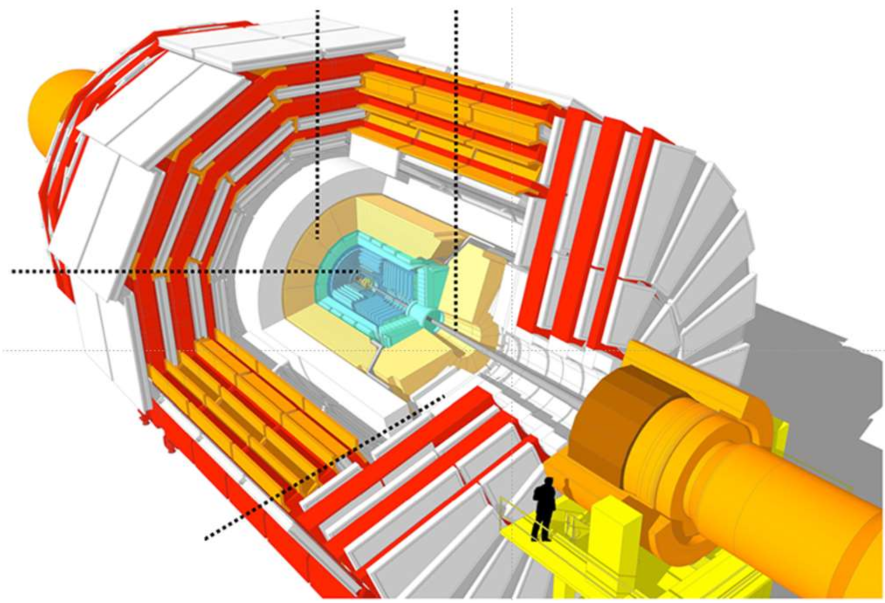
What is Needed to Define a Trigger?

- LLP Jet Trigger
  - What is a **tagger**?
    - An algorithm whose output indicates the **probability** of identifying particles of interest
  - What is a **trigger**?
    - A more complex algorithm defined by **a set of requirements at object level** for p_T, $|\eta|$, etc. on each event whose output contributes to whether or not the **event is saved**
  - What is needed to define this LLP Jet Trigger?
    - **ML-LLP Jet Tagger** → machine learning algorithm to tag these LLP jets
    - A **trigger definition** that incorporates this tagger and a set of requirements for LLP jet

- My part in this larger project
  - Developing the machinery to **comparing trigger performances** via:
    - **Signal efficiency** using simulated signal samples
    - **Event rate** using simulated background samples

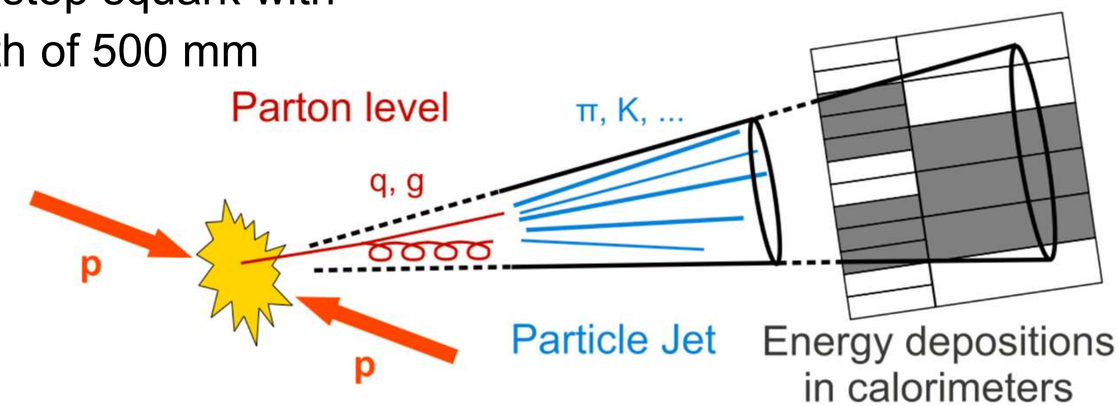
What We'll Cover…

- LLP Jet Trigger and Trigger Comparisons
- Ntuples and Jet Construction
- Jet Feature Plots
- L1 Trigger
- Trigger Seeds
- Signal Efficiencies
- Trigger Efficiency Turn-On Curve
- JetConstructTriggerAnalysis.py
- Sharing the Code



