MVA For 2021 SIMP Search

HPS Collaboration Meeting 6/4/25
Rory O'Dwyer

Table of Contents

Introduction

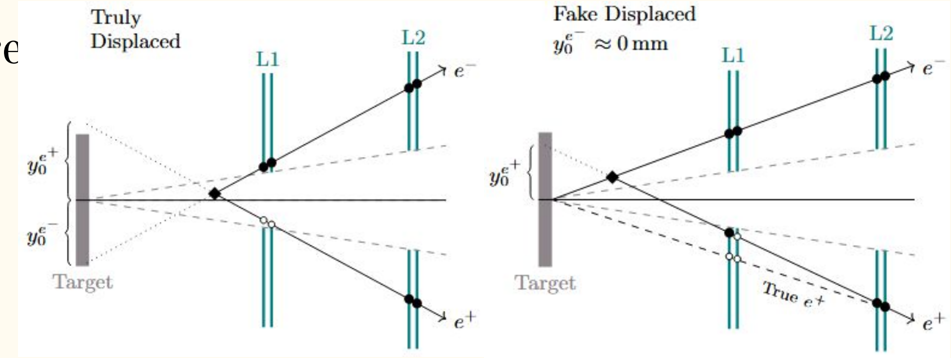
Introduction: Adversarial NN and SIMP analysis

Alic and Tom's l1l1 and l1l2 studies were based on intuitionistic cuts.

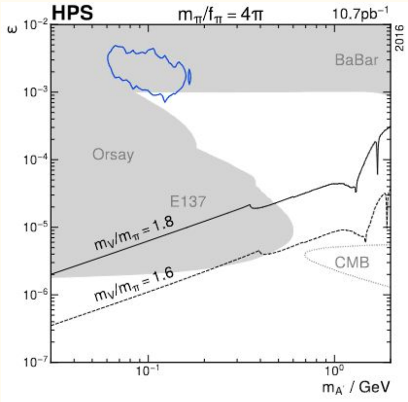
A number of these were considered; among them the displaced z as well as y_{0} significance (which is better at finding false displacements).

These pushed us into new exclusion space; we hope to obtain even further improvements with the following two additions:

- 6-7x more EOT at a different energy.
- MVA Techniques, to push past intuition based cuts.



y0 parameter and sensitivity to fake displacements.



Reach curve for 2016 SIMP study.

Introduction: Adversarial NN and SIMP analysis

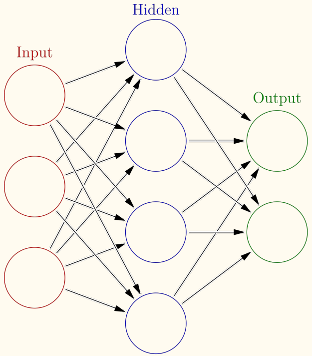
An adversarial NN is trained on two classification tasks:

- Signal vs Background
- Control vs Non-Control Region

It computes a reward function from its classification error for both, but subtracts CNCR from SNB.

It performs two consecutive backpropagation stages for each.

You arrive at a network with high SNB but incapable of distinguishing the control from the Non-Control region.



NN image from wikipedia.

$$P_{\text{sum}} = |\vec{p}_{e^-}| + |\vec{p}_{e^+}|$$

Trident prod rate CR	1.8 GeV<Psum<2.4 GeV
SIMP signal region	1.0 GeV<Psum<1.9GeV

CR chosen for 2016 based on presence or lack of signal*

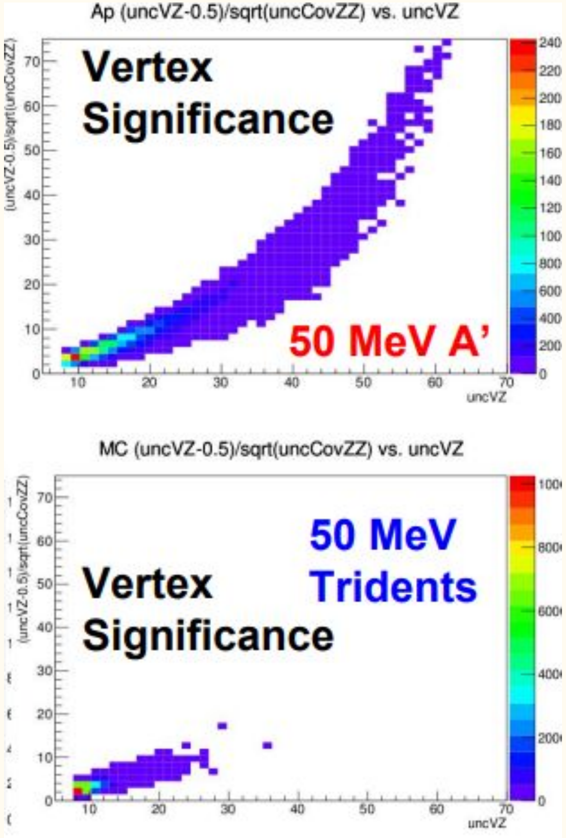
Matt Solt’s MVA analysis

Matt Solt, in this analysis, use vx,vy,vz,d0, and some significances in a NN MVA.

He found significant improvements in discriminating power; a 70% improvement in signal retention at equivalent high z rates.

He also looked at high z mis reconstruction; didn’t express numeric results but I’ve included his plots here.

Other work I’ve surveyed, or been directed to, seems still intuition cut based (Alic, Tom, Holly, etc.)



Vertex Significance vs. vertex position in Z for signal (Top) and background (Bottom)

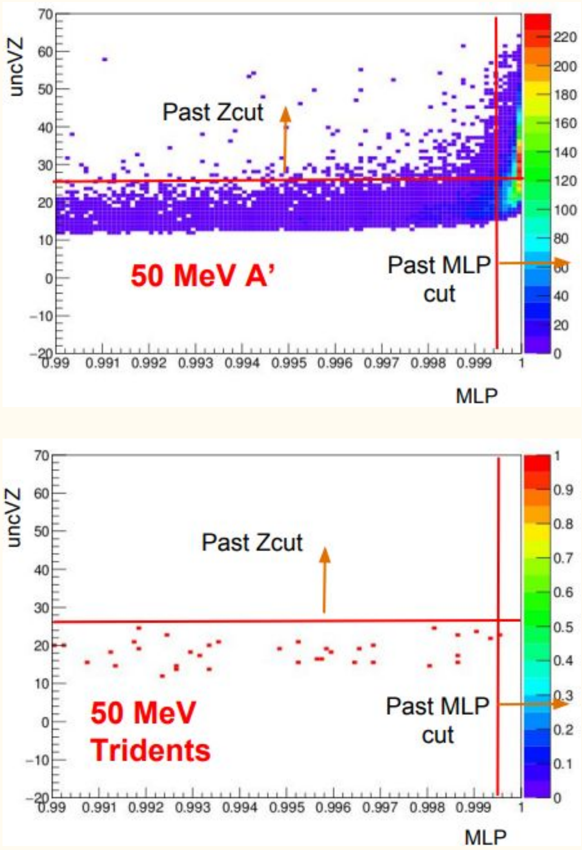
Matt Solt's MVA analysis

Matt Solt, in this analysis, use vx,vy,vz,d0, and some significances in a NN MVA.

