BETHEL
UNIVERSITY

Event Selection

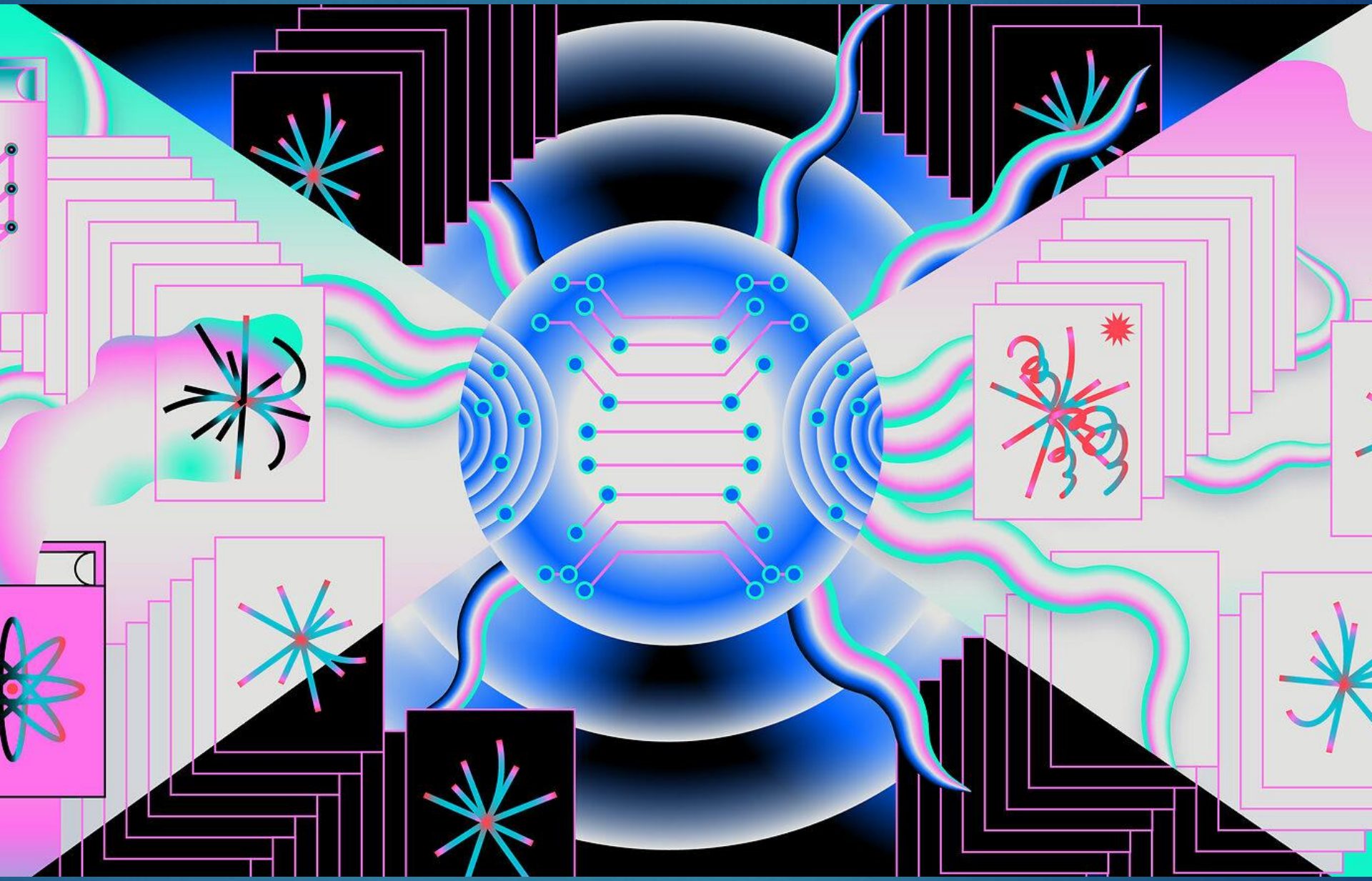
OPEN DATA WORKSHOP 2023

JULIE HOGAN

7/11/2023

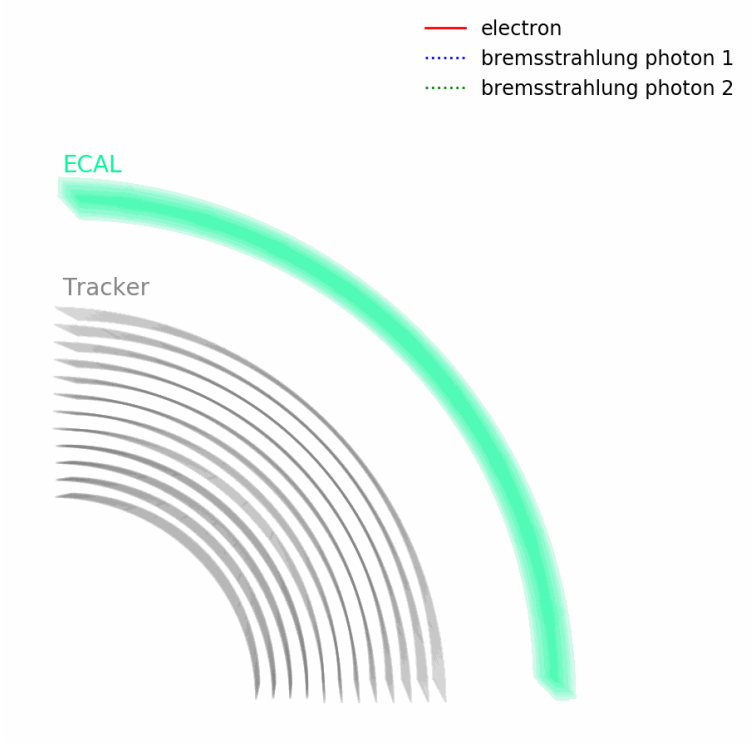
Symmetry on triggers

