Reconstructing boosted resonance using CA/FatJets and jet pruning

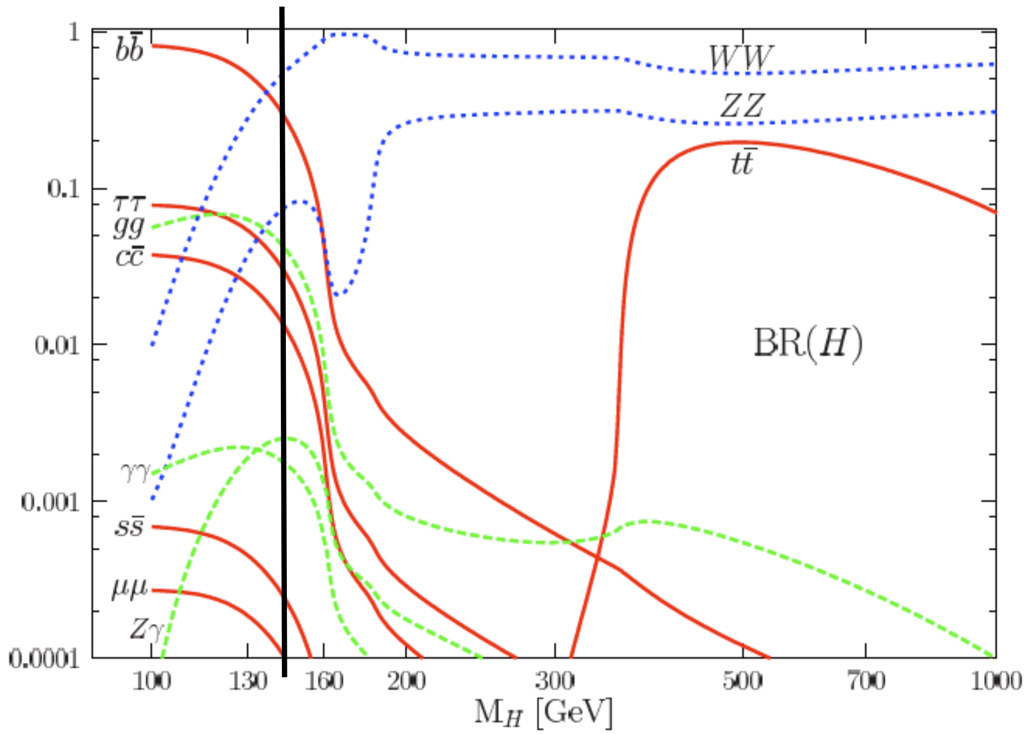
Kalanand Mishra

Fermilab

Motivation

HV - Higgs discovery channel

[Butterworth, Davison, Rubin, Salam PRL 100 (2008)]