He found significant improvements in discriminating power; a 70% improvement in signal retention at equivalent high z rates.

He also looked at high z mis reconstruction; didn't express numeric results but I've included his plots here.

Other work I've surveyed, or been directed to, seems still intuition cut based (Alic, Tom, Holly, etc.)



Plot of displaced vertex points vs. MVA score for signal (top) and background (bottom).

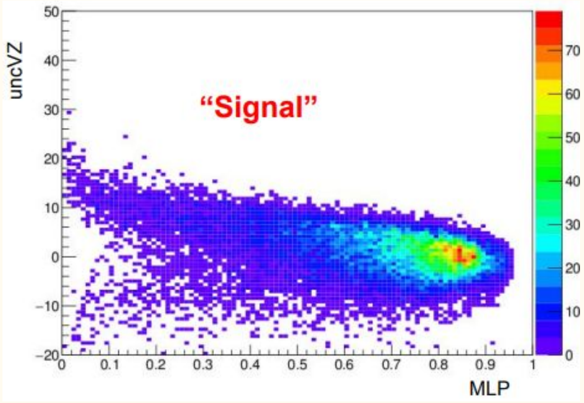
Matt Solt's MVA analysis

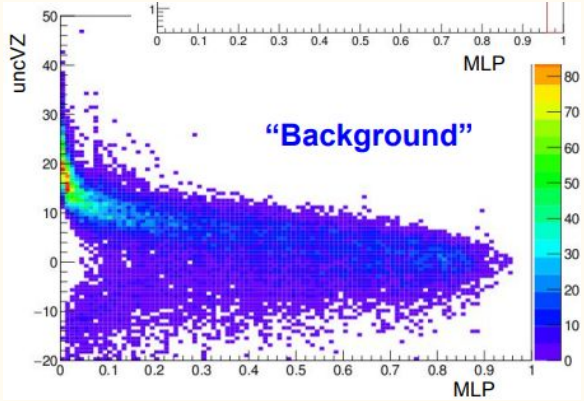
Matt Solt, in this analysis, use vx,vy,vz,d0, and some significances in a NN MVA.

He found significant improvements in discriminating power; a 70% improvement in signal retention vs existing intuition based cuts at the same level of background rejection.

He also looked at high z mis reconstruction; didn't express numeric results but I've included his plots here.

Other work I've surveyed, or been directed to, seems still intuition cut based (Alic, Tom, Holly, etc.)





Plot of displaced vertex points vs. MVA score for well reconstructed tracks (top) and scattering (bottom)

NN Sample Distributions for 30,60,90,120 MeV SIMP and Tridents.

Generated Sampled Used

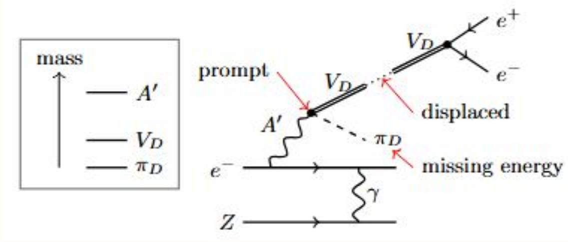
The principle background to differentiate from SIMPs are tridents.

Sarah generated 4 samples at pass 6 with the v_{d} at different values and other quantities set at constant default *.

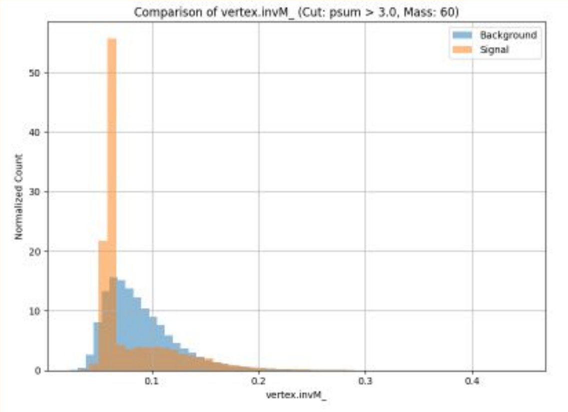
We used the v6 2021 detector alignment for these studies.

I reran reconstruction on each (and on a larger trident file) to include kinks and residuals.

The sample sizes are ~10000k passing preselection for SIMPs and tridents each.



Feynman diagram for simps, with mass hierarchy included



Reconstructed invariant for tridents (blue) and a 60 MeV SIMP signal in orange.

What do the old intuition based cuts look like?

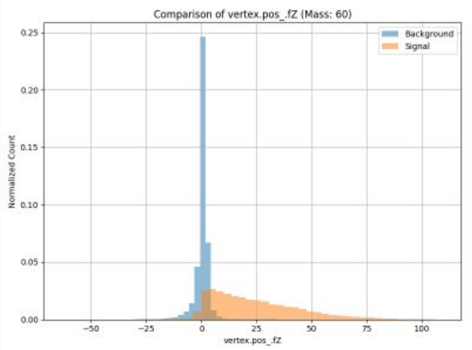
As a cross check on our NN, we should look at the old intuition based cuts.

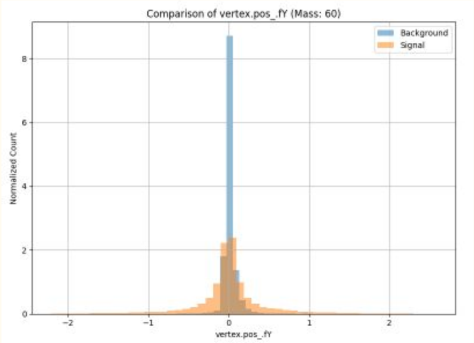
The vertex z and y0 distributions look quite discriminating, which is a good sign.

Most variables which by intuition should be different (nhits on track, etc.) are, and those which don't (chi2 of track) don't.

