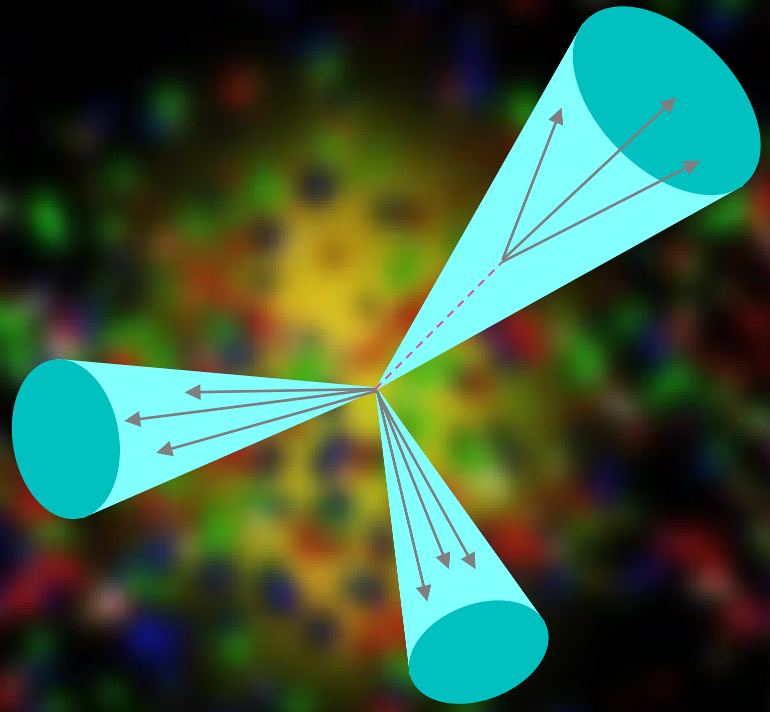
FEASIBILITY OF TAGGING HEAVY FLAVOR JETS AT RHIC WITH ML

GEORGE HALAL (LEHIGH)
HELEN CAINES (YALE), ROSI REED (LEHIGH)
JETSCAPE WORKSHOP – JAN. 13, 2019

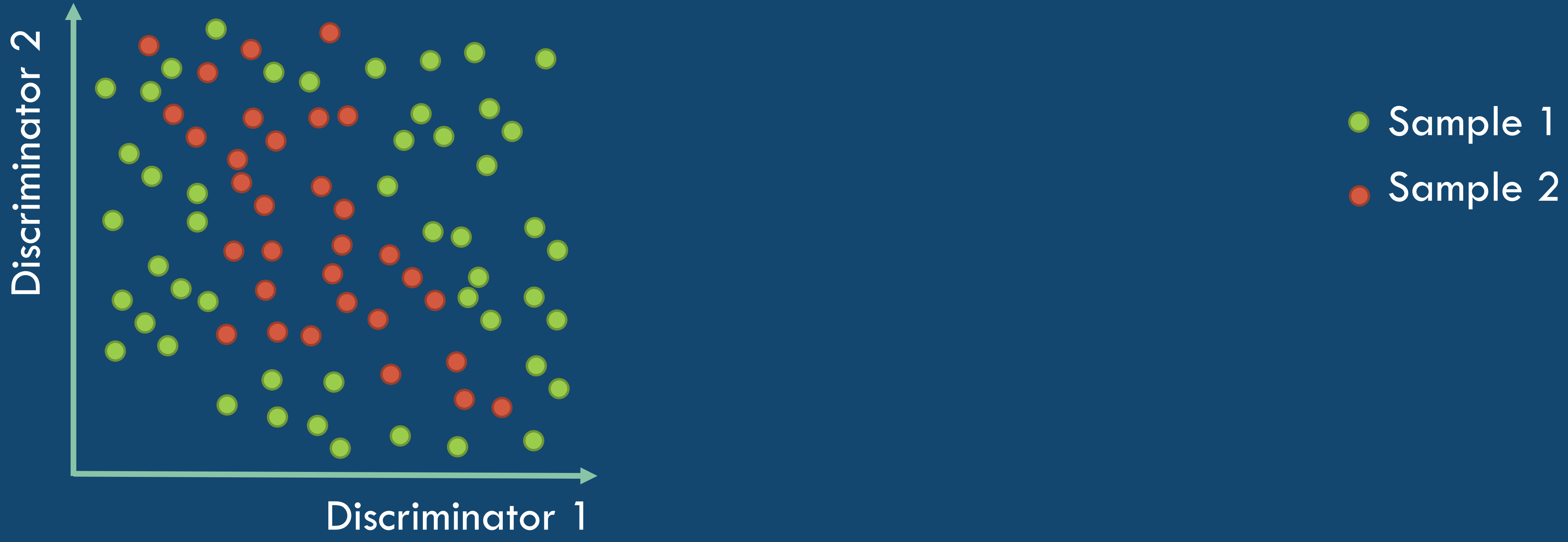
IMPORTANCE OF JET FLAVOR TAGGING



Heavy flavor jets interact differently with the QGP

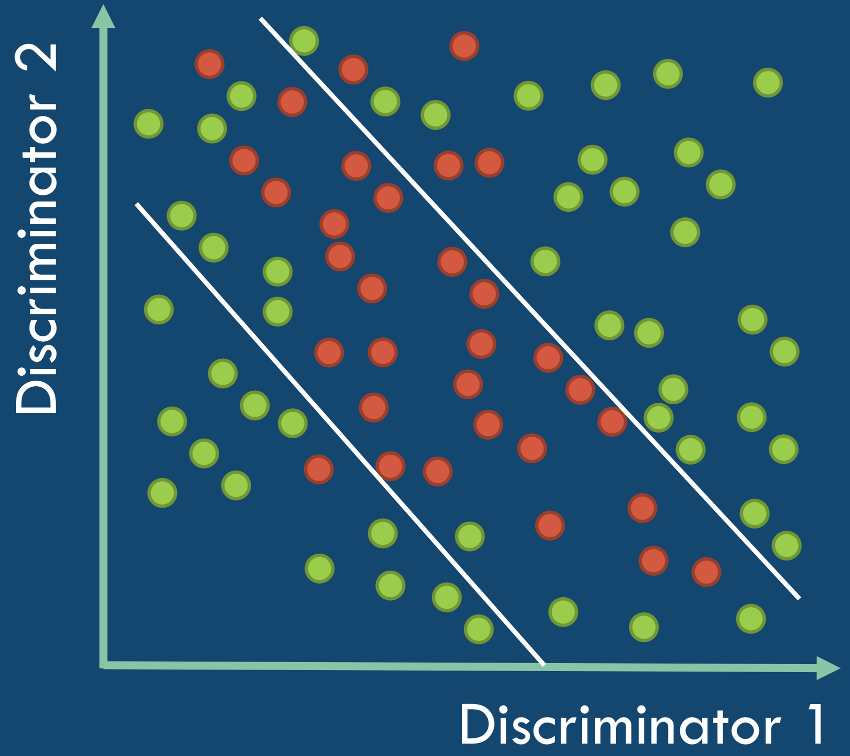
Different levels of collisional & radiative energy losses

CLASSIFICATION



CLASSIFICATION

GEOMETRIC CUTS



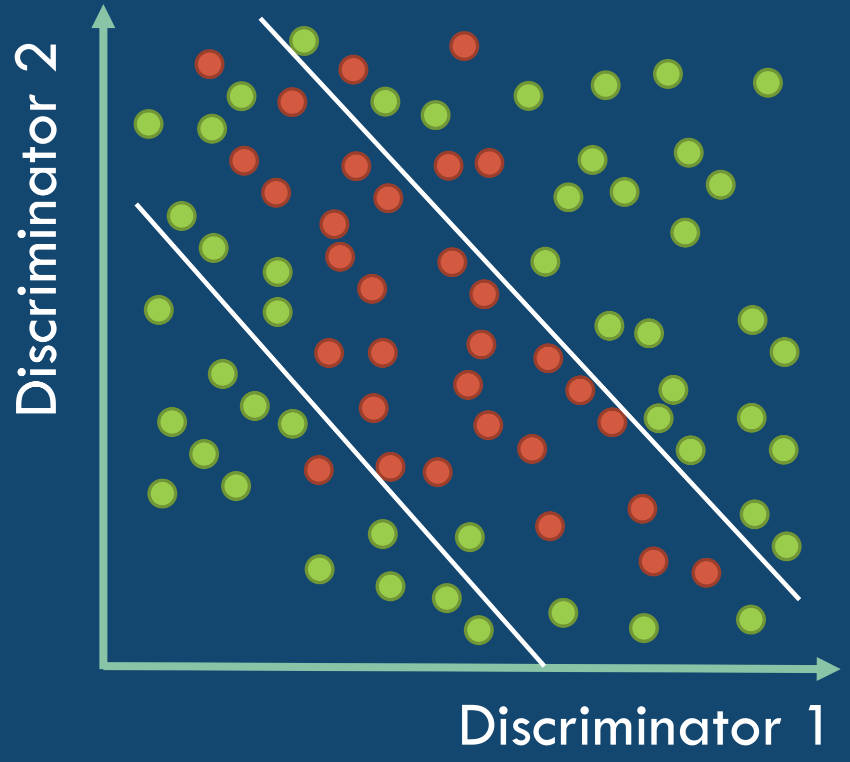
$$Efficiency = \frac{\#Sample\ 1\ in\ selected\ region}{Total\ \#Sample\ 1}$$

$$Impurity = \frac{\#Sample\ 2\ in\ selected\ region}{Total\ in\ selected\ region}$$

- Sample 1
- Sample 2

CLASSIFICATION

GEOMETRIC CUTS



MACHINE LEARNING

- Form of artificial intelligence
- Smaller errors and higher computational efficiency
- Used for image and speech recognition, prediction, etc.