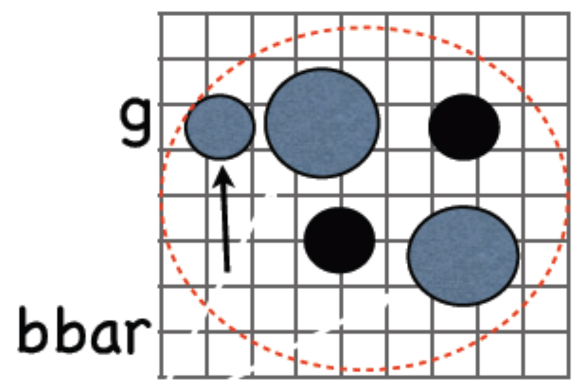
H -> b,bbar

If highly boosted, a single merged fat jet observed in detector

Z -> l^+l^-, W→lv

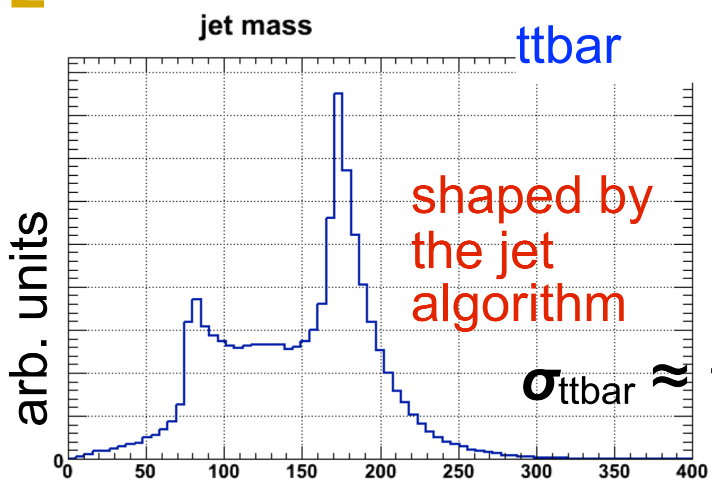
Apply filtering and take 3 hardest subjets

Use b-tagging on 2 hardest subjets

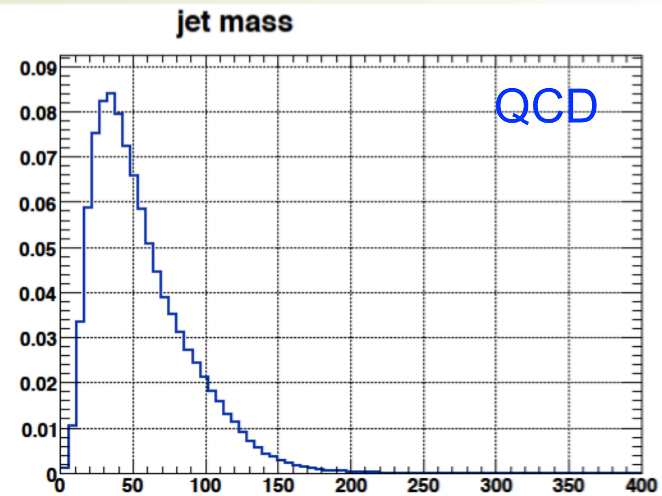


- But first need to validate the technique on top, W.
- After good understanding can apply this to other BSM searches with boosted topology.

Use top quark as surrogate

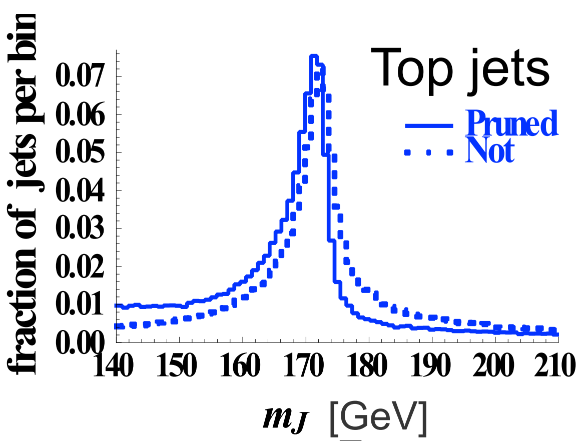


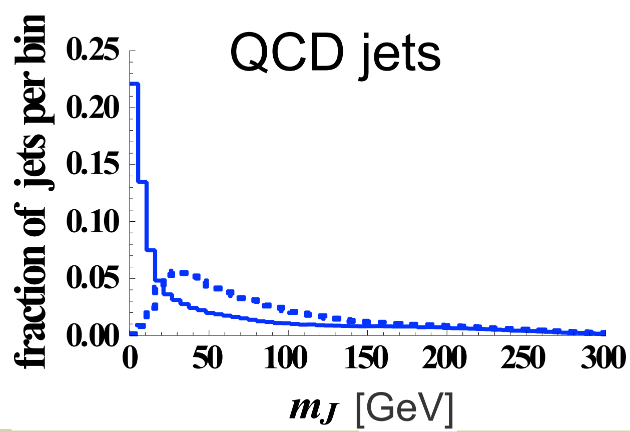
$\sigma_{ttbar} \approx 10^{-3}\sigma_{jj}$