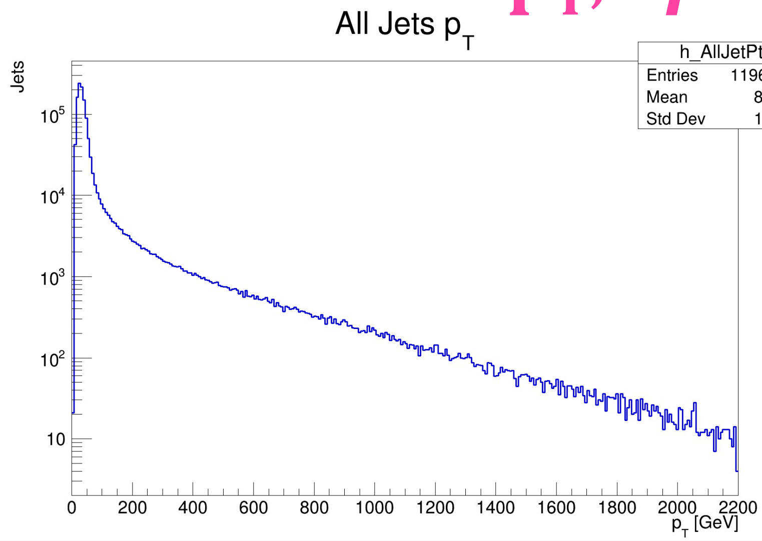
Ntuples & Jet Construction

- Events simulated by Monte Carlo, processed via L1 Emulator, stored as **ntuples**
  - pfTuple_DisplacedSUSY_stopToBottom_M_800_500mm.root **(50,000 events)**
    - LLP **Signal** sample containing stop squark with mass of 800 GeV and decay length of 500 mm
  - pfTuple_QCD.root **(99,676 events)**
    - **Background** sample



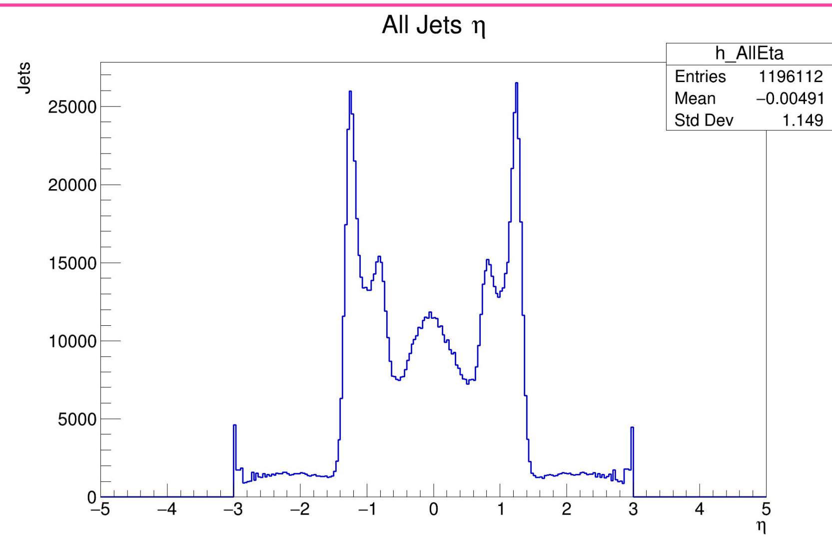
- Seeded Jet Clustering Algorithm
  1. Identify highest available p_T PF or PUPPI particle as **seed**
  2. Sum over the **TLorentzVectors** of all particles defined in a cone within $\sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} = \Delta R < 0.4$
  3. Remove jet constituent particles from further event processing
  4. Repeat these for a maximum of 12 jets per event in the ntuple

Jet Feature Plots – p_T, η

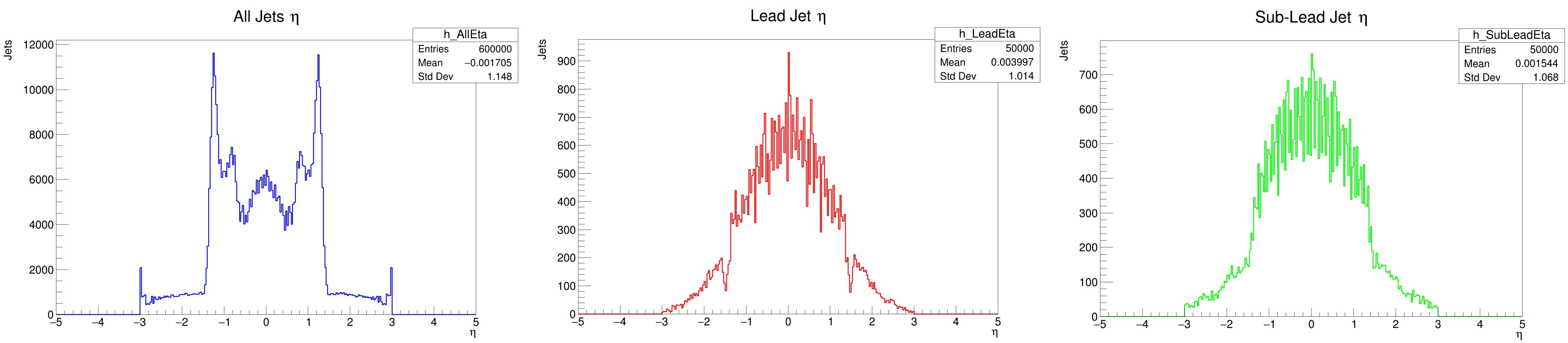


- Jets constructed from **background PF** particles with a **2 p_T cut** on jet constituents
- Sharp peak of low p_T jets