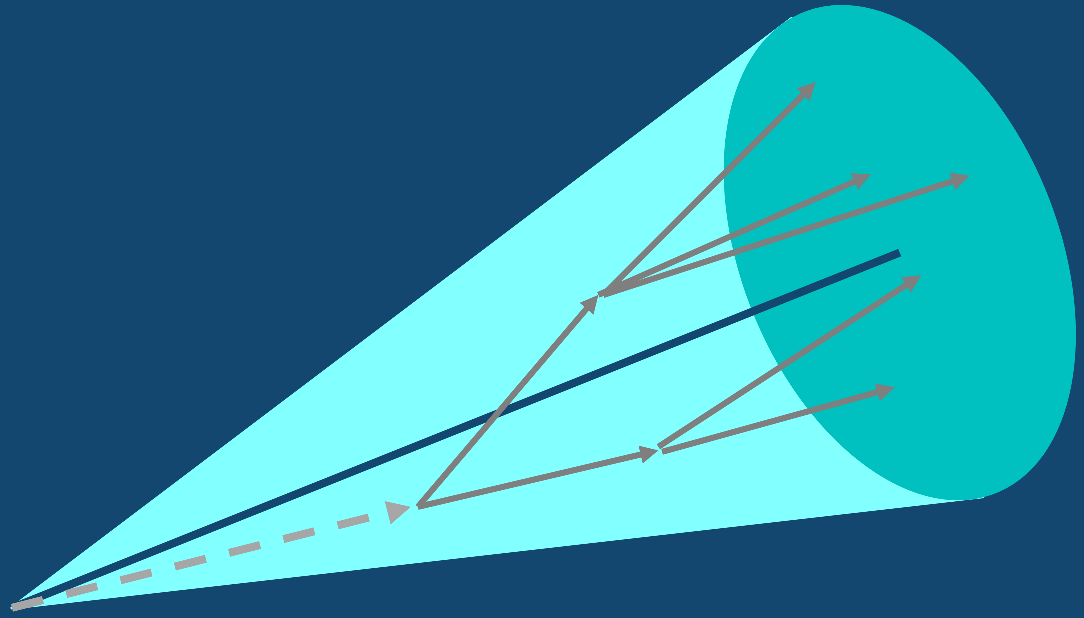
MONTE CARLO SIMULATION PARAMETERS

PYTHIA8 + FastJet:

- p+p: $\sqrt{s}$ = 200 GeV
- $|\eta| < 1.0$ (STAR)
- Full jets (charged + neutral)
- Anti-k_t R = 0.4
- Jet p_T > 10.0 GeV/c
- light jets : HardQCD:all = on
  charm jets : HardQCD:hardccbar = on, Charmonium:all = on
  bottom jets : HardQCD:hardbbbar = on, Bottomonium:all = on

JET-PARTON MATCHING

- Method 1: Trace back the decay cascades of the jet constituents to original hadron
- Method 2: $\sqrt{(\phi^{Jet} - \phi^{Parton})^2 + (\eta^{Jet} - \eta^{Parton})^2} < 0.3$ requirement from a list of $Partons$ resulting from the initial hard scattering

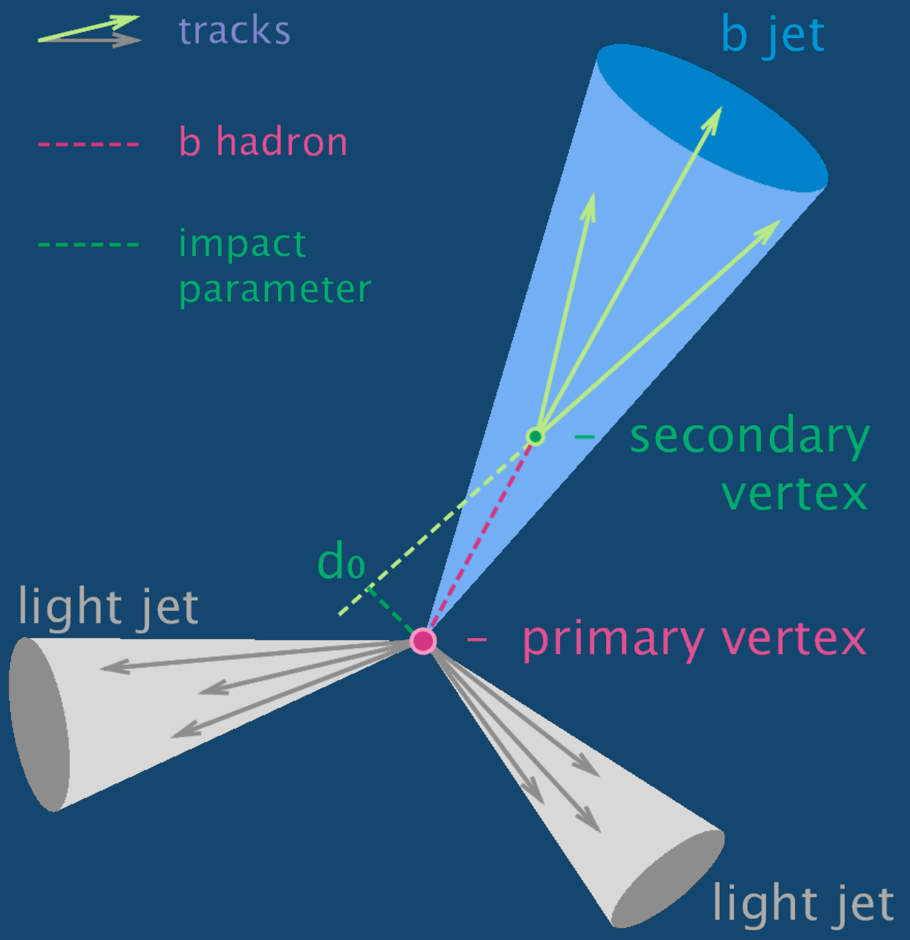


JET-PARTON MATCHING

- Method 1: Trace back the decay cascades of the jet constituents to original hadron
- Method 2: $\sqrt{(\phi^{Jet} - \phi^{Parton})^2 + (\eta^{Jet} - \eta^{Parton})^2} < 0.3$ requirement from a list of $Partons$ resulting from the initial hard scattering
- If jet is matched to more than one parton → use heaviest parton
- Throw away jets that aren't matched
- Only consider jets for which the 2 methods agree

DISCRIMINATORS

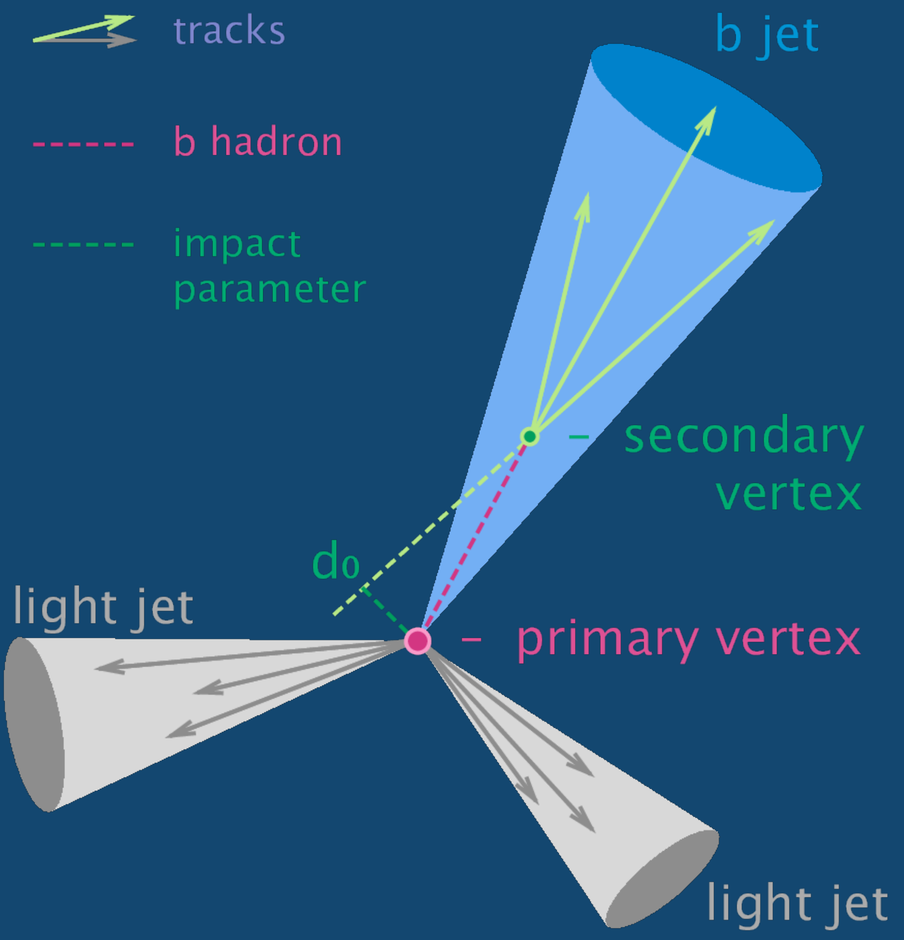
- Largest, 2nd largest, and 3rd largest impact parameter (d_0) of jet constituents



George Halal – JETSCAPE Workshop – Jan 2019

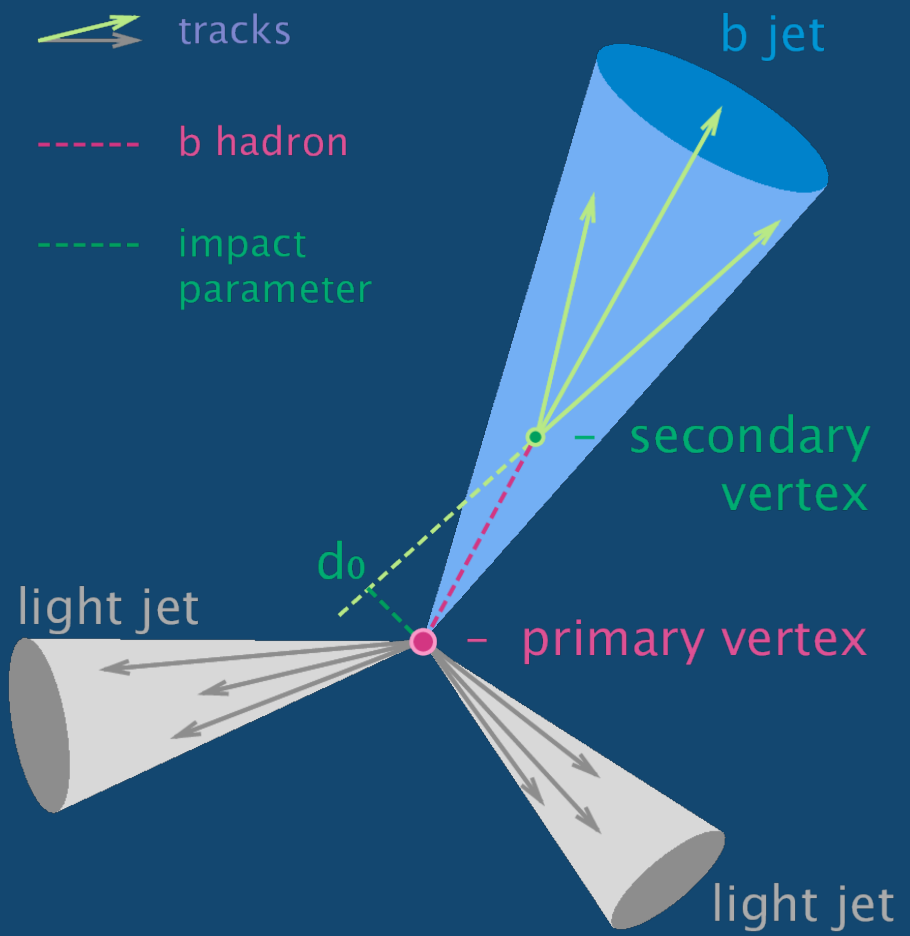
DISCRIMINATORS

- Largest, 2nd largest, and 3rd largest impact parameter (d_0) of jet constituents
- Largest and 2nd largest distance between primary and secondary vertices of jet constituents



