Semi-visible jets: Sample production in MadGraph and CMSSW

Eshwen Bhal, Ben Krikler

05th April, 2018

Outline of current progress

◎ Performed complete private sample production of s- and t-channel processes using UFO files provided by theorists

◎ Used MadGraph5@aMCatNLO as generator and created gridpacks

◎ Ran gridpacks through FullSim CMSSW chain

◎ Pythia 8.226 used as hadroniser, GEANT4 as detector simulation

◎ Processed events through to nanoAOD format for easier analysis

◎ Emulating 2016 MC with 2017 re-processing

◎ All my instructions and files for reproducing are in https://github.com/eshwen/SemivisibleJets

Gridpack generation with MadGraph

◎ Followed instructions outlined by generators group. Required scripts in https://github.com/cms-sw/genproductions

◎ Used MadGraph5_aMCatNLO v2.6.0, NNPDF3.0, at leading order. Set `xqcut = 100` with MLM matching to mimic 2017 paper

◎ Gridpack tarball created on batch containing MadGraph release, model files, input cards, and scripts required to generate LHE file

Parameter (t-channel model)	Value
$m_{Z'}$	1000 GeV
m_d (dark hadron mass scale)	10 GeV
g_d (coupling between dark hadrons and Z')	1
α_d (running dark coupling strength, set in Pythia parameters)	0.1 at 1 TeV
r_{inv} (set in Pythia parameters?)	?

FullSim chain in CMSSW

◎ Steps:

1. Gridpack to LHE-GEN-SIM (includes Pythia hadronisation with Hidden Valley module). Used CMSSW_7_1_30
2. GEN-SIM to AOD (step 1). Used CMSSW_8_0_21
3. AOD step 1 to AOD step 2. Used CMSSW_8_0_21
4. AOD step 2 to miniAOD. Used CMSSW_8_0_21
5. MiniAOD to nanoAOD. Used CMSSW_9_4_4

Several versions required as backward compatibility is an issue when emulating samples produced over several eras

◎ Currently able to run all steps locally and via HTCondor. Developing version to run on CRAB (processed up to GEN-SIM at the moment)

Hidden Valley part of gen fragment

```
    processParameters = cms.vstring(
    #'TimeShower:nPartonsInBorn = 2', #number of coloured particles (before resonance decays) in
born matrix element
    'HiddenValley:ffbar2Zv = on', #it works only in the case of narrow width approx
    #'HiddenValley:Run = on', # turn on coupling running
    'HiddenValley:fragment = on', # enable hidden valley fragmentation
    #'HiddenValley:NBFlavRun = 0', # number of bosonic flavor for running
    #'HiddenValley:NFFlavRun = 2', # number of fermionic flavor for running
    'HiddenValley:alphaOrder = 1', # order at which running coupling runs
    'HiddenValley:Lambda = 0.1', # parameter used for running coupling
    'HiddenValley:nFlav = 1', # this dictates what kind of hadrons come out of the shower, if
nFlav = 2, for example, there will be many different flavor of hadrons
    'HiddenValley:probVector = 0.75', # ratio of number of vector mesons over scalar meson, 3:1
is from naive degrees of freedom counting
    'HiddenValley:pTminFSR = 10', # cutoff for the showering, should be roughly confinement scale
    ),
```

◎ Copied from Giorgia's Pythia-generated code

◎ Not sure if s-channel and t-channel processes will need different arguments

◎ In Giorgia's version, r_{inv} specified below when adding channels

Setbacks

◎ Theorists recommended changing PDGIDs of several particles after LHE step, but before hadronisation, so Pythia recognised them as certain HV particles and decayed them properly

- Required hack of a script in gridpack to change them during LHE-GEN-SIM step in CMSSW

◎ Took a while to get the correct layout of gen fragment and correct `cmsDriver` commands

