Decorrelated Jet Substructure Tagging using Adversarial Neural Networks

Chase Shimmin

ML for Phenomenology
IPPP Durham
April 5th, 2018

Yale University
LUX ET VERITAS

Decorrelated Jet Substructure Tagging using Adversarial Neural Networks

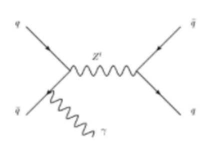
Chase Shimmin, Peter Sadowski, Pierre Baldi, Edison Weik, Daniel Whiteson (UC, Irvine), Edward Goul (MIT, Cambridge, Dept. Phys.), Andreas Søgaard (Edinburgh U.)

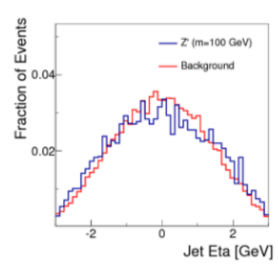
Mar 9, 2017 - 10 pages

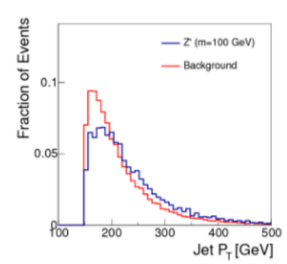
Phys.Rev. D96 (2017) no.7, 074034
(2017-10-30)
DOI: 10.1103/PhysRevD.96.074034
e-Print: arXiv:1703.03507 [hep-ex] | PDF

Abstract (APS)
We describe a strategy for constructing a neural network jet substructure tagger which powerfully discriminates boosted decay signals while remaining largely uncorrelated with the jet mass. This reduces the impact of systematic uncertainties in background modeling while enhancing signal purity, resulting in improved discovery significance relative to existing taggers. The network is trained using an adversarial strategy, resulting in a tagger that learns to balance classification accuracy with decorrelation. As a benchmark scenario, we consider the case where large-radius jets originating from a boosted resonance decay are discriminated from a background of nonresonant quark and gluon jets. We show that in the presence of systematic uncertainties on the background rate, our adversarially trained, decorrelated tagger considerably outperforms a conventionally trained neural network, despite having a slightly worse signal-background separation power. We generalize the adversarial training technique to include a parametric dependence on the signal hypothesis, training a single network that provides optimized, interpolatable decorrelated jet tagging across a continuous range of hypothetical resonance masses, after training on discrete choices of the signal mass.
Abstract (arXiv)

Keyword(s): INSPIRE: track data analysis: jet | jet: mass | gluon: jet | resonance: hadronic decay | boosted particle | resonance: mass | neural network | background | structure | network | parametric | benchmark | quark: jet | programming | statistical analysis | data analysis method | experimental results







Show more plots

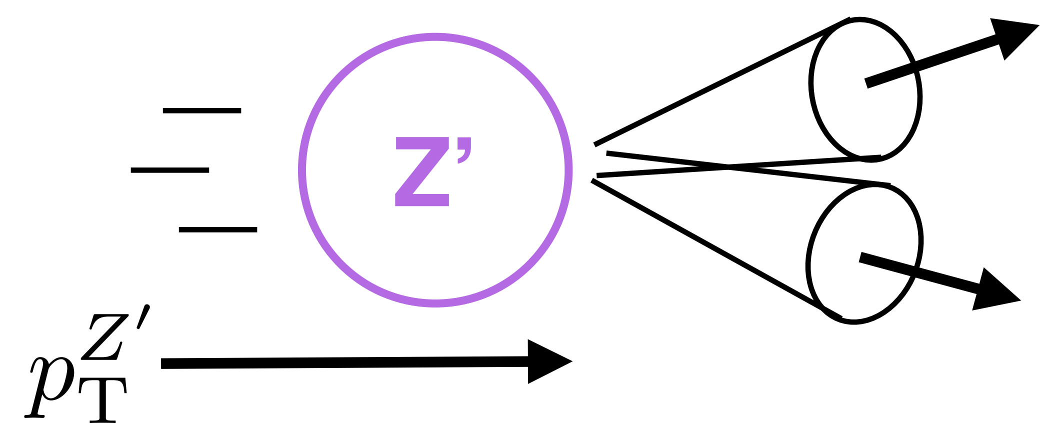
Record added 2017-03-13, last modified 2017-11-09

Boosted Objects

at rest

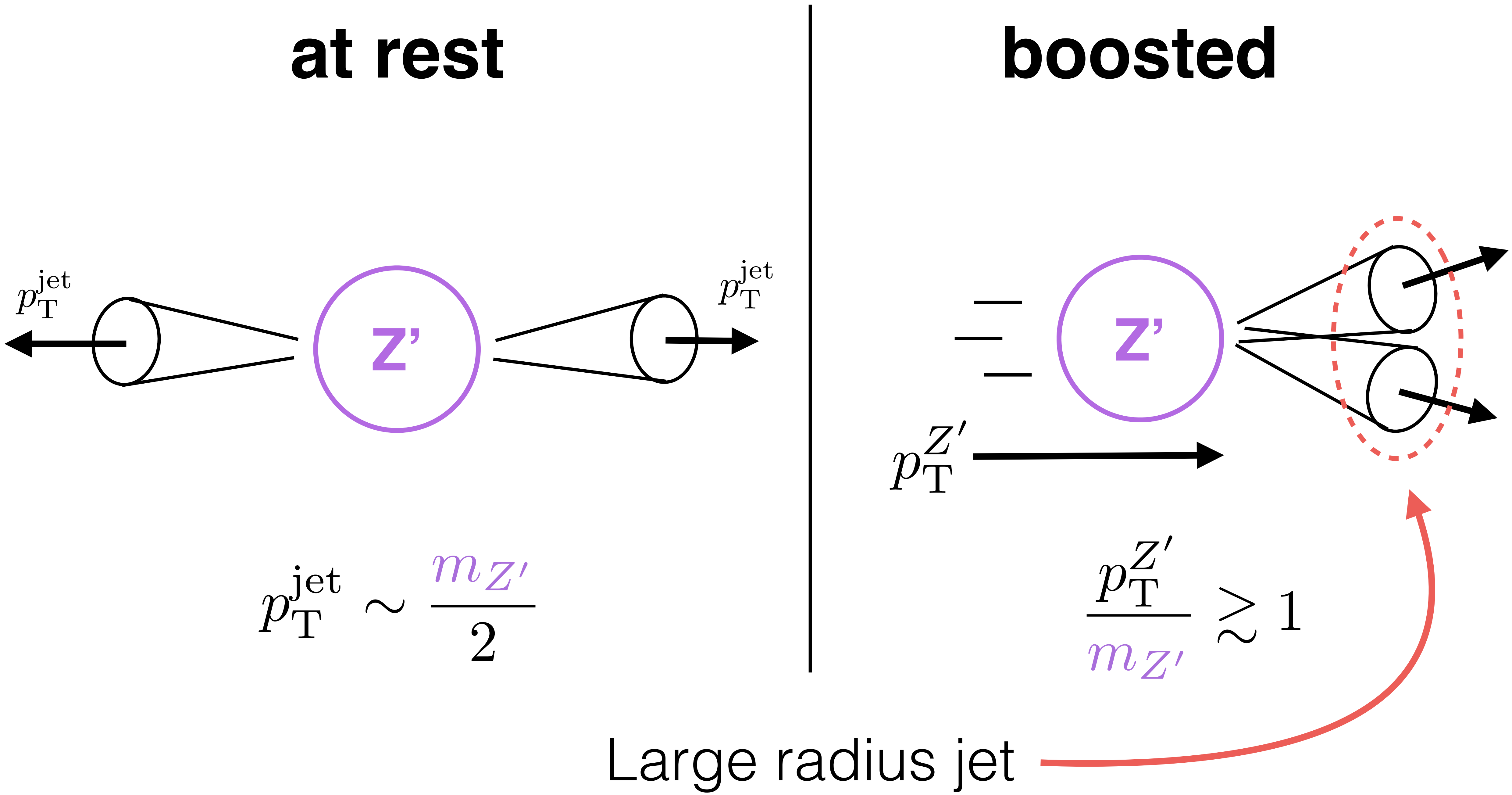
$$p_{\mathrm{T}}^{\mathrm{jet}} \sim \frac{m_{Z'}}{2}$$

boosted



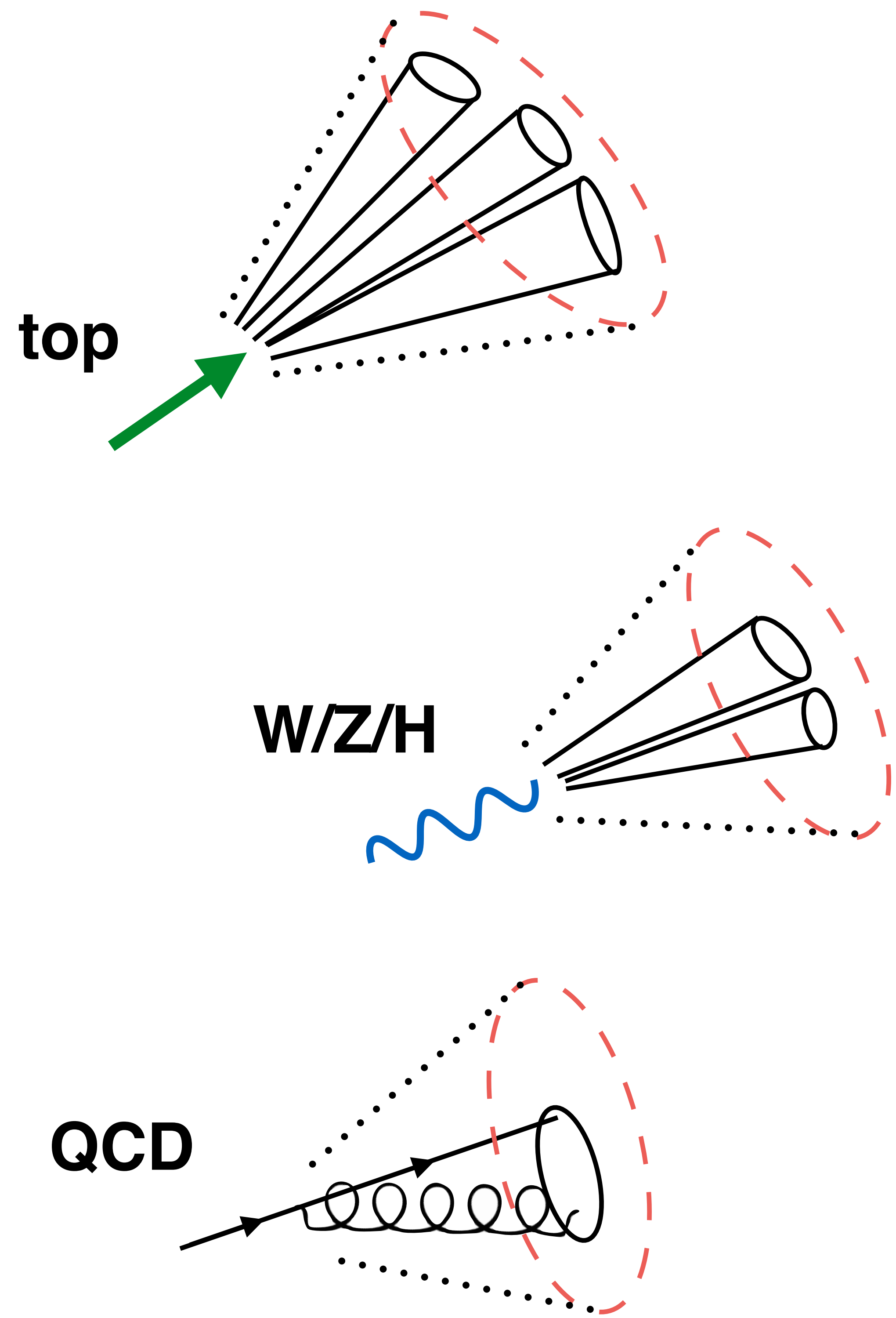
$$\frac{p_{\mathrm{T}}^{Z'}}{m_{Z'}} \gtrsim 1$$

Boosted Objects



(boosted) Jet Tagging

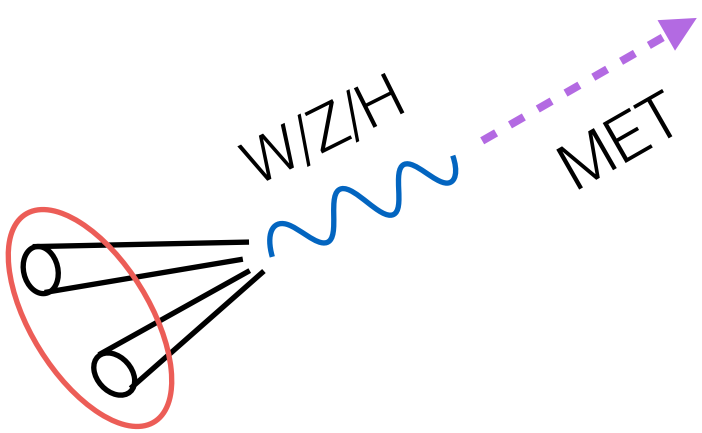
Goal: identify initial particle that caused the jet



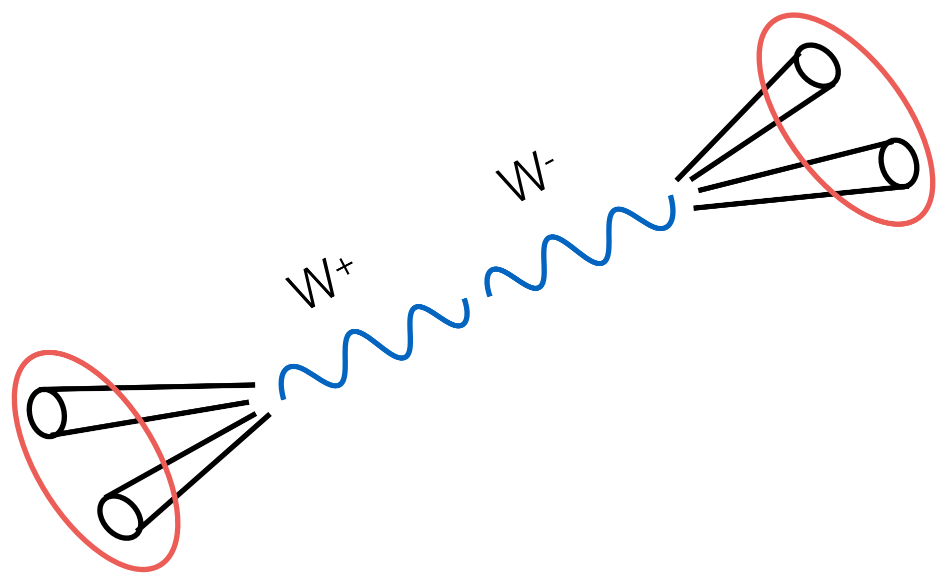
Analysis Applications

Generally want to enhance signal containing **known objects** over QCD background:

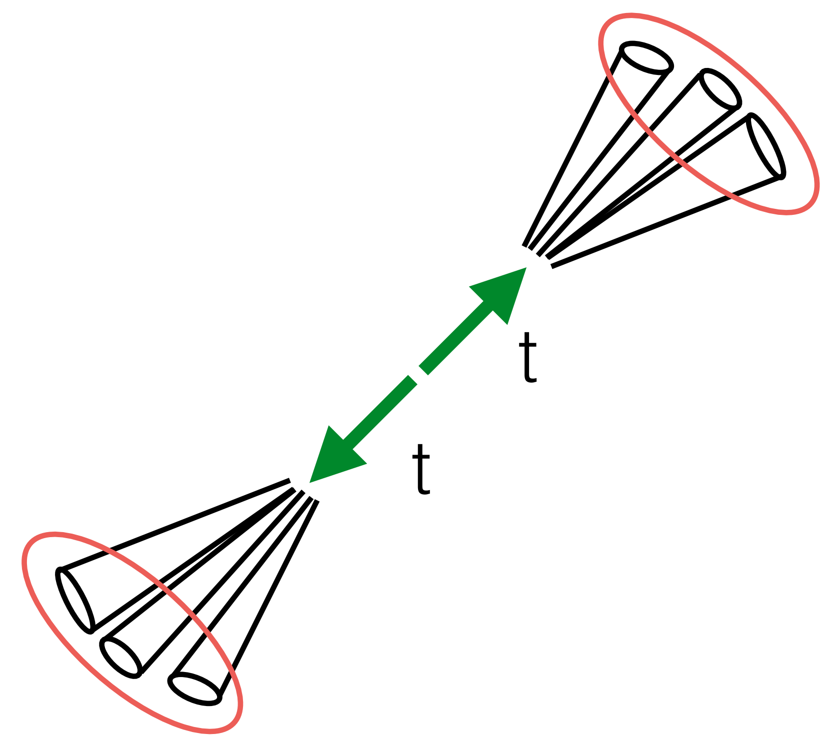
“Mono-X”



VV resonance

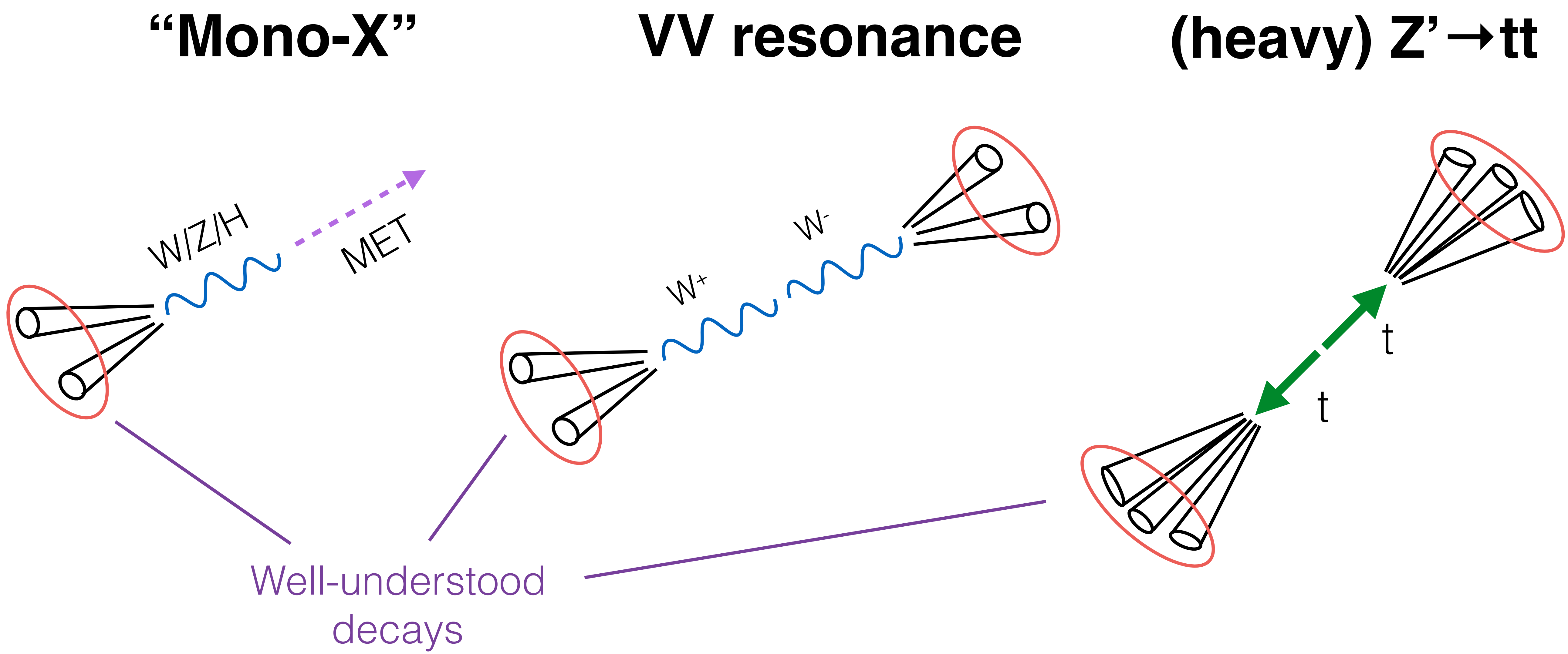


(heavy) Z’→tt



Analysis Applications

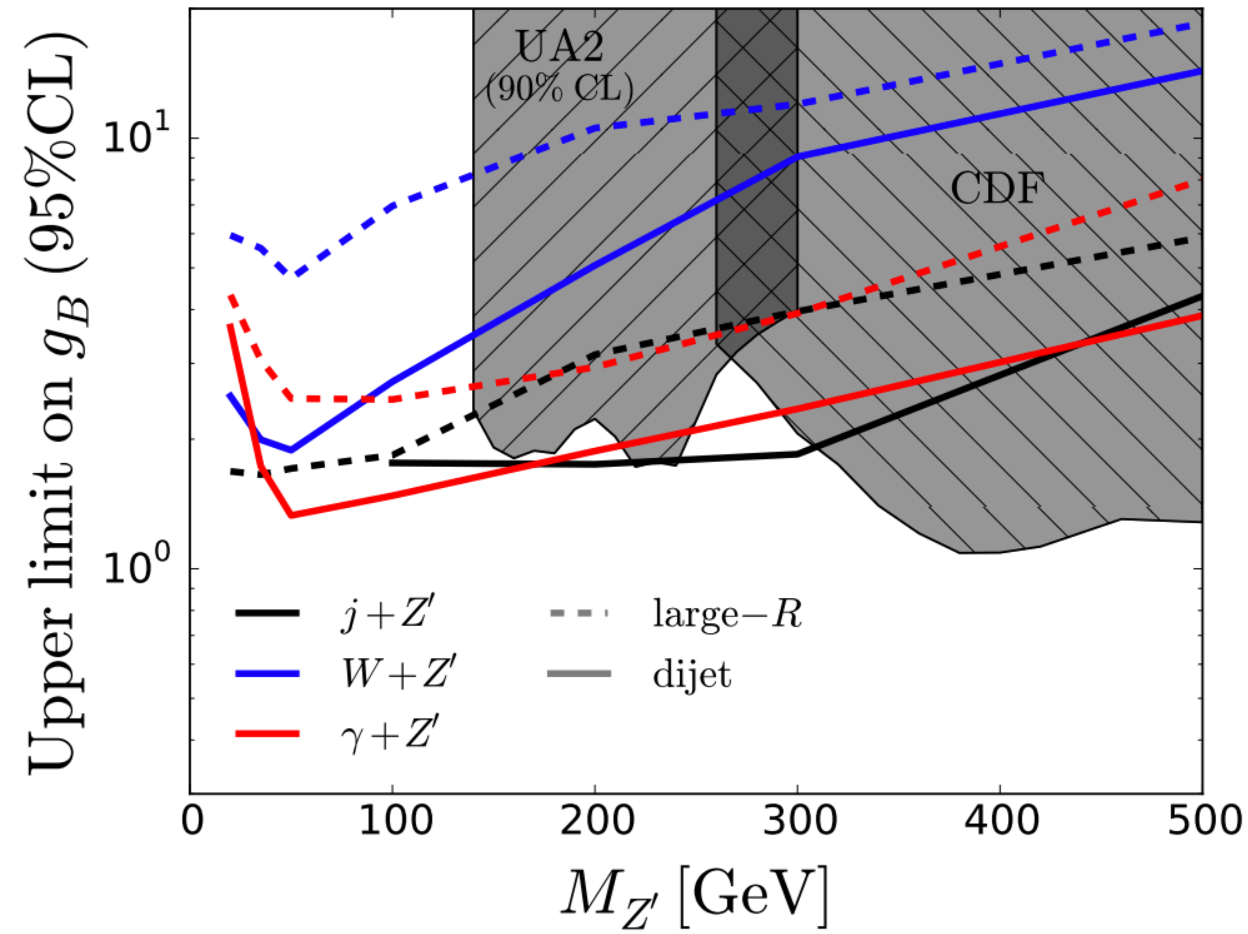
Generally want to enhance signal containing **known objects** over QCD background:



Analysis Applications

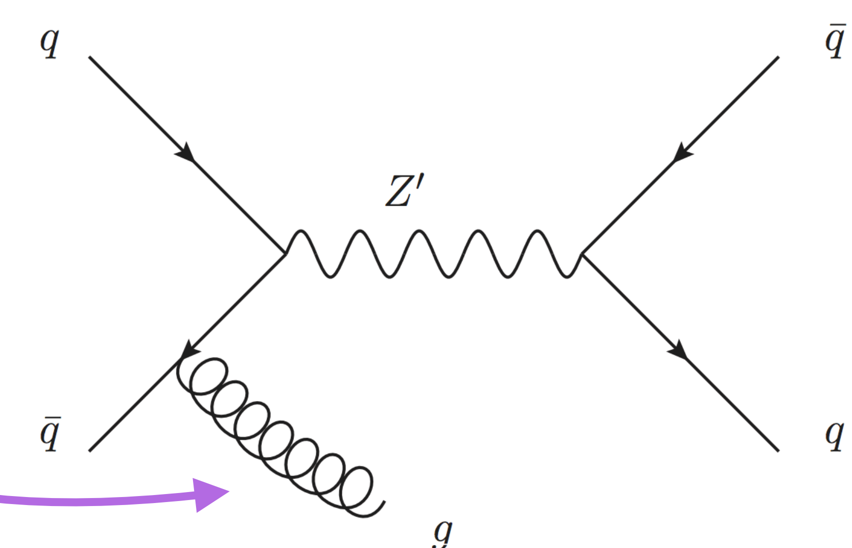
Pet project:
Very low-mass resonances

- Existing direct limits were set in the 90's!
- Typically hard to access: trigger thresholds increase with luminosity and sqrt(s)!



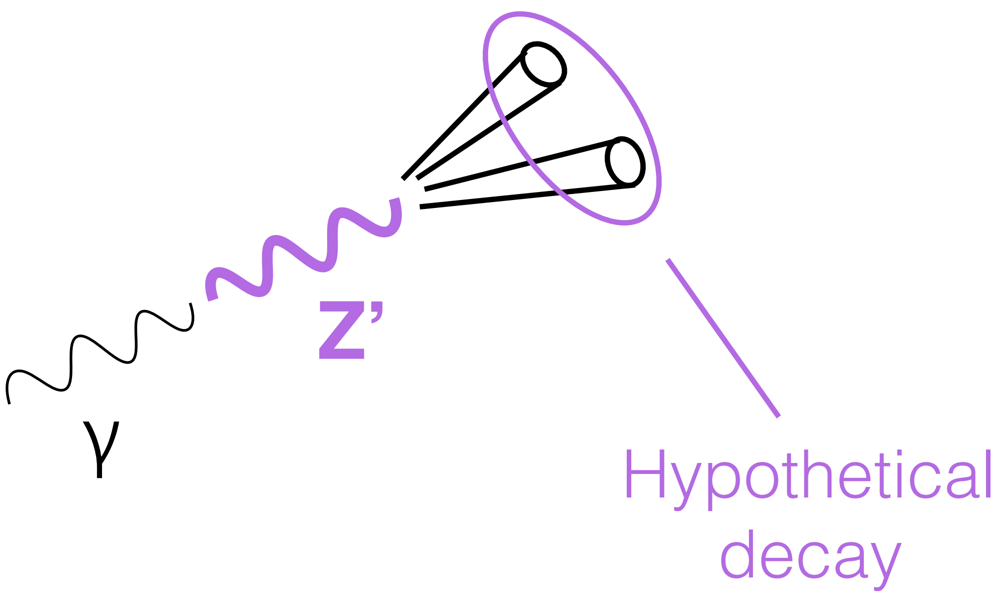
arXiv:1602.07727

Solution: Trigger on something else!



