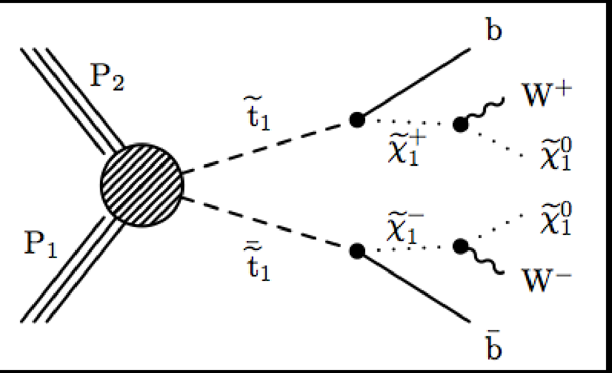
TAGGING LOW p_T B-JETS

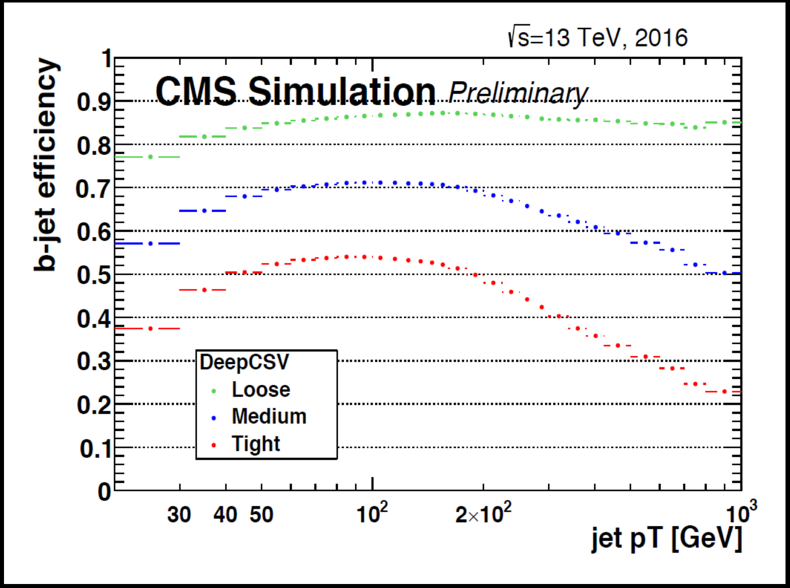
Erich Schmitz

WHY A LOW p_T B-TAGGER?

- Certain SUSY stop production models have a compressed mass scenario
    - mass difference between the stop and LSP is small
    - b quarks can be produced in intermediate steps of these stop decays, with a relatively low p_T
        - Want a tagger optimized for finding these b-jets