⇒The mass resolution of *pruned* top jets is narrower **Win**

⇒ *Pruned* QCD jets have lower mass, sometimes much lower **Win**





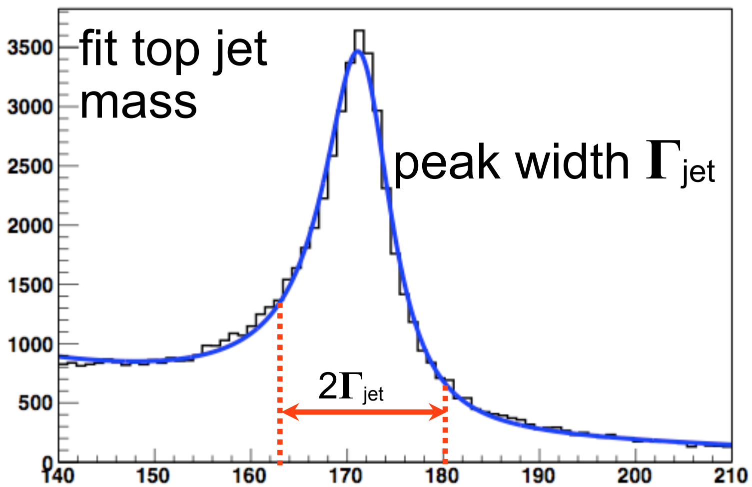
For details see Steve Ellis's talk at LPC (Nov 18, 2010) http://indico.cern.ch/conferenceDisplay.py?confId=113974

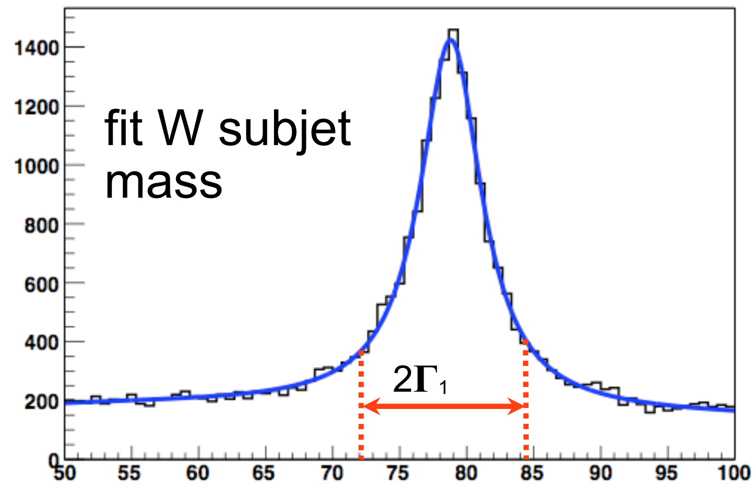
Defining Reconstructed Tops

From Steve's talk

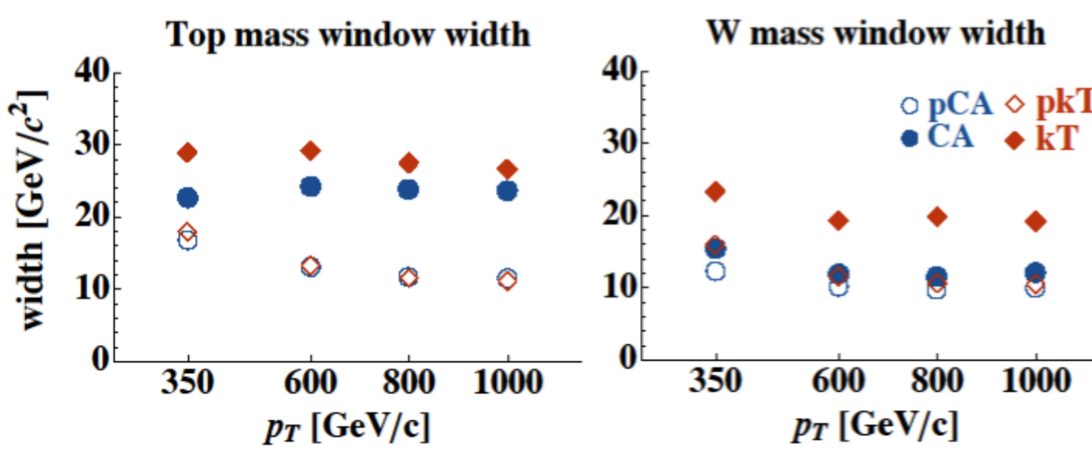
fit mass windows to identify a reconstructed top quark

cut on masses of jet (top mass) and subjet (W mass)