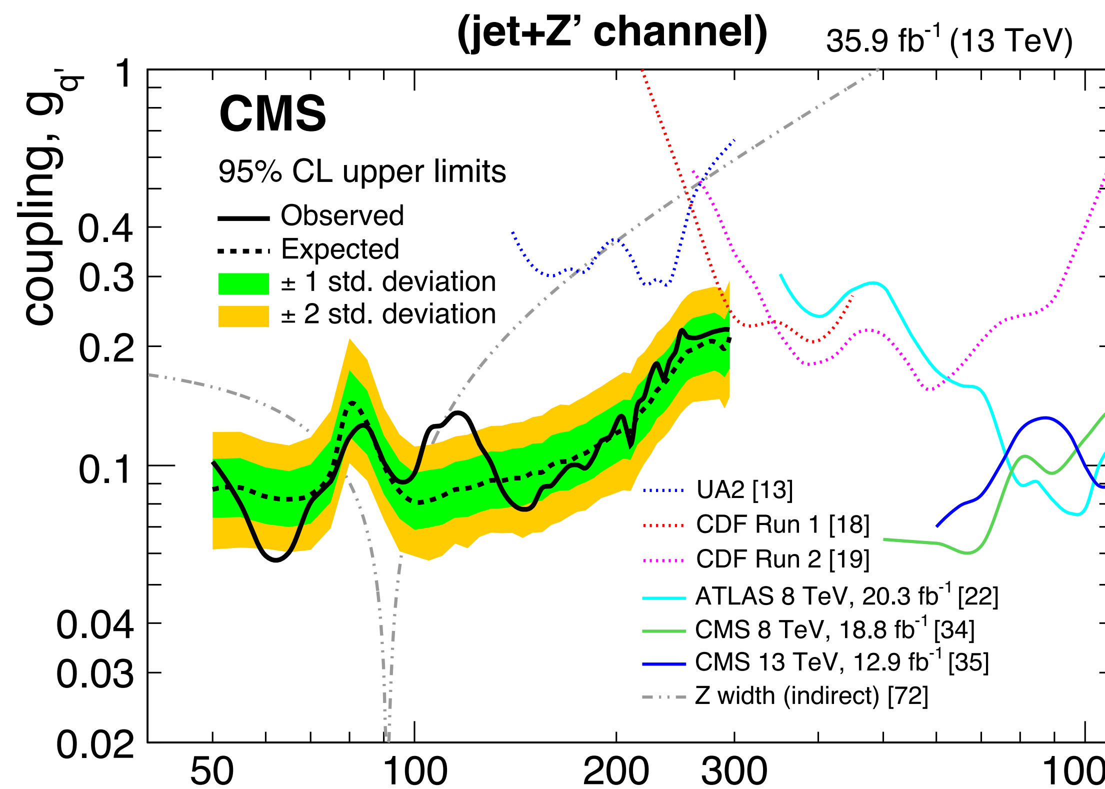
Analysis Applications

Low-mass leptophobic resonance

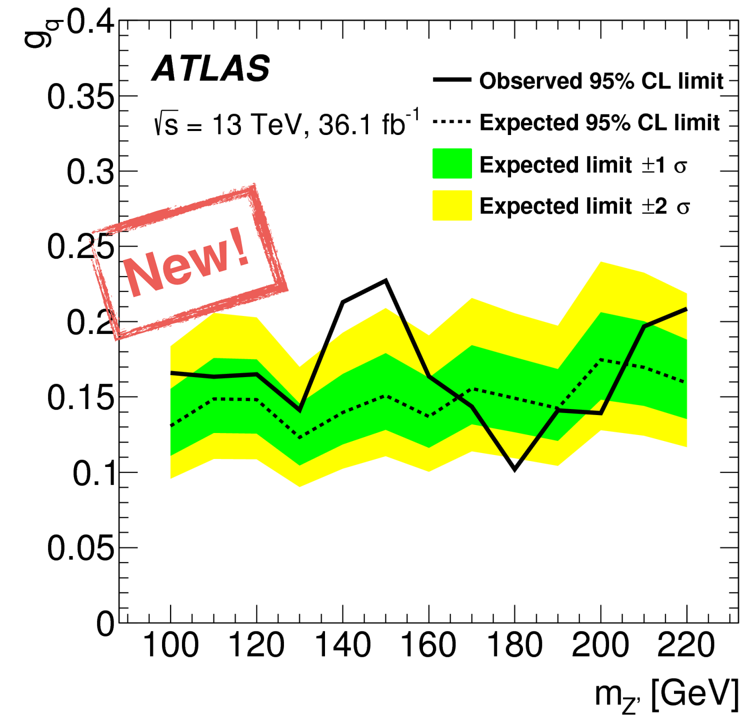


$$p_{\mathrm{T}}^{\gamma} \sim 150 \ \mathrm{GeV}$$

$$m_{Z'} \lesssim 200 \ \mathrm{GeV}$$



CMS-EXO-17-001

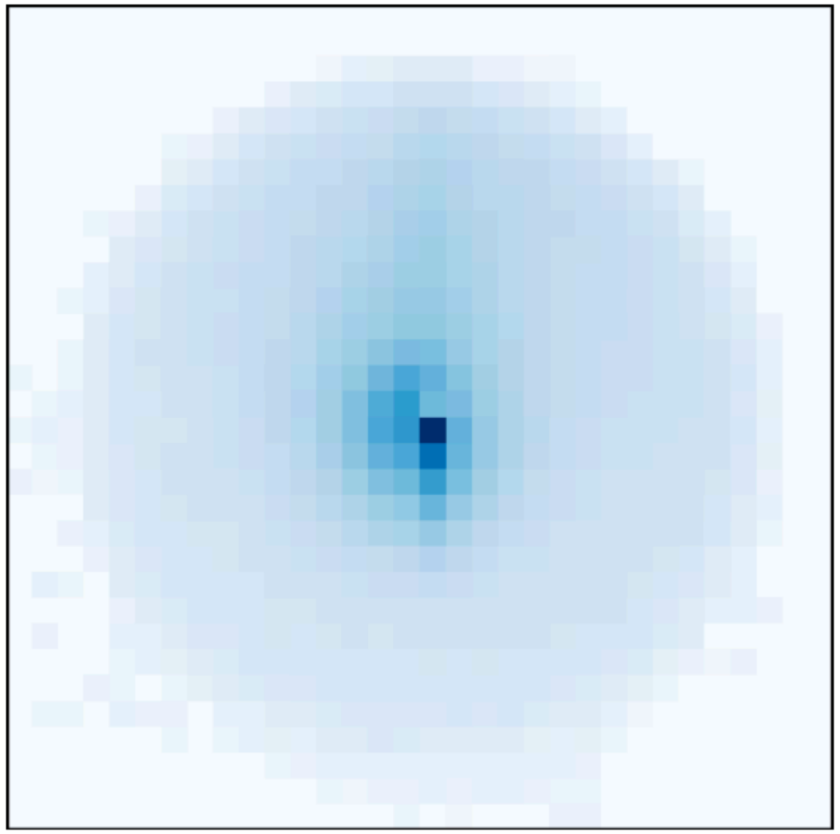


ATLAS-EXOT-2017-01

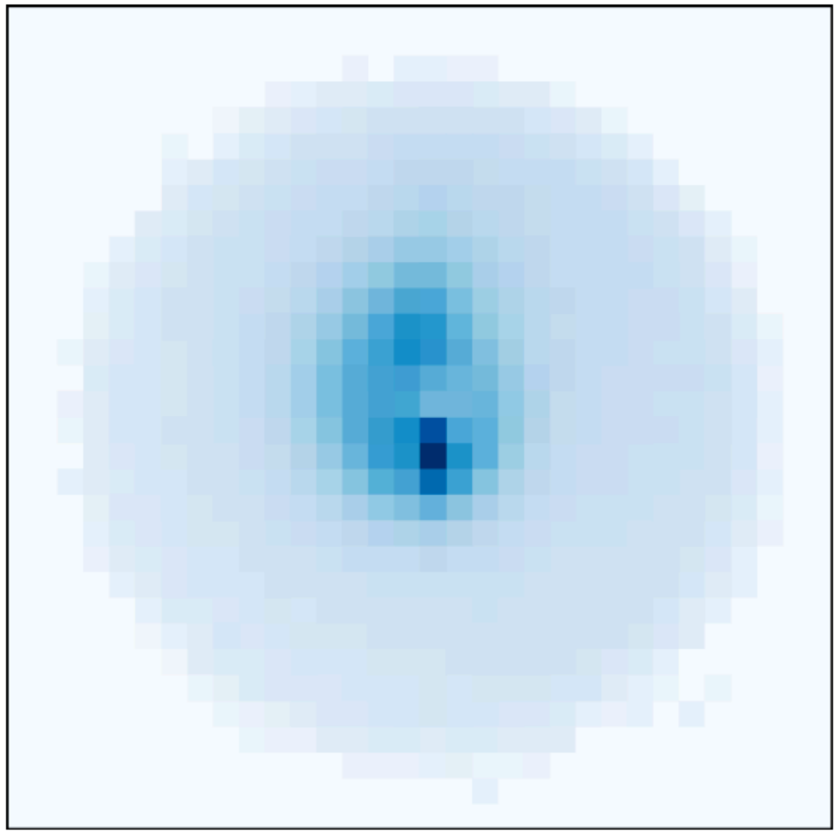
Jet Substructure

In addition to possible resonance mass, boosted jets have distinctive structure:

QCD jet



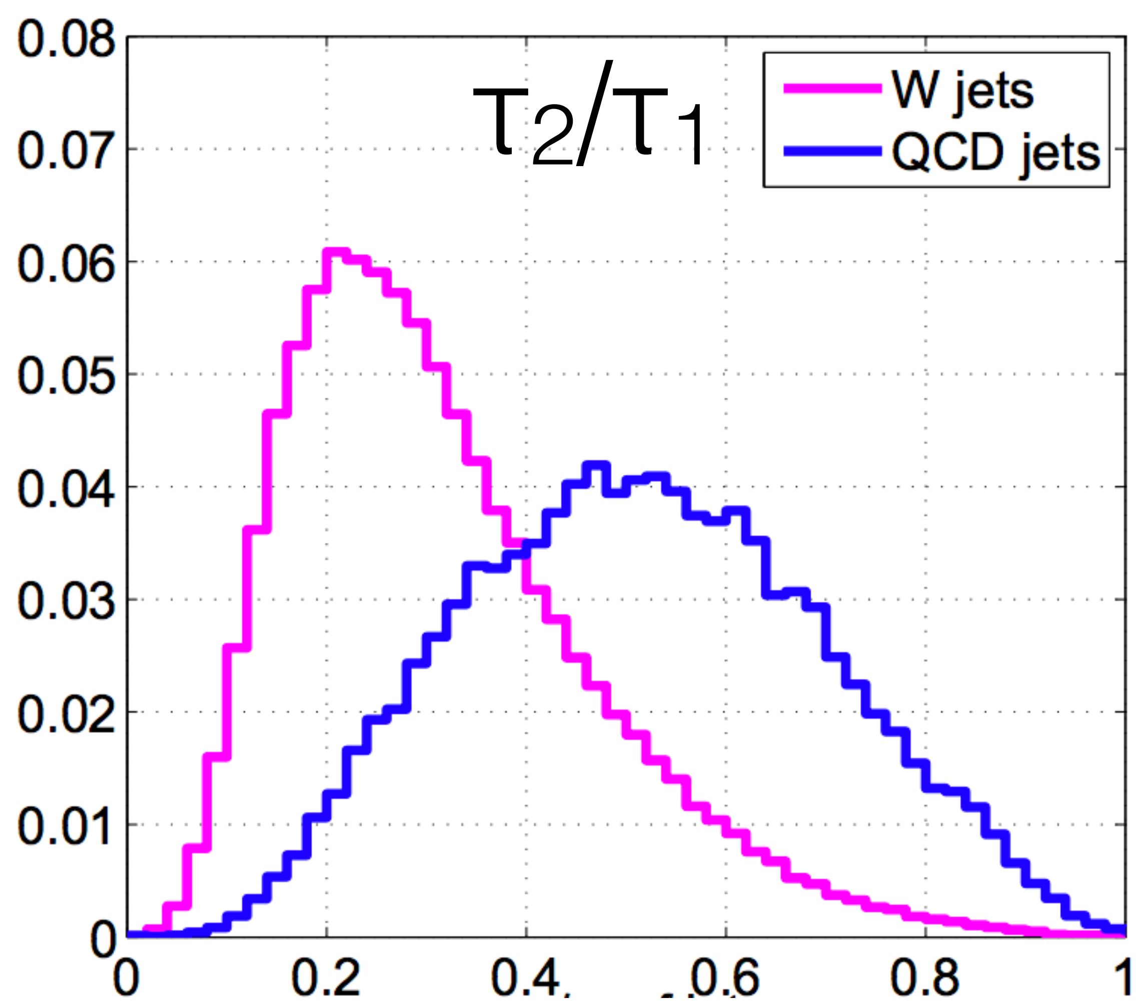
W jet

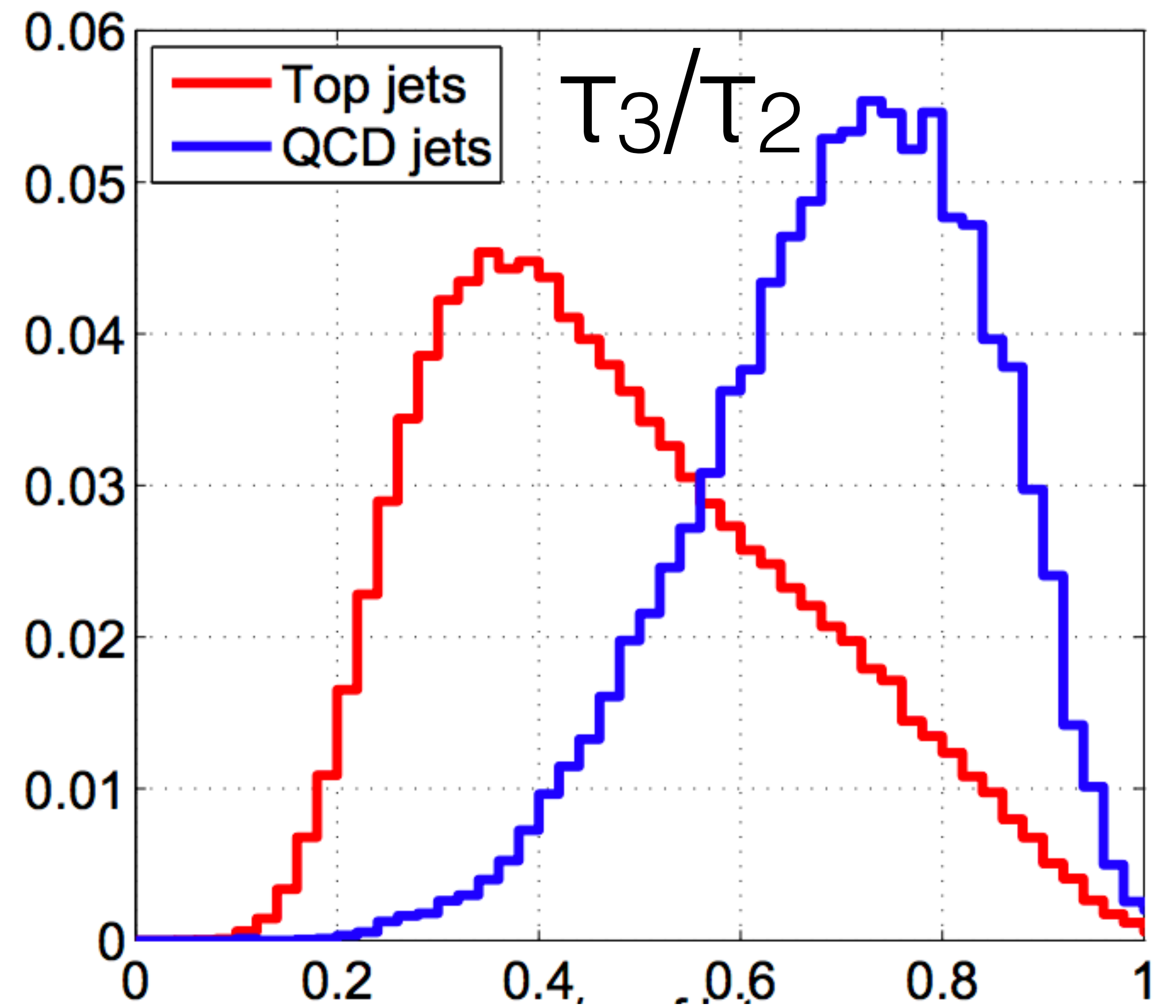


arXiv:1603.09349

Substructure Variables

- Many theoretically motivated tools to quantify jet substructure, e.g. N-subjettiness, ECF…

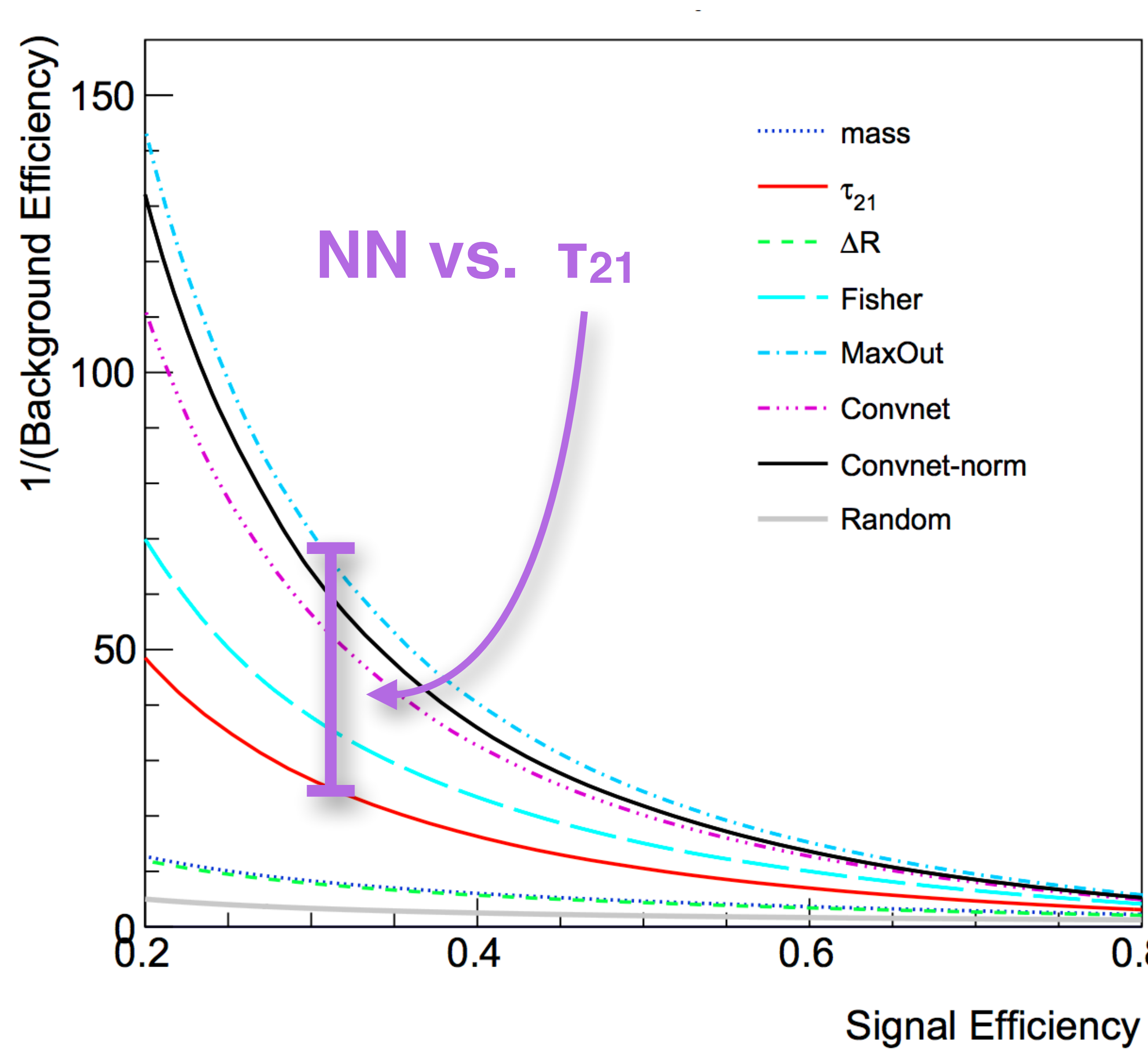




arXiv:1011.2268

Multivariate Taggers

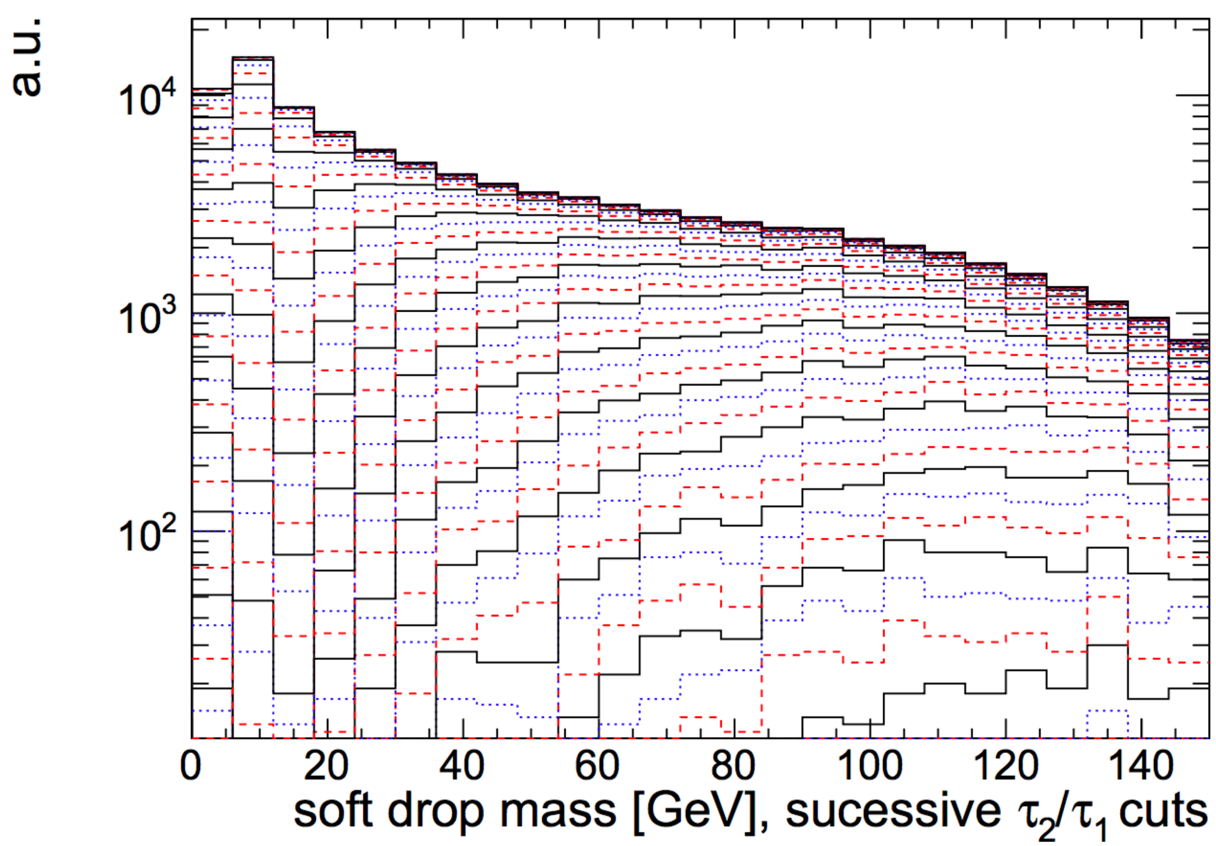
Multivariate taggers (BDT, NN) in general can do even better!