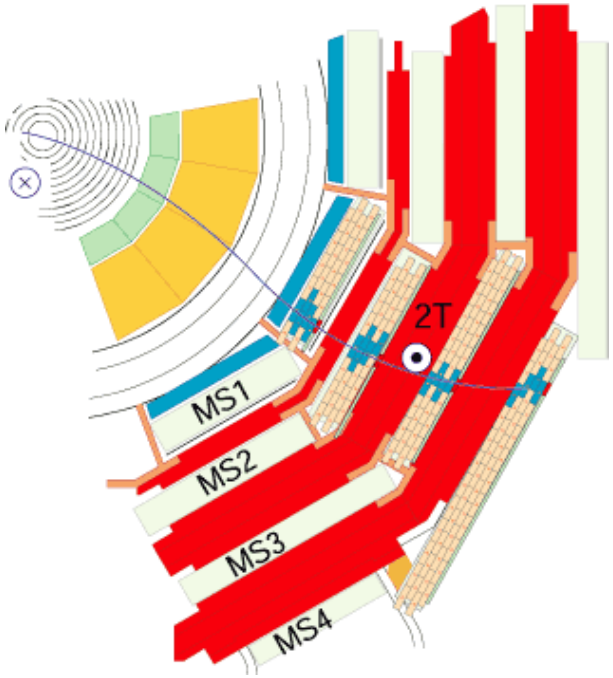
BETHEL UNIVERSITY



Physics object pre-learning

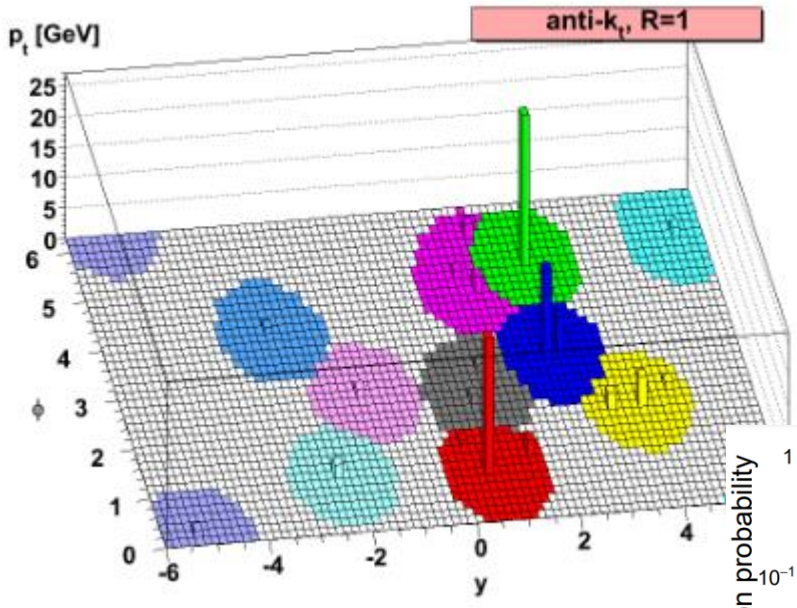
- Electrons & photons
- Muons
- Tau leptons