- ✦Pruned windows narrower ➔ better mass resolution
- ✦Pruned window widths fairly consistent between algorithms (not true of unpruned), over the full range in pT

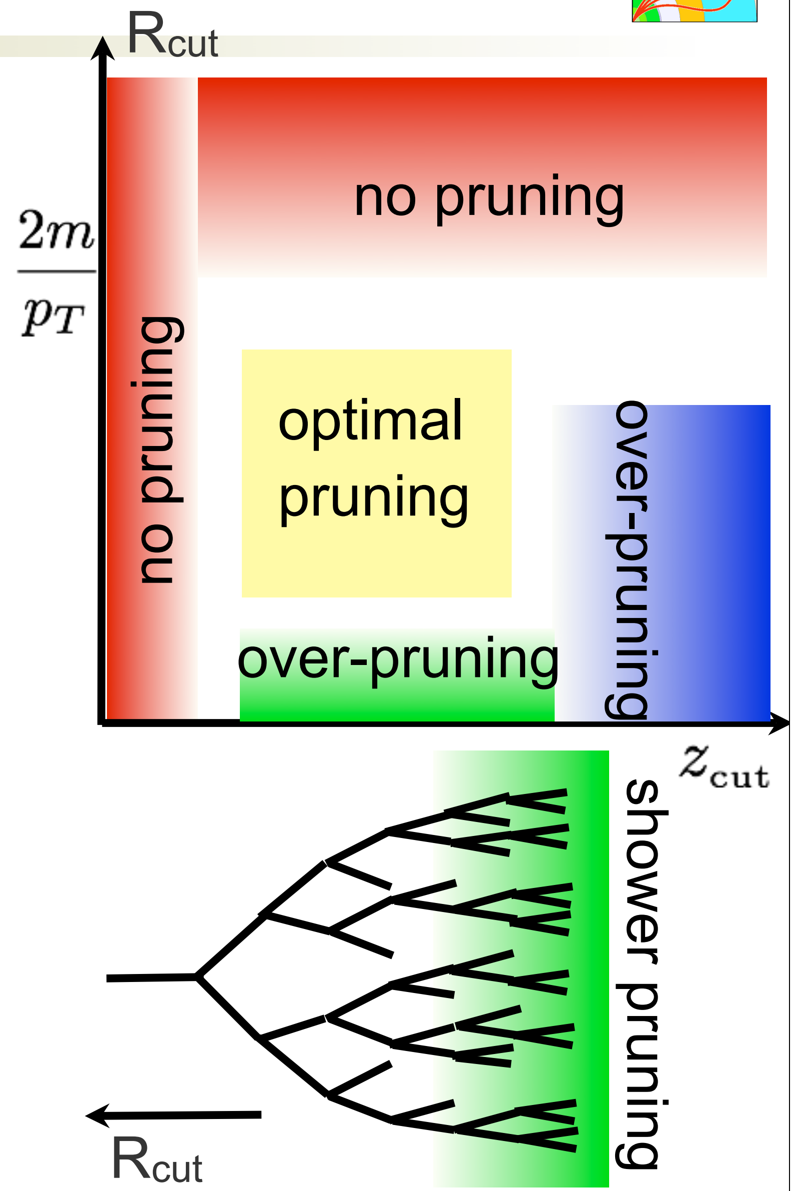


CMSSW software implementation of
fat (Cambridge-Aachen) jets and pruning

What it all works ?