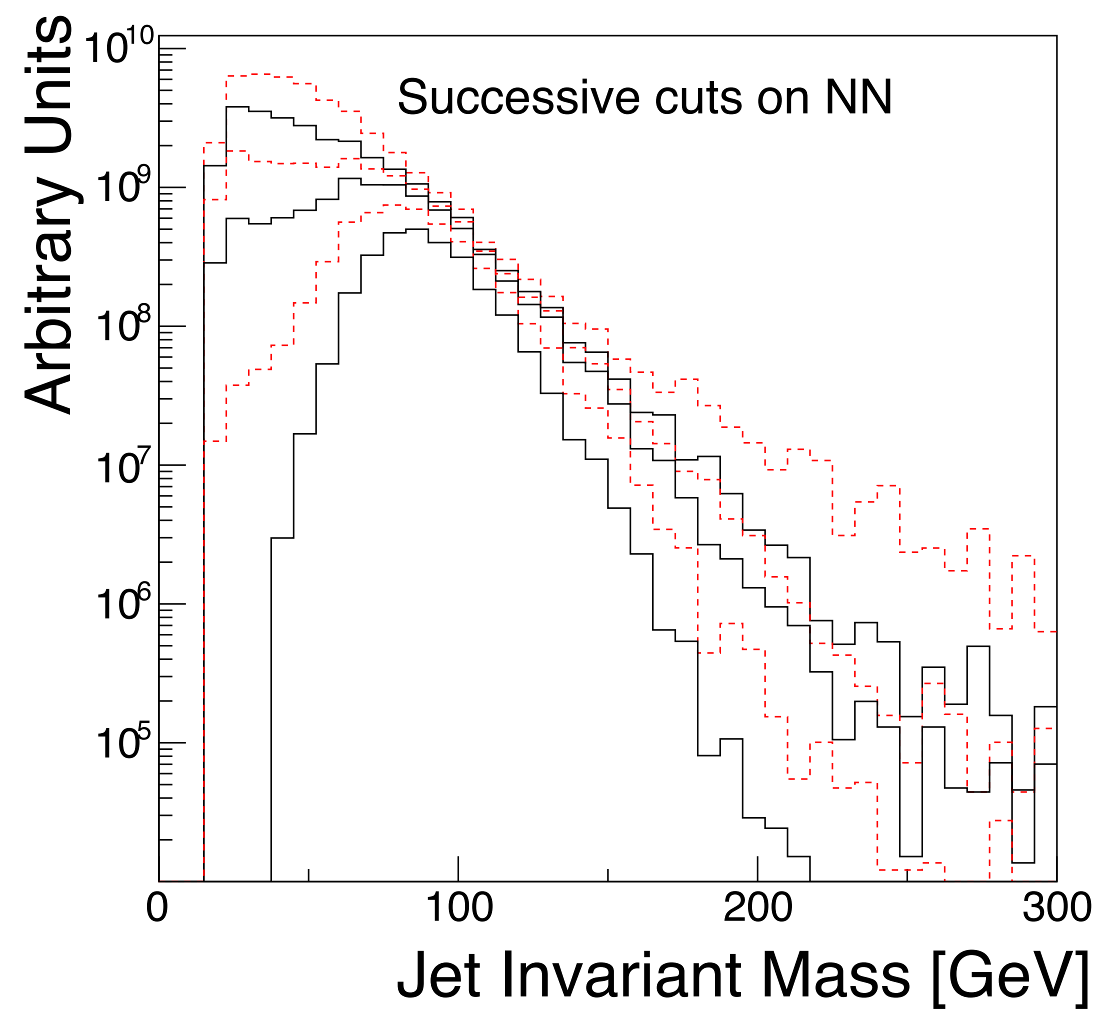
arXiv:1511.05190

Mass Correlation

Problem: cutting on taggers **distorts mass spectrum**



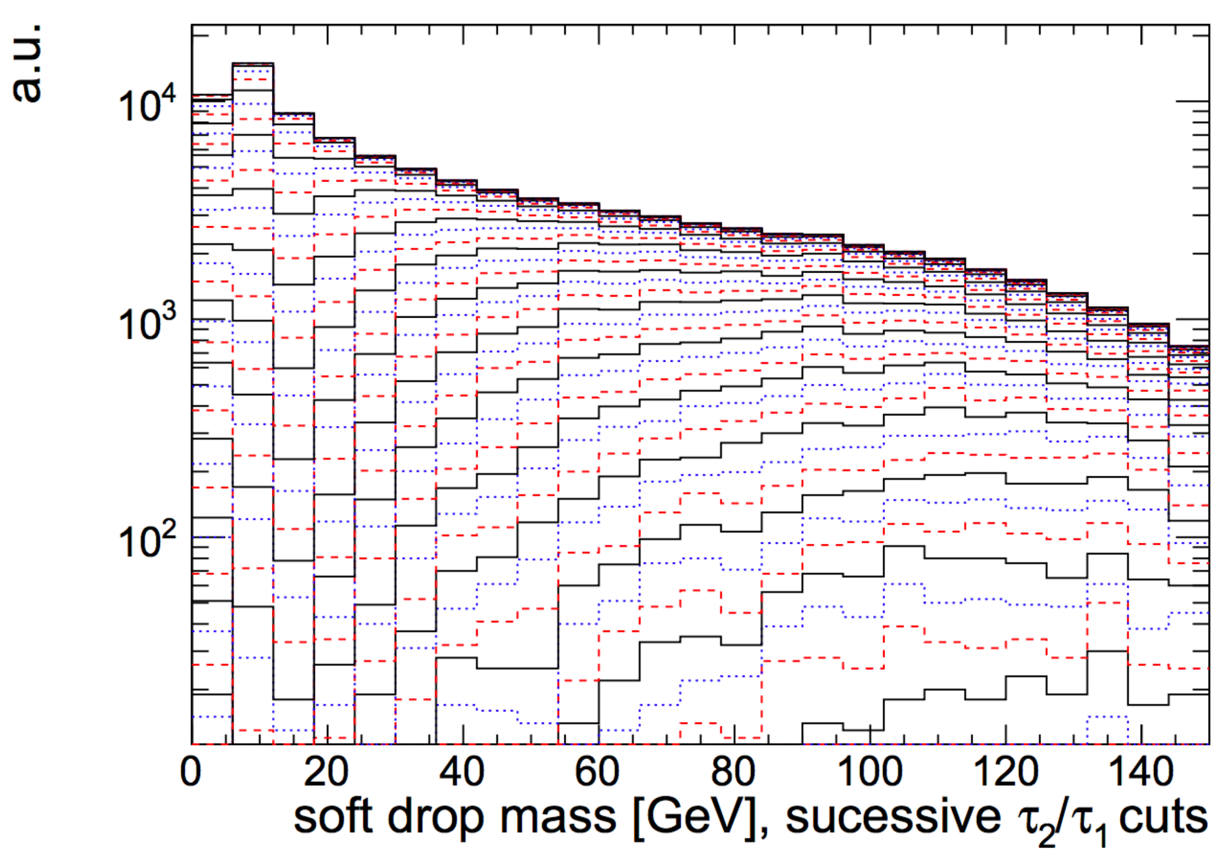
arXiv:1603.00027



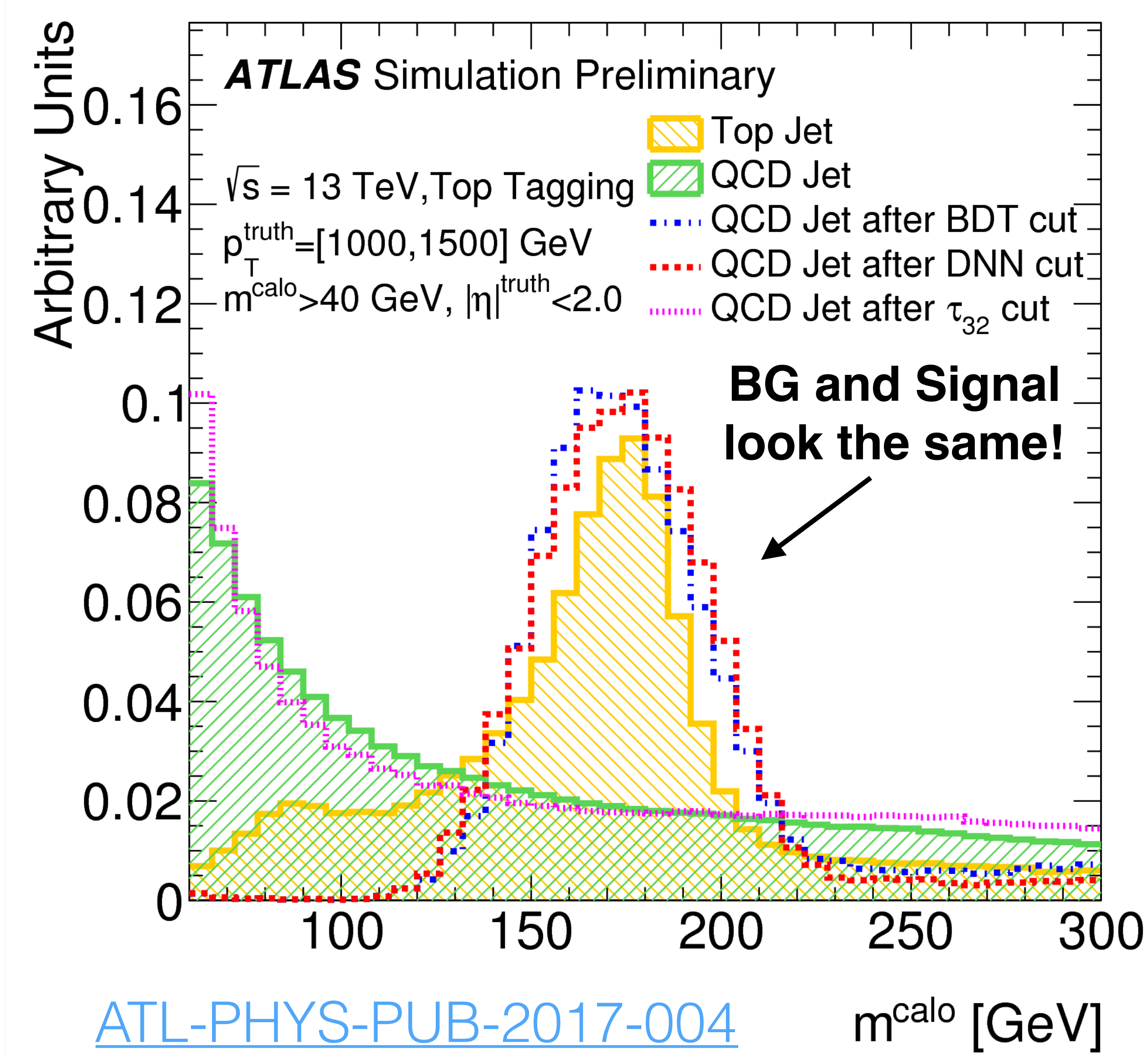
arXiv:1703.03507

Mass Correlation

Problem: cutting on taggers **distorts mass spectrum**



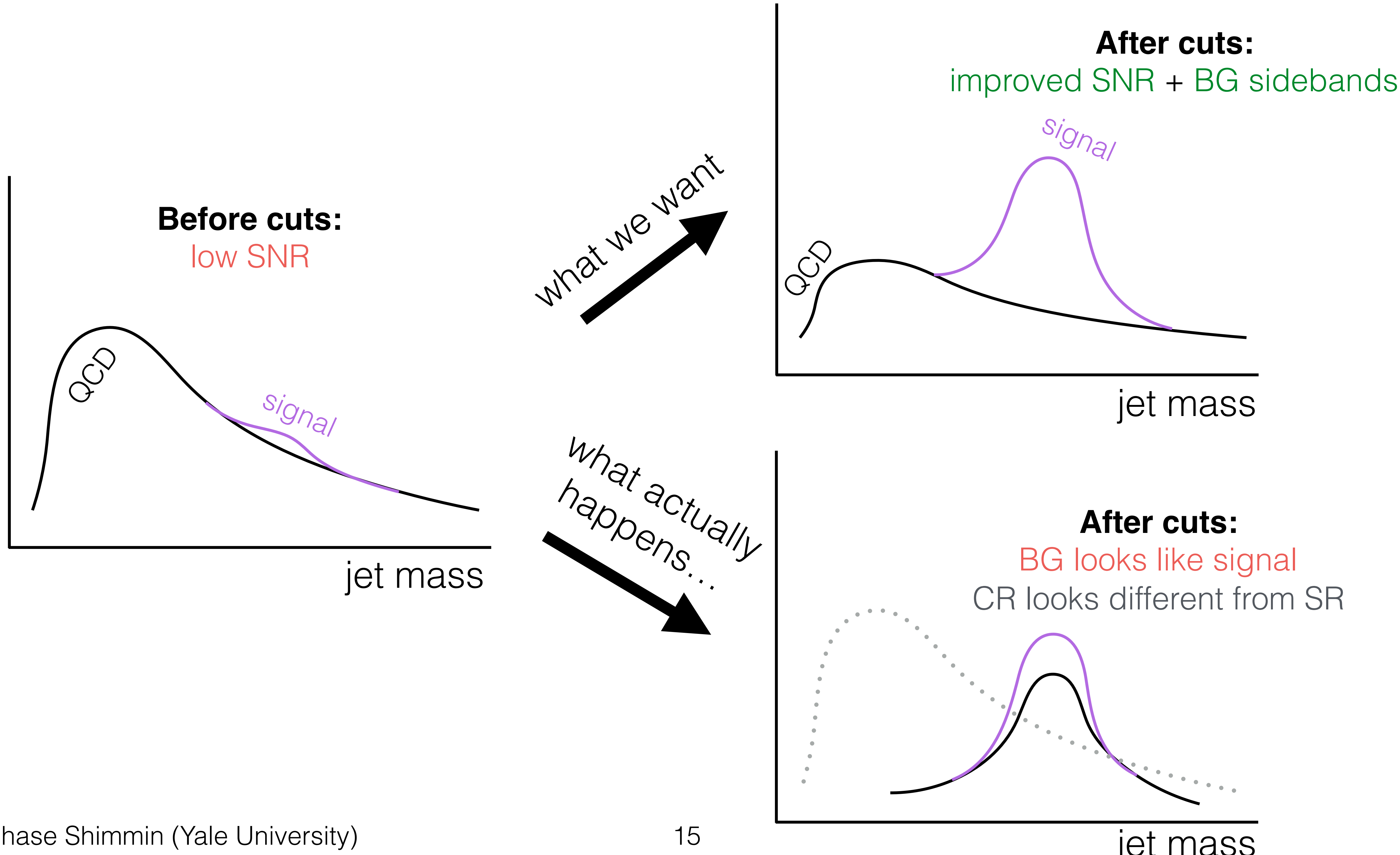
arXiv:1603.00027



ATL-PHYS-PUB-2017-004

Mass Correlation

Correlation with the observable of interest is bad!

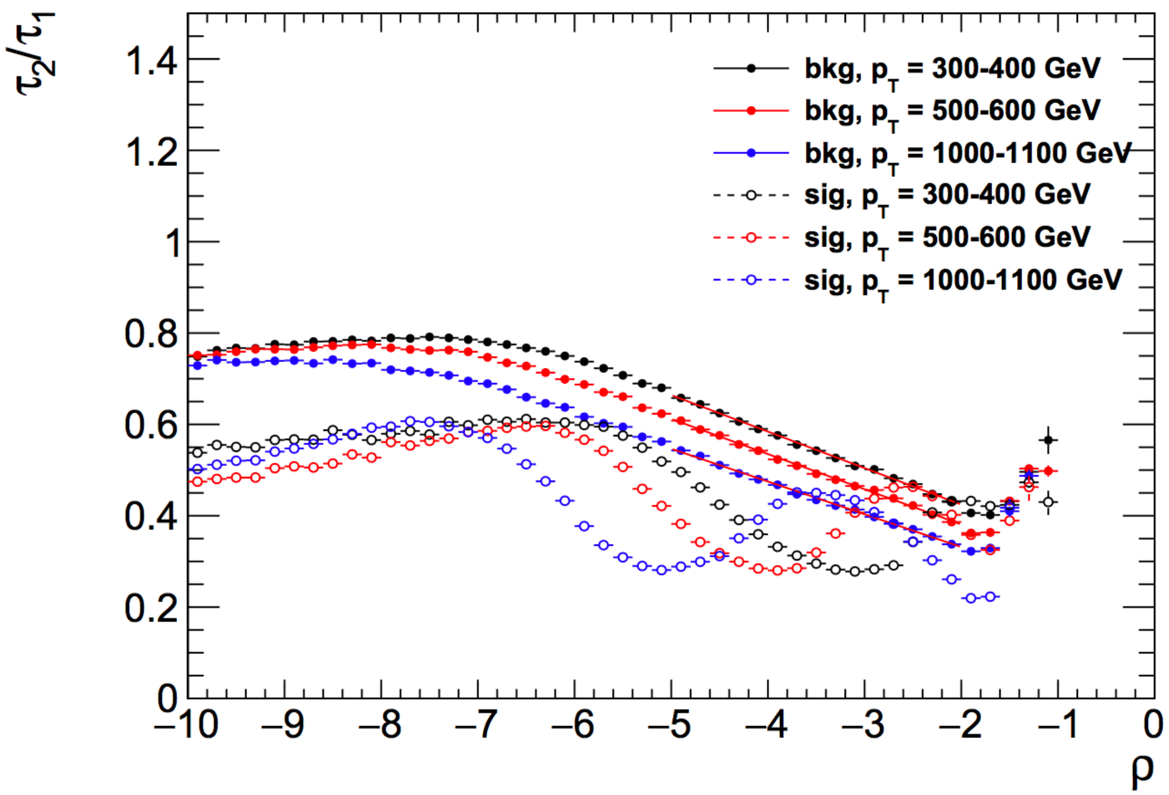


De-Correlation

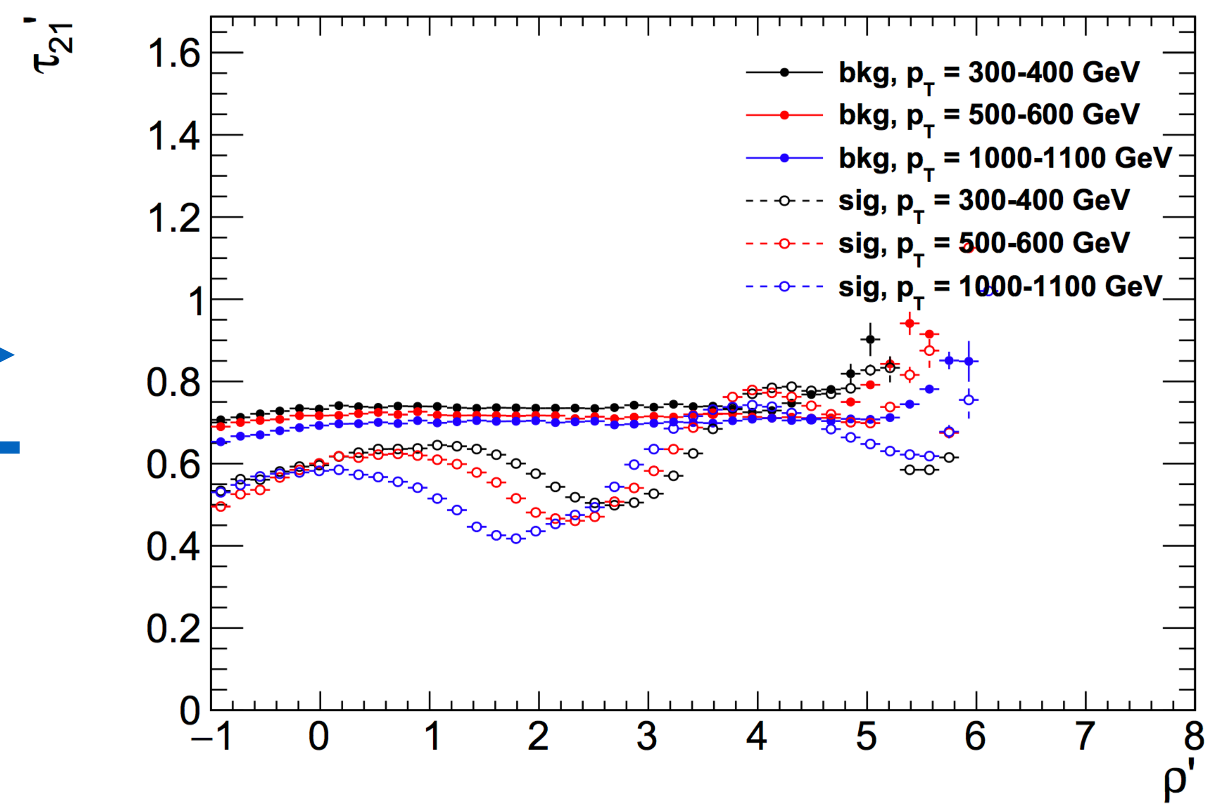
“DDT” (Designing Decorrelated Taggers) paper:

Proposes explicit transformation to decorrelate τ_{21} variable

(τ_{21} from N-subjetiness substructure)



DDT

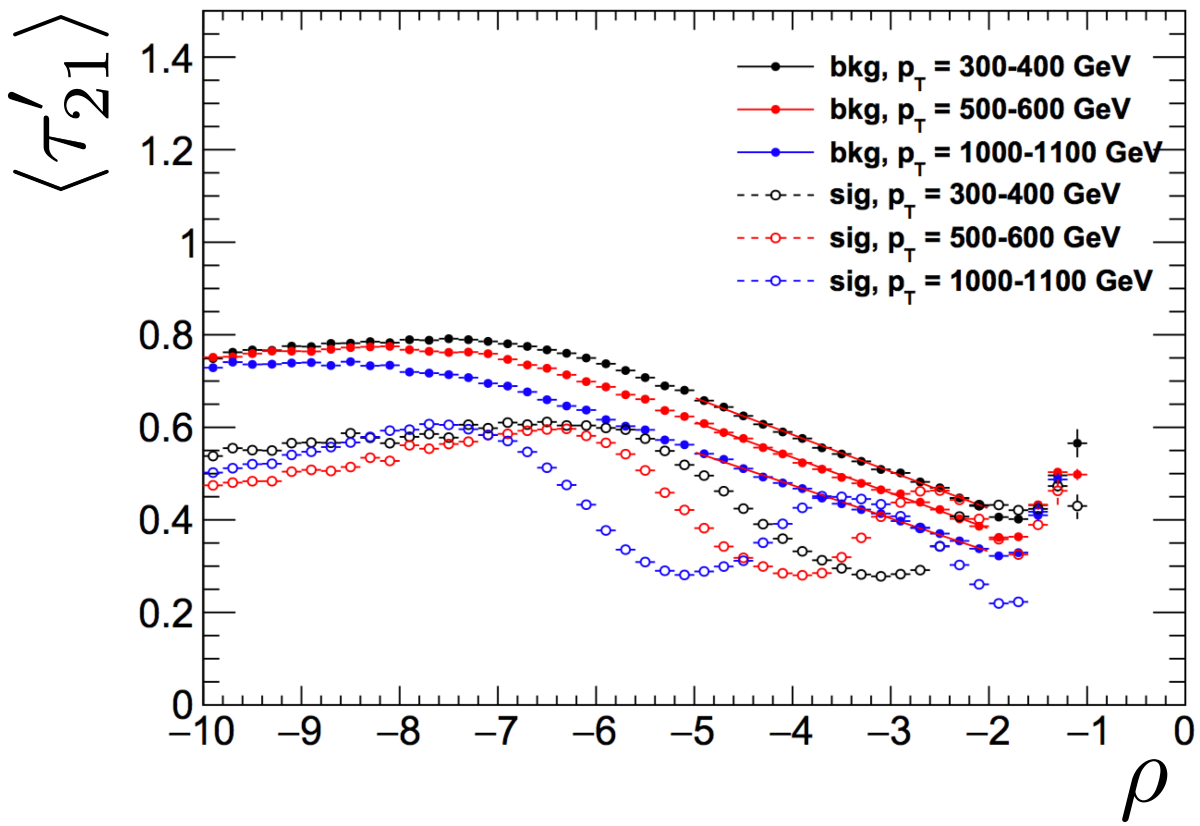


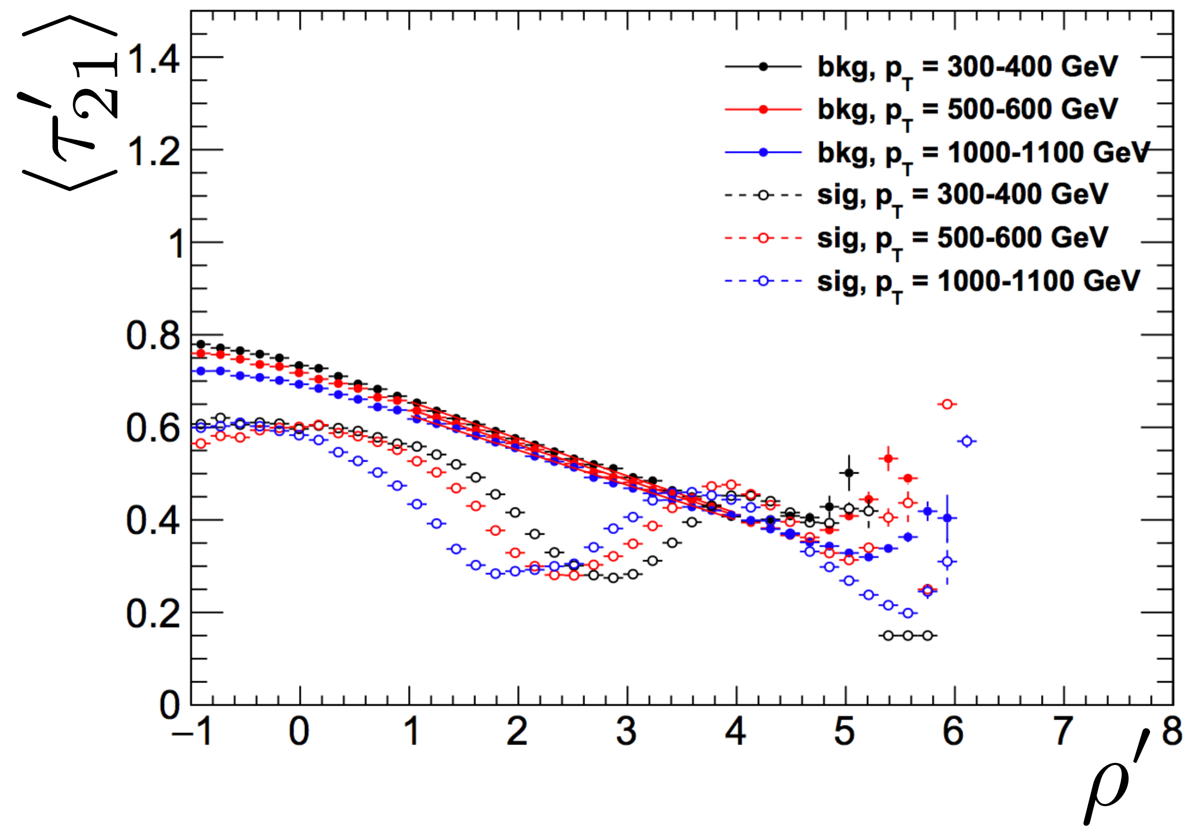
arXiv:1603.00027

DDT Method

arXiv:1603.00027

First, dependence of τ_{21} on p_T removed





$$\rho = \log(m^2/p_{\mathrm{T}}^2)$$

$$\rho' = \rho + \log\frac{p_{\mathrm{T}}}{\mu} = \log\left(\frac{m^2}{p_{\mathrm{T}}\mu}\right)$$

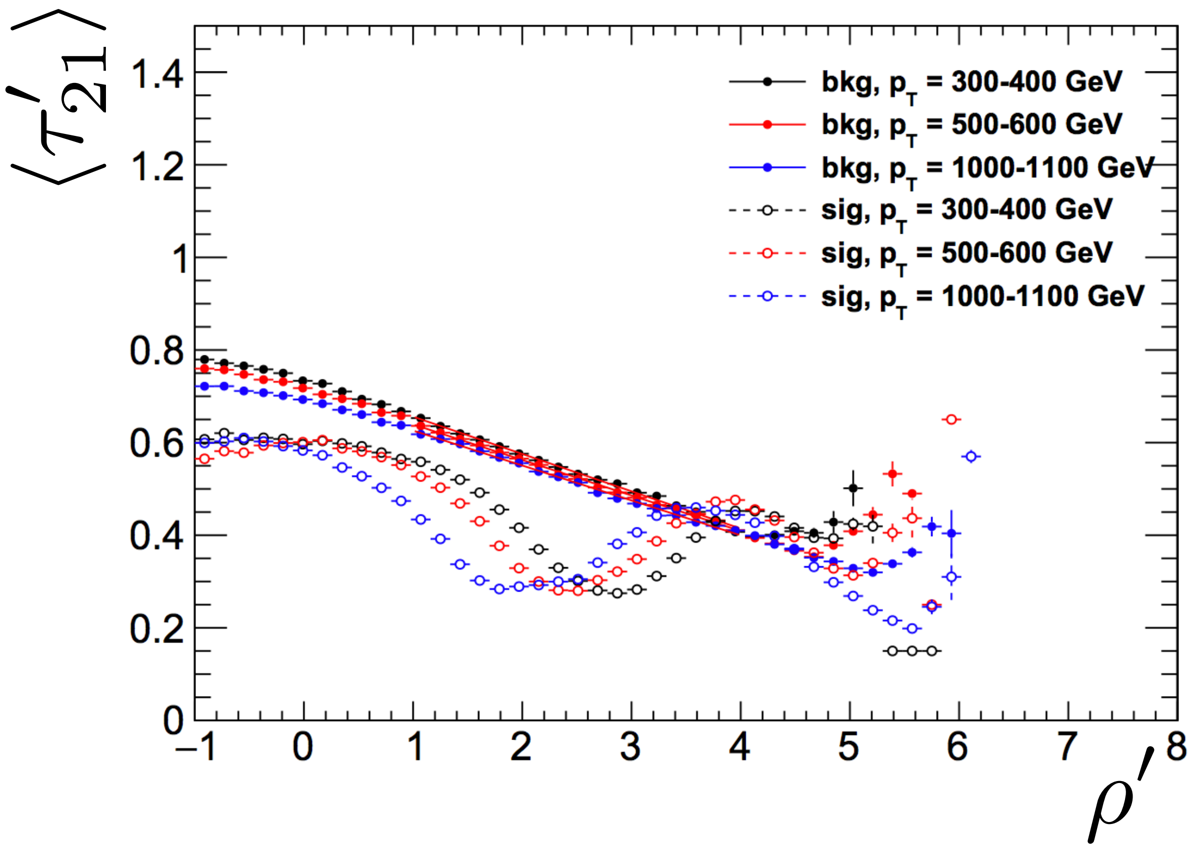
(heuristically)

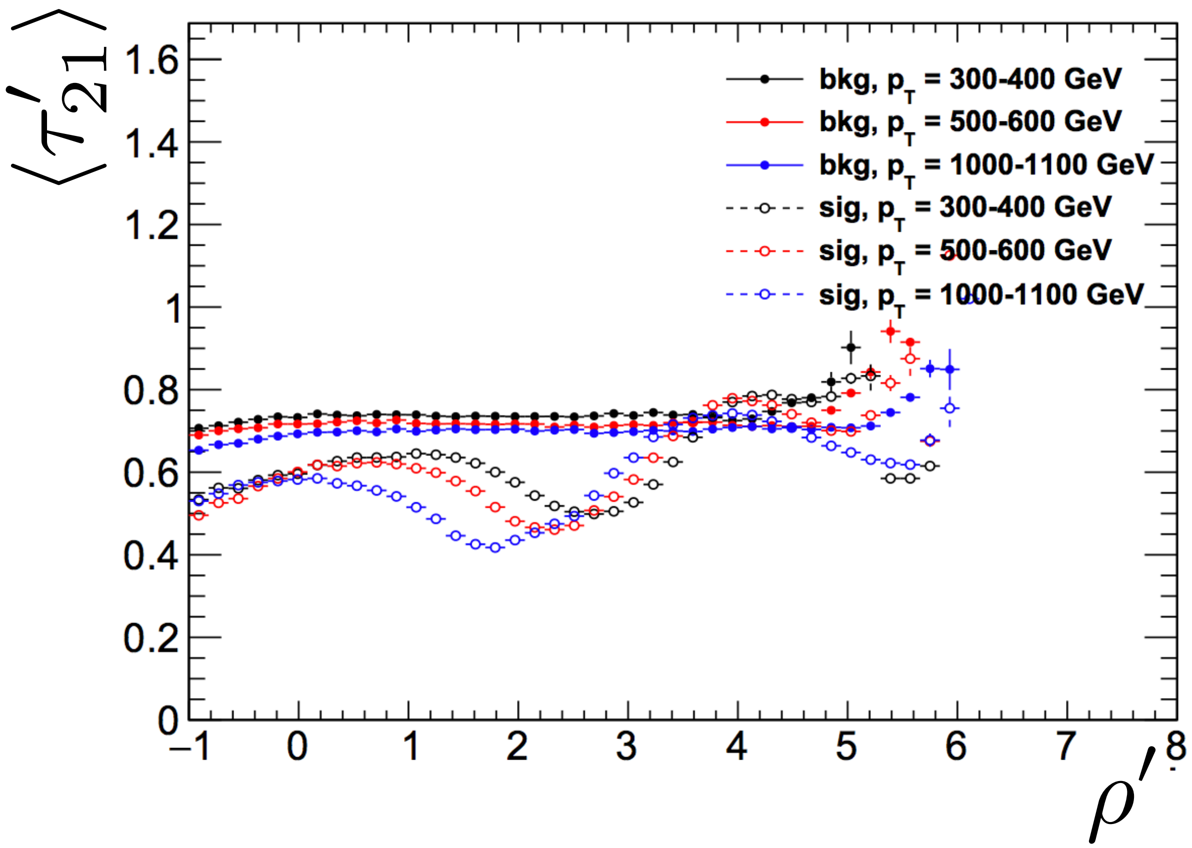
$$\mu \sim 1\ \mathrm{GeV}$$

DDT Method

arXiv:1603.00027

Then, linear trend explicitly subtracted





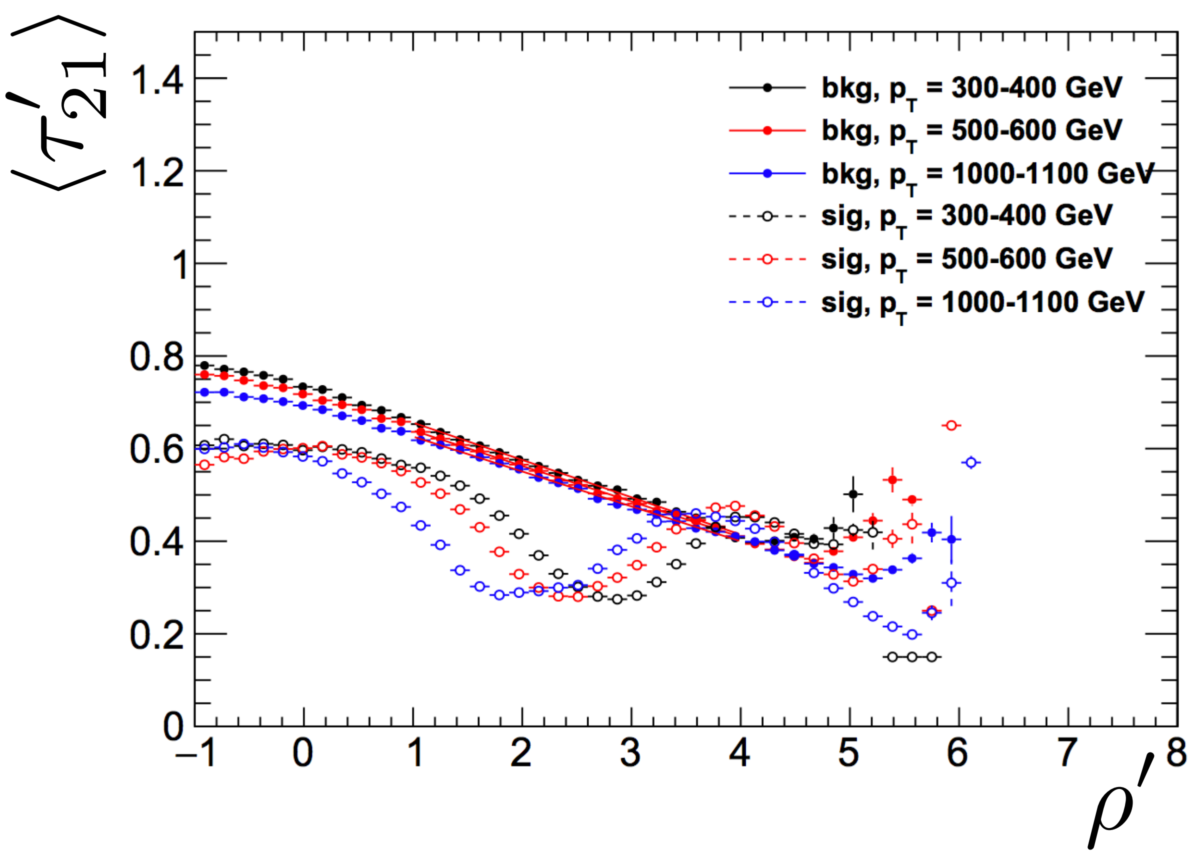
$$\tau'_{21} = \tau_{21} - M \times \rho'$$

