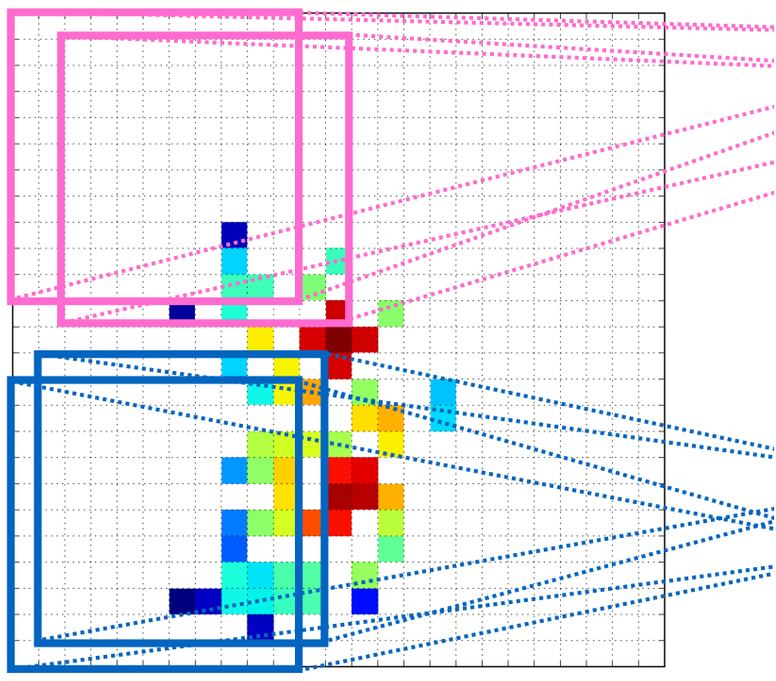
Jack Collins, Kiel Howe, **BPN**, Phys. Rev. Lett. 121, 241803 (2018)

A model agnostic machine learning search for resonant new physics

Benjamin Nachman

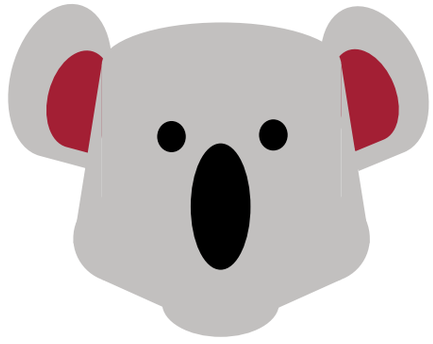
Lawrence Berkeley National Laboratory



UCI Seminar
January 15, 2019

Outline

- Why model agnostic searches?



Intermezzo: Learning directly from data

- Why it is important to learn directly from data

- How to learn without labels

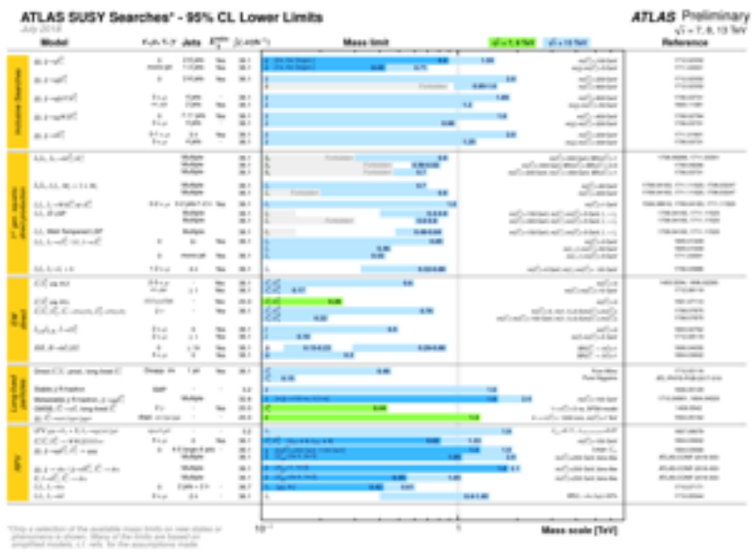
- BSM Searches background / signal model independent

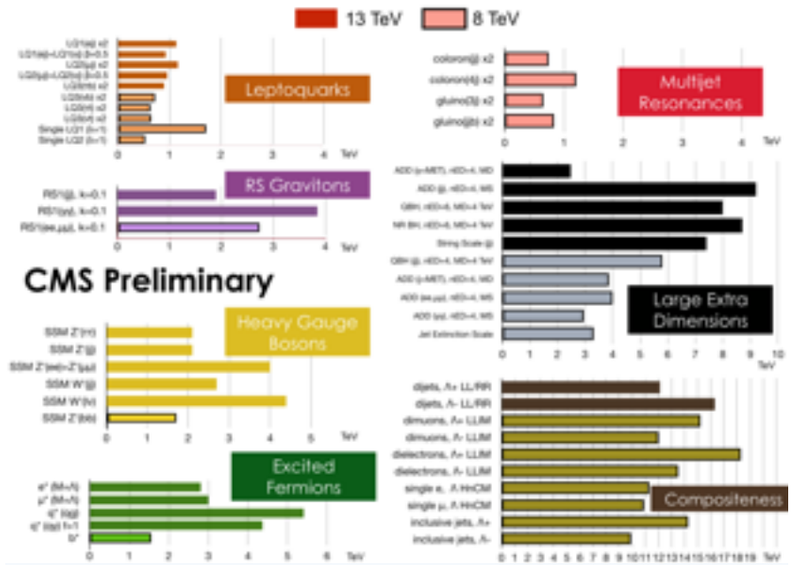
Most searches at the LHC (with or without ML)	Data versus SM simulation (“general search”)
CWoLa Hunting (focus of this talk)	Autoencoders

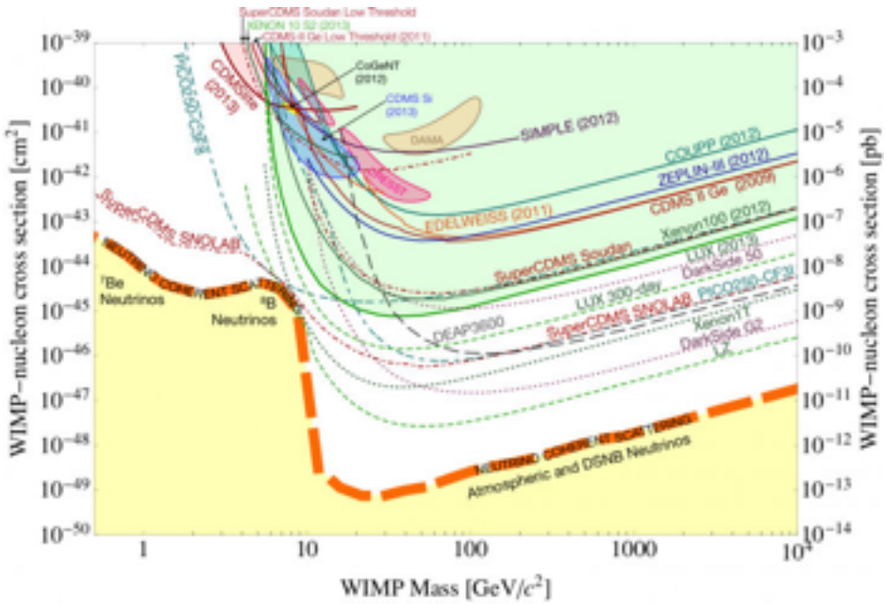
- The future

Why model agnostic?

Fact: no conclusive evidence for BSM in HEP*







Three possibilities

There is no BSM (SM all the way to Planck)

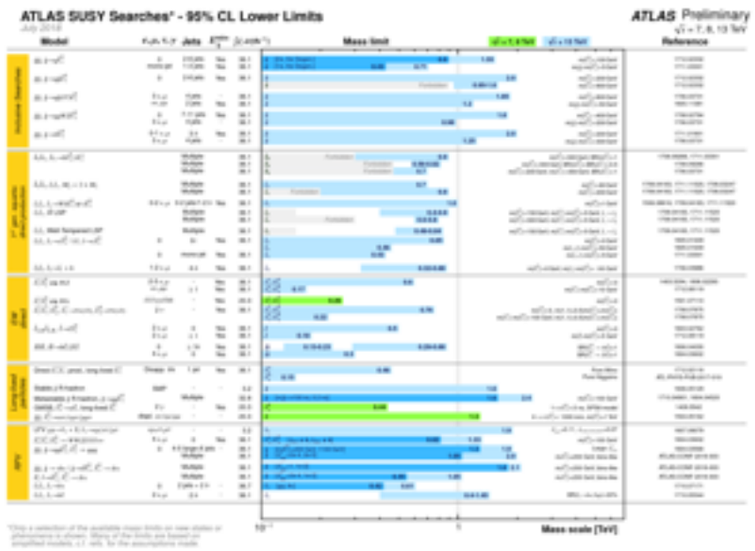
Patience! (BSM is just rare)

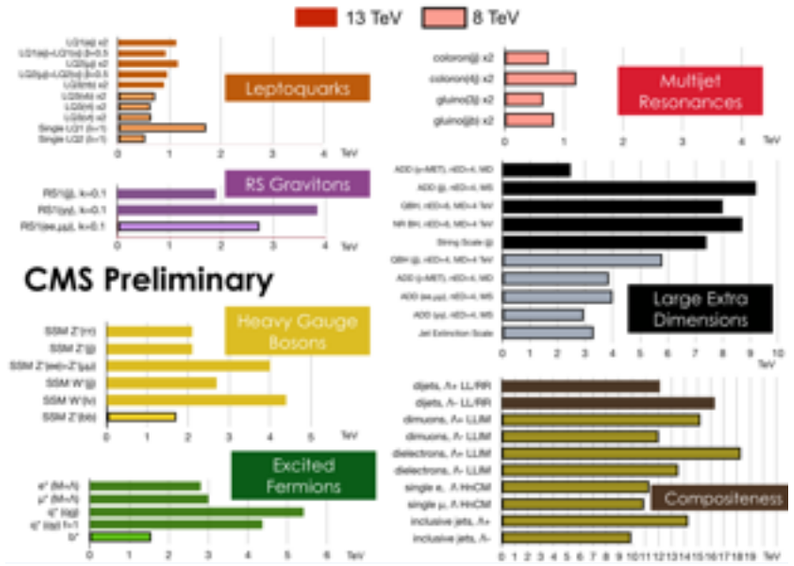
We are just not looking in the right place

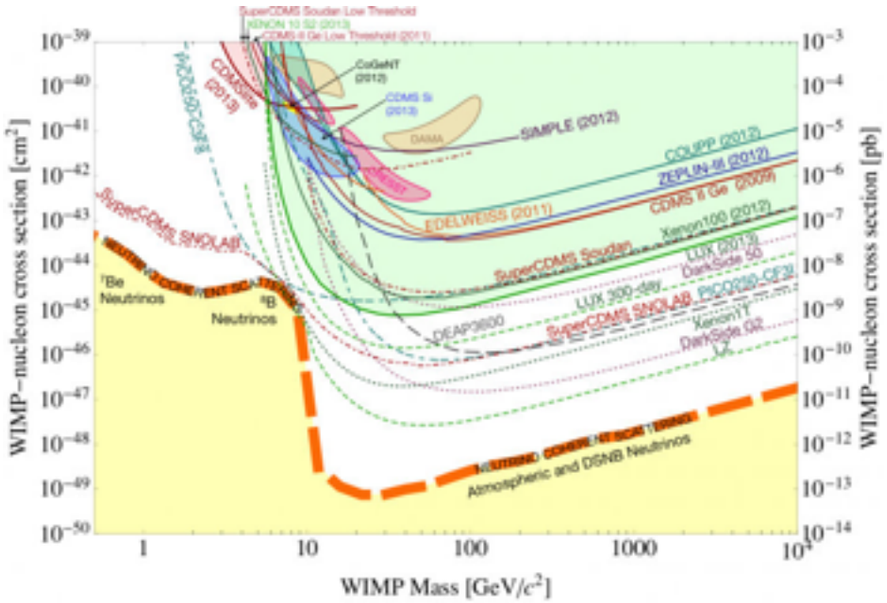
*neutrinos are *technically* BSM, but depending on their origin of their mass, may not lead to "new" short-distance physics. We do know there is dark matter, but we don't even know if it is composed of particles (such as WIMPs). There are also multiple B physics anomalies, but they are not conclusive.

Why model agnostic?

Fact: no conclusive evidence for BSM in HEP*







Three possibilities

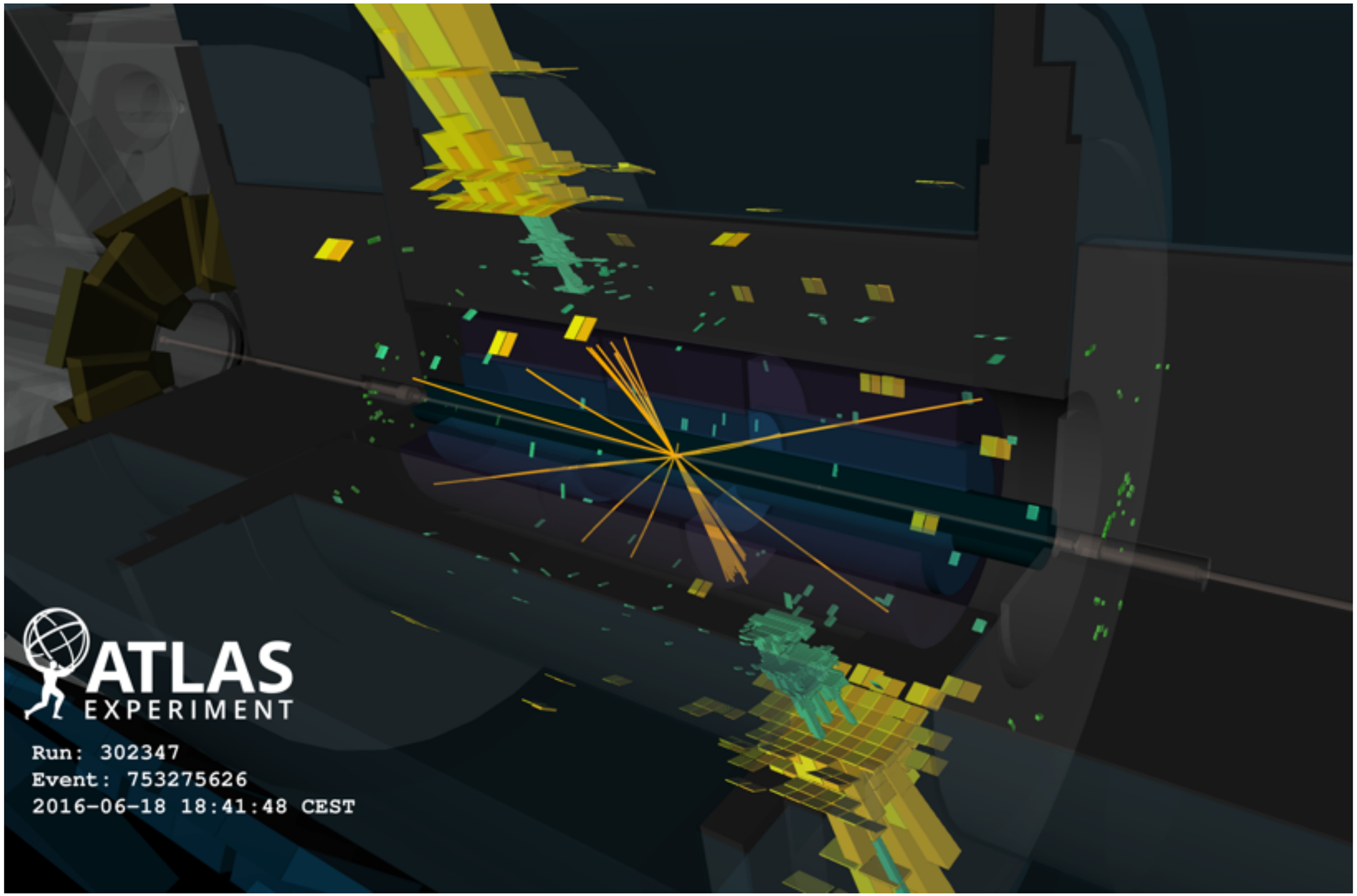
There is no BSM (SM all the way to Planck)

Patience! (BSM is just rare)

We are just not looking in the right place

*neutrinos are *technically* BSM, but depending on their origin of their mass, may not lead to "new" short-distance physics. We do know there is dark matter, but we don't even know if it is composed of particles (such as WIMPs). There are also multiple B physics anomalies, but they are not conclusive.

Consider the collider di-object resonance search.



Signatures, not models

Consider the collider di-object resonance search.

	e	μ	τ	γ	j	b	t	W	Z	h
e	$\pm\mp$[4],$\pm\pm$[5]	$\pm\pm$[5, 6]	$\pm\mp$[6, 7]	[7]	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
μ		$\pm\mp$[4],$\pm\pm$[5]		[7]	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
τ				[8]	$\varnothing$	$\varnothing$	[9]	$\varnothing$	$\varnothing$	$\varnothing$
γ					[10]	[11–13]	$\varnothing$	$\varnothing$	[14]	[14]	$\varnothing$
j						[15]	[16]	[17]	[18]	[18]	$\varnothing$
b							[16]	[19]	$\varnothing$	$\varnothing$	$\varnothing$
t								[20]	[21]	$\varnothing$	$\varnothing$
W									[22–25]	[23, 24, 26, 27]	[28–30]
Z										[23, 25, 31]	[28, 30, 32, 33]
h											[34–37]

Signatures

p p → X → SM SM

1610.09392 (Yvonne Ng, D. Whiteson, et al.)

	e	μ	τ	γ	j	b	t	W	Z	h
e	$Z', H^{\pm\pm}$	$\not{R}, H^{\pm\pm}$	$\not{R}, H^{\pm\pm}$	L^*	$LQ, \not{R}$	$LQ, \not{R}$	$LQ, \not{R}$	L^*, ν_{KK}	L^*, e_{KK}	L^*
μ		$Z', H^{\pm\pm}$	$\not{R}, H^{\pm\pm}$	L^*	$LQ, \not{R}$	$LQ, \not{R}$	$LQ, \not{R}$	L^*, ν_{KK}	L^*, μ_{KK}	L^*
τ			$Z', H, H^{\pm\pm}$	L^*	$LQ, \not{R}$	$LQ, \not{R}$	$LQ, \not{R}$	L^*, ν_{KK}	L^*, τ_{KK}	L^*
γ				$H, G_{KK}, \mathcal{Q}$	Q^*	Q^*	Q^*	$W_{KK}, \mathcal{Q}$	$H, \mathcal{Q}$	Z_{KK}
j					Z', ρ, G_{KK}	$W', \not{R}$	$T', \not{R}$	Q^*, Q_{KK}	Q^*, Q_{KK}	Q'
b						Z', H	$W', \not{R}, H^{\pm}$	T', Q^*, Q_{KK}	Q^*, Q_{KK}	B'
t							H, G', Z'	T'	T'	T'
W								H, G_{KK}, ρ	$W', \mathcal{Q}$	$H^{\pm}, \mathcal{Q}, \rho$
Z									H, G_{KK}, ρ	A, ρ
h										H, G_{KK}

Models

There are many uncovered combinations !

(since the writing of this paper, some have been filled in)

Signatures, not models

Consider the collider di-object resonance search.

	e	μ	τ	γ	j	b	t	W	Z	h	*BSM*
e	$\pm\mp$[4],$\pm\pm$[5]	$\pm\pm$[5, 6]	$\pm\mp$[6, 7]	[7]	∅	∅	∅	∅	∅	∅	
μ		$\pm\mp$[4],$\pm\pm$[5]		[7]	∅	∅	∅	∅	∅	∅	
τ				[8]	∅	∅	∅	[9]	∅	∅	∅	
γ					[10]	[11–13]	∅	∅	[14]	[14]	∅	
j						[15]	[16]	[17]	[18]	[18]	∅	
b							[16]	[19]	∅	∅	∅	
t								[20]	[21]	∅	∅	
W									[22–25]	[23, 24, 26, 27]	[28–30]	
Z										[23, 25, 31]	[28, 30, 32, 33]	
h											[34–37]	
BSM												

Signatures

For a generic p p → X → B C, mostly uncovered!

Some of the BSM elements are partially covered with e.g. XH and the photon jets search.

Training Directly on (unlabeled) Data

I hope I've convinced you that it is important to have an open mind when looking at data.

Now, let's take a step back and discuss why it is important to train on data **in general**.

Intermezzo: Learning directly from data

- Why it is important to learn directly from data
- How to learn without labels

The importance of training on data

Let's start with a small thought experiment.

Are these equivalent?

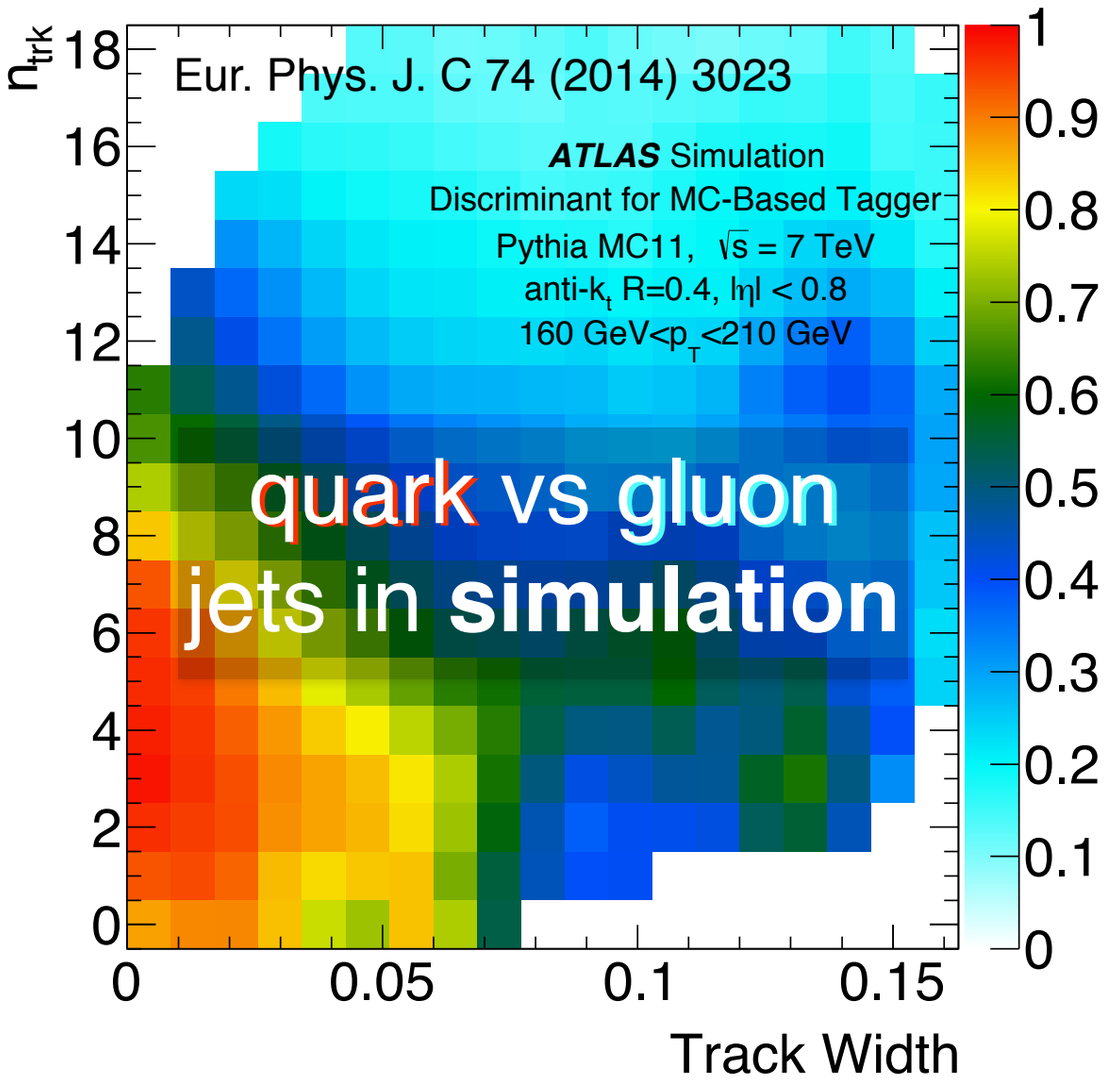
(1) train in simulation & calibrate in data

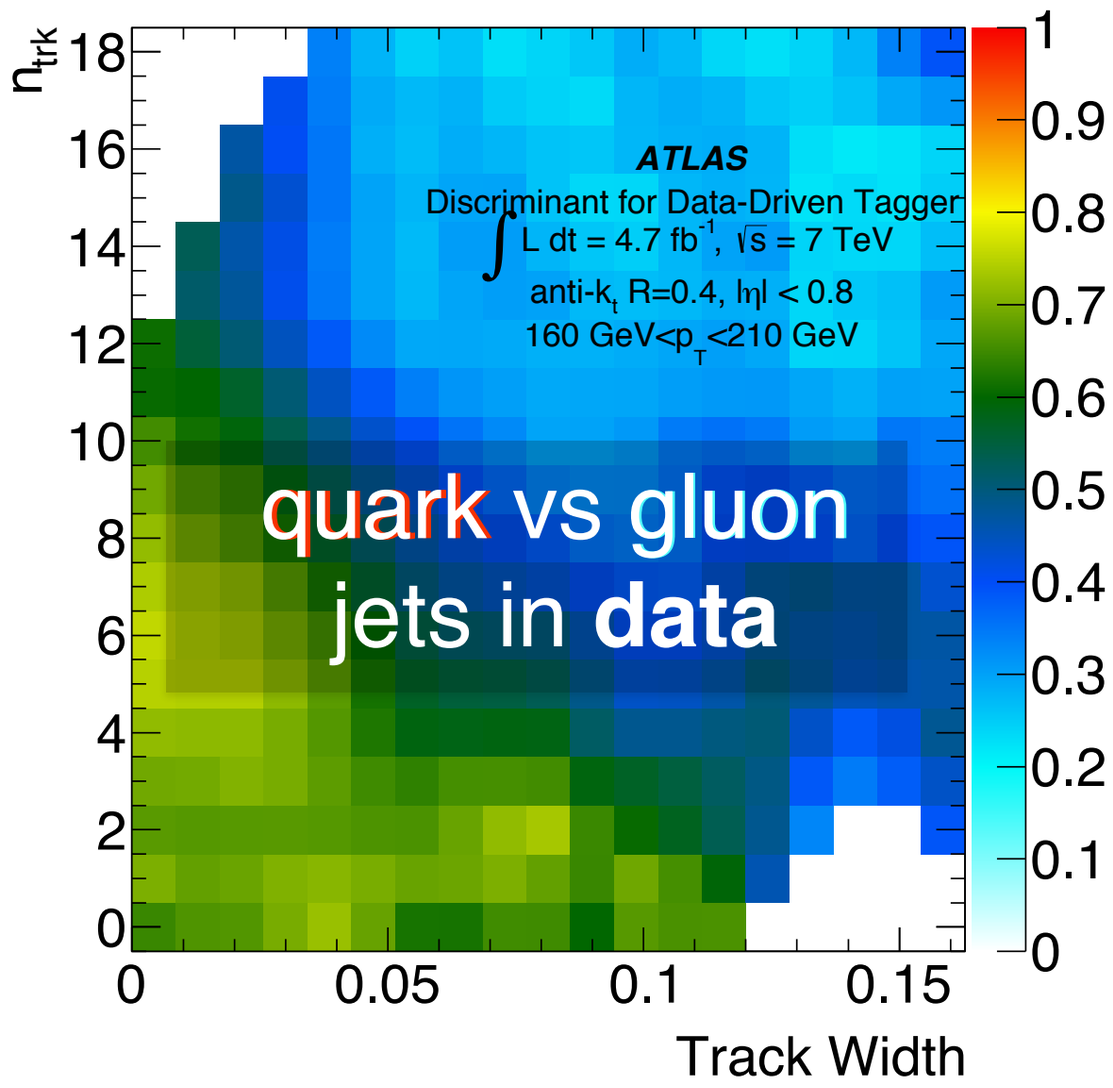
(2) train in data

← what we mostly do right now

Background and Motivation

Usual paradigm: train in simulation, validate on data, test on data.



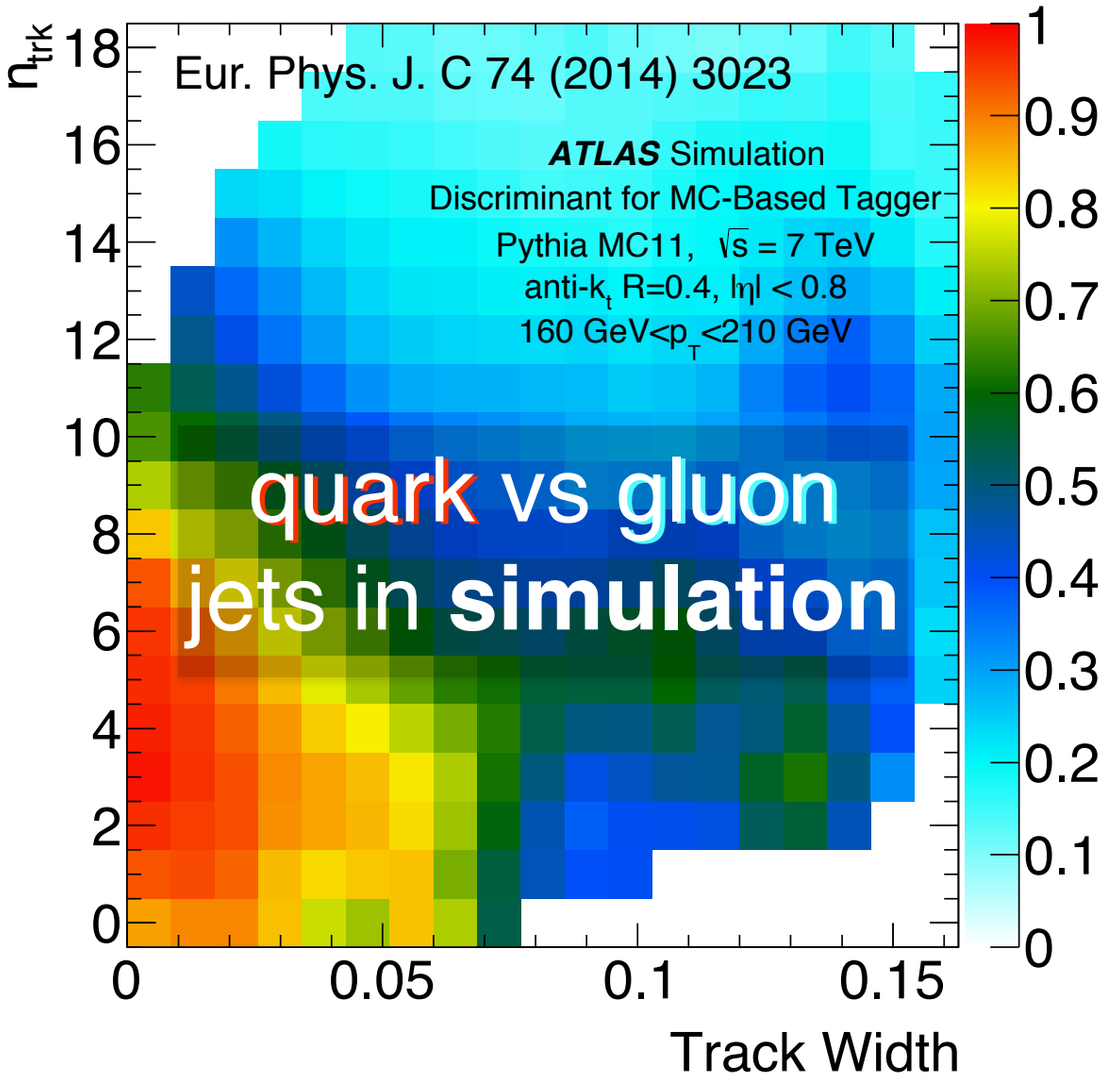


If data and simulation differ, this is sub-optimal!

(answer from previous slide is no!)

Background and Motivation

Usual paradigm: **train in simulation**, validate on data, test on data.



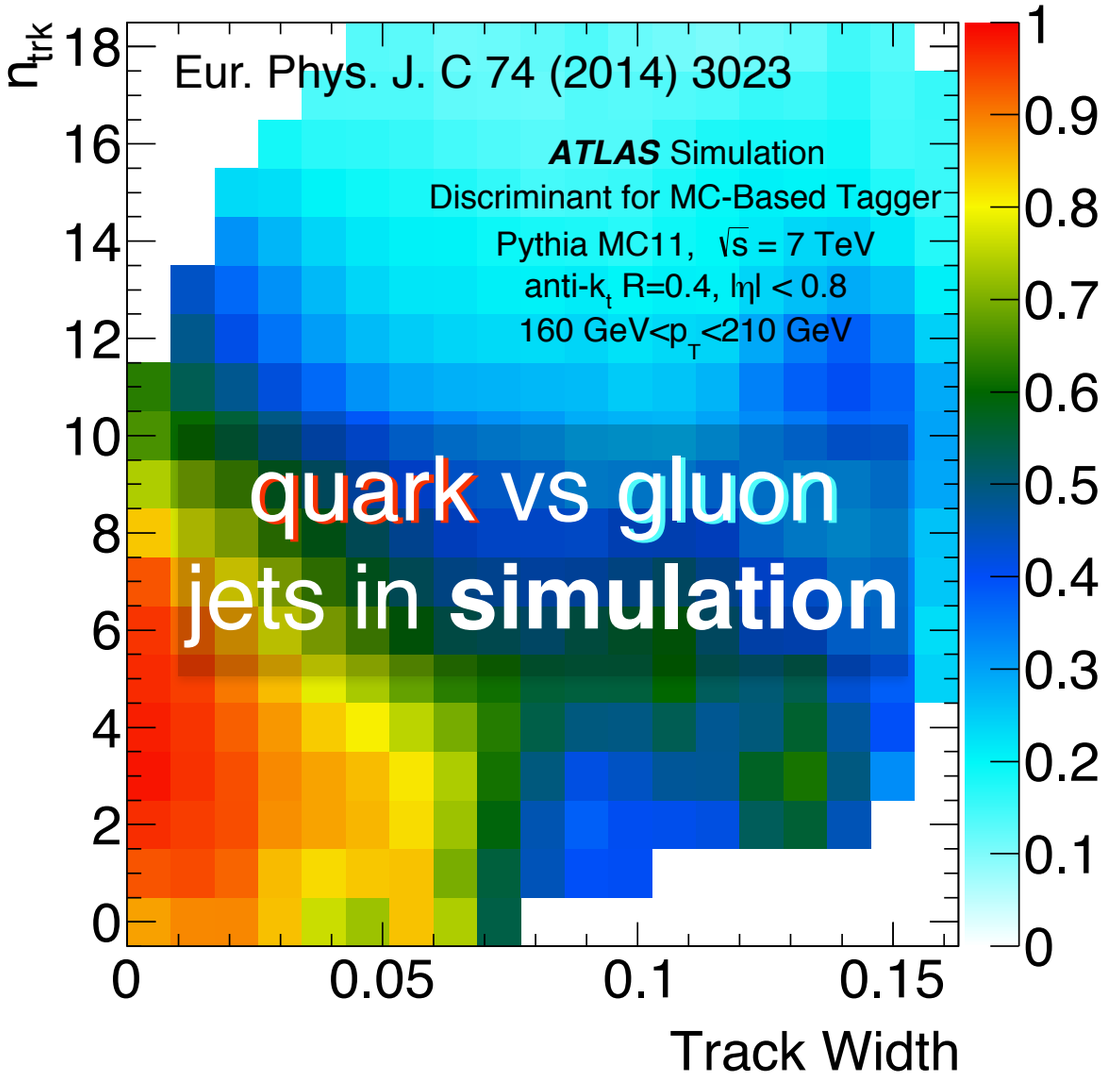
Recall: optimal classifier (by Neyman-Pearson NP) is a threshold cut on the likelihood ratio.

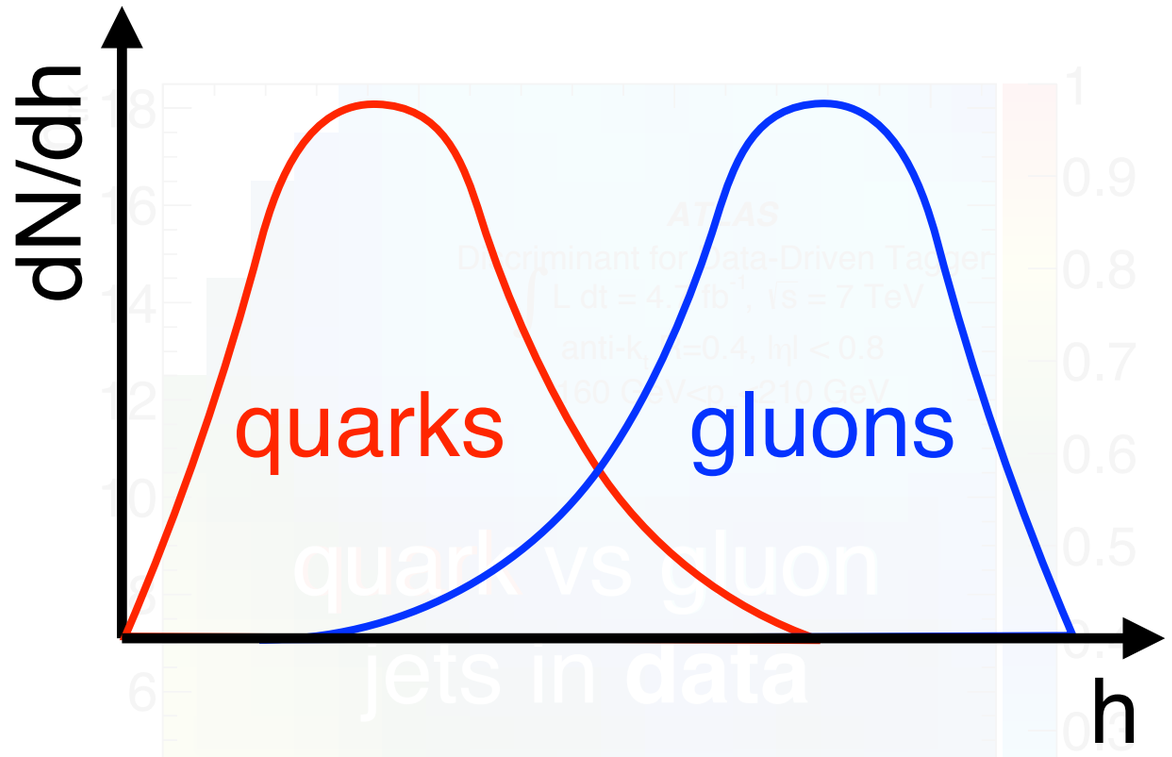
For a 2D feature space, no need for a NN or BDT - can use a histogram to “train” the classifier.

$h(n_{trk}, \text{Track Width}) \rightarrow [0,1]$

Background and Motivation

Usual paradigm: **train in simulation**, validate on data, test on data.