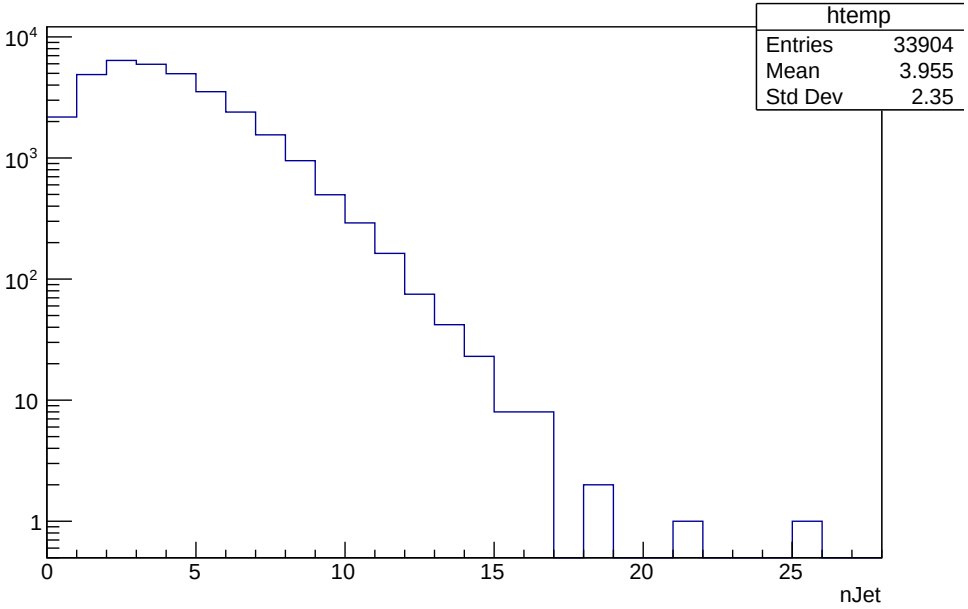
- Took inspiration from commands used to generate dataset **/Axial_MonoJ_NLO_Mphi-100_Mchi-1_gSM-0p25_gDM-1p0_13TeV-madgraph/RunIISummer*/***

◎ Developing a CRAB-friendly version took longer than expected

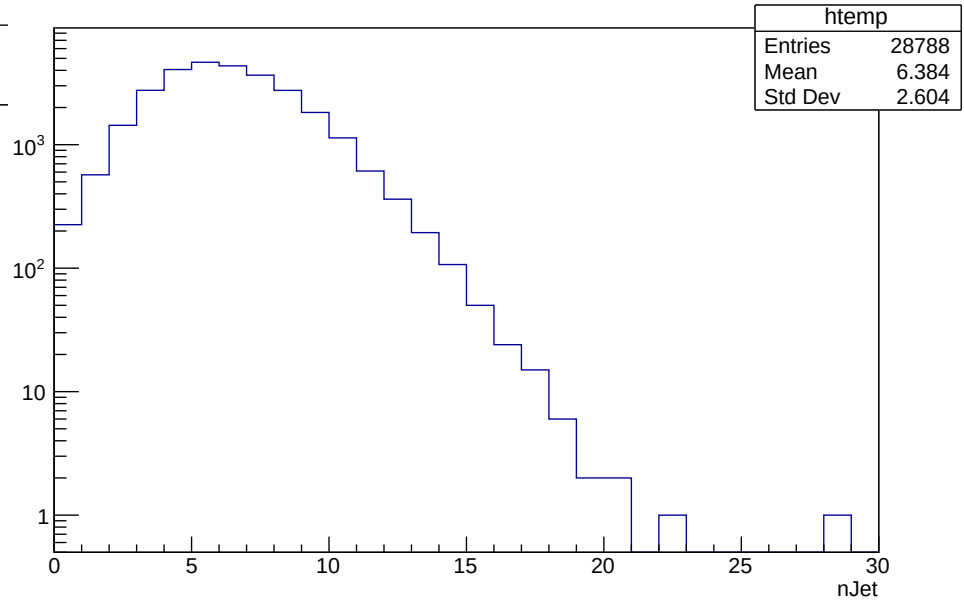
- CRAB unable to find gridpack for some reason. Private production using local gridpacks has poor documentation, but solution eventually found
- CRAB blacklists Bristol T2 site often so progress can be halted from that

First look at distributions (n_{jet})

s-channel (spin-1)

