Deep learning for LHC classification, regression, generation, and beyond

Benjamin Nachman

Lawrence Berkeley National Laboratory

cern.ch/bnachman

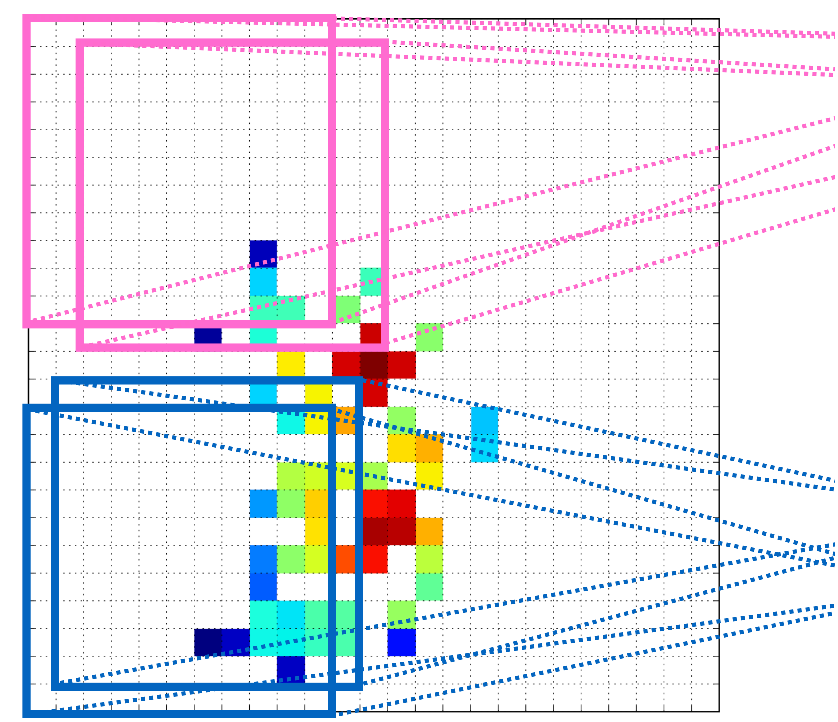
bpnachman@lbl.gov

@bpnachman

bnachman

BERKELEY LAB

BERKELEY EXPERIMENTAL PARTICLE PHYSICS



DANCE 2019

October 29, 2019

Deep learning for LHC classification, regression, generation, and **beyond**

Benjamin Nachman

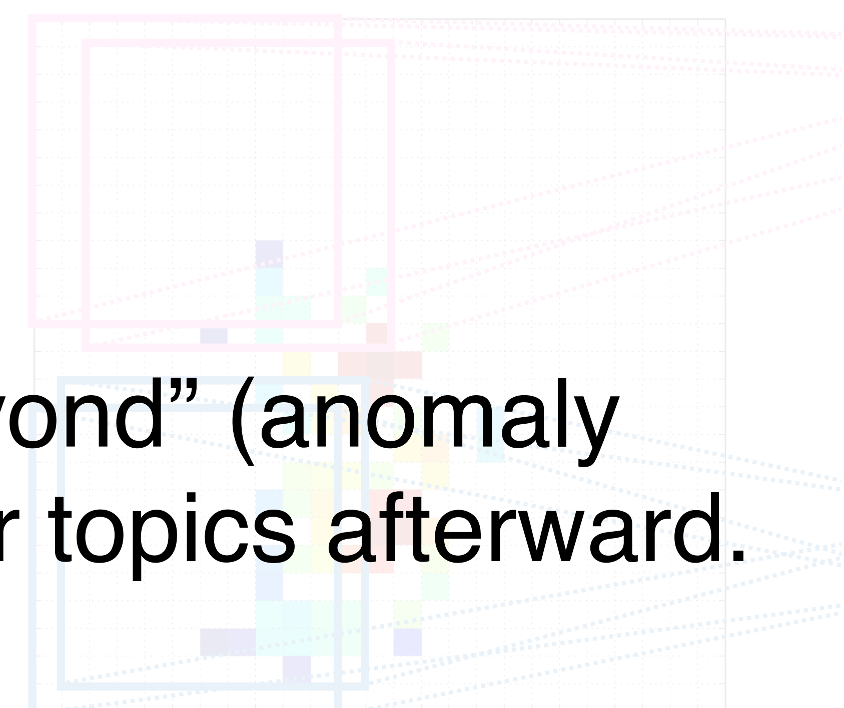
Lawrence Berkeley National Laboratory

cern.ch/bnachman

@bnachman

bnachman

bpnachman@lbl.gov

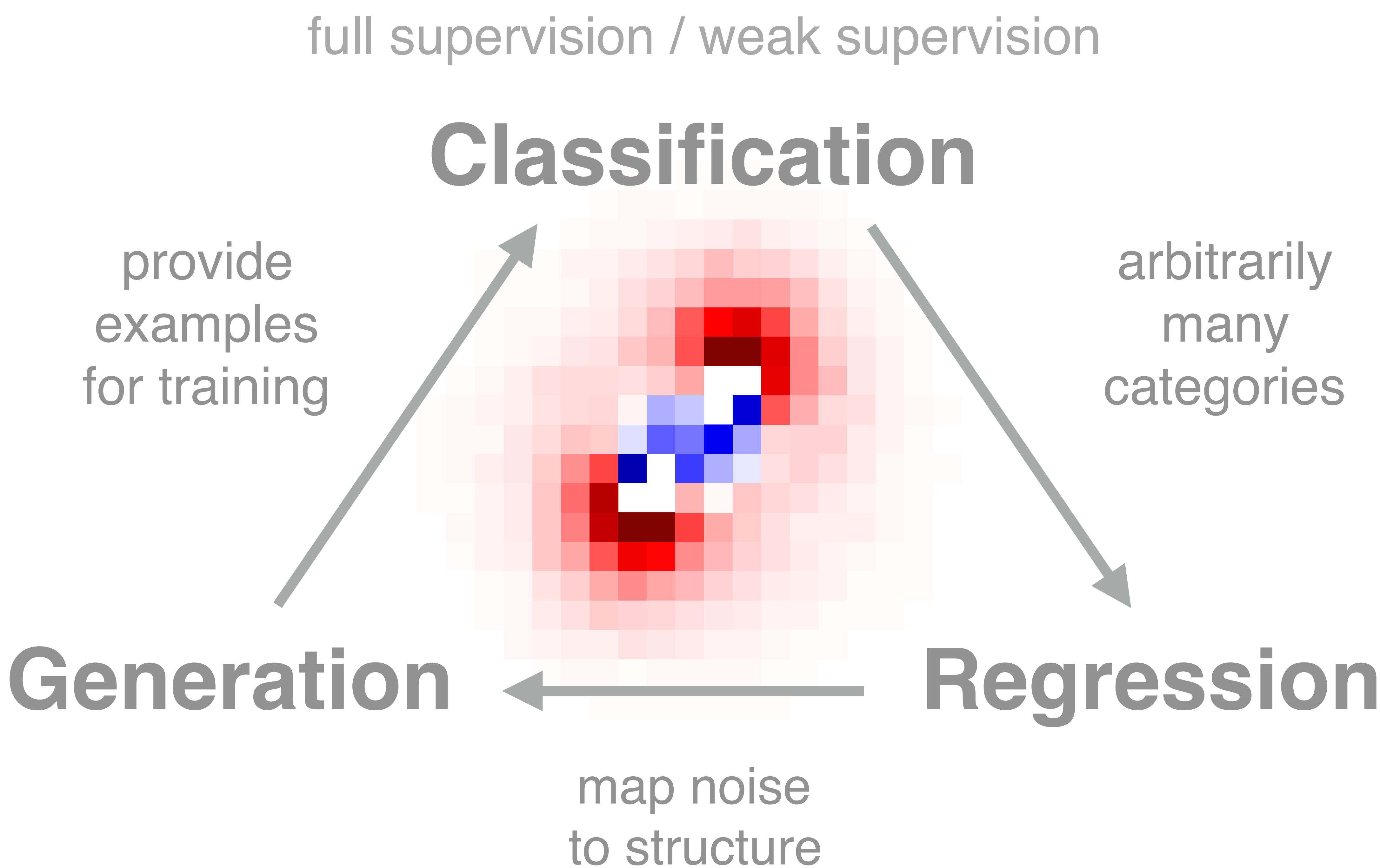


…today I will mostly talk about the "beyond" (anomaly detection), but I am happy to discuss other topics afterward.

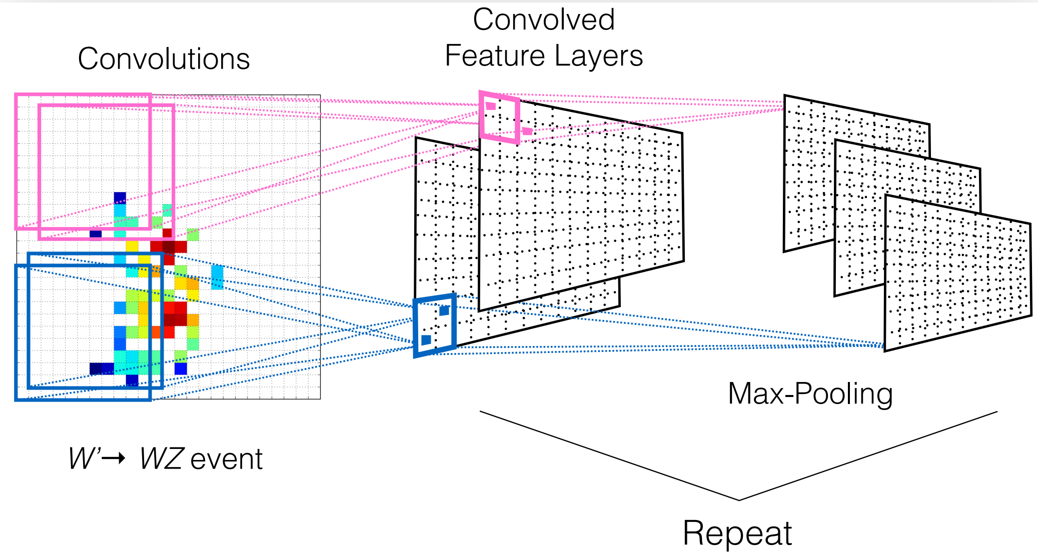
BERKELEY EXPERIMENTAL PARTICLE PHYSICS

DANCE 2019

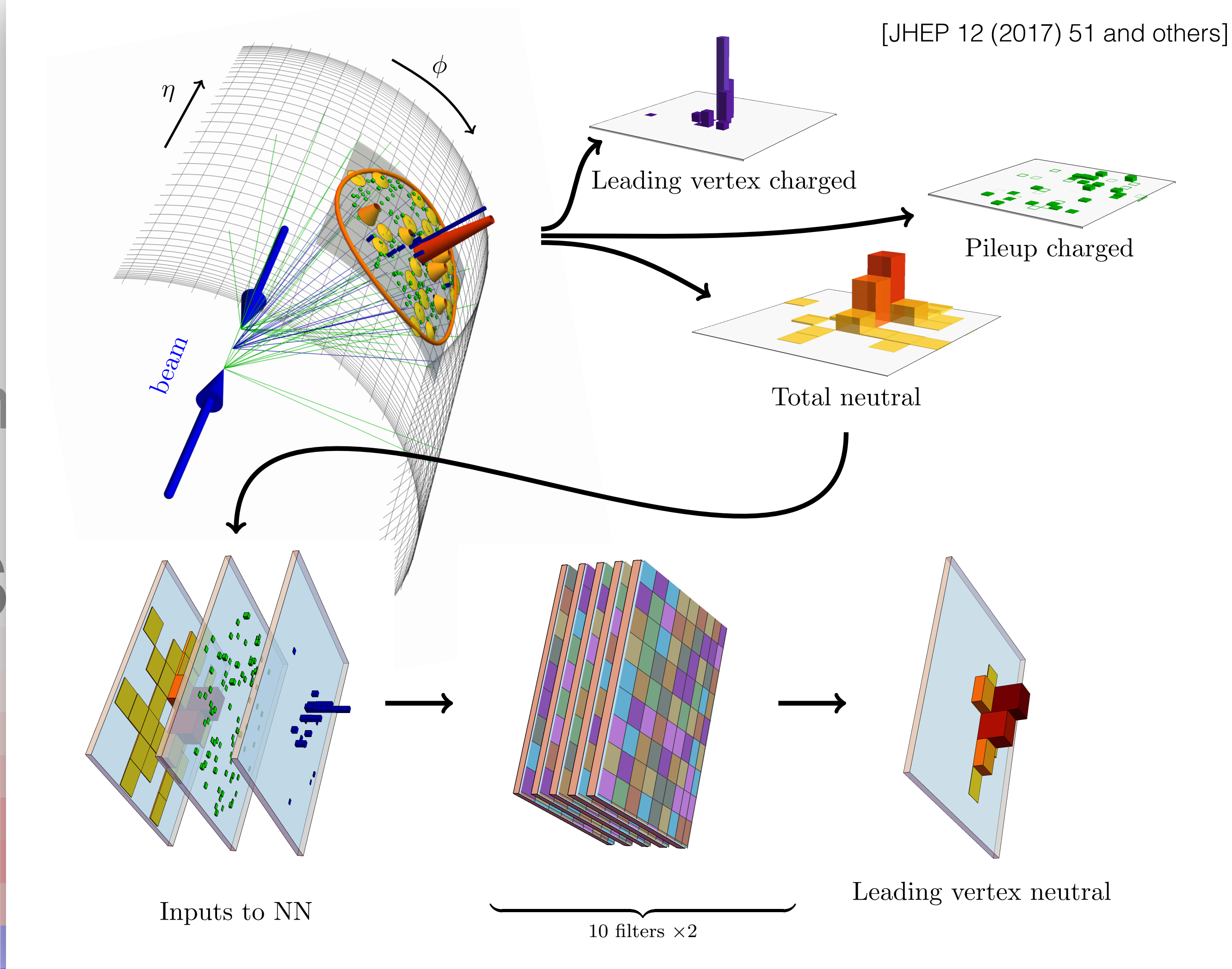
October 29, 2019


full supervision / weak supervision
Classification
provide examples for training
arbitrarily many categories
Generation
Regression
map noise to structure