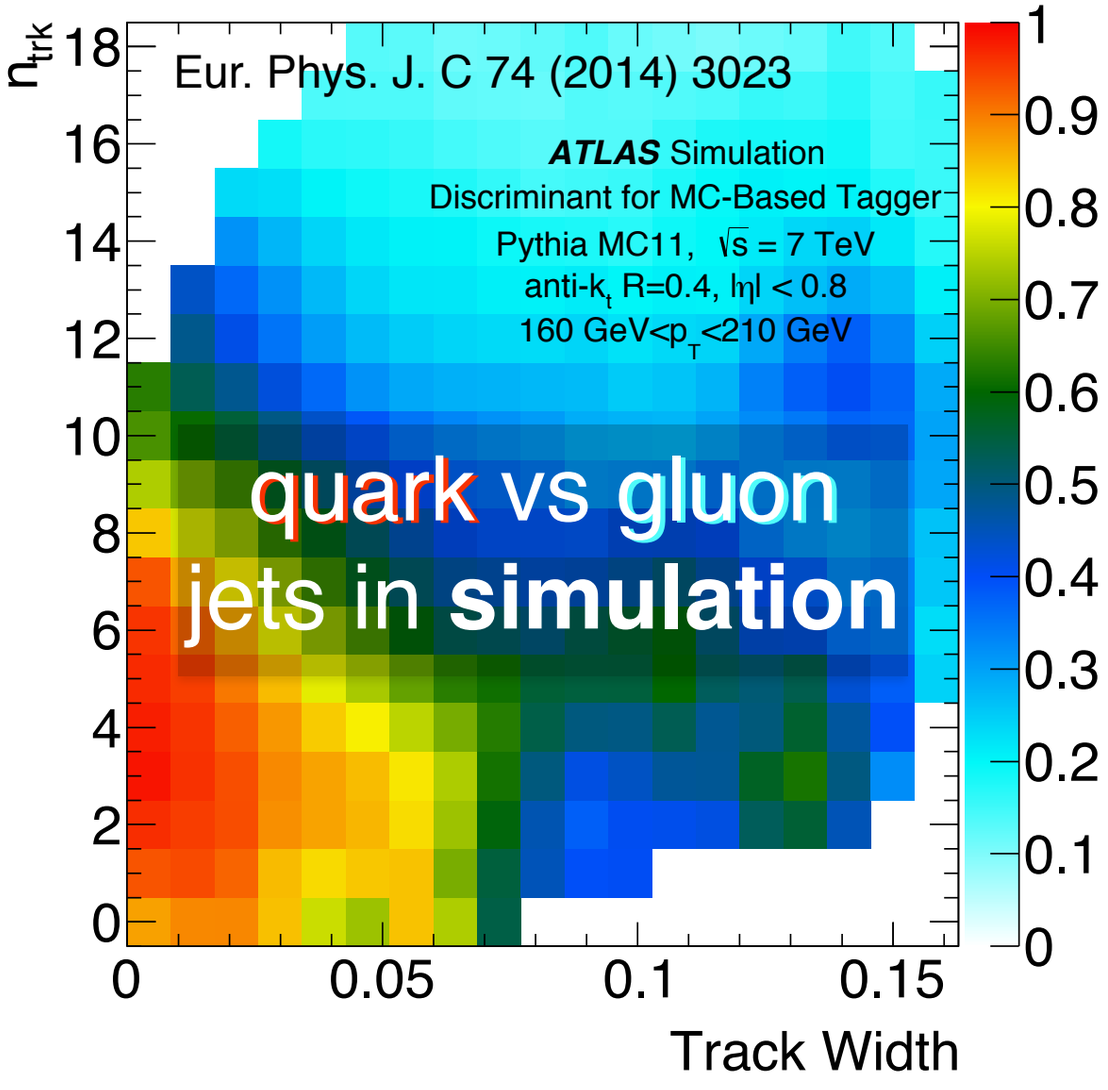


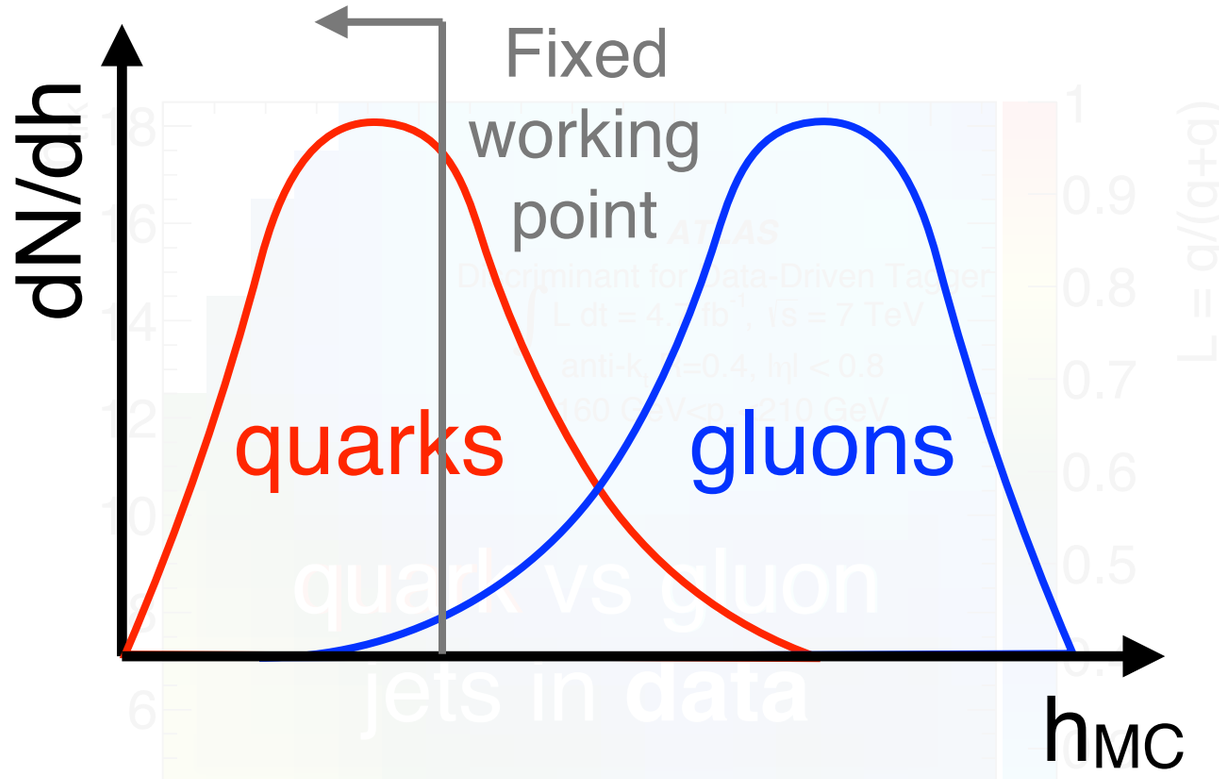
For a 2D feature space, no need for a NN or BDT - can use a histogram to “train” the classifier.

$h(n_{trk}, \text{Track Width}) \rightarrow [0,1]$

Background and Motivation

Usual paradigm: **train in simulation**, validate on data, test on data.



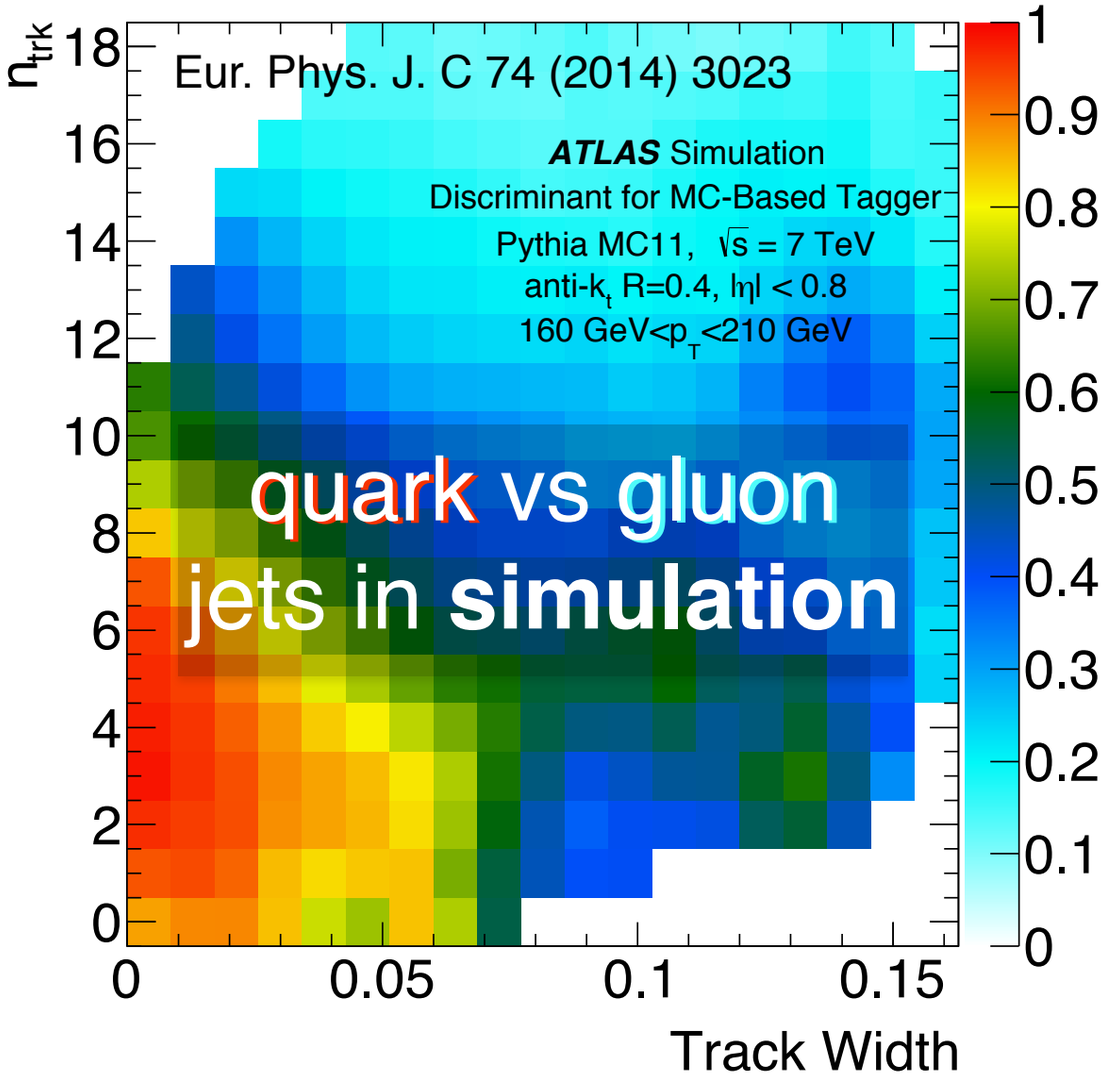


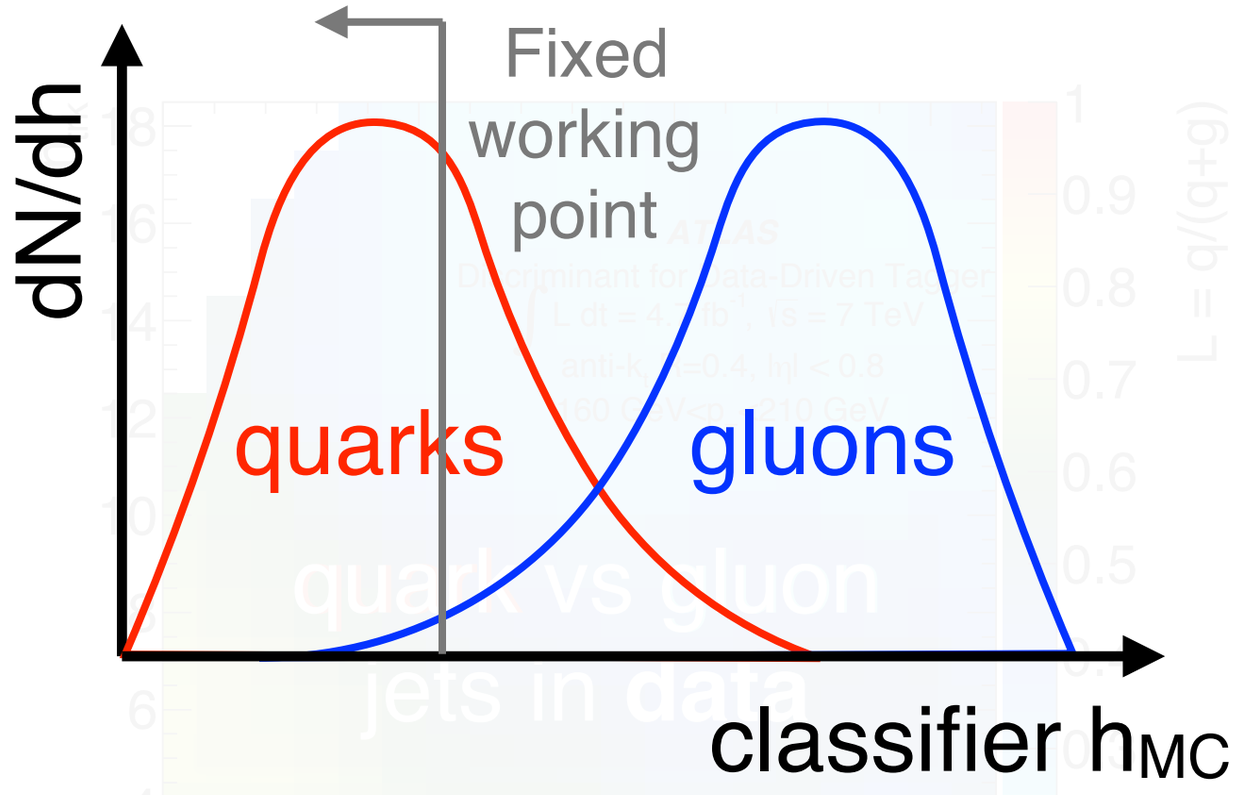
For a 2D feature space, no need for a NN or BDT - can use a histogram to “train” the classifier.

$h_{MC}(n_{trk}, \text{Track Width}) \rightarrow [0,1]$

Background and Motivation

Usual paradigm: **train in simulation**, validate on data, test on data.





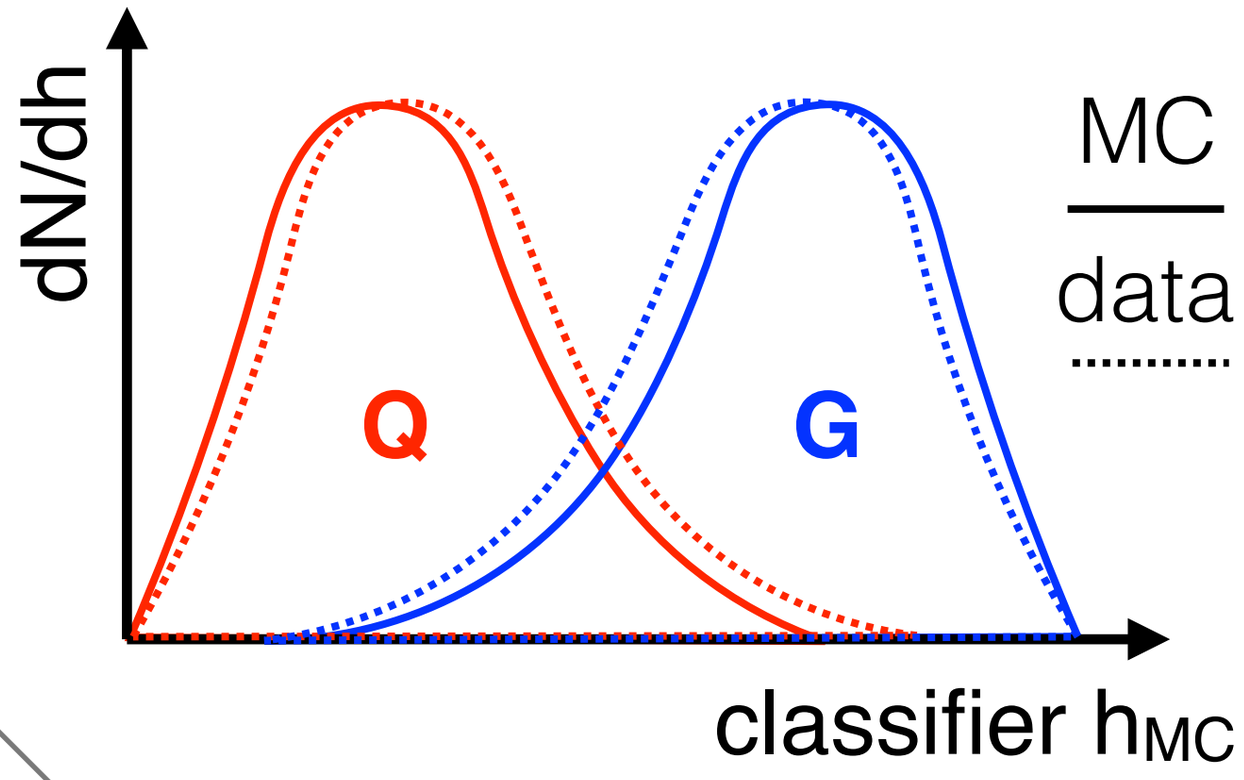
WP in simulation:

$\varepsilon_{signal,MC}$, $\varepsilon_{back,MC}$

Background and Motivation

Usual paradigm: train in simulation, **validate on data**, test on data.

Determine the performance of the WP in data.



How did we get this?

dijets = f_q x **Q** + (1-f_q) x **G**

Z+jets = g_q x **Q** + (1-g_q) x **G**

2 equations, 2 unknowns (**Q**, **G**)

two event samples with different q/g fractions

(N.B. f & g from simulation and selection can't bias Q and G)

Background and Motivation

Usual paradigm: train in simulation, **validate on data**, test on data.

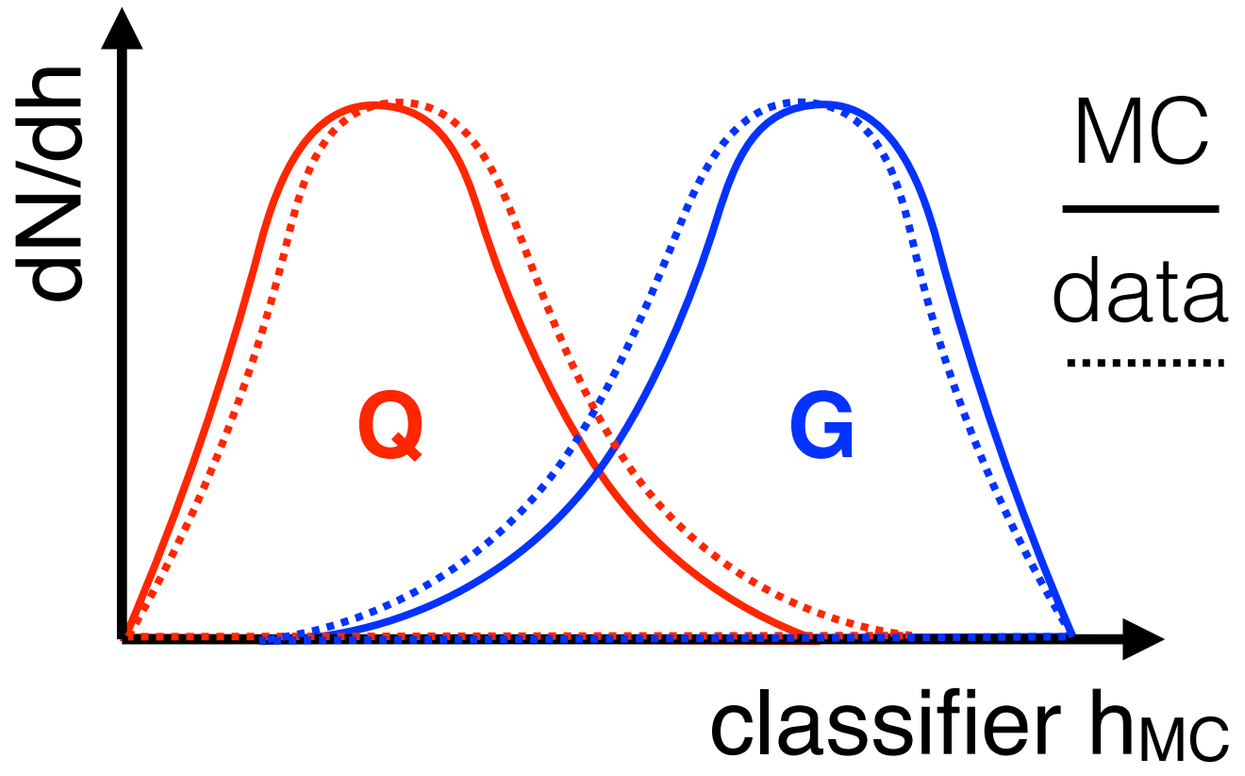
Determine the performance of the WP in data.

How did we get this?

dijets = f_q x **Q** + (1-f_q) x **G**

Z+jets = g_q x **Q** + (1-g_q) x **G**

2 equations, 2 unknowns (**Q**, **G**)



WP in data:

$\varepsilon_{signal,data}$, $\varepsilon_{back,data}$

Can correct the MC to have the same performance as data.

Background and Motivation

Usual paradigm: train in simulation, validate on data, **test on data**.

Once we have scale factors (& their uncertainty), we can ensure that our analysis will be **accurate**.

…so what is the problem?

remember my claim from earlier:

If data and simulation differ, this is sub-optimal!

This is an **accuracy** versus **precision** problem. It is "easy" to achieve accuracy through calibration, but the results may not be the best one possible.

Background and Motivation

In this 2D feature space, we can actually derive h_{data}.

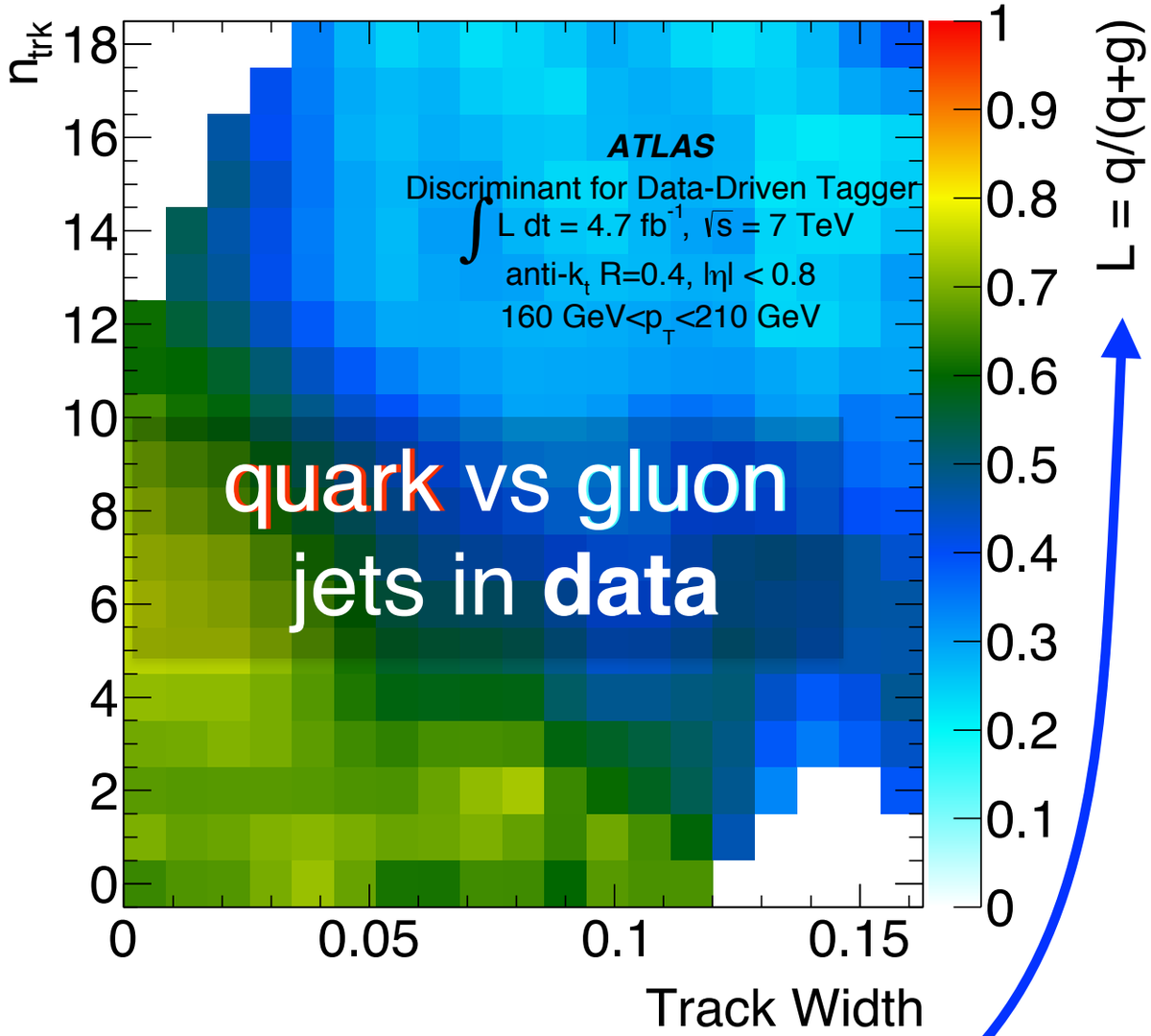
Using the same trick as earlier:

dijets = f_q x Q + (1-f_q) x G

Z+jets = g_q x Q + (1-g_q) x G

2 equations, 2 unknowns (Q, G)

(now Q and G are 2D histograms)



in general:

$h_{MC}(n_{trk}, \text{Track Width}) \neq h_{data}(n_{trk}, \text{Track Width})$

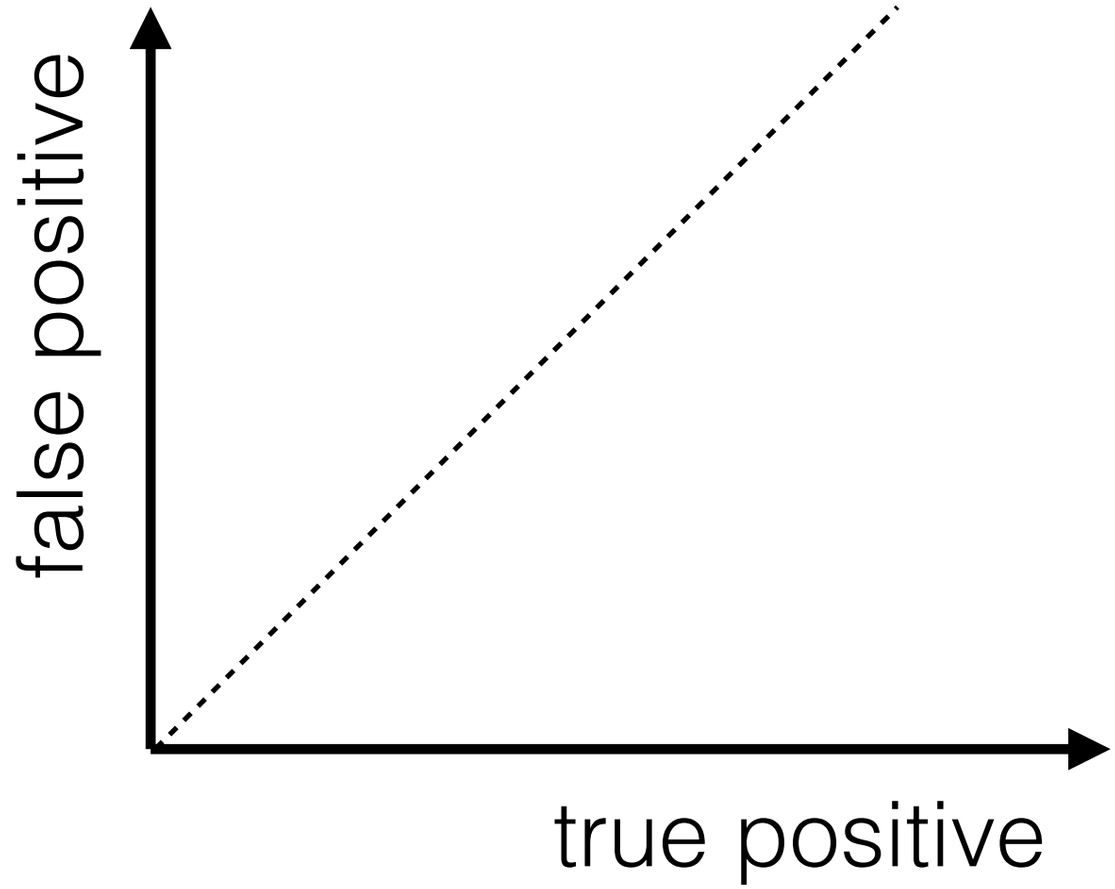
Take it to the extreme

To stress this point, suppose that h_{MC} is the random classifier:

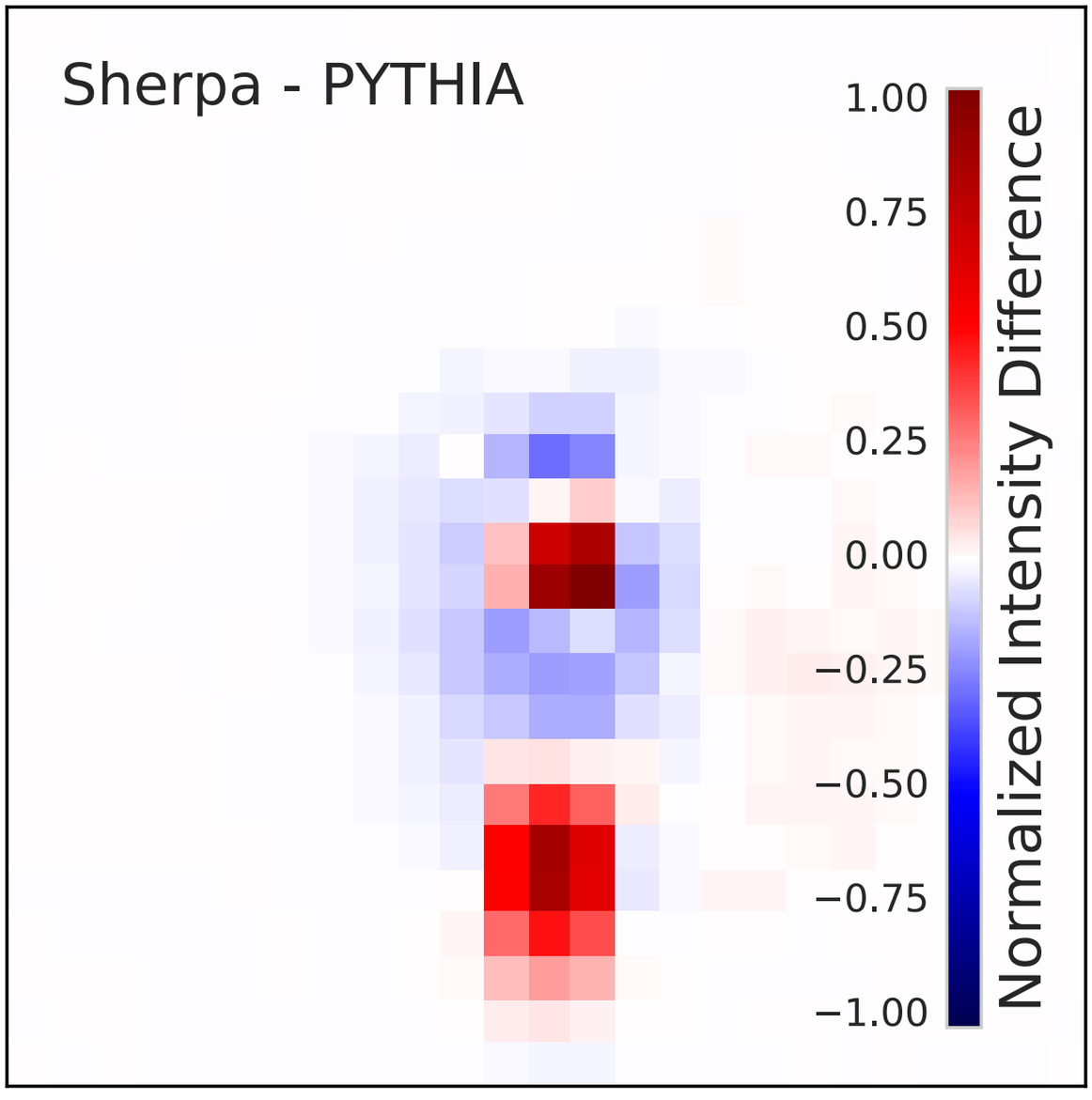
h_{MC} = 0 if you pick a random number x in [0,1] and x < ε

1 otherwise

We can calibrate this classifier in data, but clearly, it is sub-optimal !!



One more slide about why it matters



J. Barnard, E. Dawe, M. Dolan, N. Rajcic,
Phys. Rev. D 95 (2017) 014018

Especially important for **deep learning** using subtle features → hard to model!

W boson radiation pattern - same physics, different simulators!

Achieving the Optimal Classifier

Two ways around the problems mentioned earlier:

(1) Derive the classifier in MC, but don't let it use information that is not present in data.

"Learning to pivot"

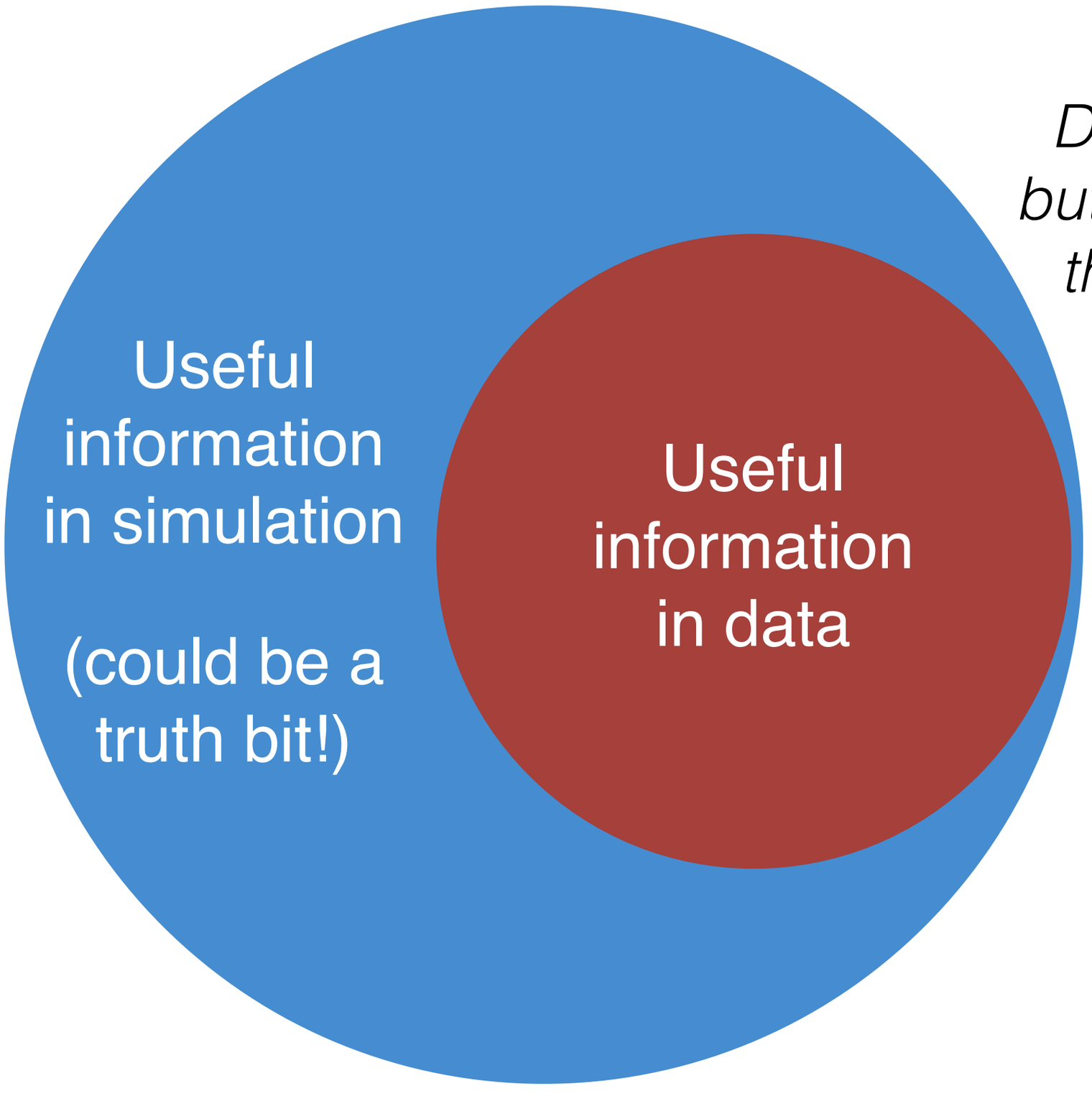
G. Louppe, M. Kagan, K. Cranmer, 1611.01046

(2) Train on unlabeled data.

"Weak supervision"

*L. Dery, **BPN**, F. Rubbo, A. Schwartzman, JHEP 05 (2017) 145*
*E. Metodiev, **BPN**, J. Thaler, JHEP 10 (2017) 174*

When option 1 works



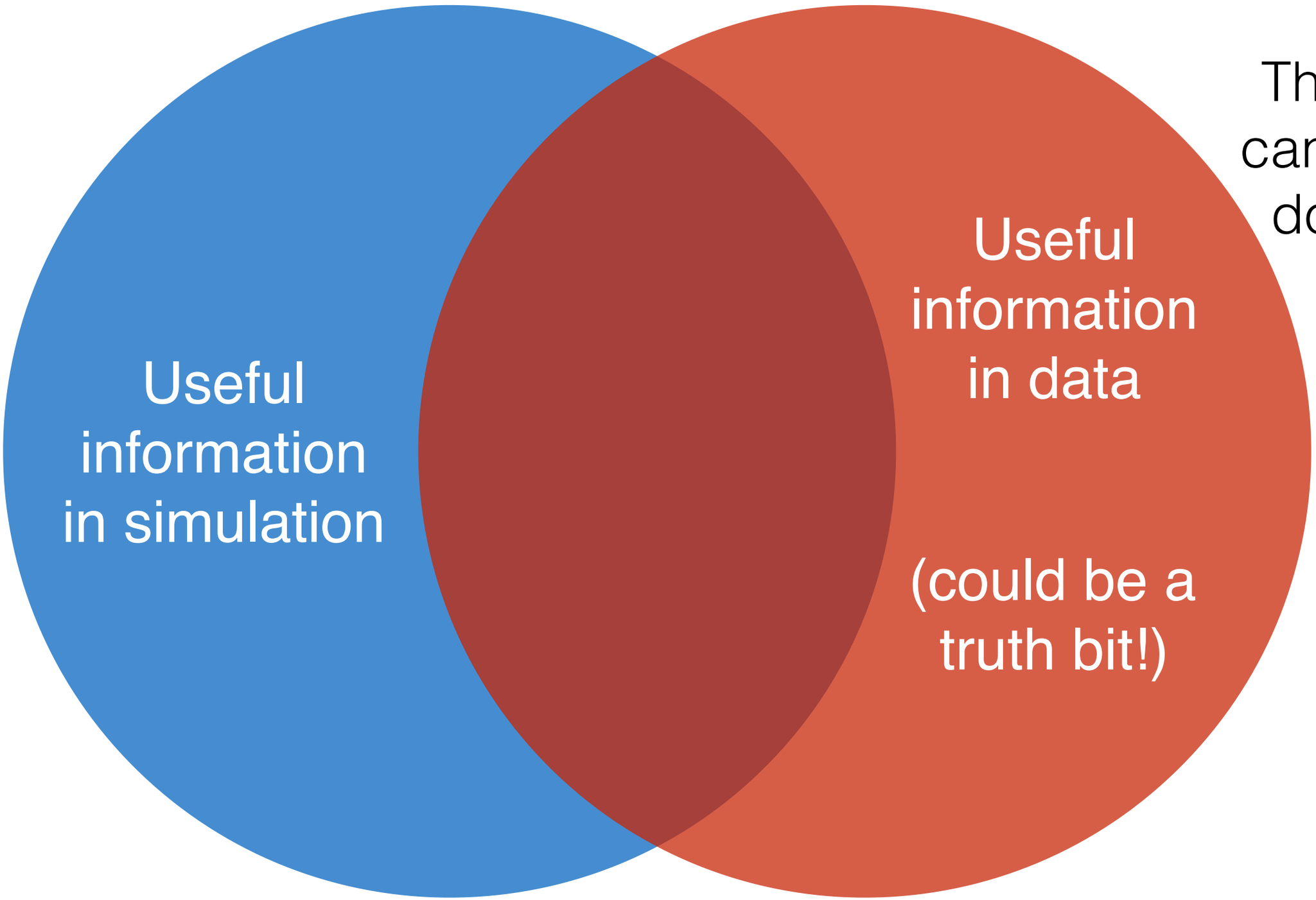
Option 1:

Derive the classifier in MC, but don't let it use information that is not present in data.

Classifier can't use information from simulation that is not useful in data and can learn the (data) optimal classifier.

When option 1 is sub-optimal

The simulation can't use what it doesn't know.

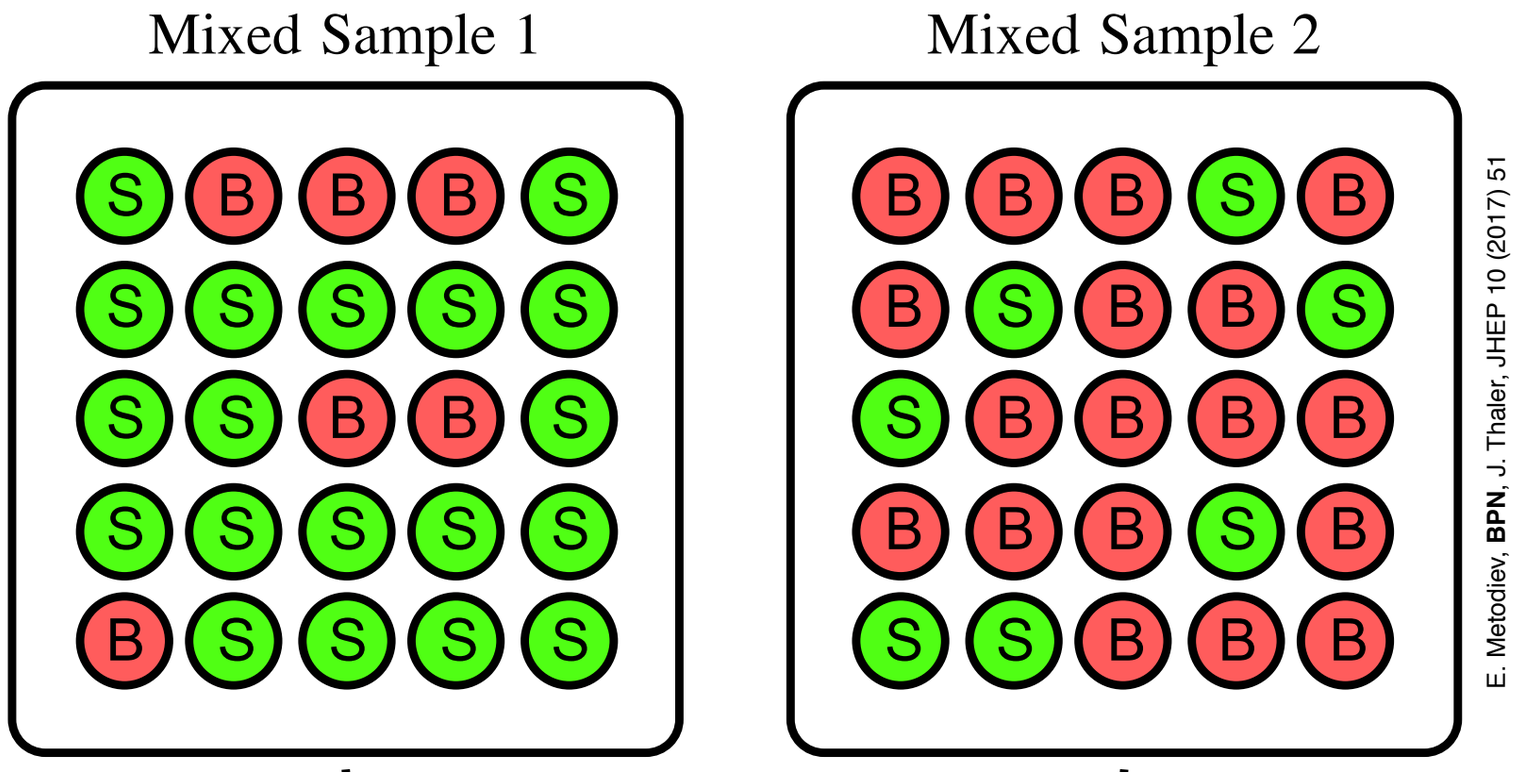


…many other applications of this approach, such as reducing sensitivity to systematic uncertainties, unwanted correlations between features, etc.

