**Credits to Amelia Binau**

**Level 1: Tests on Best of Fit Curves**

Make best-of-fir curves on momenta spectra such as gen\_lep\_pt, reco\_b\_pt, gen\_tt\_M, etc. Some possible curves to try:

* Polynomials (lines, quadratic, cubic, square root, etc.)
* Gaussian/Poisson Distributions
* Relativistic Breit-Wigner function (<https://en.wikipedia.org/wiki/Relativistic_Breit%E2%80%93Wigner_distribution>), (<https://en.wikipedia.org/wiki/Resonance_(particle_physics)>)
* Exponential/Logarithmic Decay

Also consider taking the log of the counts axis and the histogram axis (both axes) and fitting a power law to your data. These are very helpful when studying trends in experimental data because the power law will then often form a linear relationship between the two variables (<https://en.wikipedia.org/wiki/Power_law>, see Scale Invariance).

For the center of each bin in the histograms, there will be a corresponding value on the y-axis of the curve fit. For each bin center and corresponding curve-fit y-value, you will run a test of statistical significance. This is essentially a regression test that will yield various statistical parameters that will tell you how good your curve fit is vs. the original data counts. (<https://en.wikipedia.org/wiki/Chi-squared_test>).

I’ll leave it to you guys to find a Python package that has a chi-squared function.

dataset: /home/he614/phys323/minitree\_200.root

**Level 2: Response Function Analysis**

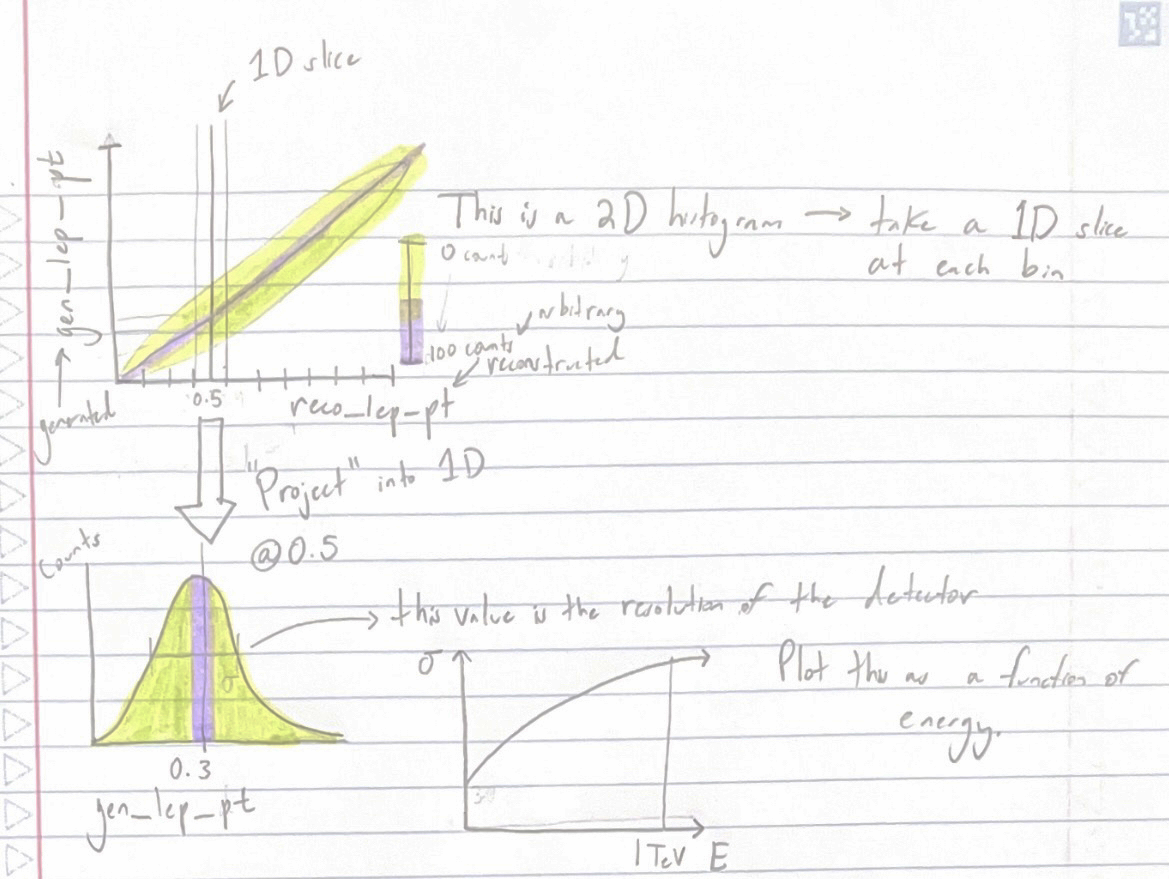


Figure 1.