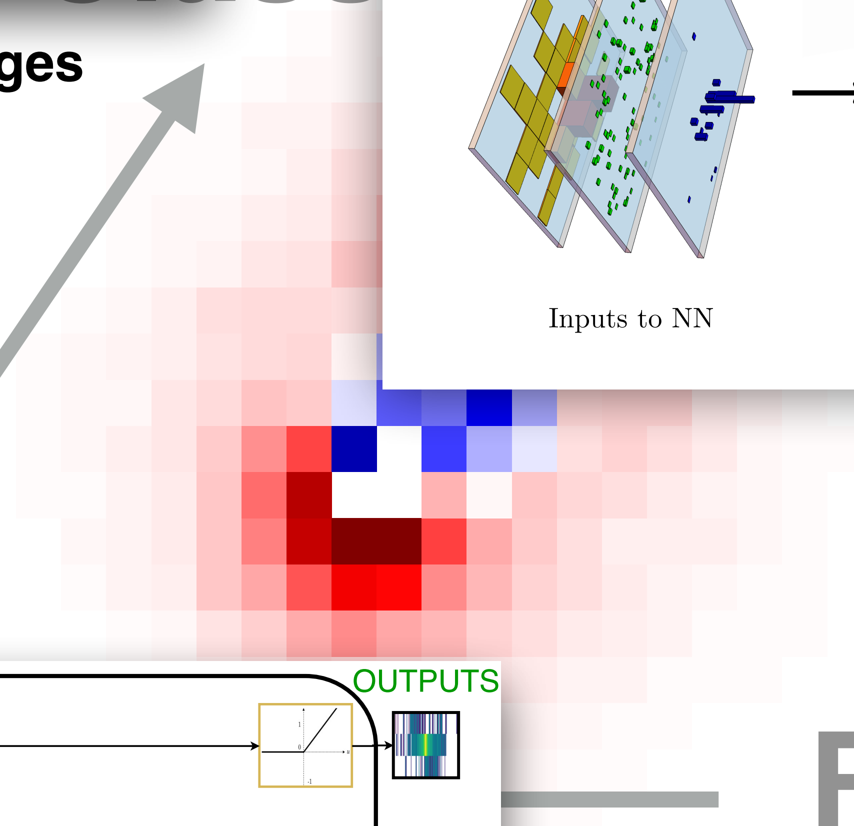
[JHEP 07 (2016) 069 and papers that cite it]



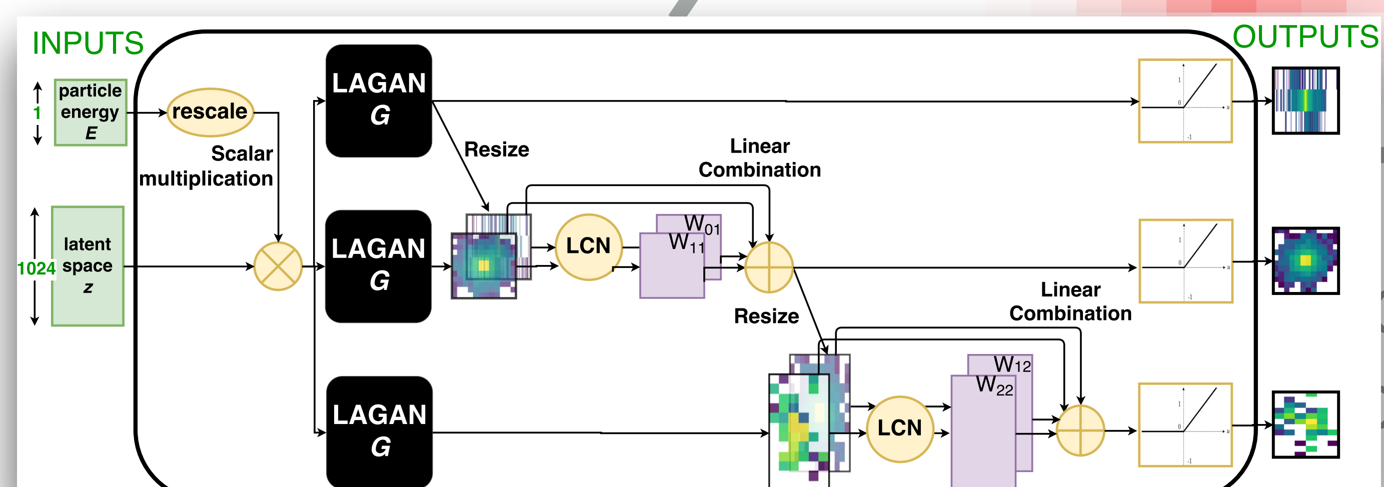
Representing detectors as images



Convolutional Neural Networks for Noise Mitigation



[PRL 120 (2018) 042003 and papers that cite it]



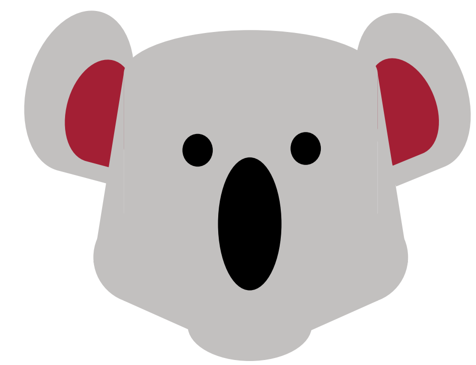
Generative Adversarial Networks to accelerate simulation

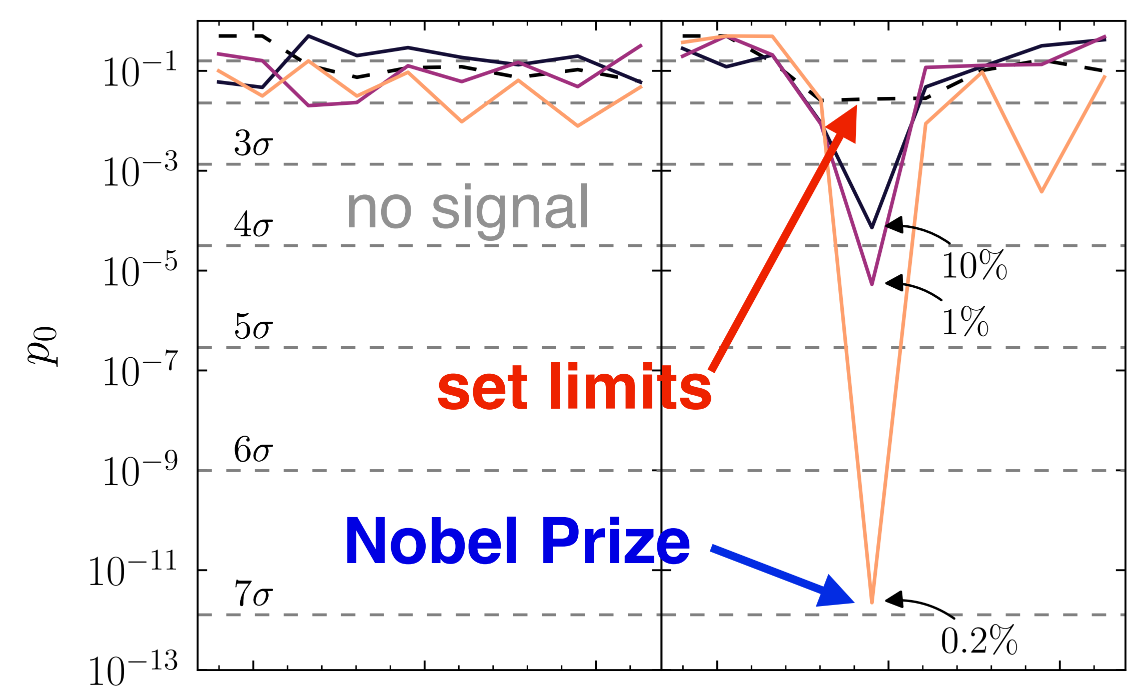
Regression

Many places where these are already part of collaboration workflows … many other places where this is still active work!

Outline for today

- Uncertainties in a NN-based analysis
- Searches at the LHC
- Learning without labels
- Model agnostic searches
- The future





Uncertainties

"But what are the uncertainties on the NN"?

- question asked by every review board

Uncertainties

"But what are the uncertainties on the NN"?

- question asked by every review board

- Before this can happen, need to better understand statistical and systematic properties of DNN based discriminators

3. Absence of rigorous treatment of statistical/systematic errors

(snippets from yesterday's slides)

Uncertainties for a NN-based analysis

Precision / Optimality: $\mathrm{NN(x)} \neq \frac{p_{\mathrm{true}}(x|\mathrm{S+B})}{p_{\mathrm{true}}(x|\mathrm{B})}$

limited training statistics

$p_{\mathrm{train}}(x) \neq p_{\mathrm{true}}(x)$

inaccurate training data

$\mathrm{NN(x)}|_{p_{\mathrm{true}}=p_{\mathrm{train}}} \neq \frac{p_{\mathrm{true}}(x|\mathrm{S+B})}{p_{\mathrm{true}}(x|\mathrm{B})}$

model/optimization flexibility

Statistical uncertainty | **Systematic uncertainty**

limited prediction statistics

$p_{\mathrm{prediction}}(x) \neq p_{\mathrm{true}}(x)$

inaccurate prediction data

Accuracy / Bias: $p_{\mathrm{prediction}}(\mathrm{NN}) \neq p_{\mathrm{true}}(\mathrm{NN})$

[1909.03081]

High-dimensional Uncertainty

One word of caution: current paradigm for uncertainties may be too naive for hypervariate analysis!