DISCRIMINATORS

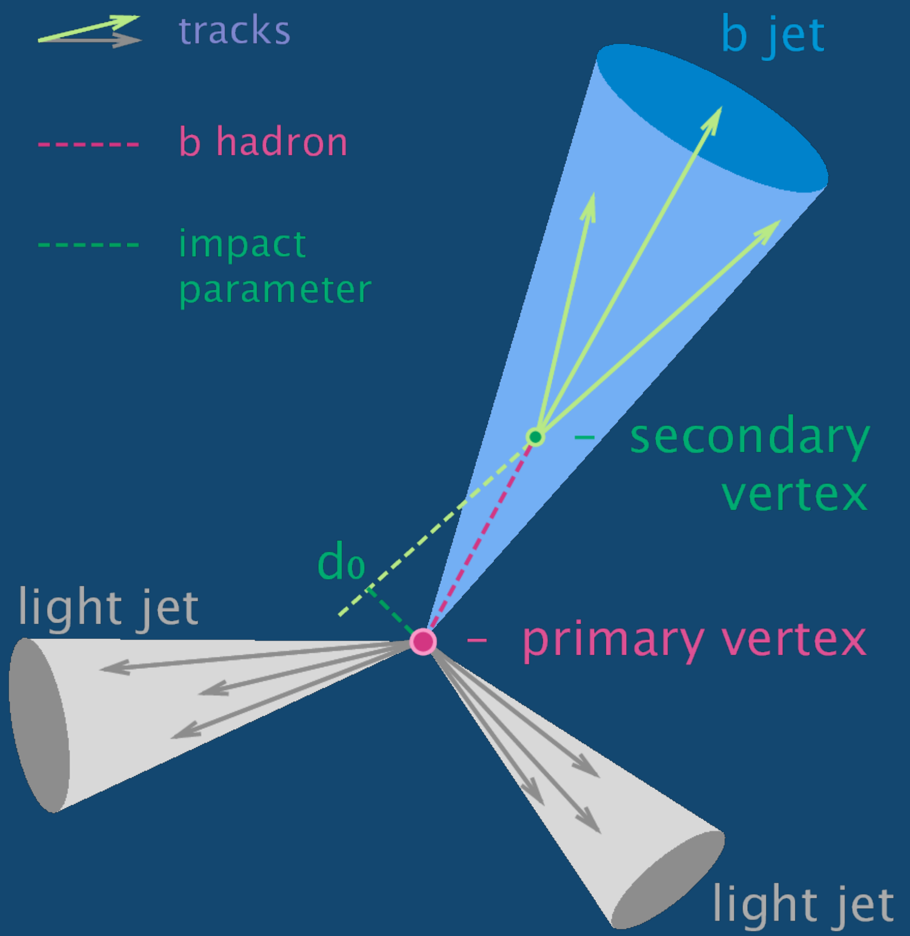
- Largest, 2nd largest, and 3rd largest impact parameter (d_0) of jet constituents
- Largest and 2nd largest distance between primary and secondary vertices of jet constituents
- Number of jet constituents with displaced vertices



George Halal – JETSCAPE Workshop – Jan 2019

DISCRIMINATORS

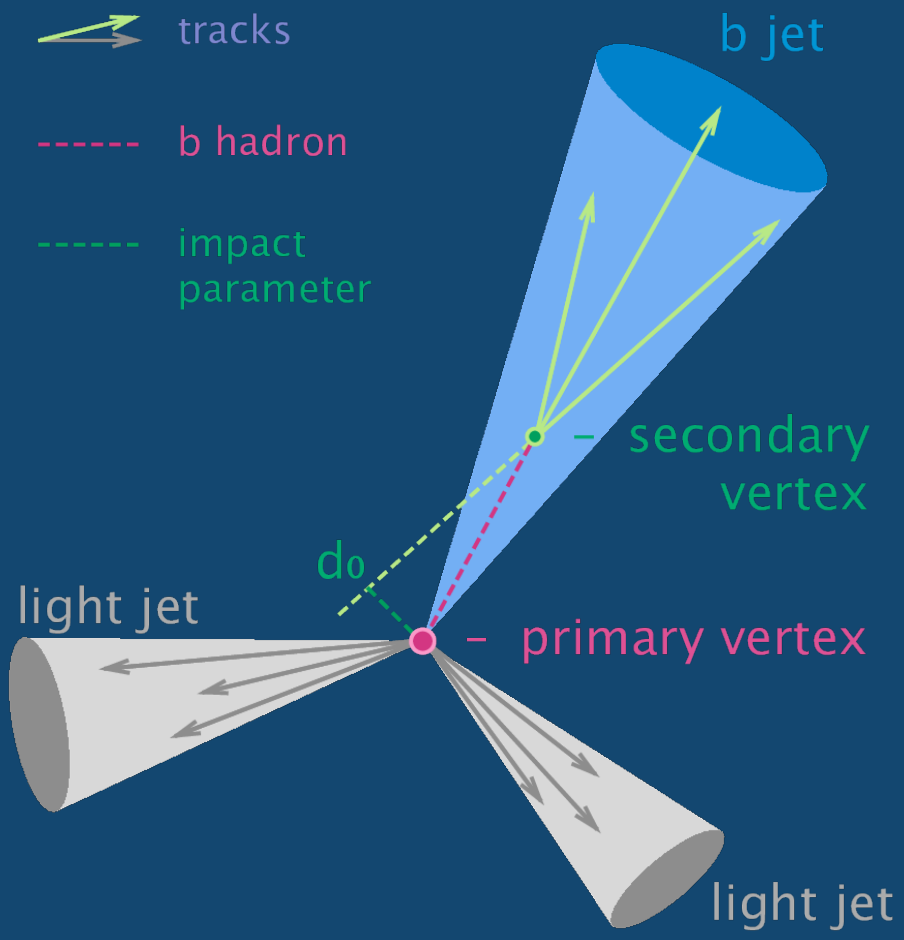
- Largest, 2nd largest, and 3rd largest impact parameter (d_0) of jet constituents
- Largest and 2nd largest distance between primary and secondary vertices of jet constituents
- Number of jet constituents with displaced vertices
- Total number of jet constituents



George Halal – JETSCAPE Workshop – Jan 2019

DISCRIMINATORS

- Largest, 2nd largest, and 3rd largest impact parameter (d_0) of jet constituents
- Largest and 2nd largest distance between primary and secondary vertices of jet constituents
- Number of jet constituents with displaced vertices
- Total number of jet constituents
- Presence of soft leptons (due to semi-leptonic decays of heavy-flavor hadron)



DATA PREPARATION

INPUTS

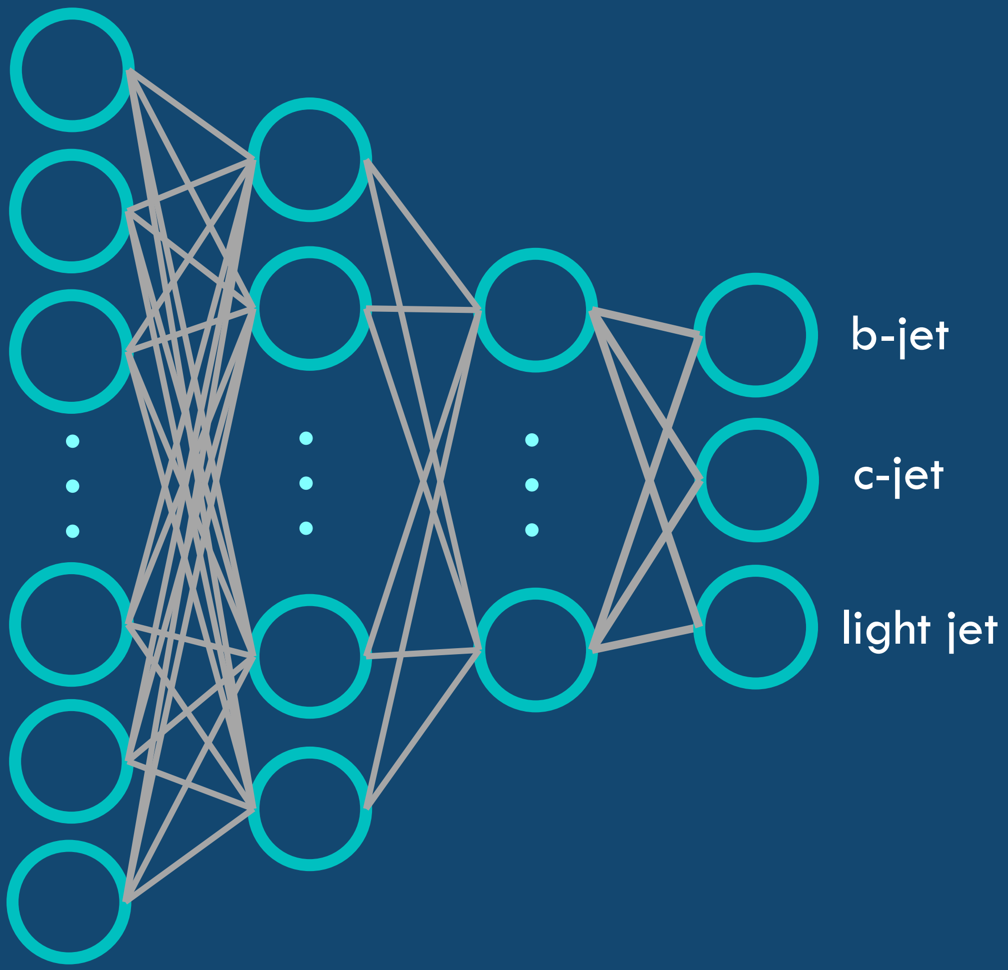
OUTPUTS

	DCA-1st	DCA-2nd	DCA-3rd	SV-1st	SV-2nd	N_{SV}	Muon?	e^-?	$N_{constit.}$	b-jet?	c-jet?	light jet?
Jet 1										0	0	1
Jet 2										1	0	0
Jet 3										0	1	0
x180k												