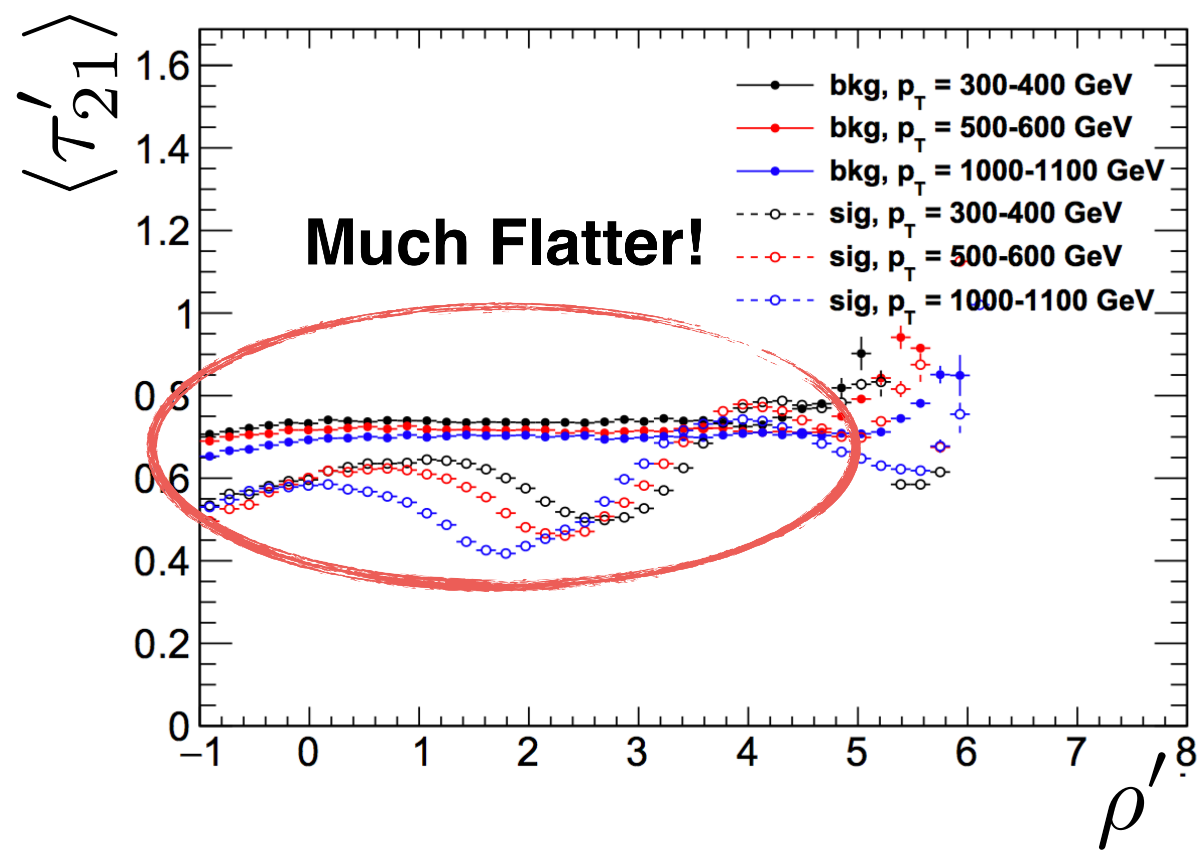
Determined empirically

DDT Method

arXiv:1603.00027

Then, linear trend explicitly subtracted

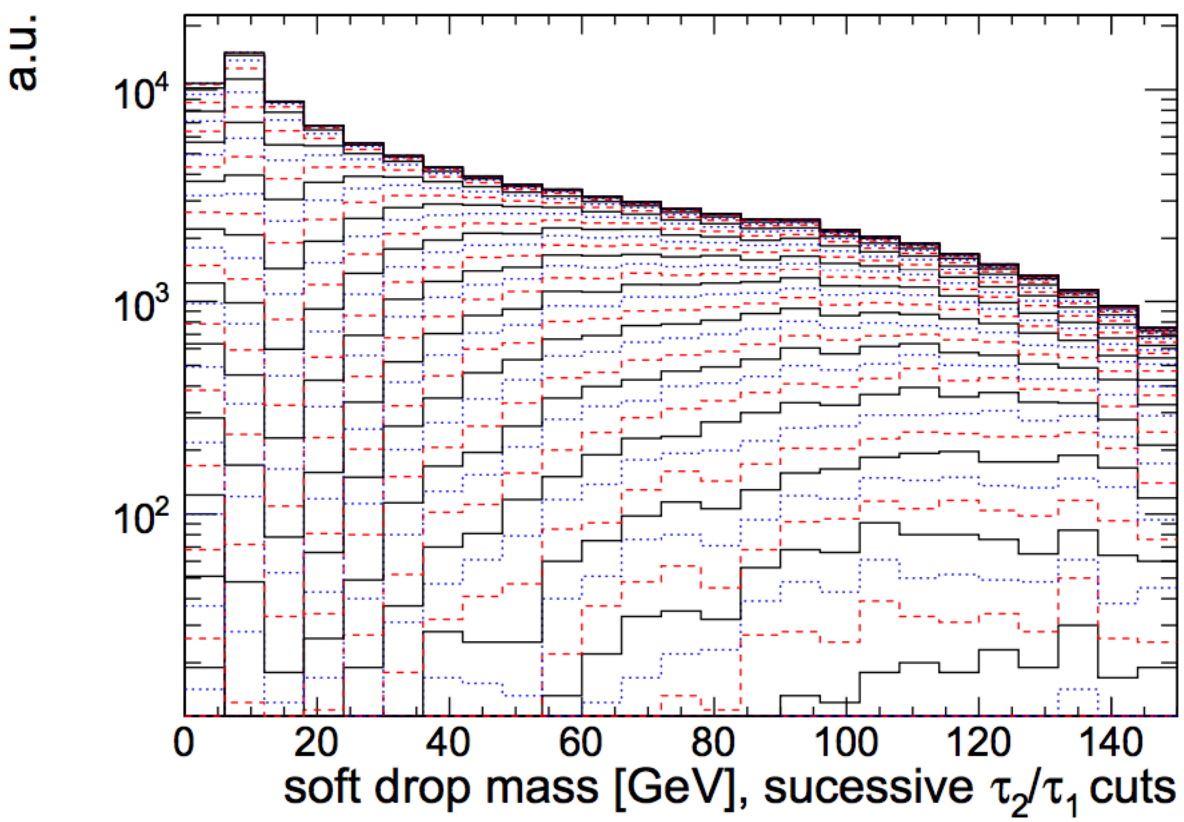


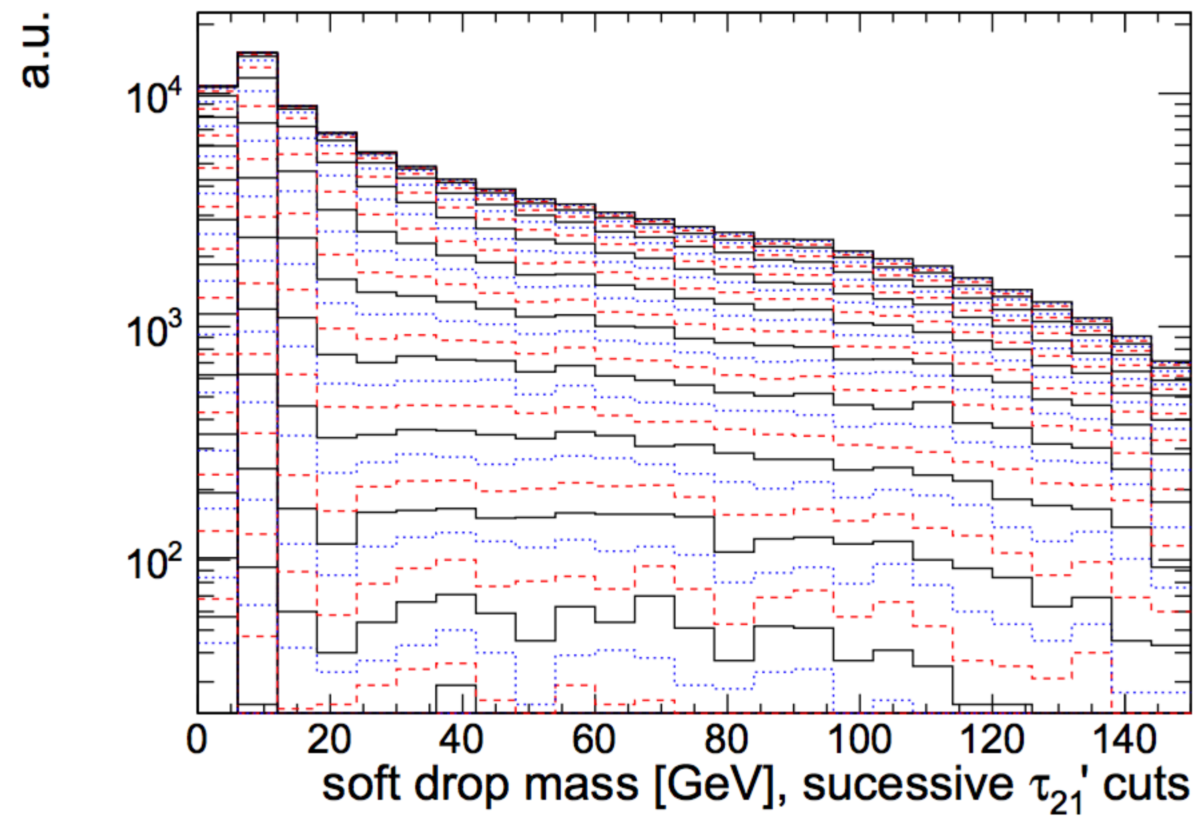


$$\tau'_{21} = \tau_{21} - M \times \rho'$$

Determined empirically

DDT Method

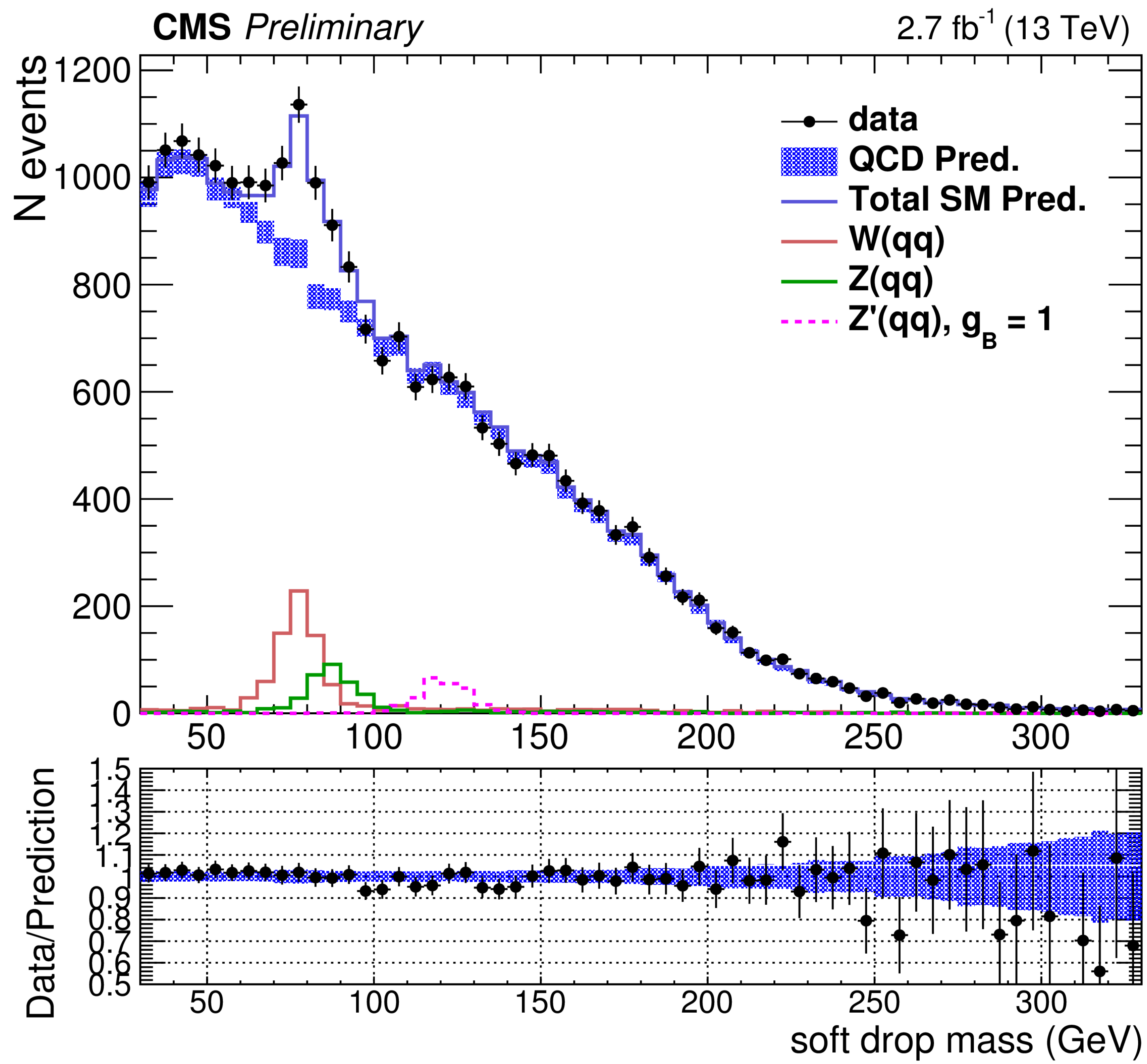




Much less sculpting of background

DDT Method

- DDT method used very successfully by CMS in low-mass Z' search

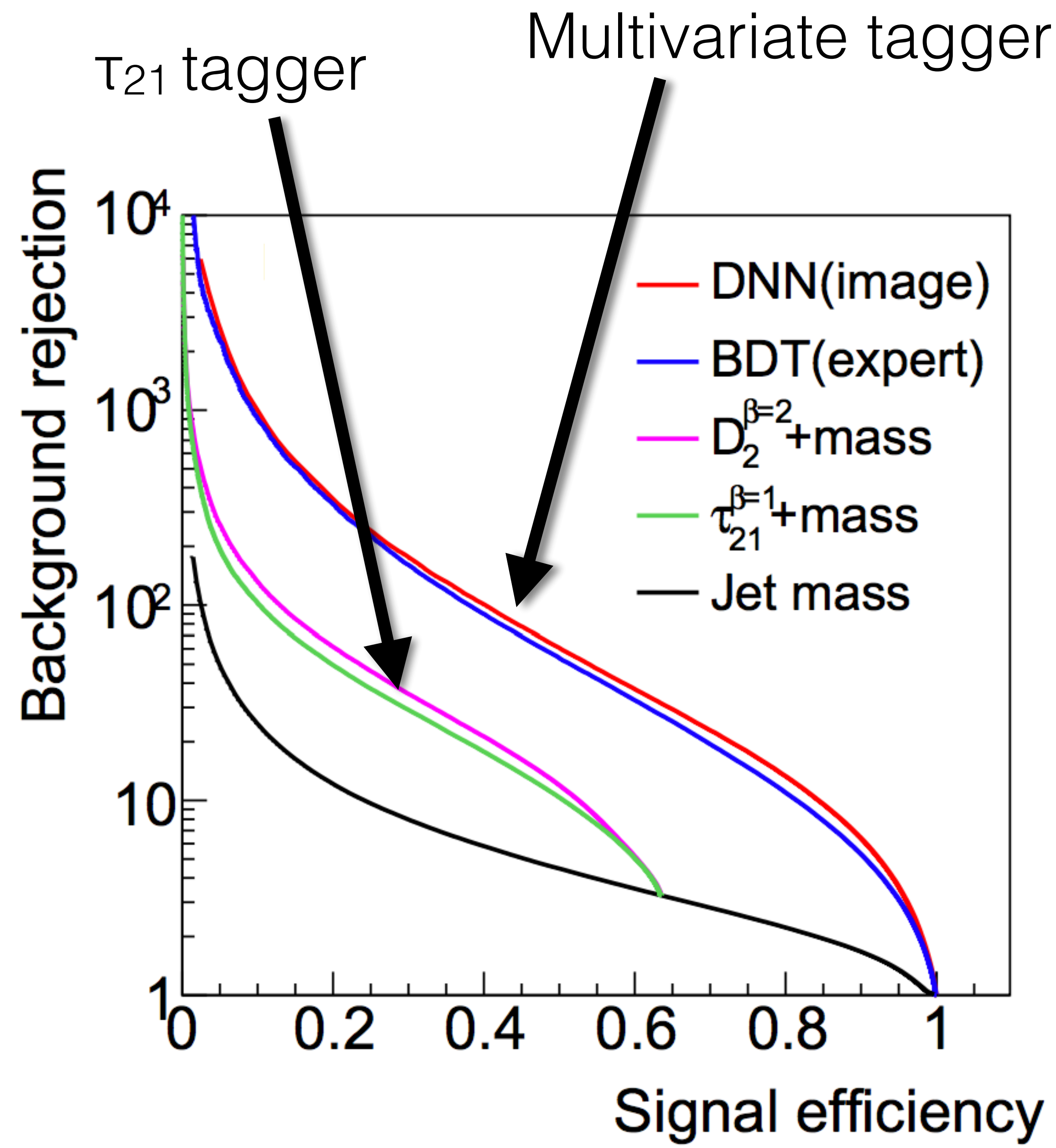


CMS-PAS-EXO-16-030

DDT Method

However:

- It has been shown that combining more information in tagger gives better results
- DDT is doesn't seem to work well for other variables
- Difficult to generalize to multiple variables



arXiv:1603.09349

Generalization

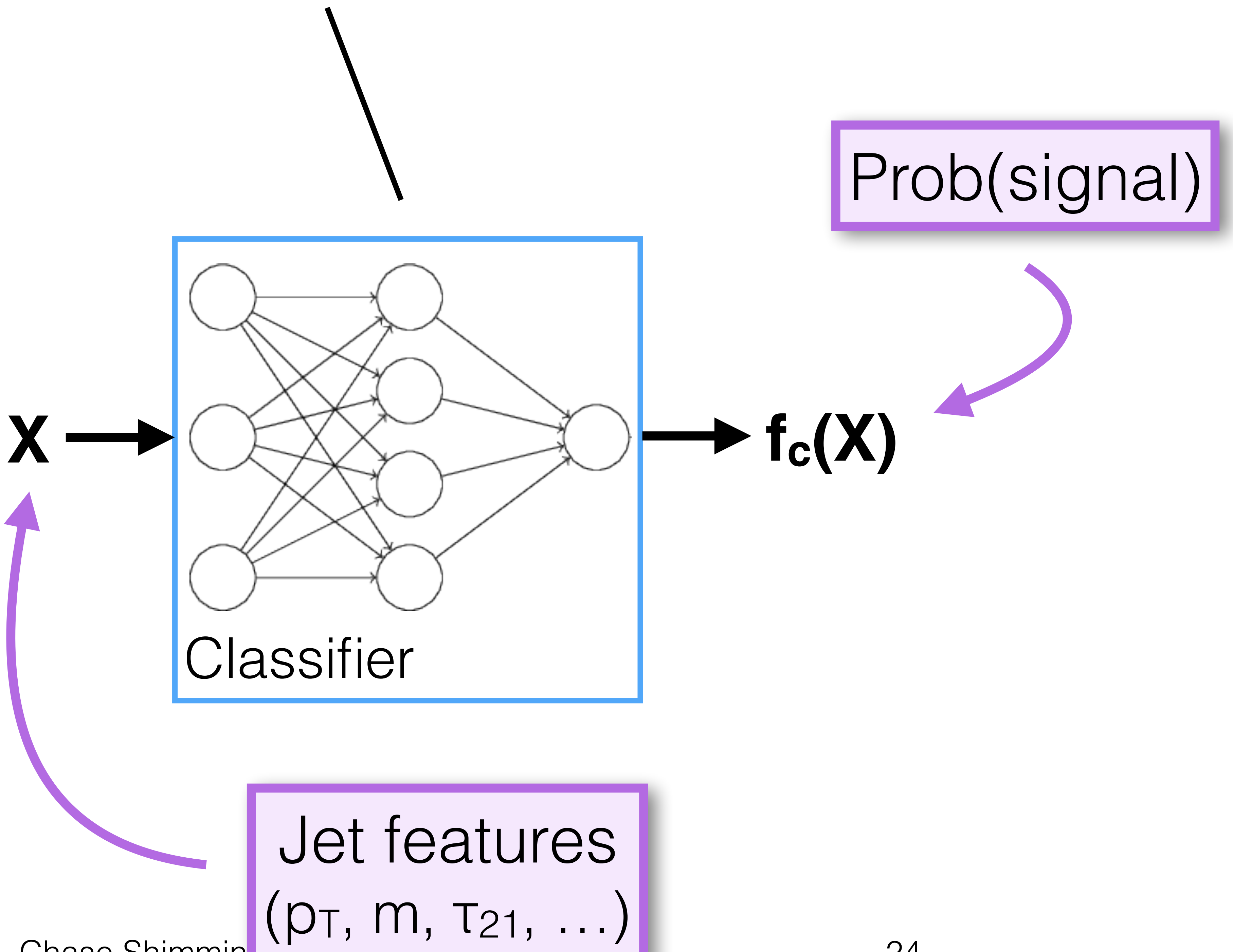
- We would like to **generalize** this decorrelation approach for arbitrary classifiers
- Some proposed approaches:
  - multivariate DDT via PCA arXiv:1603.00027
  - uGBoost: add loss to enforce “flatness” arXiv:1410.4140
  - ★ Adversarial “pivot” / domain adaptation: arXiv:1611.01046

We investigate this approach

Adversarial Decorrelation

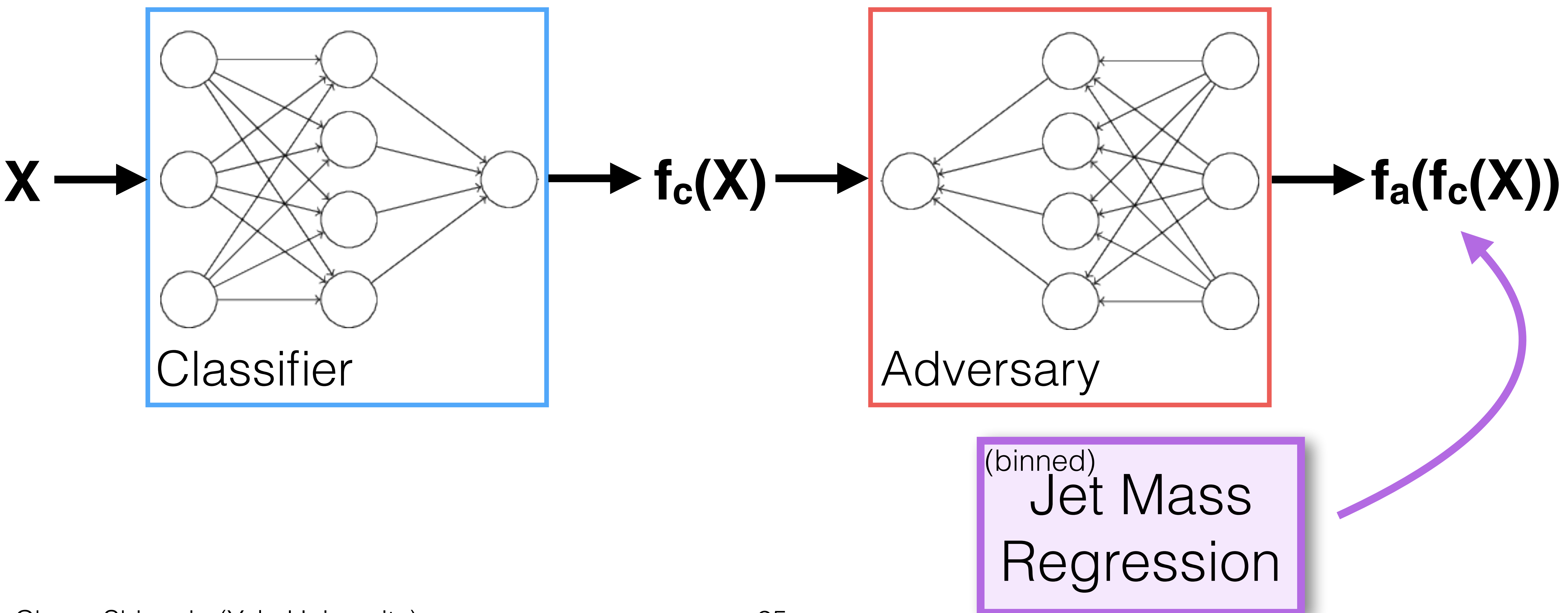
Basic idea:

Classifier is trained to identify signal jets

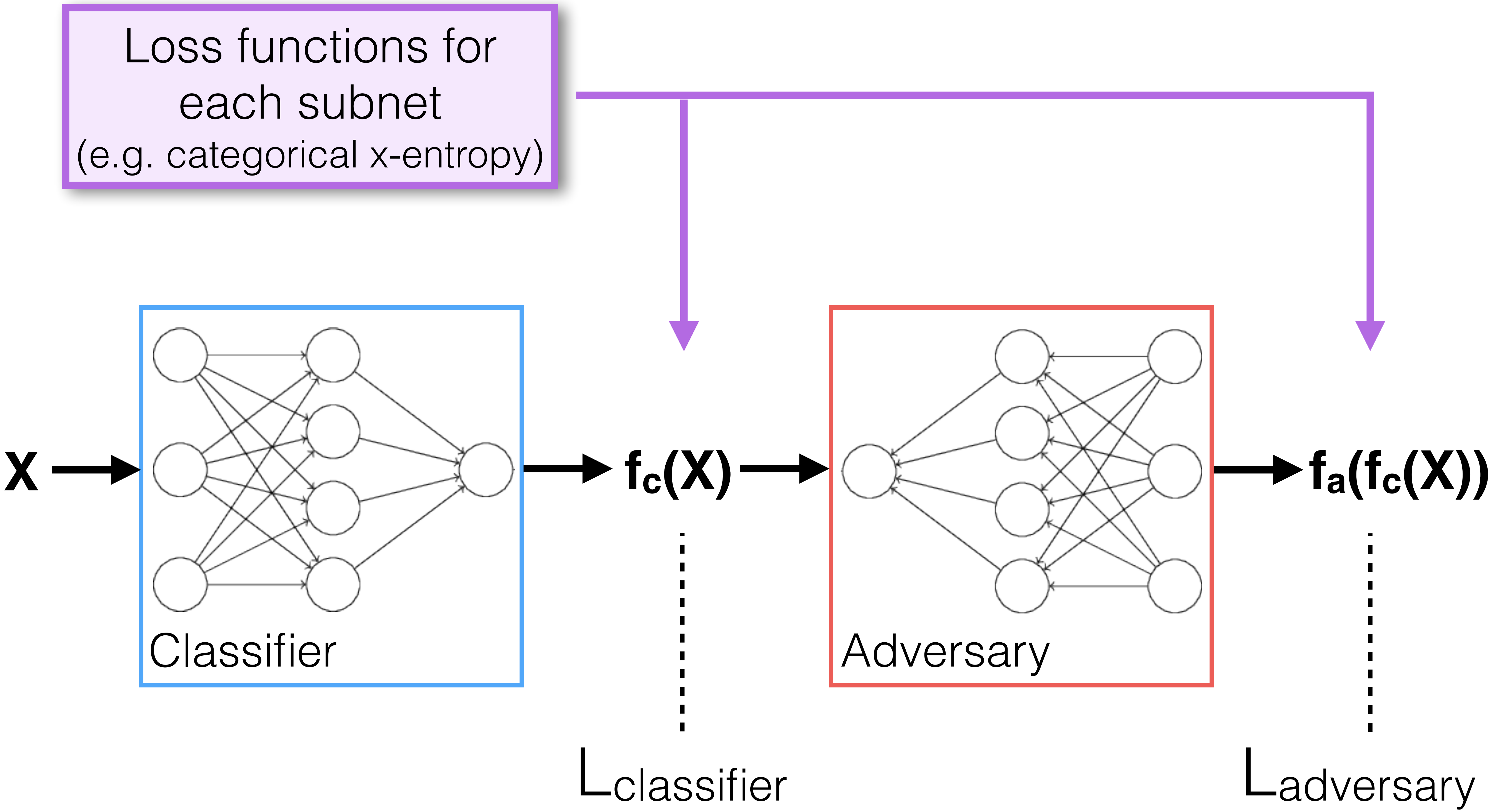


Adversarial Decorrelation

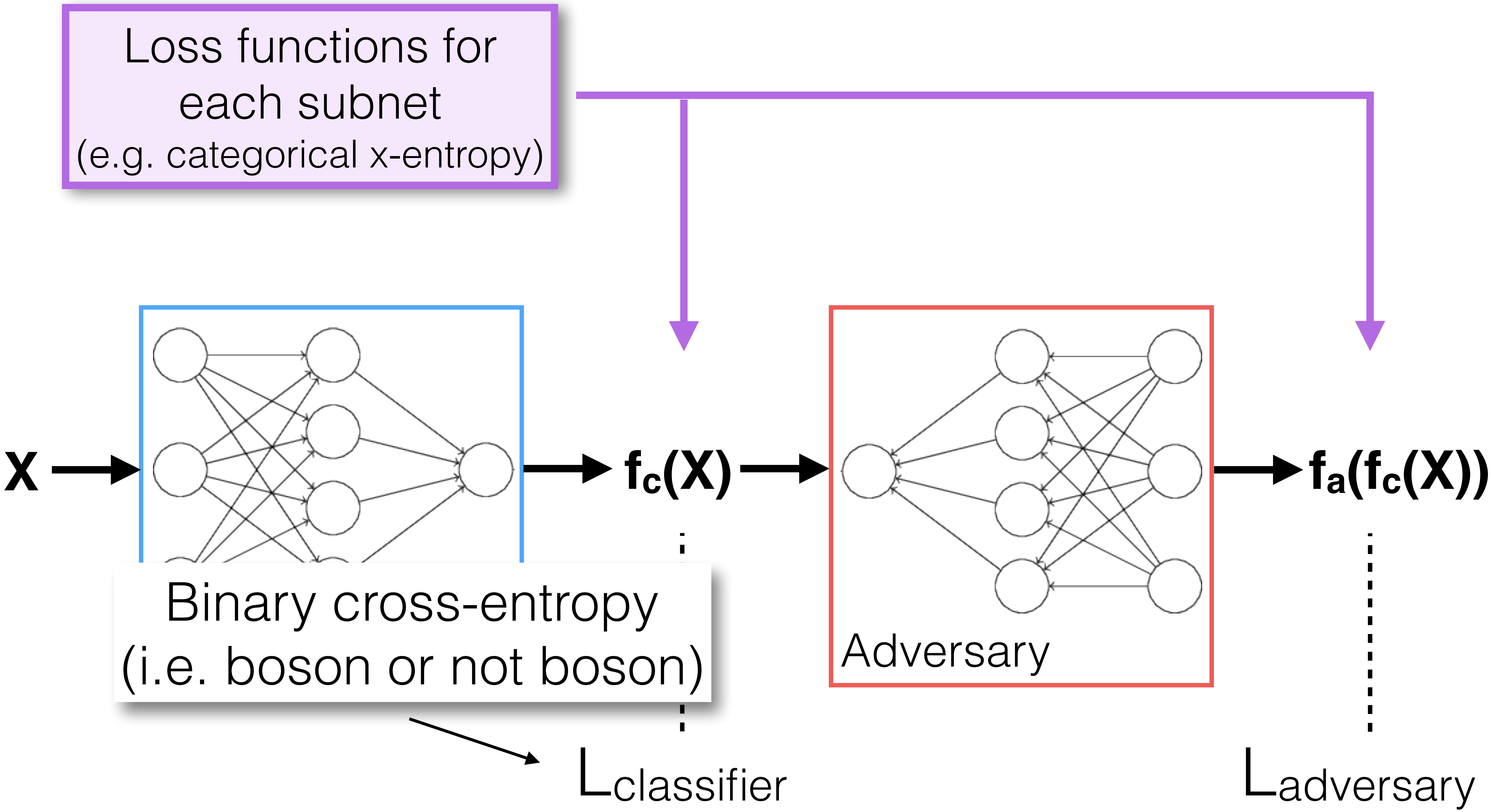
Adversary is trained to predict jet mass



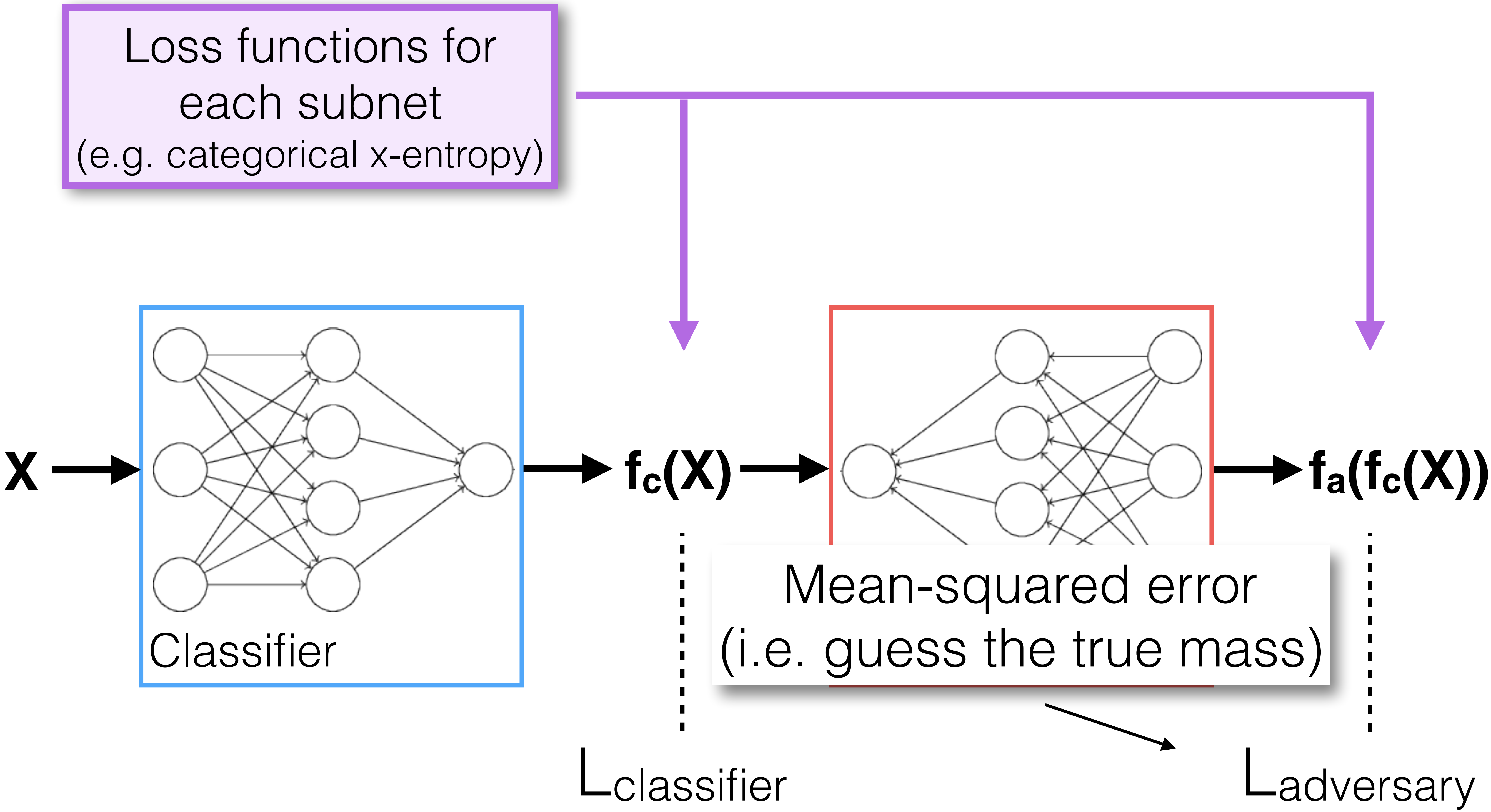
Adversarial Decorrelation



Adversarial Decorrelation



Adversarial Decorrelation



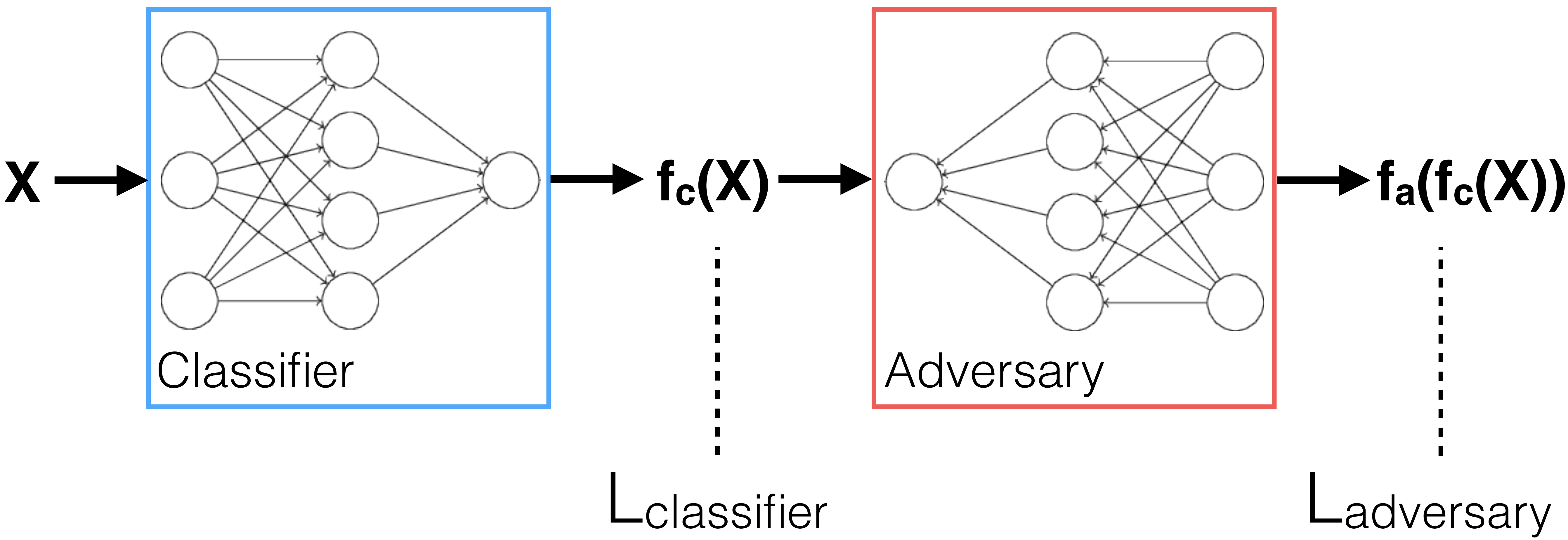
Adversarial Decorrelation

Simultaneously minimize:

$$L_{adversary}$$

and

$$L_{tagger} = L_{classifier} - \lambda L_{adversary}$$



Adversarial Decorrelation

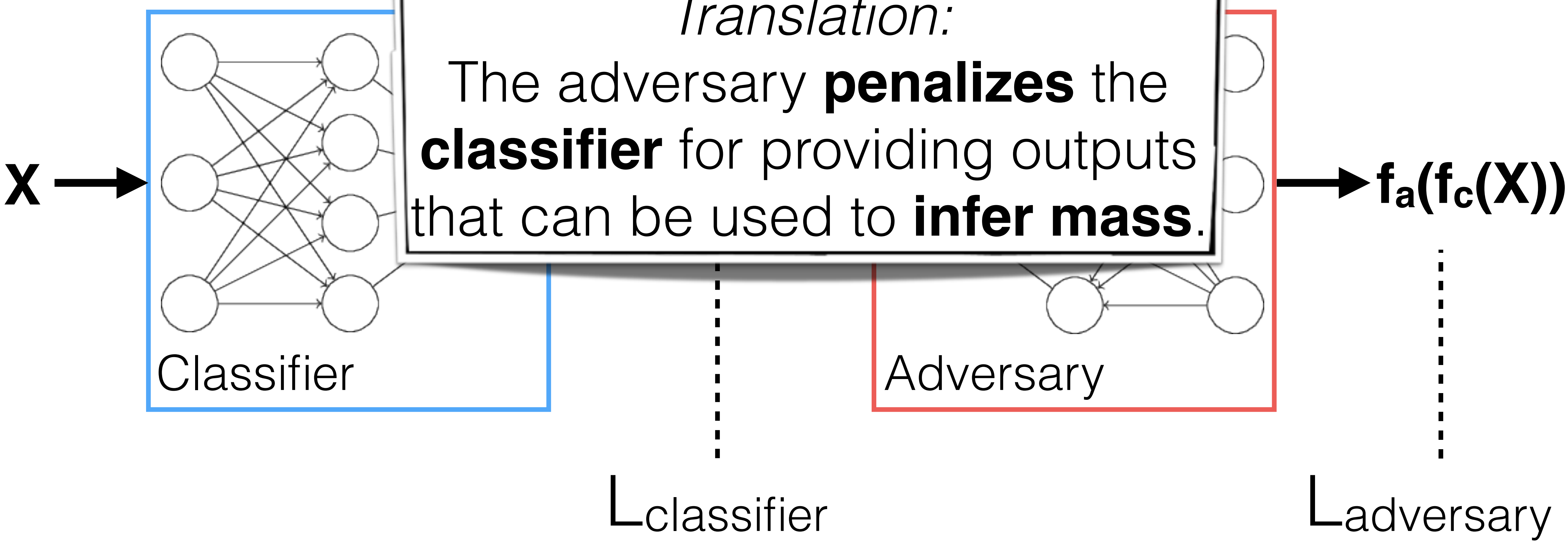
Simultaneously minimize:

$L_{adversary}$

and

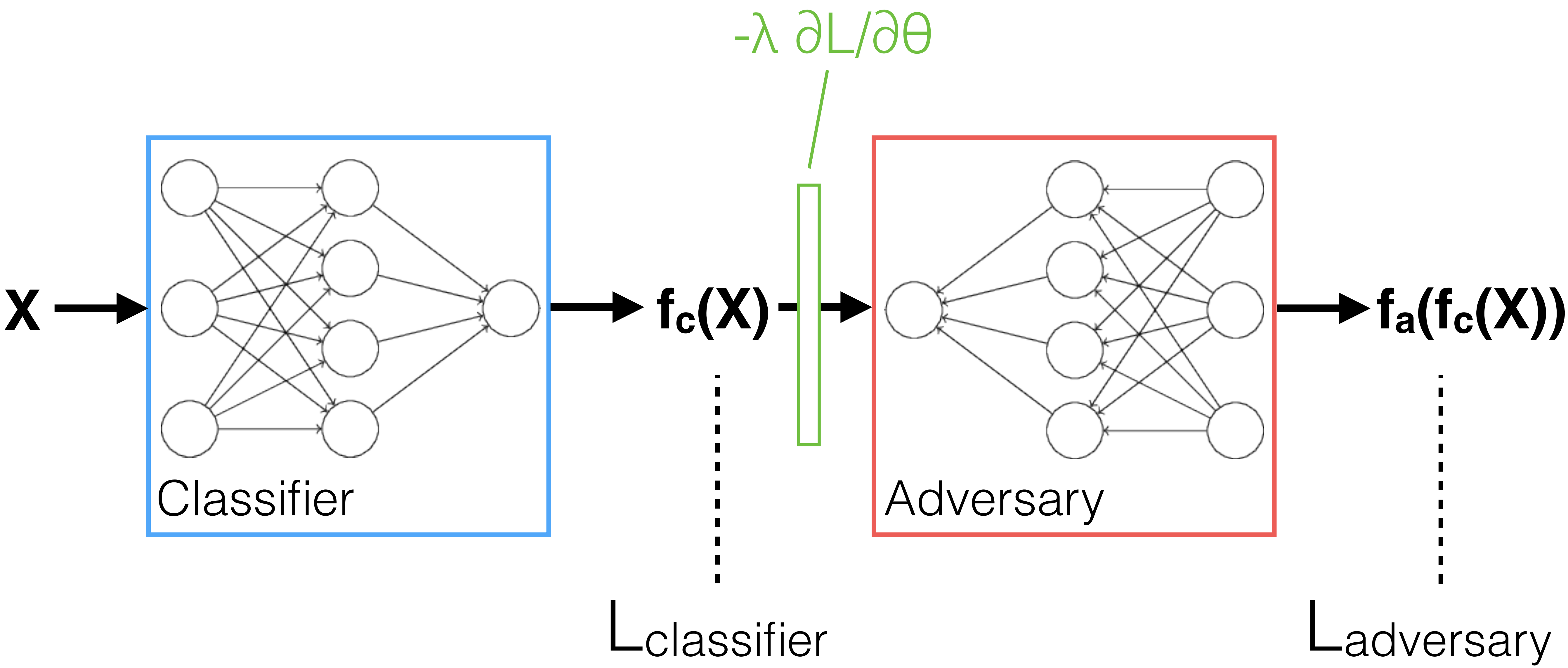
$L_{tagger} = L_{classifier} - \lambda L_{adversary}$

Translation:
The adversary **penalizes** the **classifier** for providing outputs that can be used to **infer mass**.



Training

- Simultaneous optimization achieved with **gradient scaling layer**
- Signal events are given zero weight in adversary loss



Implementation Note

- In Keras, this is implemented as a network with *two outputs* and *two loss functions*
- The whole network is trained w/ loss:
  $L_{full} = L_1 + w_2 L_2$
- So the effective value of λ for gradient-reversal scaling of g will be:
  $\lambda = g/w_2$

```
classifier = classifier_network(n_features)
adversary  = regression_network()
reversal_layer = GradientReversalLayer(lambda_reversal)

input_layer = layers.Input(shape=(n_features,))
classifier_output = classifier(input_layer)

reverse = reversal_layer(classifier_output)
adversary_output = adversary(reverse)

model = models.Model(input_layer, [classifier_output, adversary_output])
model.compile(loss=['categorical_crossentropy', 'mse'], loss_weights=[1, adversary_weight],
              optimizer=keras.optimizers.Adam())
```

Implementation Note

- In Keras, this is implemented as a network with *two outputs* and *two loss functions*
- The whole network is trained w/ loss:
  $L_{full} = L_1 + w_2 L_2$
- So the effective value of λ for gradient-reversal scaling of g will be:
  $\lambda = g/w_2$

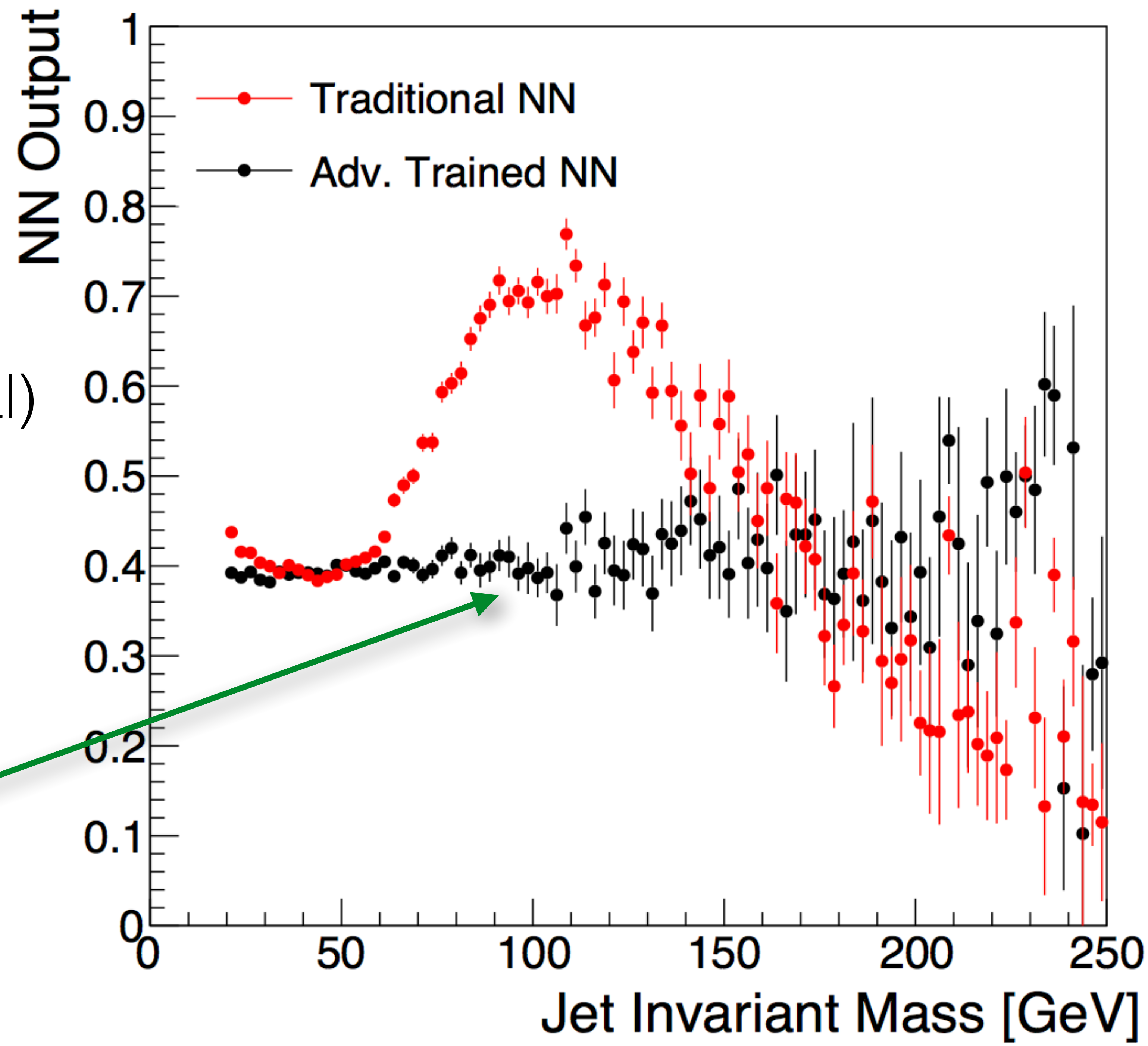
```
classifier = classifier_network(n_features)
adversary  = regression_network()
reversal_layer = GradientReversalLayer(lambda_reversal)

input_layer = layers.Input(shape=(n_features,))
classifier_output = classifier(input_layer)

reverse = reversal_layer(classifier_output)
adversary_output = adversary(reverse)

model = models.Model(input_layer, [classifier_output, adversary_output])
model.compile(loss=['categorical_crossentropy', 'mse'], loss_weights=[1, adversary_weight],
              optimizer=keras.optimizers.Adam())
```

Results

Training on ~200k MC events:

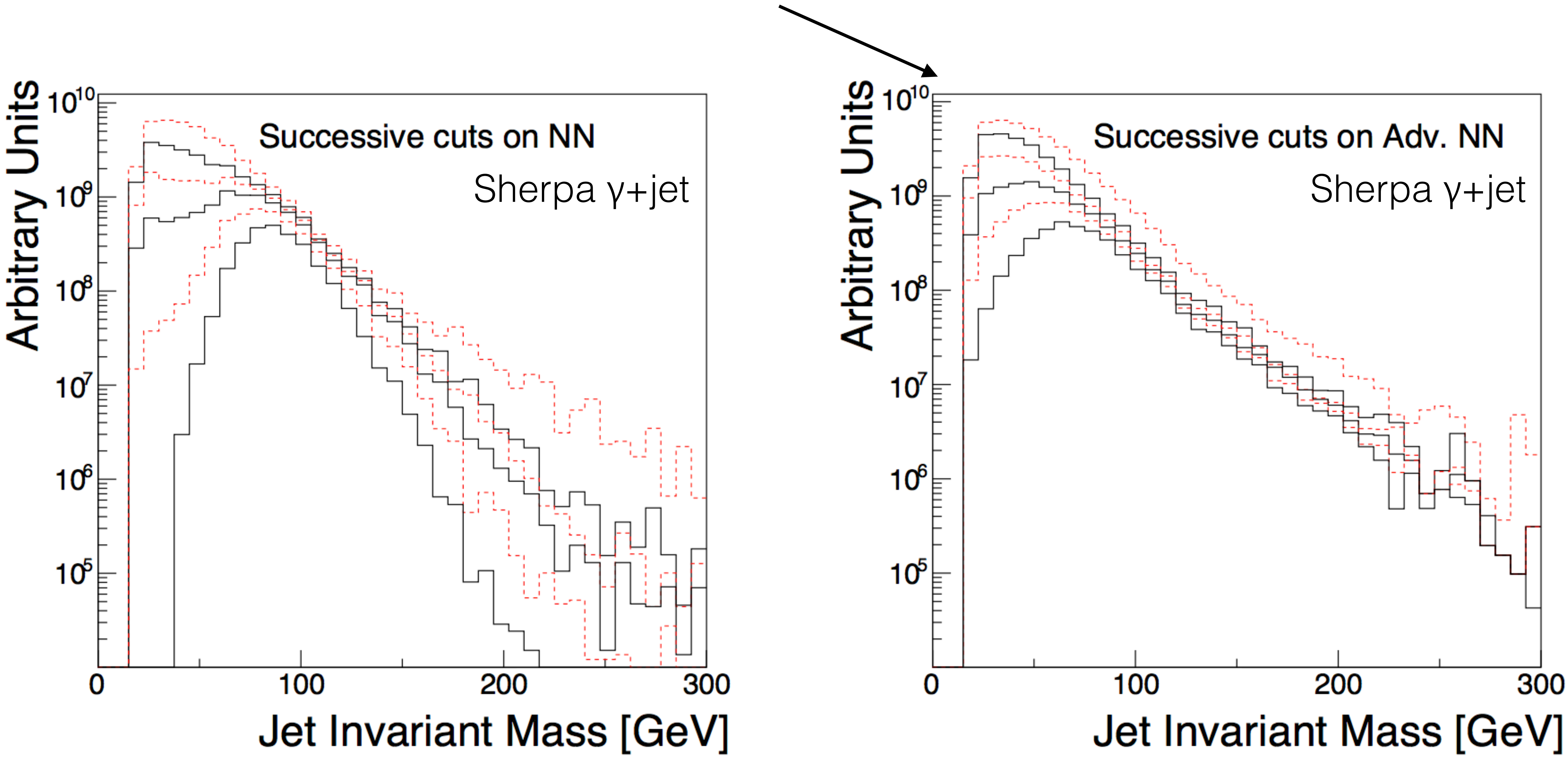
Sherpa γ+jet	(BG)
MG5 γ+Z'	(Signal)
Pythia + Delphes	(Both)