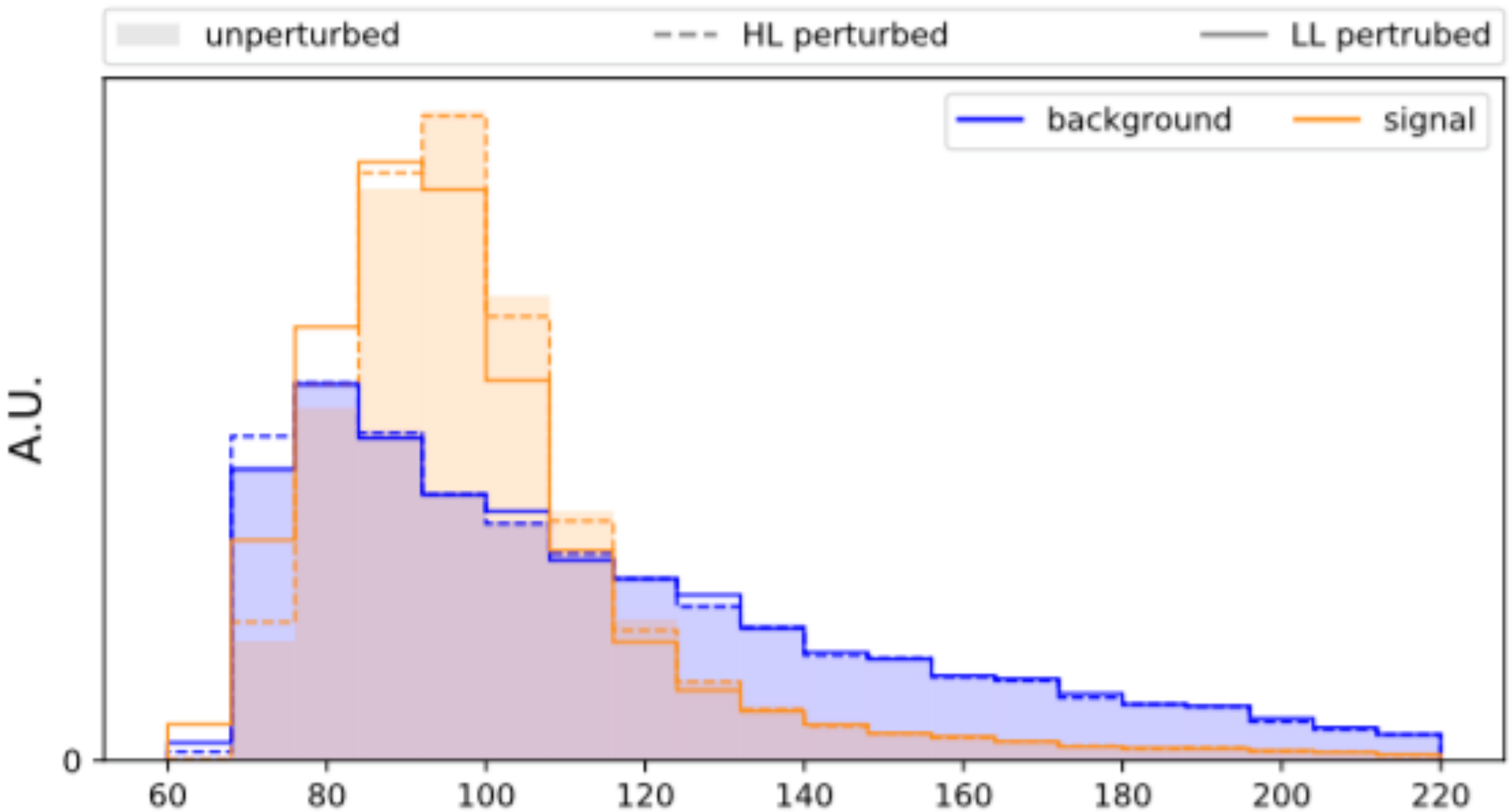
(truly end-to-end)

e.g. for some uncertainties, we often compare two different models - one nuisance parameter.

High-dimensional Uncertainty

One word of caution: current paradigm for uncertainties may be too naive for hypervariate analysis!

e.g. for some uncertainties, we often compare two different models - one nuisance parameter.



Some observable we want to validate

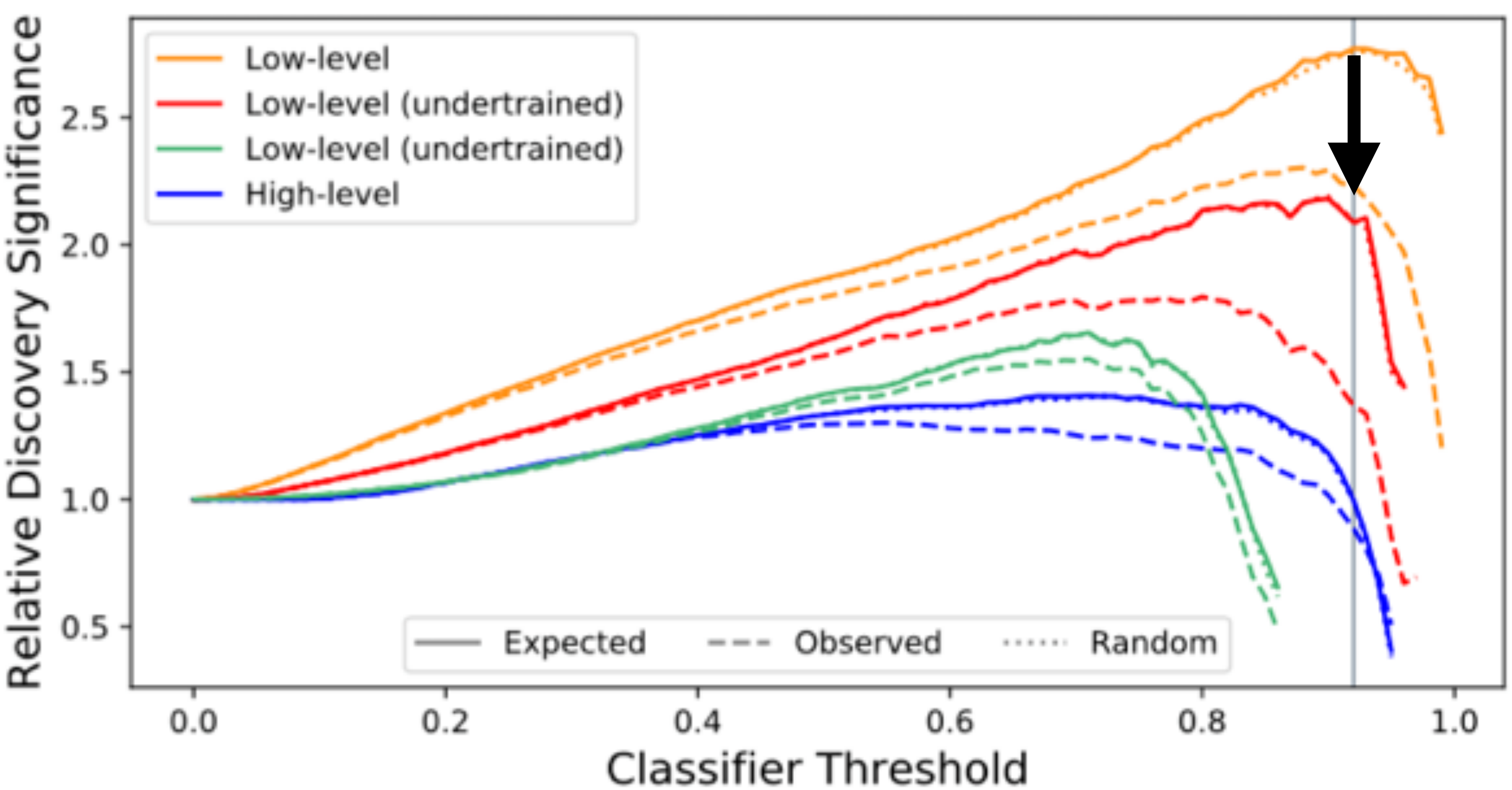
Borrowing ideas from **AI safety**, one can show that small perturbations can make big changes in NN output while preserving "control region" performance.

[1910.08606]

High-dimensional Uncertainty

One word of caution: current paradigm for uncertainties may be too naive for hypervariate analysis!

e.g. for some uncertainties, we often compare two different models - one nuisance parameter.



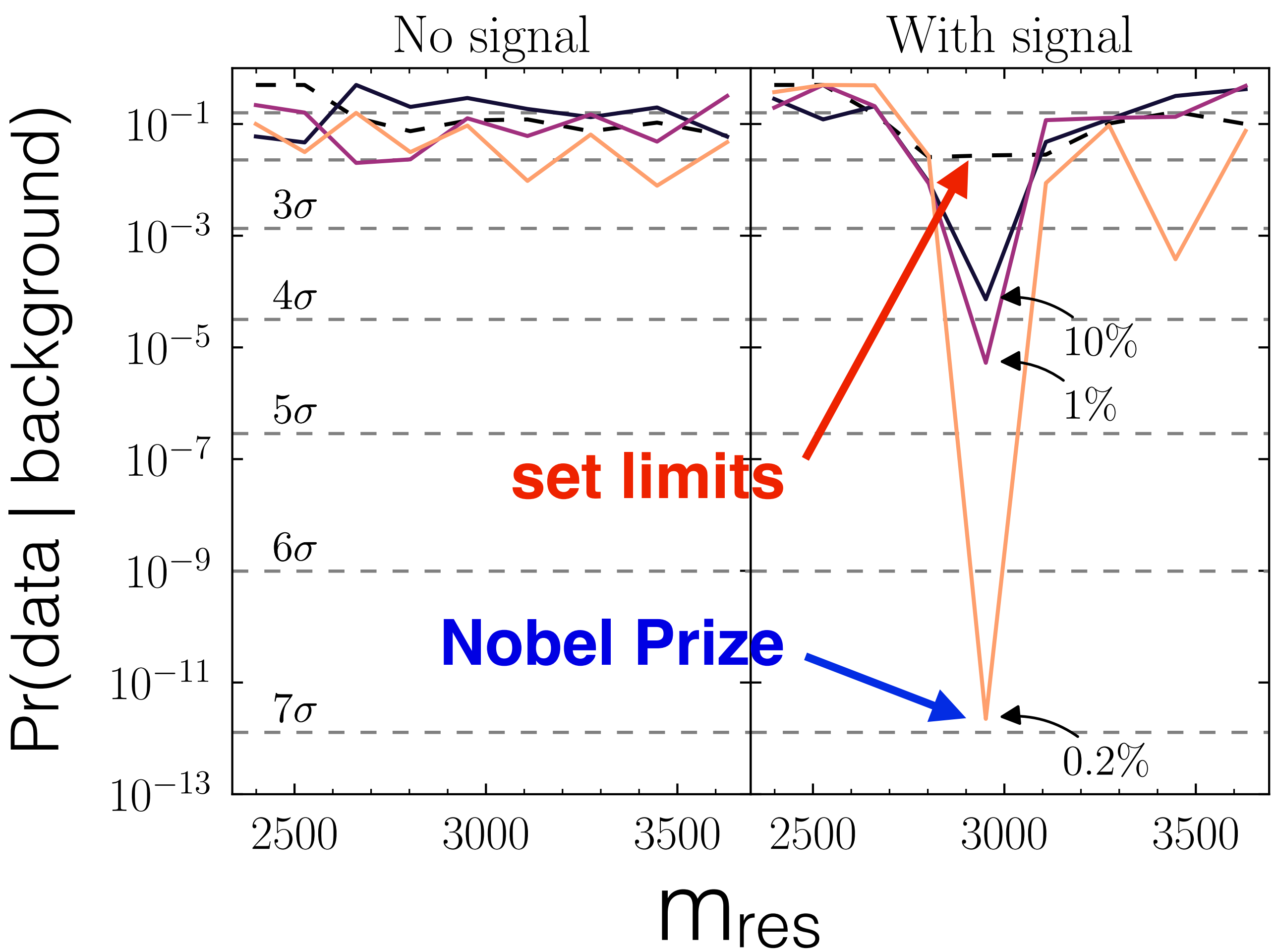
Borrowing ideas from **AI safety**, one can show that small perturbations can make big changes in NN output while preserving “control region” performance.