Convert TTrees (ROOT) to NumPy arrays (Python)

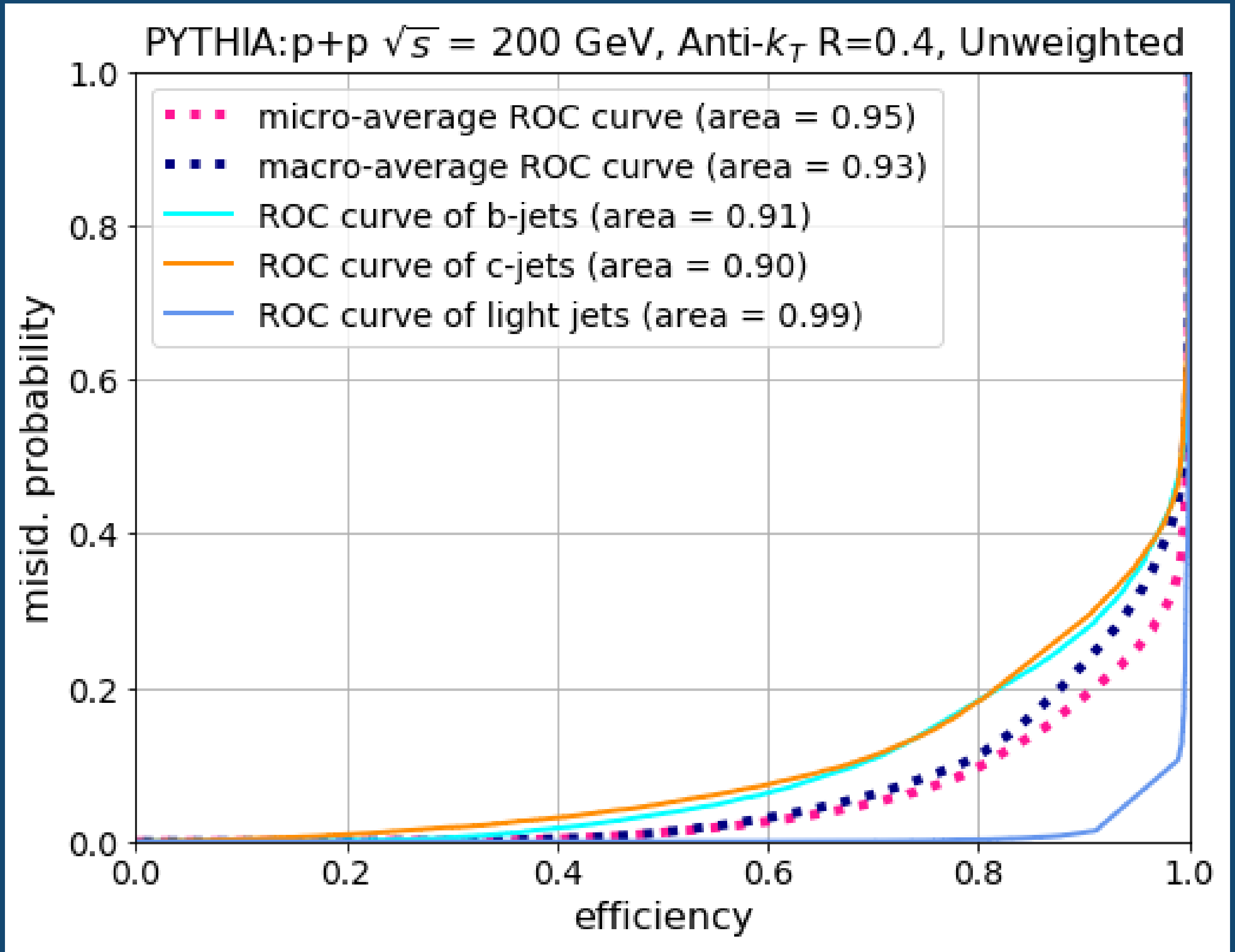
NEURAL NETWORK ARCHITECTURE

DISCRIMINATORS (x9)



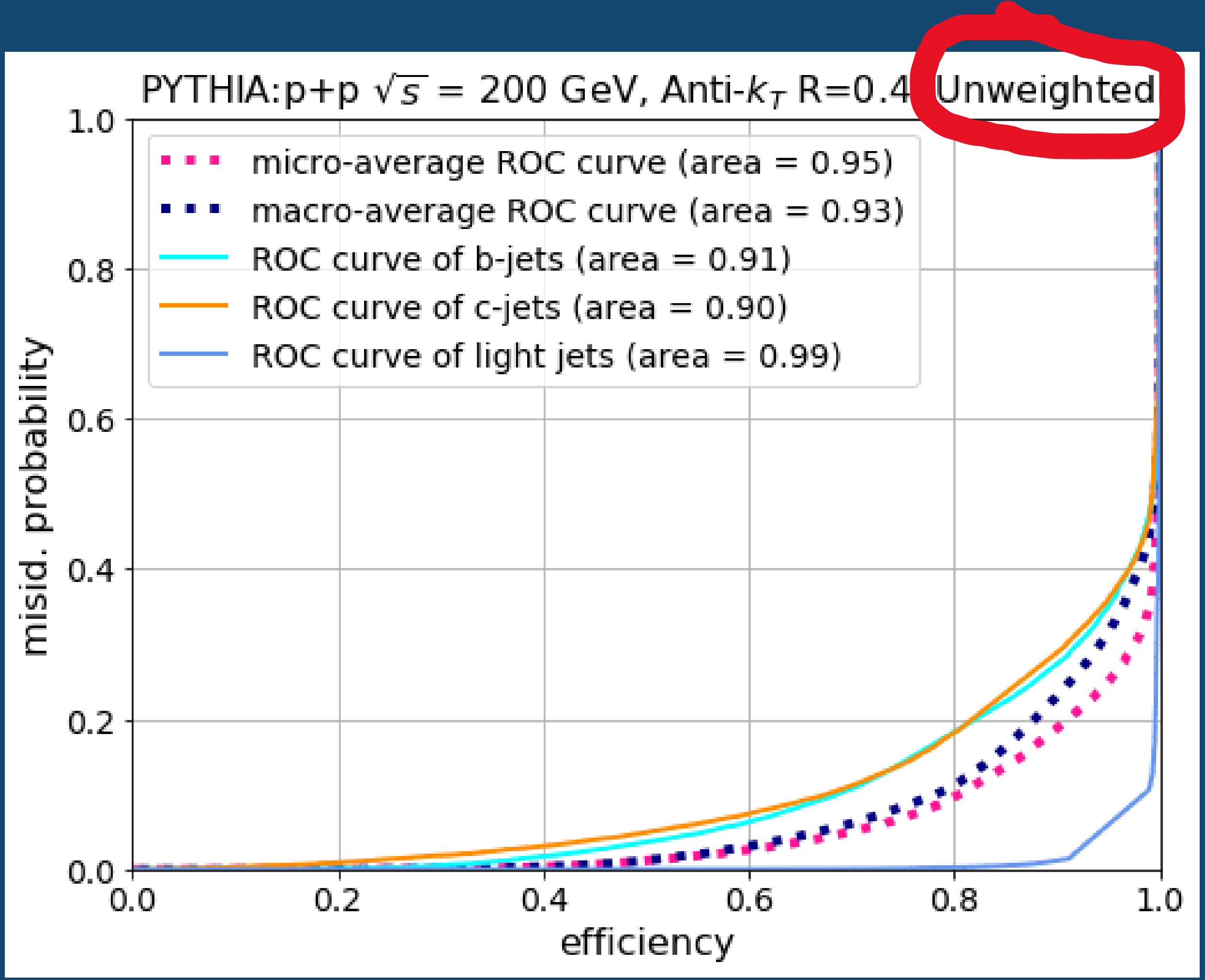
- Use Keras (TensorFlow) & scikit-learn packages in Python
- Input layer (9 nodes), 2 hidden layers (8 nodes, 5 nodes), & output layer (3 nodes)
- Kernel Initializer: Normal
- Activations: ReLU for hidden layers & softmax for output
- Loss: categorical_crossentropy
- Optimizer: adam
- Epochs = 200
- Batch Size = 200
- Scale inputs before training
- Train on equal samples of each flavor

BEAUTIFUL?