- Jets constructed from **background PF** particles with **no cut** on jet constituents
- Characteristic **“jet horns”** observed
    - Large number of **low-quality jets** in data for a certain $|\eta|$ range



Jet η Plots – Full, Lead, Sublead Jets



- Jets constructed from **signal PF** particles with **no cut** on jet constituents
- **Lead** Jets → first jet of each event; **Sub-Lead** Jets → second jet of each event
- Effect is observably **less prominent** in the leading jet and more so in the sub-leading jet

L1 Trigger – CMS Trigger System and Data Flow

- For each collision at CMS (1 per 25 nanoseconds), the detector processes 40 million events per second
  - As each event is ~1 MegaByte, data on the **order of TeraBytes** is produced **each second**
    - It is physically impossible to reconstruct and store all 40 million events per second
- To address this, a system of triggers is employed to select **only the most "interesting"** events

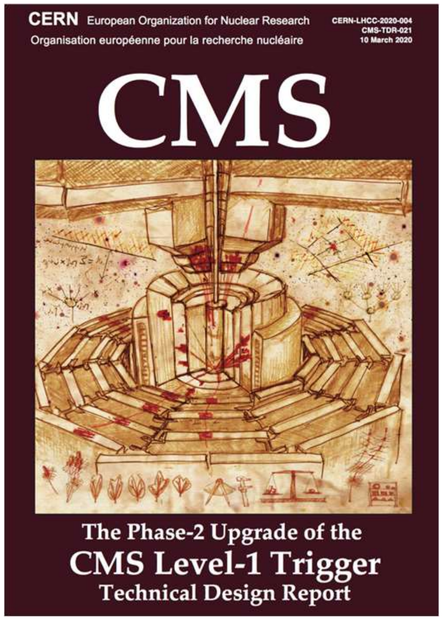
- For the purposes of my project, the most important number is: **100 kHz**
  - That is, the **maximum event output rate** passed by L1
    - This means the sum of each L1 seed rate in this menu must output **less than** 100 kHz

L1 Trigger Seeds and their Requirements

- In general, the ideal trigger has a low event rate and a high signal efficiency
  - **Lower** Event Rate → **Less** Space Used in L1 Trigger Menu
  - **Higher** Signal Efficiency → **Higher** Probability of Correct Event Classification

L1 Trigger seeds	Offline Threshold(s) at 90% or 95% (50%) [GeV]	Rate $\langle PU \rangle = 200$ [kHz]	Additional Requirement(s) [cm, GeV]	Objects plateau efficiency [%]
Hadronic seeds (jets,H_T)				
Single PuppiJet	180	70	$\lvert\eta\rvert < 2.4$	100
Double PuppiJet	112,112	71	$\lvert\eta\rvert < 2.4, \Delta\eta < 1.6$	100
PuppiH_T	450(377)	11	jets: $\lvert\eta\rvert < 2.4, p_T > 30$	100
QuadPuppiJets-PuppiH_T	70,55,40,40,400(328)	9	jets: $\lvert\eta\rvert < 2.4, p_T > 30$	100,100
E_T^{miss} seeds				
PuppiE_T^{miss}	200(128)	18		100
Cross Lepton seeds				
TkMuon-TkIsoElectron	7,20	1	$\lvert\eta\rvert < 2.4, \Delta z < 1$	95, 93
TkMuon-TkElectron	7,23	3	$\lvert\eta\rvert < 2.4, \Delta z < 1$	95, 93
TkElectron-TkMuon	10,20	1	$\lvert\eta\rvert < 2.4, \Delta z < 1$	93, 95
TkMuon-DoubleTkElectron	6,17,17	0.1	$\lvert\eta\rvert < 2.4, \Delta z < 1$	95, 93
DoubleTkMuon-TkElectron	5,5,9	4	$\lvert\eta\rvert < 2.4, \Delta z < 1$	95, 93
PuppiTau-TkMuon	36(27),18	2	$\lvert\eta\rvert < 2.1, \Delta z < 1$	90, 95
TkIsoElectron-PuppiTau	22,39(29)	13	$\lvert\eta\rvert < 2.1, \Delta z < 1$ $\Delta R > 0.3$	93, 90