Option 2: Learn from data!

Challenge is that data are **unlabeled**.

The setup: suppose you have (at least) two mixed samples, each composed of two classes (say **q** and **g**).



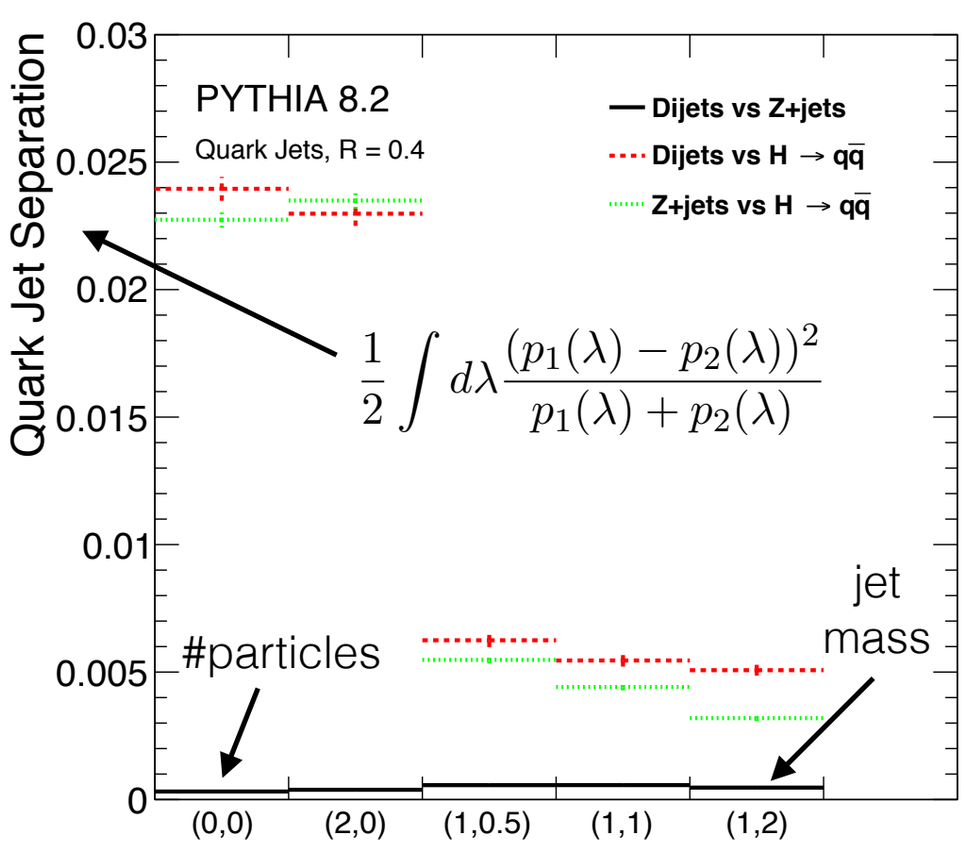
Assumption: The two classes are well-defined i.e. **q** in sample 1 is statistically identical to **q** in sample 2.

There is an interesting connection between what I'm calling "weak supervision" and the topic of "label noise".

Weak supervision, caveats up front

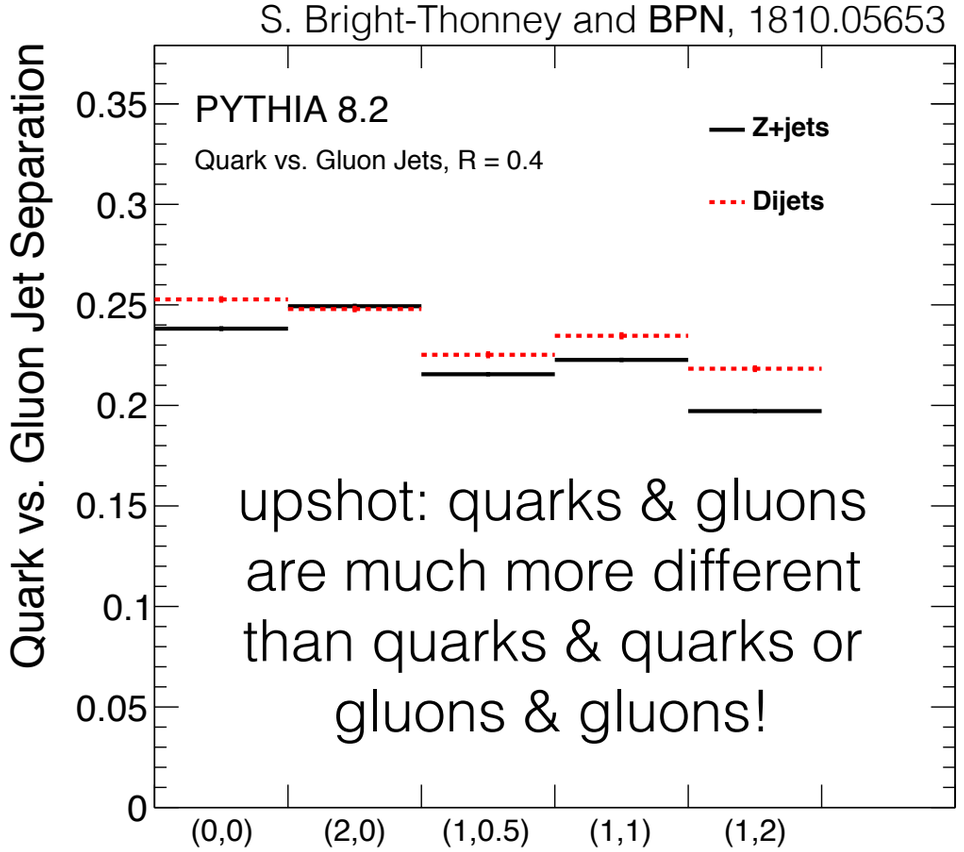
The two classes are well-defined (i.e. q in sample 1 is statistically identical to q in sample 2).

This is often not exactly true, but is often nearly true.



parameterizes jet features ⟶ (κ,β)

S. Bright-Thonney and **BPN**, 1810.05653



(κ,β)

Weak sup. option 1: Use class proportions

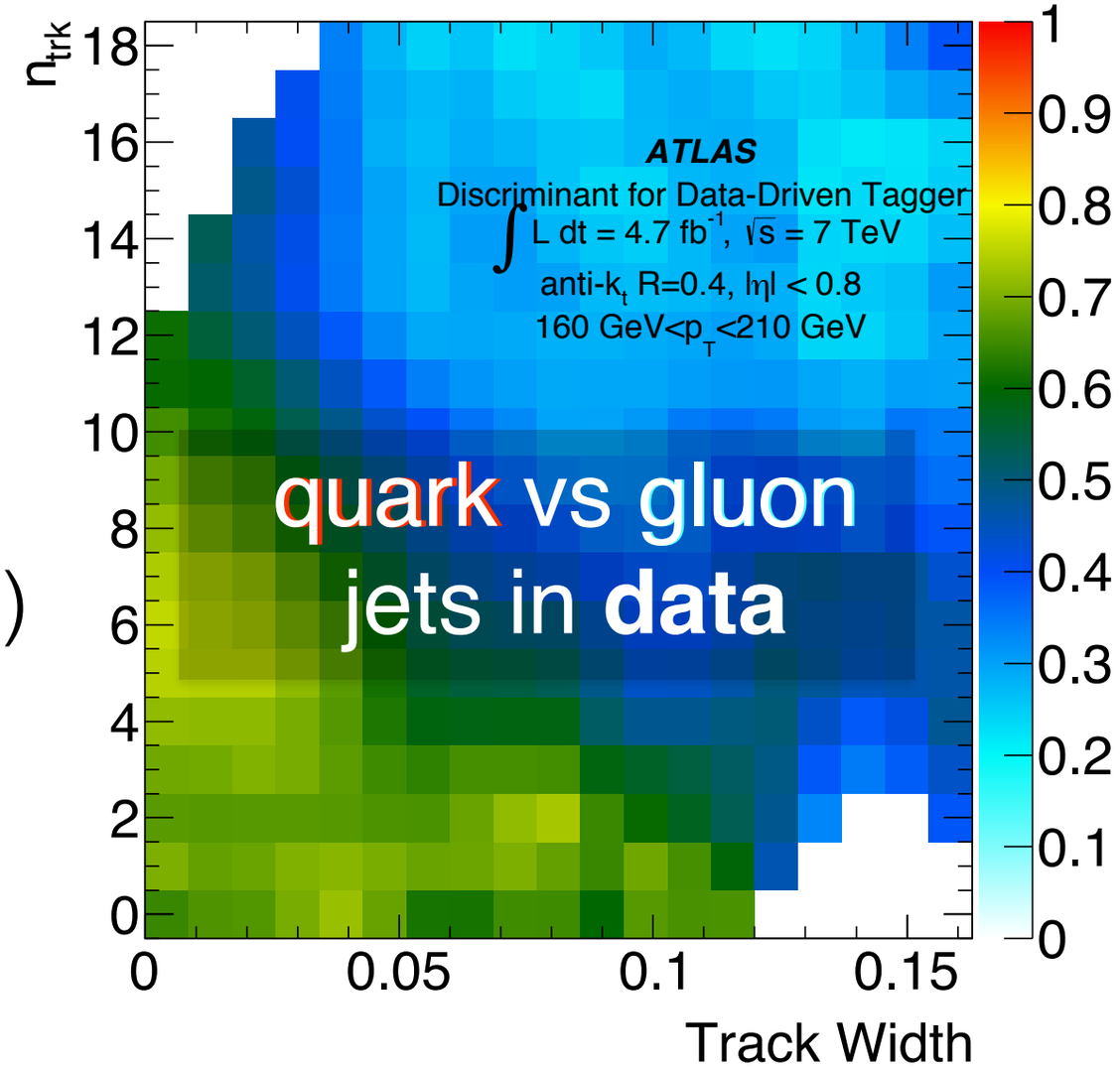
Remember this plot?

$$\text{dijets} = f_q \times Q + (1-f_q) \times G$$

$$\text{Z+jets} = g_q \times Q + (1-g_q) \times G$$

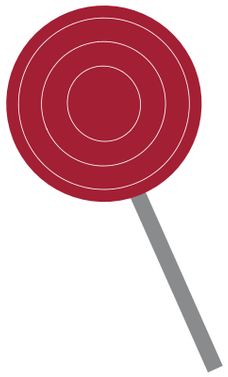
two equations, two unknowns (Q, G)

We often know f, g
(from ME + PDF) much better than
full radiation pattern inside jets.



This doesn’t work well when you have more than 2 observables because the templates become sparse.

Method 1: Learn from Proportions



$$f_{\text{full}} = \text{argmin}_{f':\mathbb{R}^n \to \{0,1\}} \sum_{i=1}^{N} \ell\left(f'(x_i) - t_i\right)$$

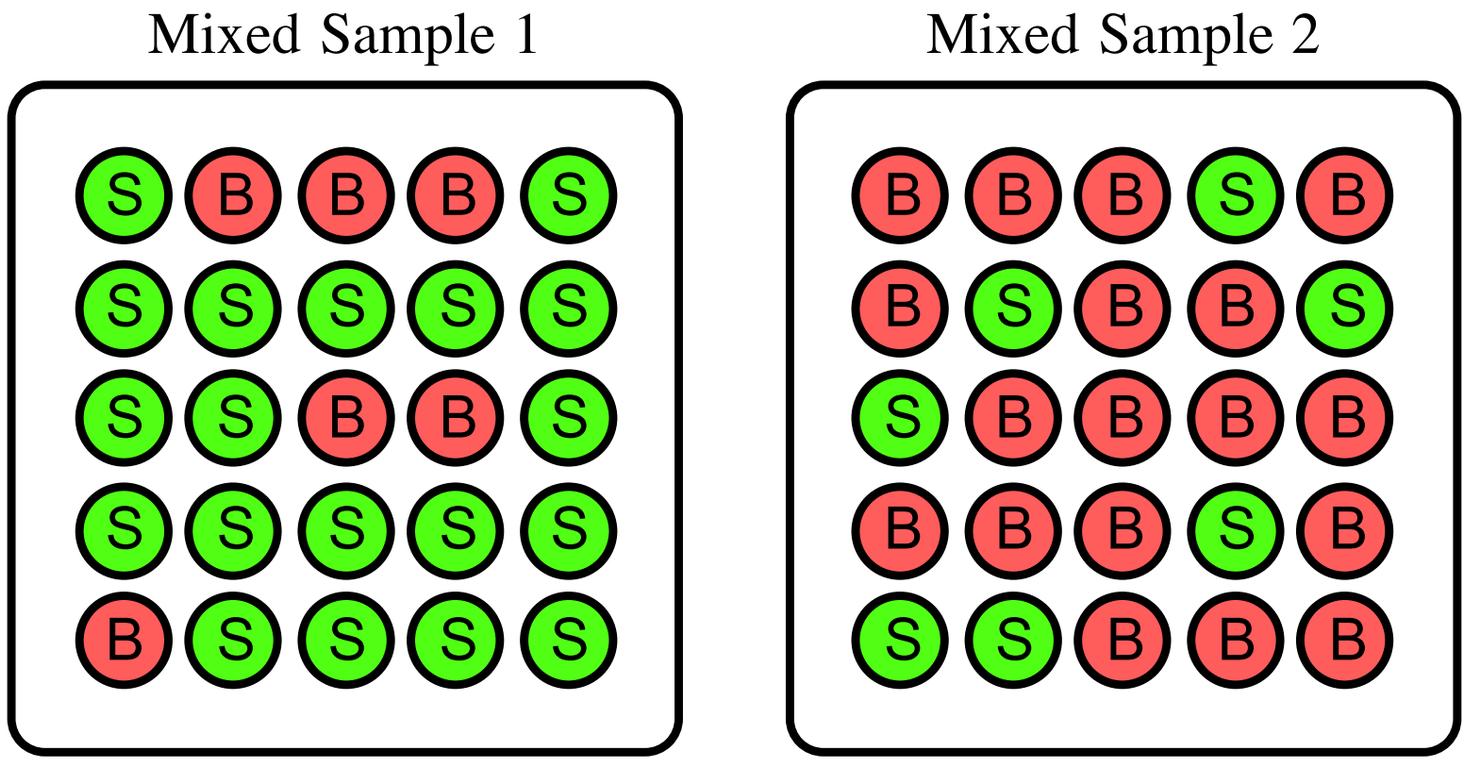
loss fcn.

labels

LoLiProp

Learning from Label Proportions

Solution: Train using class proportions. Work "on average"

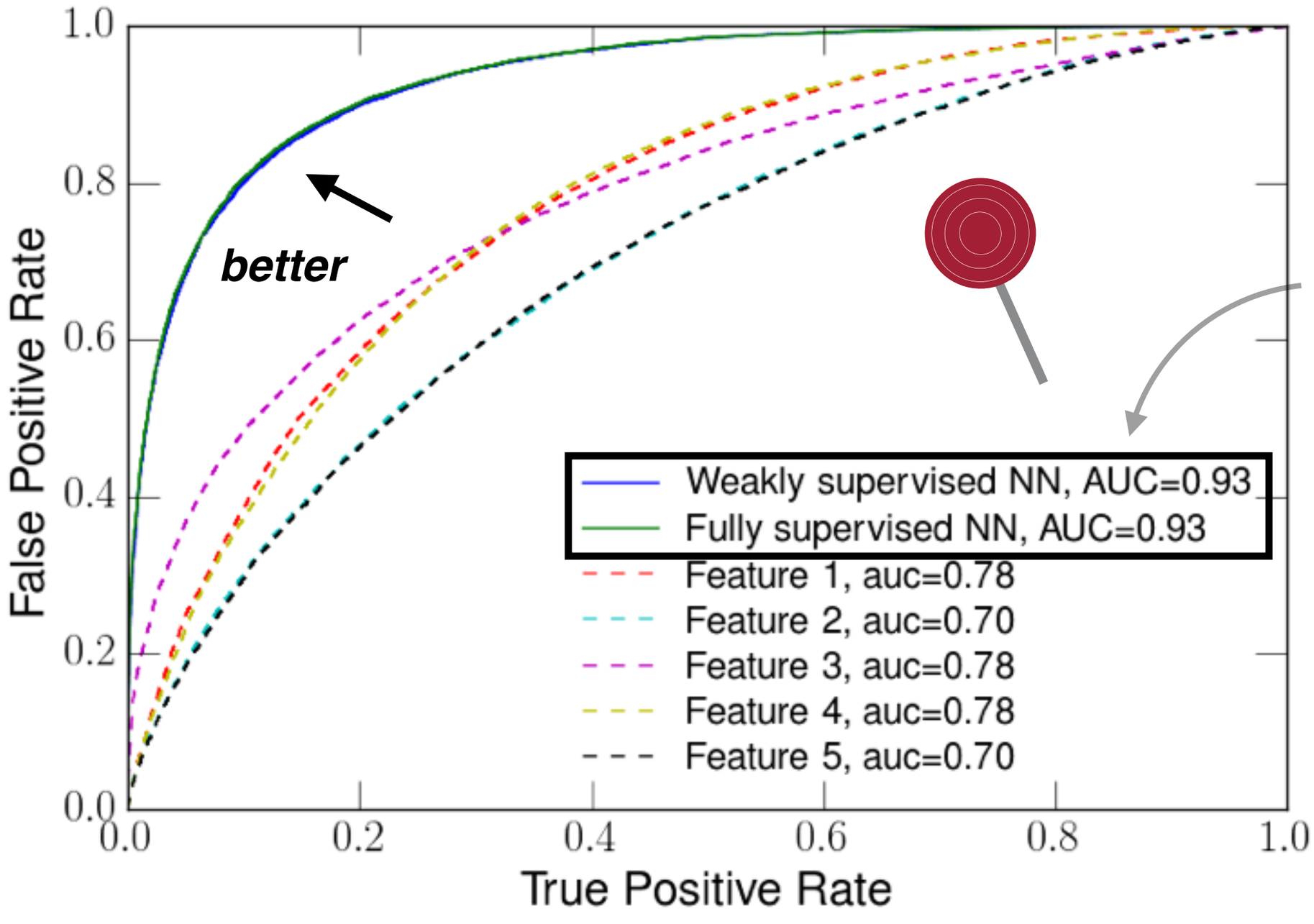


$$f_{\text{weak}} = \text{argmin}_{f':\mathbb{R}^n \to [0,1]} \ell\left(\sum_{i=1}^{N} \frac{f'(x_i)}{N} - y\right)$$

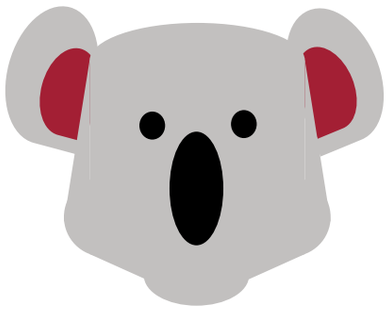
proportions

L. Dery, **BPN**, F. Rubbo, A. Schwartzman, JHEP 05 (2017) 145

Works!


1.0
0.8
0.6
0.4
0.2
0.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
True Positive Rate
better
area under
the curve
Weakly supervised NN, AUC=0.93
Fully supervised NN, AUC=0.93
Feature 1, auc=0.78
Feature 2, auc=0.70
Feature 3, auc=0.78
Feature 4, auc=0.78
Feature 5, auc=0.70

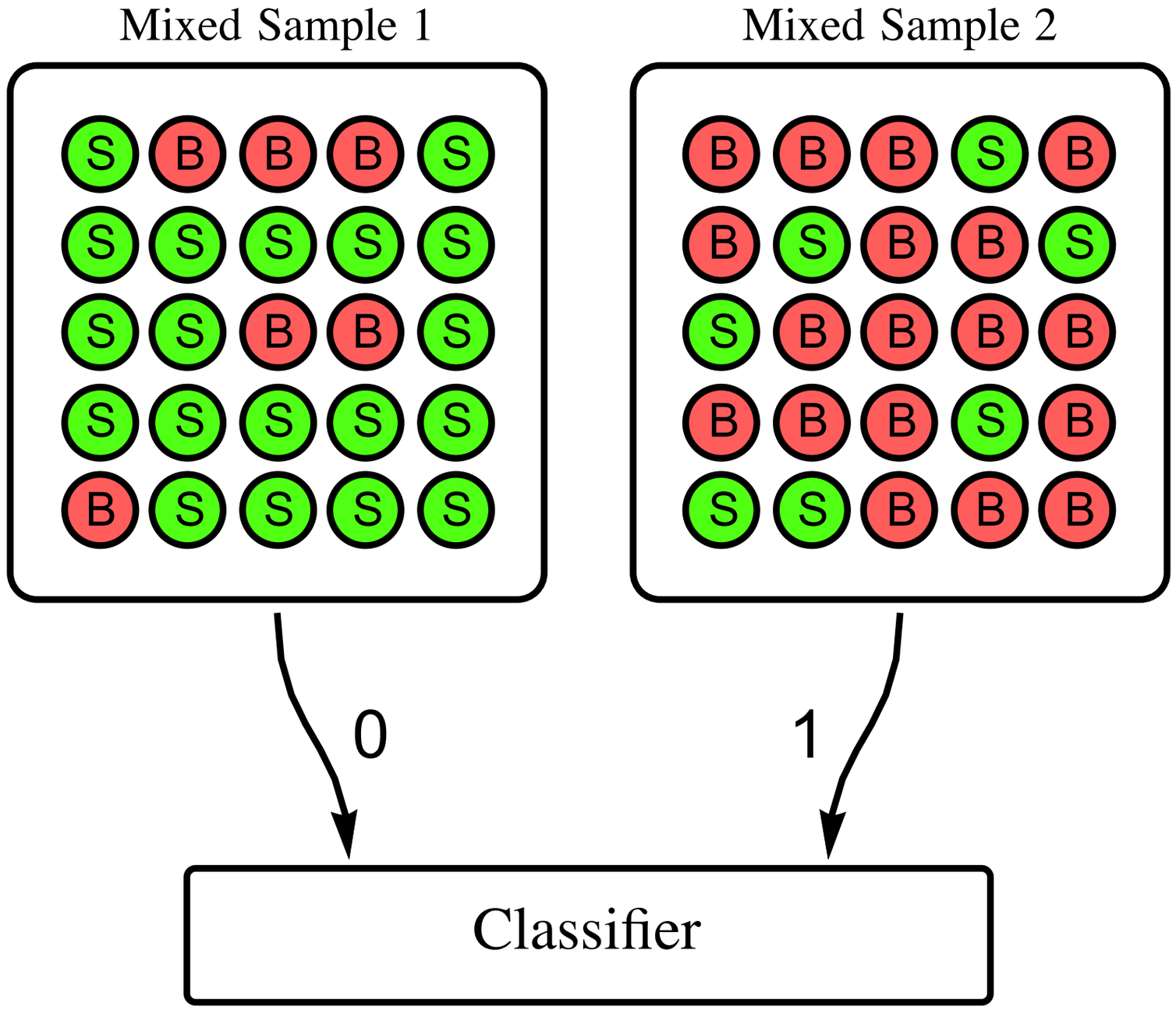
Method 2: Learning without Proportions



CWoLa

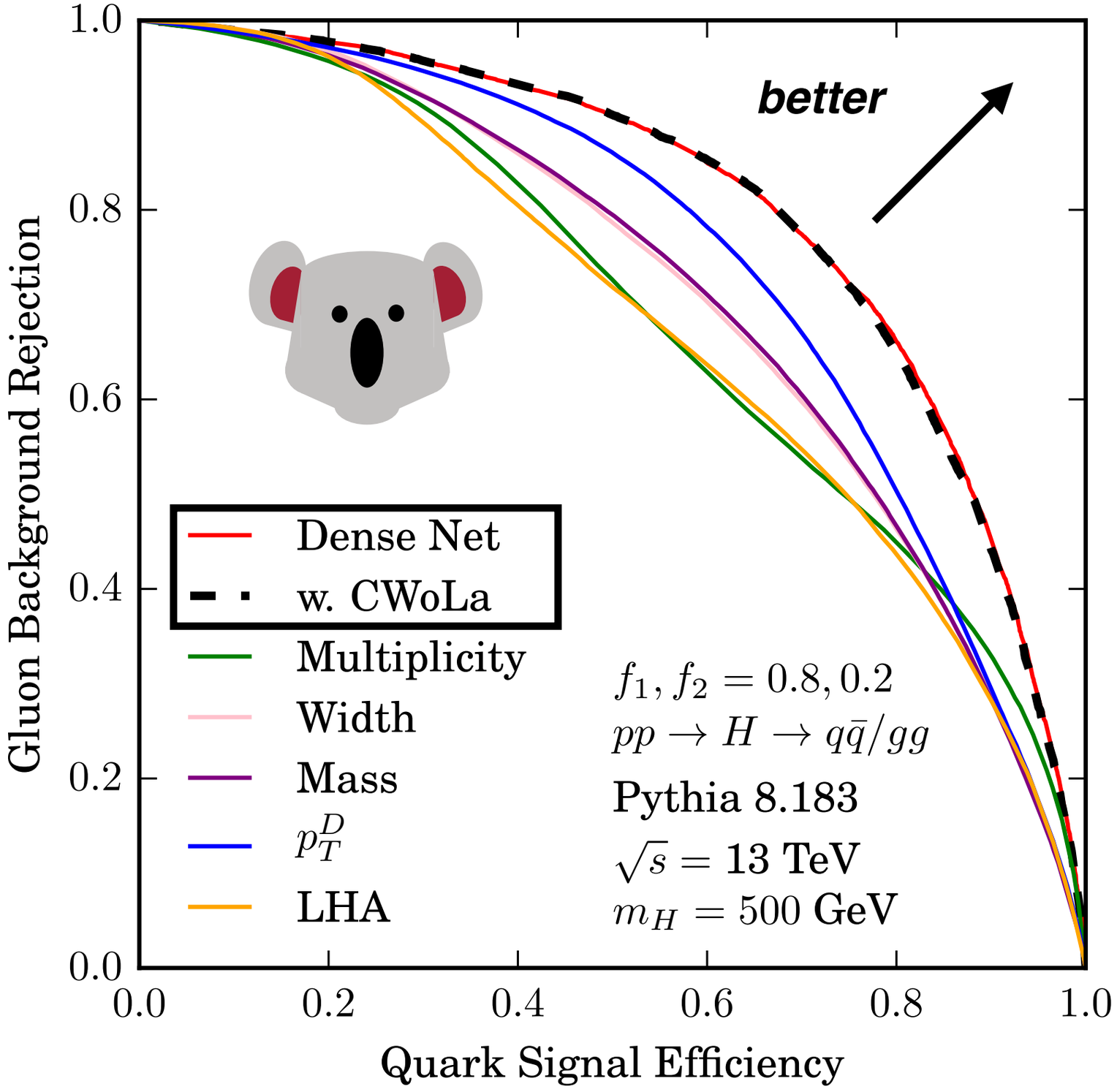
Classification Without Labels

Solution: Train **directly on data** using mixed samples

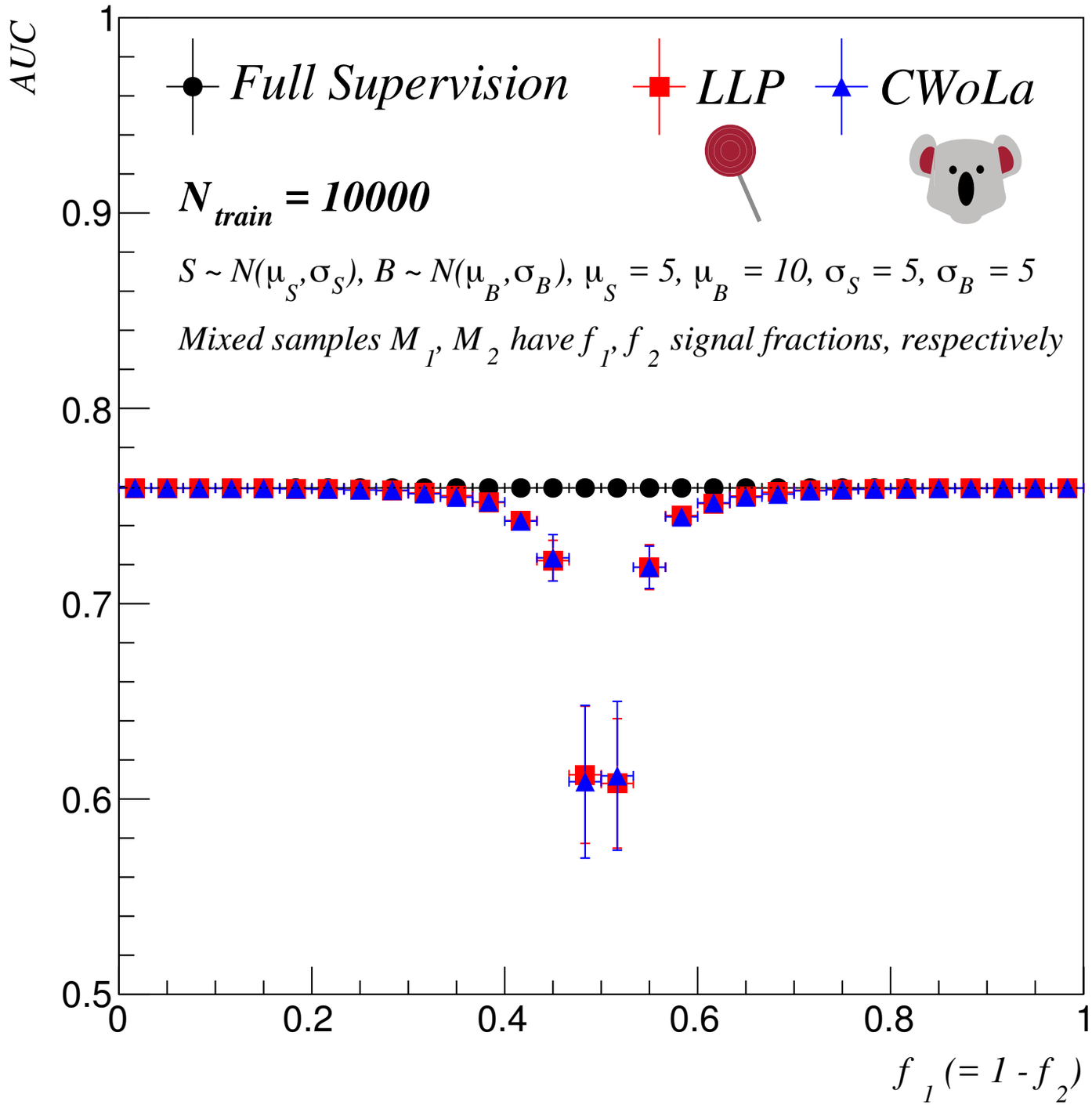


E. Metodiev, **BPN**, J. Thaler, JHEP 10 (2017) 51

Works!


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Gluon Background Rejection
Quark Signal Efficiency
better
Dense Net
w. CWoLa
Multiplicity
Width
Mass
p_T^D
LHA
$f_1, f_2 = 0.8, 0.2$
$pp \to H \to q\bar{q}/gg$
Pythia 8.183
$\sqrt{s} = 13$ TeV
$m_H = 500$ GeV

A note about training statistics



signal fraction of mixed sample 1

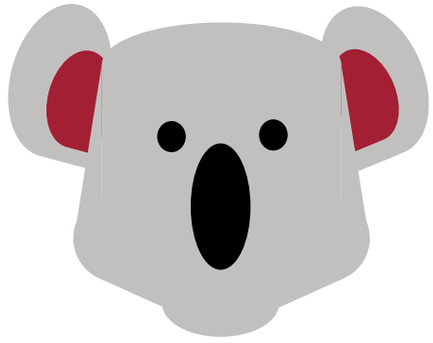
Can't learn when the two proportions are the same.

The more similar they are, the worse the performance.

(lower effective statistics)

CWoLa Hunting for BSM

- Why model agnostic searches?



Intermezzo: Learning directly from data

- Why it is important to learn directly from data

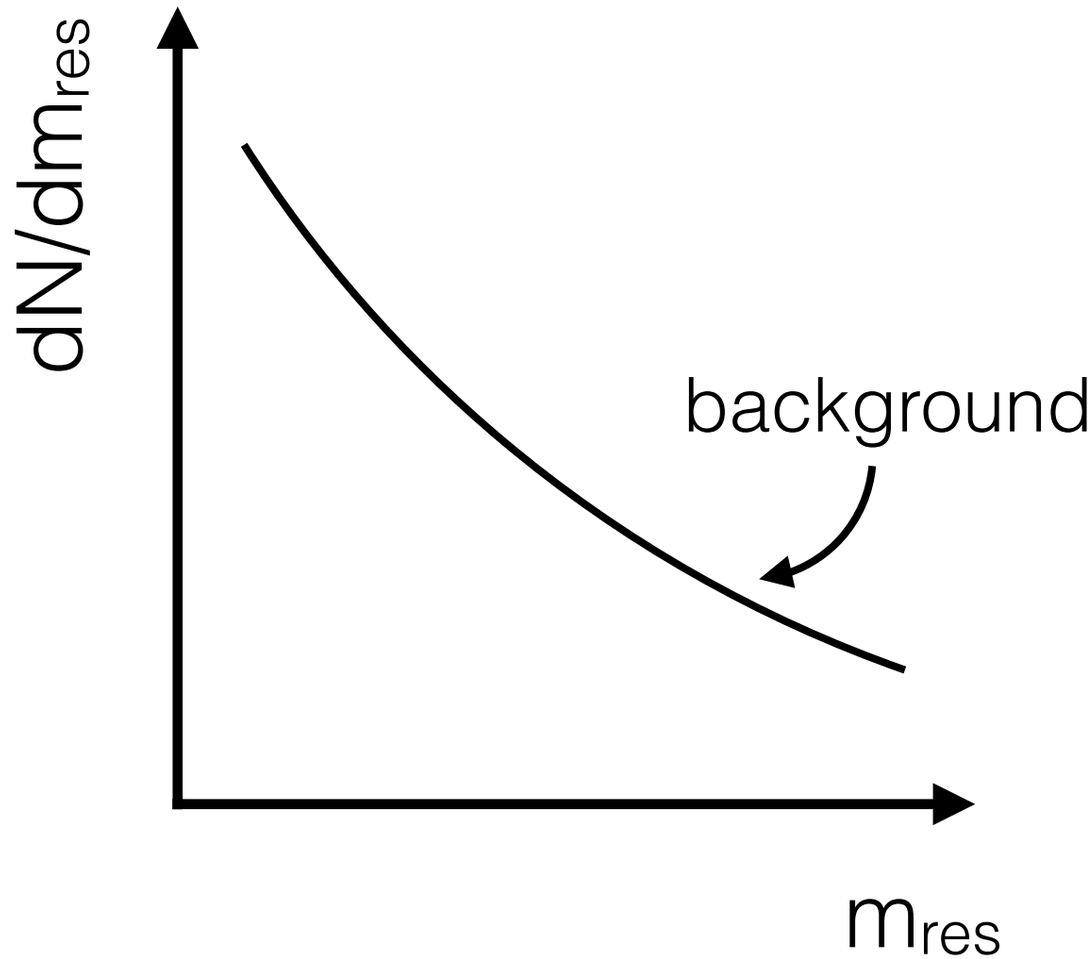
- How to learn without labels

- BSM Searches background / signal model independent

Most searches at the LHC (with or without ML)	Data versus SM simulation (“general search”)
CWoLa Hunting (focus of this talk)	Autoencoders

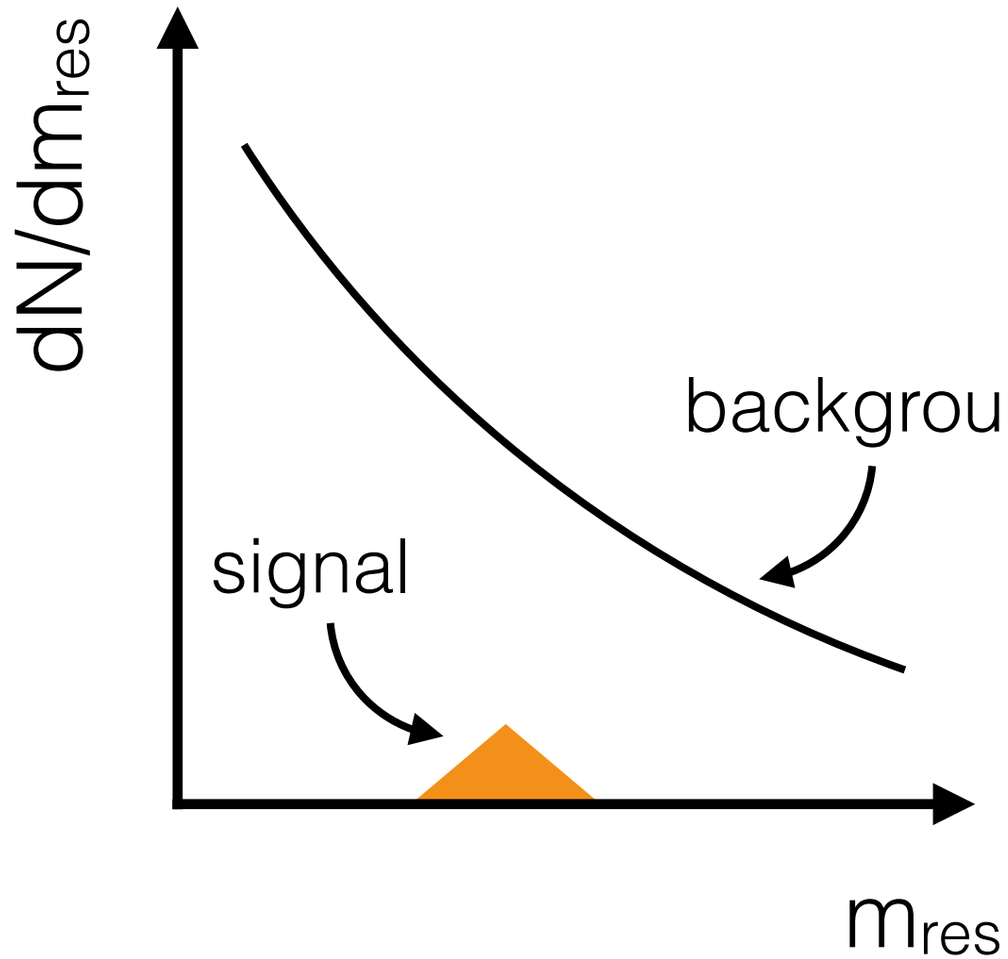
- The future

CWoLa Hunting for BSM



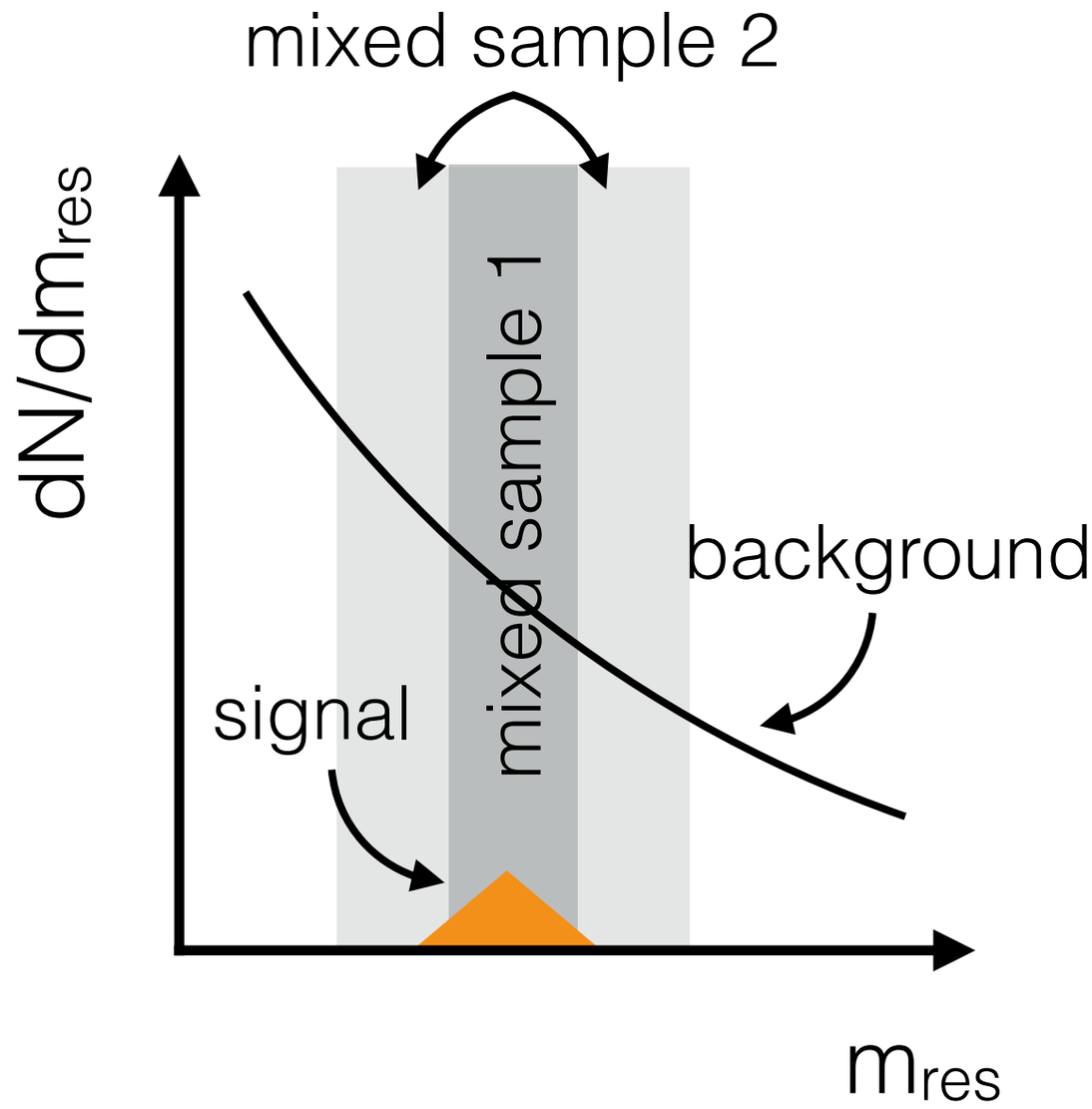
Assumption: there is a feature that we know about where the background is smooth and the signal (if it exists) is localized.