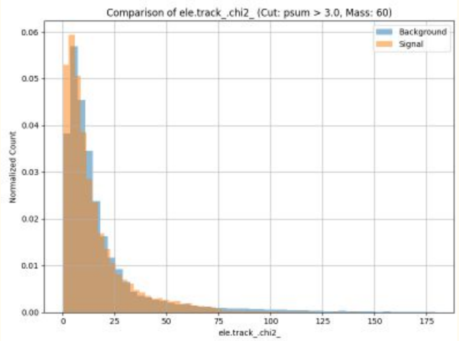
There is one variable that I don't quite believe; the track time for mass not equal to 60 is visibly displaced

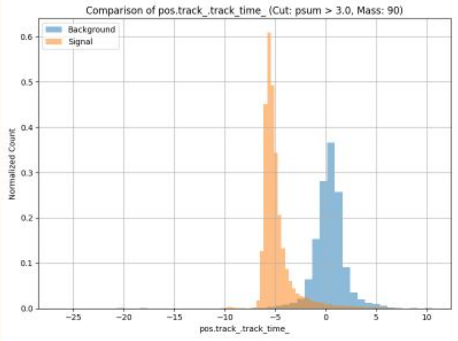
I wasn't sure if this was physical so its not included.





Simp and Trident distributions z0 left and y0 right





Simp and Trident chi left and track time right

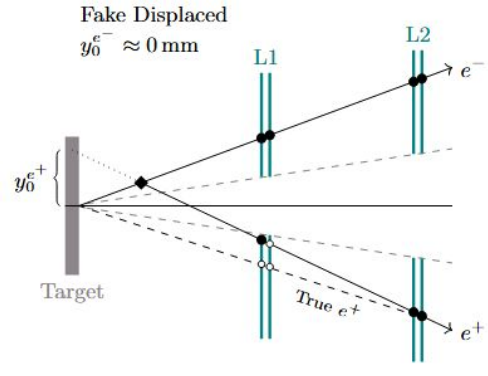
What do the kinks and residuals look like?

The HPS 2016 test focused somewhat on differentiating true from misconstructed fake displaced vertices.

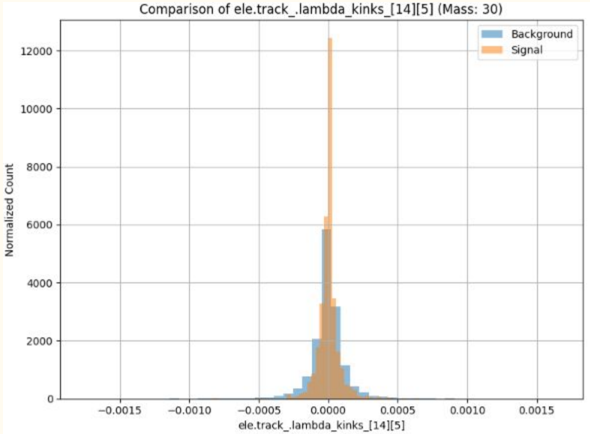
This occur due to secondary scattering in a given layer; motivated including kinks to establish whether secondary scattering occured.

The distribution of kinks, however, seems to demonstrate little differentiating power alone.

The best shot it has is if its correlations are highly differentiating; I have not found evidence of this in any of my NNs yet.



Fake displaced due to scattering in the first layer.



Lambda kink distribution for layer 7 between tridents and Simps. All distributions can be found here.

Details about NN Implementation (and First Psum Results)

Details about NN Model Implementation

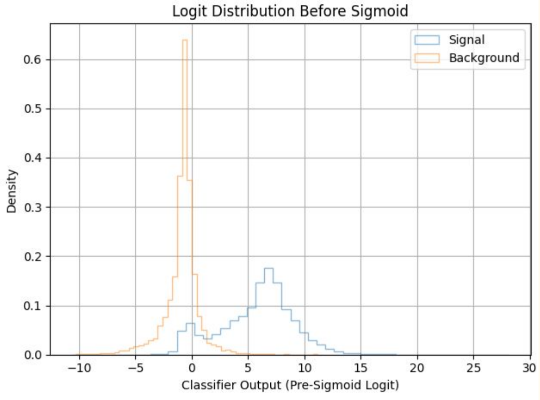
The neural network uses torch,sklearn, and uproot to build two 1 layer NN with 64 (classifier) and 32 (adversary) hidden nodes.

It labels data with a signal and background label, and 5 labels corresponding to a Psum bin. We will also consider performance with the labels corresponding to invM bins.

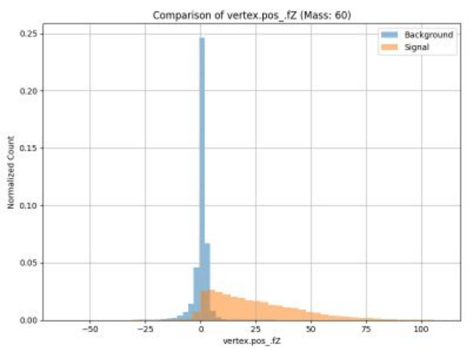
It splits data that lies 5 MeV from signal mass into test and train (30% test).

It runs these over 1000 epochs (to stabilize performance), running over the 3 characteristic steps of an adversarial network on each epoch.

It uses all variables used contained here (kinks weren't used for plots I will show).



Plot of logits prior to application of sigmoid (i.e. result of NN linear transformations on data coordinates). For mass=60, you can see that we have appreciable discrimination.



Original discriminating z variable for reference

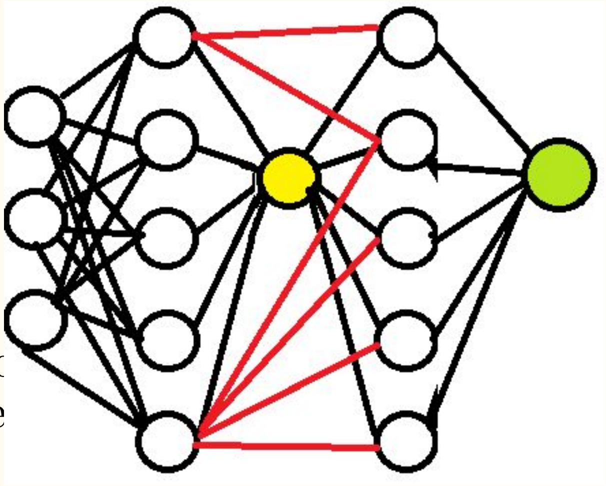
How the Adversarial Network Works

Two NN, the classifier and adversary, compete on data.