CMS-TDR-021

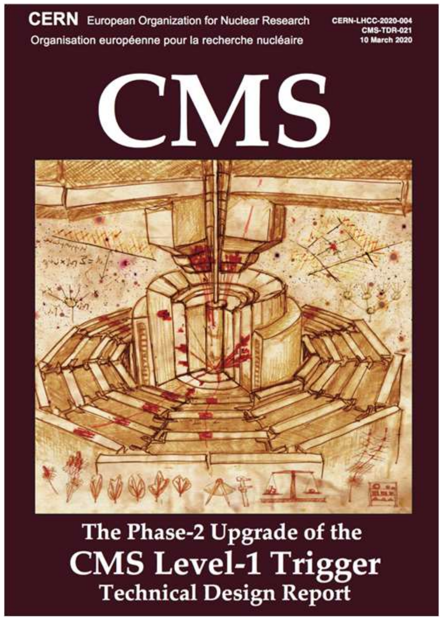


L1 Trigger Seeds and their Requirements

- In general, the ideal trigger has a low event rate and a high signal efficiency
  - **Lower** Event Rate → **Less** Space Used in L1 Trigger Menu
  - **Higher** Signal Efficiency → **Higher** Probability of Correct Event Classification

L1 Trigger seeds	Offline Threshold(s) at 90% or 95% (50%) [GeV]	Rate $\langle PU \rangle = 200$ [kHz]	Additional Requirement(s) [cm, GeV]	Objects plateau efficiency [%]
Hadronic seeds (jets,H_T)				
Single PuppiJet	180	70	$\|\eta\| < 2.4$	100
Double PuppiJet	112,112	71	$\|\eta\| < 2.4, \Delta\eta < 1.6$	100
PuppiH_T	450(377)	11	jets: $\|\eta\| < 2.4, p_T > 30$	100
QuadPuppiJets-PuppiH_T	70,55,40,40,400(328)	9	jets: $\|\eta\| < 2.4, p_T > 30$	100,100
E_T^{miss} seeds				
PuppiE_T^{miss}	200(128)	18		100
Cross Lepton seeds				
TkMuon-TkIsoElectron	7,20	1	$\|\eta\| < 2.4, \Delta z < 1$	95, 93
TkMuon-TkElectron	7,23	3	$\|\eta\| < 2.4, \Delta z < 1$	95, 93
TkElectron-TkMuon	10,20	1	$\|\eta\| < 2.4, \Delta z < 1$	93, 95
TkMuon-DoubleTkElectron	6,17,17	0.1	$\|\eta\| < 2.4, \Delta z < 1$	95, 93
DoubleTkMuon-TkElectron	5,5,9	4	$\|\eta\| < 2.4, \Delta z < 1$	95, 93
PuppiTau-TkMuon	36(27),18	2	$\|\eta\| < 2.1, \Delta z < 1$	90, 95
TkIsoElectron-PuppiTau	22,39(29)	13	$\|\eta\| < 2.1, \Delta z < 1$ $\Delta R > 0.3$	93, 90

CMS-TDR-021



How to Count the Number of Triggered Events

- Consider the Single PuppiJet and Double PuppiJet seeds with the following trigger requirements
  - Single: **at least one** jet with $\mathrm{p_T} > 180\ \mathrm{GeV}$ and $|\eta| < 2.4$
  - Double: **at least two** jets each with $\mathrm{p_T} > 112\ \mathrm{GeV}$ and $|\eta| < 2.4$ as well as a $\Delta\eta < 1.6$

- If we had an event containing 2 jets with the following values:
  - Jet 1: $\mathrm{p_T} = 524.066\ \mathrm{GeV}$, $\eta = -0.492$
  - Jet 2: $\mathrm{p_T} = 59.500\ \mathrm{GeV}$, $\eta = -0.921$

- **Only Single PuppiJet** trigger would fire for this event

How to Count the Number of Triggered Events

- Consider the Single PuppiJet and Double PuppiJet seeds with the following trigger requirements
  - Single: **at least one** jet with $\mathrm{p_T} > 180\ \mathrm{GeV}$ and $|\eta| < 2.4$
  - Double: **at least two** jets each with $\mathrm{p_T} > 112\ \mathrm{GeV}$ and $|\eta| < 2.4$ as well as a $\Delta\eta < 1.6$

- If we had another event containing 3 jets with the following values:
  - Jet 1: $\mathrm{p_T} = 397.75\ \mathrm{GeV},\ \eta = 0.243$
  - Jet 2: $\mathrm{p_T} = 385.593\ \mathrm{GeV},\ \eta = -1.099$
  - Jet 3: $\mathrm{p_T} = 334.0\ \mathrm{GeV},\ \eta = -1.335$