[1910.08606]

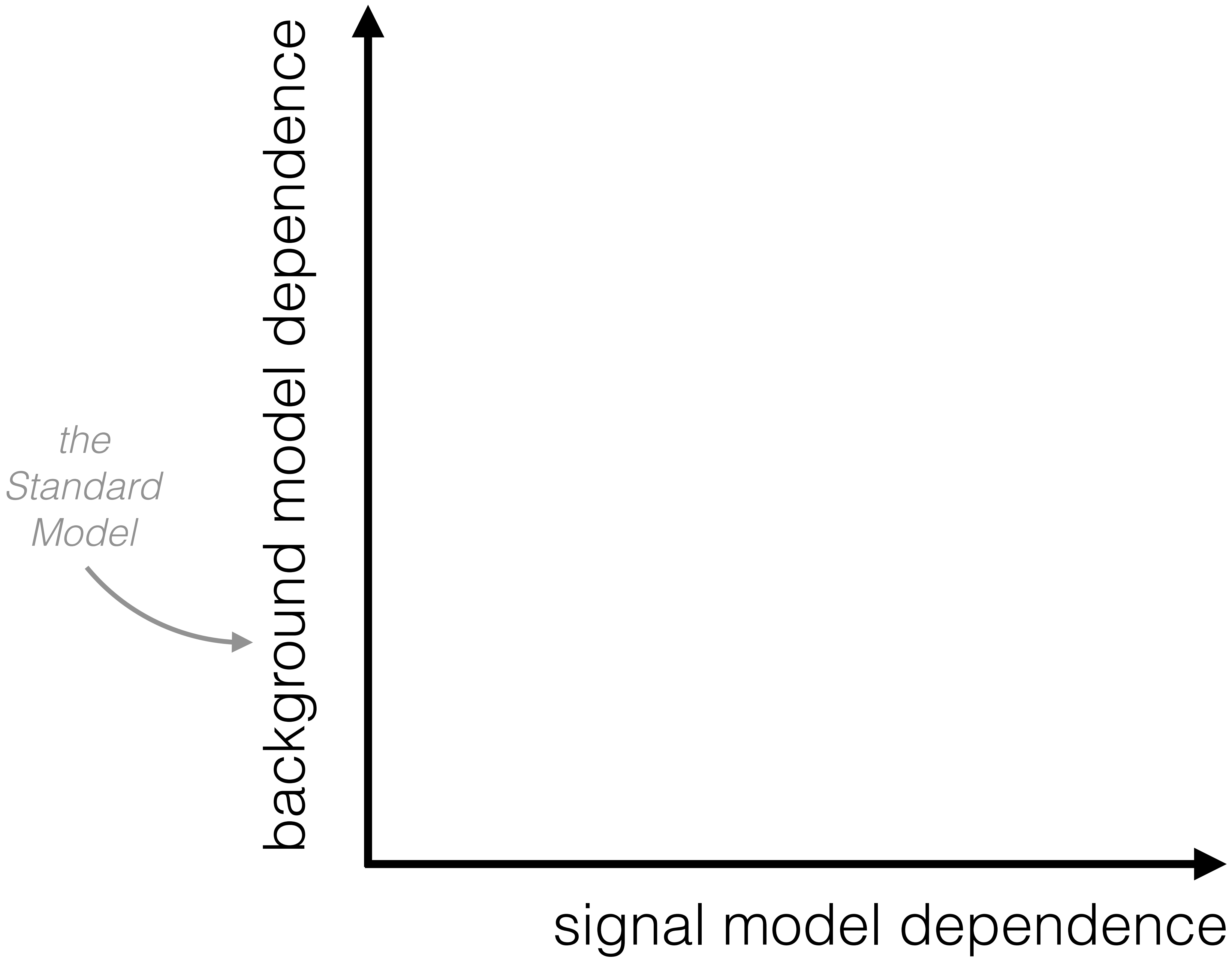
How to get around this?

Work hard to understand the true nuisance parameters in the hypervariate parameter space.

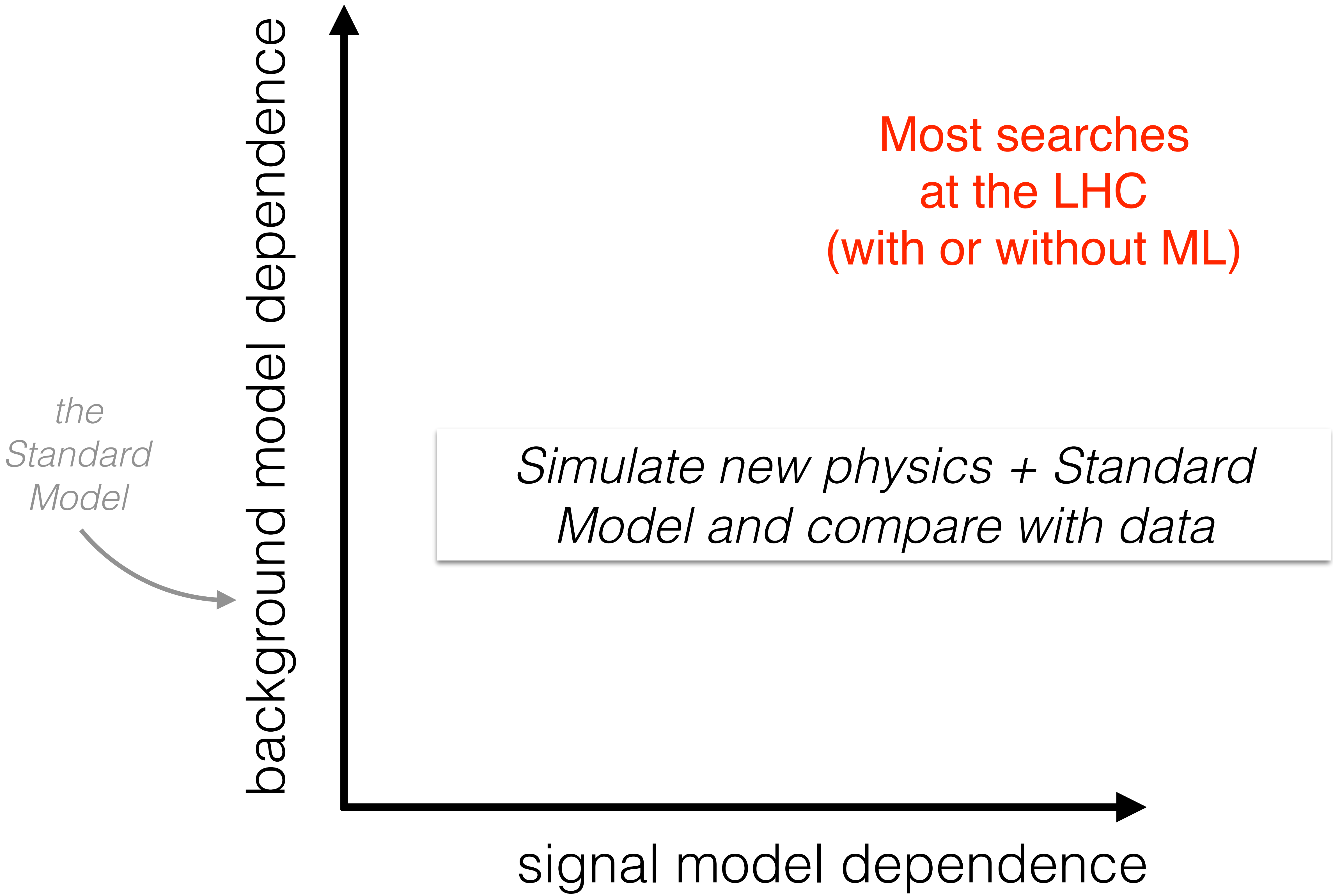
Don't use simulation!
(focus for the rest of the talk though not always possible!)



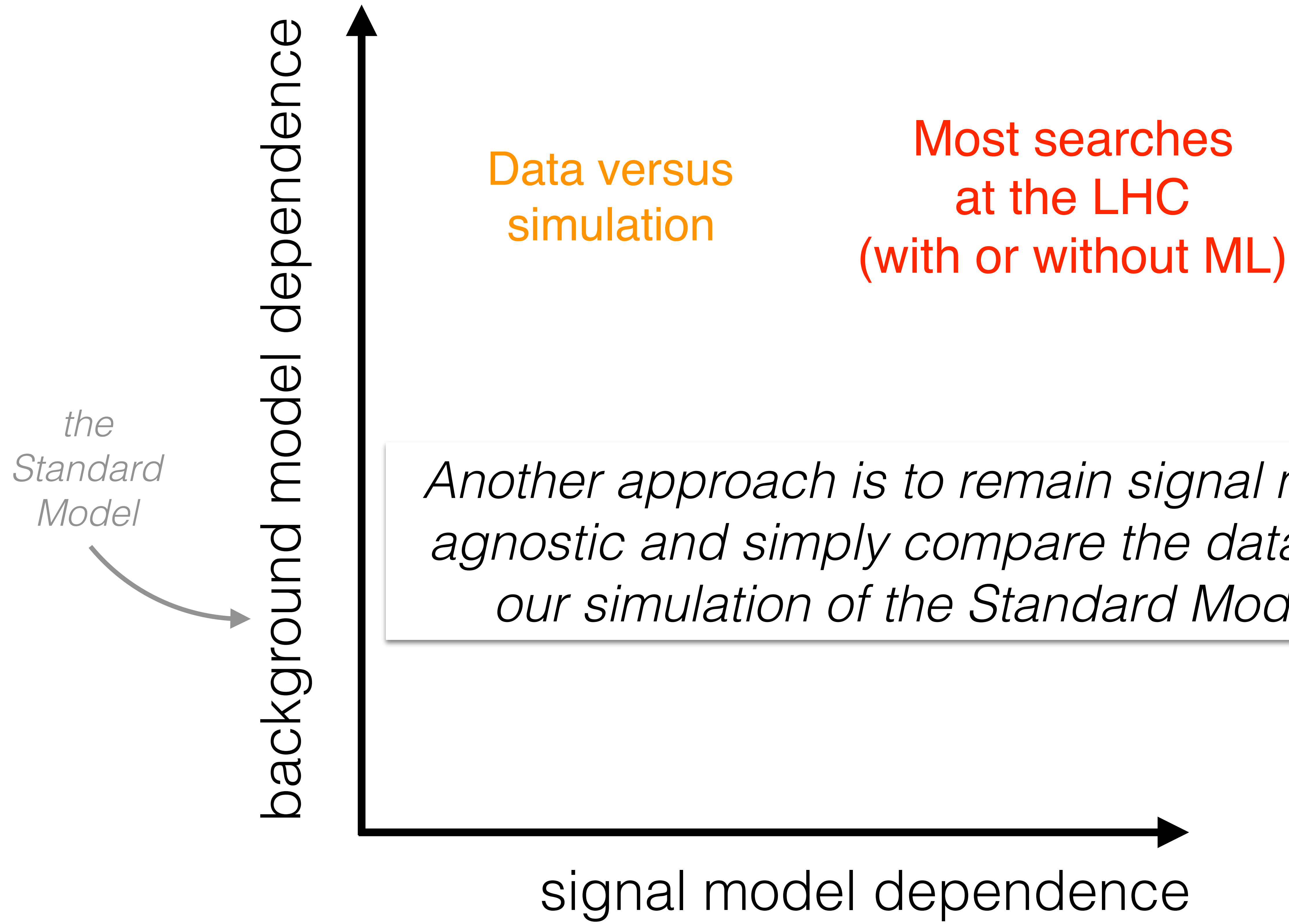
Searching for new particles / forces



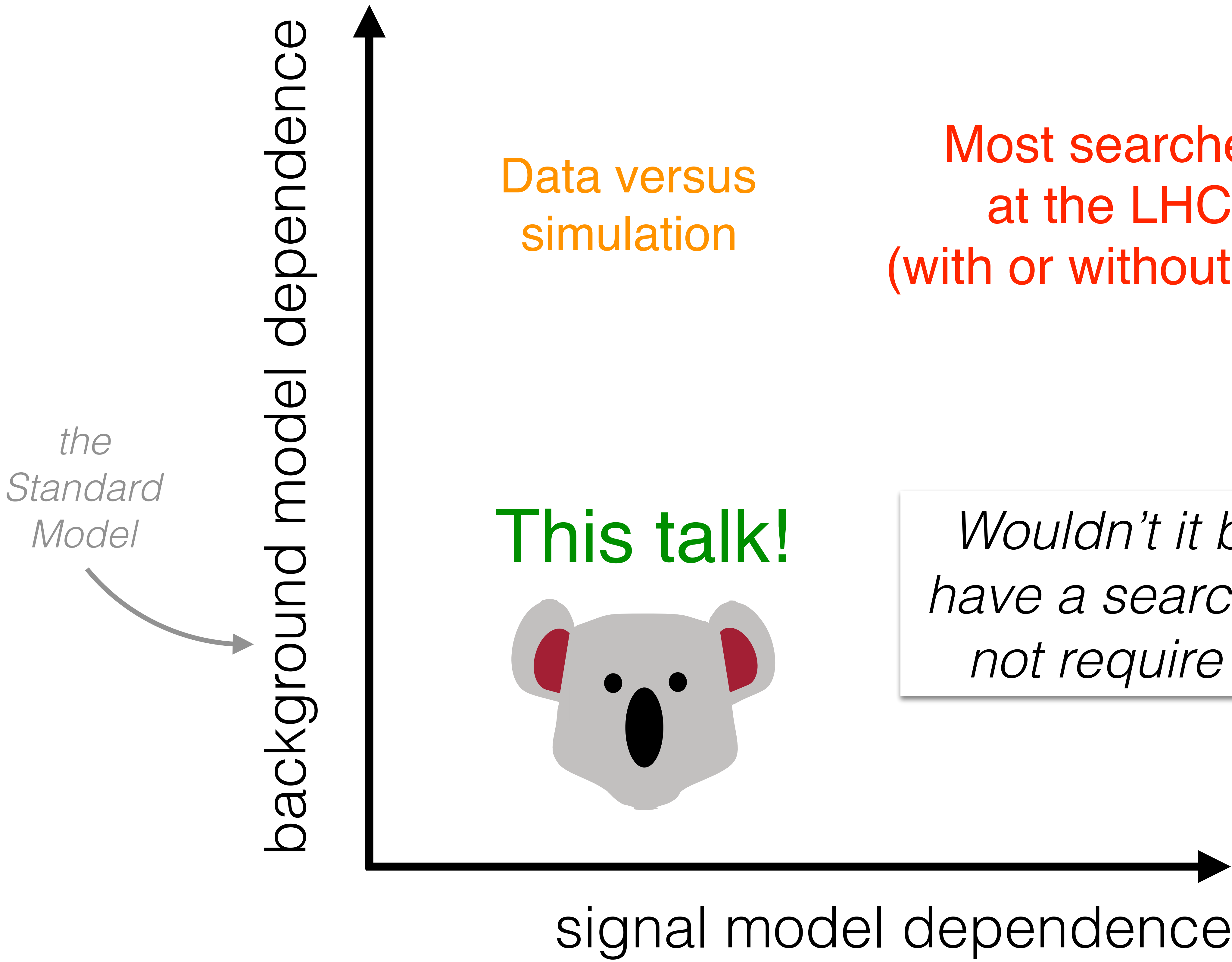
Searching for new particles / forces



Searching for new particles / forces



Searching for new particles / forces



Wouldn't it be great to have a search that does not require a model?