You will made 2D histograms (<https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.hist2d.html>) or (https://numpy.org/doc/stable/reference/generated/numpy.histogram2d.html) of the generated and reconstructed histograms for a variable (we will start with lep\_pt, the transverse momentum of leptons). This will look like something like this:

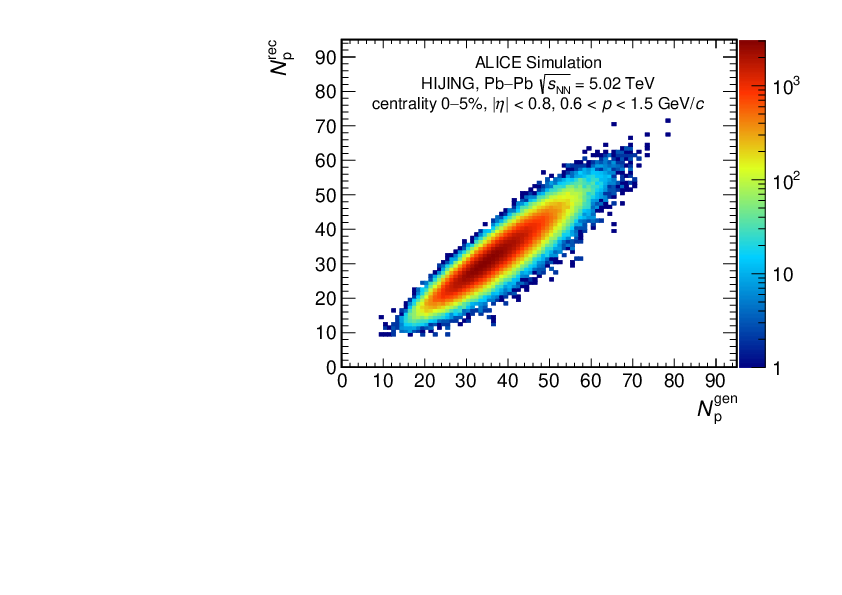


Figure 2.

But we will switch the axes (gen on y-axis and reconstructed on x-axis). As with any histogram, the x-axis here is split into bins. Selecting one of these bins and isolating it with respect to the other variable is called “slicing.” For instance, in the Figure 2 above, we could take the bin at gen=40 and slice it to get all the bins along rec that are in gen=40. We “project” this slice of the 2D histogram to make a new 1D histogram (see Figure 1). Note I accidentally reversed the color bar scale in Figure 1 for the 2D histogram; it should go from 0 to 100, not 100 to 0.

The width (full-width half-maximum <https://en.wikipedia.org/wiki/Full_width_at_half_maximum>) of the Gaussians formed in the projected 1D histograms define the energy resolution of our detector. We will call this σ. This value can be plotted as a function of energy of the beam.

dataset: /home/he614/phys323/minitree\_200.root

**Level 3:**

For students interested in HEP, the NanoAOD structure allows us to perform columnar analysis without gigantic frameworks. We can turn this into a re-discovery of Higgs Boson using CMS Open Data.

Here are additional docs of gg(qq)->Higgs->tautau->4leptons analysis with nanoaod:

<https://github.com/cms-opendata-analyses/HiggsTauTauNanoAODOutreachAnalysis>

<https://cmsdas.github.io/root-short-exercise/08-analysis-physics/index.html>

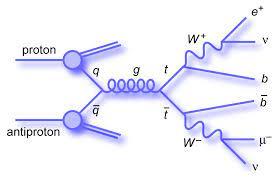
You can mimic skim.cxx using python syntaxes and a python vector module to skim the nanoaod, this will help you to understand the processes rather than getting results from a black box.

**Level 4: Separating Standard Model and SUSY Dataframes Using Neural Networks**

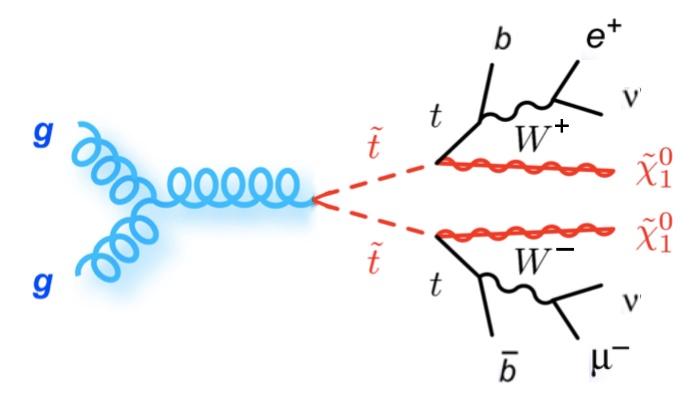
We will be using a neural network classifier (similar to those found in Lab 4) to separate a signal (contained in a SUSY minitree/dataframe) and background (Standard Model physics minitree/dataframe). See Lab 5 for more information on minitrees and ROOT.

**Background**

A pair (ttbar) decays via a weak decay channel into four particles of interest: a lepton/(anti)neutrino pair of the same flavor and an antilepton/neutrino pair of another flavor. Recall that the flavor of a particle corresponds to what generation it is a part of in the Standard Model (electron, muon, tau). A Feynman diagram of the decay channel is illustrated below.