CWoLa Hunting for BSM



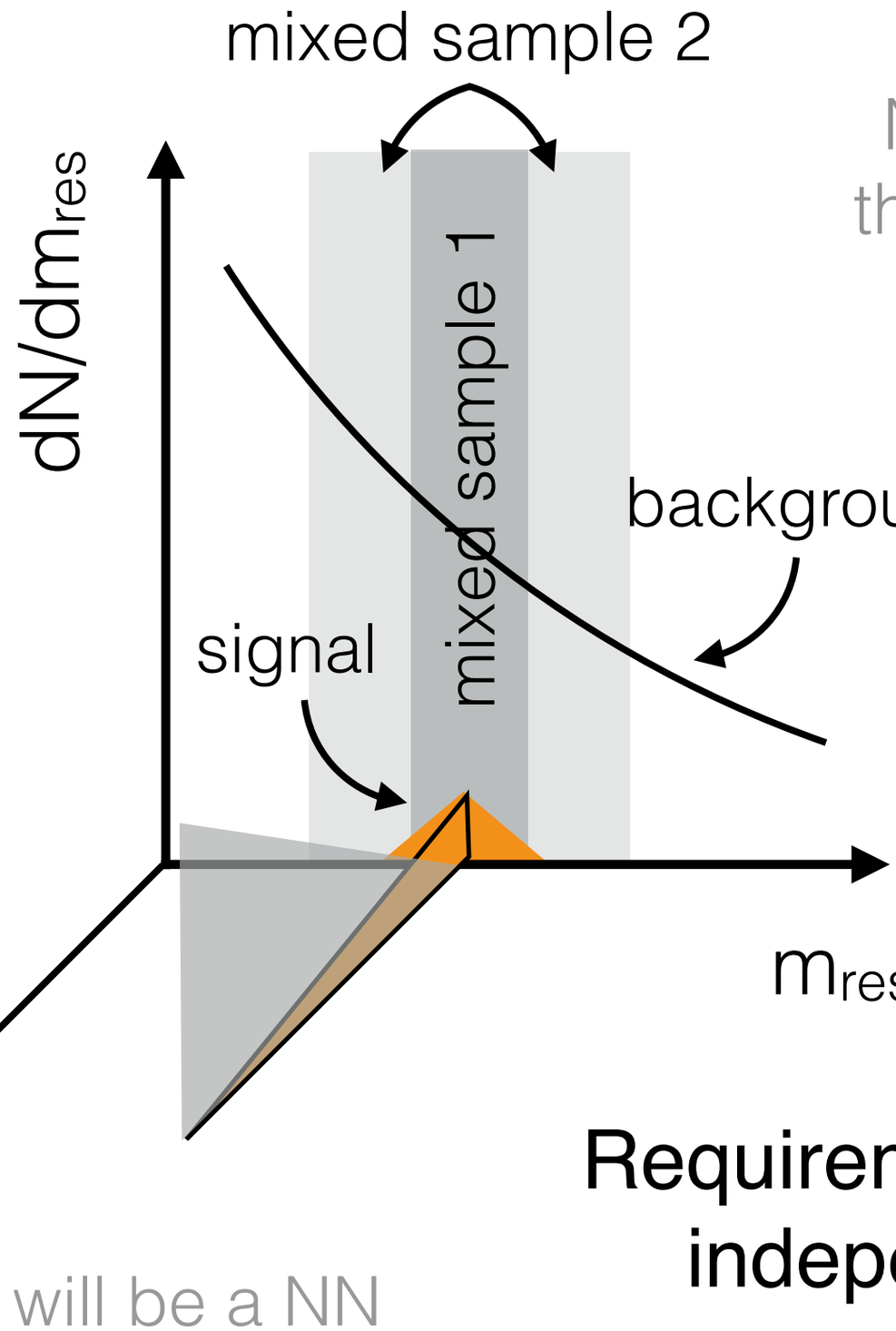
Assumption: there is a feature that we know about where the background is smooth and the signal (if it exists) is localized.

CWoLa Hunting for BSM



We don’t know where the signal is, but for a given hypothesis, we can make signal windows and sidebands.

CWoLa Hunting for BSM



N.B. it is okay that there is some signal in the sidebands!

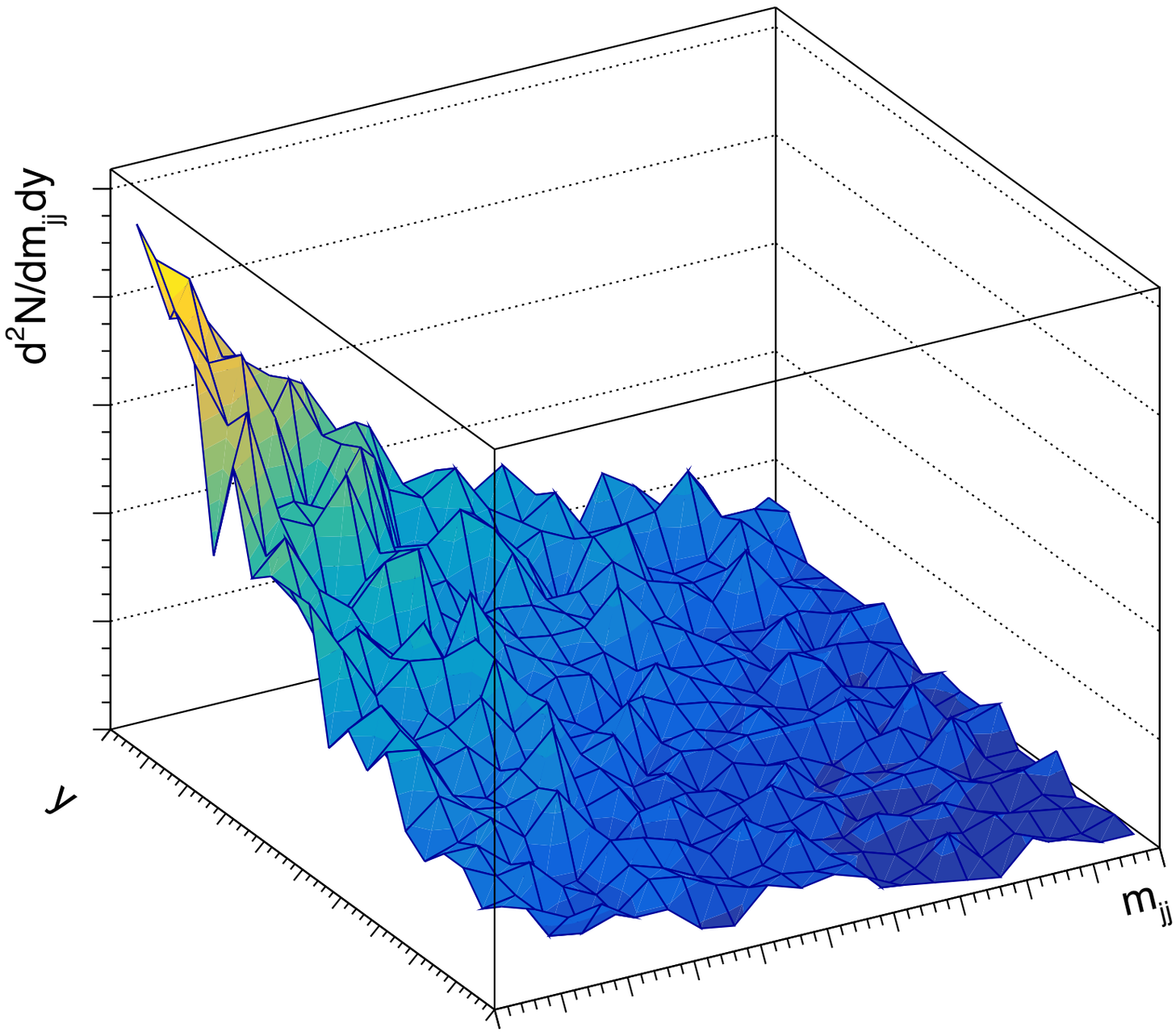
Other features can then be used to train CWoLa.

Requirement: y is (nearly) independent of m_{res}.

Overtraining & Look Elsewhere Effect

Naively, pay a huge LEE penalty because y can be high-dimensional.

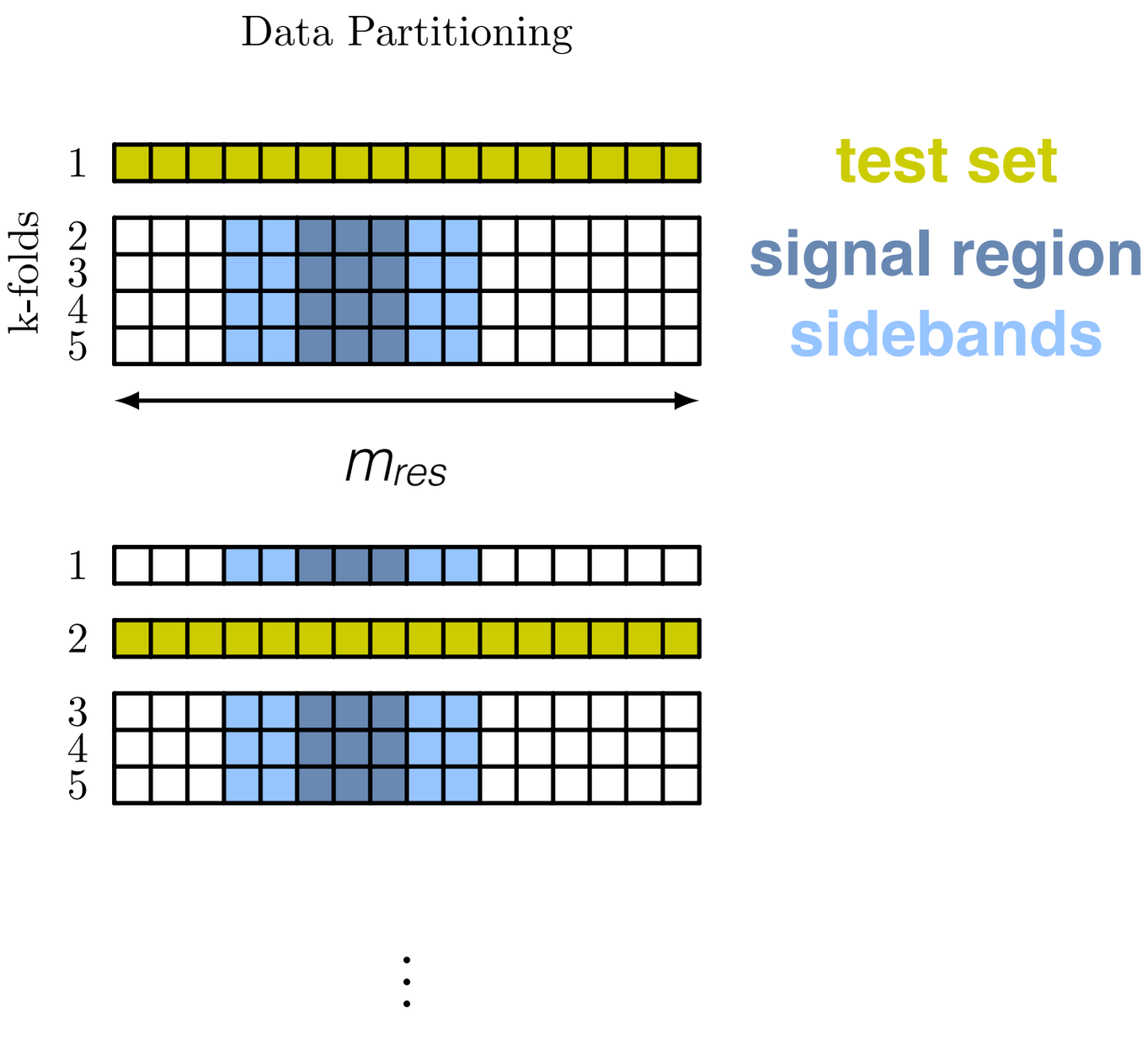
i.e. you will sculpt lots of bumps!



Solution: **(nested) cross-training**

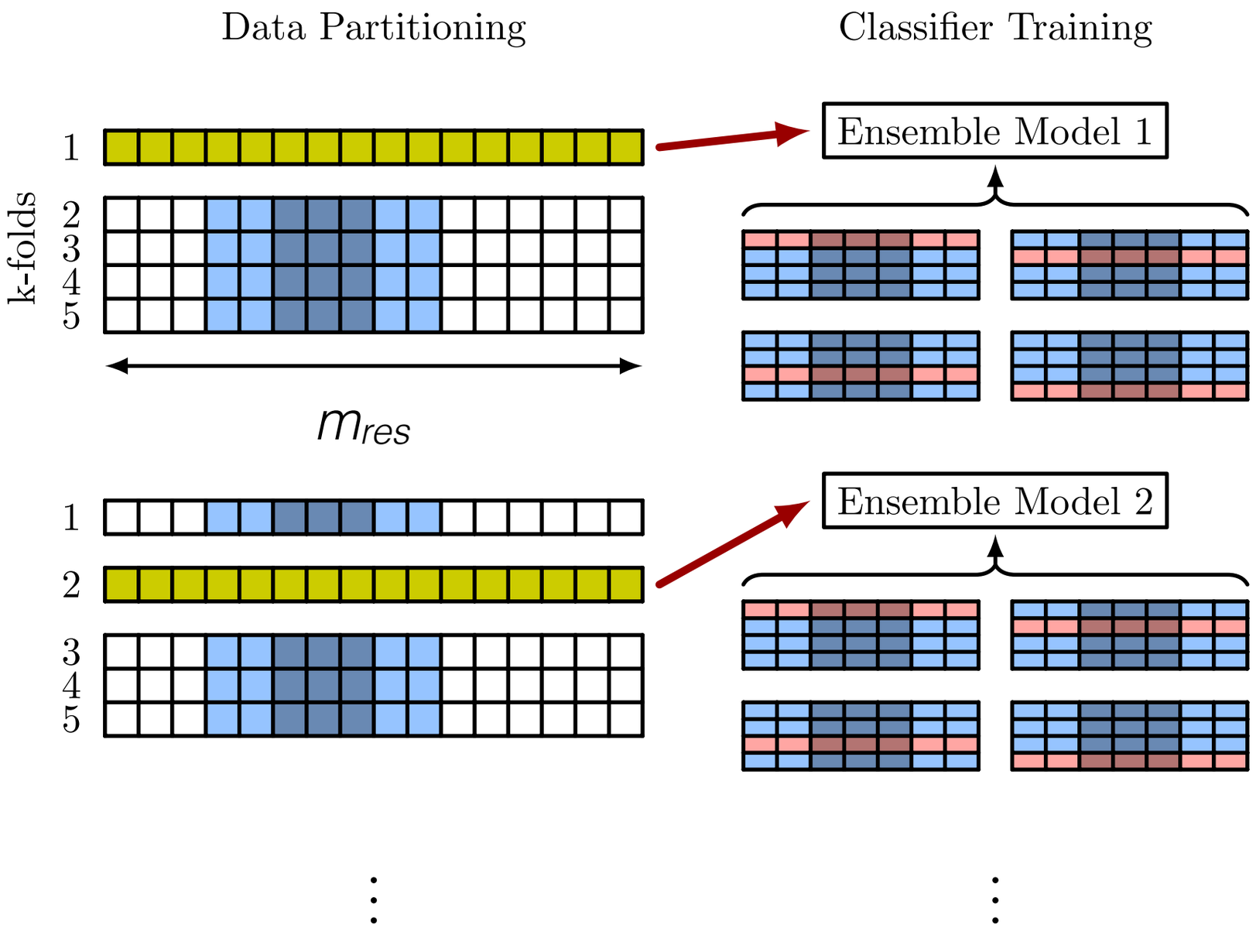
Nested cross-training

(1) Divide the entire dataset into k-folds.



Nested cross-training

(2) Train CWoLa classifiers.



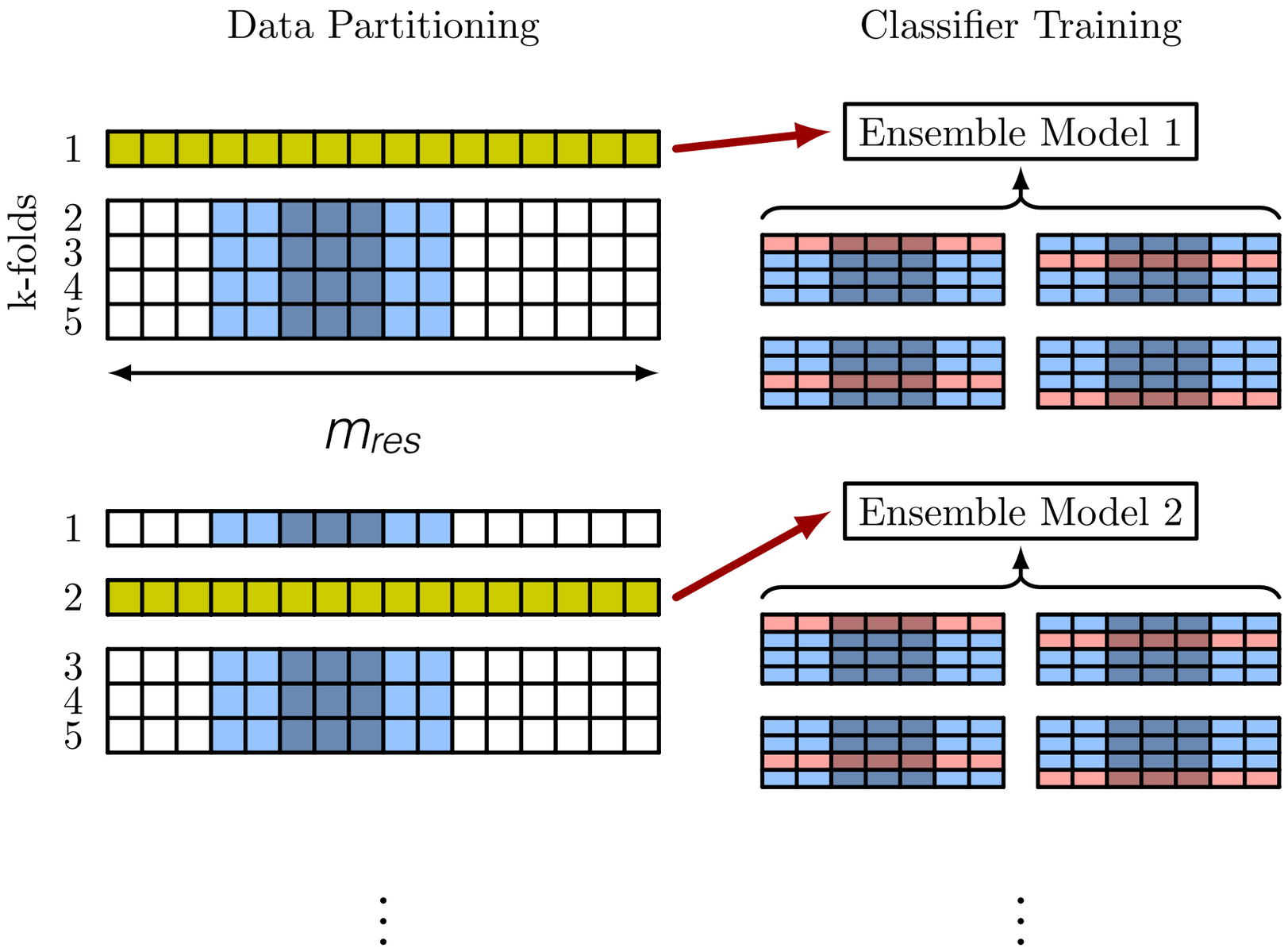
Train signal versus sideband k-1 times

rotating the validation set

In practice, also train many networks with a different initialization and take the best one per k-1 rotation.

Nested cross-training

(2) Train CWoLa classifiers.

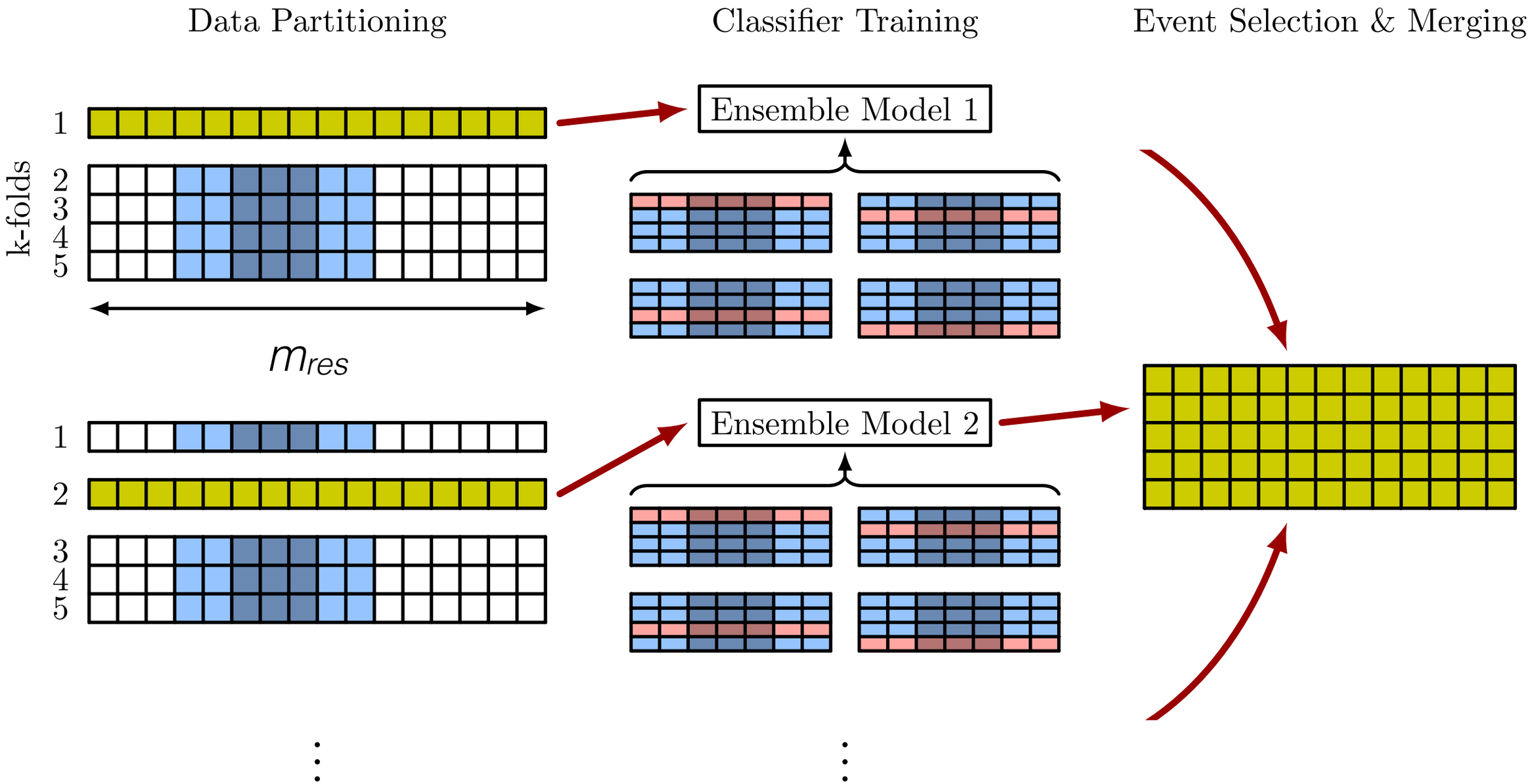


The Ensemble Model is just the average of the four networks.

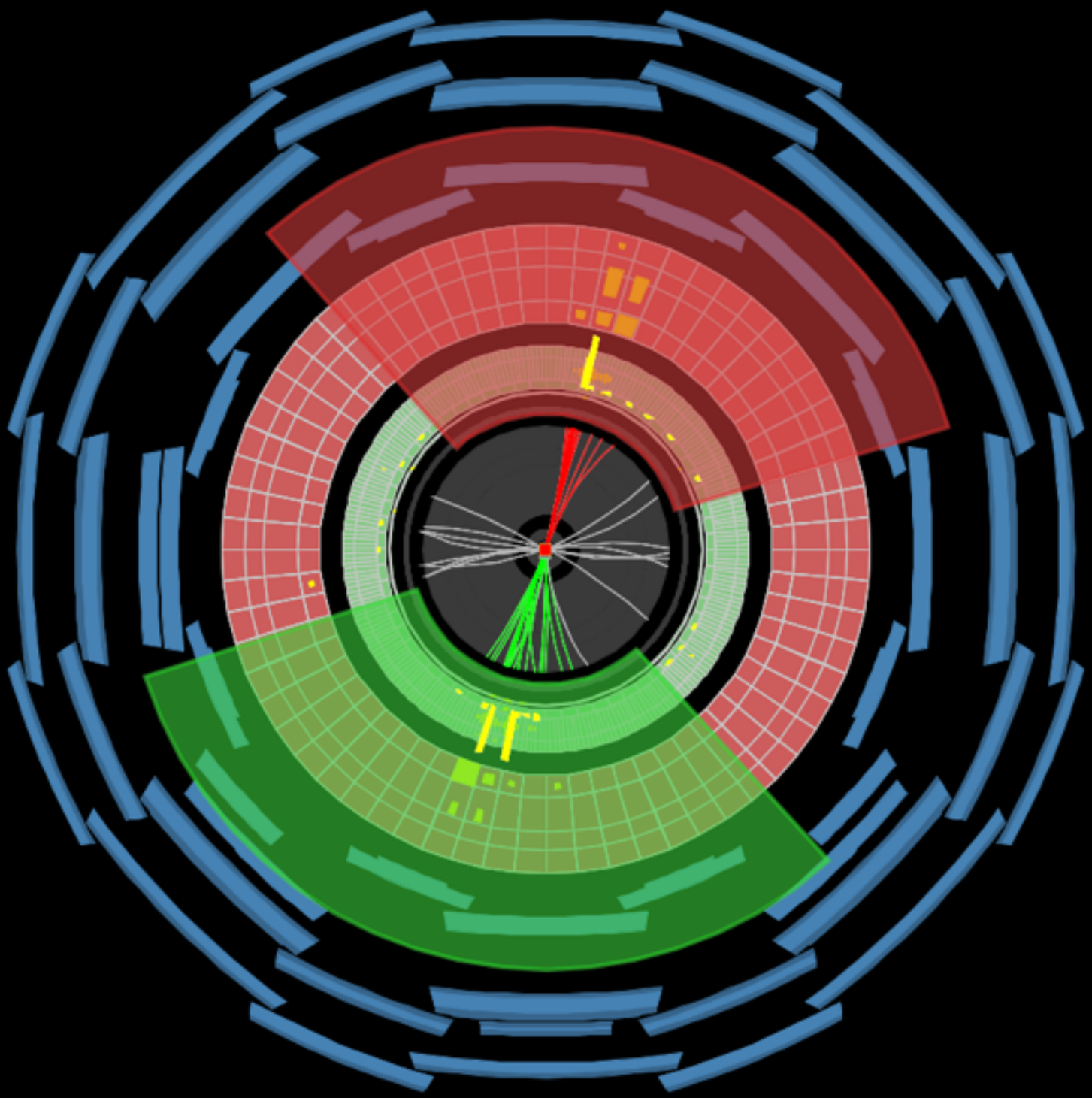
Data fluctuations will cancel destructively while signal interferes constructively.

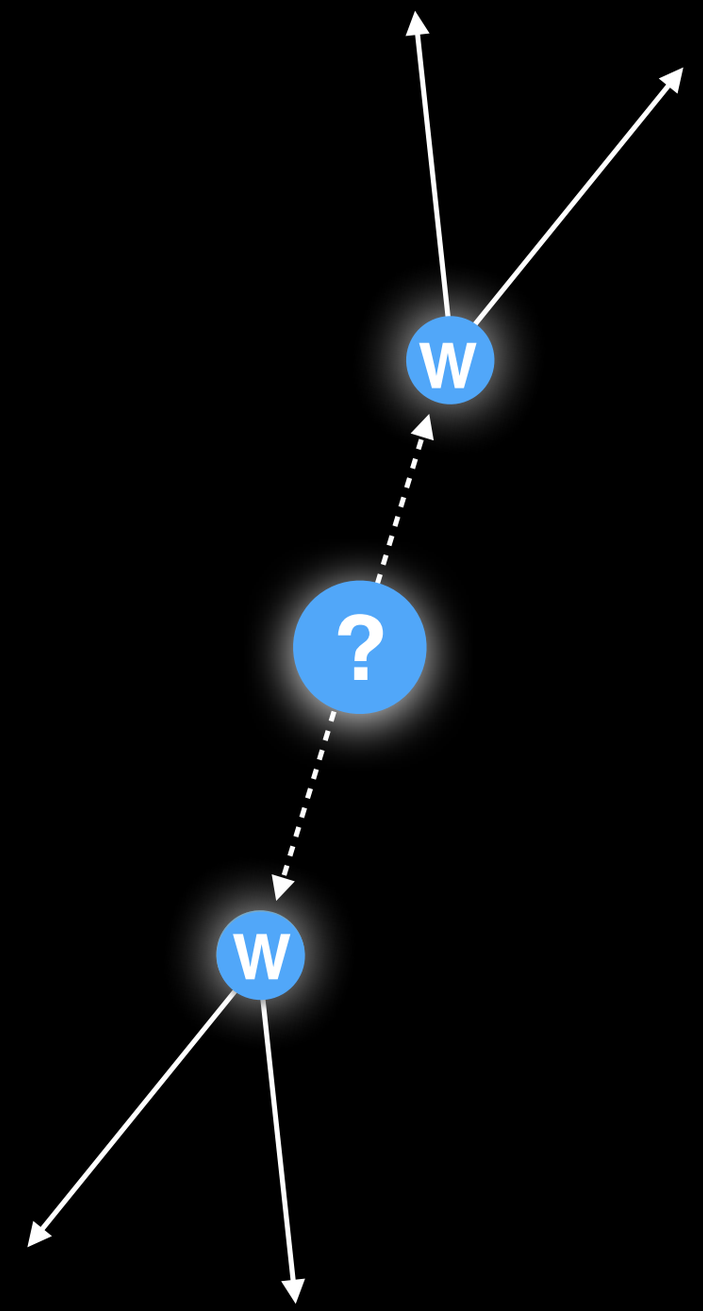
Nested cross-training

(3) Apply classifiers to holdout test sets and sum.



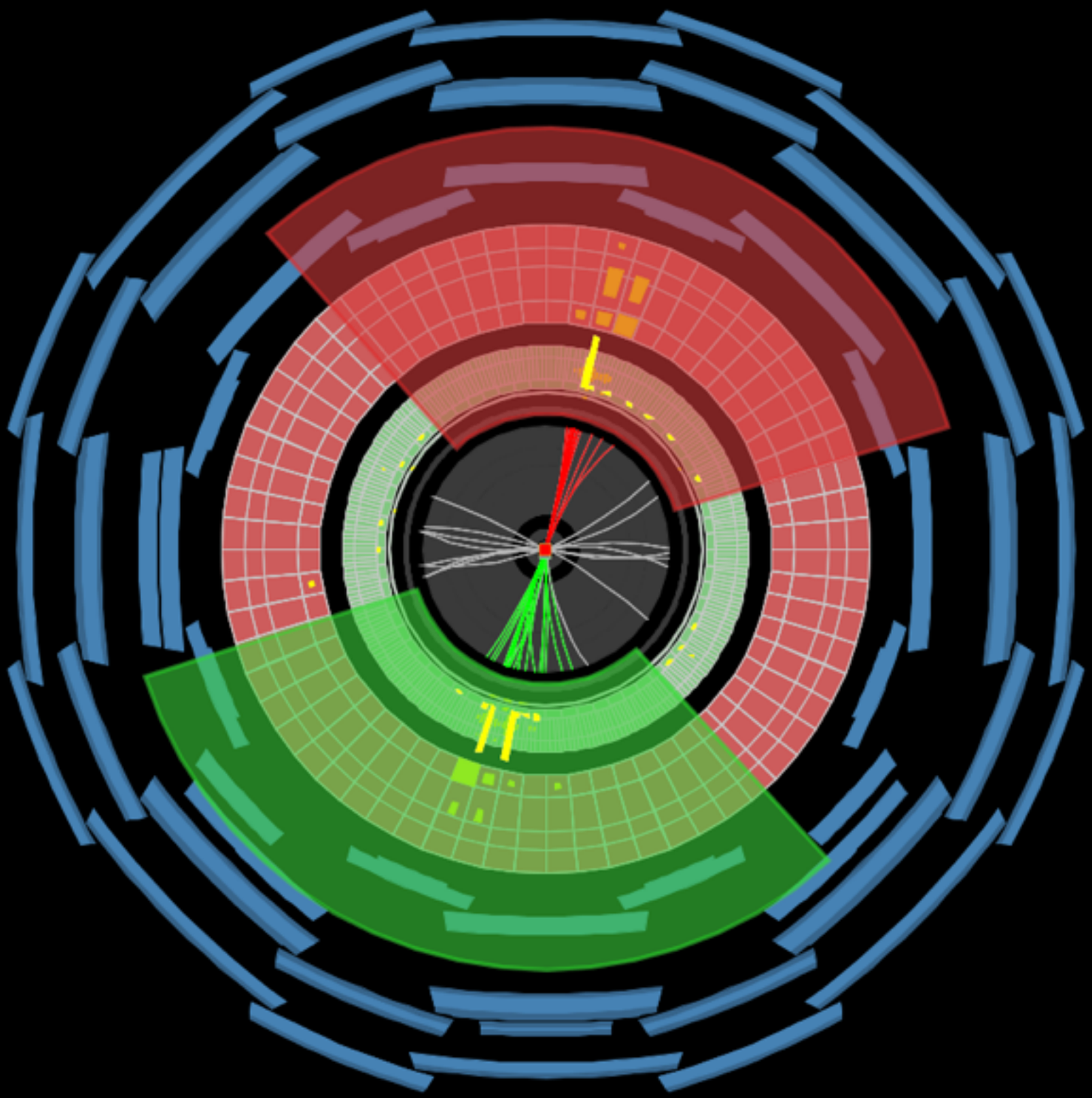
Application to dijet resonance search

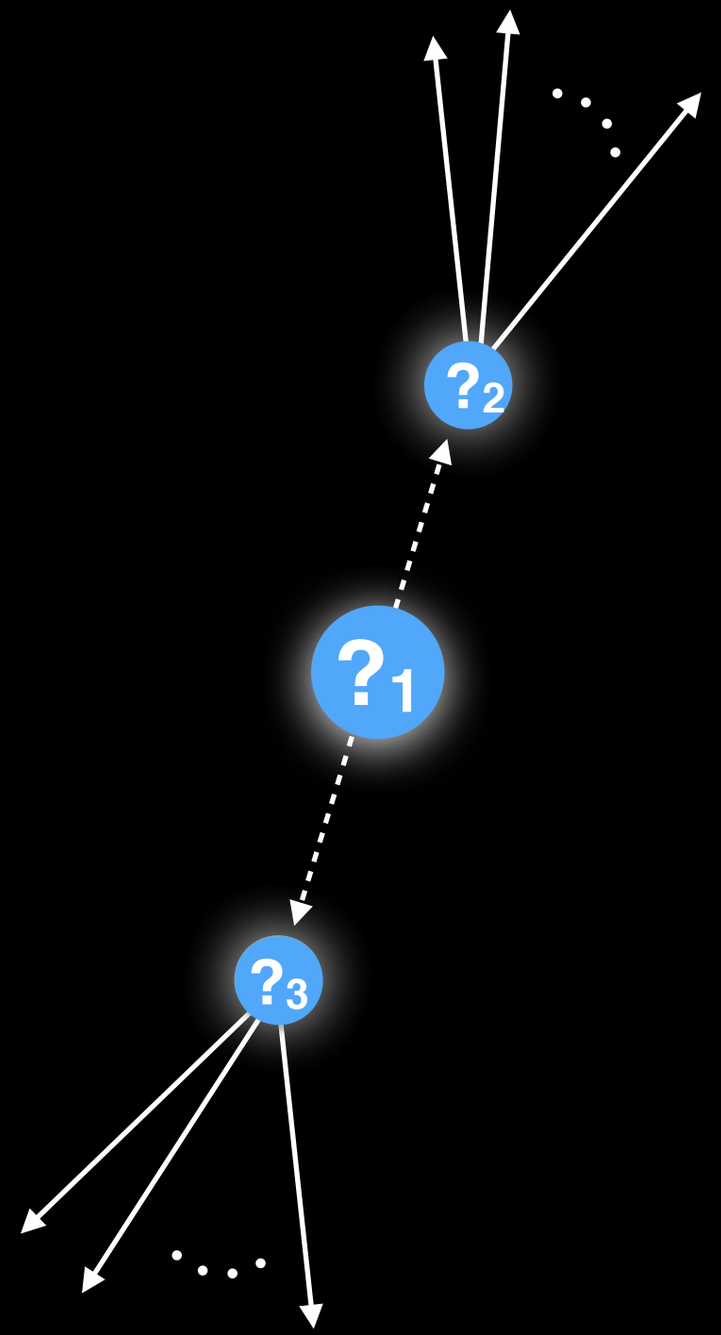




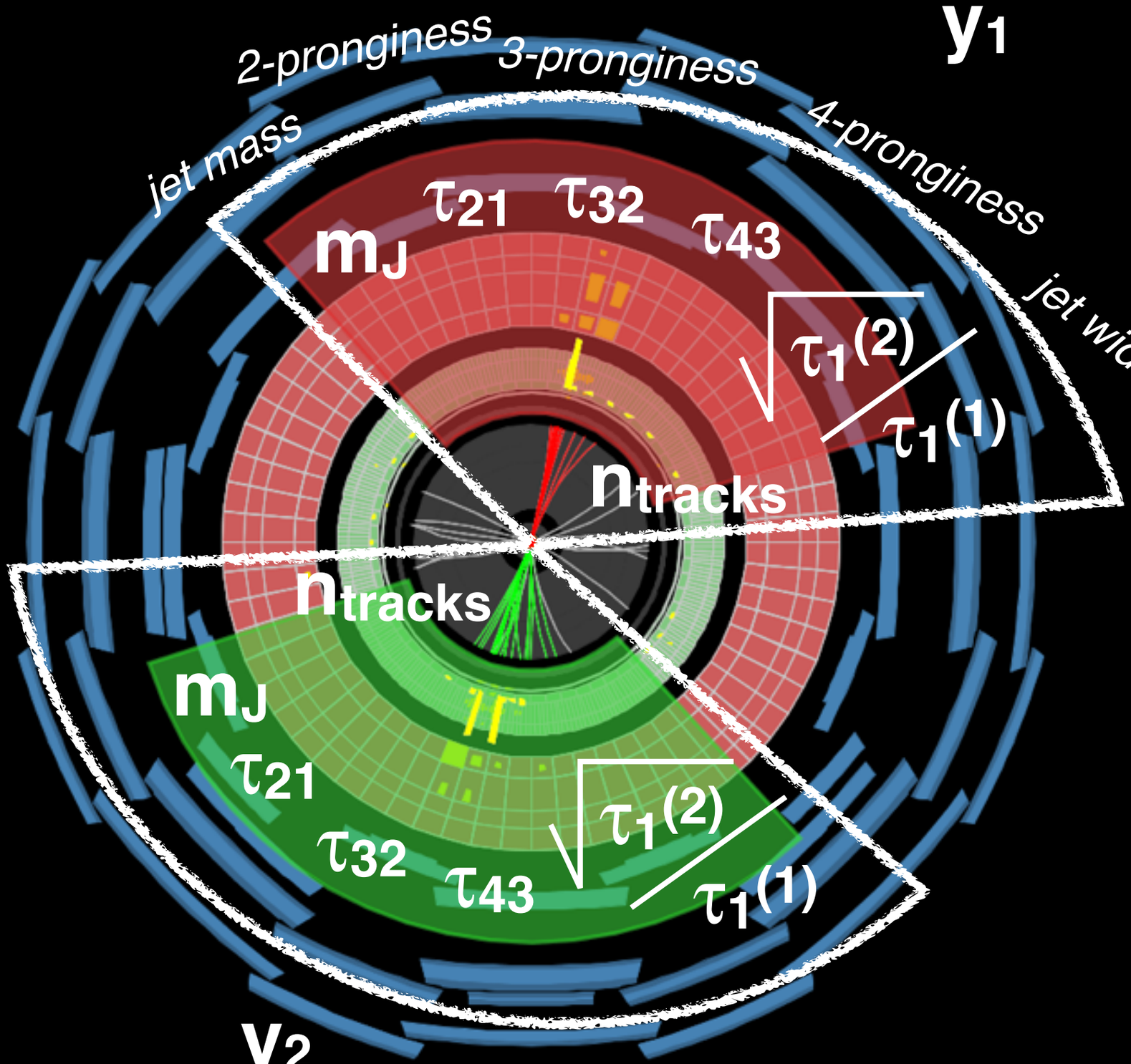
$m_? >> m_W$

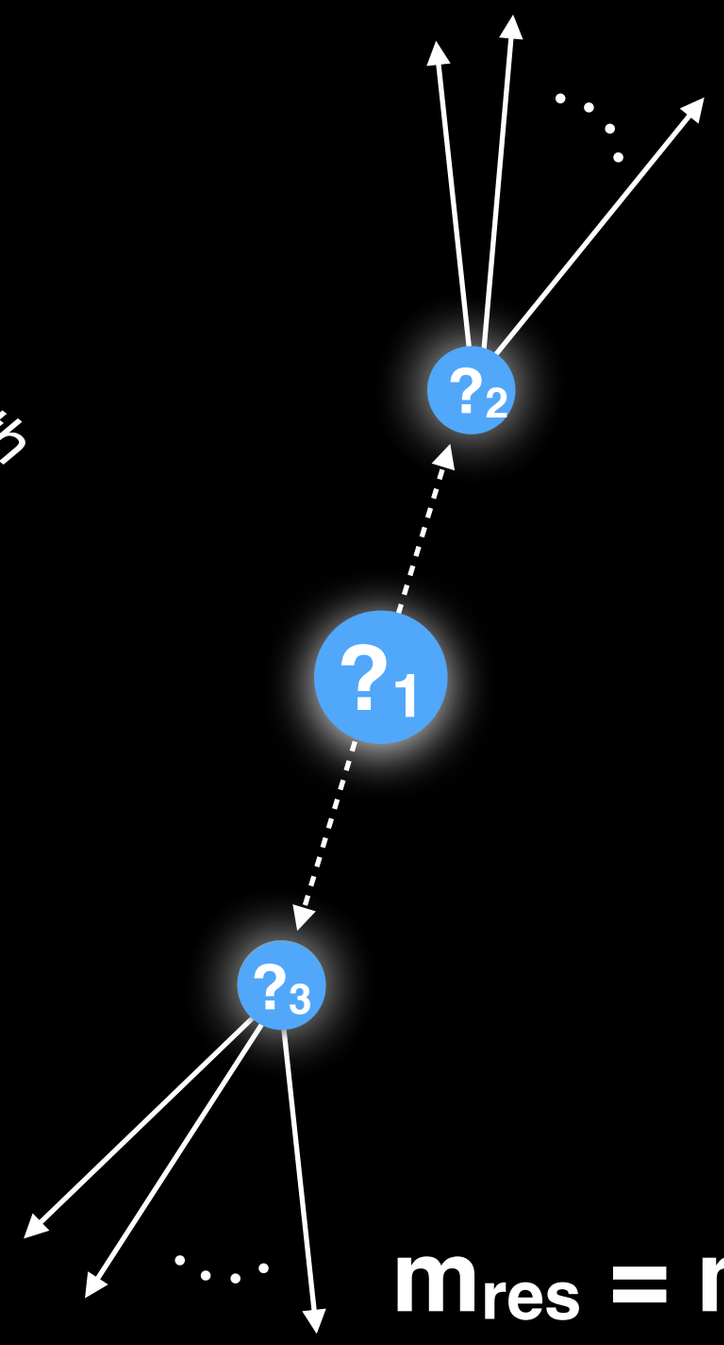
Application to dijet resonance search





Application to dijet resonance search





Note about correlations

Why? $\sqrt{\tau_1^{(2)}}/\tau_1^{(1)}$

Remember: if NN can learn m_{JJ}, violate CWoLa assumptions.