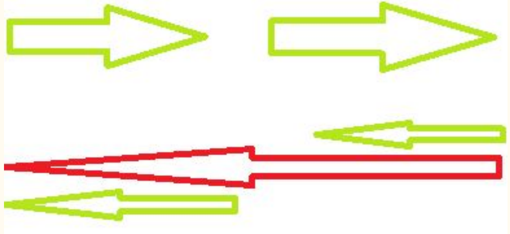
Data is input into C and then an intermediate state is fed through and adversarial network.

The gradient is reversed on backpropagatio through to the classifier, penalizing it some weighting factor if the adversary trained well.

Ultimately, if the adversary has poor classifying power, it means the classifying network made the same decisions regardless of the regions it draw data from.



The adversarial network (green) can either feed in internal features (red lines) or just the decision (yellow). Each will make the behavior invariant of region but the former is better at it.



Feedforward through both, adversarial pos. grad on itself but negative down to classifier

Adversarial Output (with 2 caveats)

I show the performance of the classifier prior to v1 draft (running takes a while).

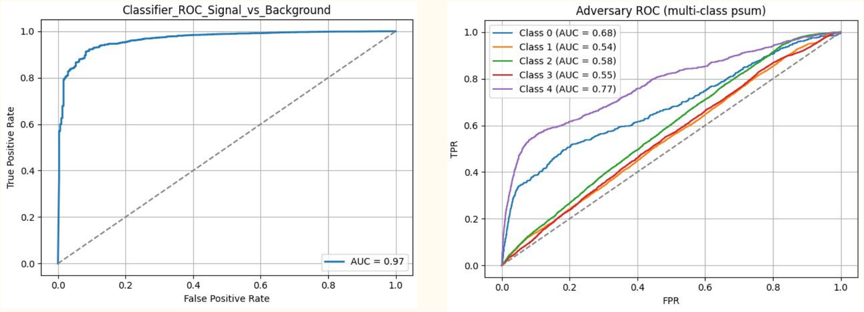
The caveats are these:

- default values/track times are included
- Adversarial network is leaky; could be improved for some p regions OR we could just fix the Psum cut.

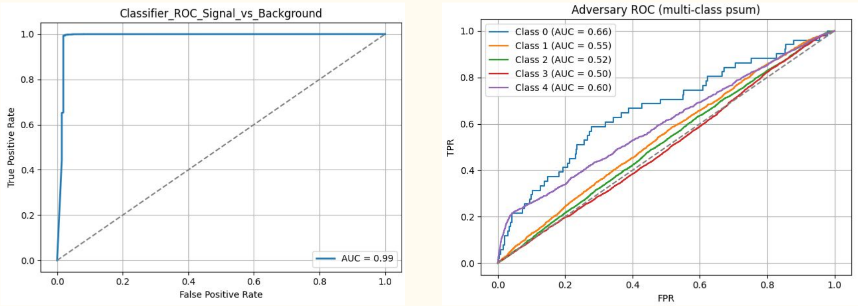
Mass 60 has no track time disparity, its AUC is 97 and seems physical. 120 is so high because of track time disparity (next page)

Default values, upon investigation, should only hurt performance. Performance is really good for mass 60 outside of adversarial considerations.

The adversarial network can be improved if we want to make it completely Psum invariant.



ROC for classifier (L) and adversarial (R) networks at 60 MeV where track time is weak



ROC for classifier (L) and adversarial (R) networks at 120 MeV where track time is strong

All Masses with Track Time Removed.

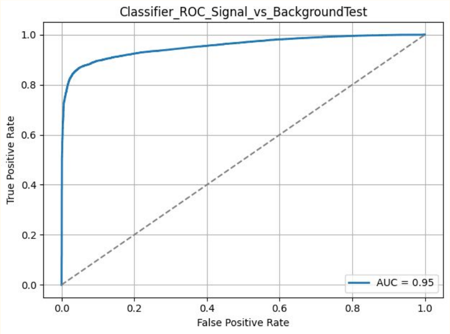
With the track time removed, the adversarial networks still seems to leaking in 60 mass, but we saw a more typical ROC curve for 120 and much less leaking leaking (not 1.0 AUC).

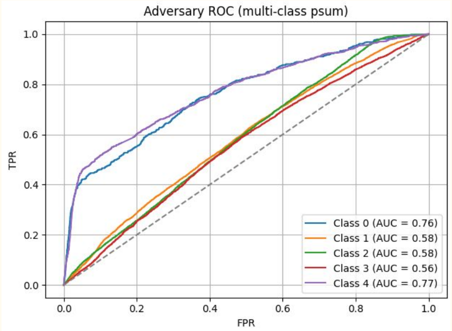
I have spent much time vetting this (using split samples to remove overfitting, plotting the presigmoid distributions, etc).

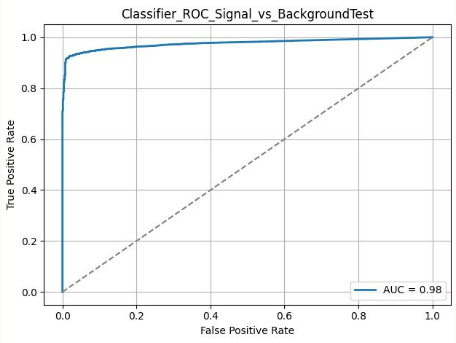
I believe beyond making the adversarial network more insensitive for some regions. this is an accurate expectation for performance.

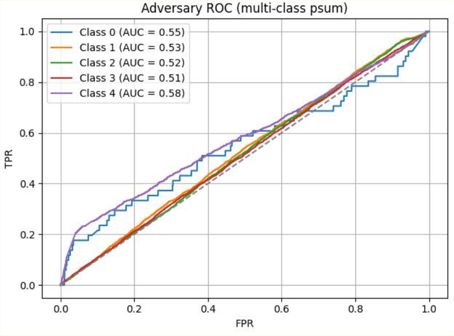
I don’t think novel ways of treating the default -1000 and -999 values will significantly affect performance

Kinks and residuals do not seem to make any statistically significant differences in classifier performance upon introduction; at least not until we consider extremities.









ROC for classifier (L) and adversarial (R) networks at 120 MeV where track time is weak

All Masses with Track Time Removed.