PYTHIA:p+p $\sqrt{s}$ = 200 GeV, Anti-k_T R=0.4, Unweighted
micro-average ROC curve (area = 0.95)
macro-average ROC curve (area = 0.93)
ROC curve of b-jets (area = 0.91)
ROC curve of c-jets (area = 0.90)
ROC curve of light jets (area = 0.99)
misid. probability
efficiency
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

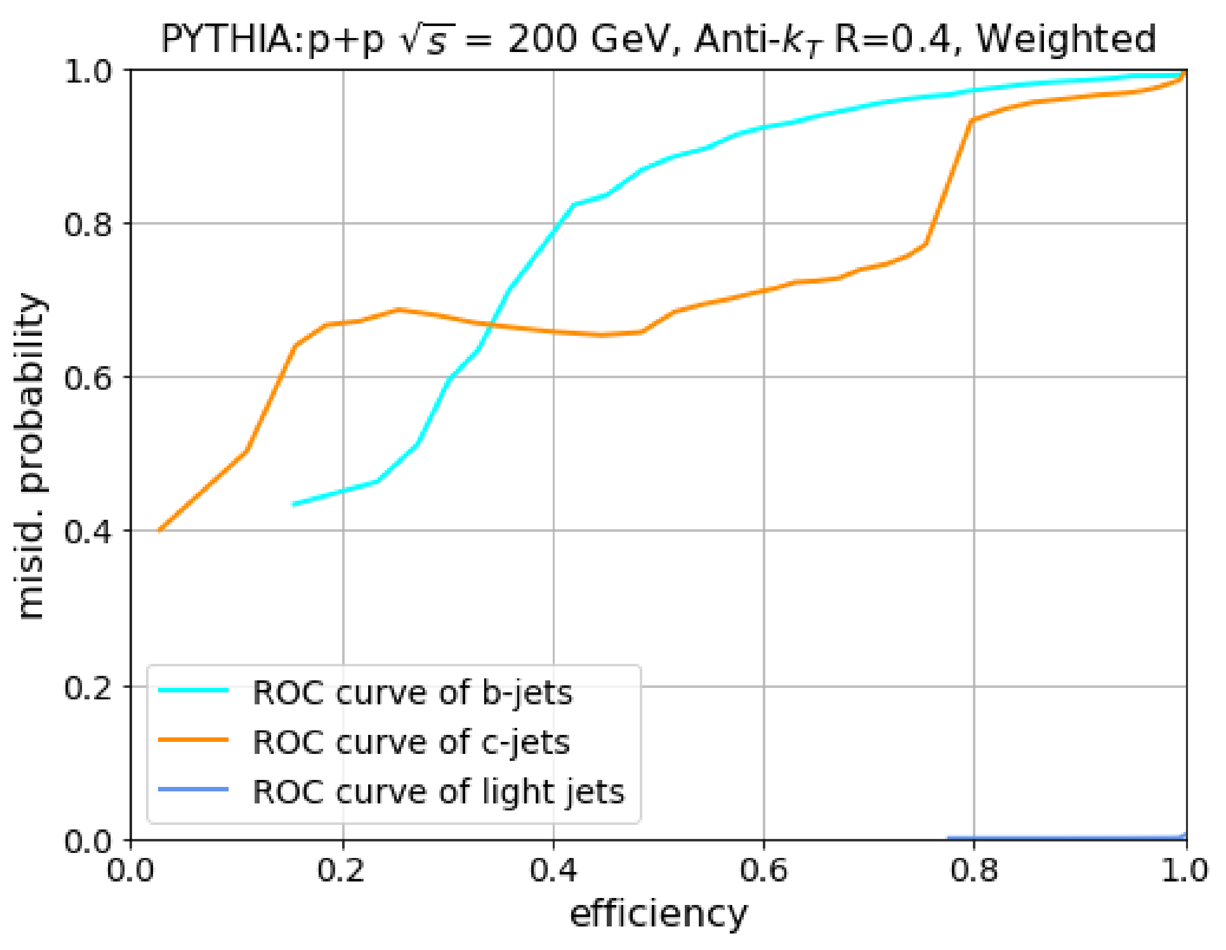
BEAUTIFUL? NOT SO FAST..



WEIGHTING

- Must take relative production rates into account
- According to what PYTHIA spits out at the end of the run, for $\sqrt{s}$=200 GeV:
  - σ_{bottom} = 1.812 e-4 mb
  - σ_{charm} = 6.998 e-4 mb
  - σ_{light} = 1.97866 e-1 mb
- Therefore, $\sigma_{bottom}:\sigma_{charm}:\sigma_{light} \approx 1:4:1092$

WHEN TAKING WEIGHTING INTO ACCOUNT



SOLUTIONS

Because the weighting is what hurts the ROC curve, only consider jets with the requirement:

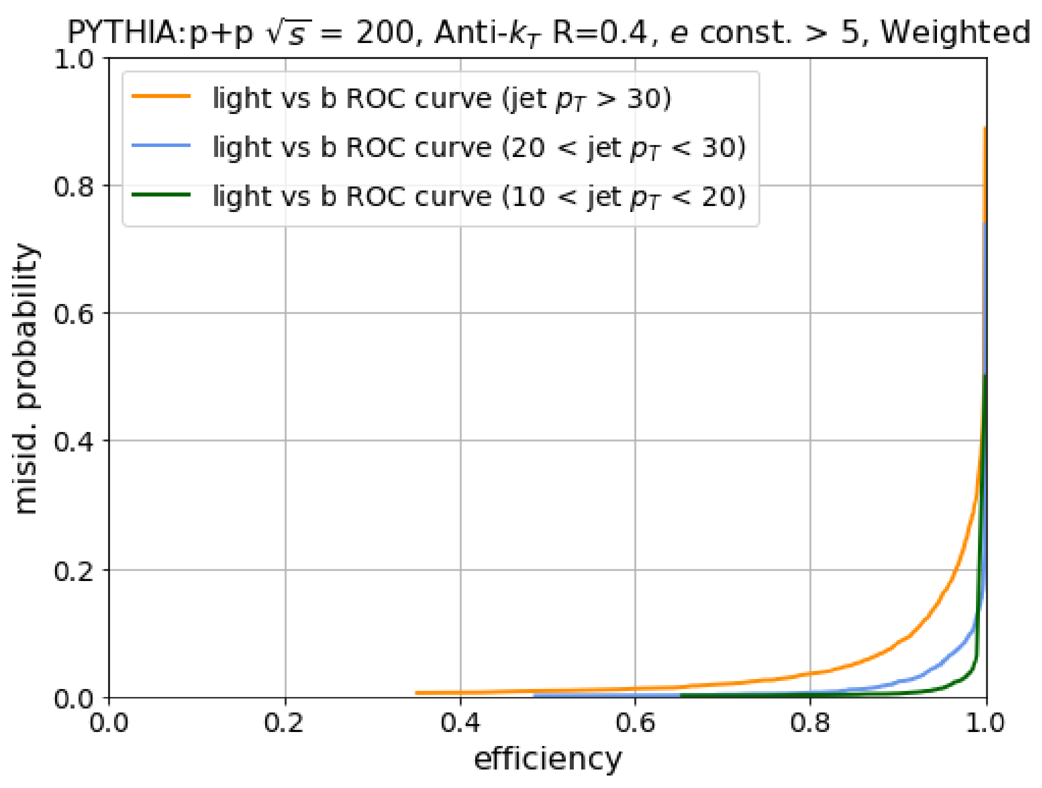
- At least 1 electron constituent with a p_T > 5.0 GeV/c (High Tower trigger)

  New relative cross-section ratio: $\sigma_{bottom}:\sigma_{charm}:\sigma_{light} \approx 1:1:1$

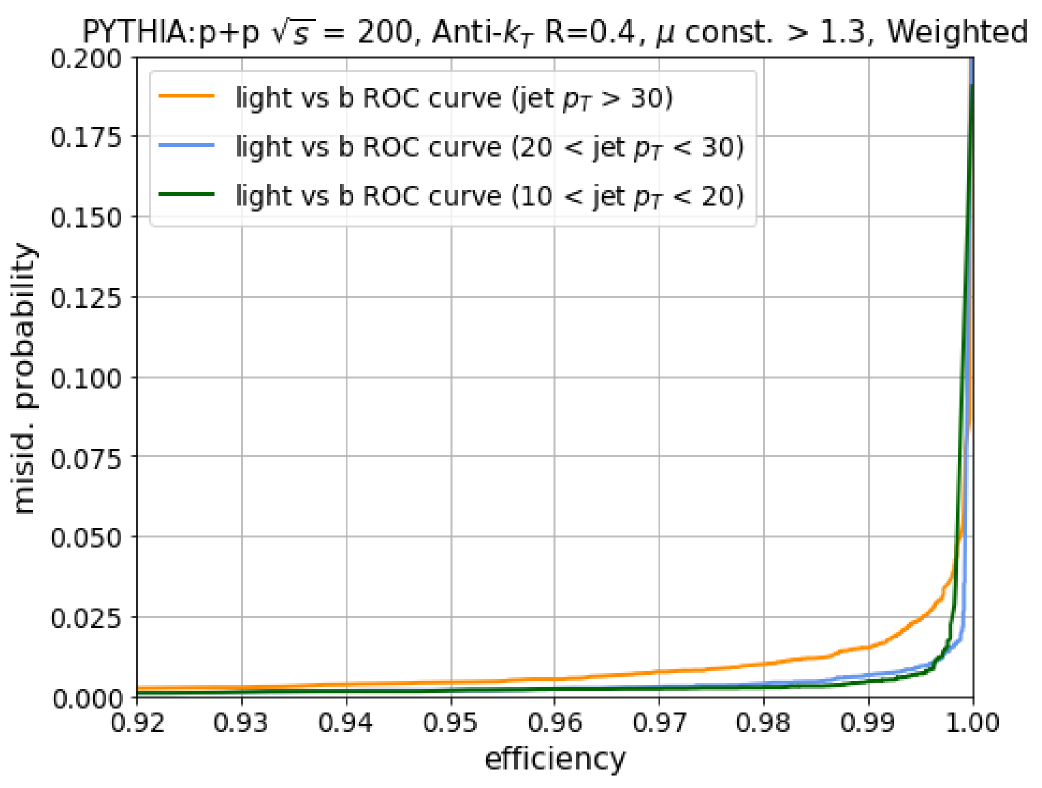
- At least 1 muon constituent with a p_T > 1.3 GeV/c (Muon Telescope Detector trigger)

  New relative cross-section ratio: $\sigma_{bottom}:\sigma_{charm}:\sigma_{light} \approx 4:9:1$