Consider generic recombination step: *i,j* ➜ *p*

- Useful variables: $z = \frac{\min(p_{T_i}, p_{T_j})}{p_{T_p}}$
  (Lab frame)
- Merging metrics: $\rho_{CA} = \Delta R / D$
- Daughter masses
  (scaled by jet mass) : $a_{i,j} \equiv \frac{m_{i,j}}{m_J}$



S. Ellis et al (also ATLAS) choose:

CA: z_{cut} = 0.1 and R_{cut} = $m_J/P_{T,J} = 0.5$

There are just two parameters to tune.
Currently CMS also uses the above cuts.
[JetMET approved, boosted top analysis]

CMSSW snippet

FatJet parameters are defined in: /CMSSW/RecoJets/JetProducers/python/SubJetParameters_cfi.py

```
import FWCore.ParameterSet.Config as cms

SubJetParameters = cms.PSet(
    #read from CompoundJetProducer:
    subjetColl = cms.string("SubJets"),    # subjet collection name
    #read from SubJetProducer:
    algorithm = cms.int32(1),               # 0 = KT, 1 = CA, 2 = anti-KT
    centralEtaCut = cms.double(2.5),        # eta cut for the "fat jet"
    jetSize = cms.double(0.8),              # jet algorithm cut-off parameter for definition of fat jet (meaning depends on jet algorithm)
    nSubjets = cms.int32(3),                # number of subjets to decluster the fat jet into (actual value can be less if less than 6 constituents in fat jet).
    enable_pruning = cms.bool(True),        # enable pruning (see arXiv:0903.5081)
    zcut = cms.double(0.1),                 # zcut parameter for pruning (see ref. for details)
    rcut_factor = cms.double(0.5)           # rcut factor for pruning (the ref. uses 0.5)
)
```

Example script to cluster CA FatJet: /UserCode/srappocc/src/Analysis/TTBSMPatTuples/test/ttbsm_cfg.py

```
###############################
###### Jet Pruning Setup ######
###############################


# Pruned PF Jets
process.caPrunedPFlow = cms.EDProducer(
    "SubJetProducer",
    PFJetParameters.clone( src = cms.InputTag('pfNoElectron'+postfix),
                           doAreaFastjet = cms.bool(True),
                           doRhoFastjet = cms.bool(True),
                           inputEMin = cms.double(1.0),
                           Ghost_EtaMax = cms.double(6.0)
                           ),
    AnomalousCellParameters,
    SubJetParameters,
    jetAlgorithm = cms.string("CambridgeAachen"),
    rParam = SubJetParameters.jetSize,
    jetCollInstanceName=cms.string("subjets")
    )

process.caPrunedPFlow.nSubjets = cms.int32(2)

# Pruned PF Jets, pileup removed
process.caPrunedPFlowPUSub = cms.EDProducer(
    "SubJetProducer",
    PFJetParameters.clone( src = cms.InputTag('pfNoElectron'+postfixPUSub),
                           doAreaFastjet = cms.bool(True),
                           doRhoFastjet = cms.bool(True),
                           inputEMin = cms.double(1.0),
                           Ghost_EtaMax = cms.double(6.0)
                           ),
    AnomalousCellParameters,
    SubJetParameters,
    jetAlgorithm = cms.string("CambridgeAachen"),
    rParam = SubJetParameters.jetSize,
    jetCollInstanceName=cms.string("subjets")
    )

process.caPrunedPFlowPUSub.nSubjets = cms.int32(2)
```

This can work both in CMSSW/Reco and PAT frameworks. There are privately-produced PAT-tuples with FatJet already stored (I can send you DBS link if interested).

Laundry list of "things-to-do"

✦The top pair production cross section at LHC is ≈ 150 pb
- -assuming 25% acceptance and 10% efficiency we should get *O*(100-150) boosted events from analyzing 36 pb^{-1} data
- -we can study the method on MC (TTbarJet and QCD samples)

✦The pruning parameters (Z_{cut} and R_{cut}) values are not necessarily optimal for our searches -- although they are set to some sensible default values.
- -perhaps we may want to explore optimization option

✦Currently jet corrections are not available for FatJets
- -apply no correction for now (the corrections for high p_T jet should be small)
- -similarly, at the moment use the same pile-up subtraction as derived for anti-kT 0.5 jets
- -both these can be improved

✦No obvious calibration source to calibrate b-bbar invariant mass
- -to understand detector resolution etc
- -perhaps photon+Z(→bb) can be explored, but currently only MC study possible