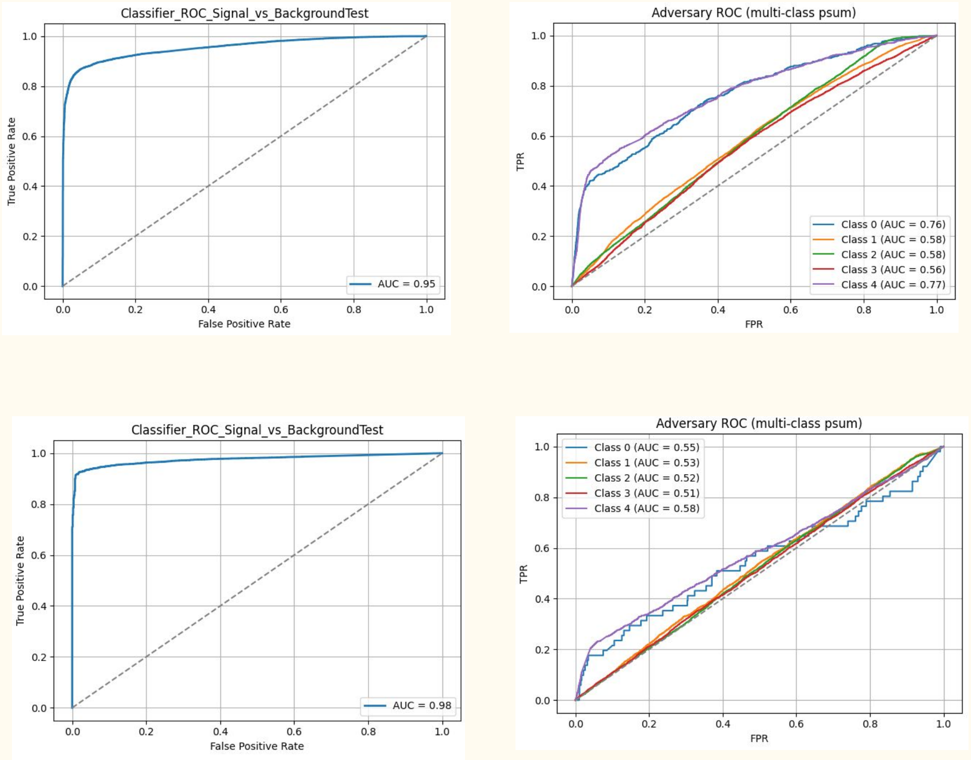
With the track time removed, the adversarial networks still seems to leaking in 60 mass, but we saw a more typical ROC curve for 120 and much less leaking leaking (not 1.0 AUC).

I have spent much time vetting this (using split samples to remove overfitting, plotting the presigmoid distributions, etc).

I believe beyond making the adversarial network more insensitive for some regions. this is an accurate expectation for performance.

I don't think novel ways of treating the default -1000 and -999 values will significantly affect performance

Kinks and residuals do not seem to make any statistically significant differences in classifier performance upon introduction; at least not until we consider extremities.



ROC for classifier (L) and adversarial (R) networks at 60 MeV (T) and 120 MeV (B) where track time is weak

Adversarial Network vs. Mass and Extremities.

Mass Invariant Network and Extremities

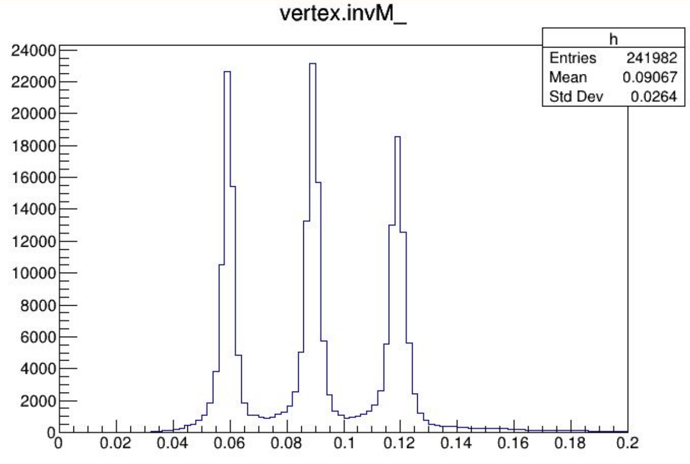
The last draft MVA had to main comments:

1. Making discriminator psum invariant is not really that useful in the analysis; why not make it invariant w.r.t. Mass?
2. We really only care about extremities, why not evaluate it on falsely displaced background?

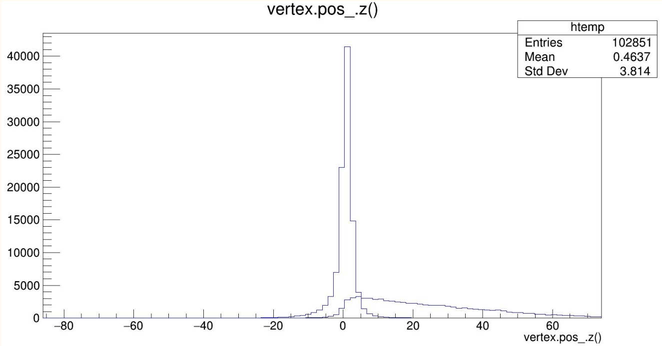
To address this, I have made a sample combining all masses for the first point.

I will demonstrate what selection power we have independent of the mass.

For the extremities, I will work both with mass independence AND evaluate performance for vertex.pos.fZ>10 (significantly displaced vertices)



invM_ profile for signal sample; we really need no leakage here.



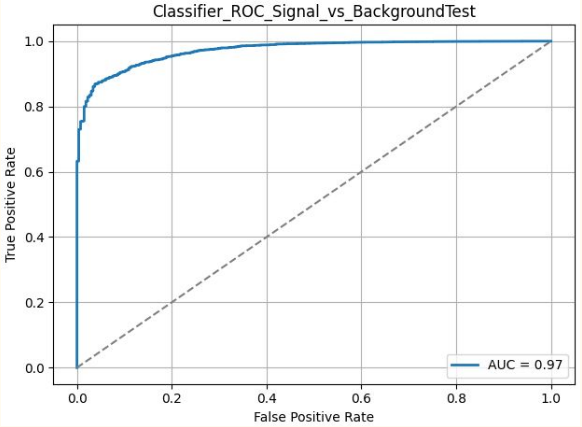
Vertex z position; extremities defined for >10

Mass 60 Network On Extremities

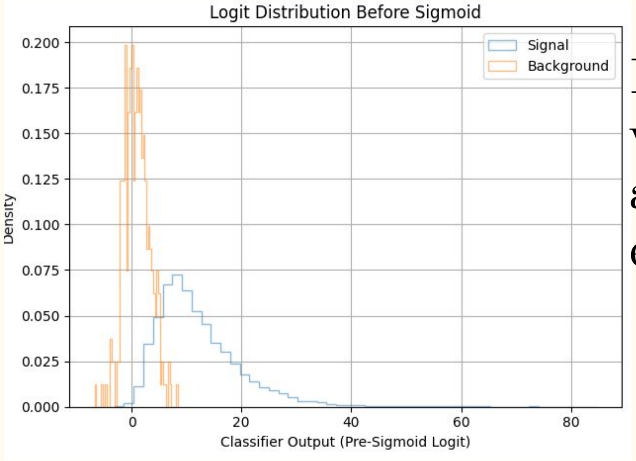
I can focus on a single mass (60 MeV in this case) and train on extremities.