$$m_{JJ} \sim p_{T,J_1} p_{T,J_2} \Delta R_{JJ}^2$$ **=> Can't learn jet p_T.**

$$\tau_1^{(j)} \sim \frac{1}{p_{T,\text{jet}}} \sum_{i \in \text{jet}} p_{T,i} \Delta R(\text{jet}, i)^j$$ **$\Delta R \sim 1/\gamma \sim m/p_T$**

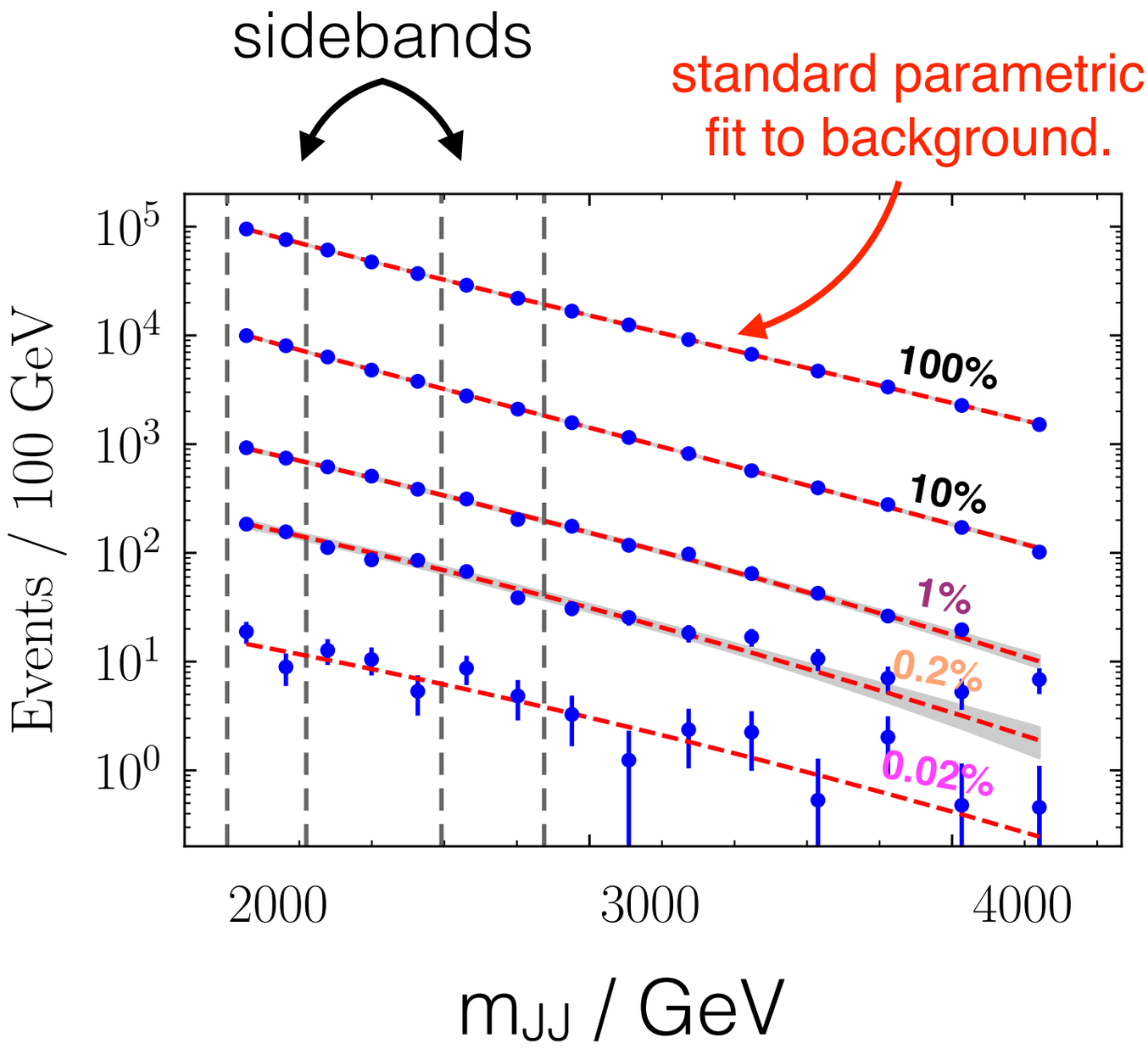
=> can't have both m and $\tau_1^{(2)}$

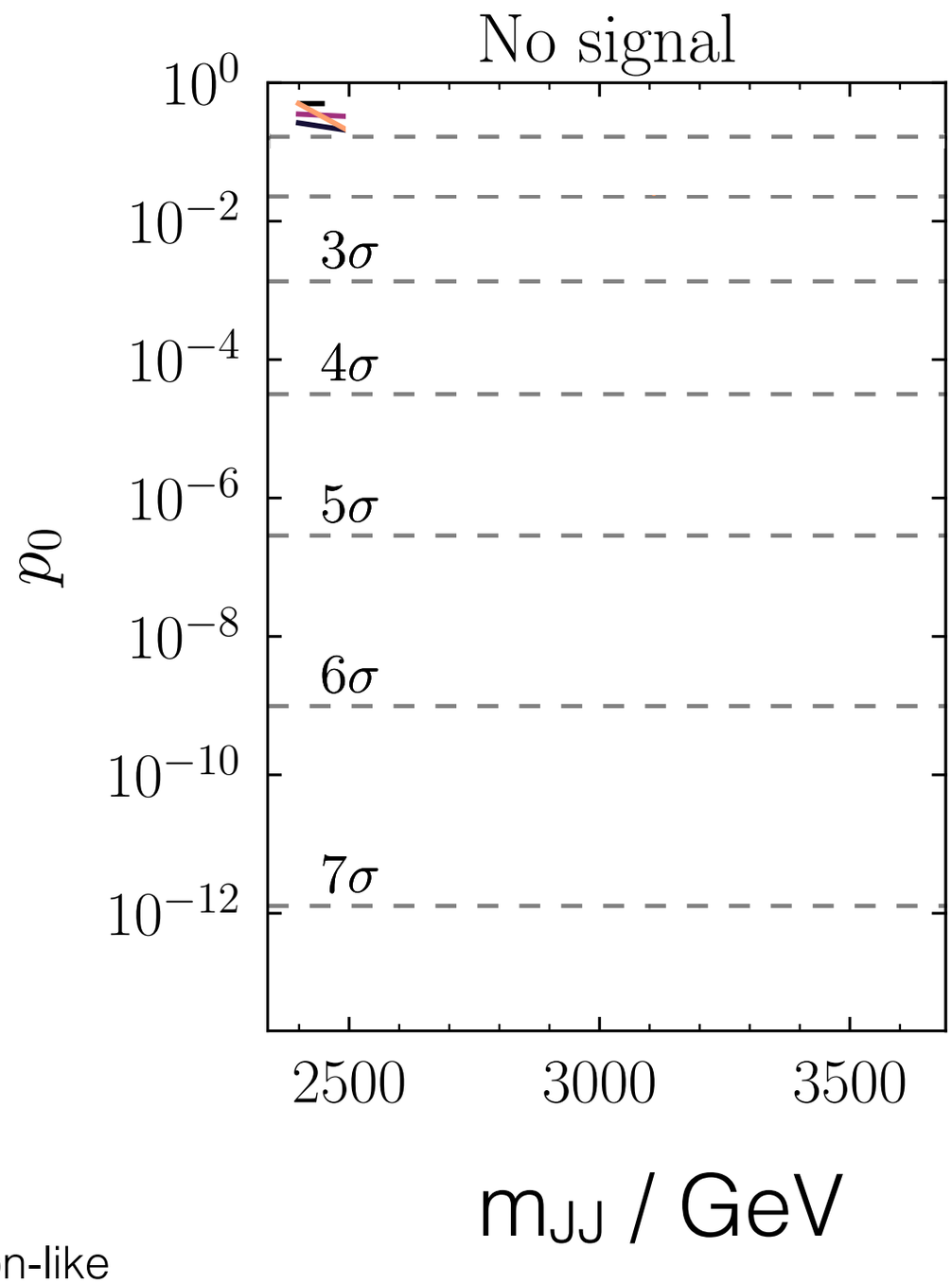
=> since $\tau_1^{(2)} \sim 1/\gamma^2$ and $\tau_1^{(1)} \sim 1/\gamma$, use $\sqrt{\tau_1^{(2)}} / \tau_1^{(1)}$

Could also imagine doing this automatically with ML - see

C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson et al. 1703.03507.

Application to dijets





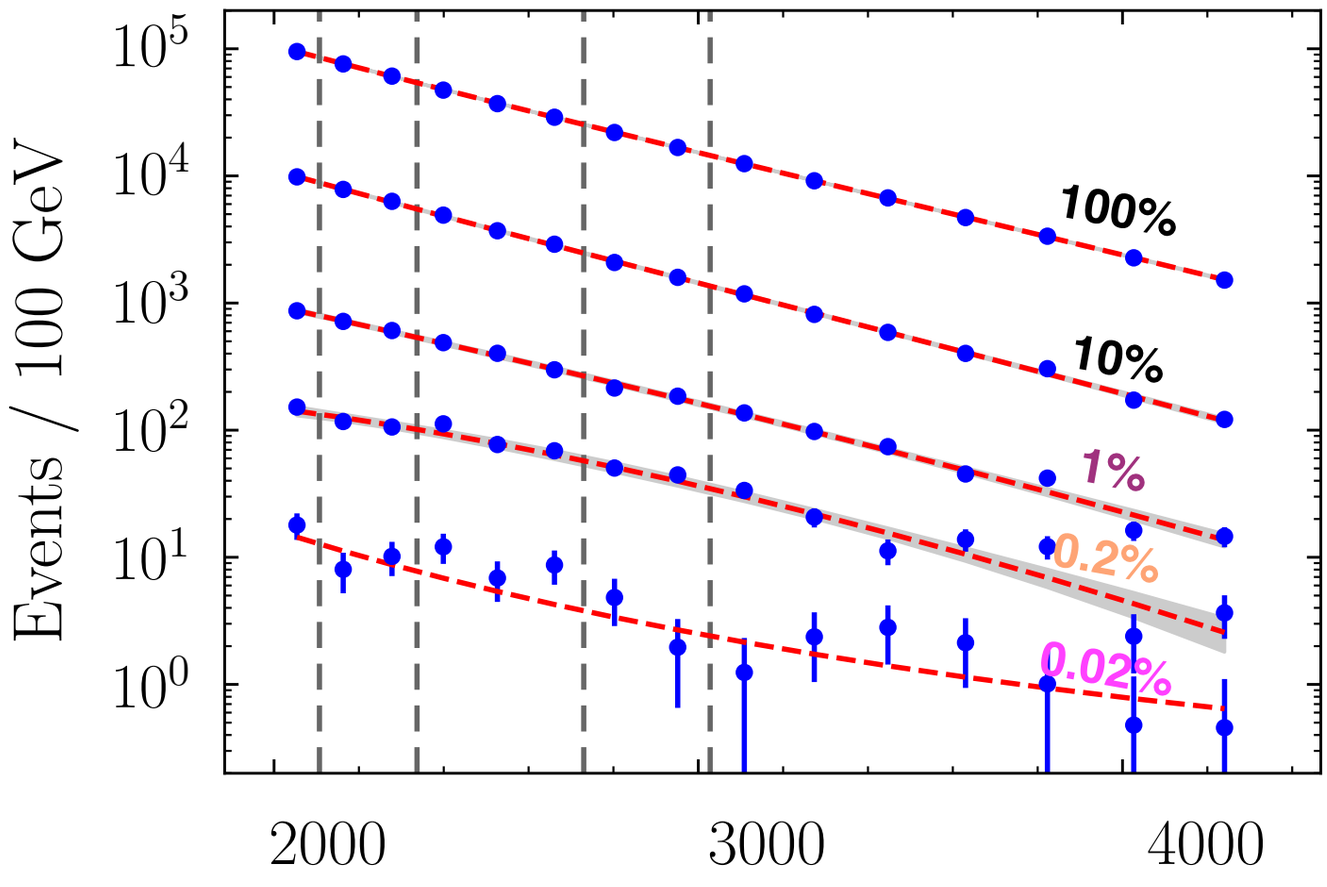
no cut on NN

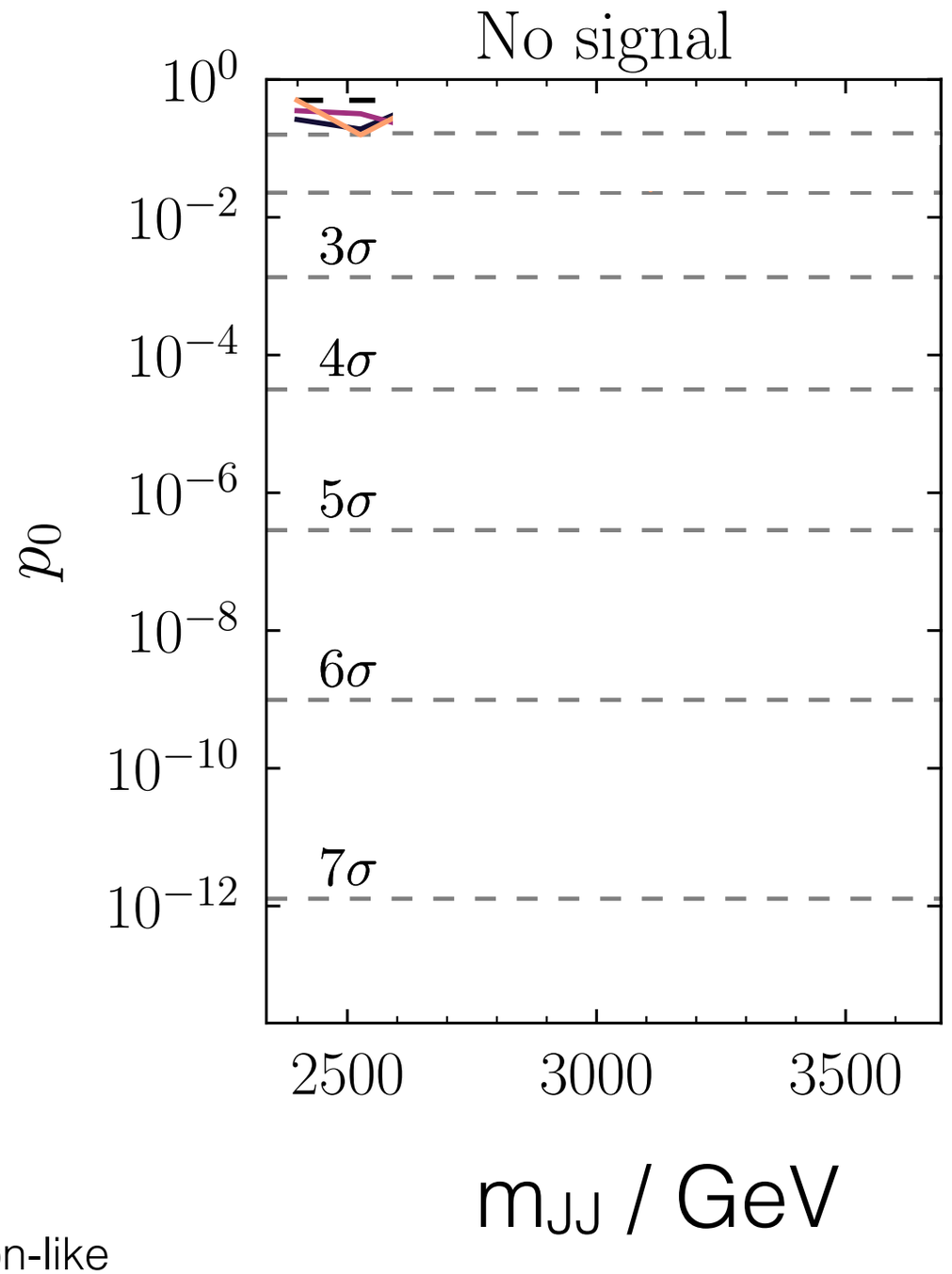
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





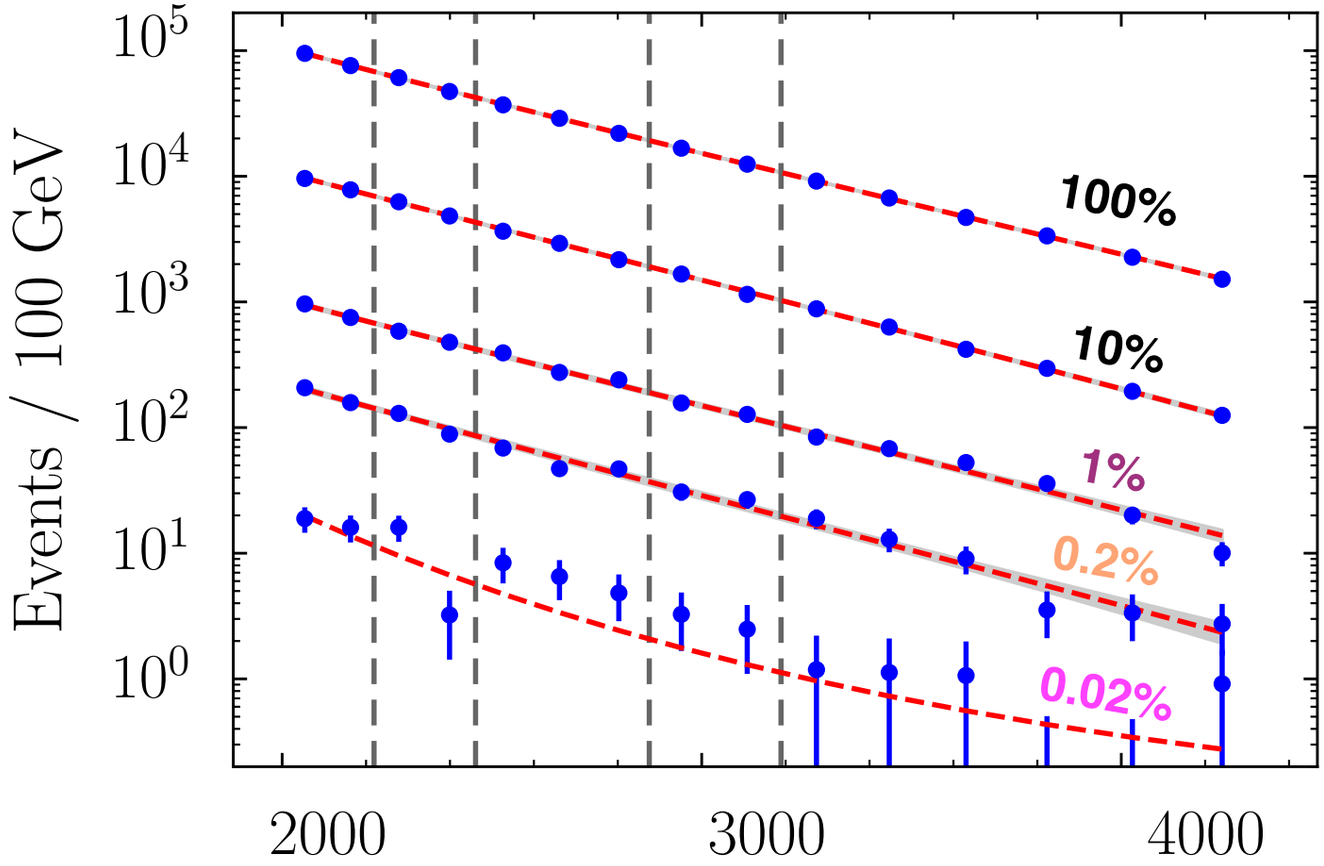
no cut on NN

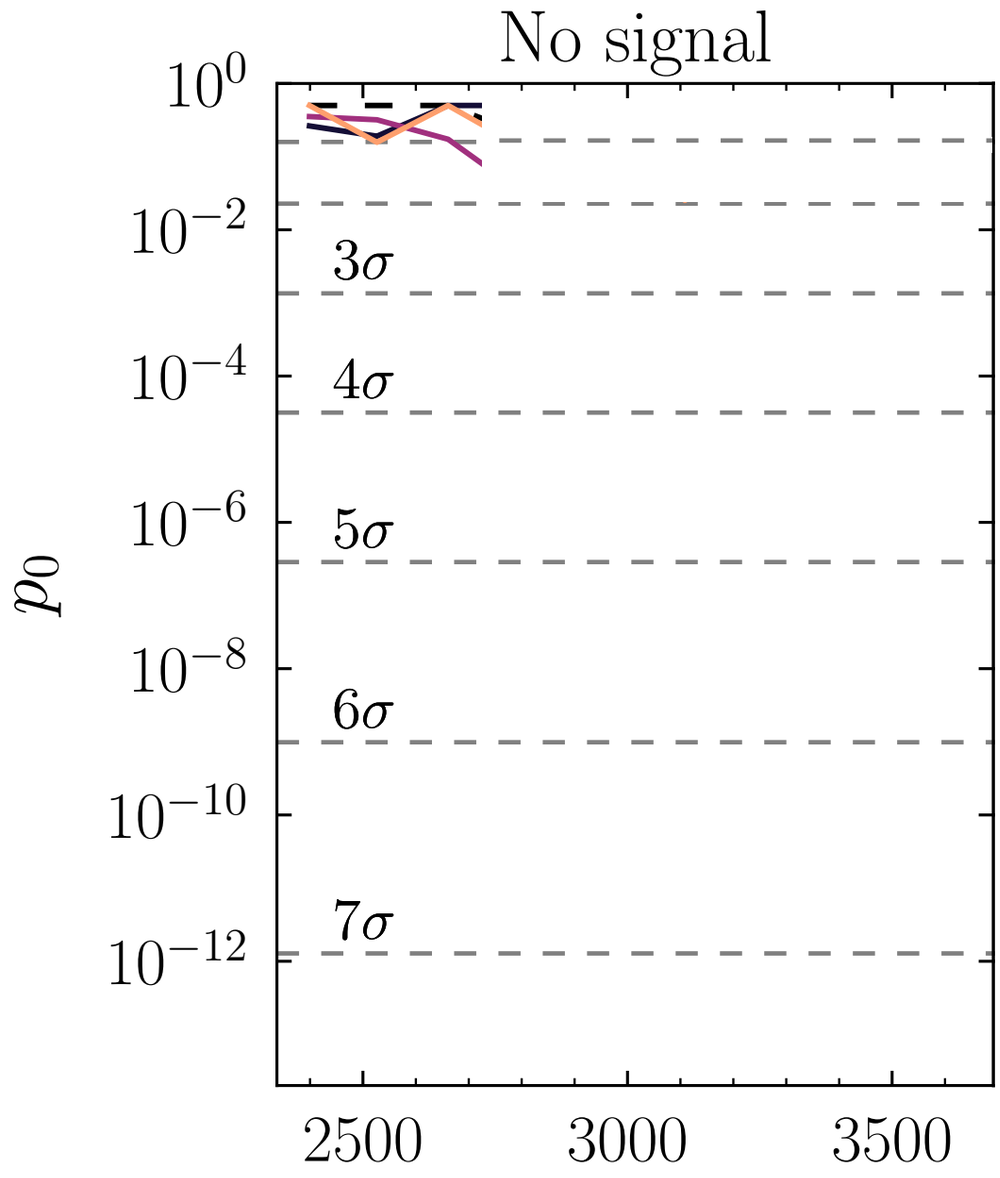
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





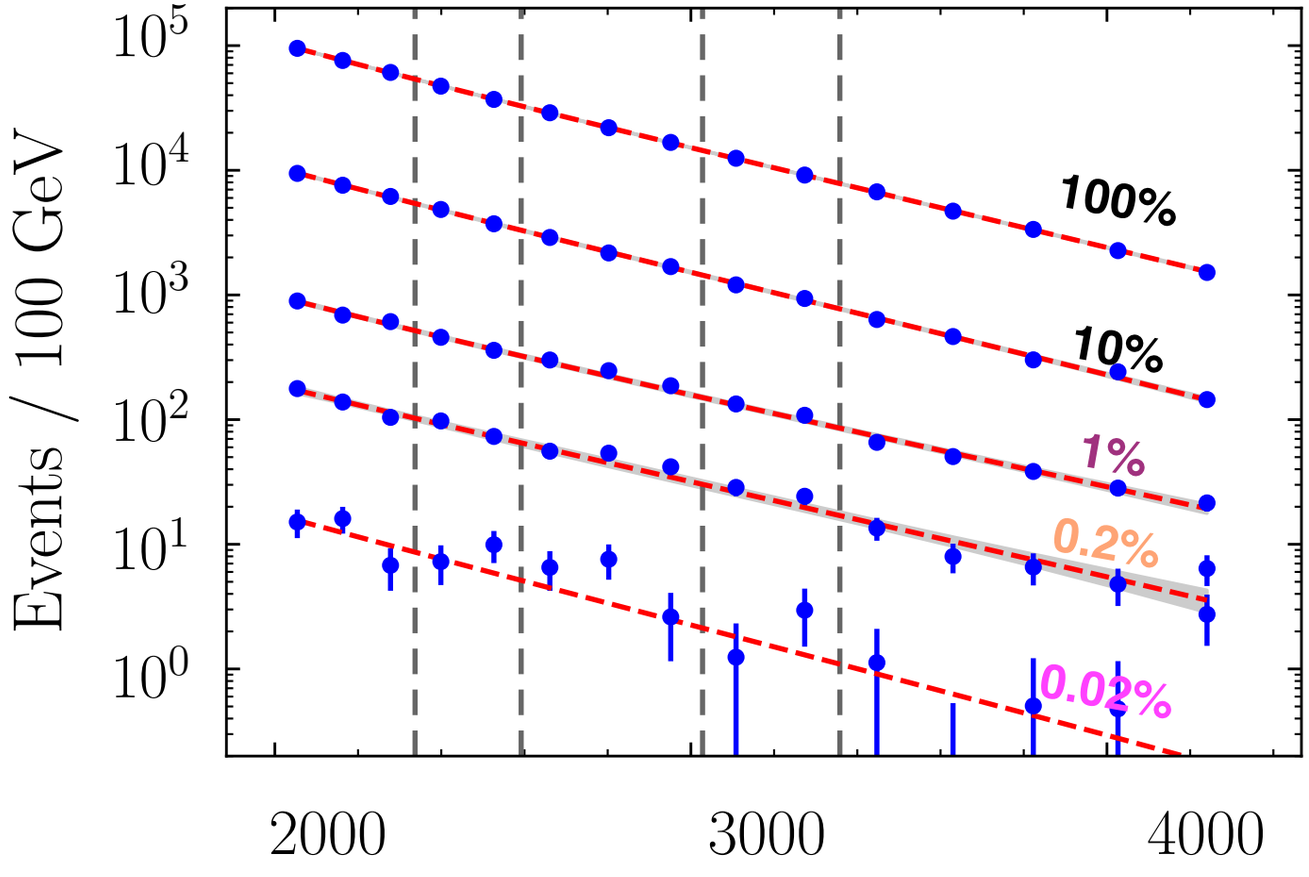
no cut on NN

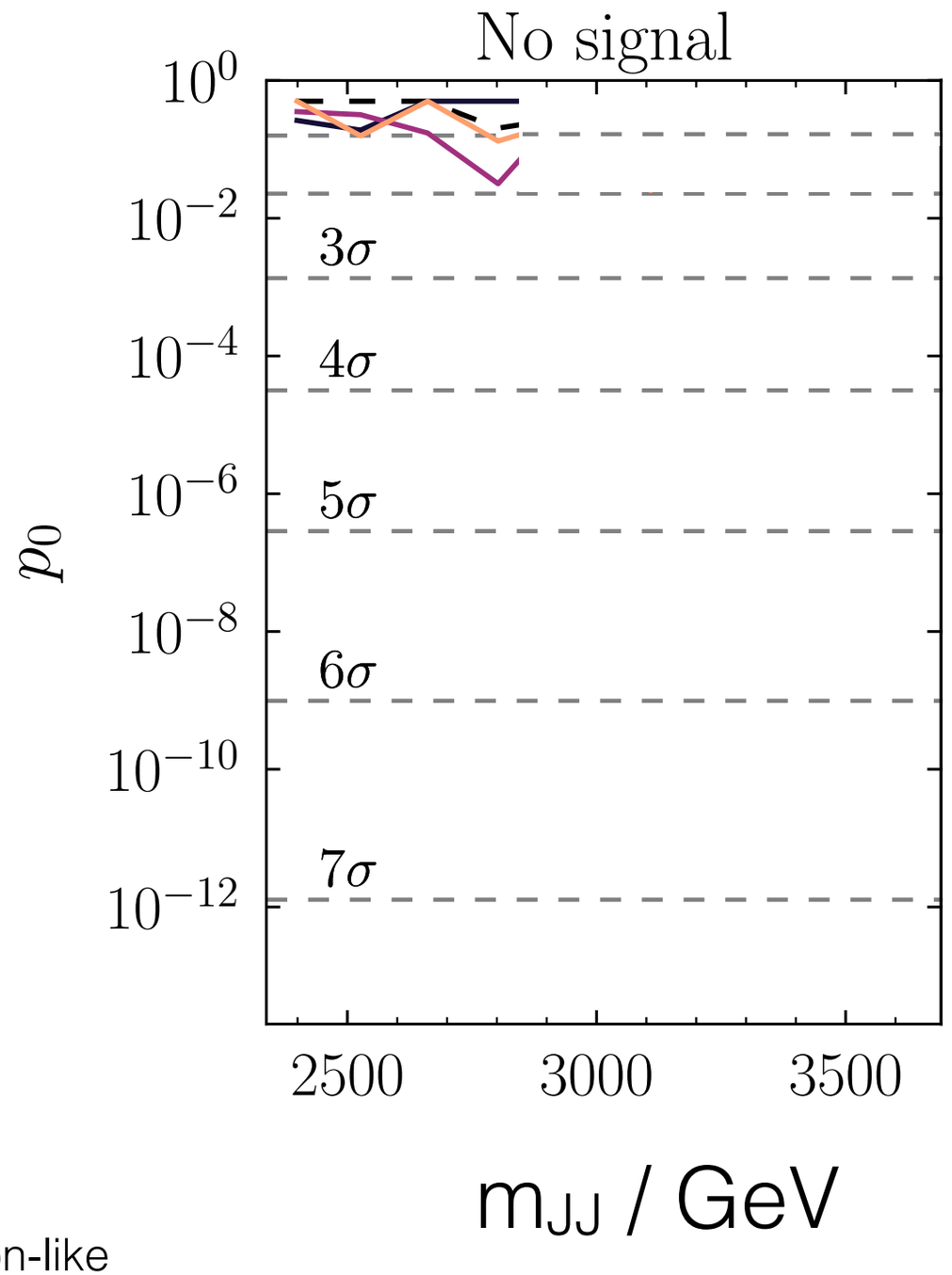
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





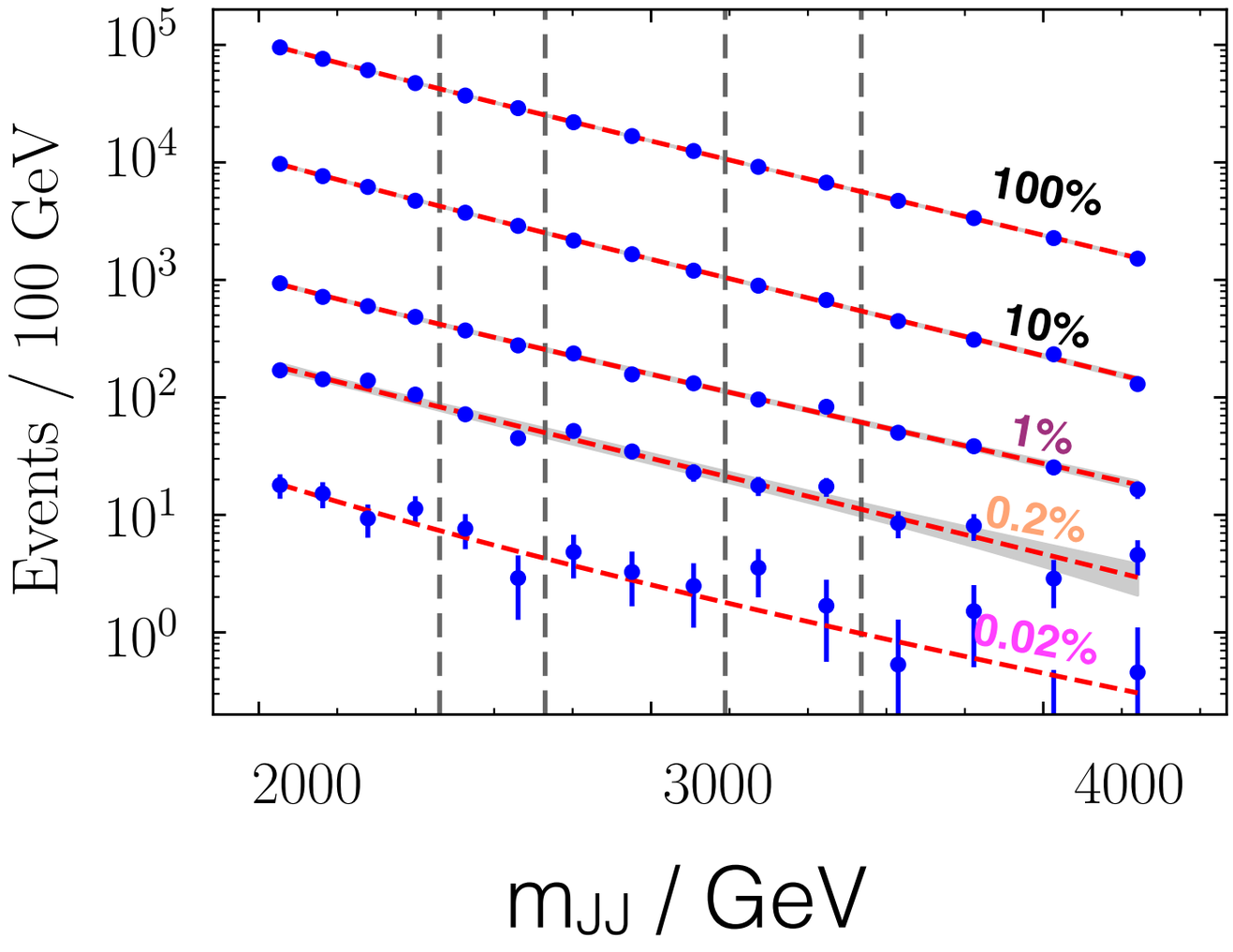
no cut on NN

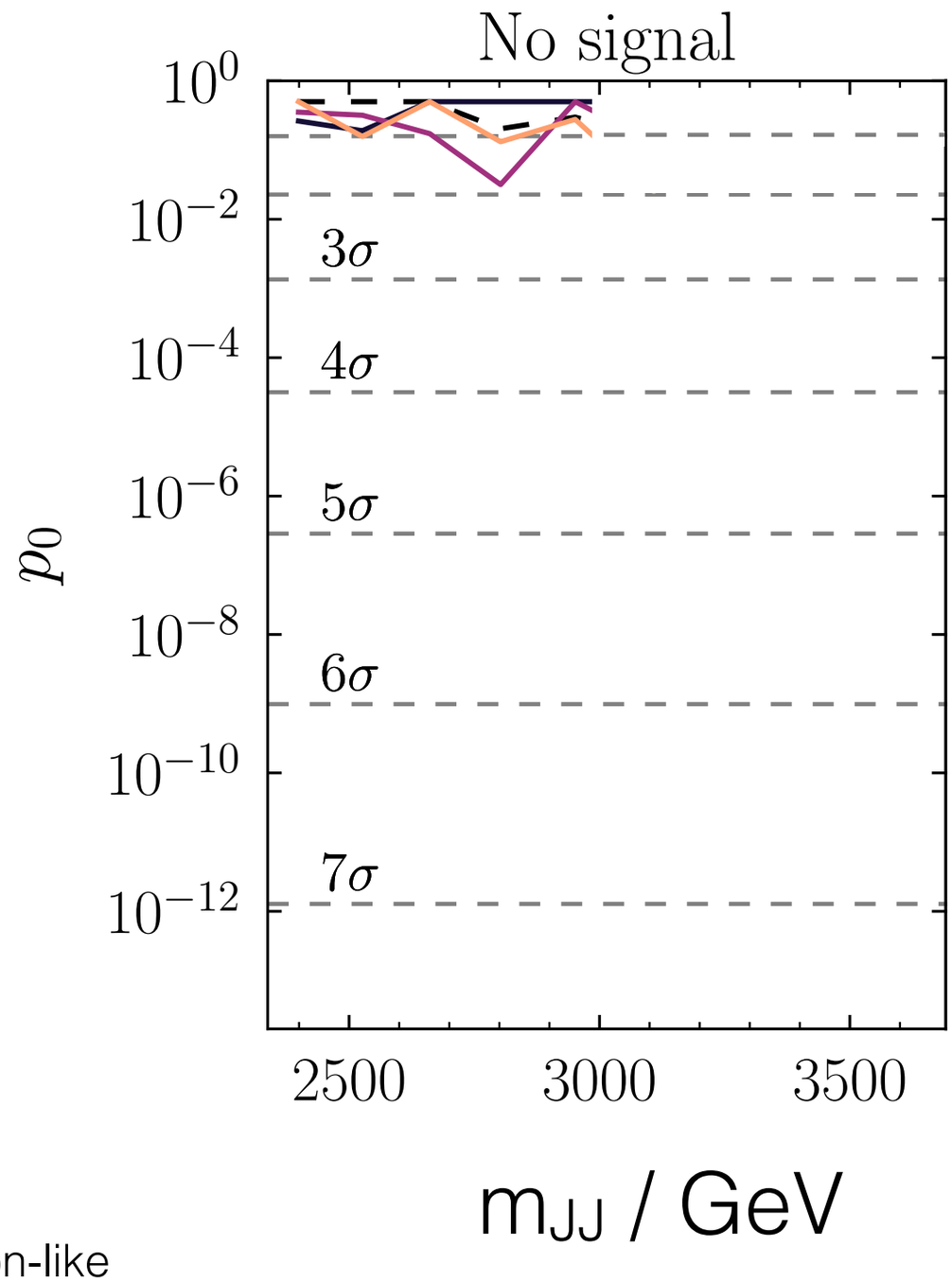
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