ONLY JETS WITH AT LEAST ONE HIGH p_T ELECTRON

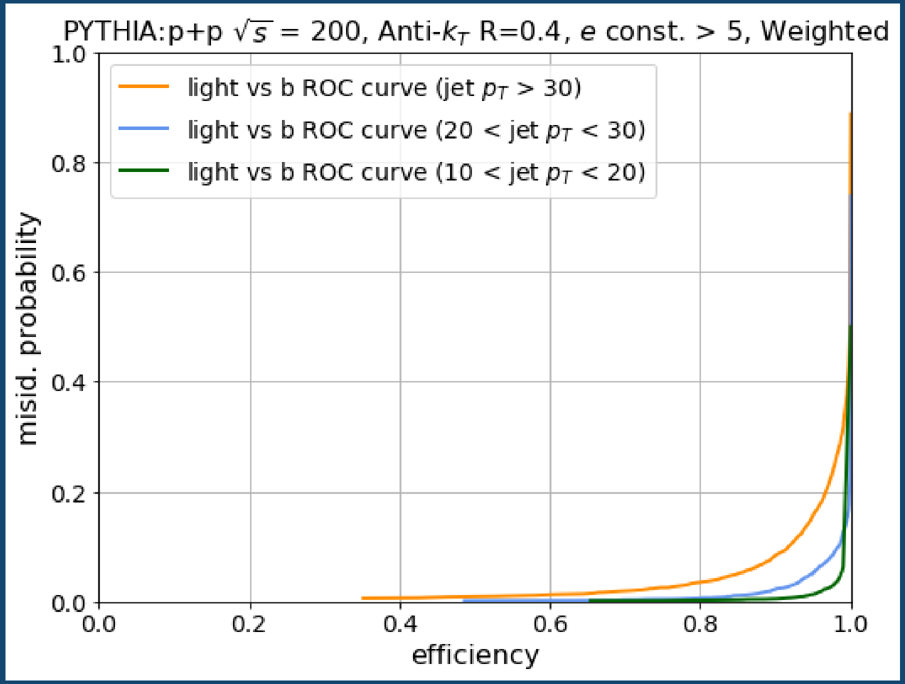


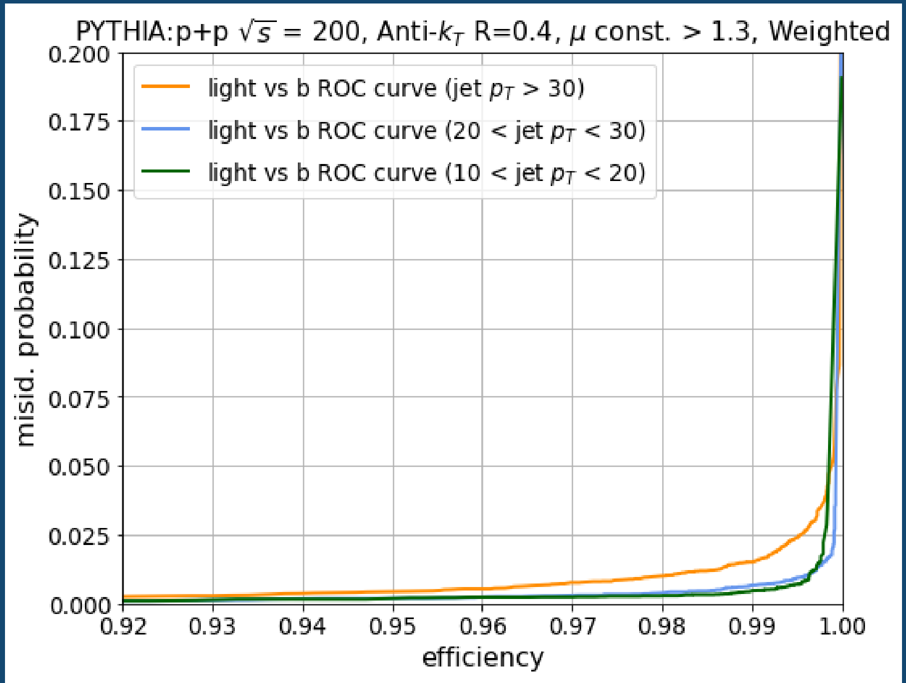
ONLY JETS WITH AT LEAST ONE HIGH p_T MUON



MISIDENTIFICATION PROBABILITY

90% Efficiency	High pT electron constituent	High pT muon constituent
Jet $p_T > 30$	8.0 %	0.18 %
$20 < \text{Jet } p_T < 30$	2.0 %	0.085 %
$10 < \text{Jet } p_T < 20$	0.43 %	0.073 %





STATISTICS

PYTHIA:pp200	jet pT range	pTHatMin
Anti-kT, R=0.4	10-20	9.5
Full Jets	20-30	19.5
	>30	29.5

STAR Run15	Trigger	Nevents
	singlemuon-5	7.40E+07
	BHT1 (3.4 GeV)	4.80E+08
	BHT2 (4.08 GeV)	2.30E+08

muon > 1.3 GeV/c	relative production cross-sections		
jet pT range	light	bottom	charm
10-20	1	4.25	9.17
20-30	1	3.54	3.65
>30	1	1.86	1.64

e > 5 GeV/c	relative production cross-sections		
jet pT range	light	bottom	charm
10-20	1.12	1.08	1
20-30	3.31	1.18	1
>30	9.47	1.2	1

no trigger	relative production cross-sections		
jet pT range	light	bottom	charm
10-20	187.5	1	5
20-30	130	1	2.74
>30	166.67	1	2.72

muon > 1.3 GeV/c	# events to produce		
jet pT range	1 light	1 b	1 c
10-20	15000	35000	16000
20-30	40000	11000	11000
>30	20000	11000	12000

e > 5 GeV/c	# events to produce		
jet pT range	1 light	1 b	1 c
10-20	79000	84000	92000
20-30	8000	23000	26000
>30	2500	19000	24000

no trigger	# events to produce		
jet pT range	1 light	1 b	1 c
10-20	8	1500	300
20-30	4	520	190
>30	3	500	184

CONCLUSION

- Relative production rates lead to the inability of tagging heavy flavor jets at RHIC energies with no triggers
- With a high tower or MTD trigger, we can tag heavy flavor jets at RHIC energies with a very high efficiency and low misidentification probability

MOVING FORWARD

- Repeat the analysis for quenched jets using JETSCAPE
- Compare to JEWEL and/or PYTHIA + HI background
- Try additional discriminators
- Add STAR/sPHENIX detector effects

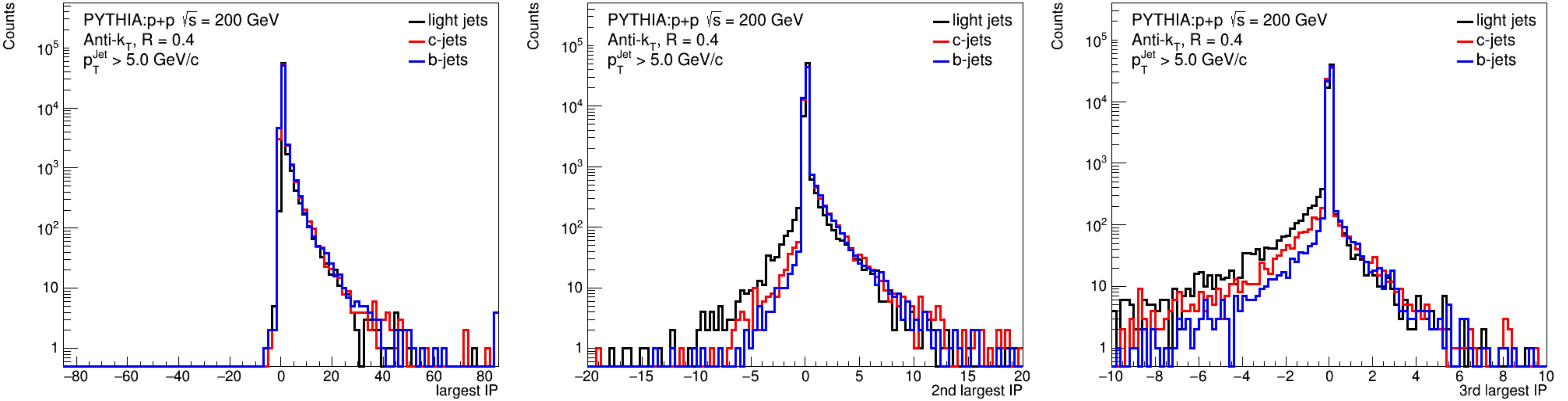
THANK YOU!

QUESTIONS?