- **Both Single and Double PuppiJet** triggers would fire for this event
  - NOTE: despite Jets 1, 2, and 3 all passing the Single PuppiJet requirements, this event is **only counted once**

Trigger Seed Signal Efficiencies

L1 Trigger Seeds	Signal Sample: Total # Events	# of Events Passing	Efficiency
Single PuppiJet	50000	49532	0.99064
Double PuppiJet + $\Delta\eta$	50000	48026	0.96052
Quad PuppiJet-HT	50000	16626	0.33252
PuppiHt	50000	13859	0.27718
PuppiMET	50000	34508	0.69016

L1 Trigger Seeds	Requirements
Single PuppiJet	pT > 180 & |η| < 2.4
Double PuppiJet + $\Delta\eta$	pT > 112 & |η| < 2.4 & Δη < 1.6
Quad PuppiJet-HT	pT > 70, 55, 40, 40 & HT > 400; jets pT > 30 & |η| < 2.4
PuppiHt	HT > 450 (Scalar sum of pT of all jets with pT > 30 and |η| < 2.4)
PuppiMET	MET > 200 (Scalar sum of pT of calorimeter deposits)

- **High** efficiency of single and double jet triggers
- **Low** efficiency of quad jet-Ht trigger
- Due to **stricter requirements** on quad jet-Ht trigger

Trigger Seed Signal Efficiencies

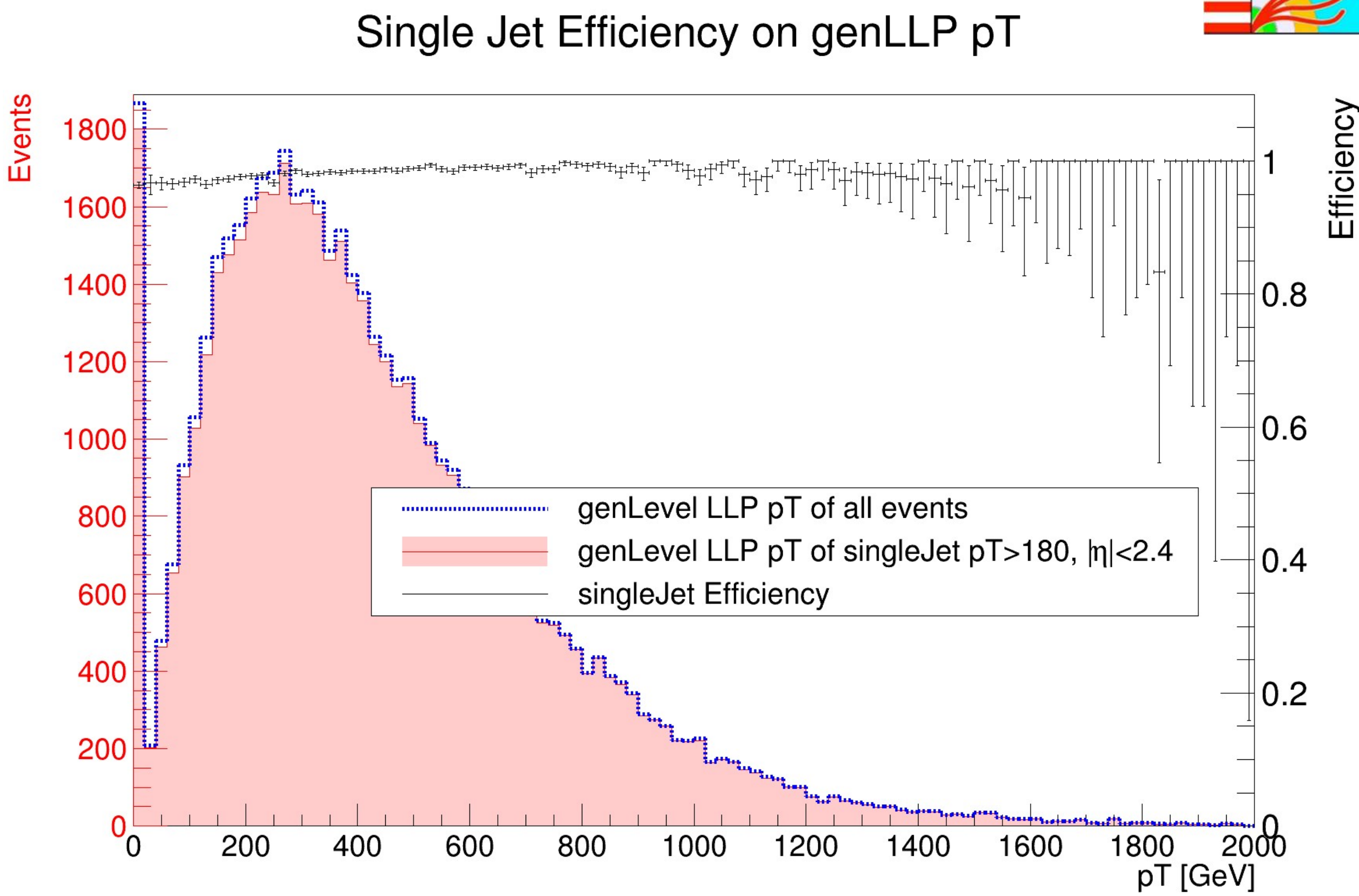
L1 Trigger Seeds	Signal Sample		
	Total # Events	# of Events Passing	Efficiency
Single PuppiJet	50000	49532	0.99064
Double PuppiJet + $\Delta\eta$	50000	48026	0.96052
Quad PuppiJet-HT	50000	16626	0.33252
PuppiHt	50000	13859	0.27718
PuppiMET	50000	34508	0.69016