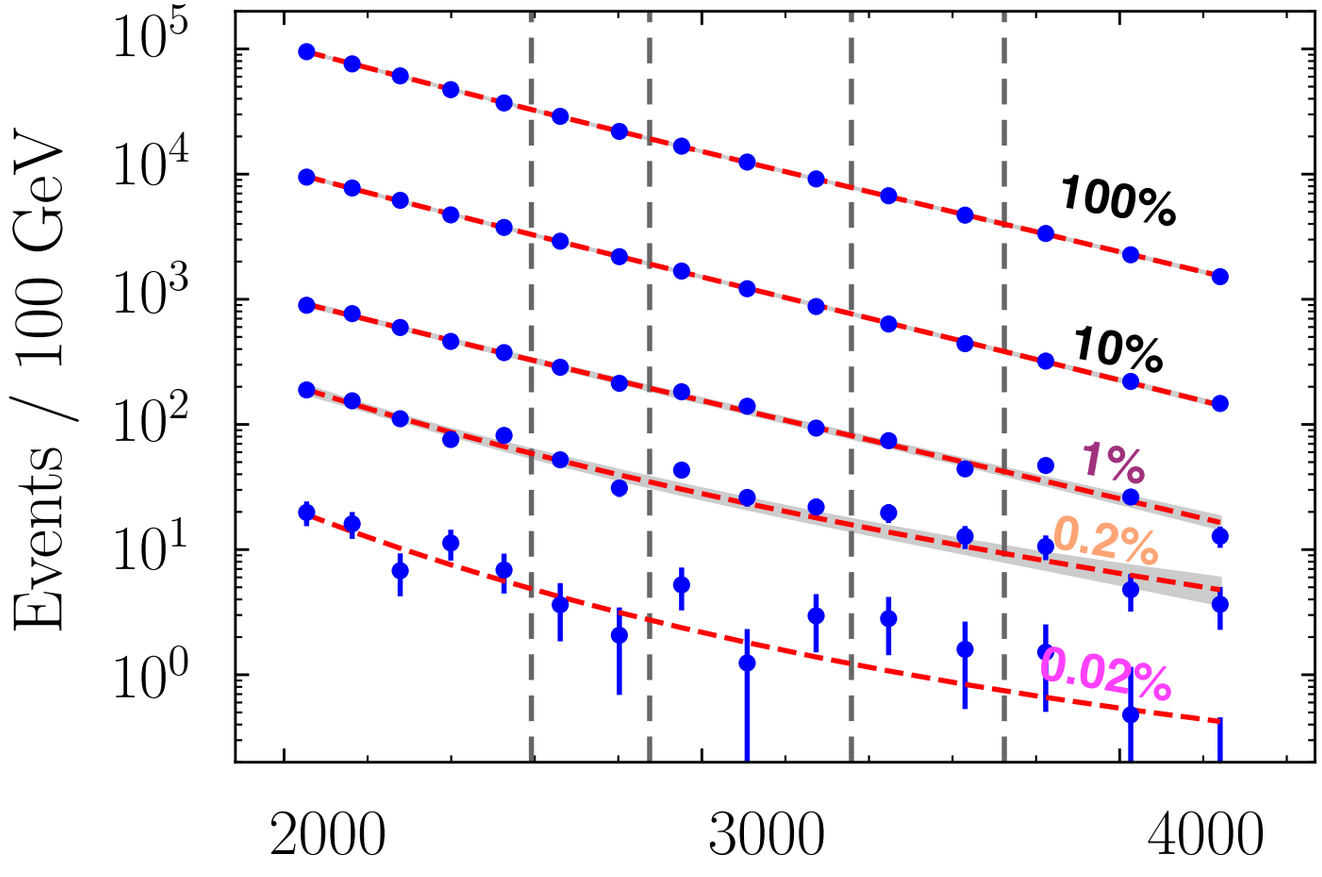
no cut on NN

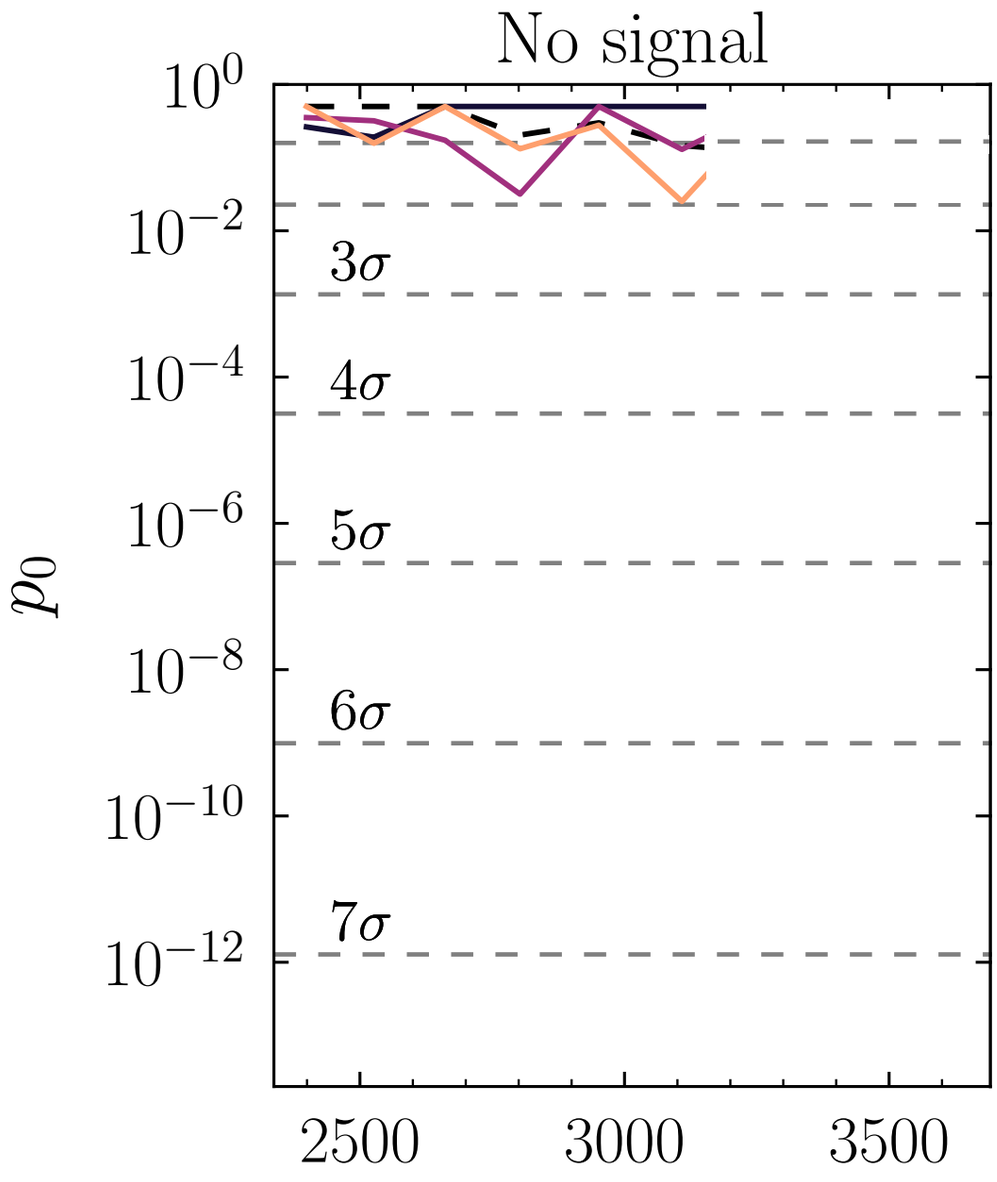
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





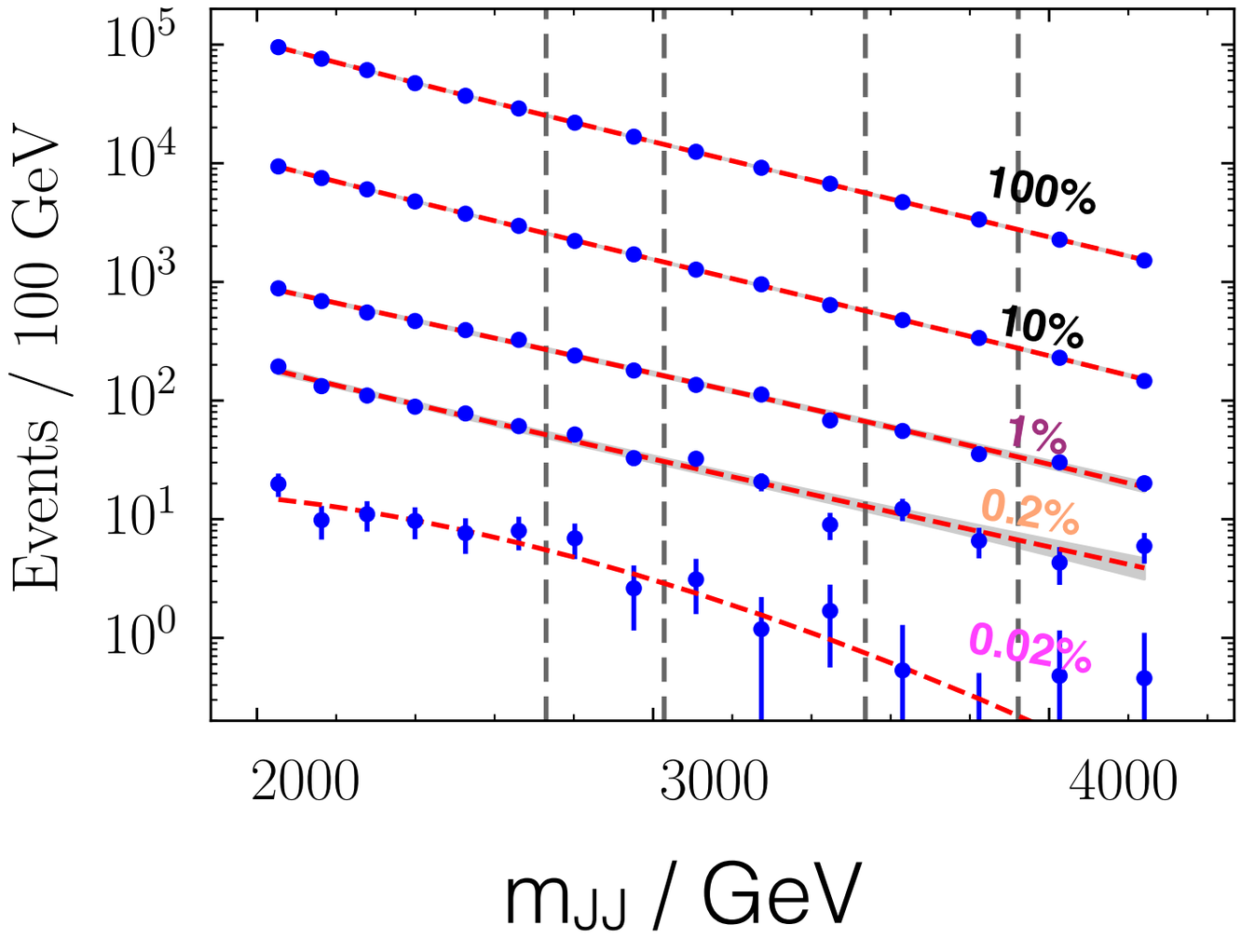
no cut on NN

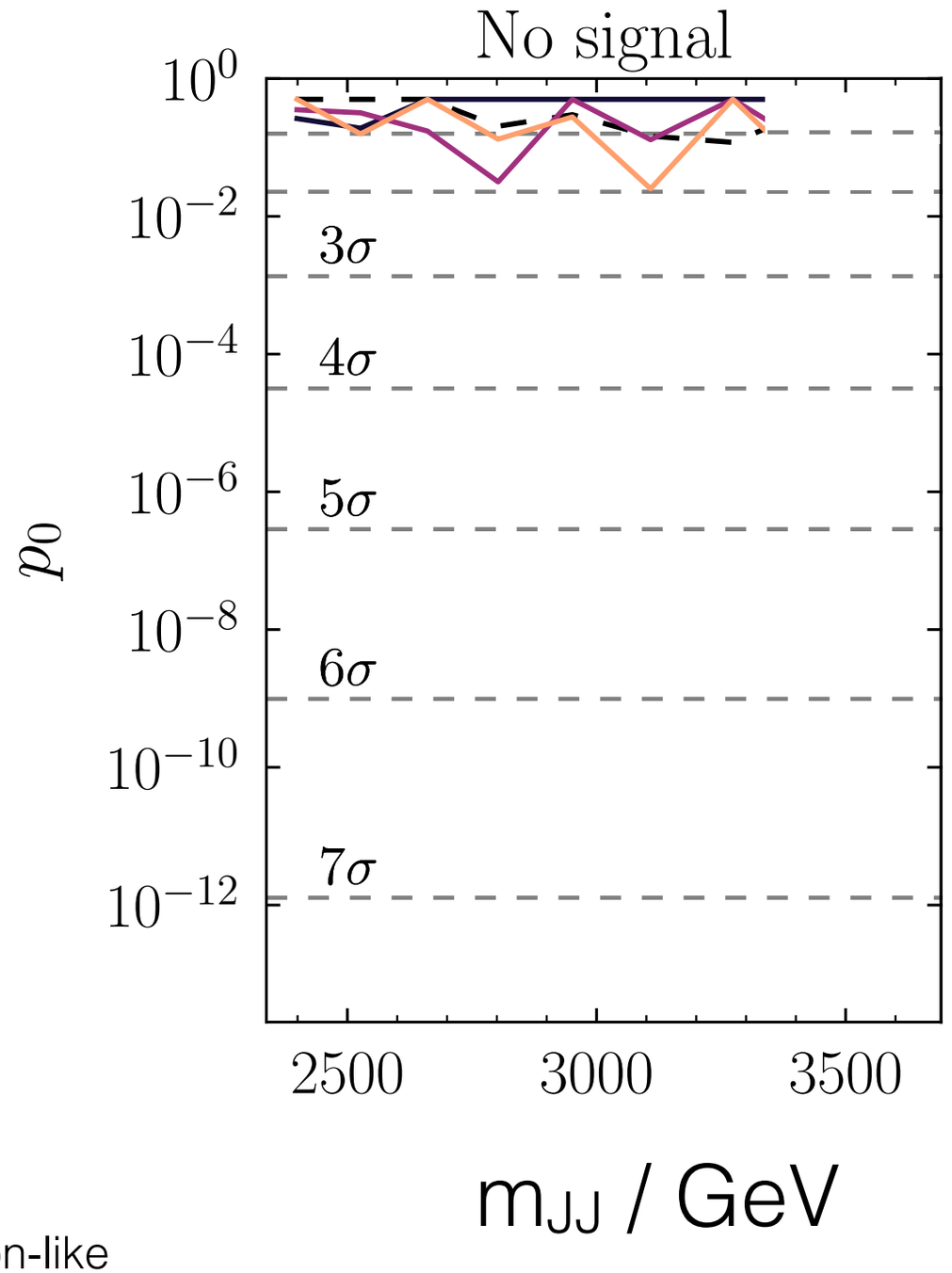
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





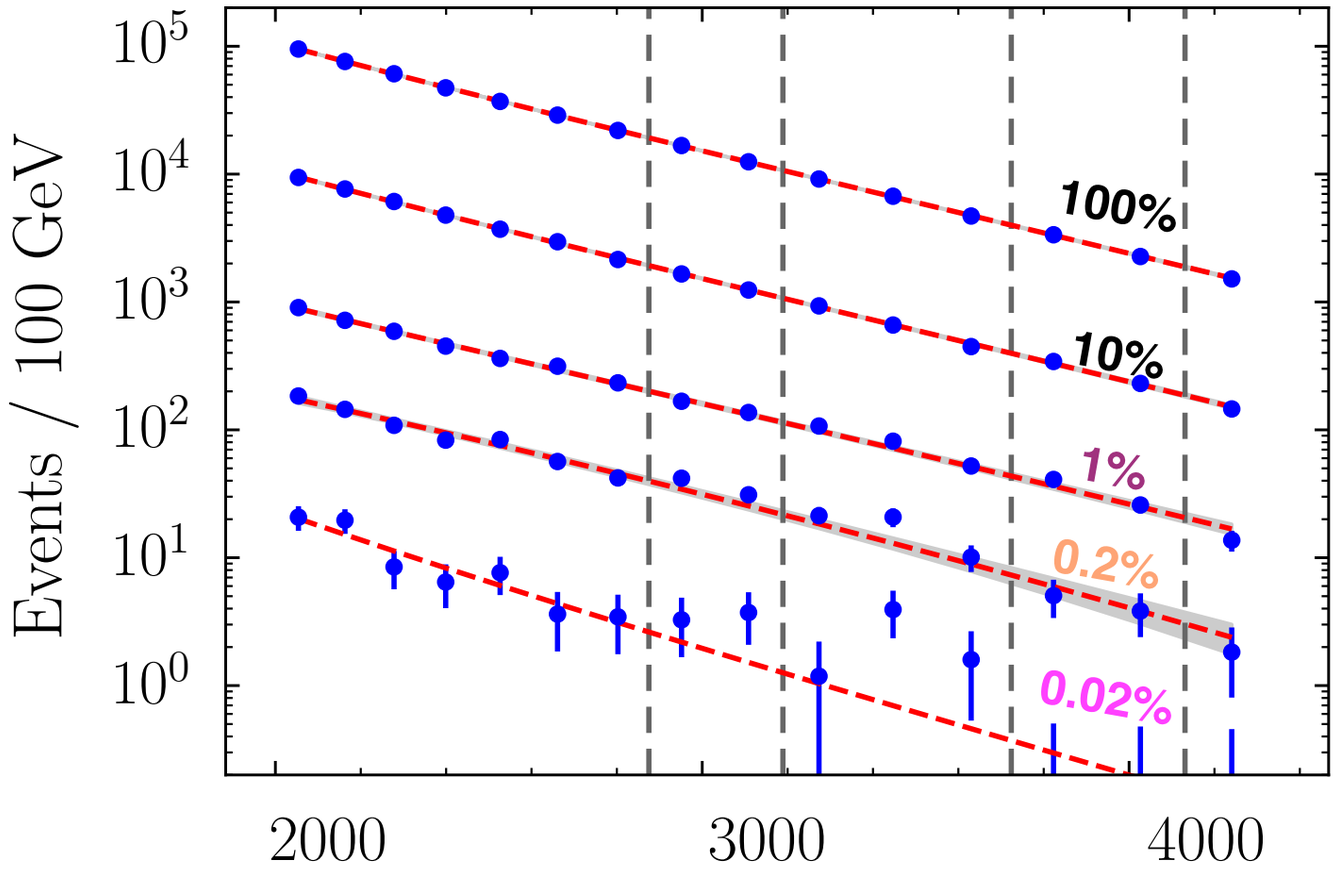
no cut on NN

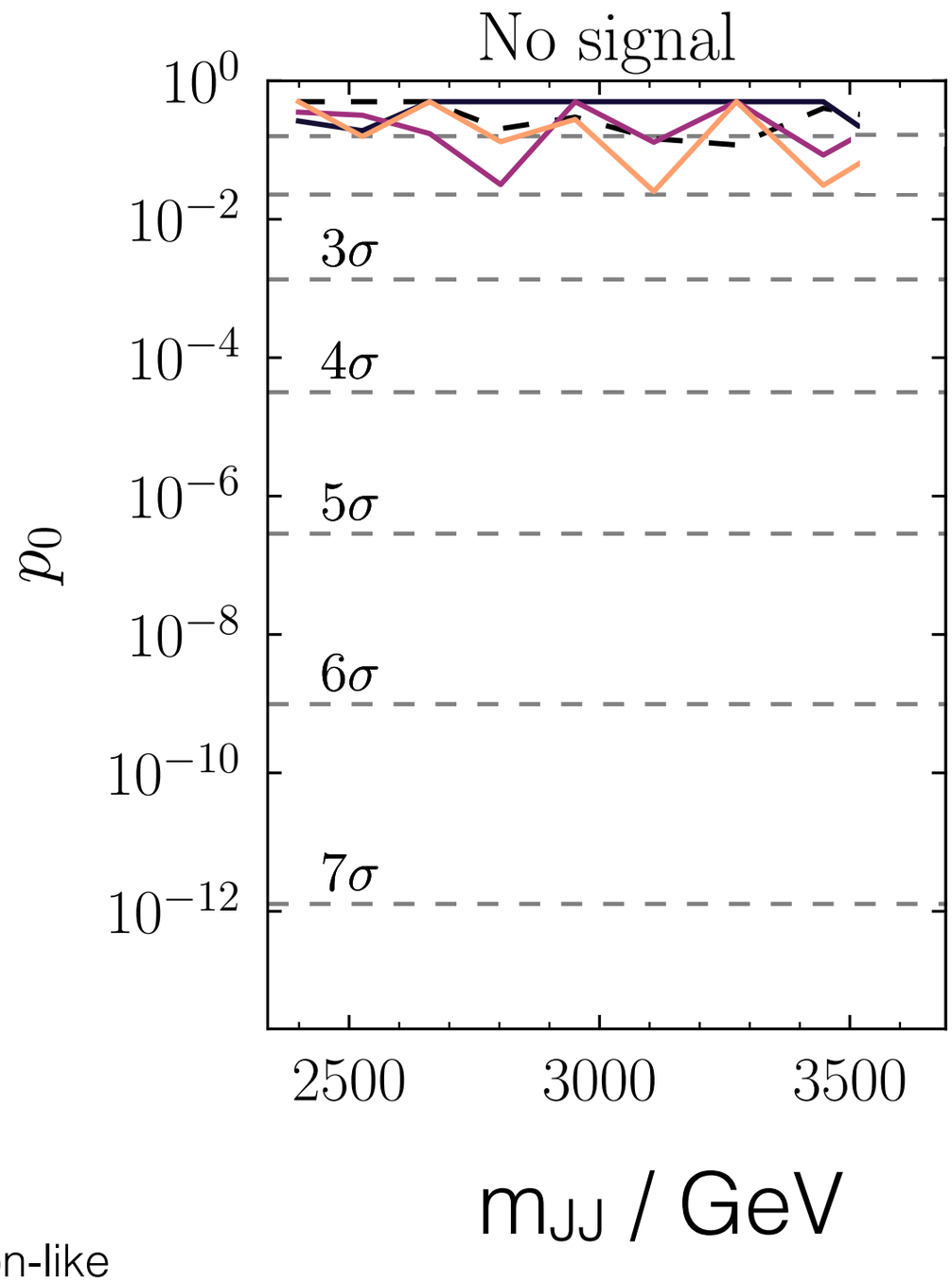
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





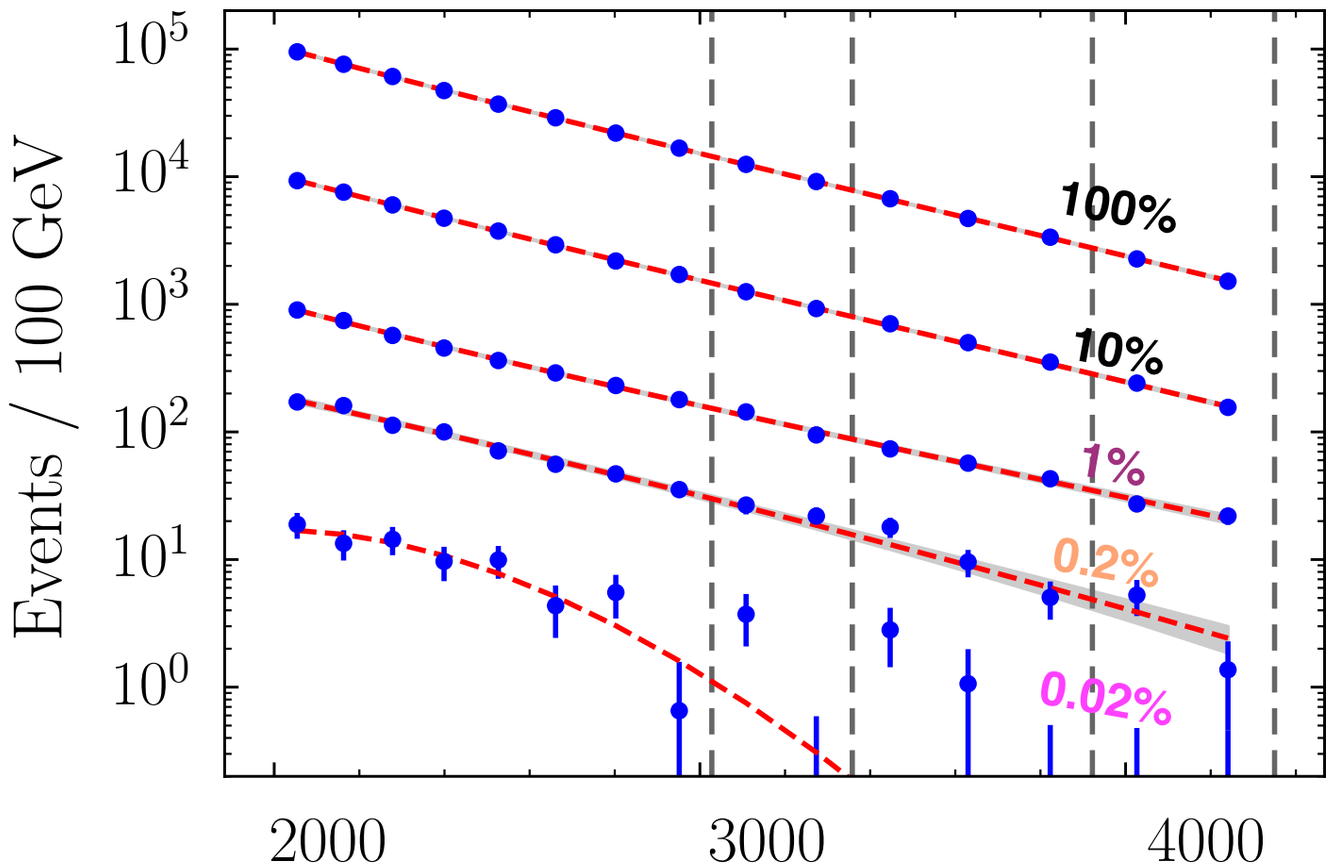
no cut on NN

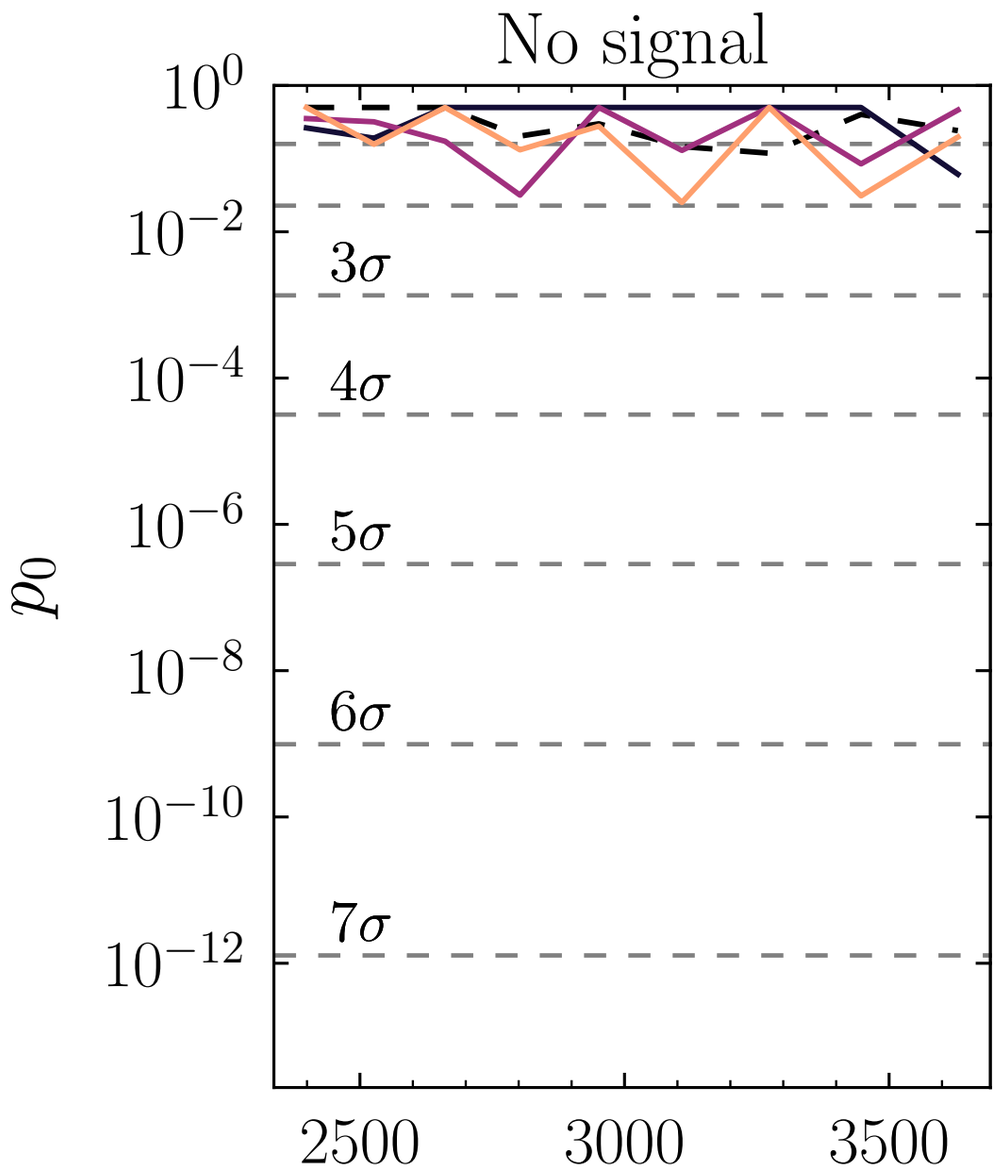
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

Application to dijets





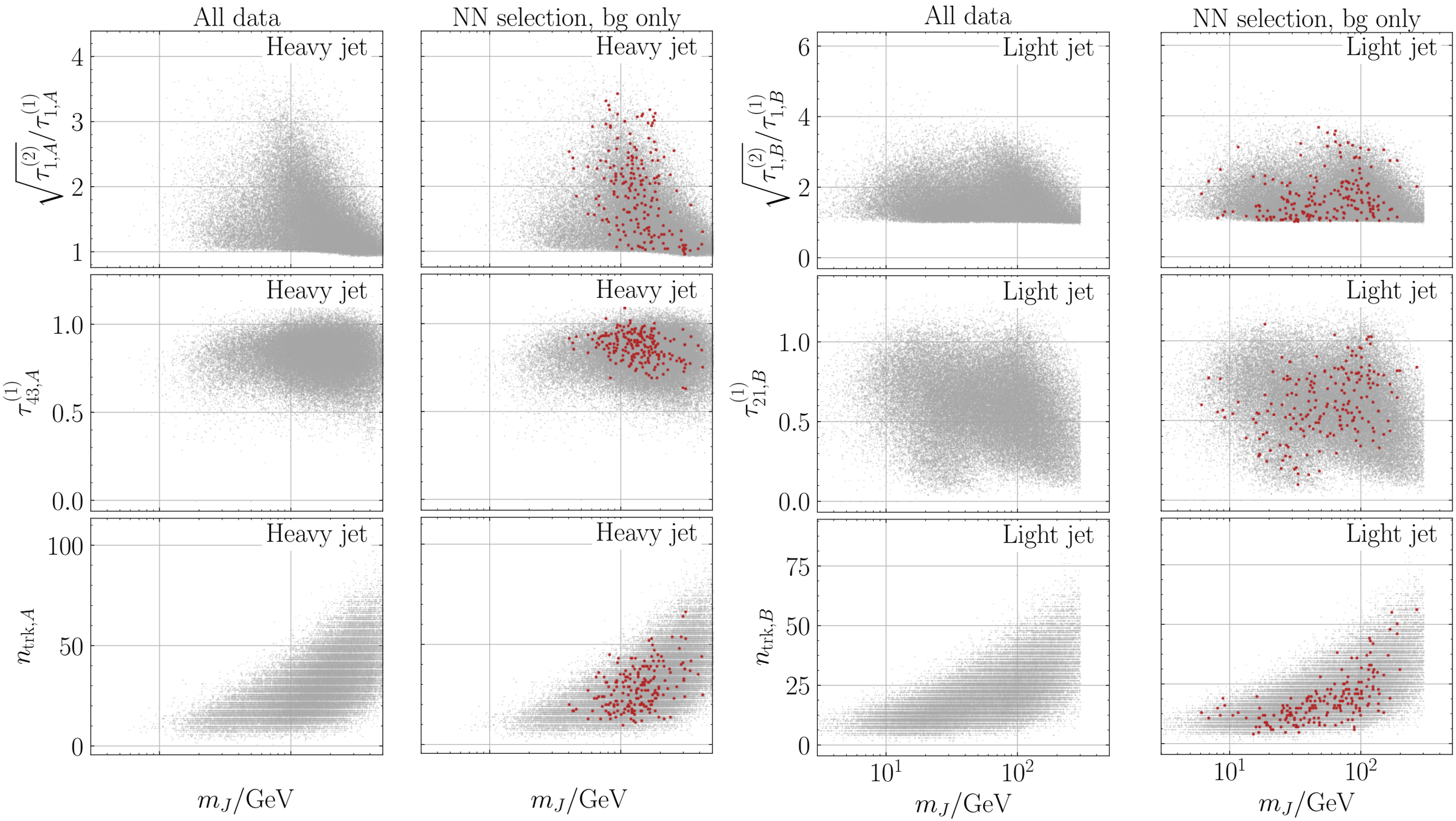
no cut on NN

most 10% signal-region-like

most 1% signal-region-like

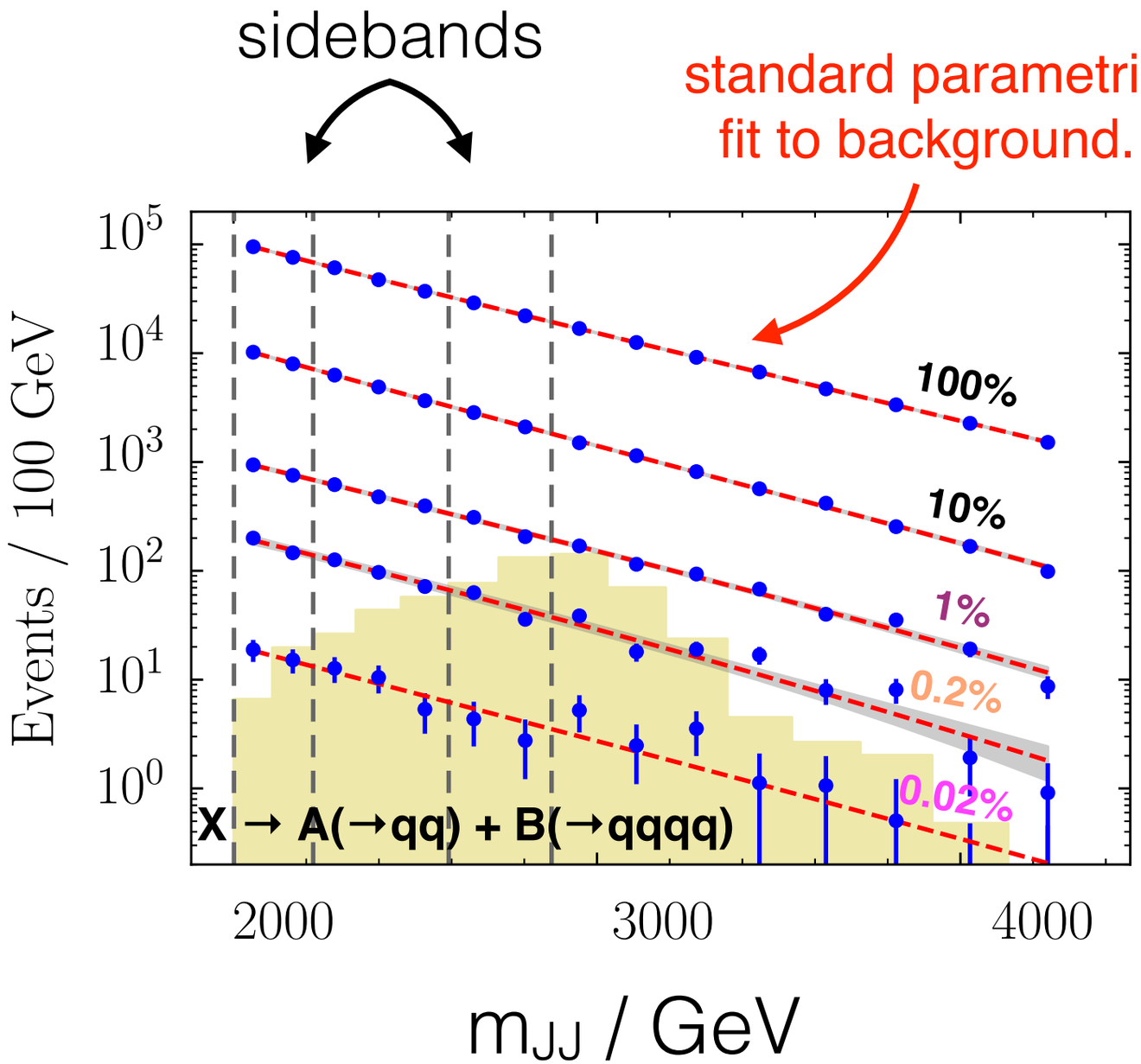
most 0.2% signal-region-like

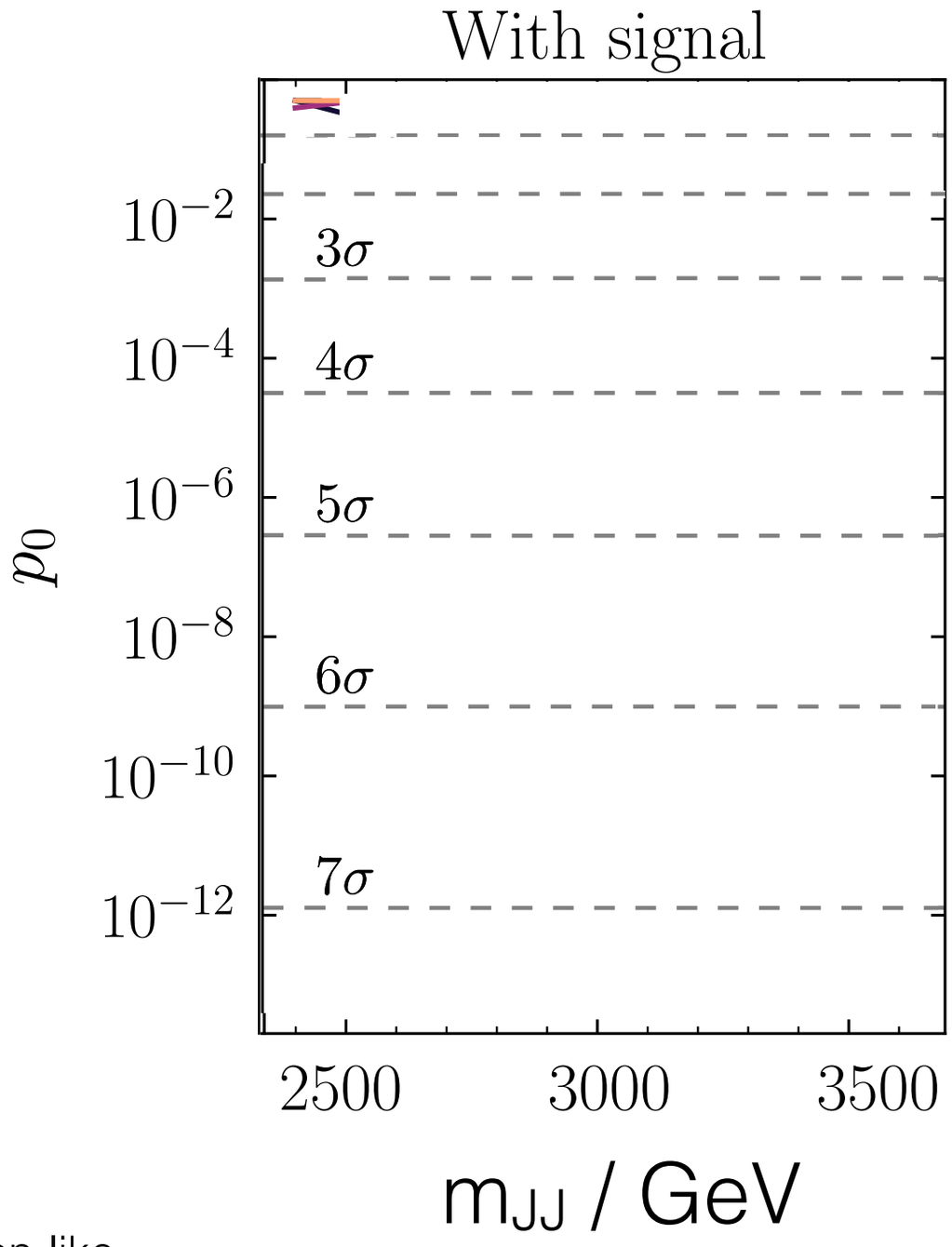
What is the network learning?



Learns nothing … as desired !

…and when there is a signal?