✓ Tagger profile much flatter

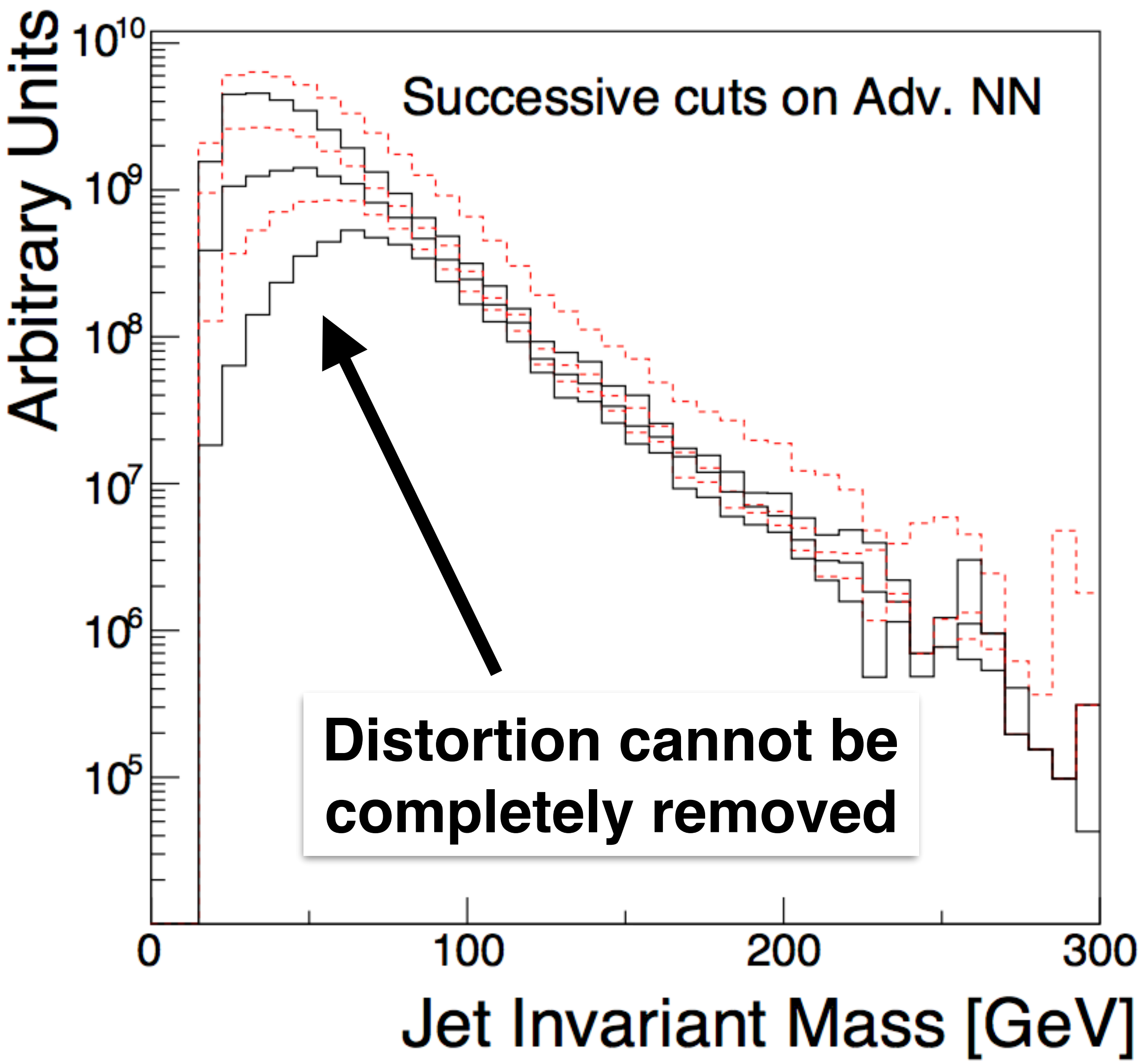


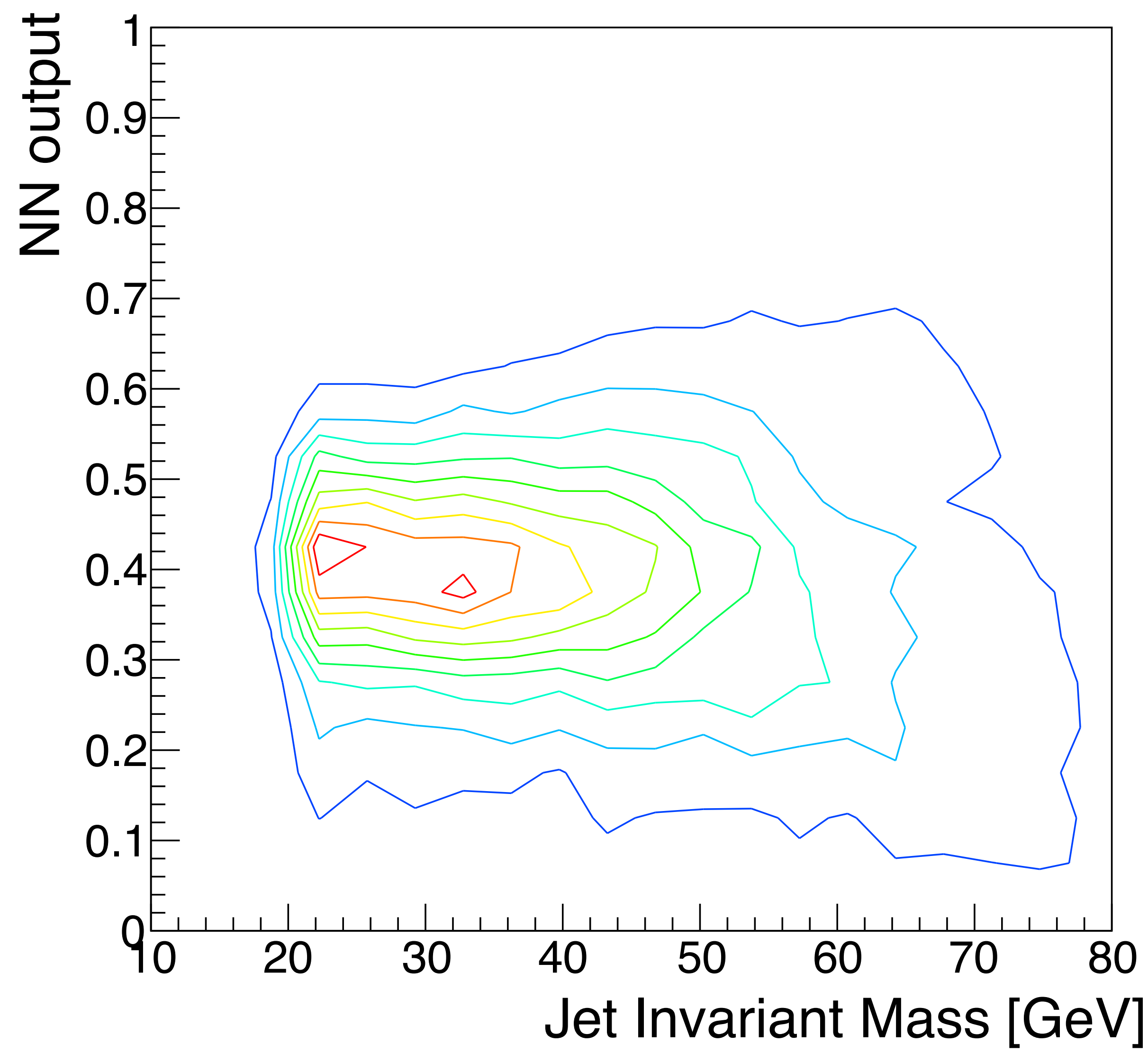
Results

✓ BG distortion considerably reduced

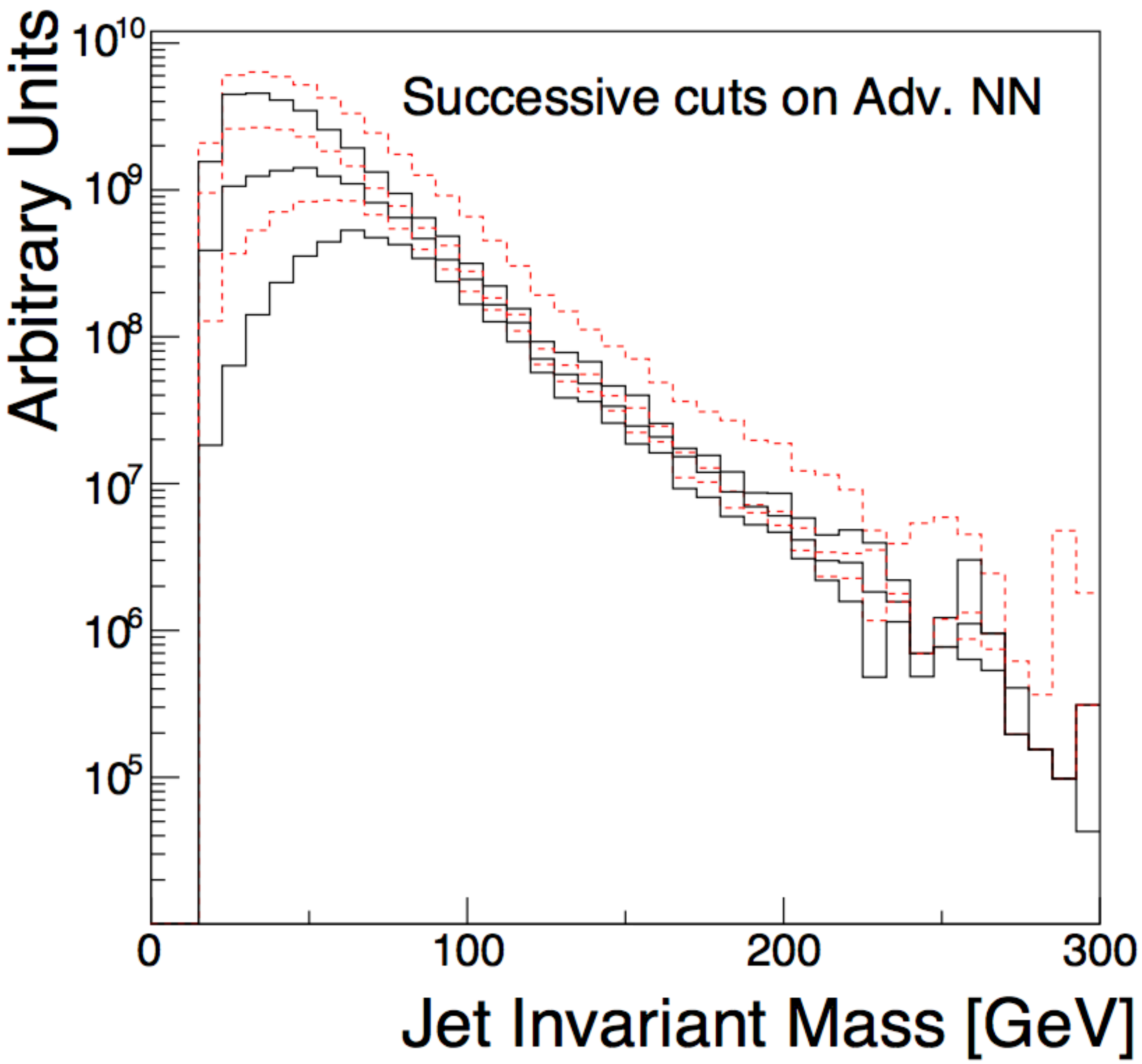


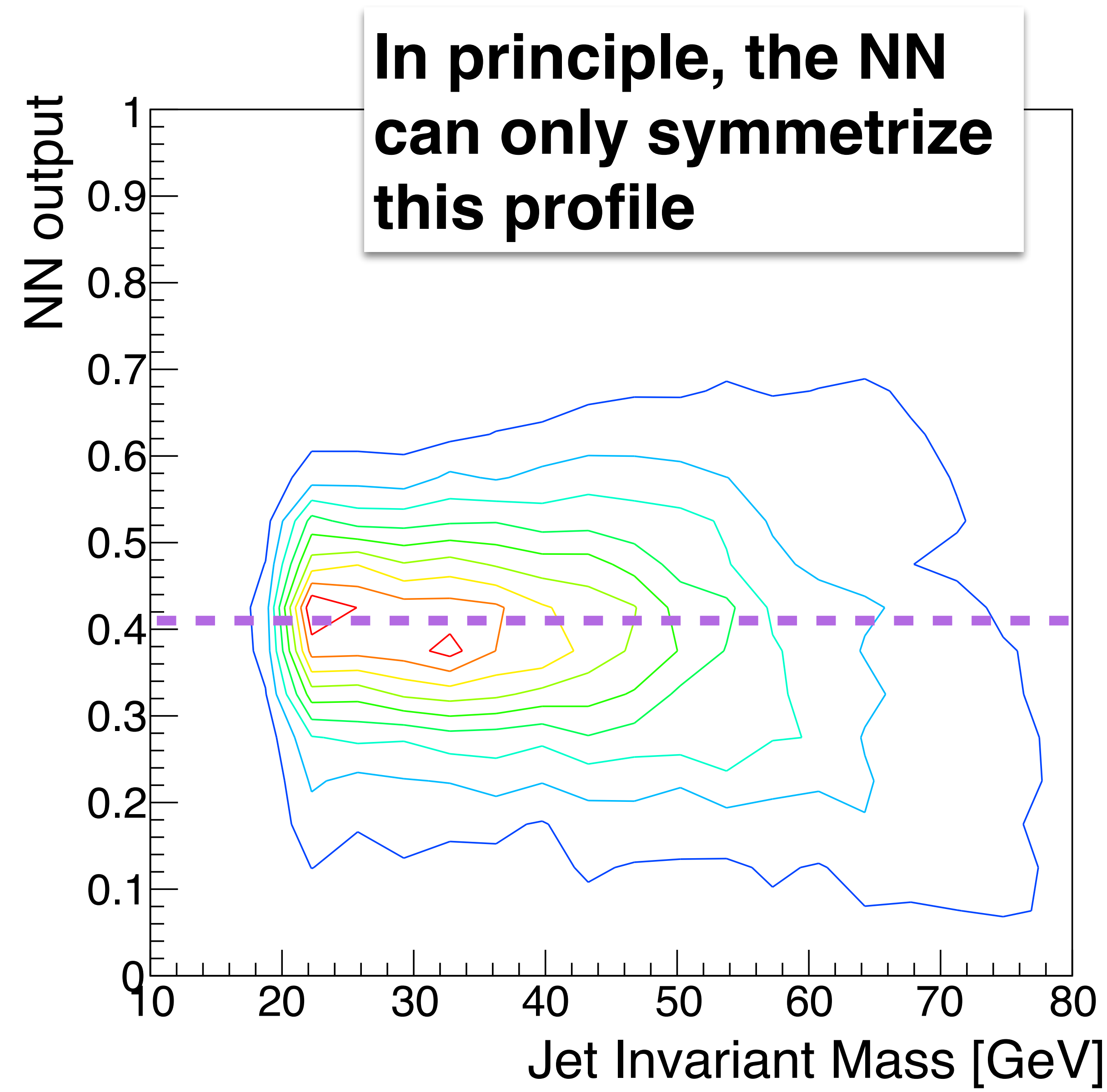
BG Distortion



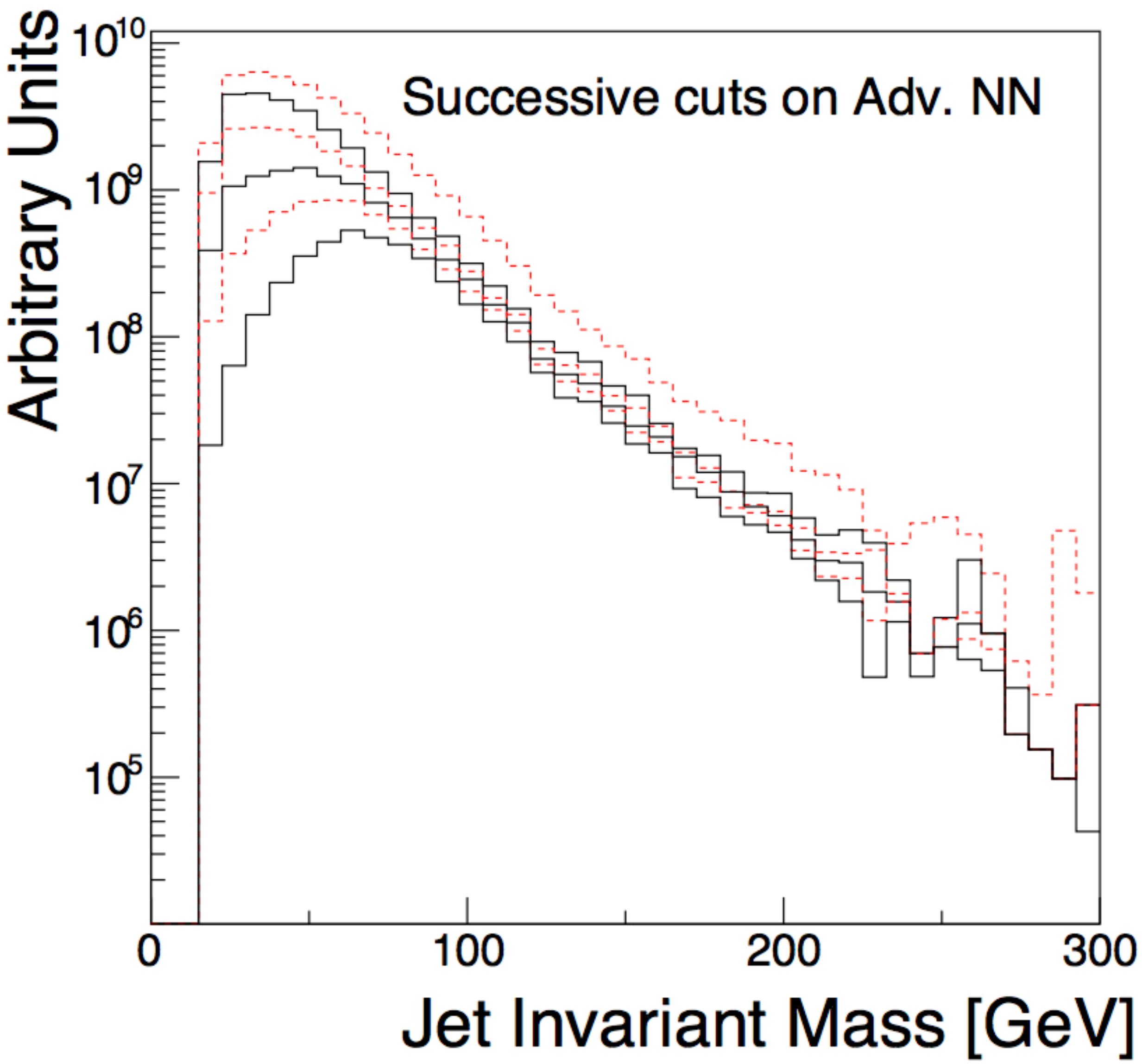


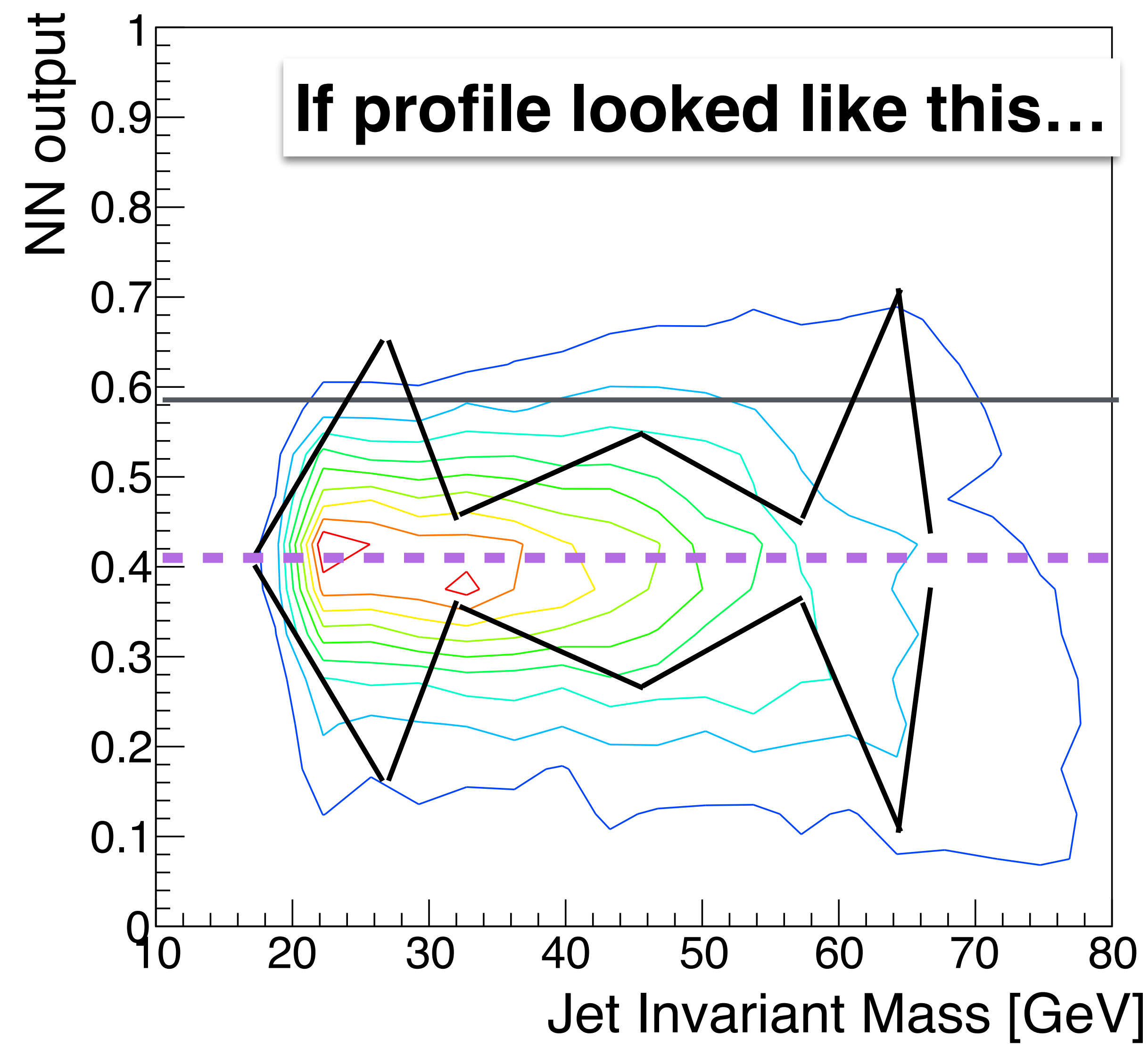
BG Distortion





BG Distortion

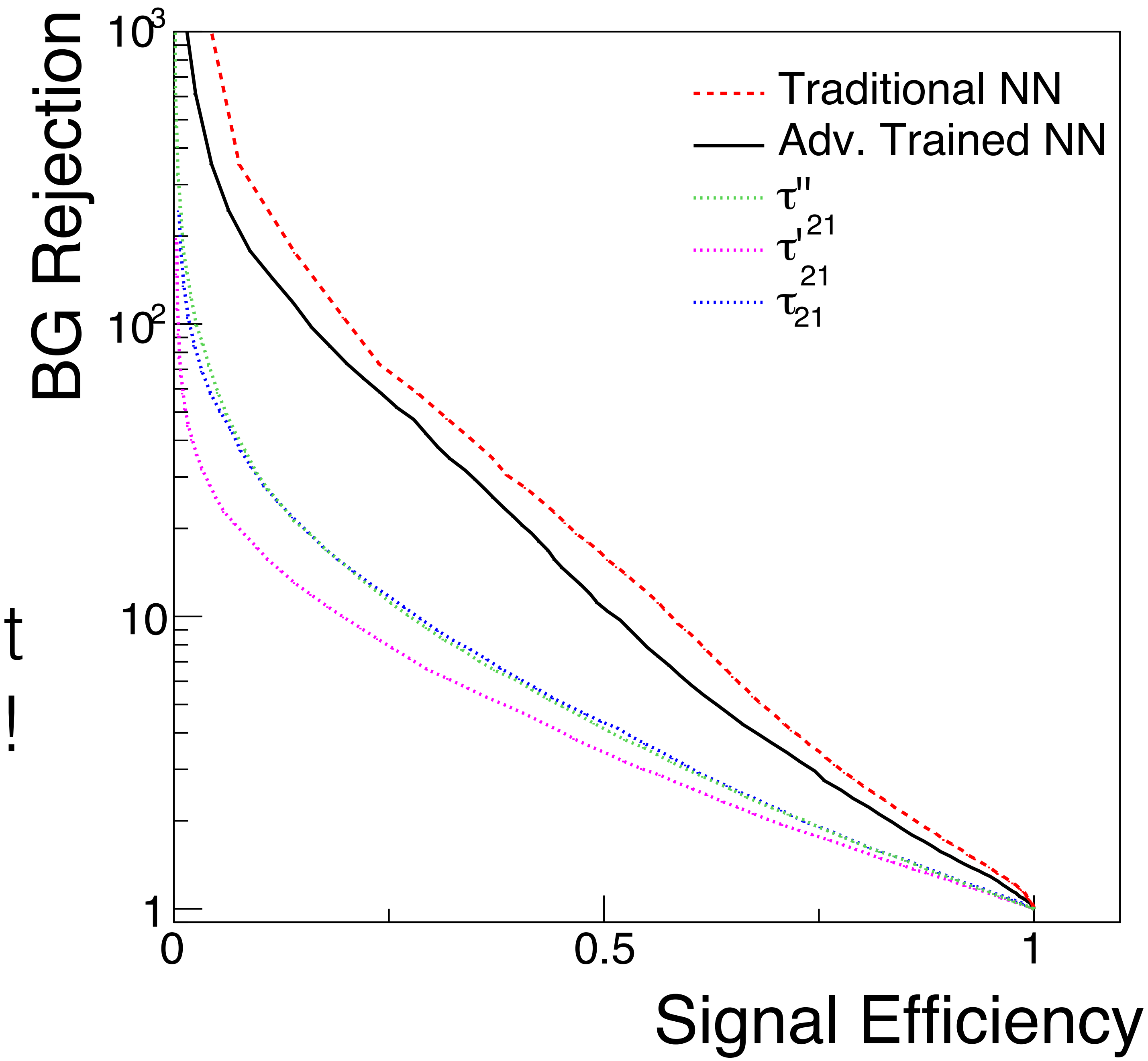




ROC Performance

Adversarial method:
slightly lower AUC

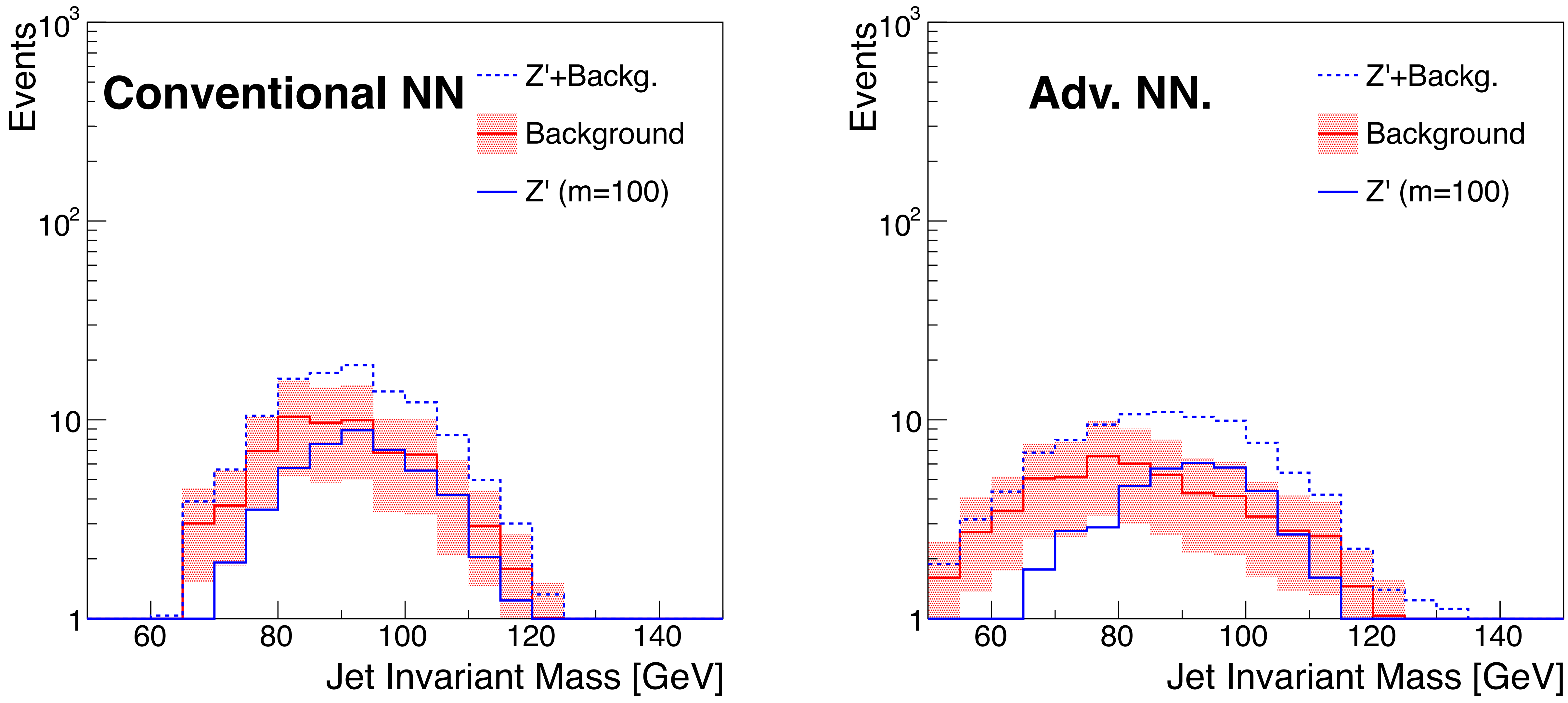
… however this is not our figure of merit!