BACKUP

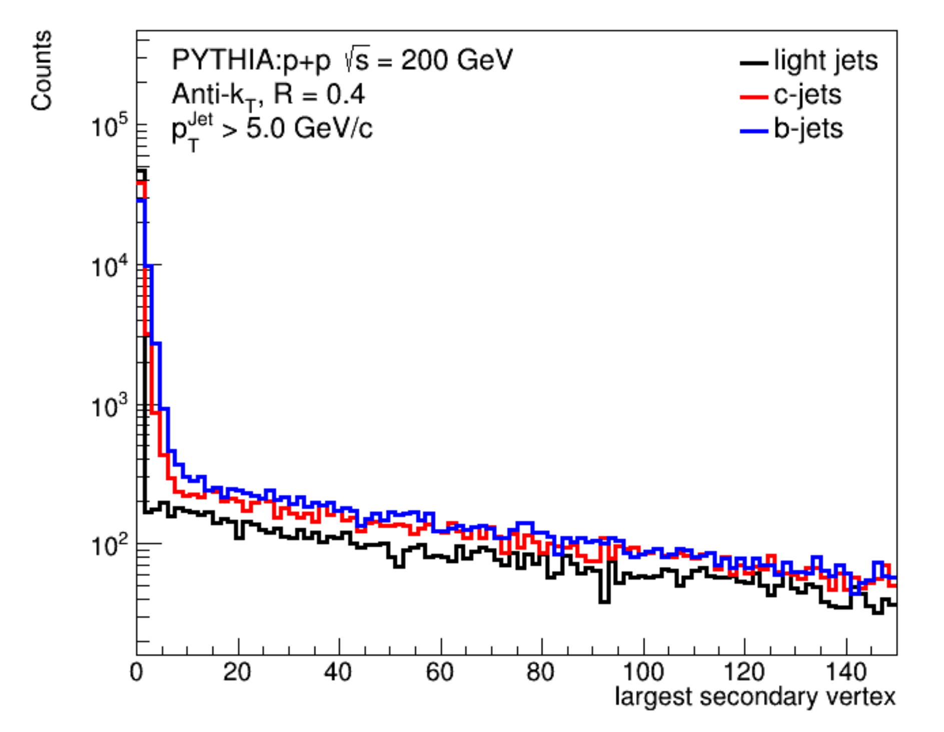
DISCRIMINATORS

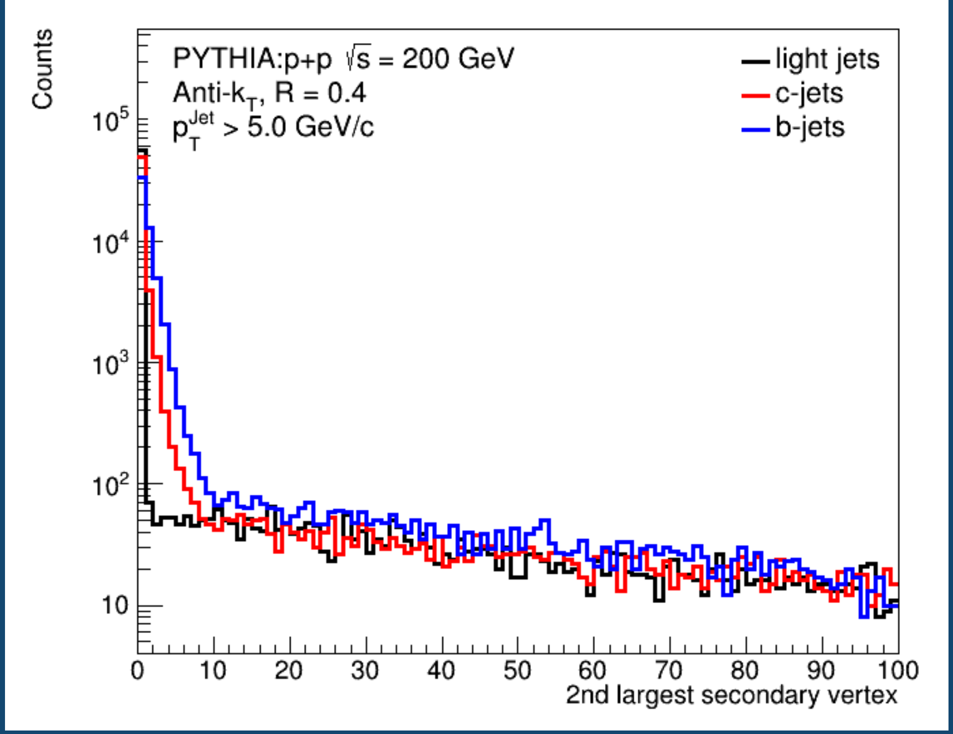
- Largest, 2nd largest, and 3rd largest impact parameter (d_0) of jet constituents



DISCRIMINATORS

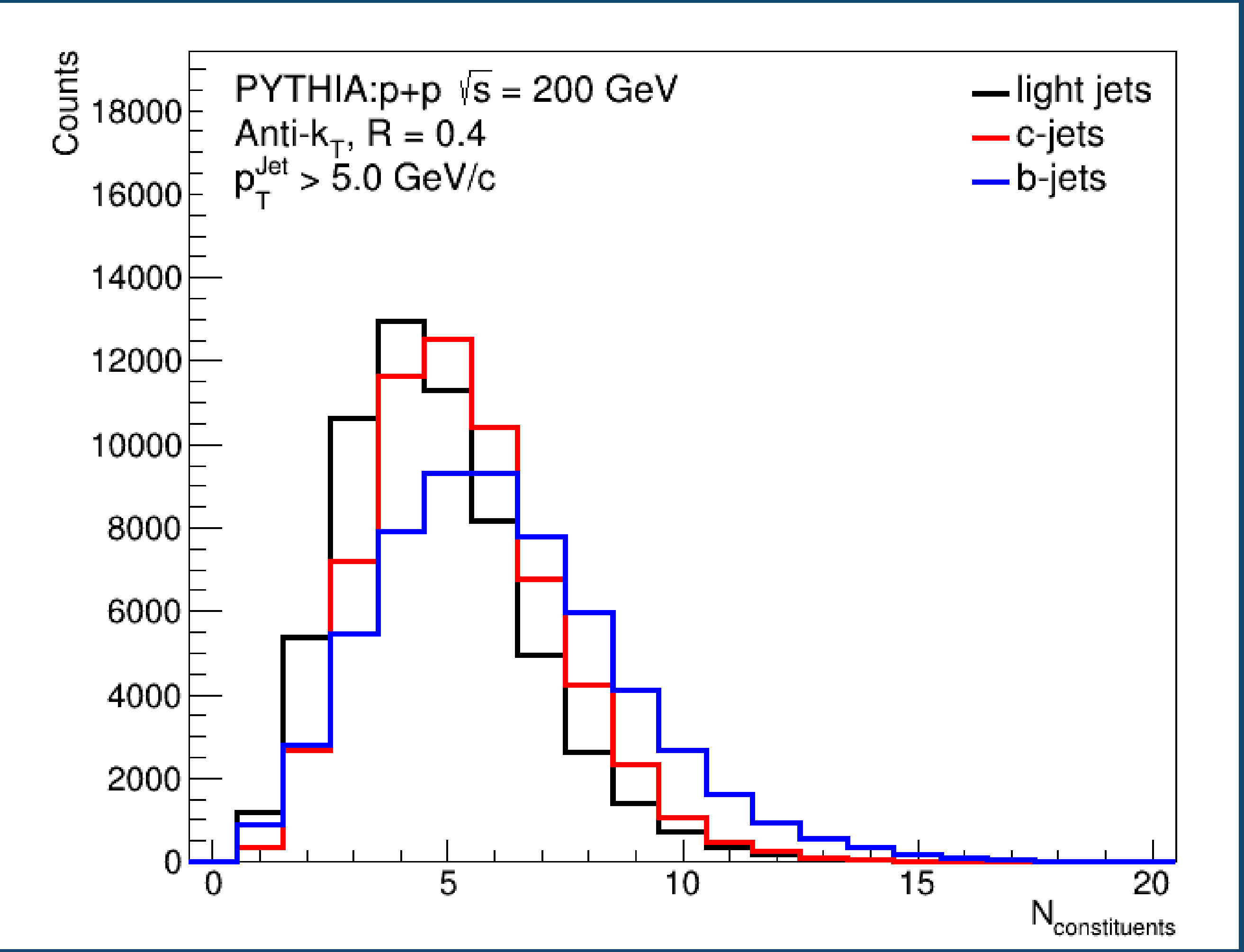
- Largest and 2nd largest distance between primary and secondary vertices of jet constituents