L1 Trigger Seeds	Requirements
Single PuppiJet	pT > 180 & |η| < 2.4
Double PuppiJet + $\Delta\eta$	pT > 112 & |η| < 2.4 & Δη < 1.6
Quad PuppiJet-HT	pT > 70, 55, 40, 40 & HT > 400; jets pT > 30 & |η| < 2.4
PuppiHt	HT > 450 (Scalar sum of pT of all jets with pT > 30 and |η| < 2.4)
PuppiMET	MET > 200 (Scalar sum of pT of calorimeter deposits)

- **High** efficiency of single and double jet triggers
- **Low** efficiency of quad jet-Ht trigger
- Due to **stricter requirements** on quad jet-Ht trigger

Trigger Efficiency Turn-On Curve

- Using **signal PF** particles with **10 p_T cut** jets
- Efficiency of single jet trigger of PF jets as a function of **gen-level LLP p_T**
- High efficiency uncertainty at high LLP p_T
- Further ntuple samples will give more sensitive turn-on



JetConstructTriggerAnalysis.py

- What the code does:
  - Jet Construction + Jet Feature Plots + Trigger Efficiency + Trigger Rate + Efficiency Curves

- How to run the code:
  - In the LPC terminal, the code can be run from the command line:

```
python JetConstructTriggerAnalysis.py "ntuple.root" 1 0 0 0 0
```

  - **Argparse** – Command Line Interface
    - Allows user to pass different ntuple files for jet construction and analysis
    - Give user the option to enable or disable features specific to their task

- How it will be used:
  - To **compare** efficiencies and rates for trigger seeds with the LLP tagger input and the same seeds without
  - To investigate the quality of PF or PUPPI jets from a given ntuple

JetConstructTriggerAnalysis.py

- What the code does:
  - Jet Construction + Jet Feature Plots + Trigger Efficiency + Trigger Rate + Efficiency Curves

- How to run the code:
  - In the LPC terminal, the code can be run from the command line:

```
python JetConstructTriggerAnalysis.py "ntuple.root" 1 0 0 0 0
```

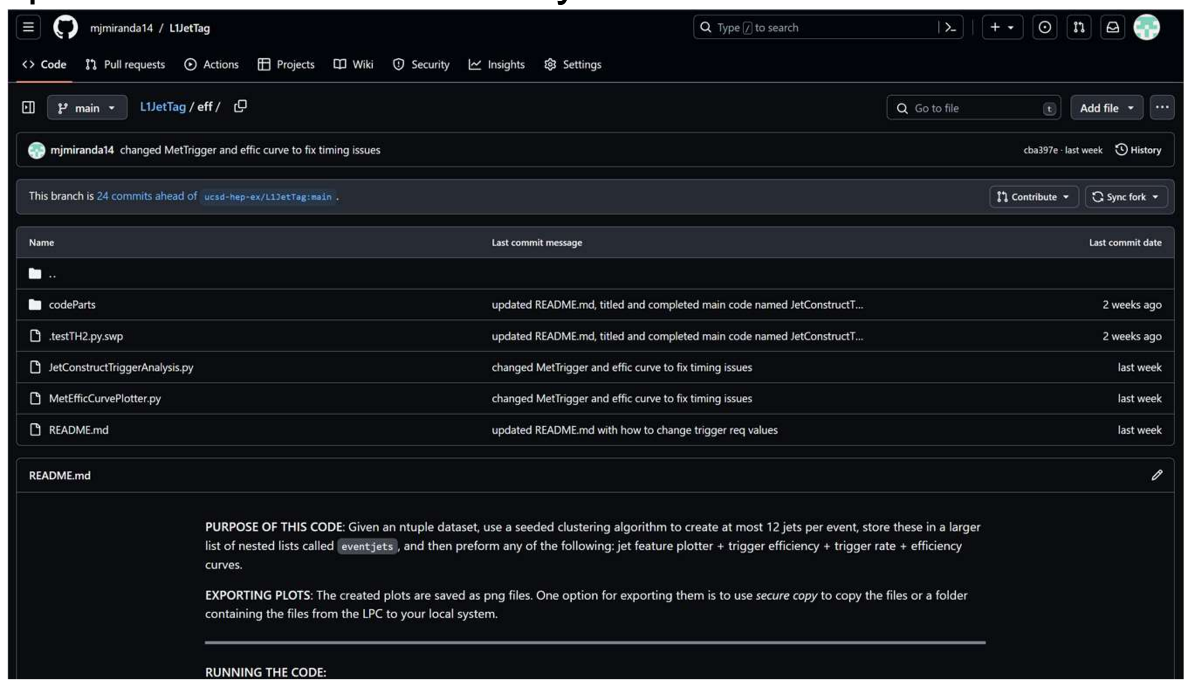
  - **Argparse** – Command Line Interface
    - Allows user to pass different ntuple files for jet construction and analysis
    - Give user the option to enable or disable features specific to their task