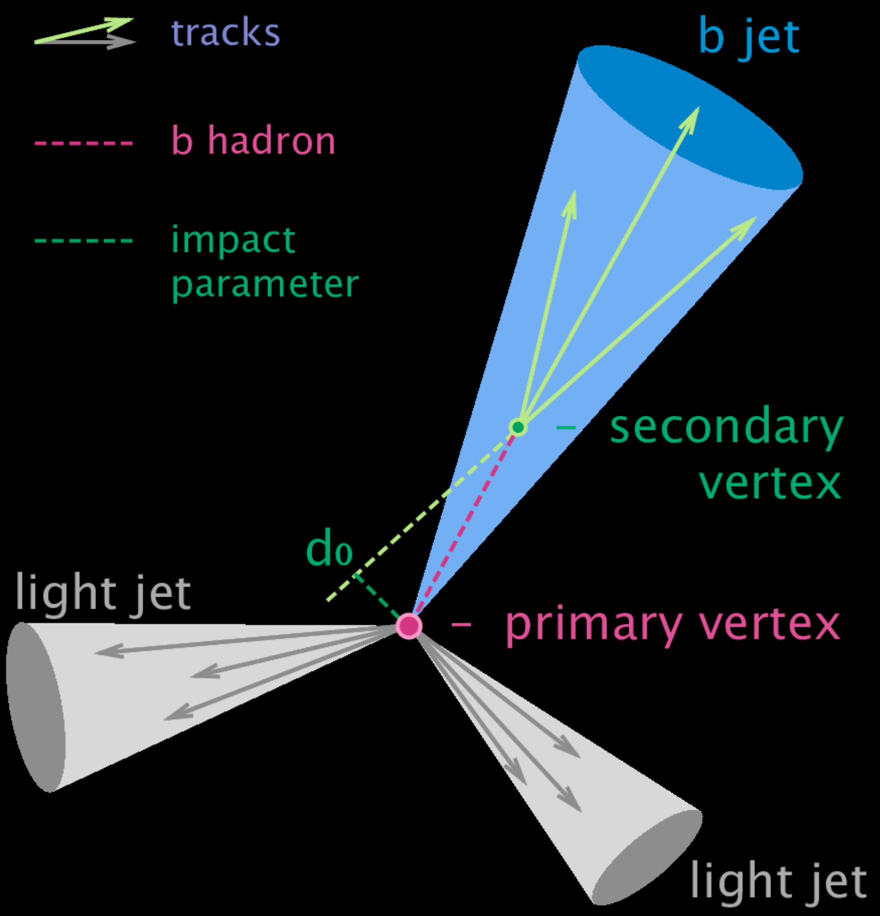


- Current taggers are supported for p_T ranges 20-1000 GeV
    - These are not optimized for a specific p_T range



JETS AND B HADRONS

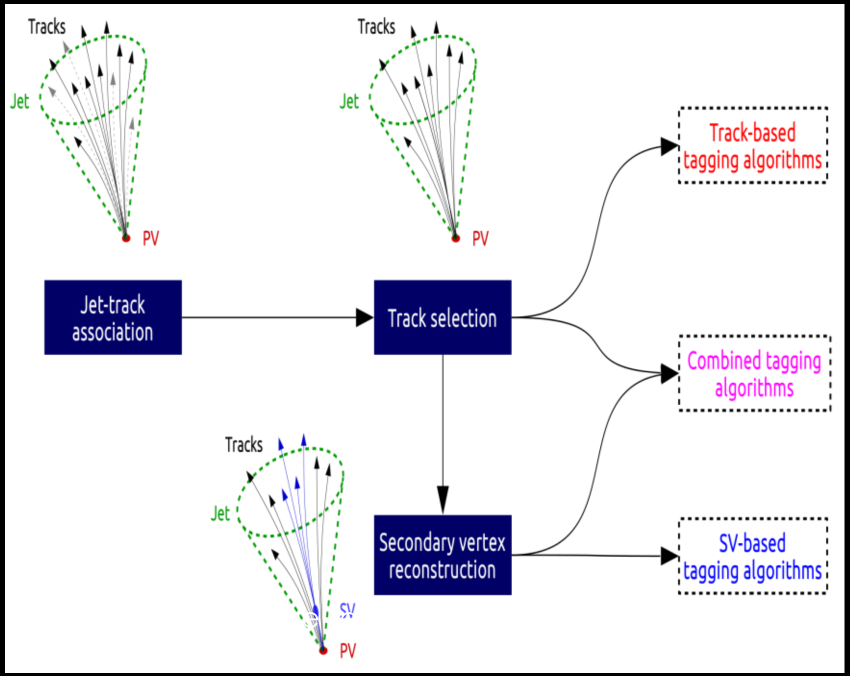
- Jets
  - narrow cone of particles produced by hadronization of quarks and gluons
- b-quarks hadronize into B-hadrons → forms jets
- b-jet identifiers
  - Displaced tracks
    - Large impact parameter
  - Displaced vertices (secondary vertex)
    - Sizable lifetime (1.5 ps)