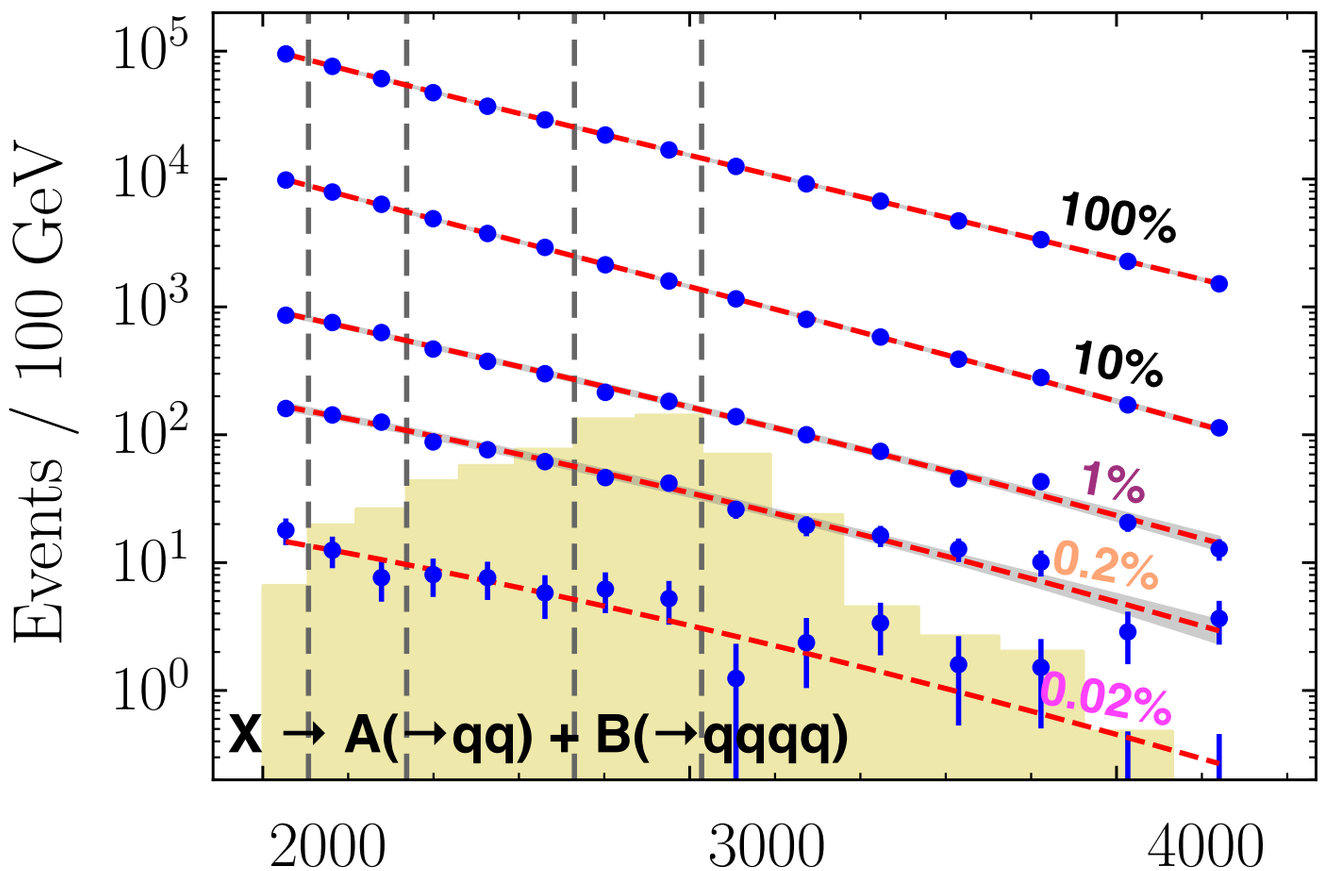
no cut on NN

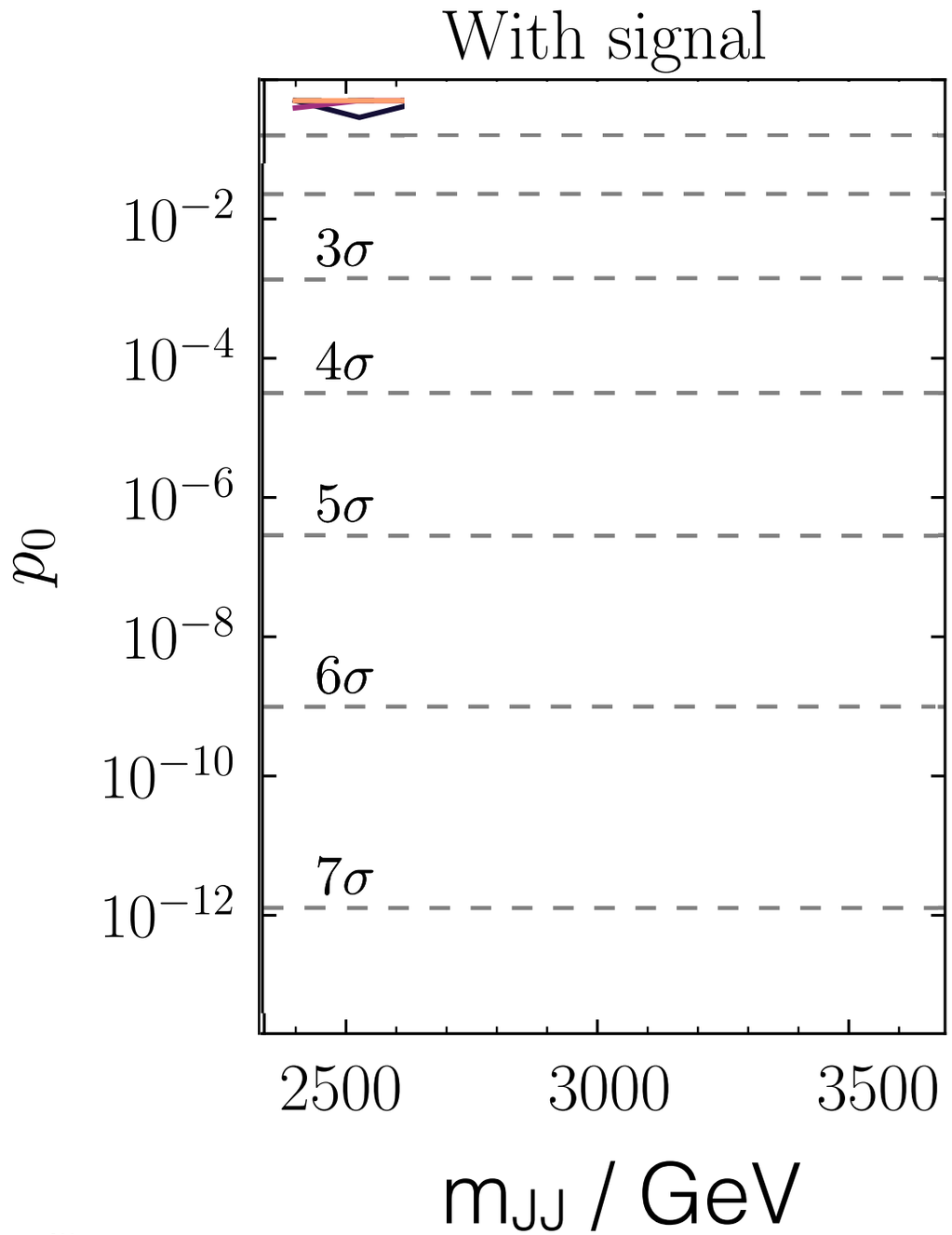
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

…and when there is a signal?





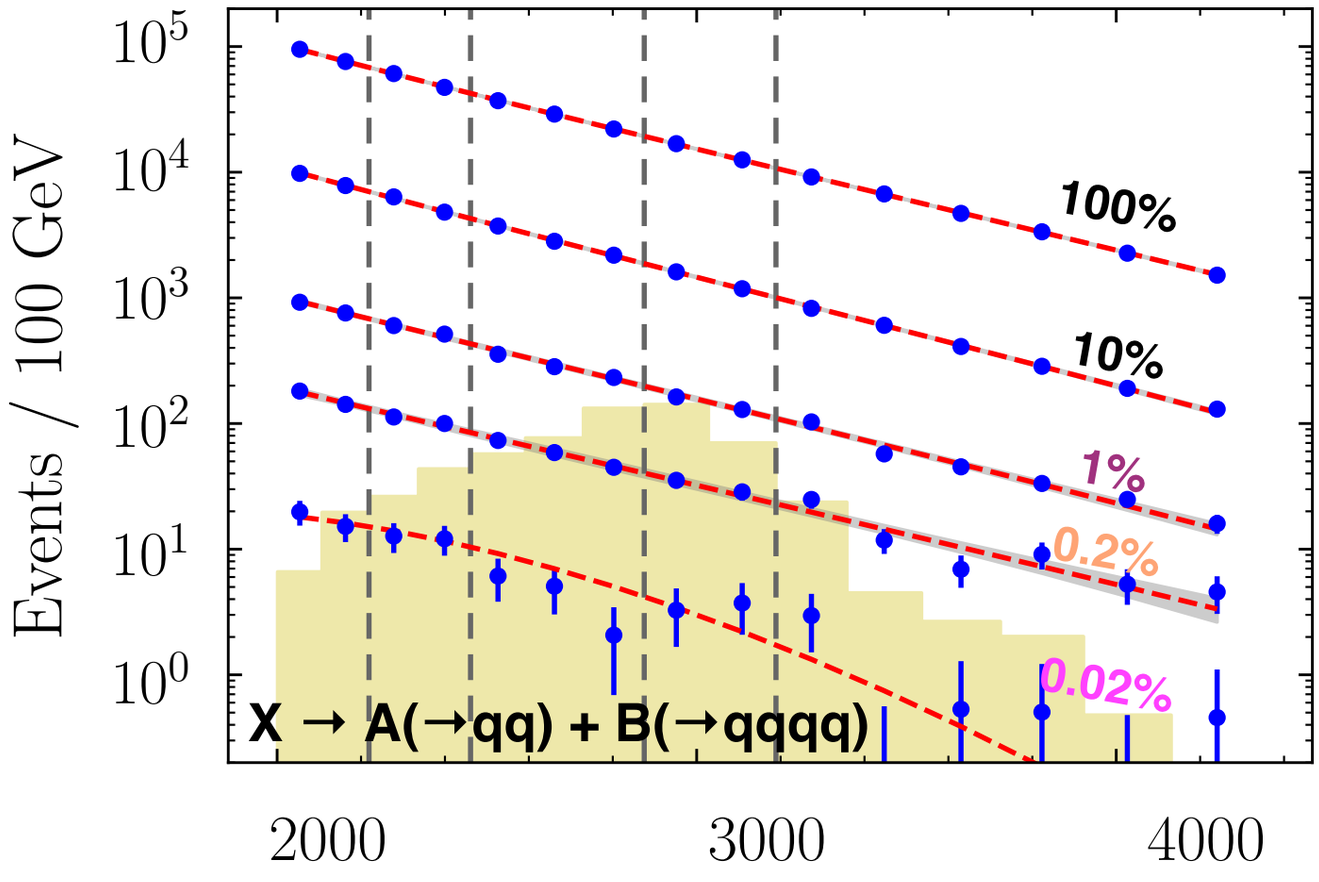
no cut on NN

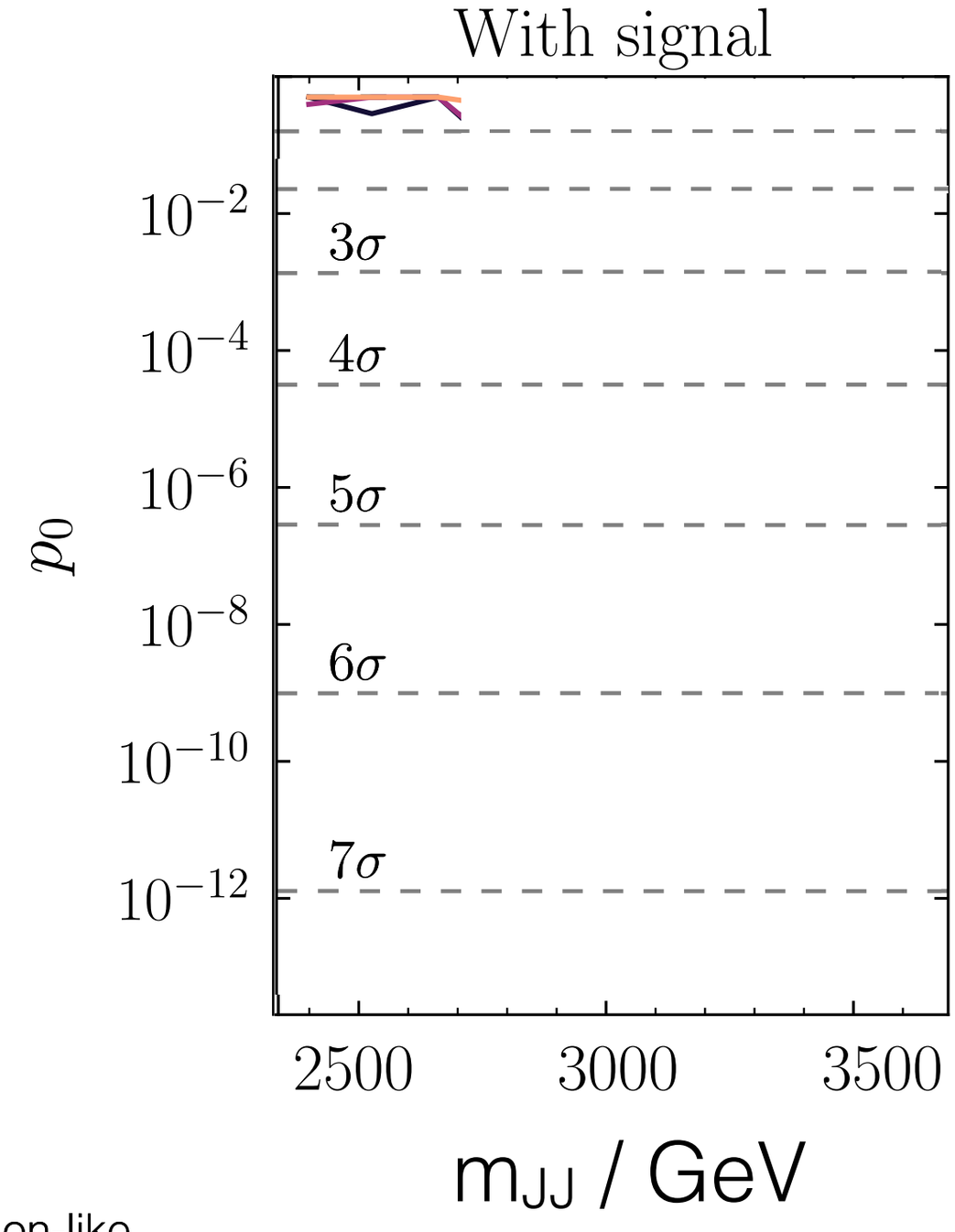
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

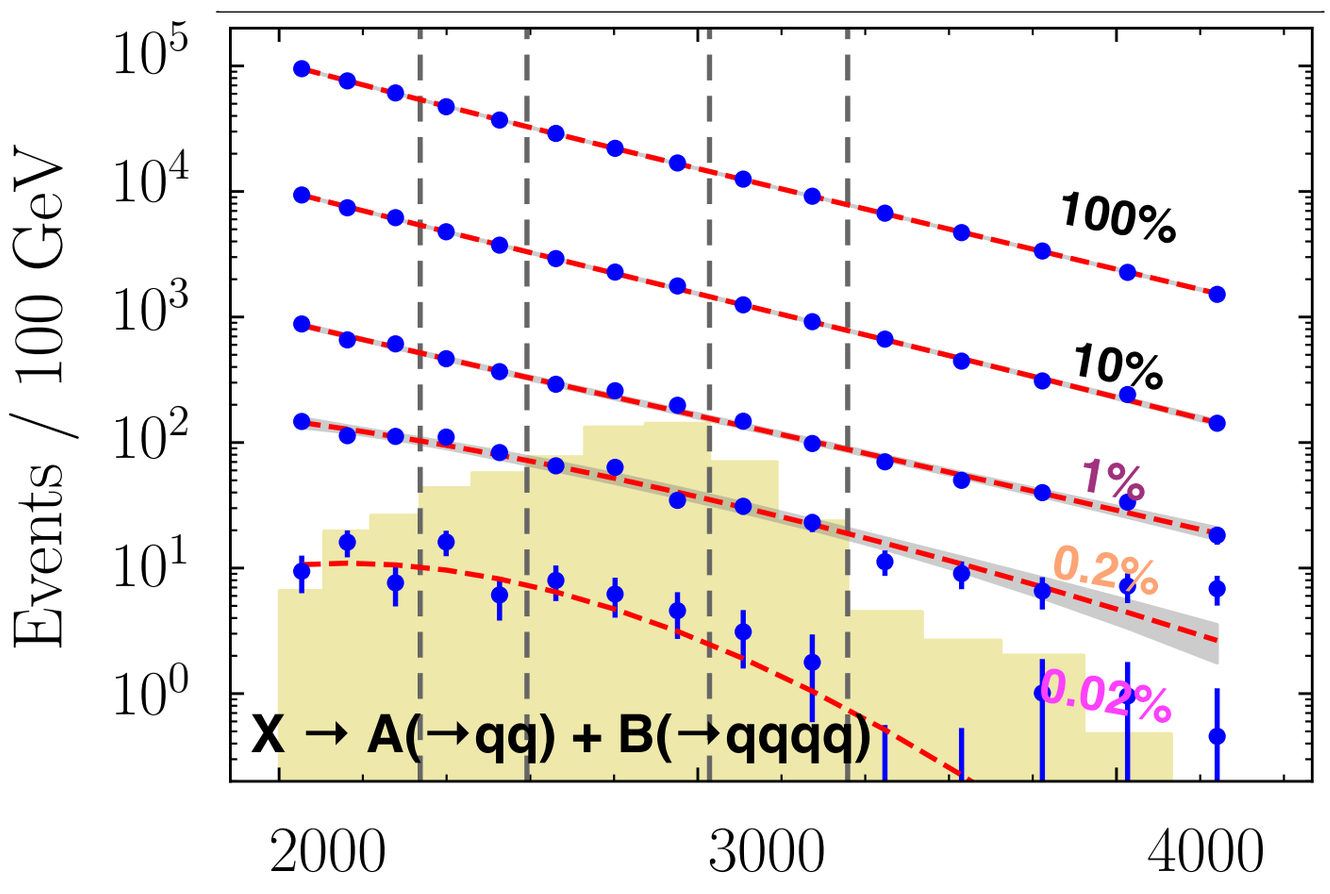
…and when there is a signal?

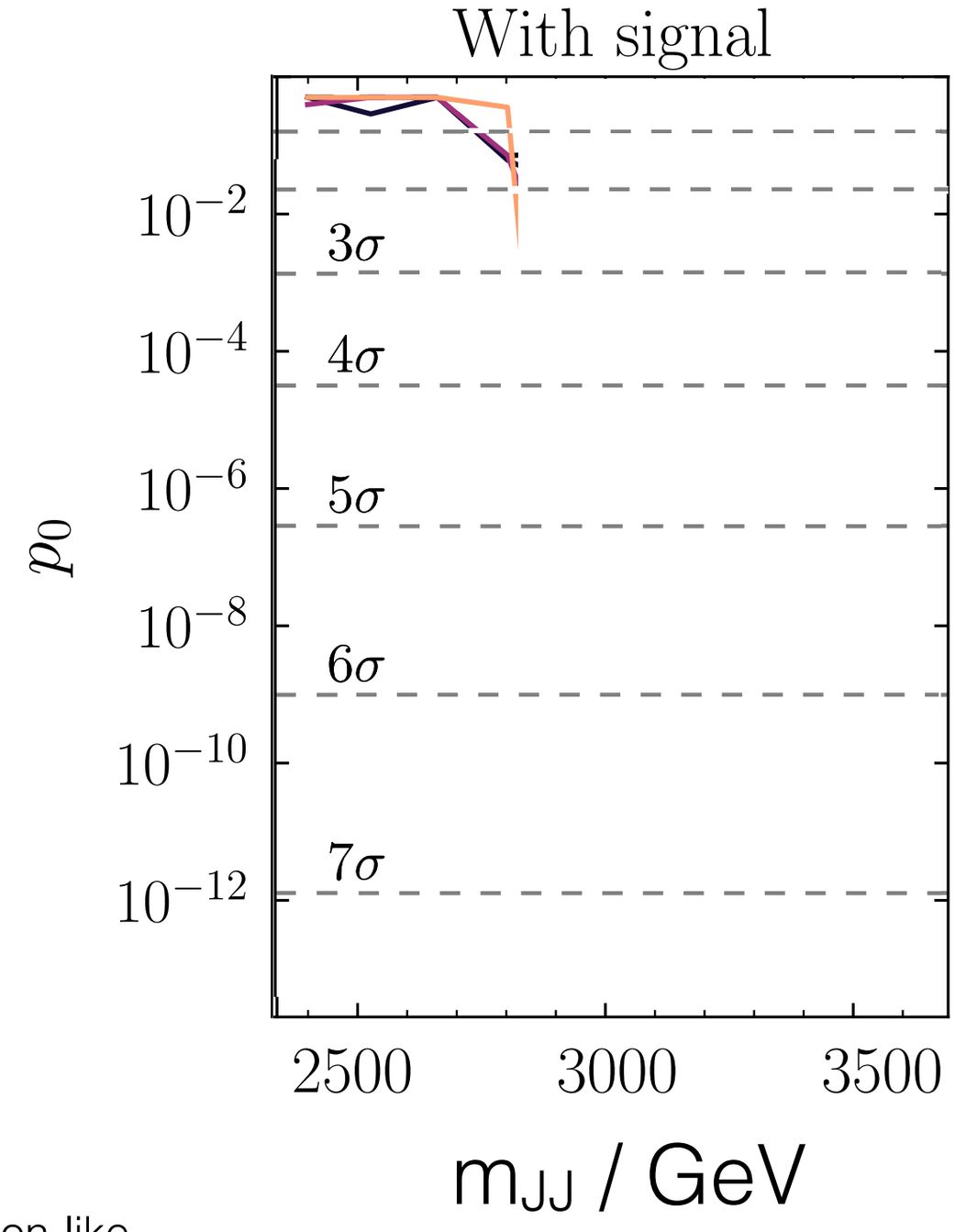




- no cut on NN
- most 10% signal-region-like
- most 1% signal-region-like
- most 0.2% signal-region-like

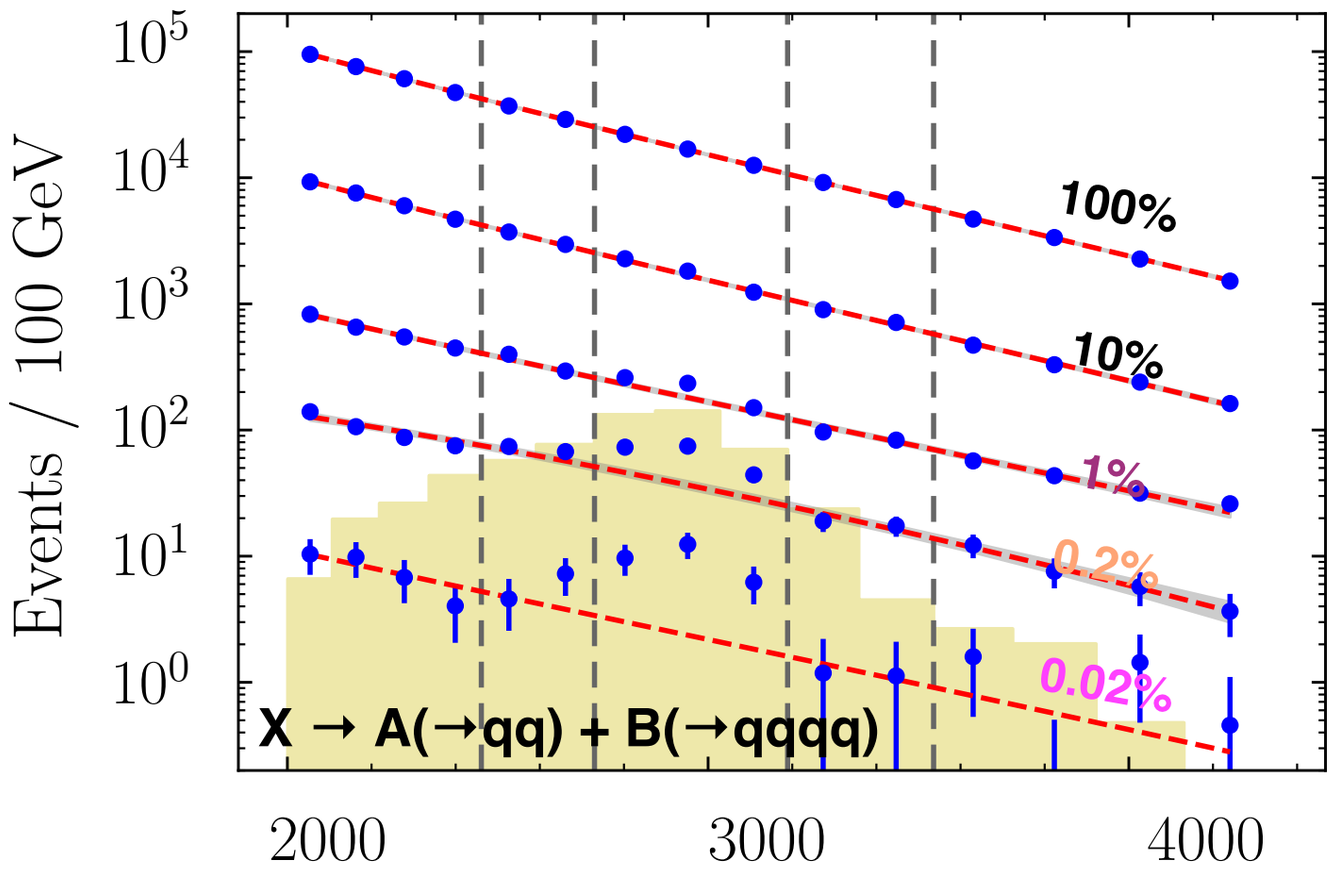
…and when there is a signal?

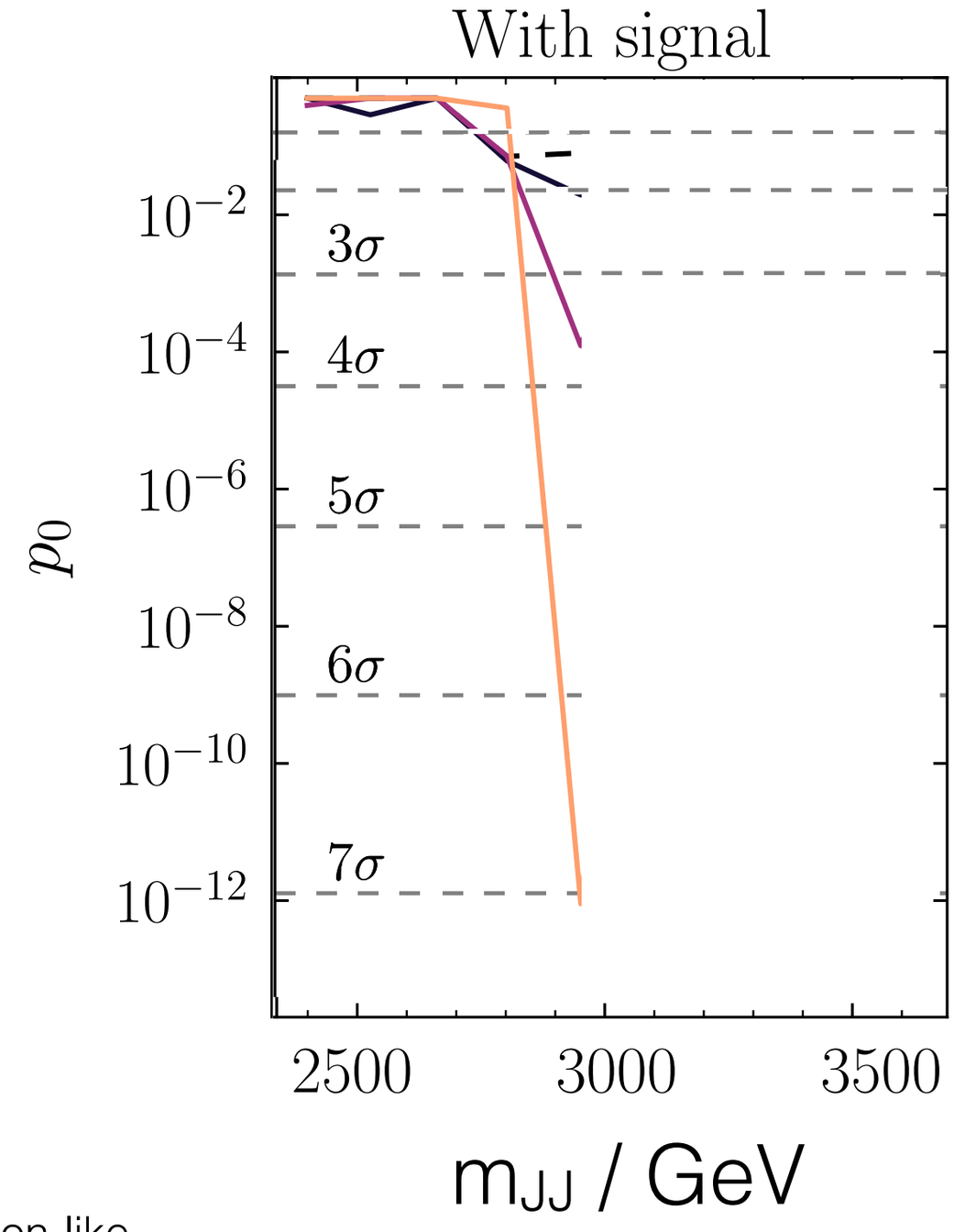




- no cut on NN
- most 10% signal-region-like
- most 1% signal-region-like
- most 0.2% signal-region-like

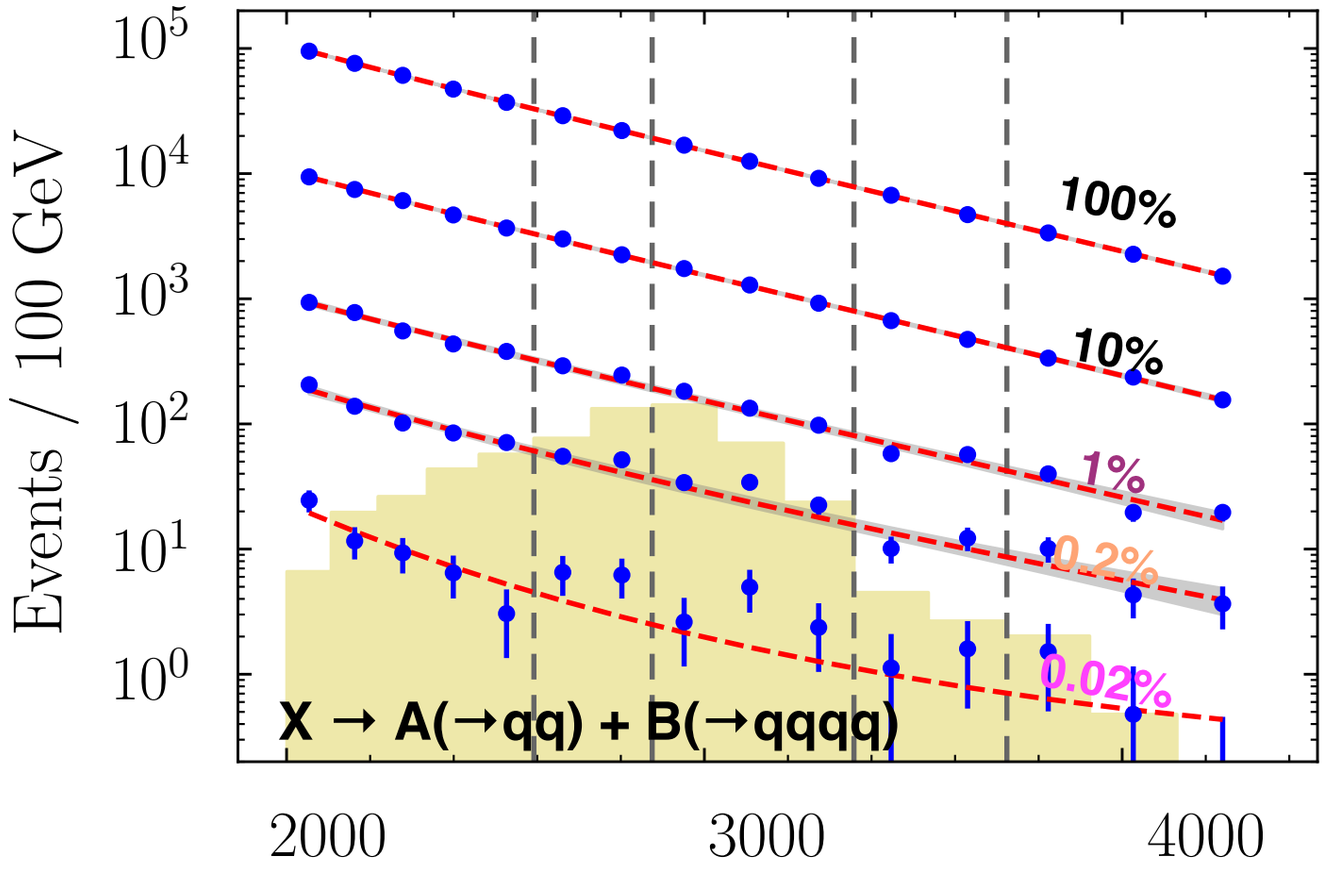
…and when there is a signal?

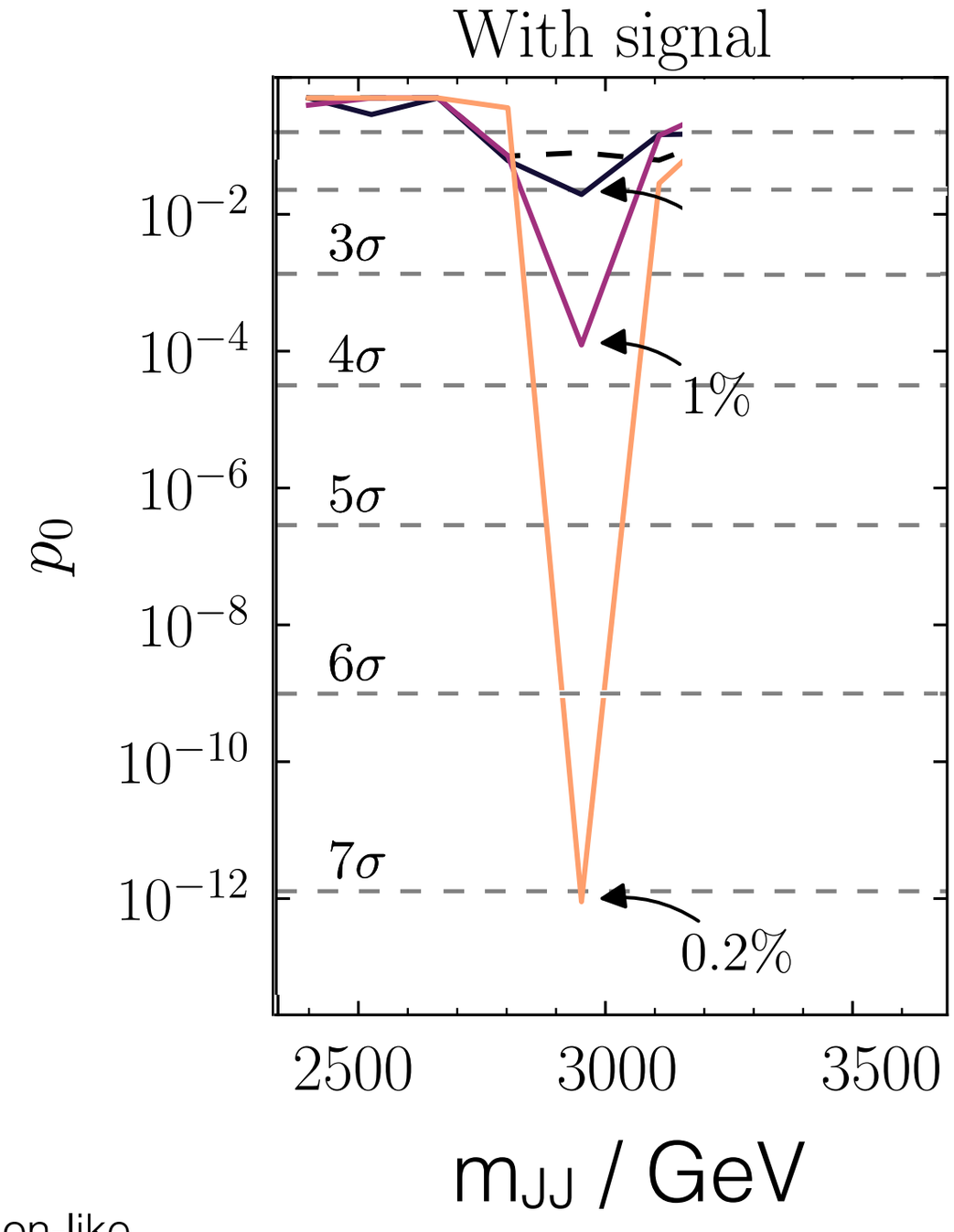




- - - - no cut on NN
——— most 10% signal-region-like
——— most 1% signal-region-like
——— most 0.2% signal-region-like

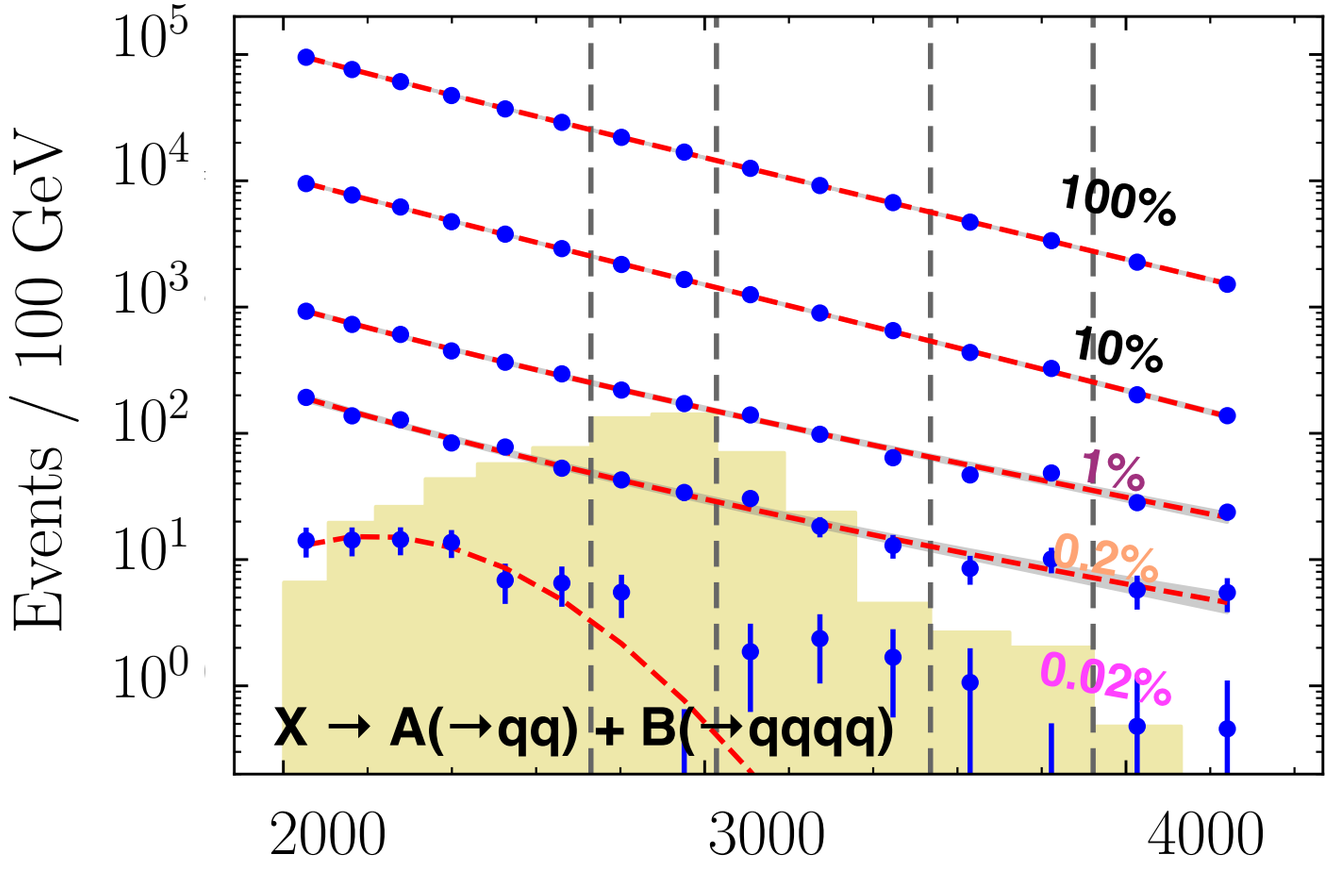
…and when there is a signal?