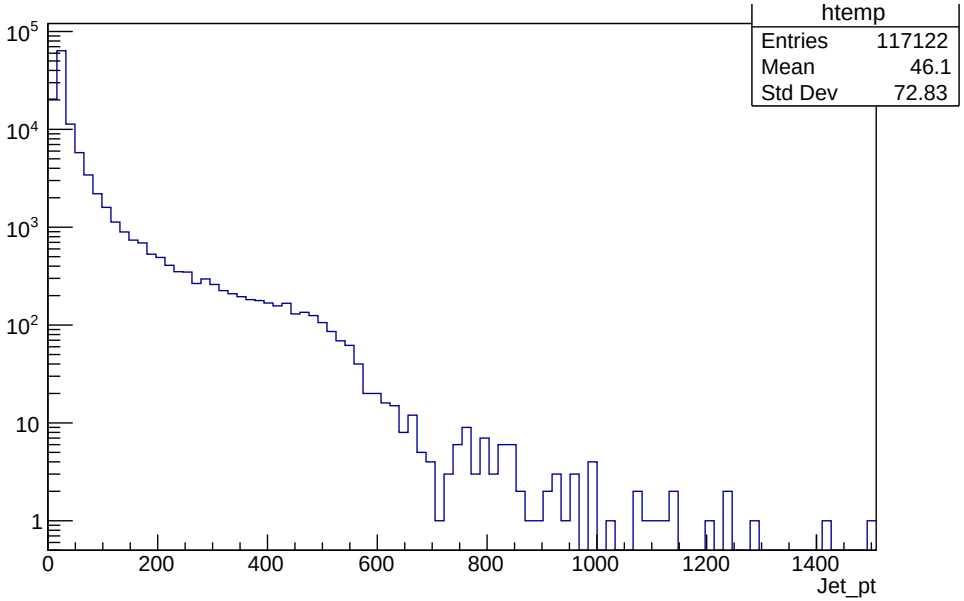


t-channel

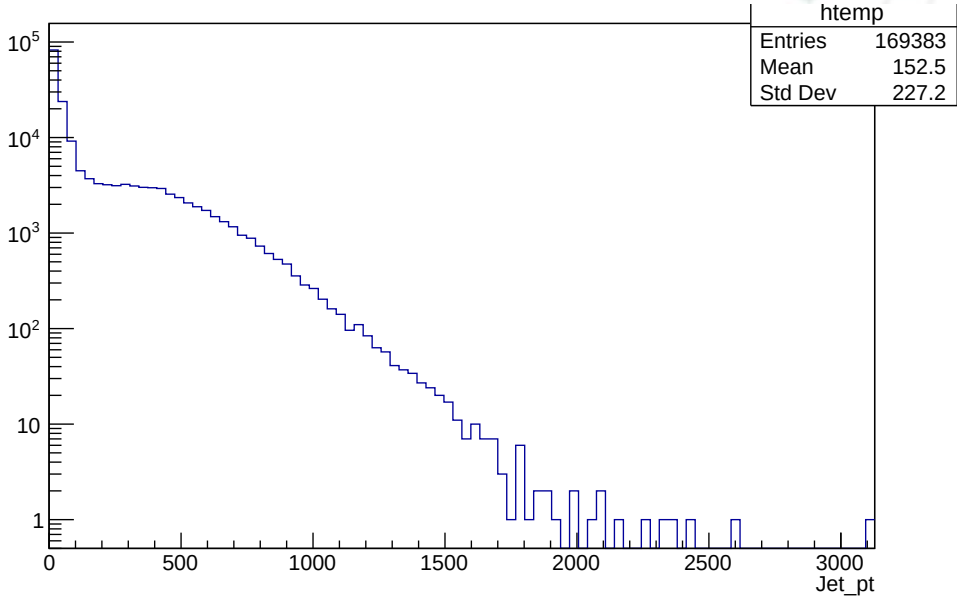


First look at distributions (jet p_T)

s-channel (spin-1)