Why is this important? A similar decay channel is hypothesized to occur in SUSY (supersymmetry) theory, which is a hypothetical expansion on the Standard Model. If we can confirm certain decays occur within the realm of supersymmetry, we could confirm that this is an accurate model for the universe, which could answer big questions in physics, including the identity of dark matter. The supersymmetric equivalent of this decay channel appears as follows:





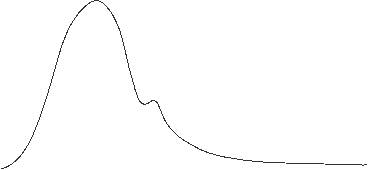
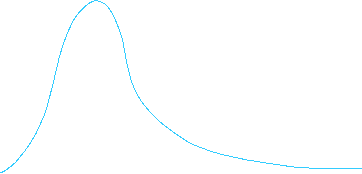
The lepton-neutrino pairs can be measured/reconstructed. Therefore, we will be using those particles to identify which of the dataframes each ttbar decay event will occur through.

**Goals**

The general goal of this project is to merge the SUSY and SM dataframes and use the 4-vector observables contained in those dataframes contain to again separate, using a neural network, those dataframes. These minitrees represent, respectively, your signal (the variables you’re trying measure) and your background (the “noise” that’s cluttering up your signal). We will plot the combined minitrees in a 2D phase-space (a 2D graph plotting each point/decay event using two kinematic variables) and the neural network will then be run on these two kinematic variables to pick apart the signal data from the background data. This method of distinguishing signal and noise data is extremely important to modern particle physics. Classically, signals (i.e. resonances/peaks) were distinguished from background by modeling the background theoretically, fitting this curve to the data, and using that to find slight bumps in the data. However, some signals are so small, that even with the best background predictions, it’s impossible to get a statistically significant separation using this technique. Therefore, using this neural network algorithm within a phase-space of kinematic variables is necessary for modern high-energy particle physics research.







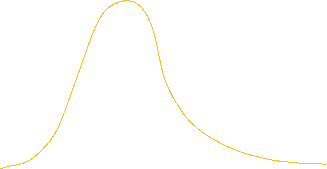




**Methods**

1. Students will obtain pre-made minitrees for the SUSY and SM data. Each of these trees will contain 4-vectors for the (electron, muon) product decay channel for a ttbar pair. These will be the two lepton-neutrino pairs we will be working with; that is, we will search for theoretical supersymmetry signal using kinematic measurements from ttbar decays that produce an electron-(anti)electron neutrino pair and an antimuon-muon neutrino pair.
2. Students will plot a mass spectrum/distribution of each minitree for these measured products. It will look something like this:









1. Students will merge the events from the two 4-vectors from each minitree. These can be plotted as a mass spectrum for practice, but the combined minitree will simply appear as a skewed normal distribution with more counts than the SM distribution.
2. Now, how do we use the neural network to separate the SM and SUSY data? Students will choose two kinematic variables from the 4-vectors. These two variables will be the same for SM and SUSY. The four variables in these vectors are as follows: pt (transverse momentum), phi (pseudorapidity), eta (rapidity), and M (mass). Ask your TA for clarification on what any of these physical observables represent.
3. All combined events from the SM and SUSY data will then be plotted in a scatter plot of the two kinematic observables (this is what we will refer to as the 2D phase-space). Students will then use a PyTorch neural network classifier to draw a line that separates data into two groups: which data it believes is in the SUSY dataframe and which are in the SM dataframe. That is, which data are signal and which are background. This may be the most complex part of the research project, as more elaboration as to the inner-working of this neural network will be required, but this portion of the project will produce conclusive results on the effectiveness of this classifier to separate the signal and background, as well as where the signal and background are located.















Phase space where x and y are the two chosen kinematic variables.

A few things to note:

1. The combined SUSY and the SM data will look almost identical by eye to the SM data alone, but the neural network should be able to find the signal.
2. There are fewer events in the SUSY dataframe than the SM array.
3. SUSY top quark mass will vary from 125 to 225 GeV.
4. The group who has already taken PHYS 323 will be using spin correlation, the angle between the lepton and antilepton, to separate the two dataframes, not kinematic observables.

The full paper describing neural networks for supersymmetric top quark (squark or stop) decay is located here: <https://cds.cern.ch/record/2813262/files/FTR-18-034-pas.pdf>

The datasets are located in Ling’s scholar workspace:

/home/he614/phys324/minitree\_total\_SM.root

/home/he614/phys324/minitree\_total\_SUSY.root

Note that this project requires special techniques to handle inconsistent definitions (ex. l\_pt and sl\_pt are leading and subleading lepton transverse momentum while lep\_pt and alep\_pt stand for lepton and anti-lepton pt. The SUSY file hasn’t gone through top reconstruction due to software dependency issues.

**Example Poster from the Past** (The poster file is also in the github repository)**:**