The mass invariance is not trained on; we get an AUC of .97 after 200 epochs.

When we train including kinks and residuals (~42 extra variables) our AUC goes to .95; this is almost certainly because at 200 epochs we haven't settled at a minimum yet.



ROC Curve for 60 MeV on extremitie s



Logit value for after 200 epochs

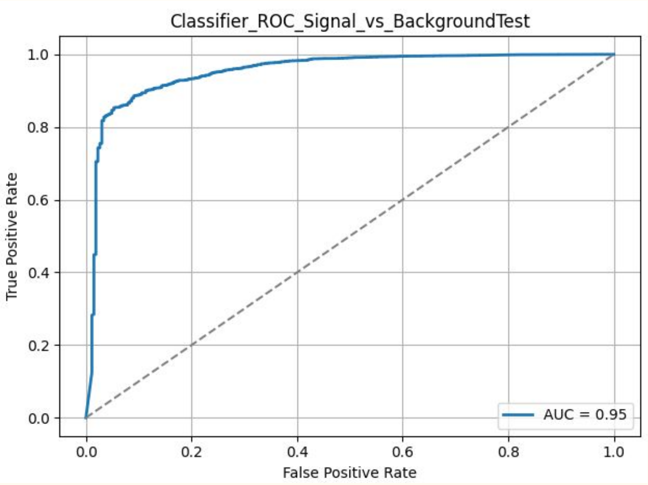
Mass 60 Network On Extremities

I can focus on a single mass (60 MeV in this case) and train on extremities.

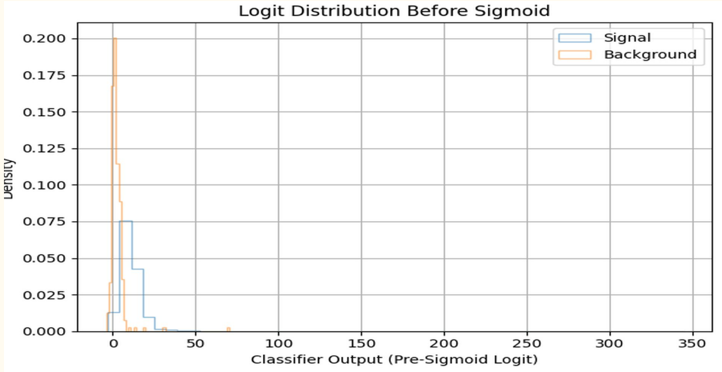
The mass invariance is not trained on; we get an AUC of .97 after 200 epochs.

When we train including kinks and residuals (~42 extra variables) our AUC goes to .95; this is almost certainly because at 200 epochs we haven’t settled at a minimum yet.

I proceeded to run it over night on 1000 epochs (it takes that long with kinks). Still does not outperform original discriminator (which means it has a really complicated parameter space).



ROC Curve with kinks included (200 epochs)



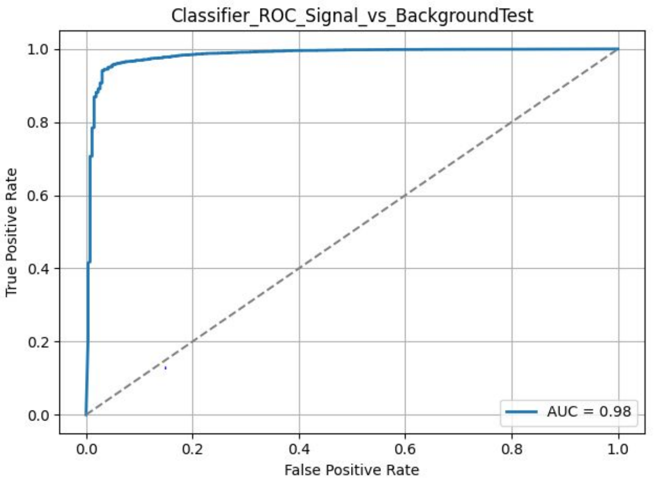
Mass 90 Network On Extremities

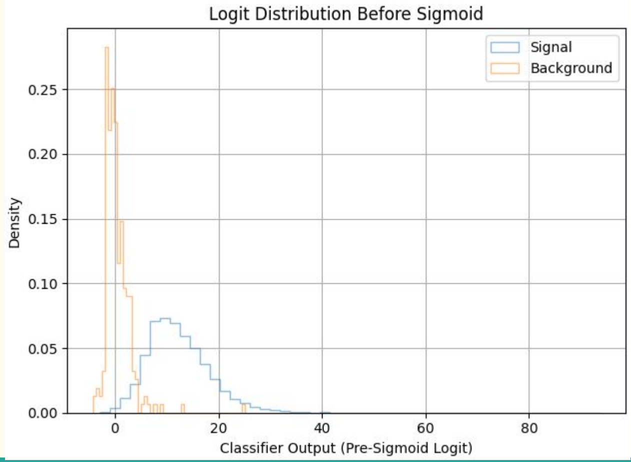
At 200 epochs, we already reach .98 AUC even on these extreme events.

In the next section, we will compare these results to the intuition based cuts of 2016 (reformulated for 2021).

It will become clear that MVA can significantly outperform them, though kinks seem to be captured in other existing information.

For instance, the kink Matt Solt used could very well have its information stored in the 2016 min(y) metric.