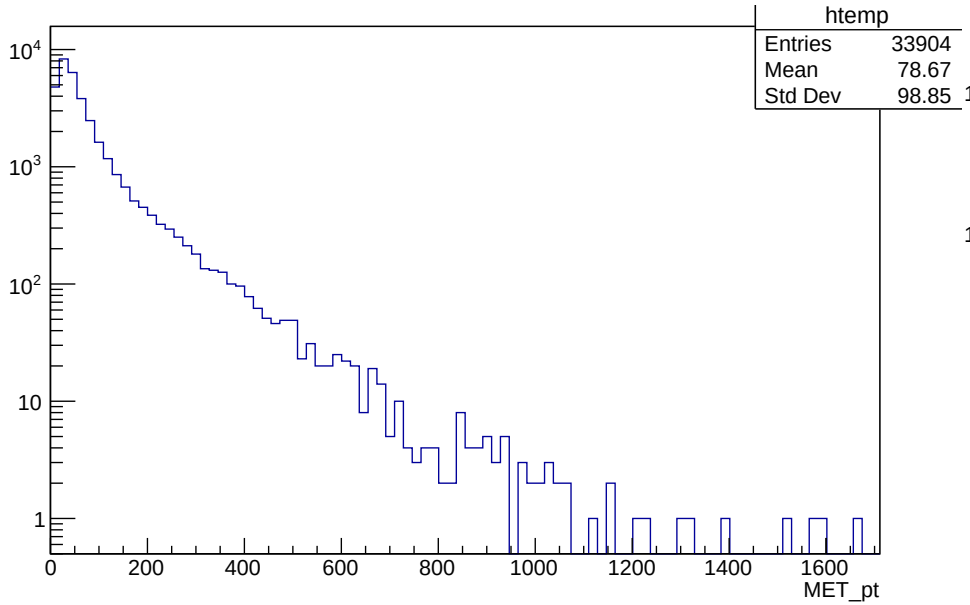


t-channel

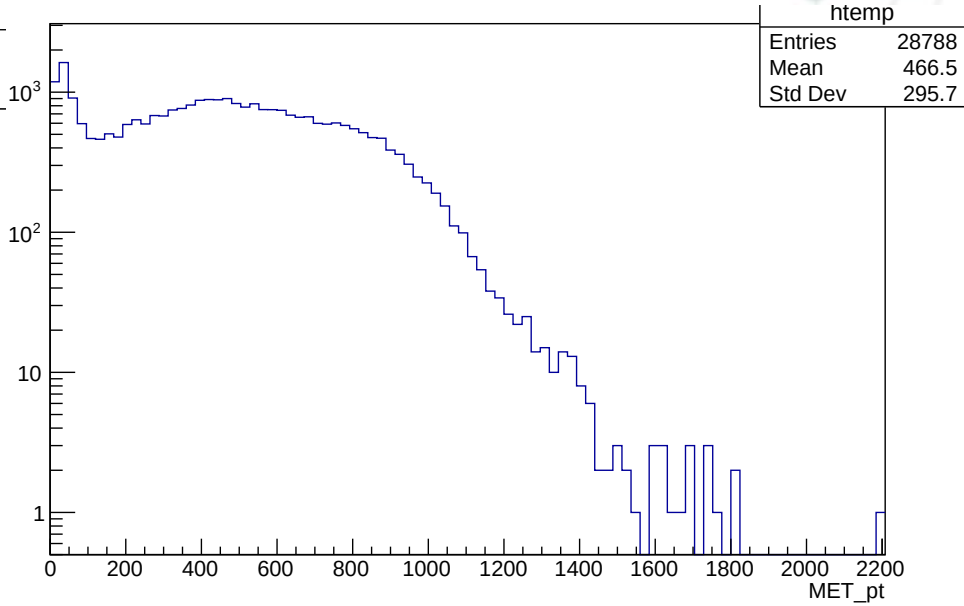


First look at distributions (E_T^{miss})

s-channel (spin-1)



t-channel



◎ Haven't had time to make plots of calculated variables (H_T, M_T, $\Delta\phi$, etc.), but will look into those

Status and future plans

◎ Generated gridpacks and run FullSim CMSSW chain on Condor with 50k events for s- and t-channel. Also processing with CRAB