Searching for new particles / forces

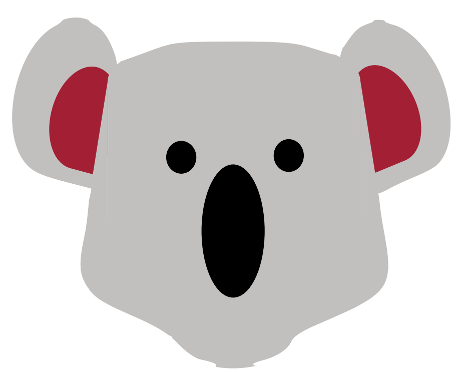
background model dependence

the
Standard
Model

The question then becomes:

how to train a classifier directly on data, without a particular signal model in mind?

This talk!

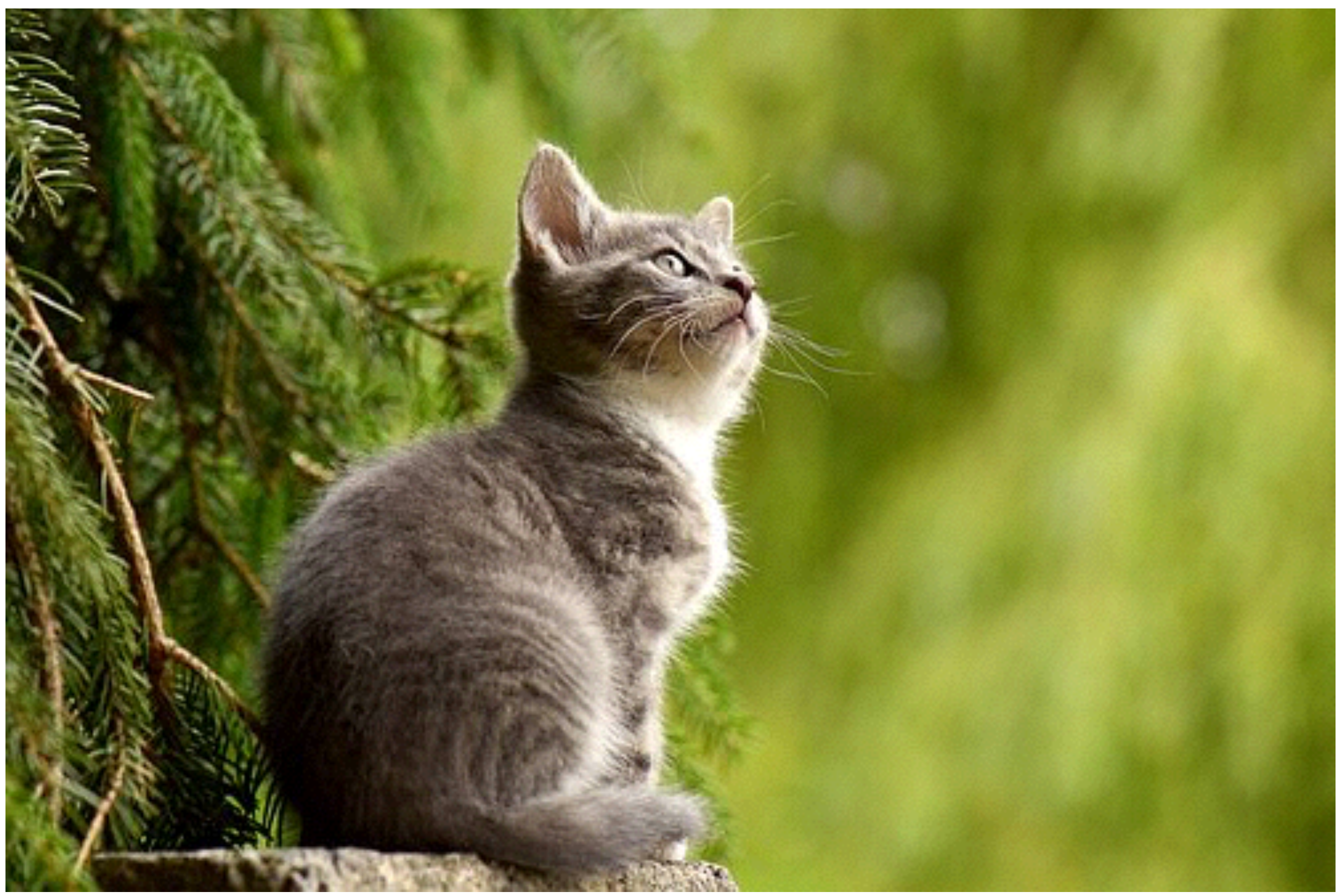


Wouldn't it be great to have a search that does not require a model?

signal model dependence

What is the problem?

Why can't I just pay some physicists to label events and then train a neural network using those labels?



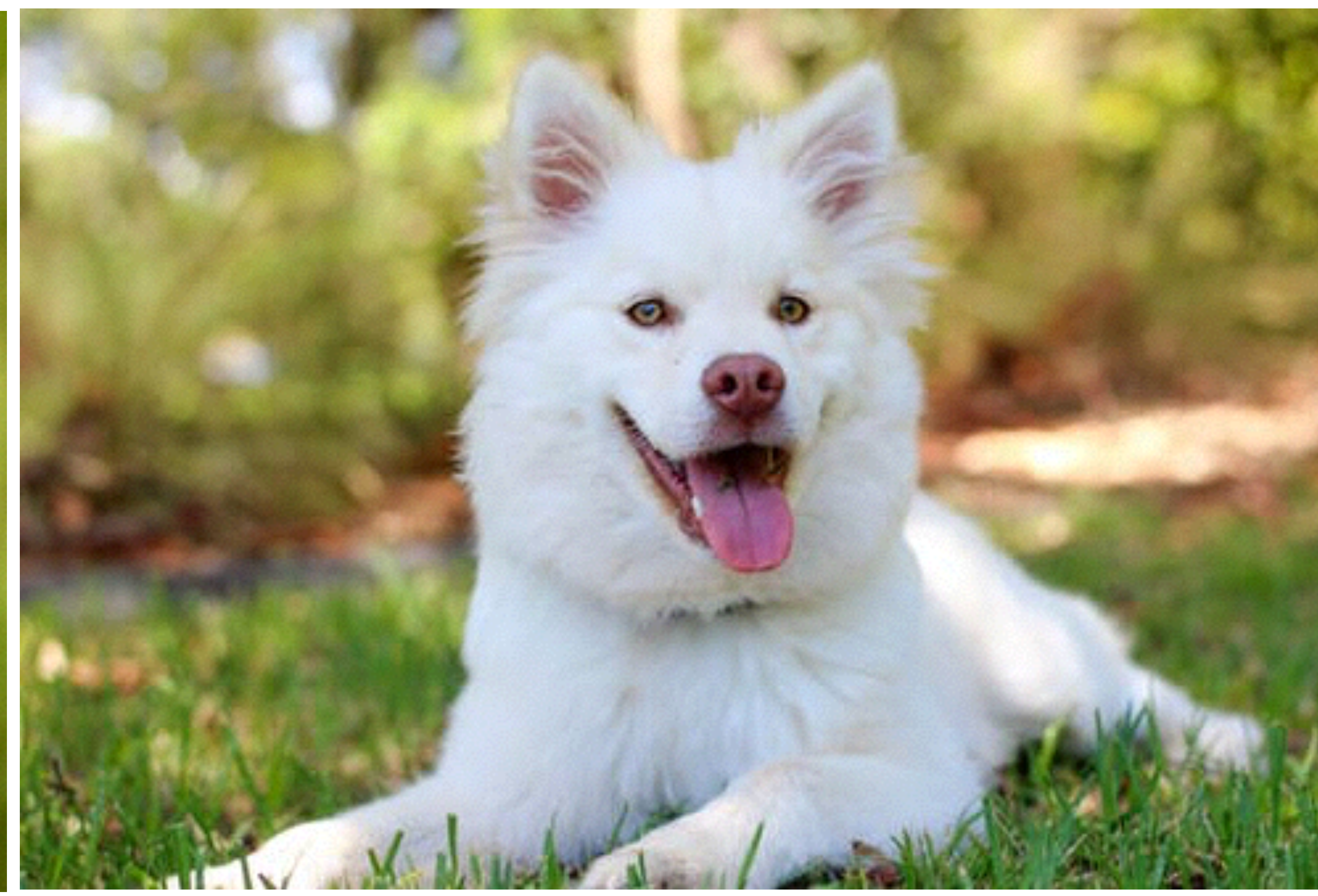
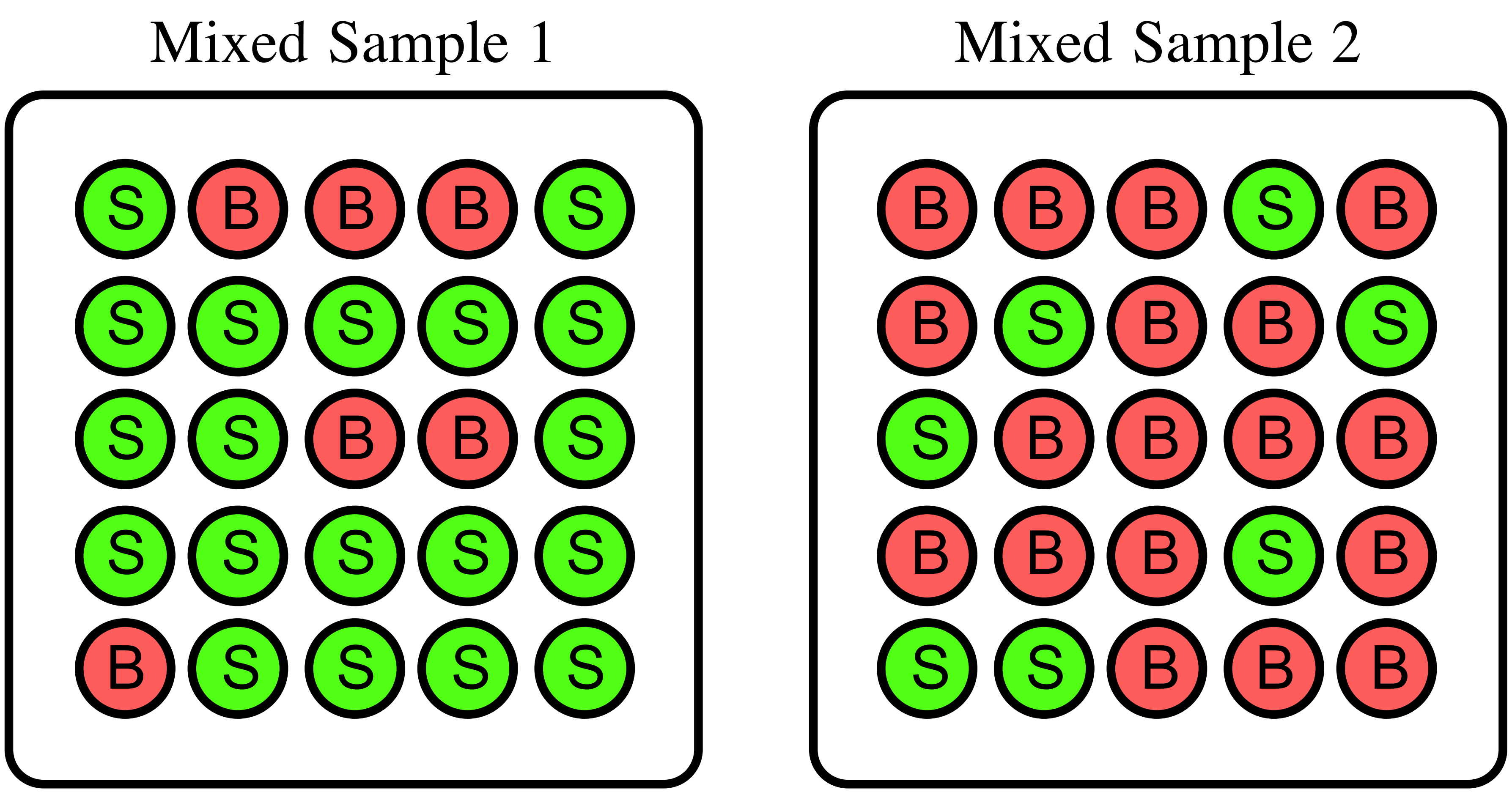


Image credit: pixabay.com

Answer: this is not cats-versus-dogs … thanks to quantum mechanics it is **not possible to know** what happened.