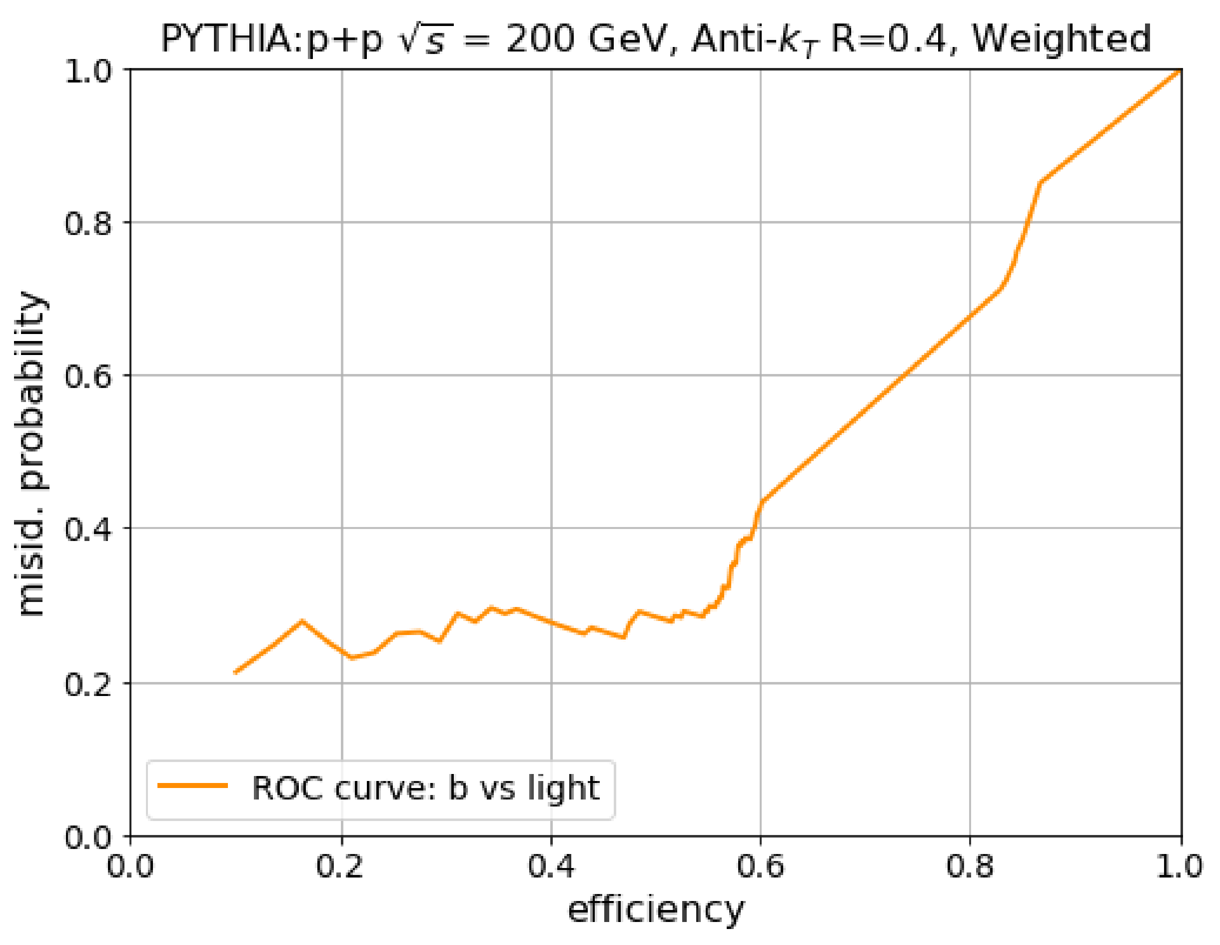
DISCRIMINATORS

- Jet $N_{Constituents}$

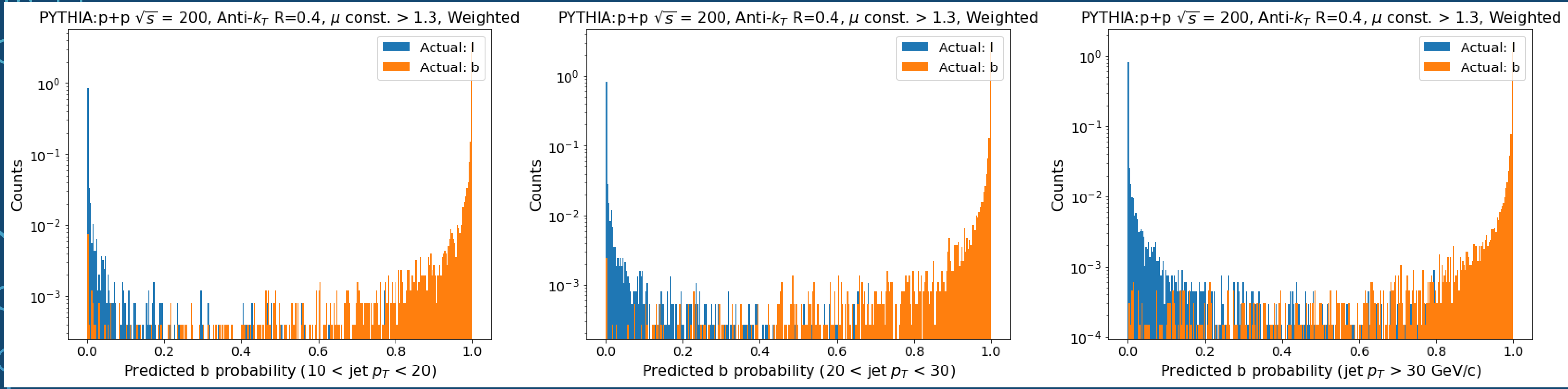


George Halal – JETSCAPE Workshop – Jan 2019

INCLUDING THE WEIGHTING IN THE TRAINING

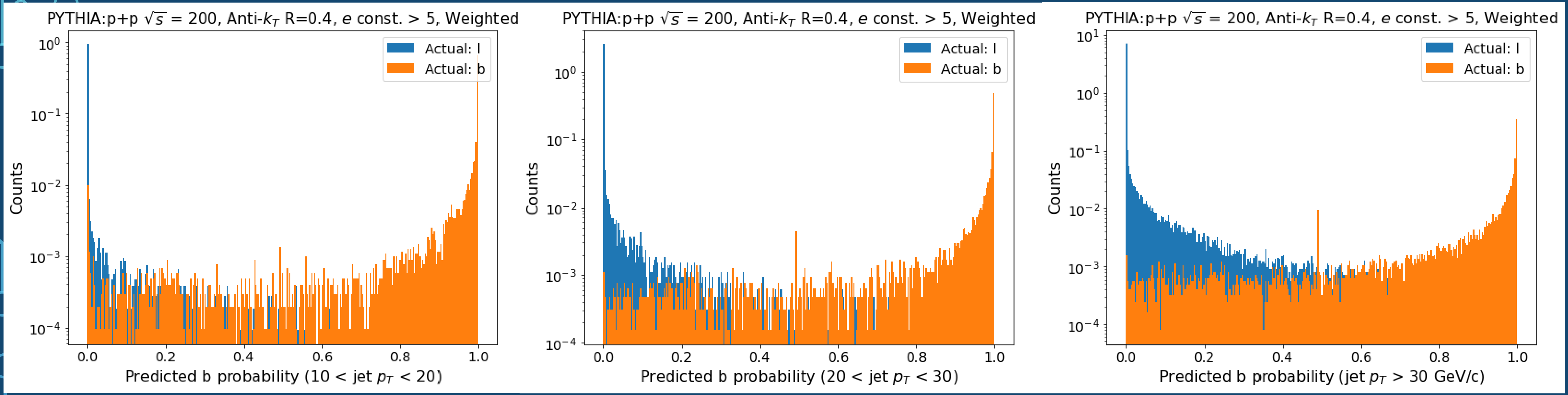


HOW THE SAUSAGE IS MADE – MUON EDITION



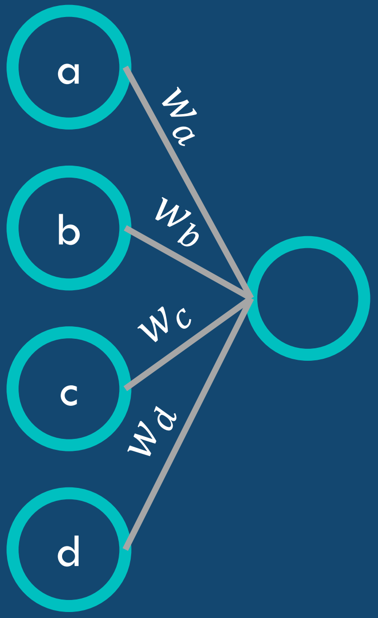
- Pick different thresholds
- Integrate to find efficiency and impurity for each threshold
- Repeat until you have enough points to make the ROC curve

HOW THE SAUSAGE IS MADE – ELECTRON EDITION

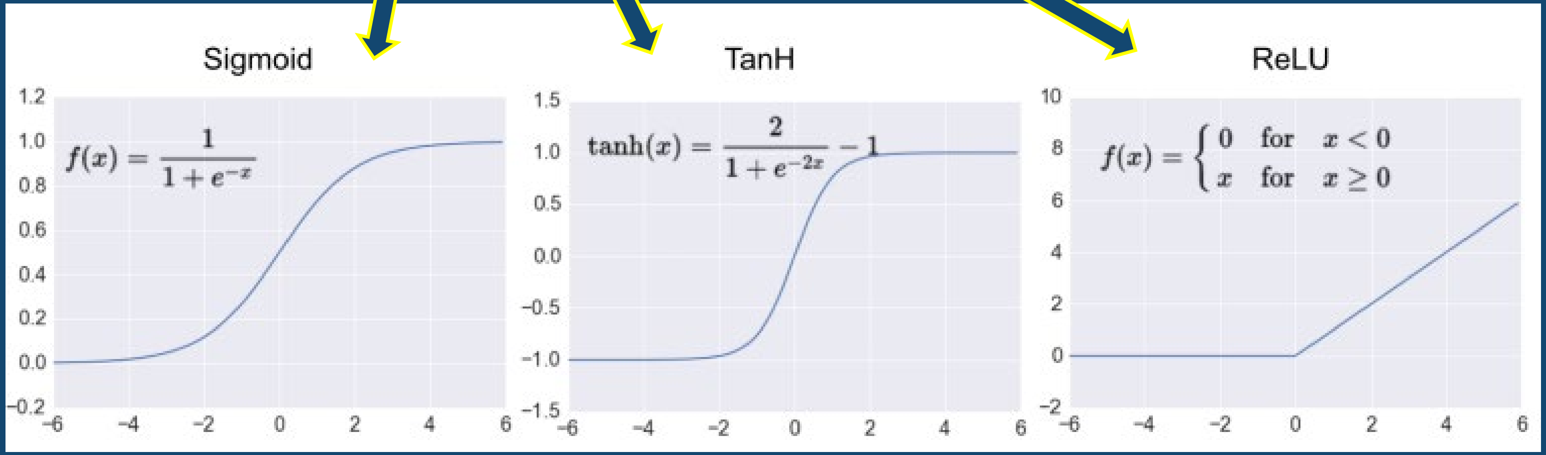


- Pick different thresholds
- Integrate to find efficiency and impurity for each threshold
- Repeat until you have enough points to make the ROC curve

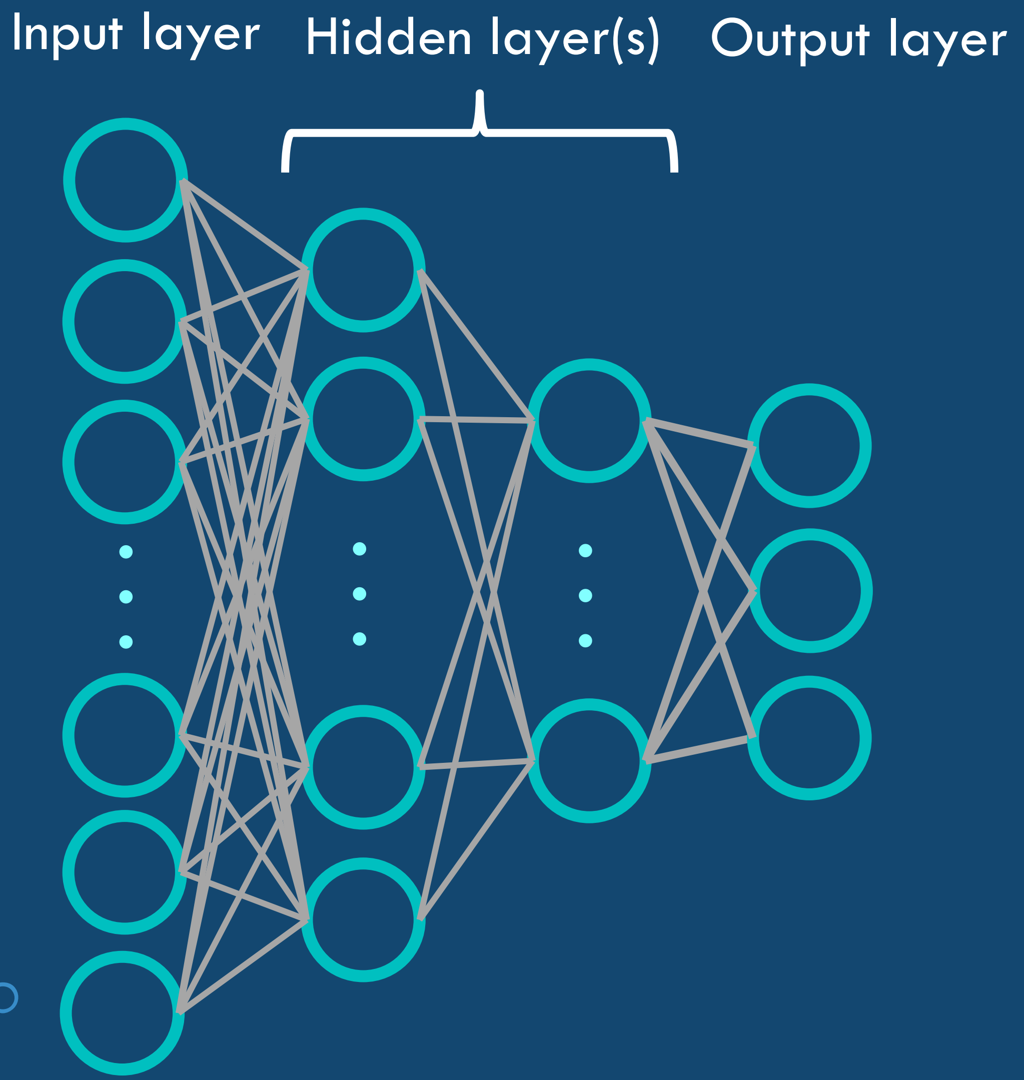
MACHINE LEARNING & NEURAL NETWORKS



$$= f\ (a\,.\,w_a + b\,.\,w_b + c\,.\,w_c + d\,.\,w_d + bias)$$



MACHINE LEARNING & NEURAL NETWORKS



- Start with random weights and biases
- Calculate the difference between predicted and actual outputs
- Update weights and biases by taking the negative gradient
- Step back to previous layer and repeat