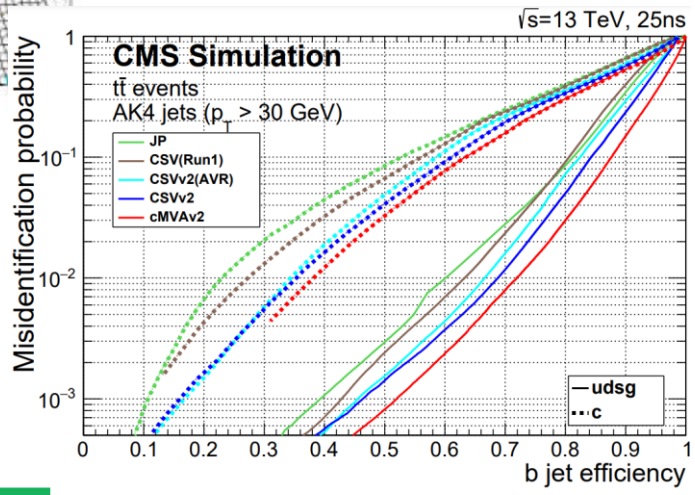


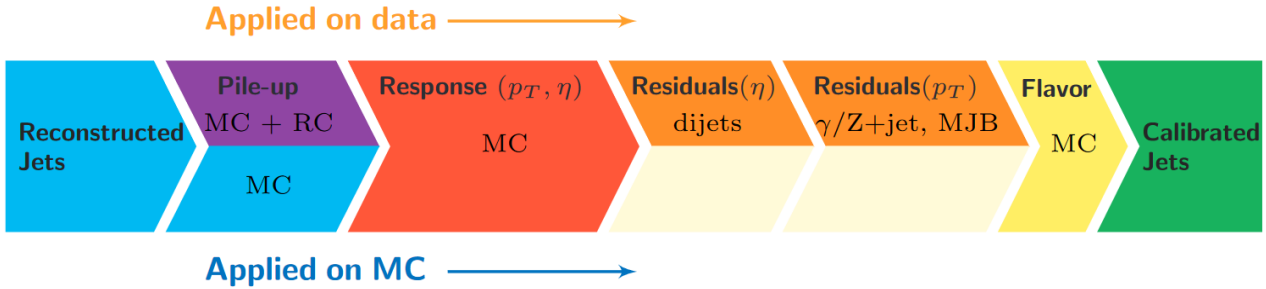


Physics object pre-learning

- Jets
- MET
- B-tagging
- Jet energy corrections







Event Selection philosophy

BETHEL UNIVERSITY

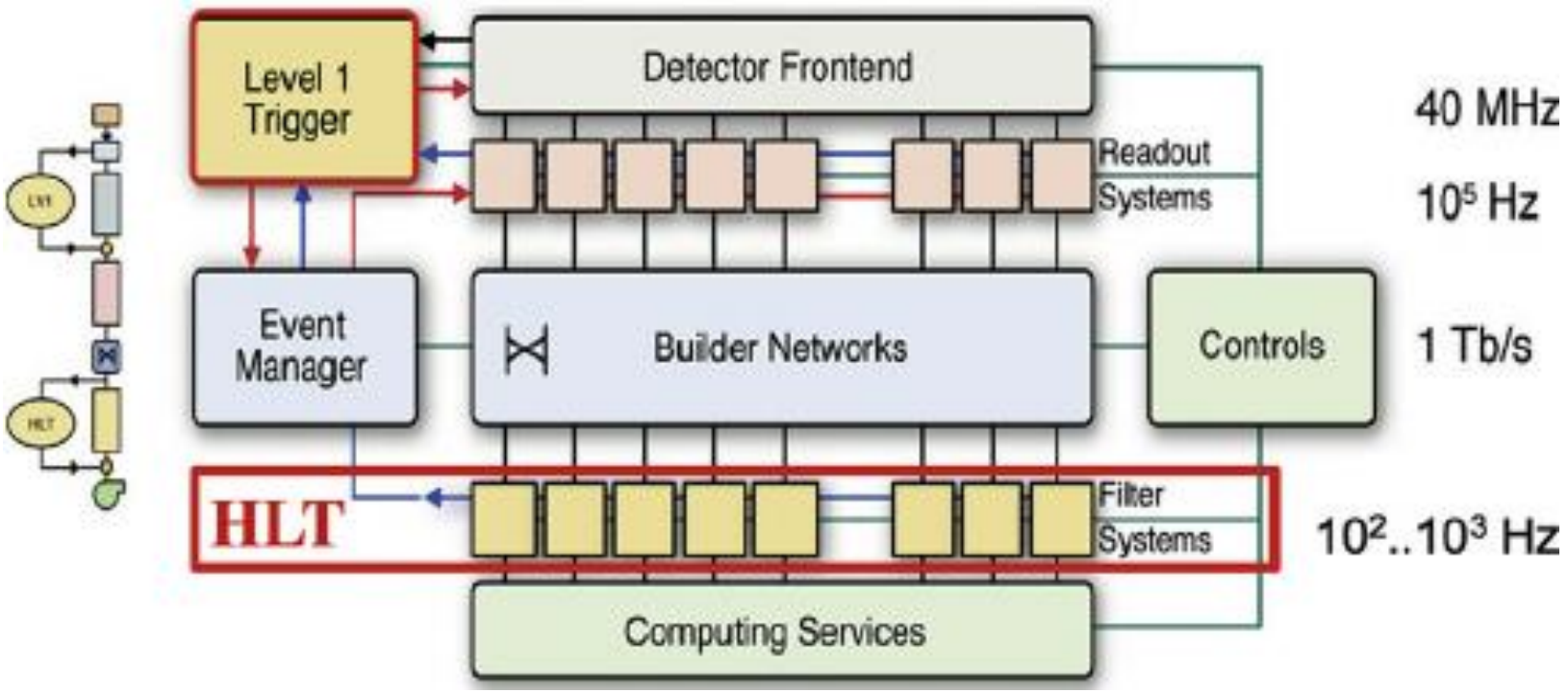
♡ Guiding principles

1. What physics objects should be present to represent the final state particles of my Feynman diagram? Should any of the objects be related to each other in a special way?
2. What physics objects should NOT be present?
3. What will cue CMS to store the types of events I want to analyze?

- Discuss!
- Any nuances or other principles we should add?

Triggers in CMS

- Level-1 trigger + high-level trigger



Triggers in CMS

- Triggers are implemented in CMSSW as collections of **filters** on different event features

Python

```
process.HLT_Mu20_v2 = cms.Path( process.HLTBeginSequence + process.hltL1sL1SingleMu16 + pro
cess.hltPreMu20 + process.hltL1fL1sMu16L1Filtered0 + process.HLTL2muonrecoSequence + proces
s.hltL2fL1sMu16L1f0L2Filtered10Q + process.HLTL3muonrecoSequence + process.hltL3fL1sMu16L1f
0L2f10QL3Filtered20Q + process.HLTEndSequence )
```

- Triggers are fairly fluid and change over time:
  - Version numbers update pretty regularly
  - Some triggers' acceptance rates are tied to the overall "luminosity" of the detector
- Prescale = X means "1 out of X passing events are kept"

Data streams

- Collision data is sorted into “streams” based on which triggers fired for each event
  - Convenient for analyzers!
  - Beware events appearing in multiple datasets
- Peak into Run 3:

AlCaLowPtJet
AlCaLumiPixelsCountsPrompt
AlCaP0
AlCaPPSPrompt
AlCaPhiSym
BTagMu
Commissioning
Cosmics
DisplacedJet
EGamma(0,1)
EphemeralHLTPhysics(0-19)
EphemeralZeroBias(0-19)

HLTPhysics
HcalNZS
IsolatedBunch
JetMET(0,1)
L1Accept
MiniDaq
MinimumBias
Muon(0,1)
MuonEG
NoBPTX
ParkingDoubleElectronLowMass
ParkingDoubleMuonLowMass(0-7)
ParkingHH