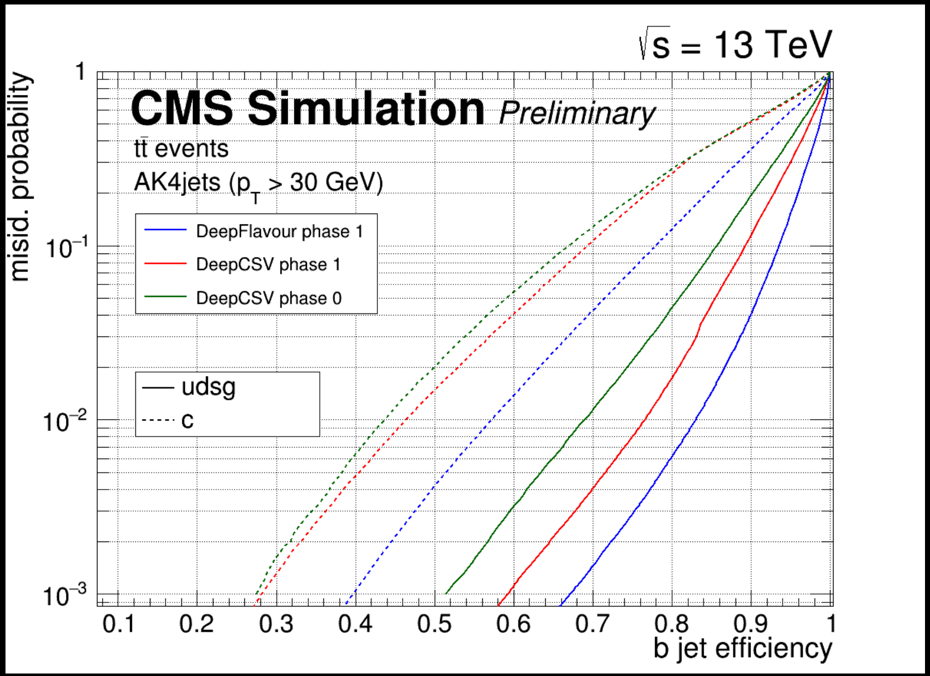


CSV ALGORITHM

- Combined Secondary Vertex (CSV):
  - Tagger that makes use of SV and track-based lifetime information
  - Iterations:
    - CSV (likelihood)
    - CSVv2 (Artificial NN)
    - DeepCSV (DNN)

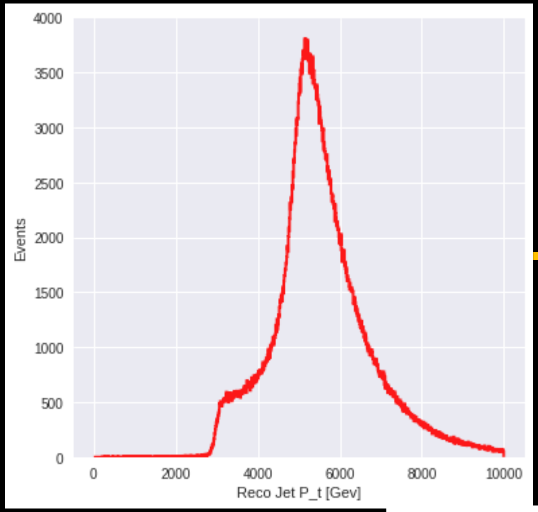
- Discriminating power for different situations
  - No vertex
  - Pseudo Vertex
  - Vertex



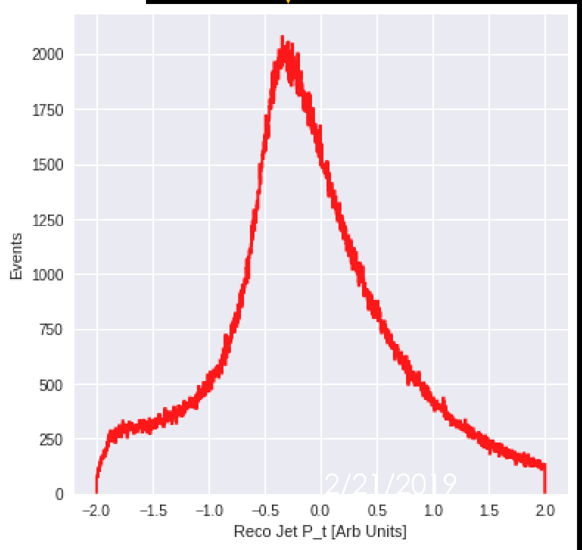


DEEPCSV – PREPROCESSING

- Data used is official CMS MC
  - QCD Multijets – Pythia8
  - $t\bar{t}$ - MadgraphMLM, Pythia8
- Data preparation
  - Original Samples converted to flat ROOT ntuples (DeepNTuples framework)
    - C
  - ROOT files converted to numpy arrays with pre-processing applied
    - Python, some compiled C modules
- Data pre-processing
  - Mean normalization
  - Zero-padding
  - p_T/η flattening of classes