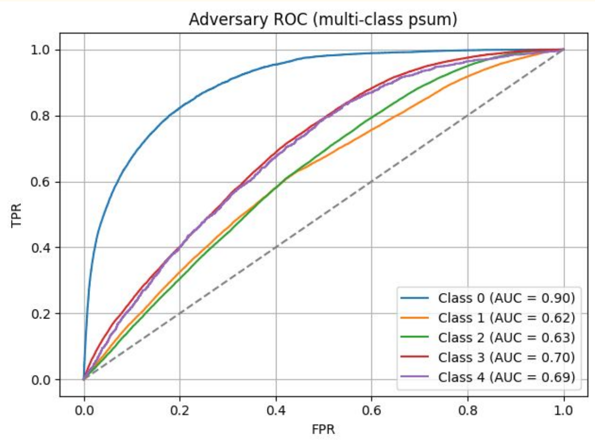


ROC Curve for extremities at 90 (120 does even better)

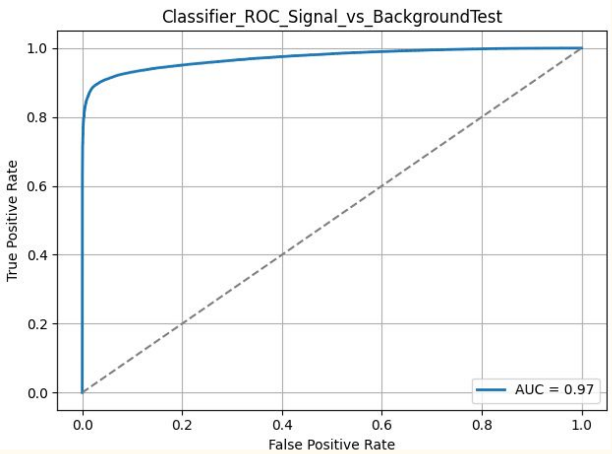
Mass Invariant Network (Not On Extremities)*

For the mass invariant network (which used the mixed sample), I seem to obtain alot of leakage.

I ran with the weight devoted to decrementing the adversarial score almost equivalent to the classifier, and only saw marginal improvements to leakage.



Adversarial ROC curves; indicative of high leakage



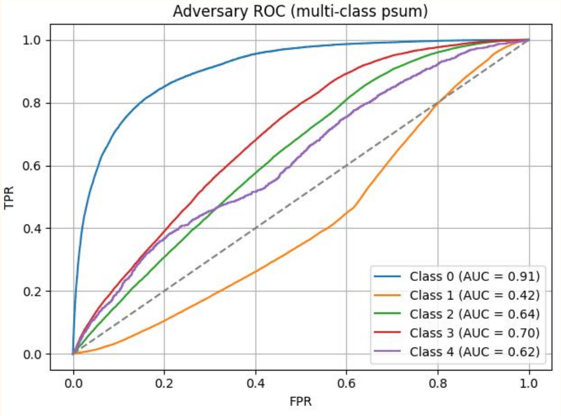
ROC curve for leaky network.

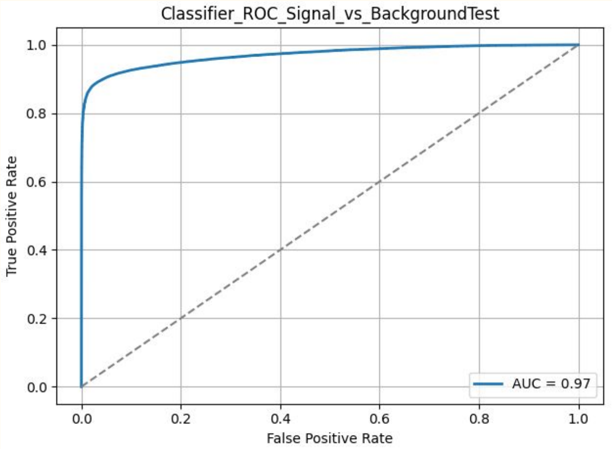
Mass Invariant Network (Not On Extremities)

For the mass invariant network (which used the mixed sample), I seem to obtain alot of leakage.

I ran with the weight devoted to decrementing the adversarial score almost equivalent to the classifier, and only saw marginal improvements to leakage.

I then set it to ~50 times the weight associated with the classifier score; I obtain these values. I believe I can do better with larger epochs, but I did not have time before this meeting to try.





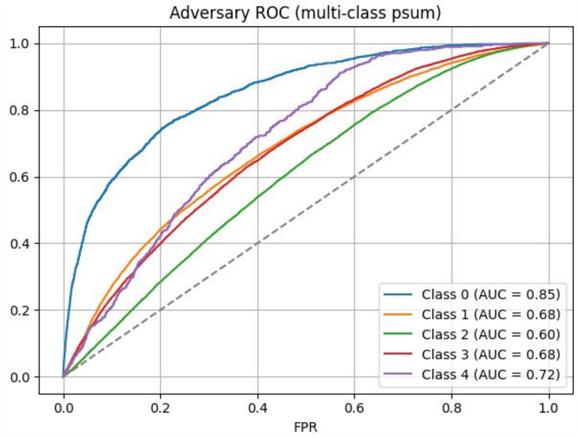
Better wrt leakage, but still quite leaky.

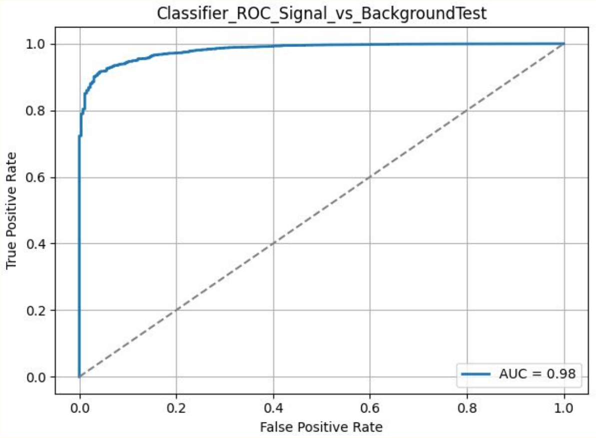
Mass Invariant Network (On Extremities)

For the mass invariant network (which used the mixed sample), I seem to obtain alot of leakage.

I ran with the weight devoted to decrementing the adversarial score almost equivalent to the classifier, and only saw marginal improvements to leakage.

I then set it to ~50 times the weight associated with the classifier score; I obtain these values. I believe I can do better with larger epochs, but I did not have time before this meeting to try.