ParkingLLP
ParkingVBF(0-7)
RPCMonitor
ReservedDoubleMuonLowMass
ScoutingPFMonitor
ScoutingPFRun3
SpecialHLTPhysics(0-31)
SpecialRandom(0-19)
Tau
TestEnablesEcalHcal
ZeroBias,(0-19)
ZeroBiasNonColliding

- Prompt Reco switched to CMSSW_13_0_9, 18 hours ago

MiniAOD triggering

- How can I get the trigger info?!
  - Directly from information stored in MiniAOD Files
  - From the CMS conditions database via “provider” classes

- First, from the files!
- WATCH JULIE – we will work through the lesson page, but the “doing” isn’t super exciting for everyone to do individually.

https://cms-opendata-workshop.github.io/workshop2023-lesson-selection/03-miniaodtrigger/index.html

Conditions DB triggering

- TriggerAnalyzer.cc has an example of storing trigger names and pass/fail * prescale information
- Checks the "HLTPrescaleProvider" to see if the settings changed

```
166     // ------------ method called when starting to processes a run  ------------
167     void TriggerAnalyzer::beginRun(edm::Run const& iRun, edm::EventSetup const& iSetup)
168     //--------------------------------------------------------------------------
169     {
170         using namespace std;
171         using namespace edm;
172
173         bool changed(true);
174         hltPrescaleProvider_.init(iRun,iSetup,processName_,changed);
175         if (changed){
176           cout<<"HLTConfig has changed for this Run. . . "<<endl;
177        }
178     } //------------------- beginRun()
```

Conditions DB triggering

- Uses the providers to regularly update trigger name list
- Then digs into each name to check prescales and acceptance

```
HLTConfigProvider const&  hltConfig = hltPrescaleProvider_.hltConfigProvider();

// sanity check
assert(triggerResultsHandle_->size()==hltConfig.size());

//Inspired in https://github.com/cms-sw/cmssw/blob/CMSSW_7_6_X/HLTrigger/HLTfilters/src/HLTHighLevel.cc
// init the TriggerNames with the TriggerResults
const edm::TriggerNames & triggerNames = iEvent.triggerNames(*triggerResultsHandle_);
bool config_changed = false;
if (triggerNamesID_ != triggerNames.parameterSetID()) {
   triggerNamesID_ = triggerNames.parameterSetID();
   config_changed = true;
}
// (re)run the initialization of the container with the trigger patterns
// - this is the first event
// - or the HLT table has changed
if (config_changed) {
   initPattern(*triggerResultsHandle_, iSetup, triggerNames);
}
```

Conditions DB triggering

- Go back to the tutorial page to check out the results of this code!

https://cms-opendata-workshop.github.io/workshop2023-lesson-selection/04-triggerselection/index.html

Event selection challenge

BETHEL UNIVERSITY

- Let's design an analysis!

Workshop analysis example: $t\bar{t} \rightarrow (bjj)(b\ell\nu)$

Later in the workshop we will use a measurement of the top quark pair production cross section as an example analysis. The signal for this measurement is one top quark that decays hadronically, and one top quark that decays leptonically, to either a muon or an electron.

Your analysis example

What is a physics process that you might study? Let's design a possible CMS event selection. If your process includes a particle with multiple possible decay modes, choose one (or a small group of very similar decay modes) as a test case for this challenge.

Event selection challenge

BETHEL UNIVERSITY

Signal

Which final state particles would you expect to observe in the detector from your “signal” process?

Based on these particles, consider:

- Which trigger or triggers would be useful to make sure your signal events are represented in the dataset?
- Which objects should you require in each event?
- What kinematic criteria might you requier for each object? (momentum, angular regions, etc)
- What quality criteria might you require for each object? (identification, isolation)
- Are there any correlations between your objects that you might exploit?

Background

Which SM backgrounds could easily mimic your signal, given a few extra physics objects, or a few missing physics objects?

Based on these processes, consider:

- Which background simulations should you include in your study?
- Should you apply any upper or lower limits on the numbers of certain physics objects in your events?
- Are there any objects you should *veto* from your events?
- What quality criteria might you choose for vetoing objects?

Event selection challenge

- Walk through the decisions for the ttbar analysis on the tutorial webpage

https://cms-opendata-workshop.github.io/workshop2023-lesson-selection/06-solutions/index.html