Mean normalization



DEEPCSV - INPUTS

- 28 variables
  - **12 jet / jet-track associated variables**
    - p_T, η, n_{SV}, vertex category, $\sum\left(E_T^{tracks}\right)/E_T^{jet}$, $\Delta R(p_{tracks}^{\mu}, jet)$, 1st track IP significance/value above Charm, $n_{selected\ tracks}$, n_{tracks} w/η_{rel}
  - **7 track variables**, keeping up to **6 tracks per jet**
    - p_T^{rel}, min track-jet distance, $\Delta R(track, jet)$, $p_T/\mathrm{E_T}$, IP sig/val, decay length
  - **1 associated track variable**, keeping up to **4 entries per jet**
    - η^{rel}
  - **8 secondary vertex variables**, keeping up to **1 vertex per jet**
    - mass, n_{tracks}, E_T^{sv}/E_T^{Total}, $\Delta\mathrm{R}(\mathrm{sv}, \mathrm{jet})$, flight distance sig/val
- Grand total of **66 inputs**

DEEPCSV - DNN

- DNN Characteristics
  - 66 inputs (x)
  - 4 truth categories (classes) (y) → isB, isBB, isC, isUDSG
  - Fully connected
  - 7 layers
  - 5 hidden layers → 100 nodes each
  - Dropout rate: 0.1
  - Activation: ReLU on hidden layers, softmax on last layer
  - Loss: categorical x-entropy
    - $CE = -\sum_i^C t_i \log(f(s)_i)$
    - $f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$
  - Learning rate: 0.003
  - Batch size: 5000
  - Epochs: 50

DEEPJET FRAMEWORK

- Training and testing is done using the DeepJet Framework
  - https://github.com/DL4Jets
  - Specifically made for jet tagging
- Pure Keras + Tensorflow for training/testing
  - Includes some compiled C modules for data processing

- Class structure for modifying data structures
  - Start with a basic structure consisting of truth inputs, basic jet variables
    - 12 usable truth classes (flavors of jets)
    - b, bb, gbb, lepb, lepb_c, c, cc, gcc, ud, s, g, undefined
  - Create new data class that inherits basic structure, and add to it
- Set of base models for training supplied as python pseudo-modules, and can be further adjusted
- Plotting macros for ROC curves, performance plots

ACTION ITEMS FOR PROJECT

- Reproduce training results for base DeepCSV structure/model as found in the DeepJet framework
  - Try to determine how this can be optimized for low p_T jets
- Look at correlation between inputs and jet p_T
  - look for new inputs (see what is available)
  - Determine collection sizes
    - Remain at 6 tracks per jet, or change?
      - Look at number of tracks against p_T
      - Will most likely be on the smaller size
  - Adjusting input weights
- Adjusting layers/NN
  - Currently using all dense layers
  - Can explore different types of NN
    - Ex) introduce convolutional layers
  - Introduce p_T related regression target on top of classification?

PLACES OF INTEREST

- Lectures from the 2018 CoDaS summer school
  - https://indico.cern.ch/event/707498/timetable/
- What I started with (documentation on DeepCSV):
  - https://indico.cern.ch/event/595059/contributions/2497371/attachments/1430948/2198064/IML_2017.pdf
- ML at CERN: the IML Working Group
  - Public meetings
    - https://indico.cern.ch/category/8009/
  - EP-IT Data Science Seminars (also public)
    - https://indico.cern.ch/category/9320/
  - Forum (requires lightweight CERN account)
    - https://account.cern.ch/account/externals/
    - Anyone should be able to make one
  - https://iml.web.cern.ch/