Comparison to Intuition Based Cuts

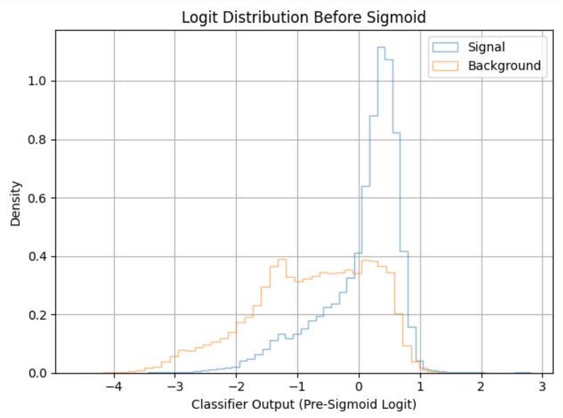
The ROC Curve for min(|y_{0}^{-}|,|y_{0}^{+}|)

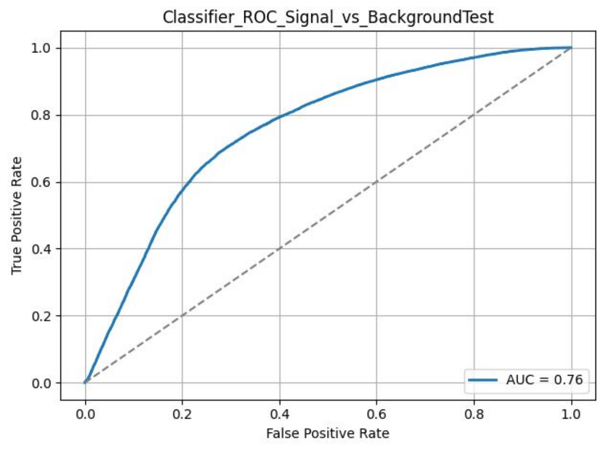
The main cut employed by the 2016 l1l1 and l1l2 search is $\min(|y_0^+|,|y_0^-|)$, the minimum of the projected y_0 among the electron and positron.

This detects displaced vertices, but has the extra effect of detecting scattering in the first layer (inducing false vertices).

Here is a discriminator based only on these two variables, for both our regular discrimination task and extremities.

You can see that our previously shown MVAs dramatically out perform the selecting power of this discriminator.





Discriminating power when training only on y_0^+ and y_0^- for 60 MeV.

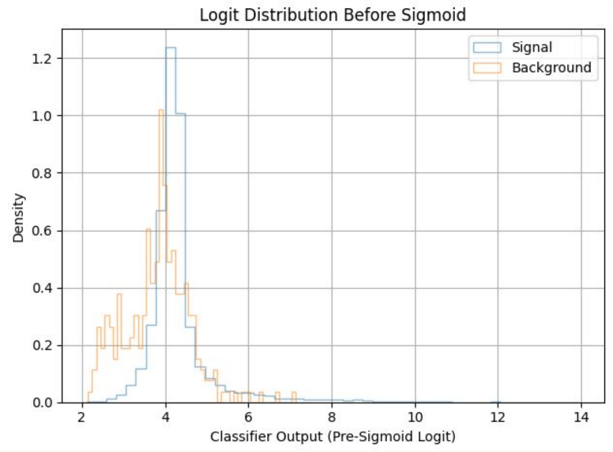
The ROC on intuition cut on extremities.

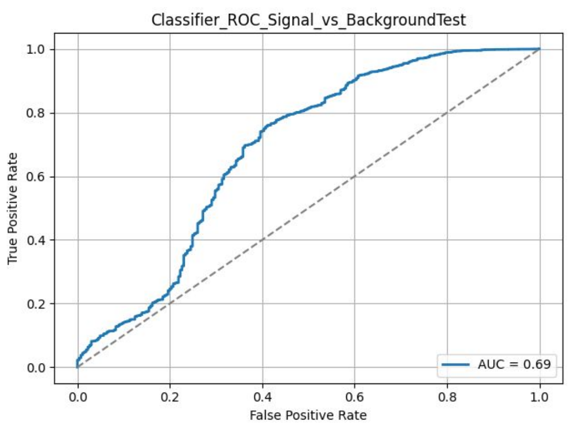
The main cut employed by the 2016 l1l1 and l1l2 search is min($|y_0^+|$,$|y_0^-|$), the minimum of the projected y_0 among the electron and positron.

This detects displaced vertices, but has the extra effect of detecting scattering in the first layer (inducing false vertices).

Here is a discriminator based only on these two variables, for both our regular discrimination task and extremities.

You can see that our previously shown MVAs dramatically out perform the selecting power of this discriminator.





Discriminating power when training only on y_0^+ and y_0^- for 60 MeV.

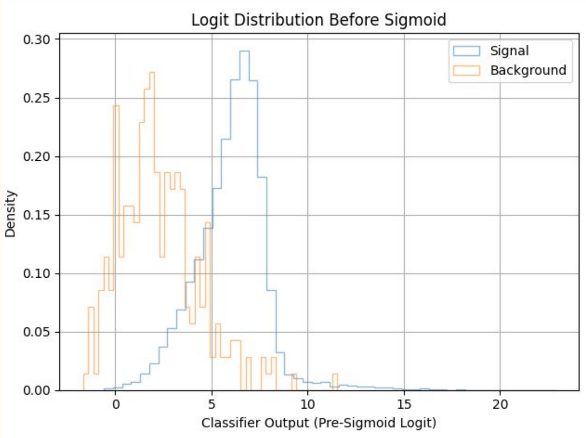
The ROC on intuition cut on extremities.

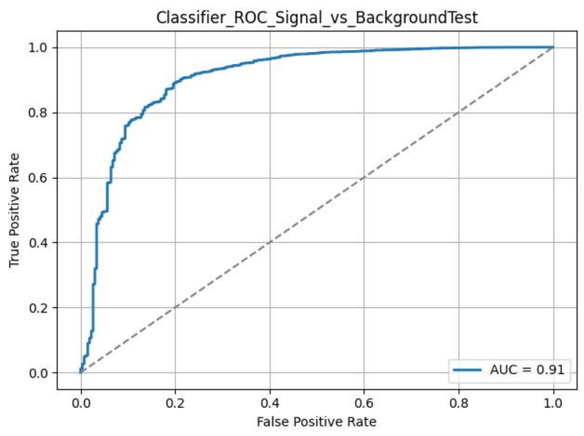
The main cut employed by the 2016 l1l1 and l1l2 search is min($|y_0^+|$,$|y_0^-|$), the minimum of the projected y_0 among the electron and positron.

This detects displaced vertices, but has the extra effect of detecting scattering in the first layer (inducing false vertices).

Here is a discriminator based only on these two variables, for both our regular discrimination task and extremities.

You can see that our previously shown MVAs dramatically out perform the selecting power of this discriminator.





Discriminating power when training only on y_0^+ and y_0^- for 90 MeV.

Conclusion

The old pSum based adversarial network showed leakages for low masses; this leakage problem continued to the mass invariant adversarial network.

The problem improves somewhat with higher weight given to the adversarial; I would like to run with higher epoch number to be sure.

Going to the non-leaky case: I have better comparisons for performance increases.

For 60 MeV, the ROC for a discriminator based on the intuition cut alone is .69 compared to .97 for same epoch number.

Cut for time: I was going to directly apply to Matt Solt's algorithm to show that the kink could supply a performance increase: presumably it's just captured by other variables.