◎ Want to run full chain with same parameters as Giorgia and compare distributions

◎ Want to vary model parameters to produce samples that make up a large (coarse or fine?) scan of phase space

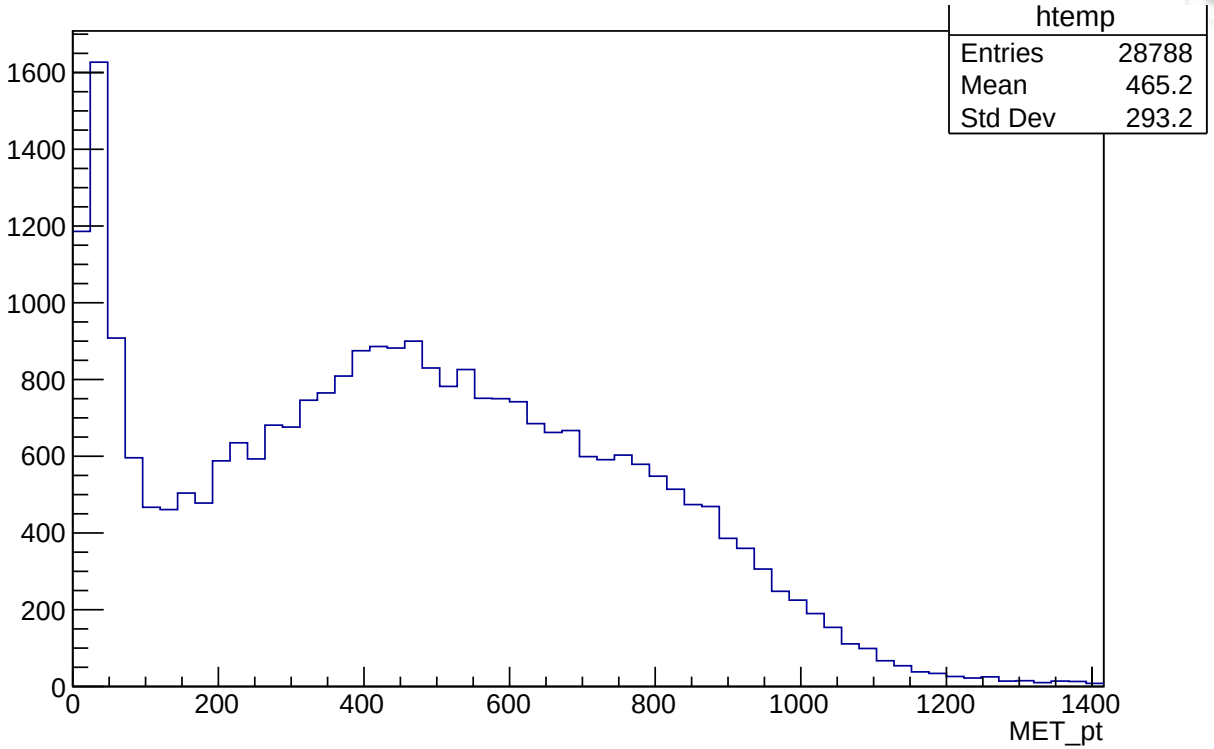
- Possible, but painstaking, to change values in UFO files. Would have one MadGraph model for each combination of parameters
- May be possible to specify arguments in MadGraph param/run card to change parameters. Only need a single model, then one input card per combination to change values

◎ Need to find out the procedure for central production and aim to request it soon (unless EXO MC contacts can “bless” private samples)

Backup

Closer look at E_T^{miss} distribution (t)

t-channel



◎ Haven't had that much time to digest plots. Unsure why there's a second peak around 450 GeV

Gen fragment, part 1 (LHE portion)

```
import FWCore.ParameterSet.Config as cms

# Needed as I'm using an external generator
externalLHEProducer = cms.EDProducer("ExternalLHEProducer",
    args =
cms.vstring('/afs/cern.ch/work/e/ebhal/public/DMsimp_SVJ_t_slc6_amd64_gcc481_CMSSW_7
_1_30_tarball.tar.xz'),
    nEvents = cms.untracked.uint32(50000),
    numberOfParameters = cms.uint32(1),
    outputFile = cms.string('cmsgrid_final.lhe'),
    scriptName =
cms.FileInPath('GeneratorInterface/LHEInterface/data/run_generic_tarball_cvmfs.sh')
)
```