What is the problem?

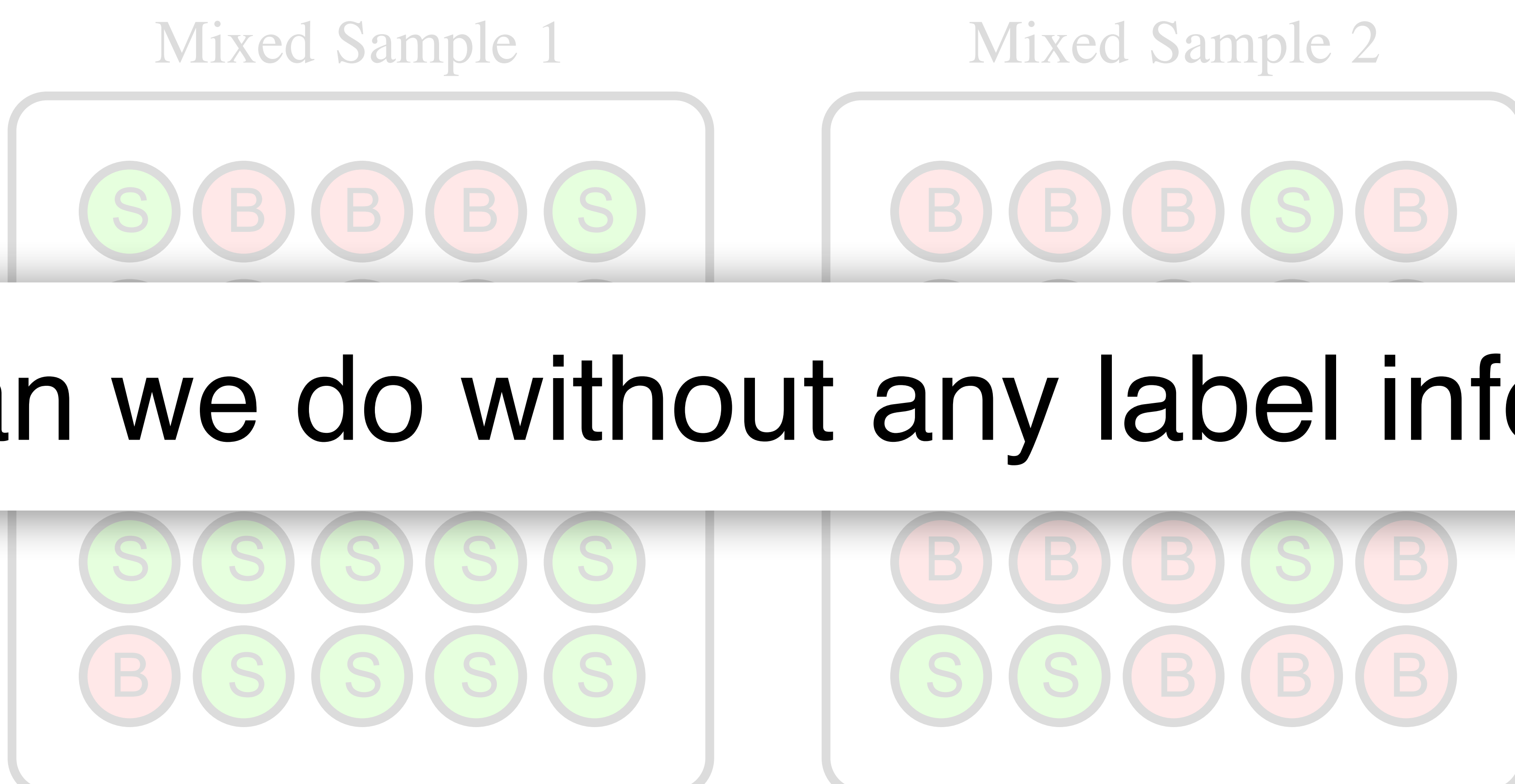
The data are unlabeled and in the best case, come to us as mixtures of two classes (“signal” and “background”).



(we don’t get to observe the color of the circles)

What is the problem?

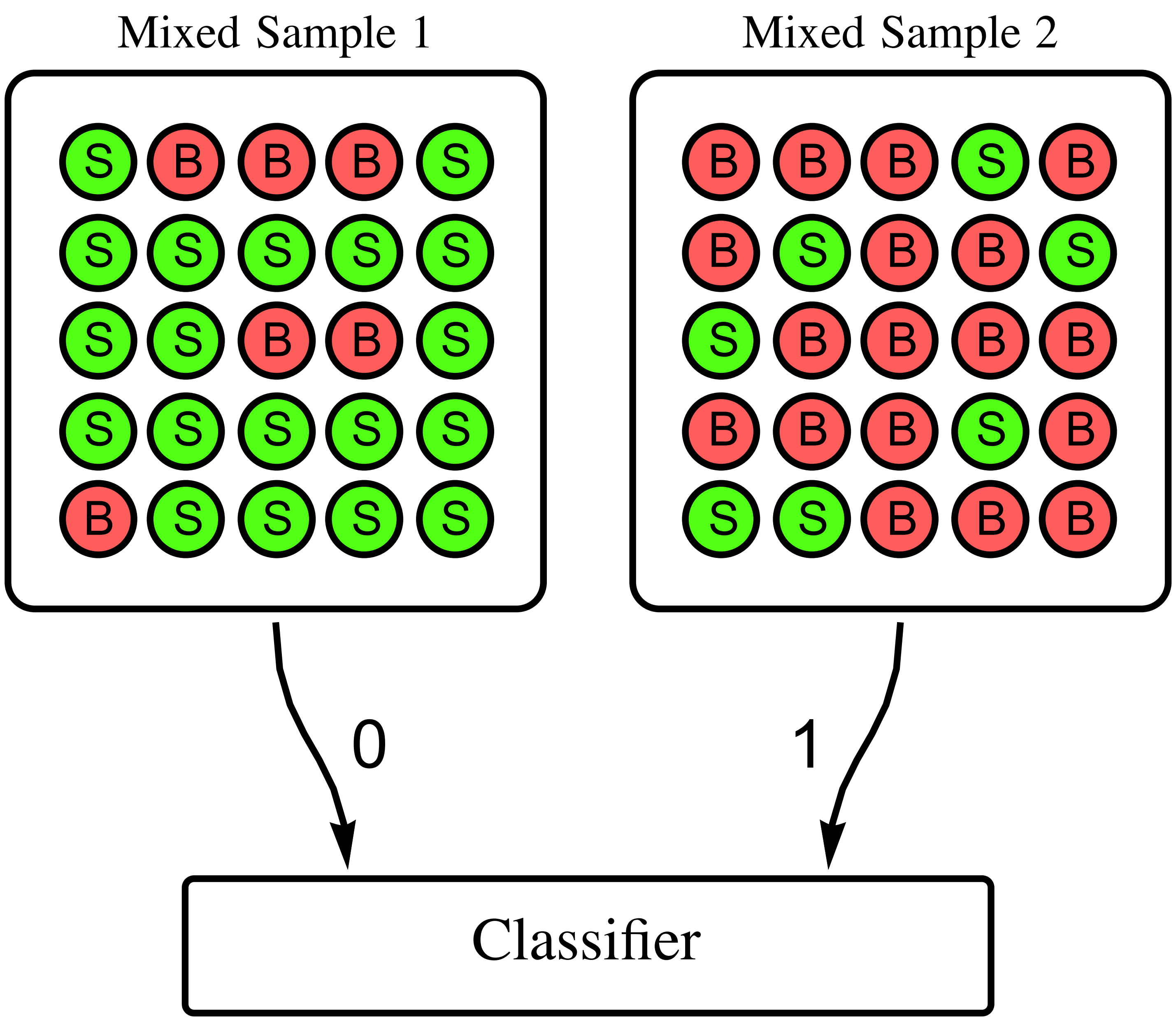
The data are unlabeled and in the best case, come to us as mixtures of two classes (“signal” and “background”).



Can we do without any label info?

(we don’t get to observe the color of the circles)

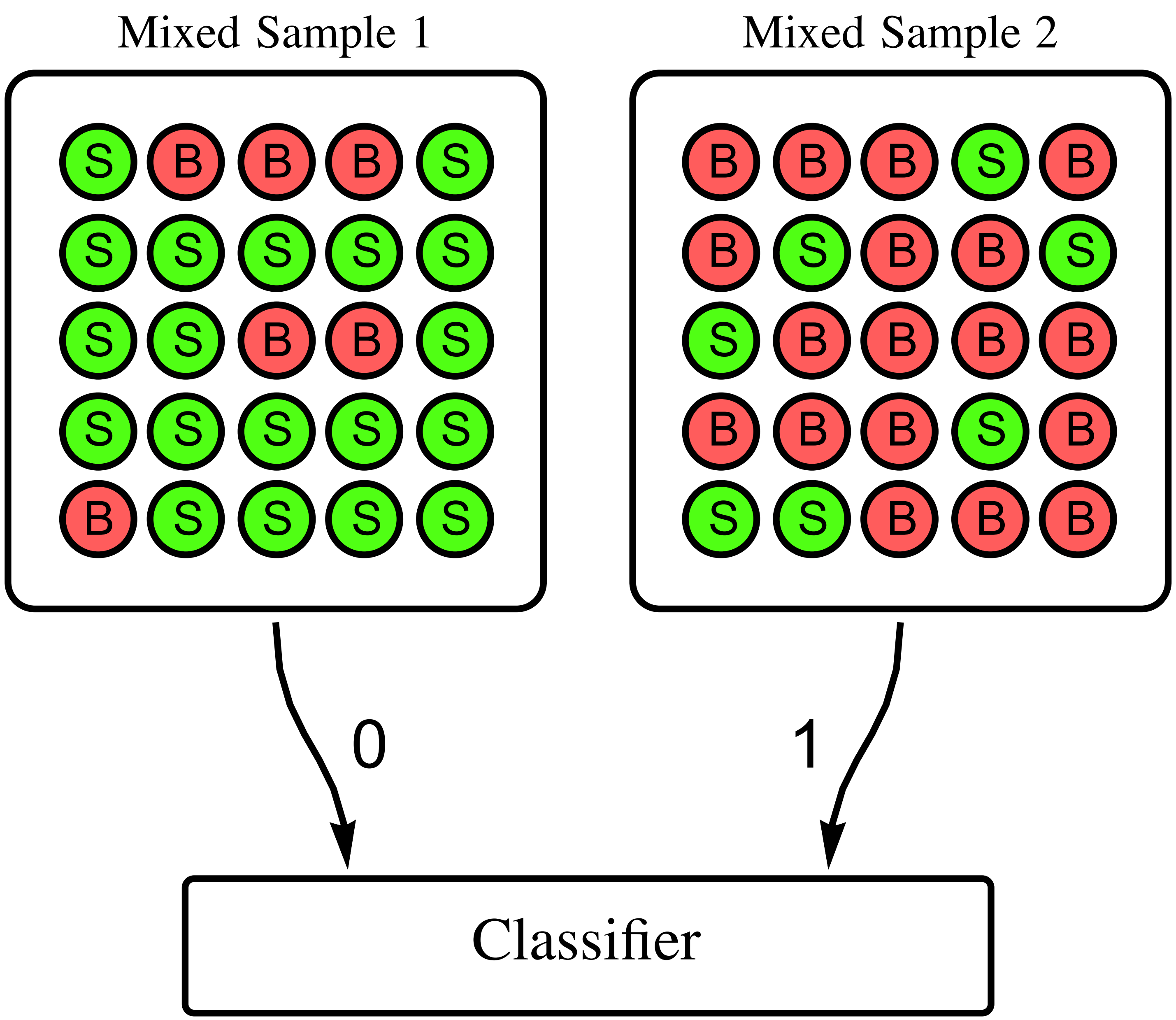
Classification Without Labels



Yes !