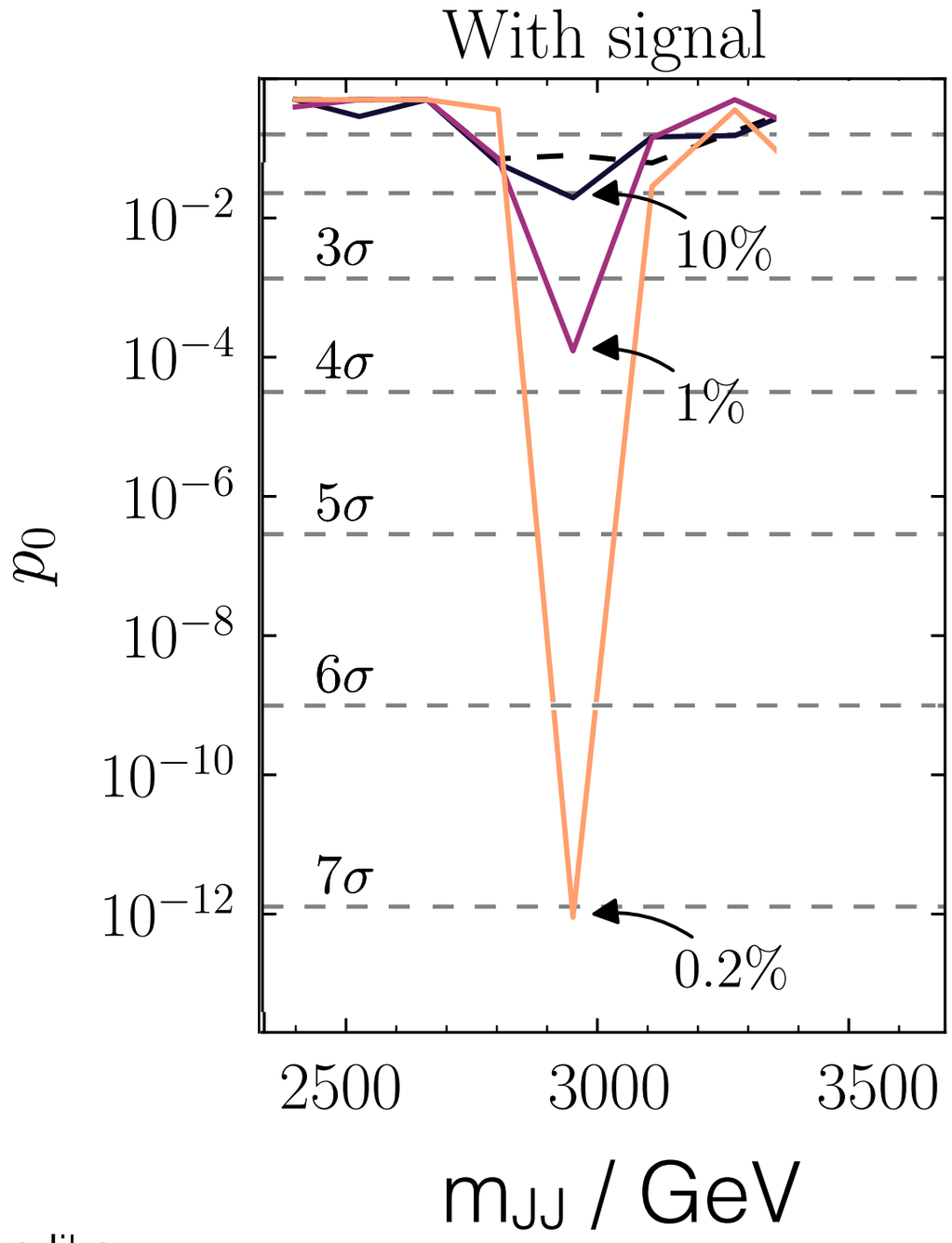




- no cut on NN
- most 10% signal-region-like
- most 1% signal-region-like
- most 0.2% signal-region-like

…and when there is a signal?





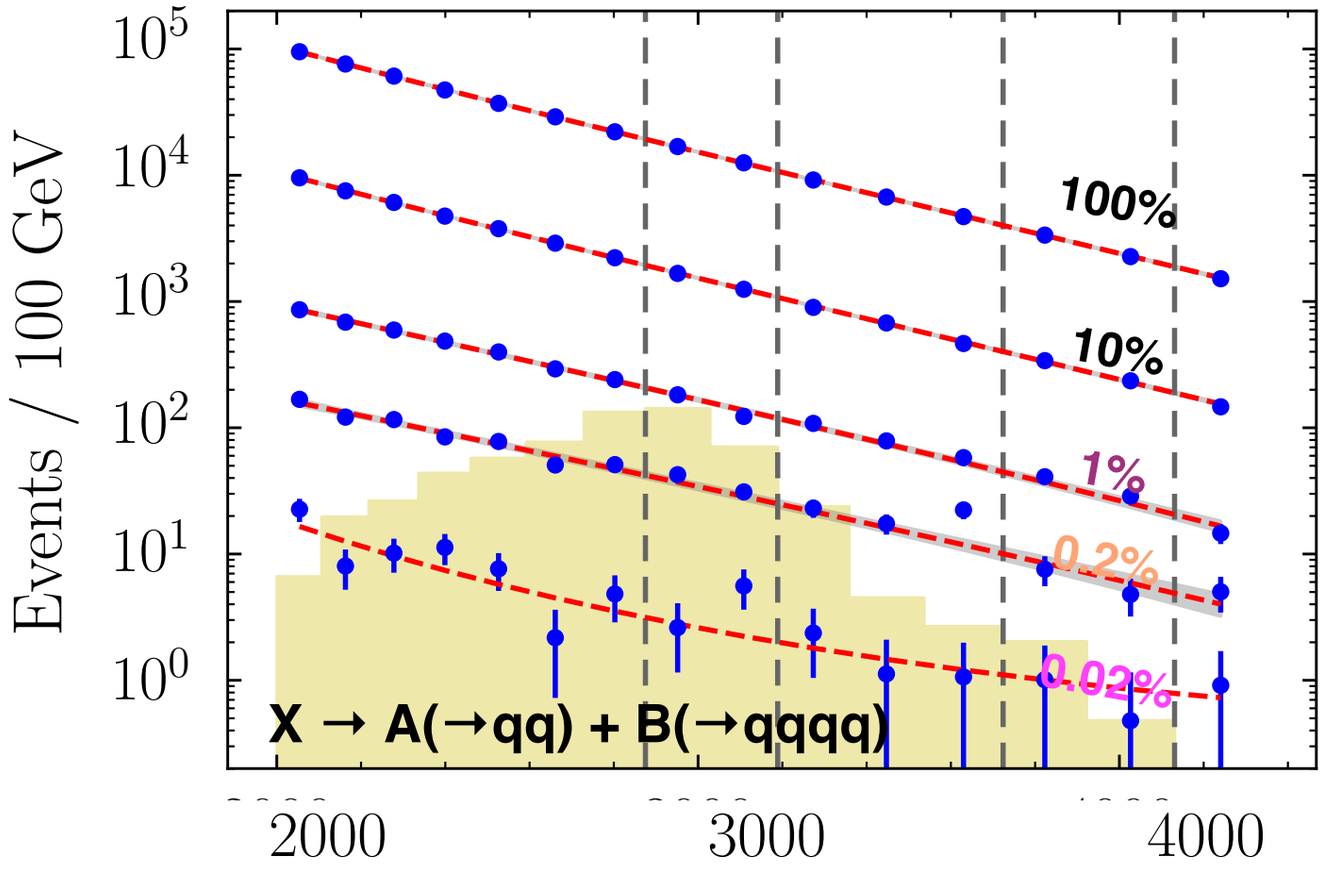
no cut on NN

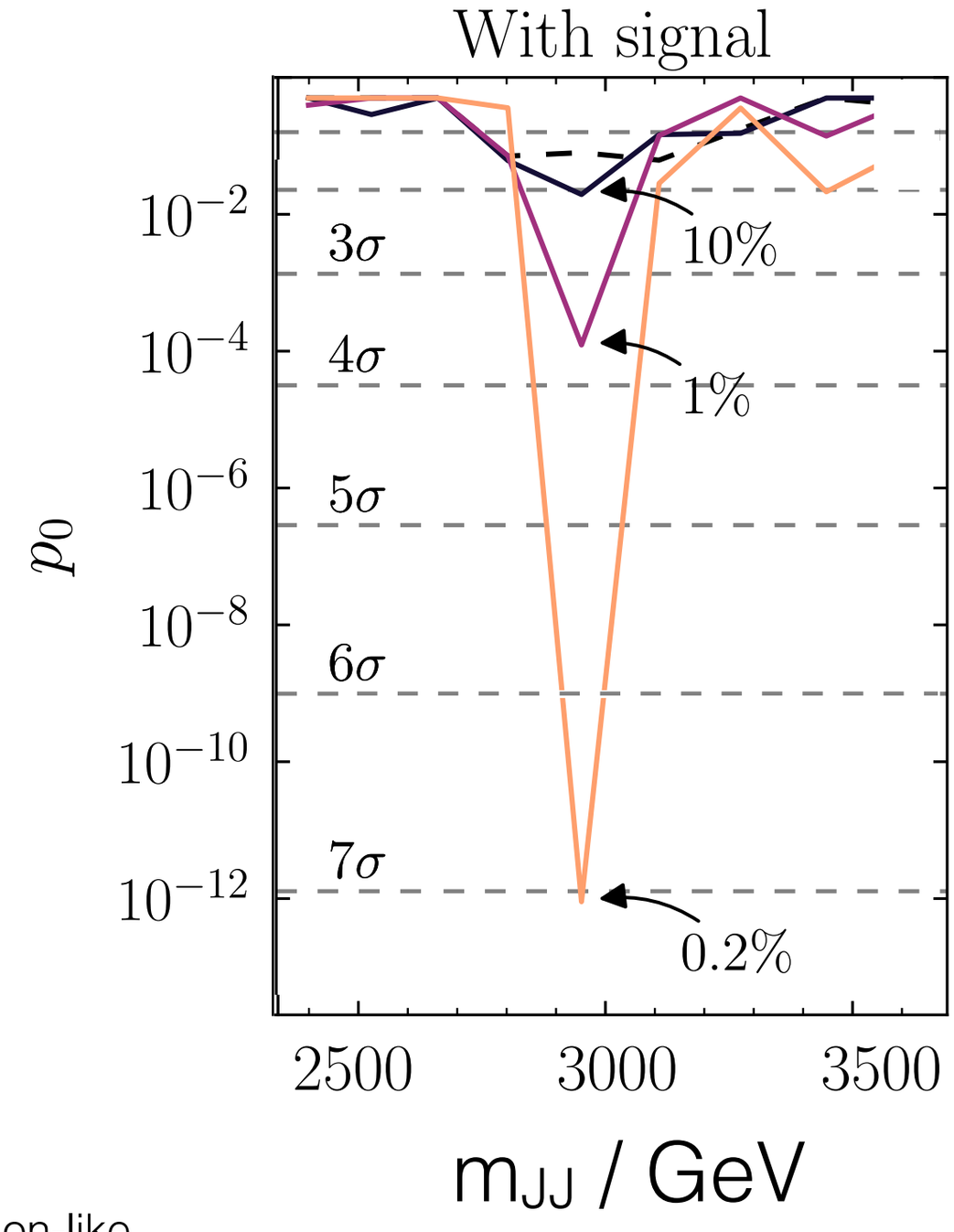
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

…and when there is a signal?





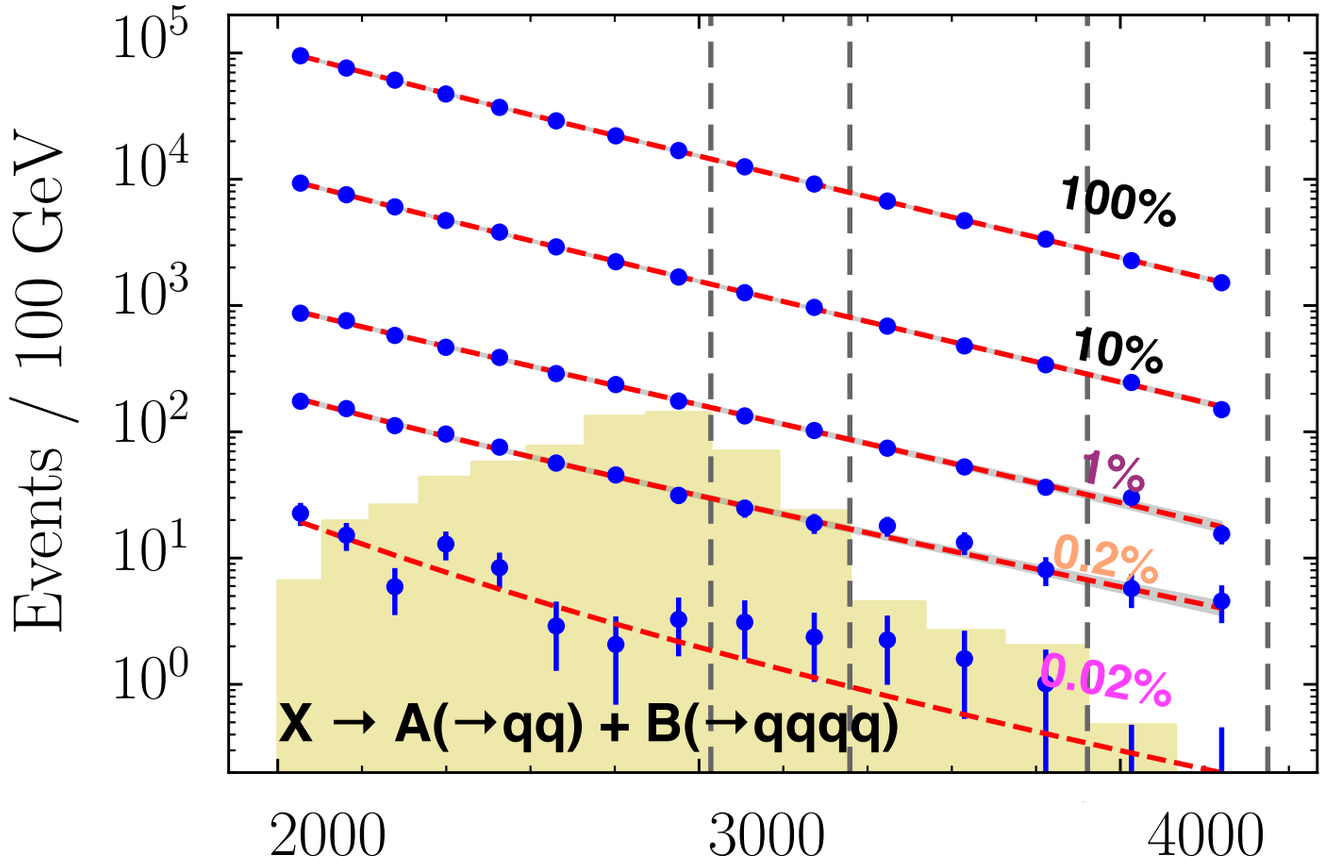
no cut on NN

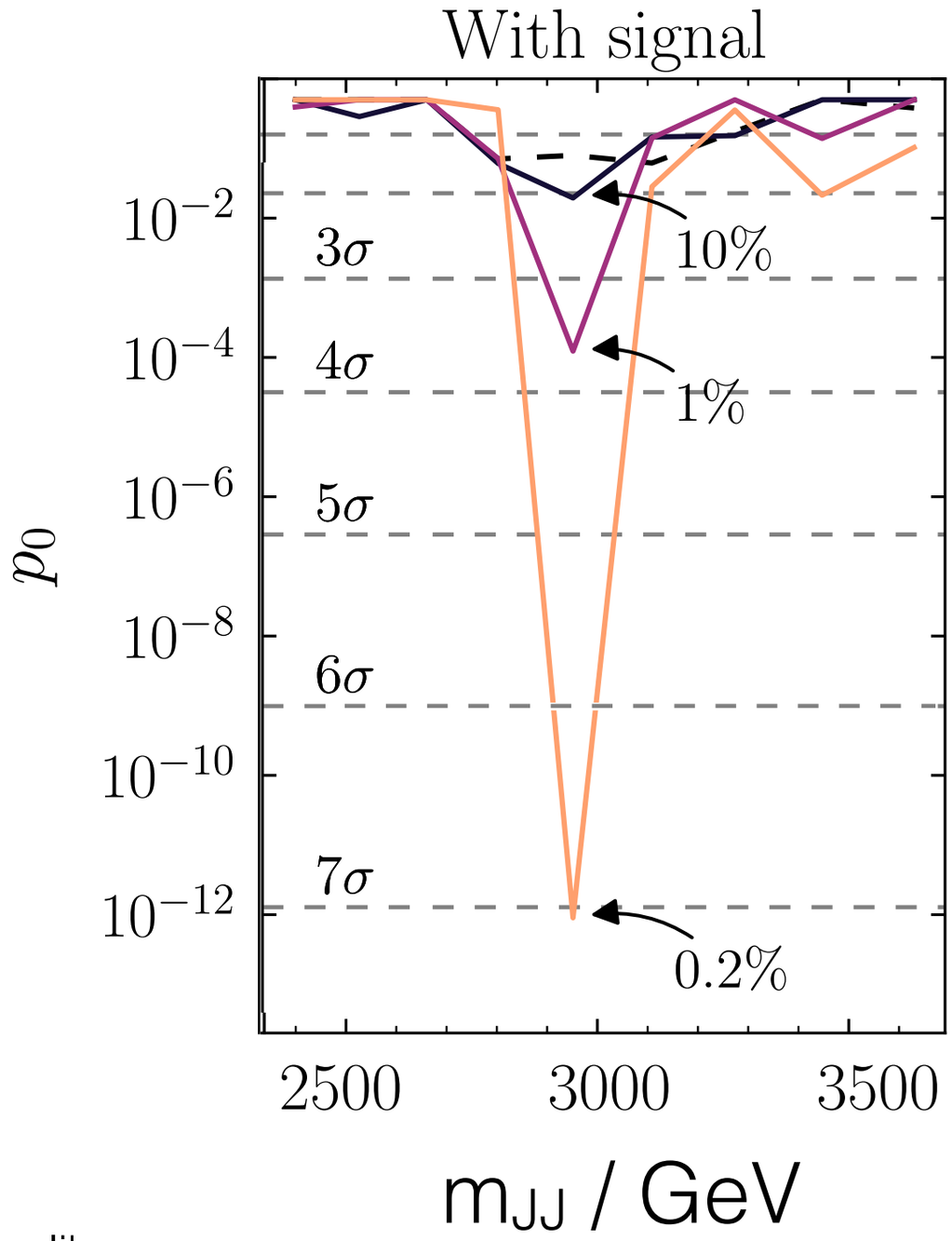
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

…and when there is a signal?





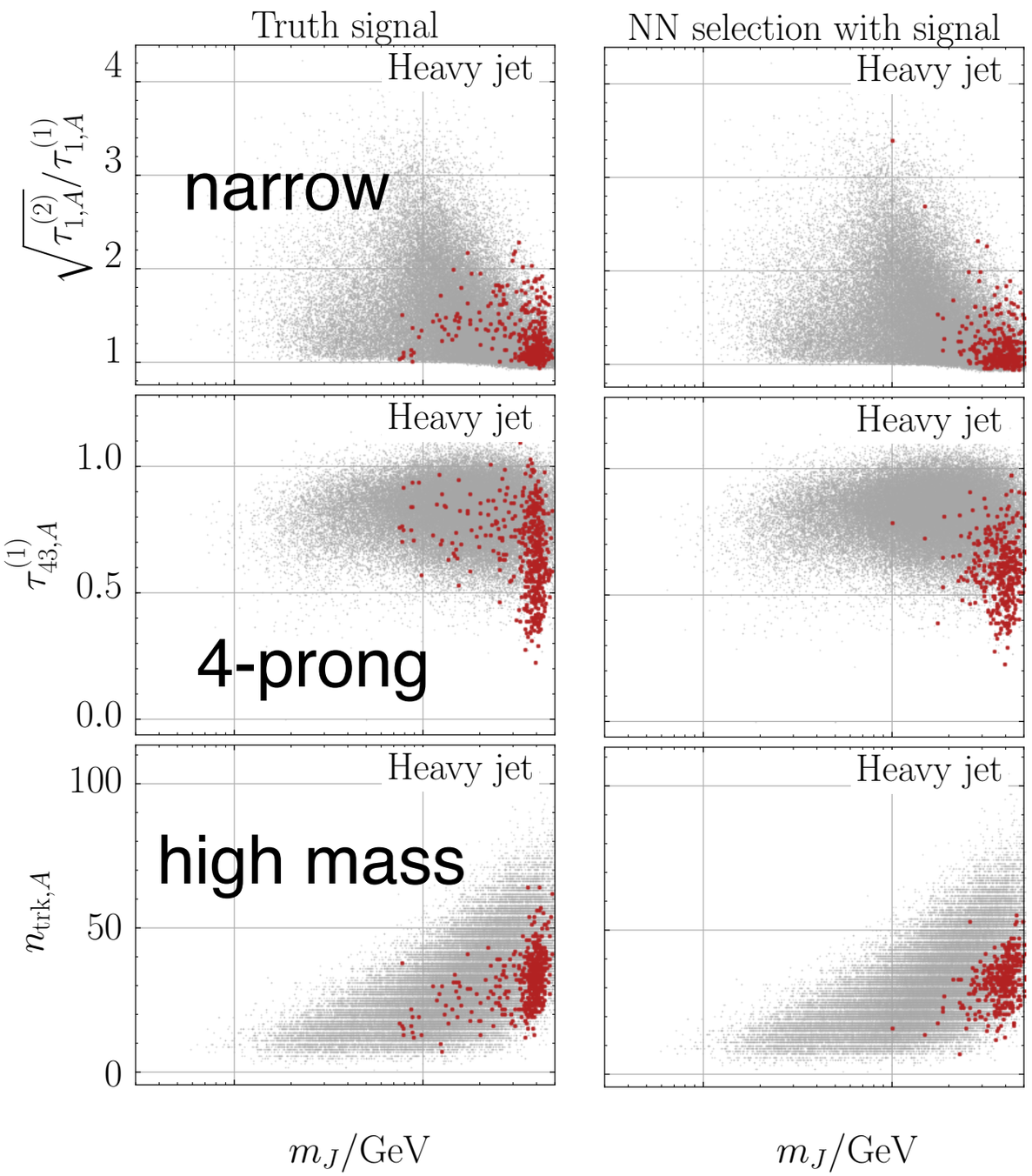
no cut on NN

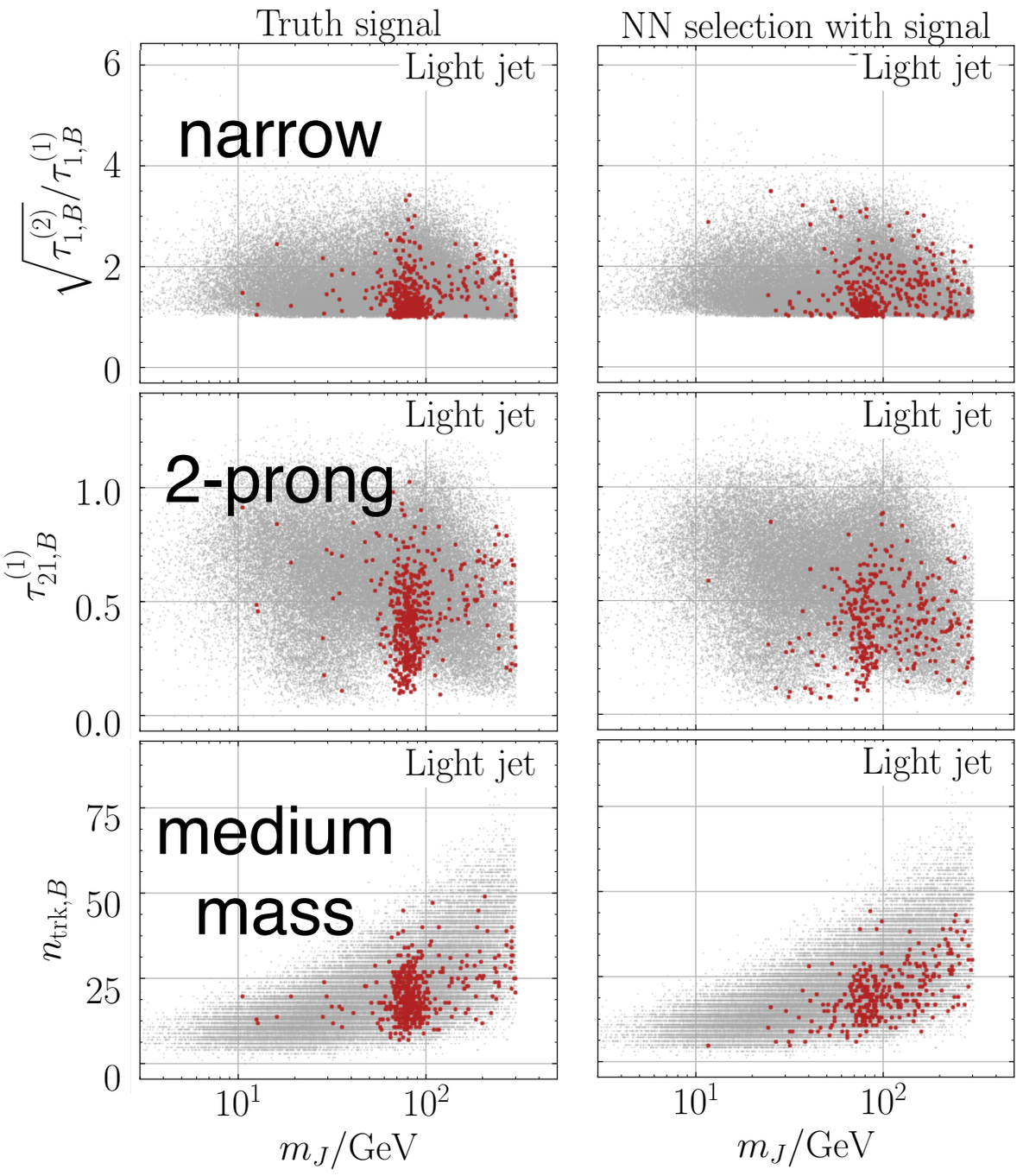
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

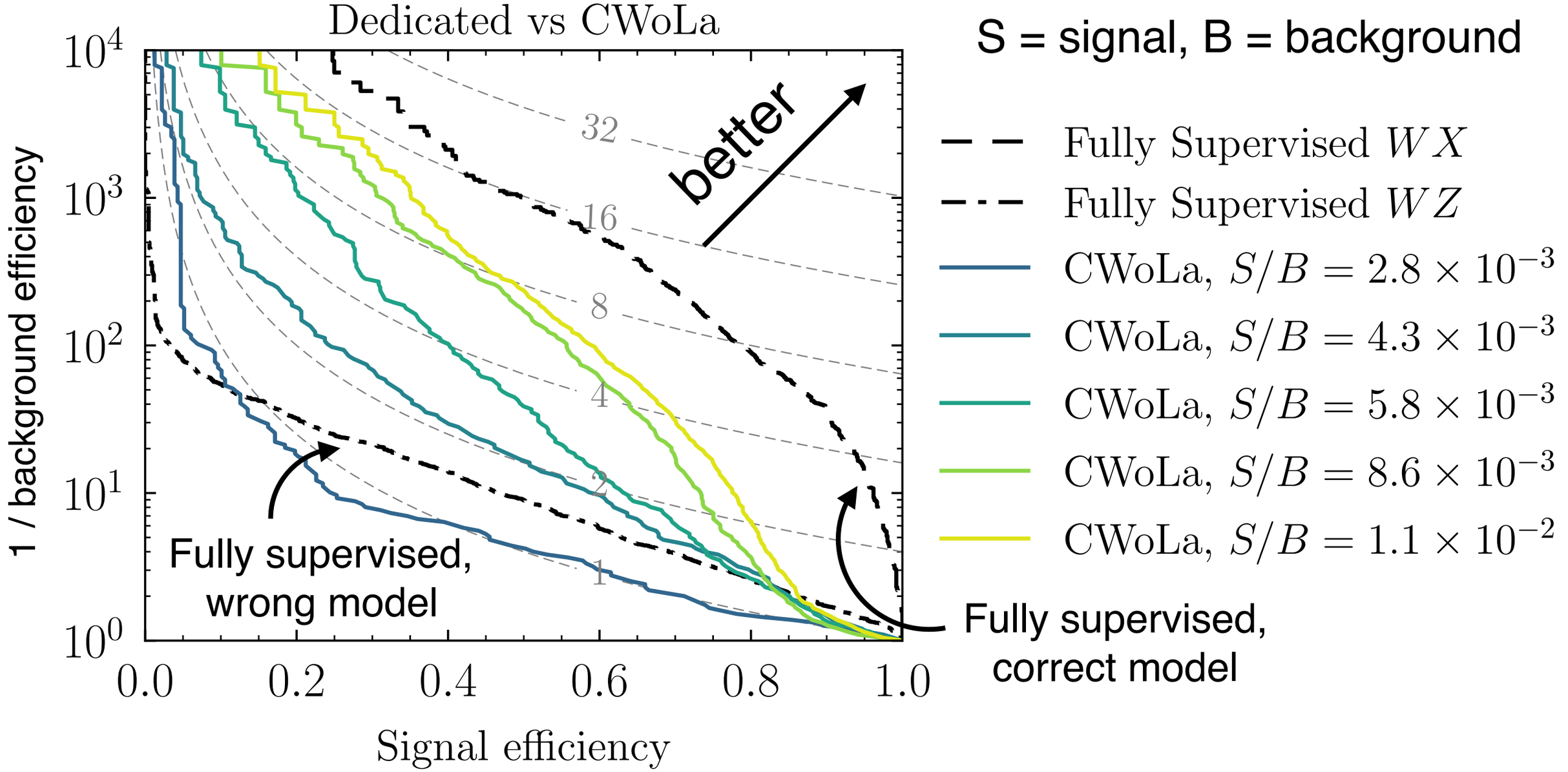
What is the network learning?





Learns to find the signal ! **X → A(→qq) + B(→qqqq)**

CWoLa hunting vs. Full Supervision

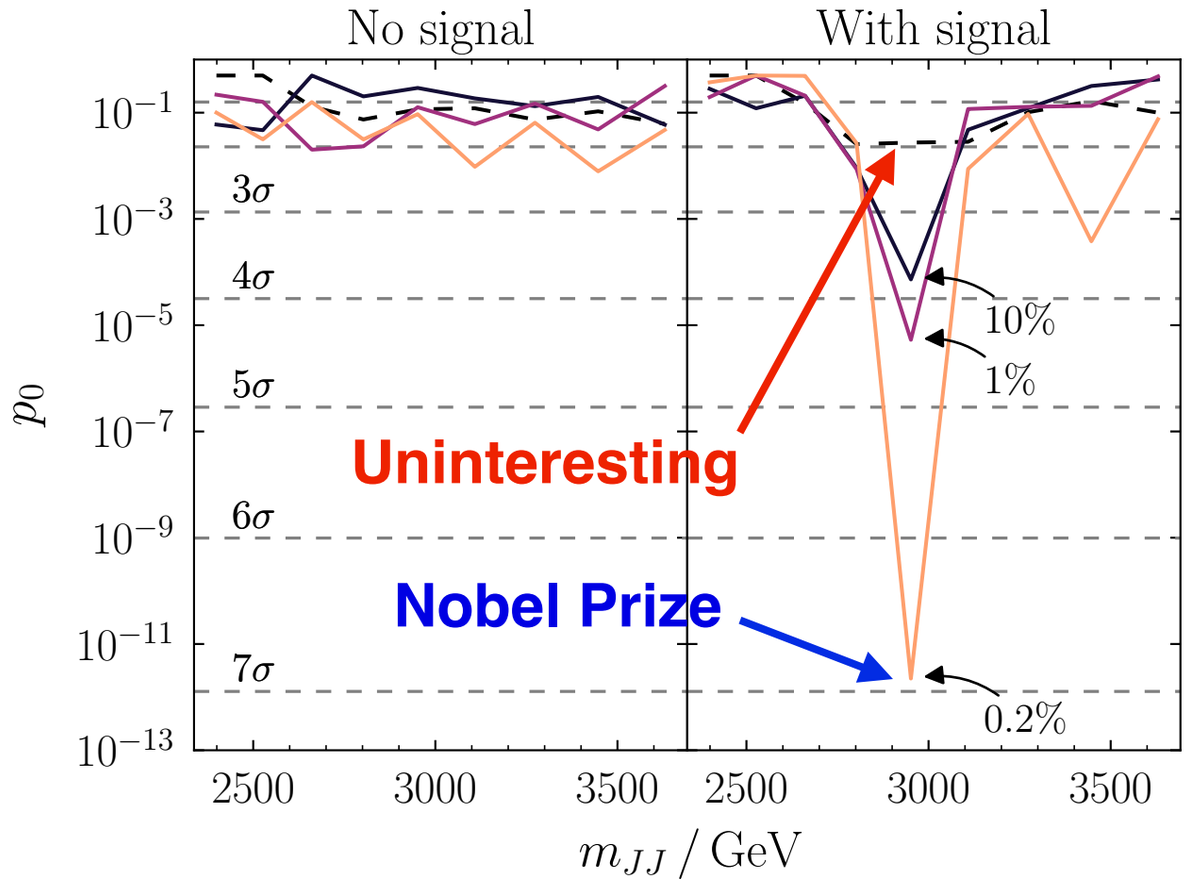


If you know what you are looking for, you should look for it. If you don't know, then CWoLa hunting may be able to catch it!

Outlook*

Jack Collins, Kiel Howe, **BPN**, PRL 121, 241803 (2018)

If we continue to find nothing (and even if we find something!), it is important to broaden our scope.



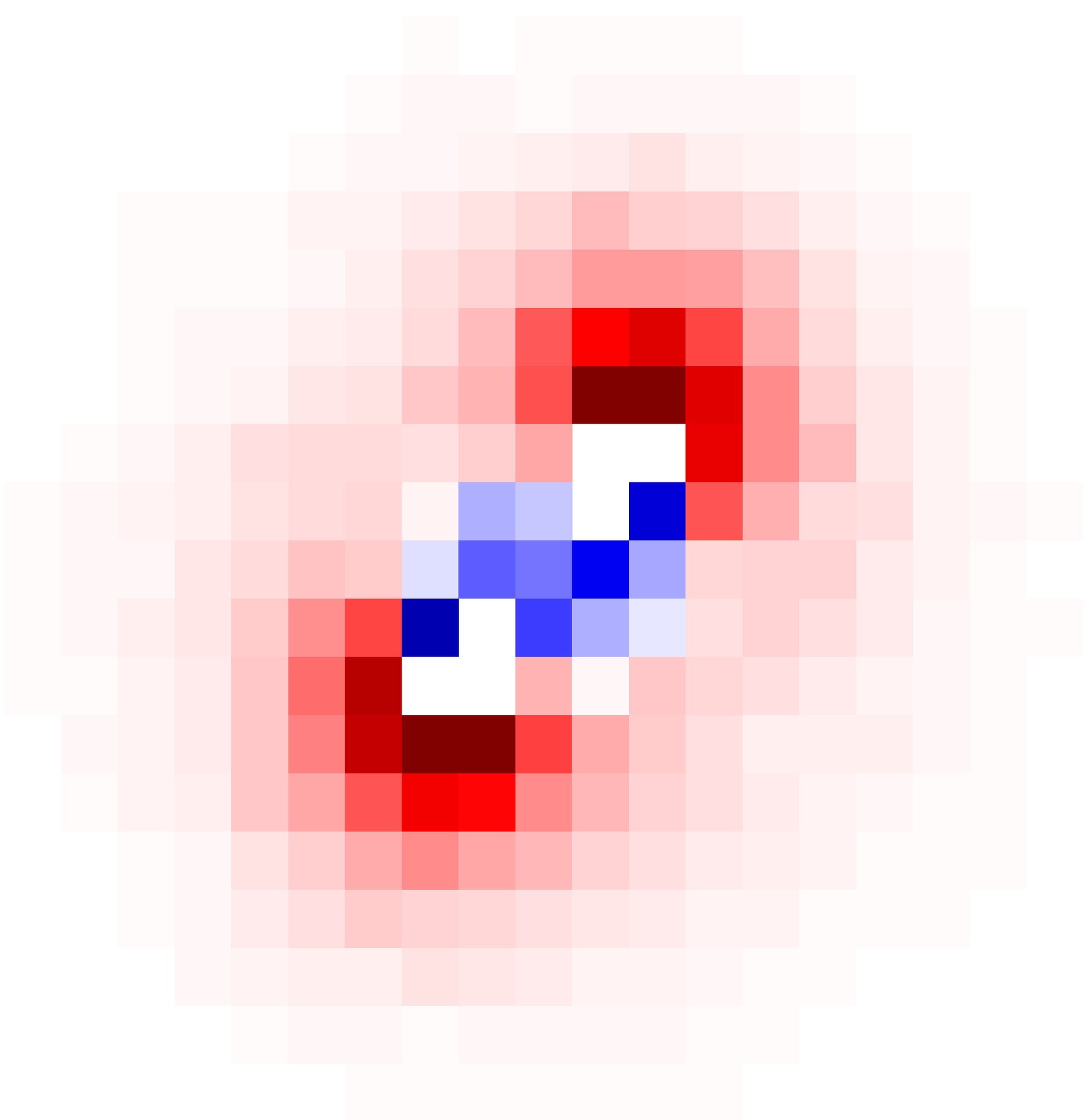
There may be BSM hiding, but we need **hyper-variate vision** to find it … ML can help us learn something **fundamental about nature** !

*Image from this article. This Koala is actually being freed - I do not condone violence against these animals!

…many thanks to a fun and insightful collaboration with these guys!

Jack Collins

Kiel Howe



Fin.

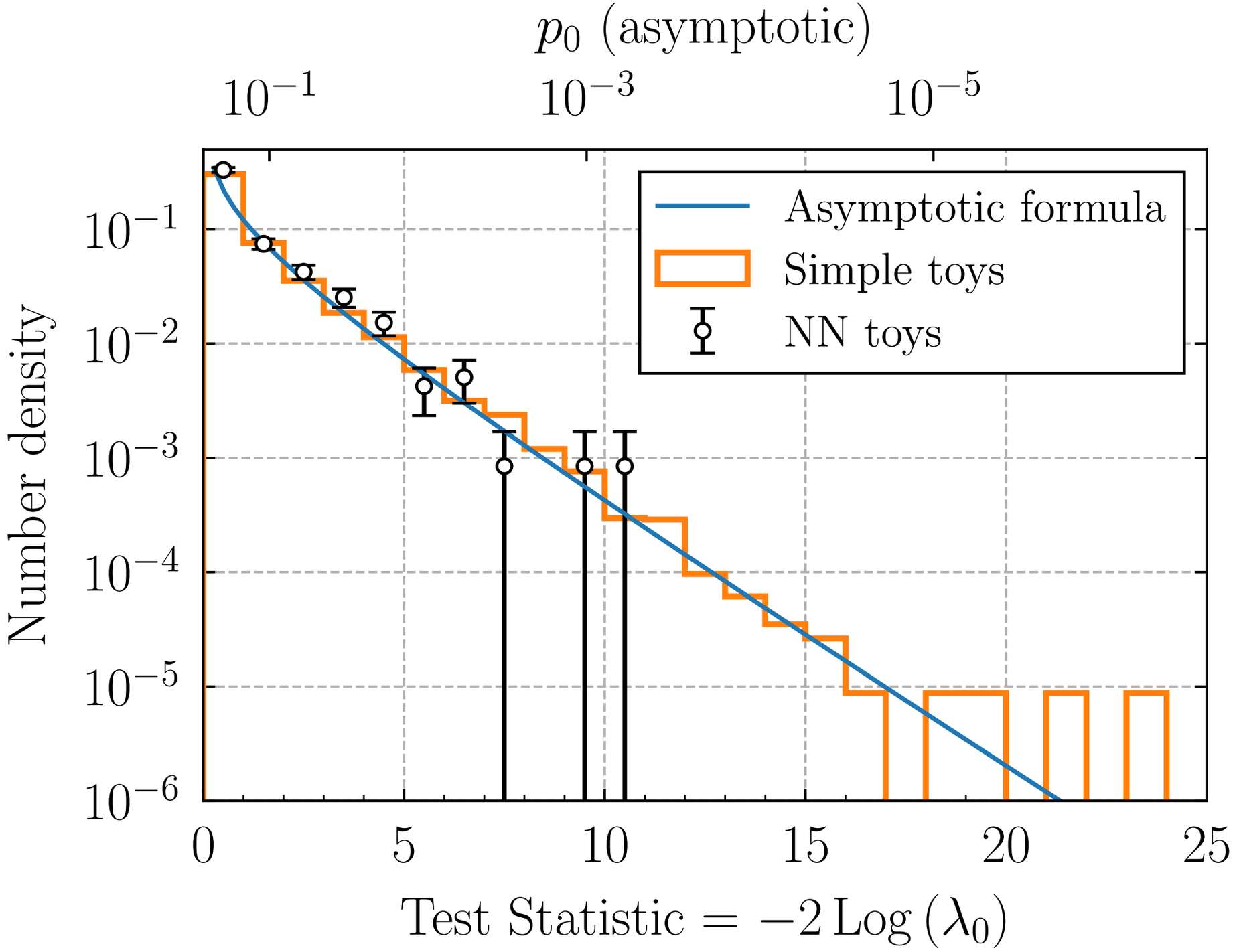
Autoencoders

Marco Farina, Yuichiro Nakai, David Shih,
https://arxiv.org/abs/1808.08992

Theo Heimel, Gregor Kasieczka, Tilman Plehn,
Jennifer M Thompson, https://arxiv.org/abs/1808.08979

Olmo Cerri, Thong Q. Nguyen, Maurizio Pierini, Maria Spiropulu,
Jean-Roch Vlimant, https://arxiv.org/abs/1811.10276

Statistics


p_0 (asymptotic)
10^{-1}
10^{-3}
10^{-5}
Asymptotic formula
Simple toys
NN toys
Number density
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
5
10
15
20
25
Test Statistic $= -2\,\mathrm{Log}\,(\lambda_0)$