- How it will be used:
  - To **compare** efficiencies and rates for trigger seeds with the LLP tagger input and the same seeds without
  - To investigate the quality of PF or PUPPI jets from a given ntuple

Sharing the Code with Lab Group

- Where does your code live?
  - A branched repository of this L1 Jet Tag repository on [GitHub]()

- And how will other people in the lab group be able to use it or edit it?
  - README.md – a document for the person who's never seen your code before
    - Virtual environment
    - Updating trigger definitions
    - Editing the plots



What Did I Do? What's Next?

- Produced JetConstructTriggerAnalysis.py with an accompanying README that implements:
  - **Jet Construction**
    - Seeded Clustering Algorithm
  - **Jet Feature Plots**
    - Plotting p_T, η, ϕ, and mass to investigate quality of constructed jets
  - **Trigger Signal Efficiency and Event Rate**
    - Calculating expected signal efficiency and expected event rate for a defined trigger seed
  - **Trigger Efficiency Curve**
    - Graphing trigger efficiency over a particular parameter to determine "turn-on" behavior
  - All with the option to turn these ON/OFF depending on the task and to change the input ntuple sample all from the command line

- **What's next**
  - Defining new trigger path that factors tagger output with specific jet properties which would require:
    - Comparing trigger performance with and without LLP tagger implemented
    - Varying accepted tagger outputs with trigger parameters such as jet p_T or η

Acknowledgements

- First, I would like to thank my mentors from University of California – San Diego:
  - Prof. Javier Duarte
  - Dr. Daniel Diaz
  - Tony Aportela

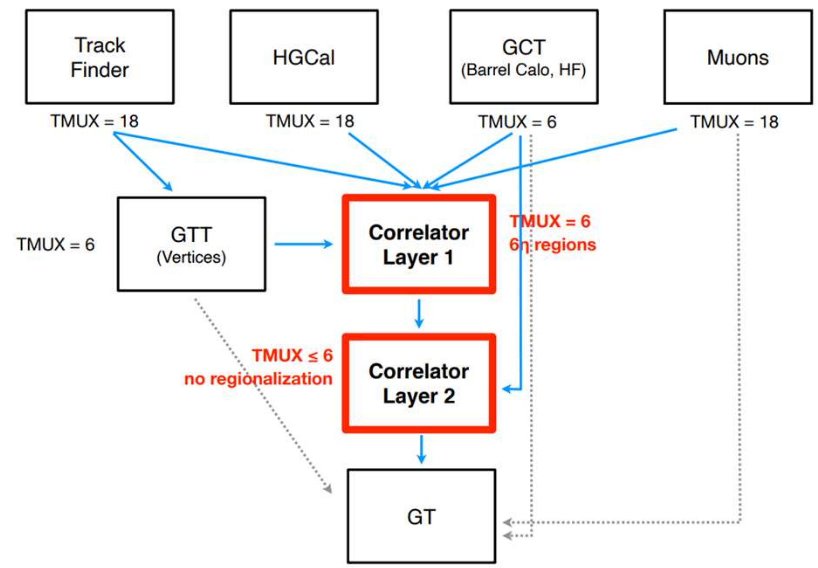
- And of course, the PURSUE program PIs, Fermilab LPC Staff, all the tutorial instructors, and all the speakers who kindly gave of their time
  - Prof. Sudhir Malik, Prof. Santanu Banerejee, Prof. Tulika Bose
  - Dr. Marguerite Tonjes
  - Maggie Slusarczyk, Guillermo Fidalgo