[Komiske, Metodiev, **BPN**, Schwartz, PRD 98 (2018) 011502]
[Cohen, Freytsis, Ostdiek, JHEP 02 (2018) 034]
[Metodiev, **BPN**, Thaler, JHEP 10 (2017) 51]
[Dery, **BPN**, Rubbo, Schwartzman, JHEP 05 (2017) 145]

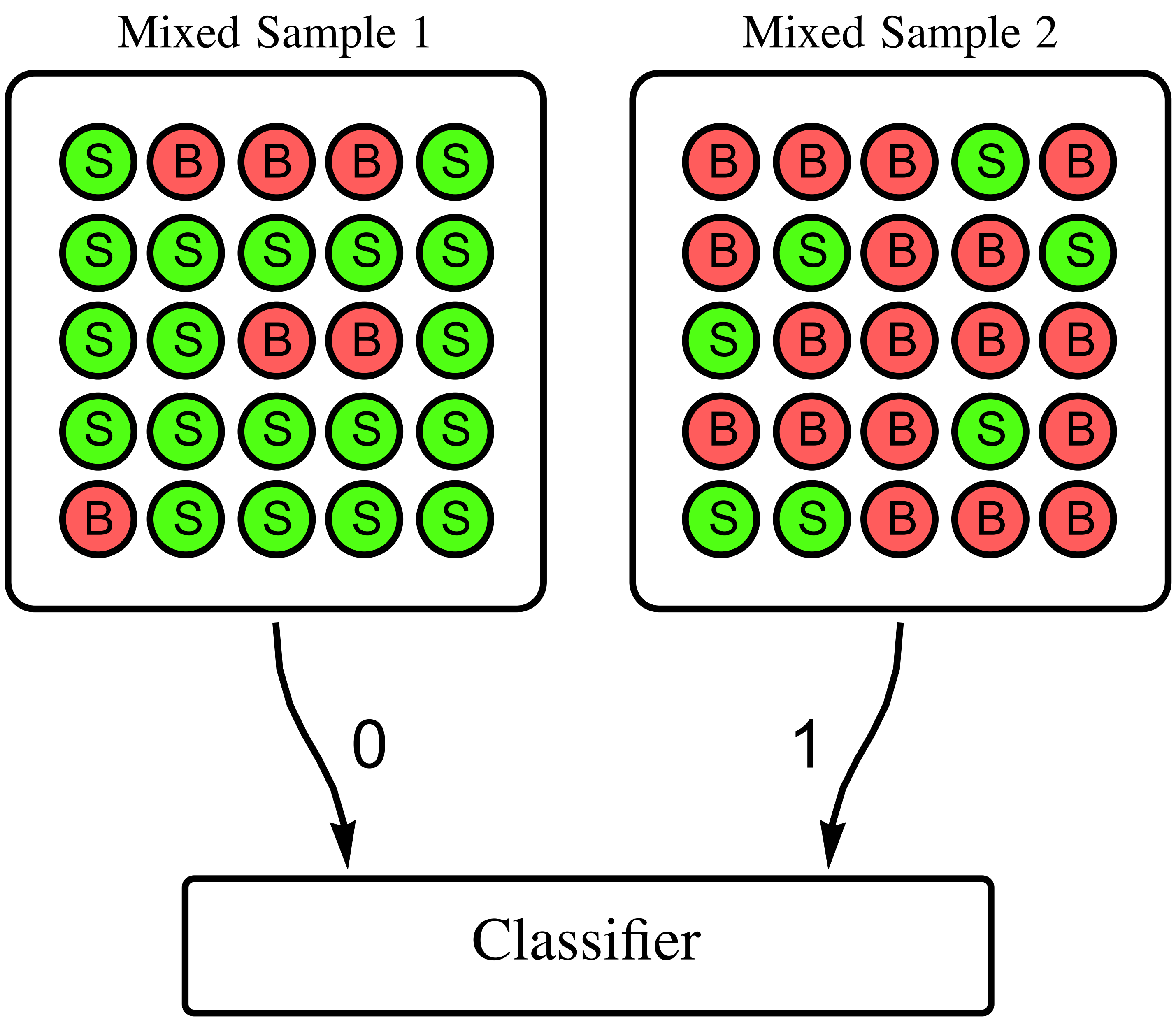
Classification Without Labels



Yes !

[Komiske, Metodiev, **BPN**, Schwartz, PRD 98 (2018) 011502]
[Cohen, Freytsis, Ostdiek, JHEP 02 (2018) 034]
[Metodiev, **BPN**, Thaler, JHEP 10 (2017) 51]
[Dery, **BPN**, Rubbo, Schwartzman, JHEP 05 (2017) 145]

Classification Without Labels

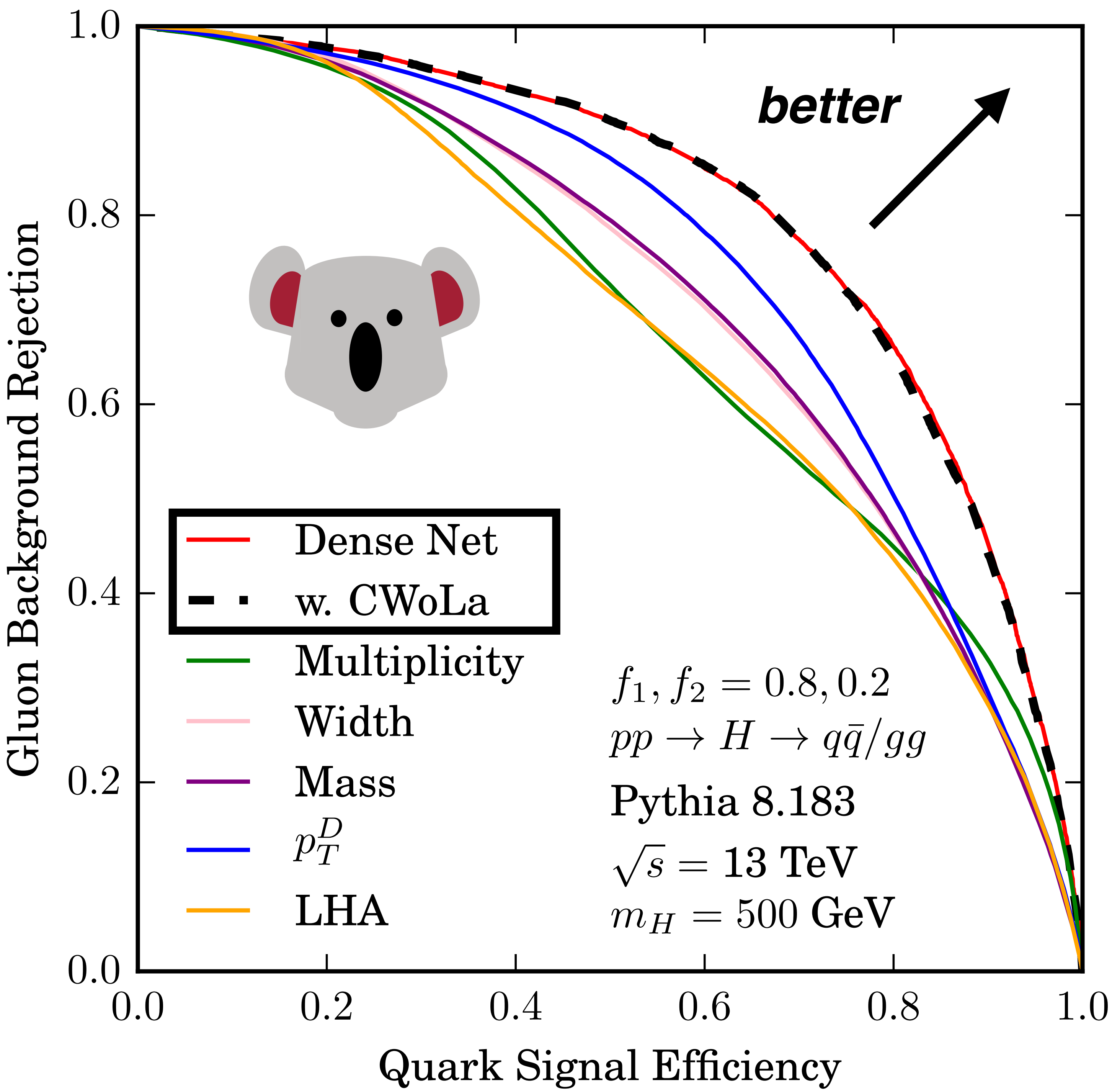


Yes !

One can show that this procedure asymptotically converges to the optimal classifier (with labels).

[Komiske, Metodiev, **BPN**, Schwartz, PRD 98 (2018) 011502]
[Cohen, Freytsis, Ostdiek, JHEP 02 (2018) 034]
[Metodiev, **BPN**, Thaler, JHEP 10 (2017) 51]
[Dery, **BPN**, Rubbo, Schwartzman, JHEP 05 (2017) 145]

Classification Without Labels



In practice, it also seems to work well, often approaching the case with 100% label information (fully supervised)

[Metodiev, **BPN**, Thaler, JHEP 10 (2017) 51]

What if we know even less?

There are many uses for CWoLa when you know the two classes. What if you don't - can we use CWoLa to **hunt for new** particles without a signal model in mind?

What if we know even less?

There are many uses for CWoLa when you know the two classes. What if you don't - can we use CWoLa to **hunt for new** particles without a signal model in mind?

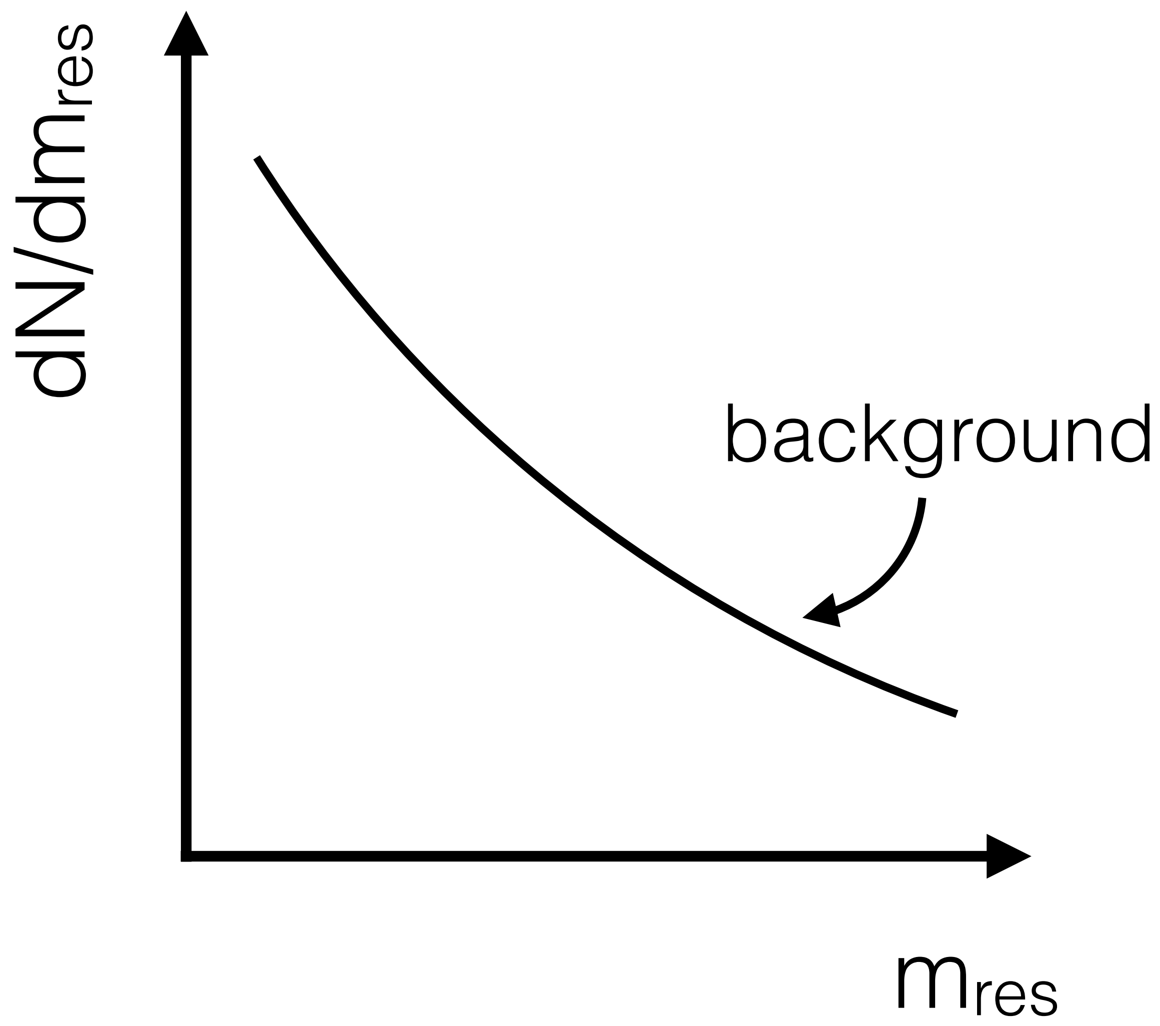
Yes!
…CWoLa hunting

[J. Collins, K. Howe, **BPN** PRL 121 (2018) 241803]