Gen fragment, part 2 (GEN-SIM)

```
from Configuration.Generator.Pythia8CommonSettings_cfi import *
from Configuration.Generator.Pythia8CUEP8M1Settings_cfi import *
from Configuration.Generator.Pythia8aMCatNLOSettings_cfi import *

generator = cms.EDFilter("Pythia8HadronizerFilter",
    maxEventsToPrint = cms.untracked.int32(1),
    pythiaPylistVerbosity = cms.untracked.int32(1),
    filterEfficiency = cms.untracked.double(1.0),
    pythiaHepMCVerbosity = cms.untracked.bool(False),
    crossSection = cms.untracked.double(97.0),
    comEnergy = cms.double(13000.),

    PythiaParameters = cms.PSet(
        pythia8CommonSettingsBlock,
        pythia8CUEP8M1SettingsBlock,
        pythia8aMCatNLOSettingsBlock,
        JetMatchingParameters = cms.vstring(
            'JetMatching:setMad = off',
            'JetMatching:scheme = 1',
            'JetMatching:merge = on',
            'JetMatching:jetAlgorithm = 2',
            'JetMatching:etaJetMax = 5.',
            'JetMatching:coneRadius = 1.',
            'JetMatching:slowJetPower = 1',
            'JetMatching:qCut = 100.', #this is the actual merging scale
            'JetMatching:nJetMax = 2', #number of partons in born matrix element for highest
multiplicity
            'JetMatching:doShowerKt = off', #off for MLM matching, turn on for shower-kT matching
        ),
```

Chosen to emulate 2017 paper as closely as possible

Gen fragment, part 3 (GEN-SIM contd)

```
    processParameters = cms.vstring(
    #'TimeShower:nPartonsInBorn = 2', #number of coloured particles (before resonance decays) in
born matrix element
    'HiddenValley:ffbar2Zv = on', #it works only in the case of narrow width approx
    #'HiddenValley:Run = on', # turn on coupling running
    'HiddenValley:fragment = on', # enable hidden valley fragmentation
    #'HiddenValley:NBFlavRun = 0', # number of bosonic flavor for running
    #'HiddenValley:NFFlavRun = 2', # number of fermionic flavor for running
    'HiddenValley:alphaOrder = 1', # order at which running coupling runs
    'HiddenValley:Lambda = 0.1', # parameter used for running coupling
    'HiddenValley:nFlav = 1', # this dictates what kind of hadrons come out of the shower, if
nFlav = 2, for example, there will be many different flavor of hadrons
    'HiddenValley:probVector = 0.75', # ratio of number of vector mesons over scalar meson, 3:1
is from naive degrees of freedom counting
    'HiddenValley:pTminFSR = 10', # cutoff for the showering, should be roughly confinement scale
    ),

    parameterSets = cms.vstring('pythia8CommonSettings',
        'pythia8CUEP8M1Settings',
        'pythia8aMCatNLOSettings',
        'processParameters',
        'JetMatchingParameters'
        )
    )
)
```

© `processParameters` copied from Giorgia's Pythia-generated code

cmsDriver commands

◎ Gridpack to LHE-GEN-SIM:

```
cmsDriver.py Configuration/GenProduction/python/${gen_frag_file} --fileout
file:${model_name}_LHE_GEN_SIM.root --mc --eventcontent RAWSIM,LHE --customise
SLHCUpgradeSimulations/Configuration/postLS1Customs.customisePostLS1,Configuration/DataProcessing
/Utils.addMonitoring --datatier GEN-SIM,LHE --conditions MCRUN2_71_V1::All --beamspot
Realistic50ns13TeVCollision --step LHE,GEN,SIM --magField 38T_PostLS1 --python_filename
${model_name}_LHE_GEN_SIM.py --no_exec -n $n_events
```