BG Sculpting

The conventionally-trained NN is “greedy”

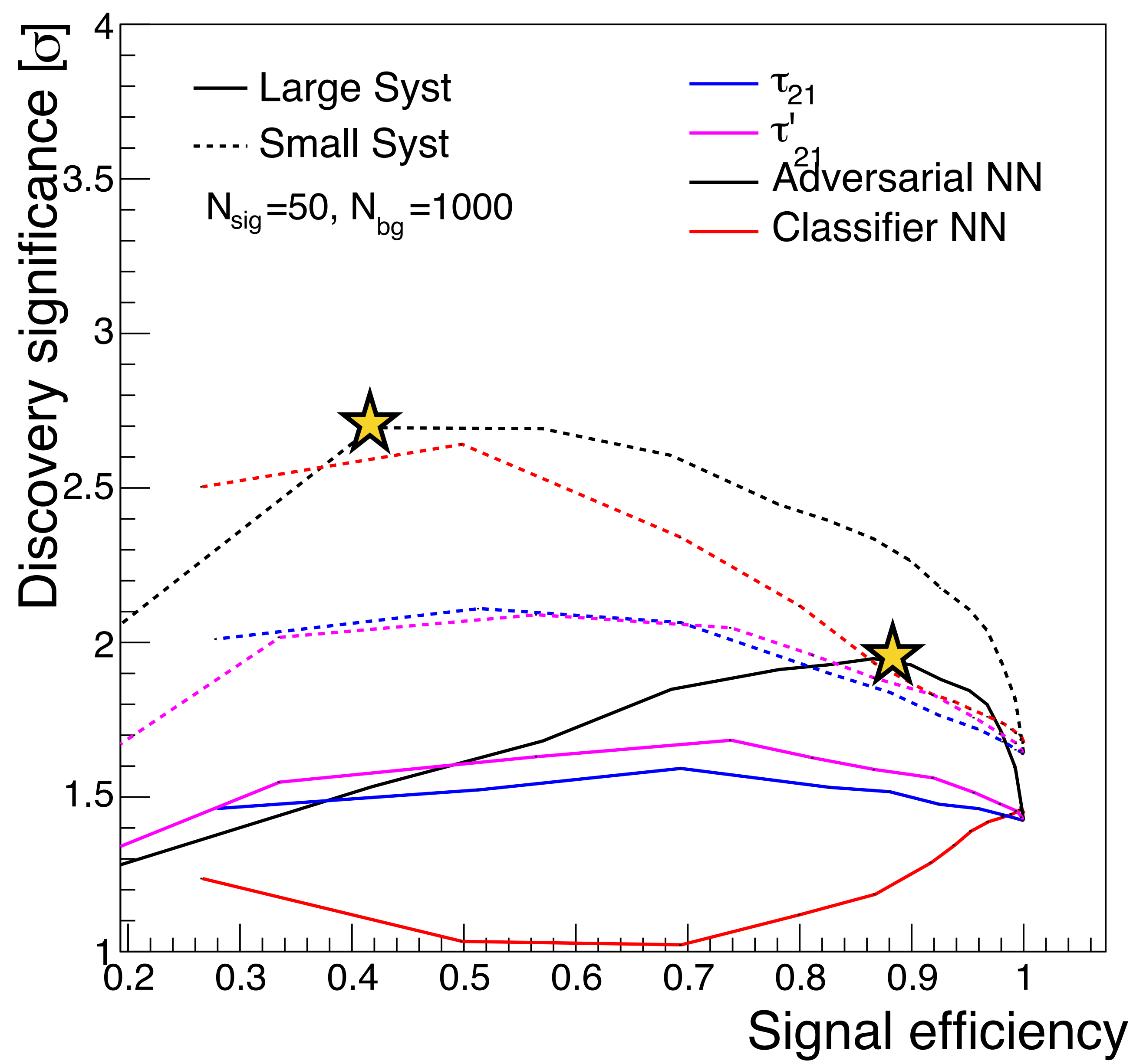
➡ Signal and BG distributions end up identical!



Statistical Significance

- Toy statistical model:
  - MC template fit
  - BG normalization uncertainty

✓ Adversarial method attains highest discovery significance

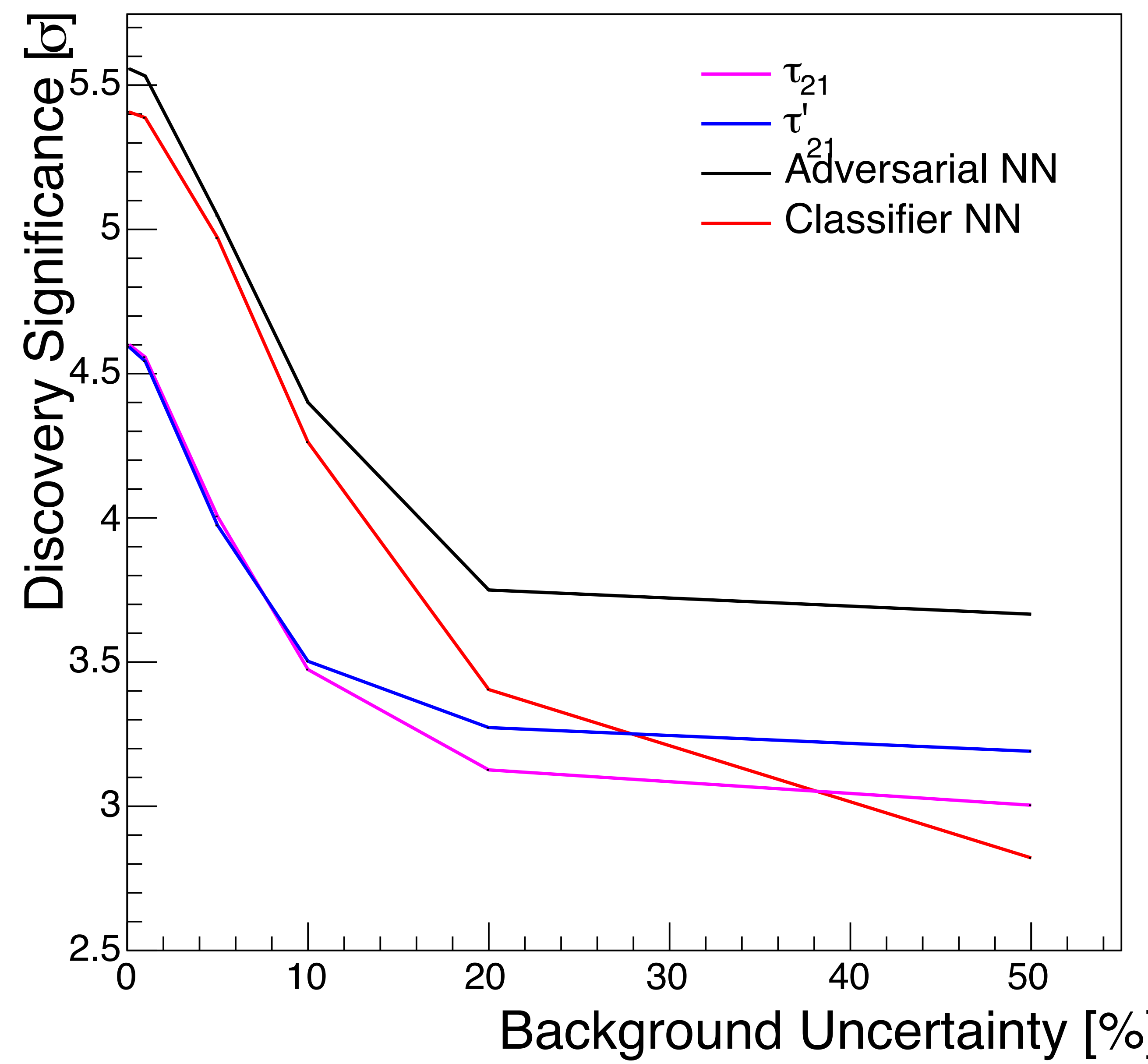


Statistical Significance

- Toy statistical model:
  - MC template fit
  - BG normalization uncertainty

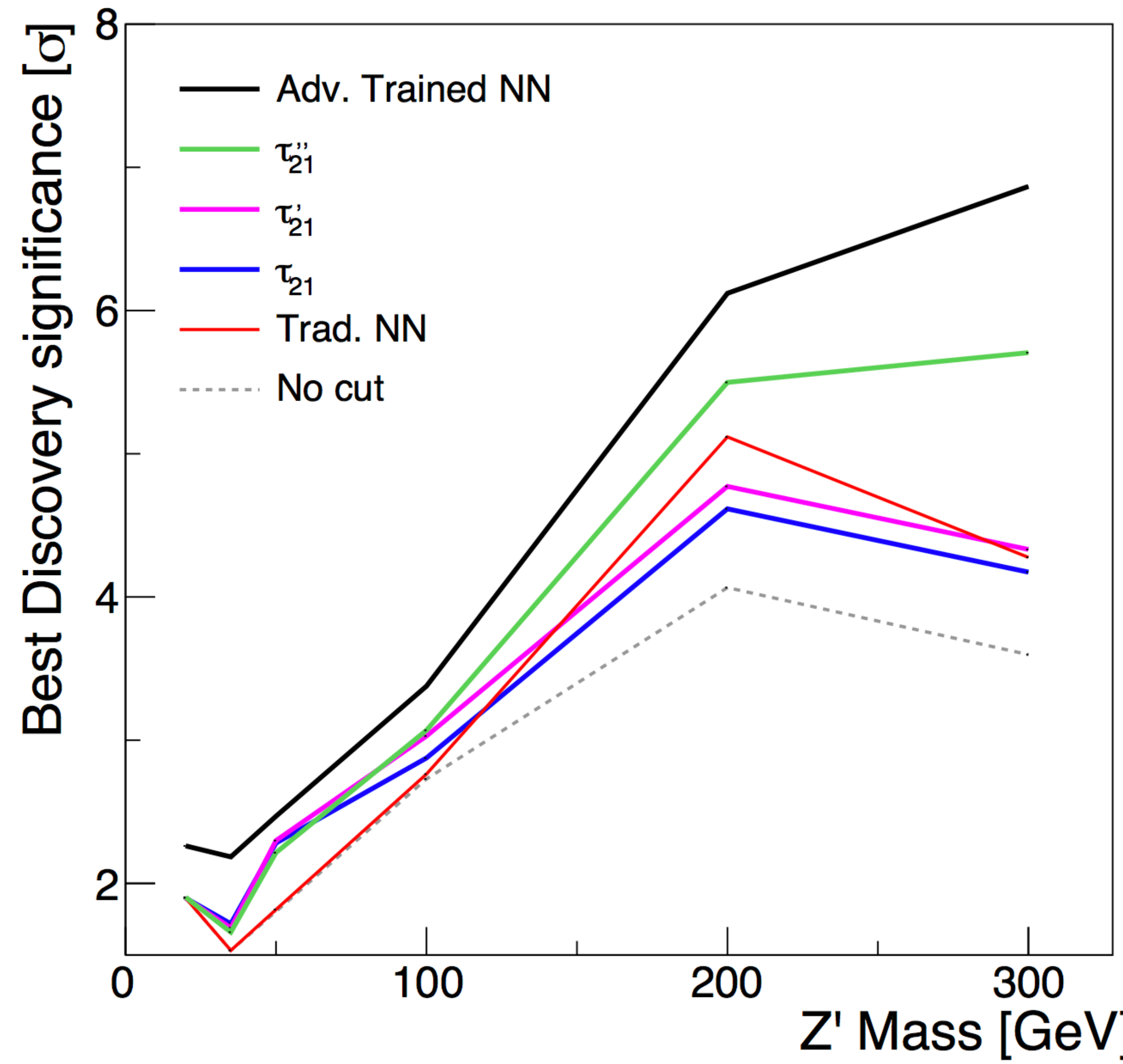
✓ Adversarial method attains highest discovery significance

- Larger systematics
  ⇒ stronger improvement



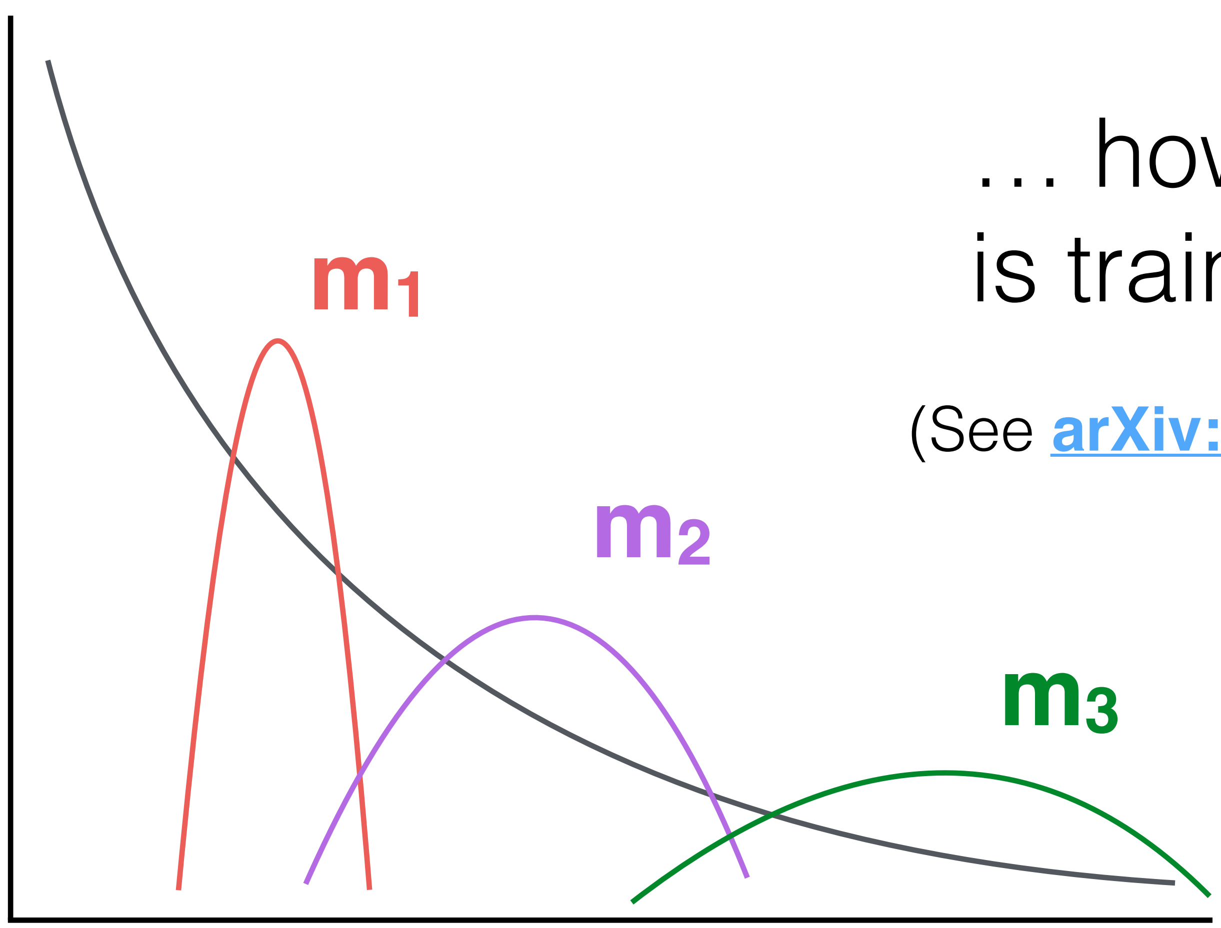
Parameter Scans

➡ Architecture can be extended to include parametric dependence on hypothesis mass, $M_{Z'}$



Parameterizing Mass

Often the case that we want to scan
a range of hypothetical mass points

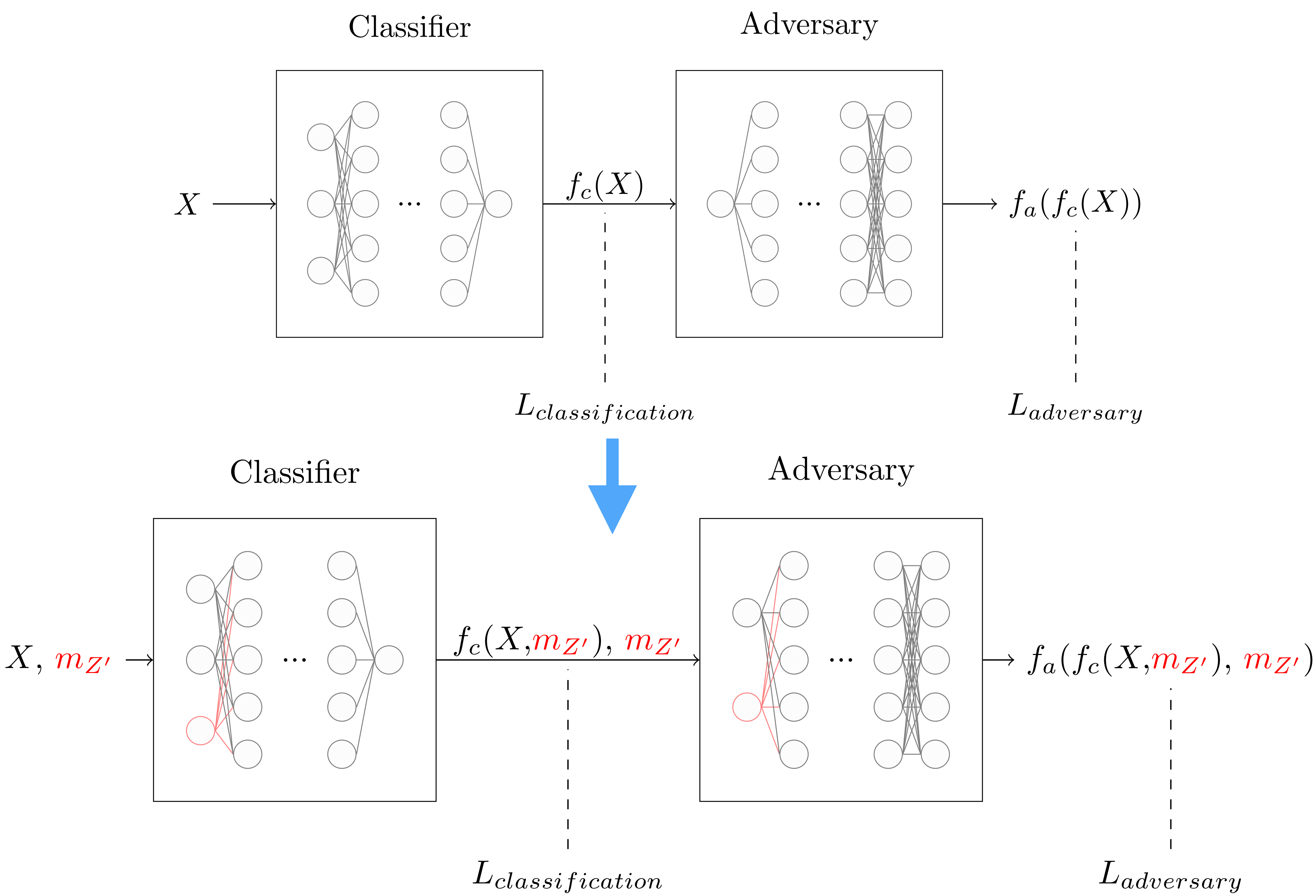


… however, the NN tagger
is trained for a specific mass

(See **arXiv:1601.07913** for treatment of this issue)

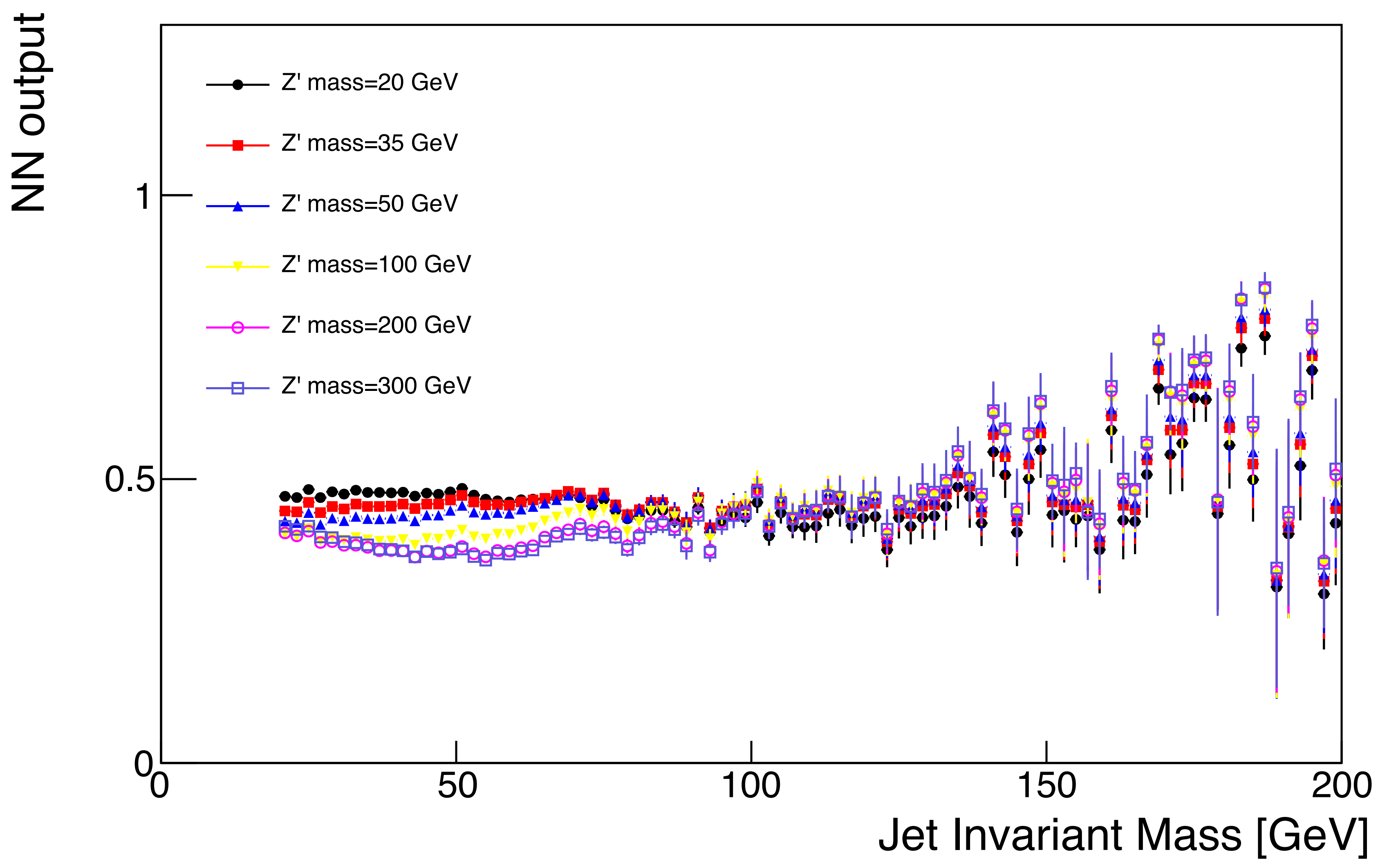
Parameterizing Mass

Simple generalization: tell (both) Neural Nets what hypothesis they are optimizing



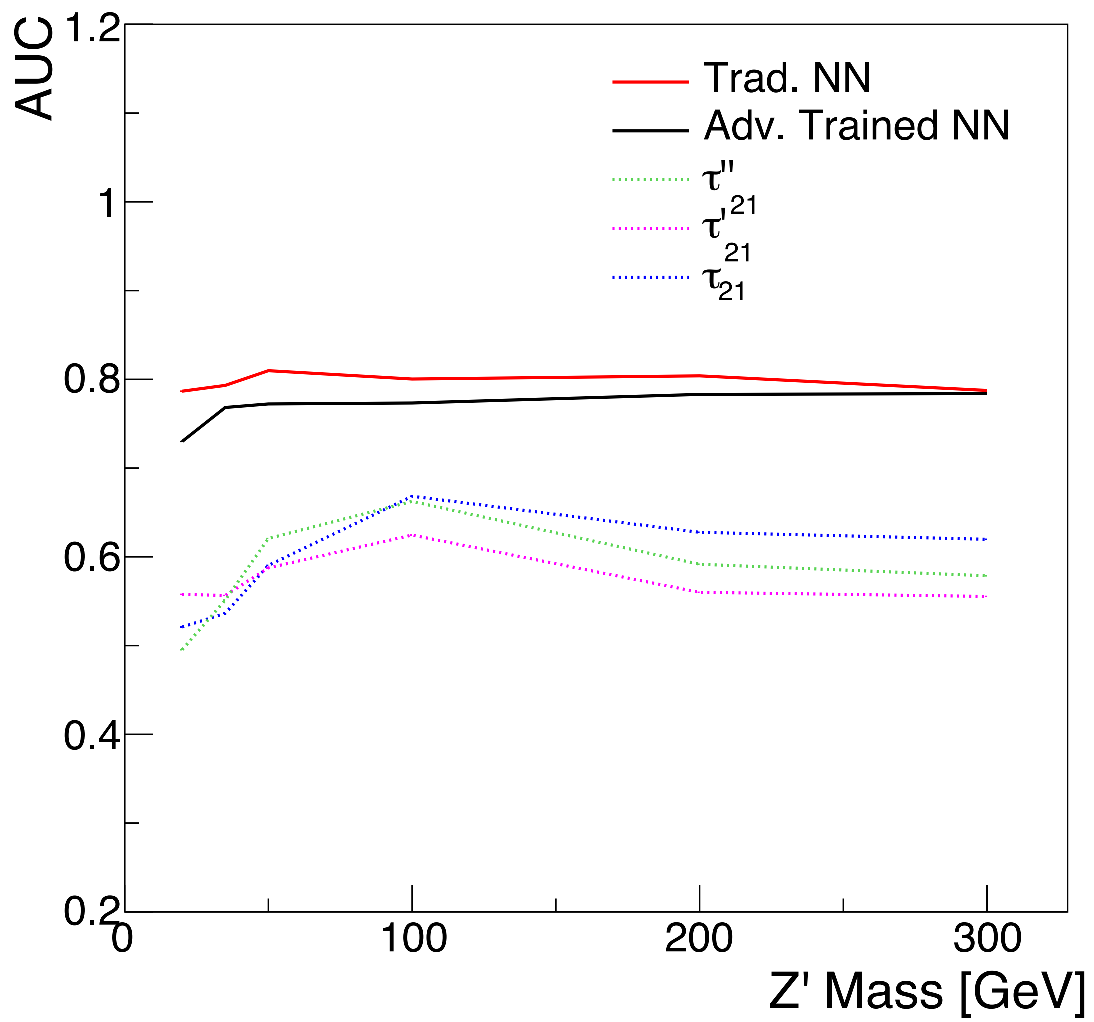
Parameterizing Mass

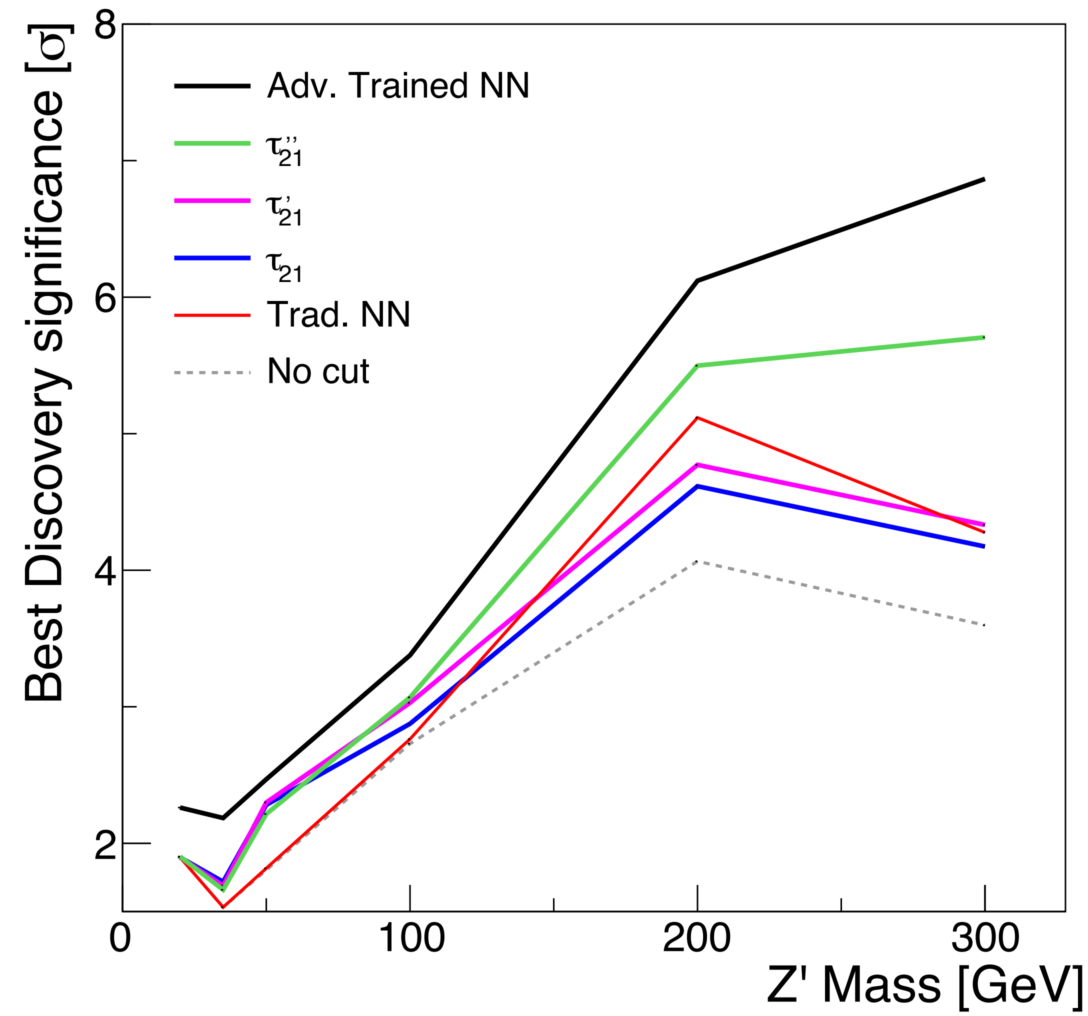
Surprisingly (to me), it works!