- Finally, I would like to thank the institutions that made this summer possible
  - Fermilab
  - US Department of Energy
  - UC San Diego
  - And every US CMS collaborating institution

Backup Slides

Phase 2 Upgrades of CMS L1 Trigger

- Upgrades will enable previously unsupported LLP tagging at L1 via:
  - Global Track Trigger (GTT)
    - Reconstruct charged particle tracks within full outer silicon tracker volume at 40 MHz collision rate to ultimately build high-level objects out of these tracks
  - Correlator Trigger (CT)
    - Aggregate inputs from all upstream systems:
      - Track Finder (TF), High Granularity Calorimeter (HGCal), Global Calorimeter Trigger (GCT), Muons, Global Track Trigger (GTT))
    - Optimally combine information from the various sub-systems to achieve the best possible trigger performance
  - FPGA arrays
    - Hardware implementation of these systems



Events with 0 or 1 jet constructed – Signal Sample

- Why are the lead and sublead jets plots missing entries from the total 50,000 event dataset?
  - Of the dataset, 15 events have only 1 jet and 1 event has 0 jets from this clustering algorithm
  - I believe this is due to candidate constituent particles having a large ΔR value (compared to a ΔR of 0.4) with the seed particles as well as low p_T candidates

			ΔR					pT				
Event #	# of Jets from Clustering	# of Particles	Mean	Median	Mode	Min	Max	Mean	Median	Mode	Min	Max
2539	1	300	2.936322779	2.946664801	2.794697522	0.228315017	5.286037765	4.282608696	3	2.25	2	18
18751	1	304	2.004010574	2.05343396	2.055537412	0.035980912	4.101377501	4.936468647	3.25	2.25	2	101
20091	1	339	2.122644076	2.16681918	1.459210356	0.004363328	3.88123425	5.874260355	3.25	2.25	2	179.5
23990	1	241	1.973987684	1.944231354	1.100604527	0.022248661	3.797928825	4.697916667	3	2.25	2	93.75
25399	1	271	1.915761538	1.917887462	0.151275868	0.034078717	3.725828391	4.525	3	2.25	2	82.75
28897	1	301	2.097197254	2.114963302	0.729810575	0.037280261	4.149013133	4.351666667	3	2.25	2	34
31964	1	270	1.915020147	1.857136108	0.278467571	0.013089895	4.03989301	4.796468401	3.25	2.25	2	104.75
32396	1	272	2.495017599	2.511531177	0.166608178	0.166608178	4.723601121	4.757380074	3.25	2.5	2	21.5
34855	1	260	2.582208516	2.769046898	3.199363455	0.356085647	4.796300468	4.346525097	3	2	2	23.25
36944	1	273	2.024366671	2.181846523	1.574966313	0.015732146	4.234703401	4.757352941	3	2.25	2	108.25
37331	1	283	2.347109666	2.37804061	0.689901923	0.162383575	4.736455077	5.548758865	3.25	2.25	2	109.75
38650	1	263	1.98487616	2.018203136	1.014216862	0.021816611	3.862497362	4.229961832	3	2.25	2	89.25
39539	1	239	1.893352802	1.966218221	2.131701147	0.008726642	3.80219981	4.545168067	3	2.25	2	74.5
46550	1	266	1.909686419	1.999535348	0.098053467	0.030853297	3.826495131	5.322641509	3.25	2.25	2	114.5
46940	1	240	1.995206766	2.00633652	1.421780659	0.006170646	3.640708861	5.076359833	3	2.25	2	110
49078	0	220	2.120648986	2.187596421	0.474961225	0.031464457	4.511368092	4.27283105	3.25	2.25	2	16.5

Efficiency/Rate Calculations – Pseudocode

Jet Triggers

- **FUNC** input: list of jets and any $p_T,\ \eta,\ \phi,\ M$ threshold values
  - **INITIALIZE** a counting variable
  - **FOR** each jet in an event loop
    - **IF** trigger requirements are met, increment the count variable
      - Add on successive **FOR** loops for each jet in an event for multi-jet triggers
      - Jets may not be re-used for the same event
  - **OUTPUT** effic = (count #) / (# of events) OR rate = (count #) / (# of events) * 40 MHz

Energy Sums Triggers

- **FUNC** input: list of jets and any $p_T,\ \eta,\ \phi,\ M$ threshold values
  - **INITIALIZE** a counting variable
  - **FOR** each jet in an event loop
    - **INTIALIZE** a variable to hold scalar or vector sum of p_T or E
    - **IF** trigger requirements are met, add p_T value on to sum variable
      - Jets may not be re-used for the same event
  - **IF** sum meets requirements, increment counting variable
  - **OUTPUT** effic = (count #) / (# of events) OR rate = (count #) / (# of events) * 40 MHz