◎ GEN-SIM to AOD (step 1):

```
cmsDriver.py step1 --filein file:${model_name}_LHE_GEN_SIM.root --fileout
file:${model_name}_AOD_step1.root --pileup_input /store/mc/RunIISpring15PrePremix/Neutrino_E-
10_gun/GEN-SIM-DIGI-RAW/PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v2-v2/100000/001EB167-
3781-E611-BE3C-0CC47A4D75F4.root --mc --eventcontent PREMIXRAW --datatier GEN-SIM-RAW --
conditions 80X_mcRun2_asymptotic_2016_TrancheIV_v6 --step
DIGIPREMIX_S2,DATAMIX,L1,DIGI2RAW,HLT:@frozen2016 --datamix PreMix --era Run2_2016 --
python_filename ${model_name}_AOD_step1.py --no_exec --customise
Configuration/DataProcessing/Utils.addMonitoring -n $n_events
```

cmsDriver commands

◎ AOD (step 1) to AOD (step 2):

```
cmsDriver.py step2 --filein file:${model_name}_AOD_step1.root --fileout
file:${model_name}_AOD_step2.root --mc --eventcontent AODSIM --runUnscheduled --datatier AODSIM -
-conditions 80X_mcRun2_asymptotic_2016_TrancheIV_v6 --step RAW2DIGI,RECO,EI --era Run2_2016 --
python_filename ${model_name}_AOD_step2.py --no_exec --customise
Configuration/DataProcessing/Utils.addMonitoring -n $n_events
```

◎ AOD (step 2) to miniAOD:

```
cmsDriver.py --filein file:${model_name}_AOD_step2.root --fileout file:${model_name}_MINIAOD.root
--mc --eventcontent MINIAODSIM --runUnscheduled --datatier MINIAODSIM --conditions
80X_mcRun2_asymptotic_2016_TrancheIV_v6 --step PAT --era Run2_2016 --python_filename
${model_name}_MINIAOD.py --no_exec --customise Configuration/DataProcessing/Utils.addMonitoring -
n $n_events
```

◎ MiniAOD to nanoAOD:

```
cmsDriver.py --filein file:${model_name}_MINIAOD.root --fileout file:${model_name}_NANOAOD.root -
-mc --eventcontent NANOAODSIM --datatier NANOAODSIM --conditions auto:run2_mc -s NANO --era
Run2_2016,run2_miniAOD_80XLegacy --python_filename ${model_name}_NANOAOD.py --no_exec -n
$n_events
```

CRAB config example (GEN-SIM)

```
from CRABClient.UserUtilities import config, getUsernameFromSiteDB
config = config()

modelName = 'DMsimp_SVJ_t_MadGraph'
datasetStr = 'mZp_1000_md_10_alphaD_0p1_NNPDF30_13TeV-GEN-SIM'

# CRAB project directory
config.General.workArea = modelName
config.General.requestName = datasetStr
config.General.transferOutputs = True
config.General.transferLogs = True

config.JobType.pluginName = 'Analysis'
# Name of the CMSSW configuration file
config.JobType.psetName = 'DMsimp_SVJ_t_MadGraph_NNPDF_13TeV_GS.py'

config.Data.inputDataset = '/mZp_1000_md_10_alphaD_0p1_NNPDF30_13TeV-LHE/ebhal-
DMsimp_SVJ_t_MadGraph_mZp_1000_md_10_alphaD_0p1_NNPDF30_13TeV-LHE-
d5245ab584434bea4faa5aea256d691f/USER'
config.Data.inputDBS = 'phys03'
config.Data.splitting = 'EventAwareLumiBased'
config.Data.unitsPerJob = 100
config.Data.totalUnits = -1 # Run over all events
config.Data.outLFNDirBase = '/store/user/%s/' % (getUsernameFromSiteDB())
config.Data.publication = True # If true, output files are published on DBS. Useful for future
steps
config.Data.outputDatasetTag = modelName + '_' + datasetStr

config.Site.whitelist = ['T2_UK_SGrid_Bristol', 'T2_CH_CERN'] # CERN site needed so CRAB worker
nodes with /afs mounted can be used
config.Site.storageSite = 'T2_UK_SGrid_Bristol'
```