Parameterizing Mass

Adv. NN results are as before, for all mass points



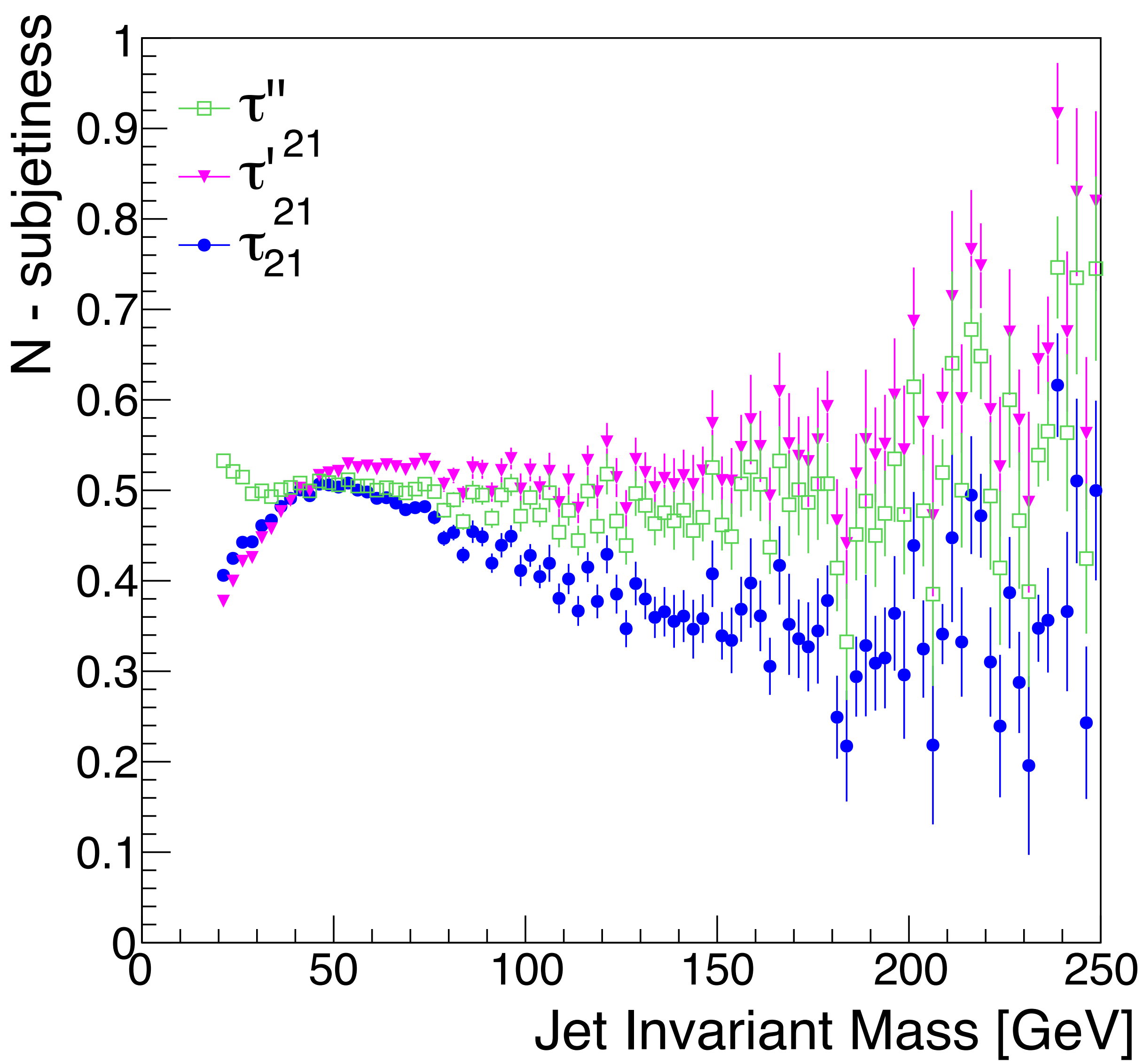


Summary / Conclusion

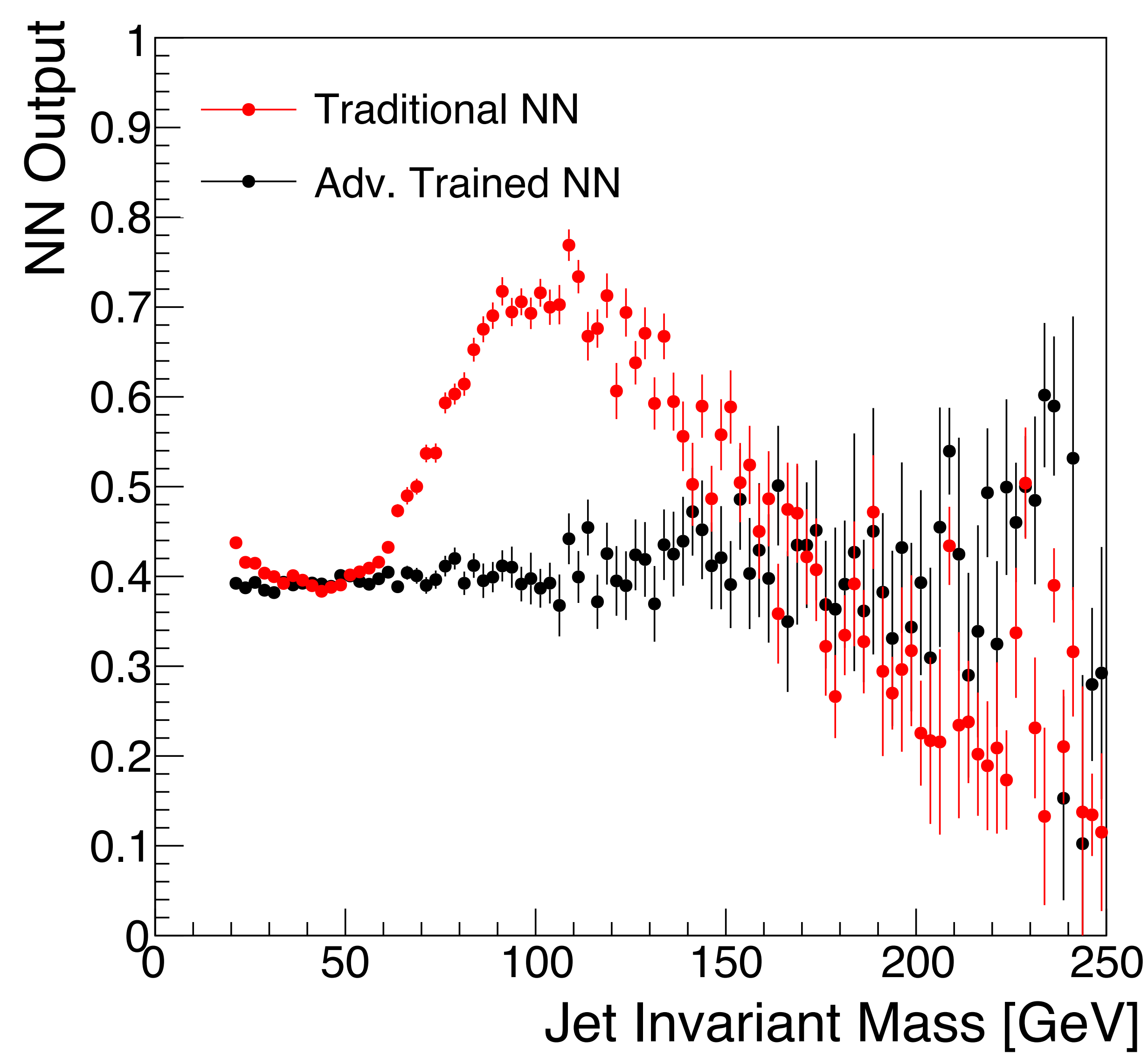
- Multivariate taggers are powerful tools for many signals
- However, correlation with analysis observables results in **reduced sensitivity** in the presence of **BG modeling systematics**
- Adversarial techniques can enforce decorrelation for **arbitrarily complex classifiers**
- Resulting classifiers may outperform both theoretically-motivated variables as well as conventional multivariate methods
- Method is **generic** and should work for different object taggers and/or analysis observables

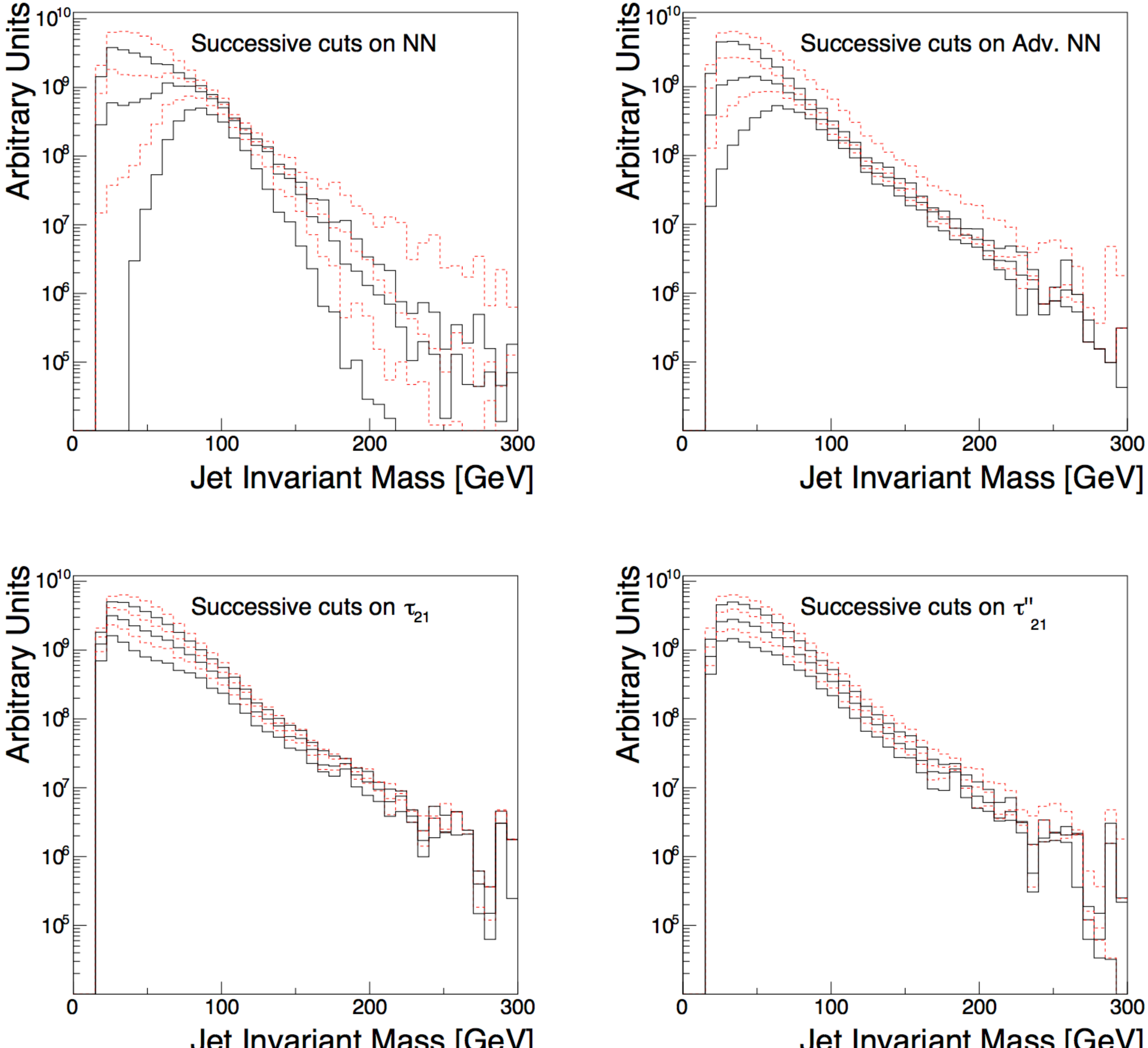
End

N-subjettiness profiles



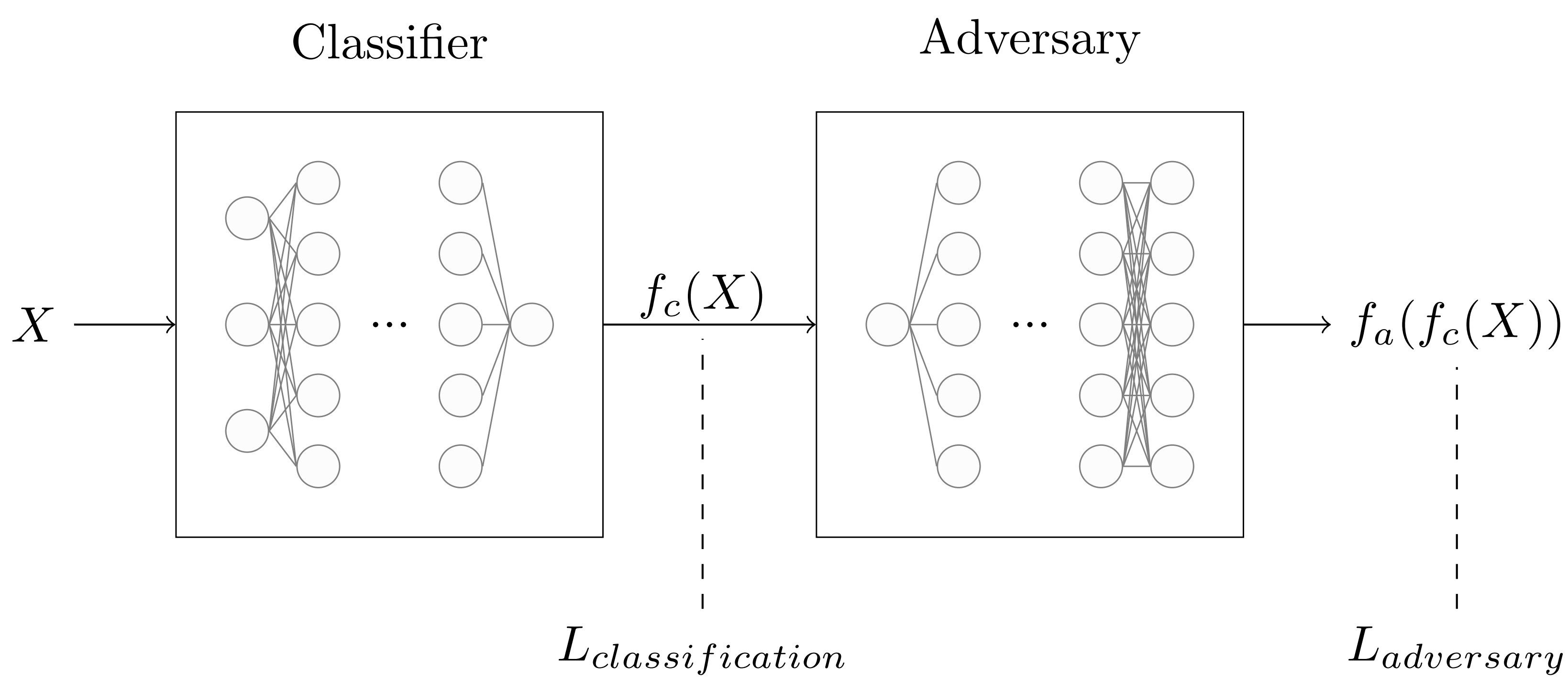
NN profiles



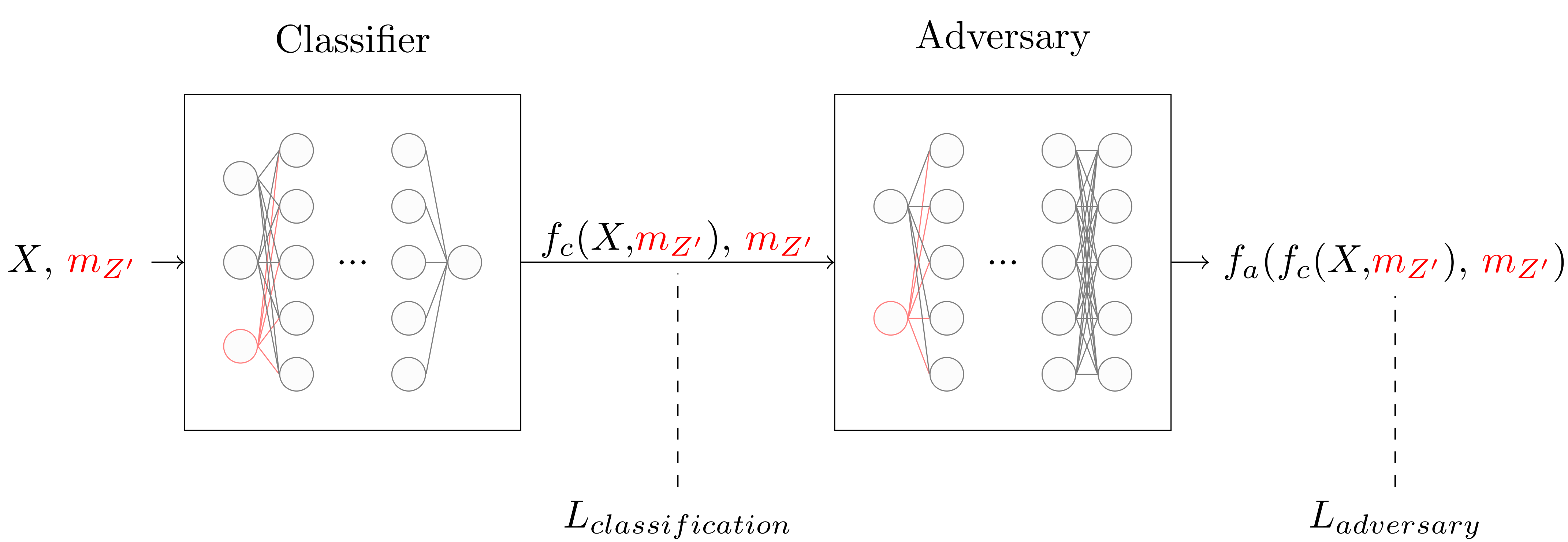


Successive cuts on NN
Arbitrary Units
Jet Invariant Mass [GeV]
Successive cuts on Adv. NN
Arbitrary Units
Jet Invariant Mass [GeV]
Successive cuts on τ_{21}
Arbitrary Units
Jet Invariant Mass [GeV]
Successive cuts on τ''_{21}
Arbitrary Units
Jet Invariant Mass [GeV]

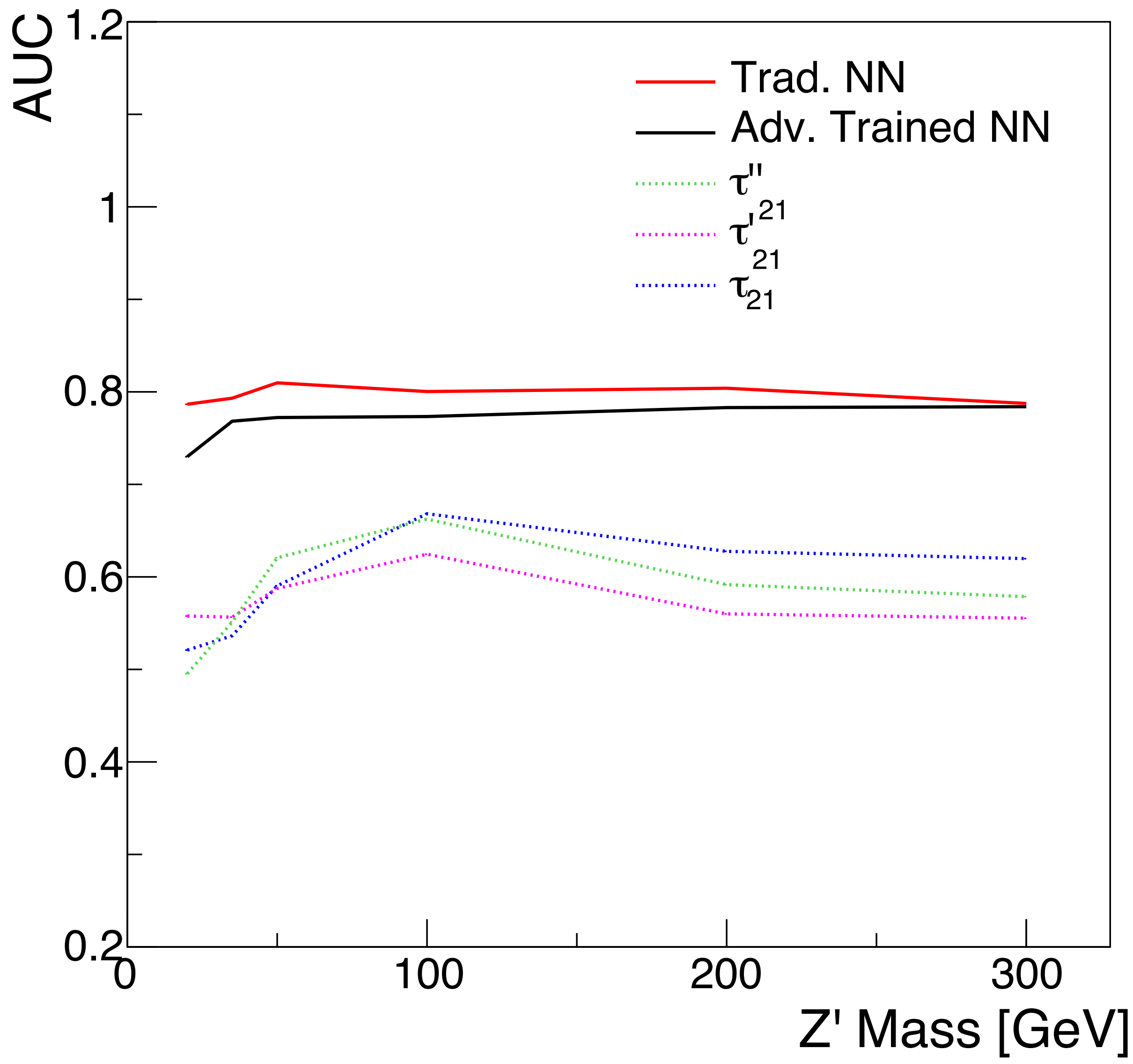
Adv. NN

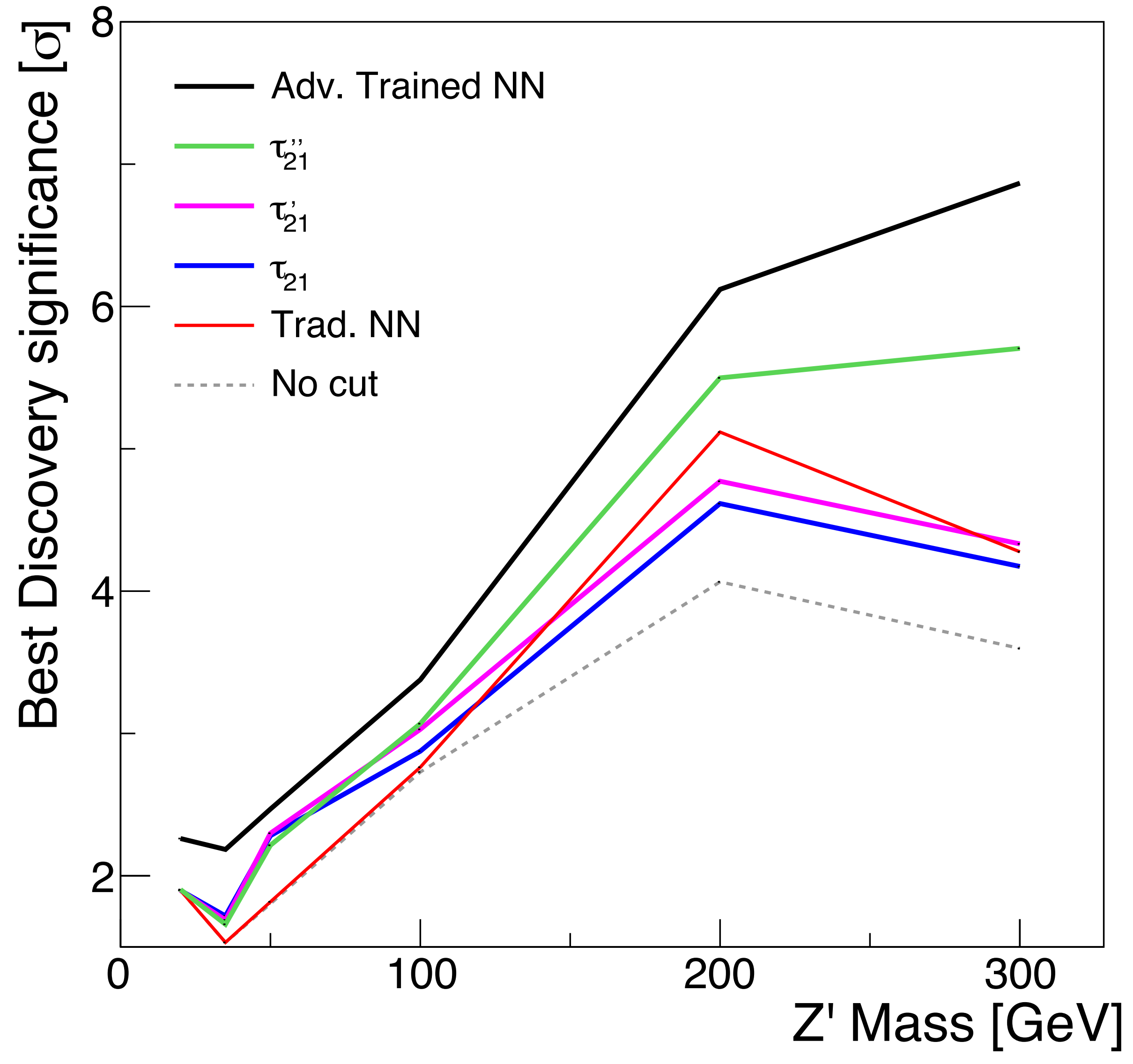


Parametric Adv. NN



AUC and significance





pT dependence