[J. Collins, K. Howe, **BPN** PRD 99 (2019) 014038]

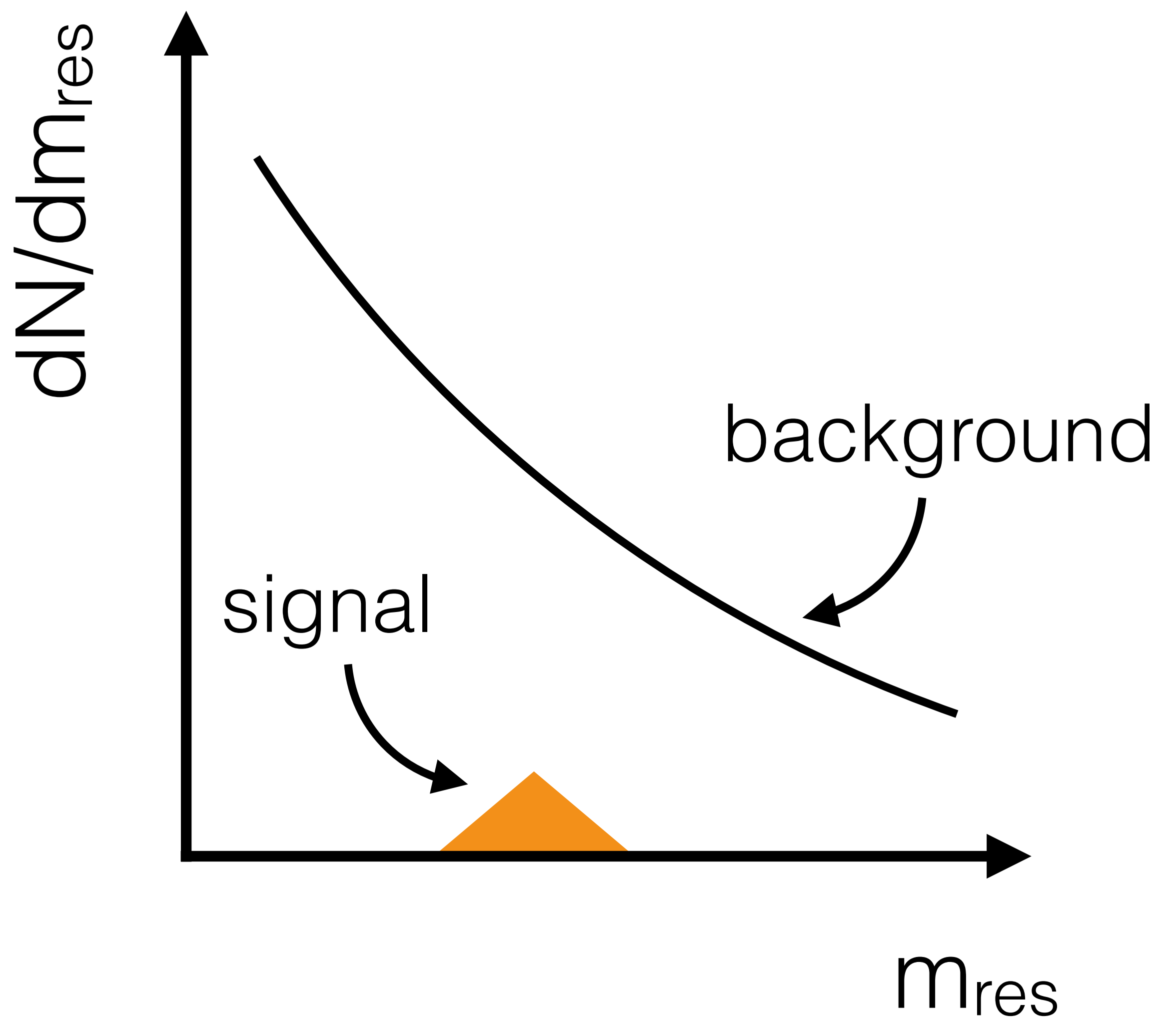
*Image from *The Courier Mail*. Koala is actually being freed - I do not condone violence against these animals!

CWoLa Hunting for new particles



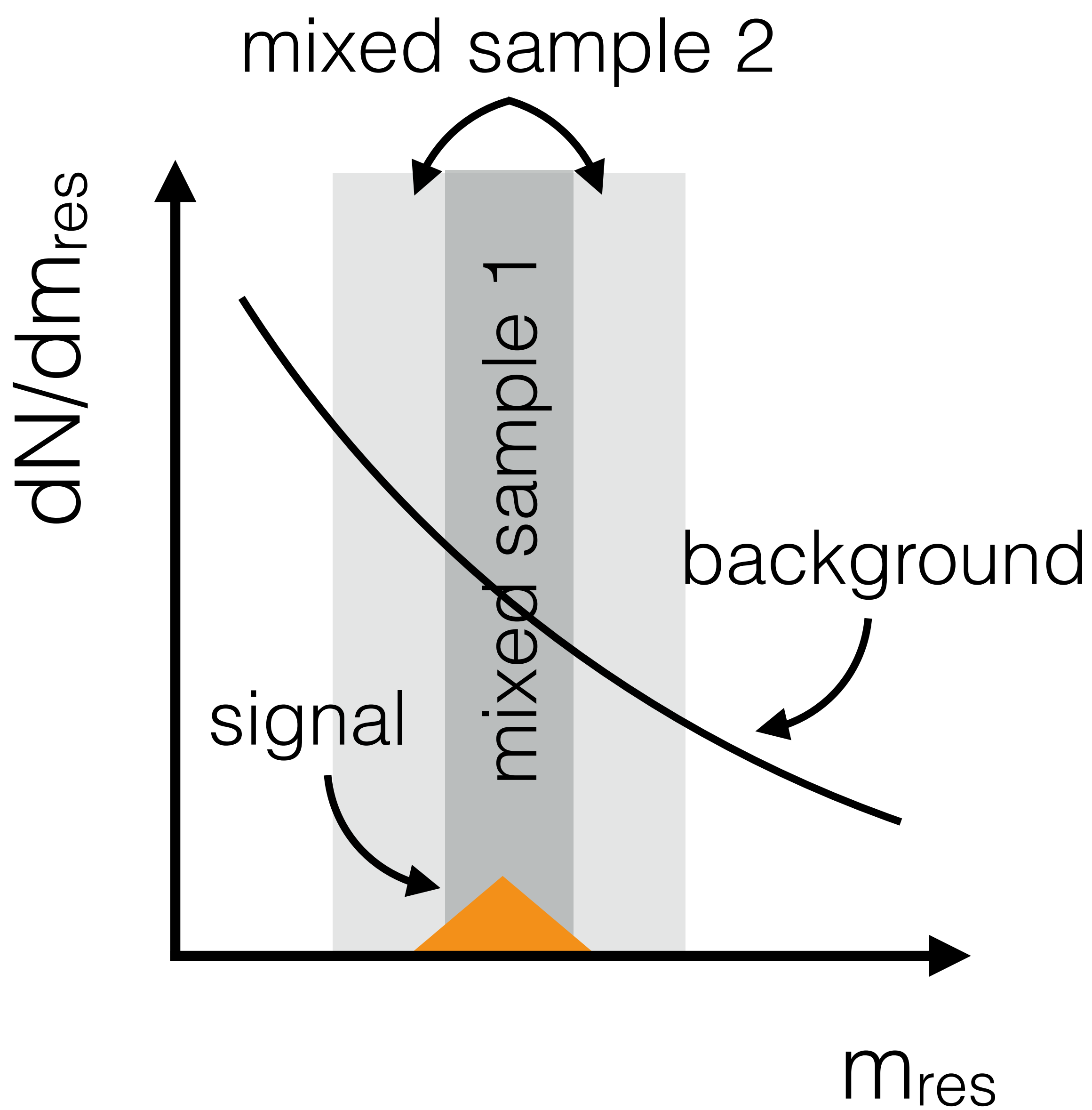
Assumption: there is a feature that we know about where the background is smooth and the signal (if it exists) is localized.

CWoLa Hunting for new particles



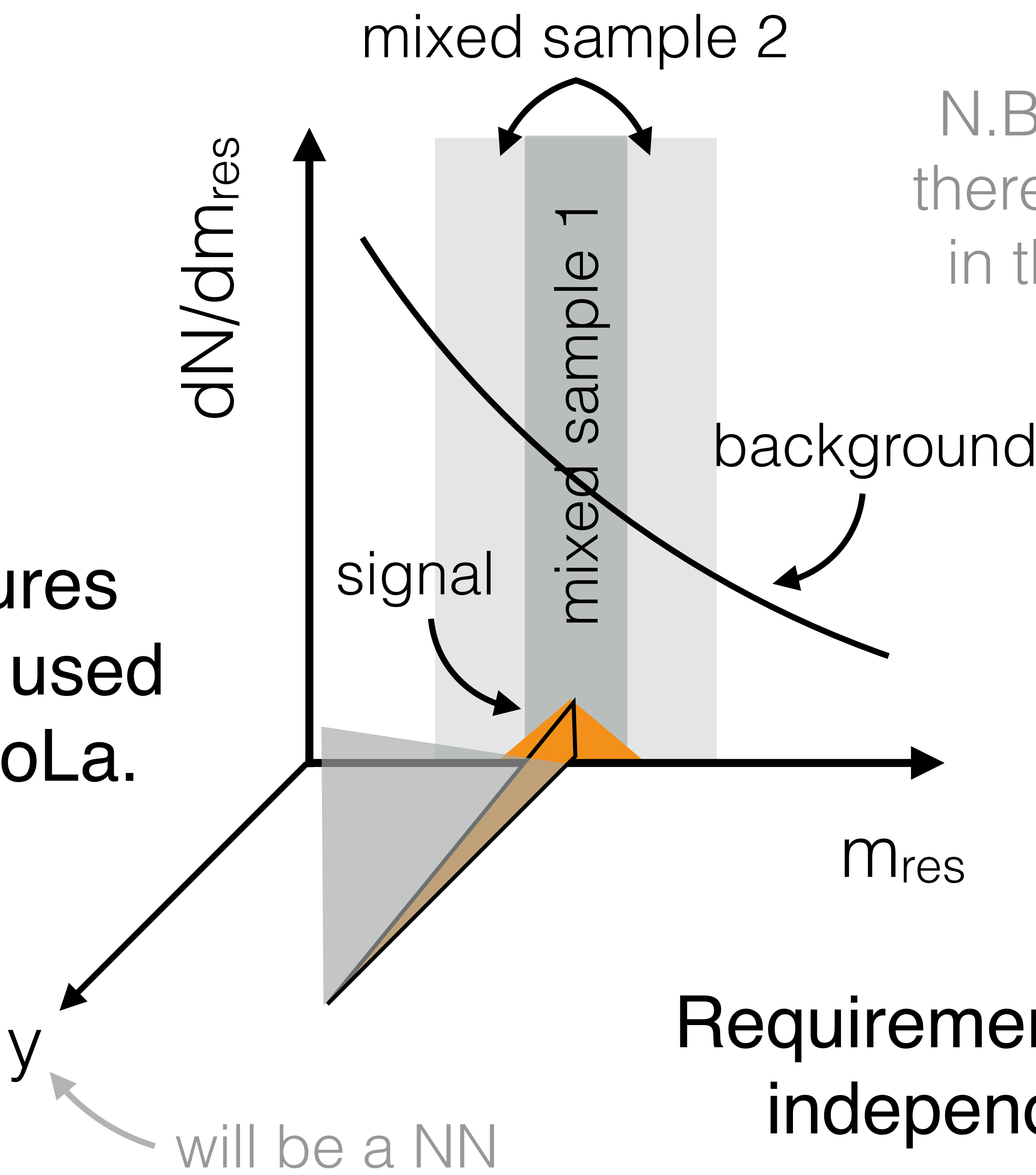
Assumption: there is a feature that we know about where the background is smooth and the signal (if it exists) is localized.

CWoLa Hunting for new particles



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

CWoLa Hunting for new particles

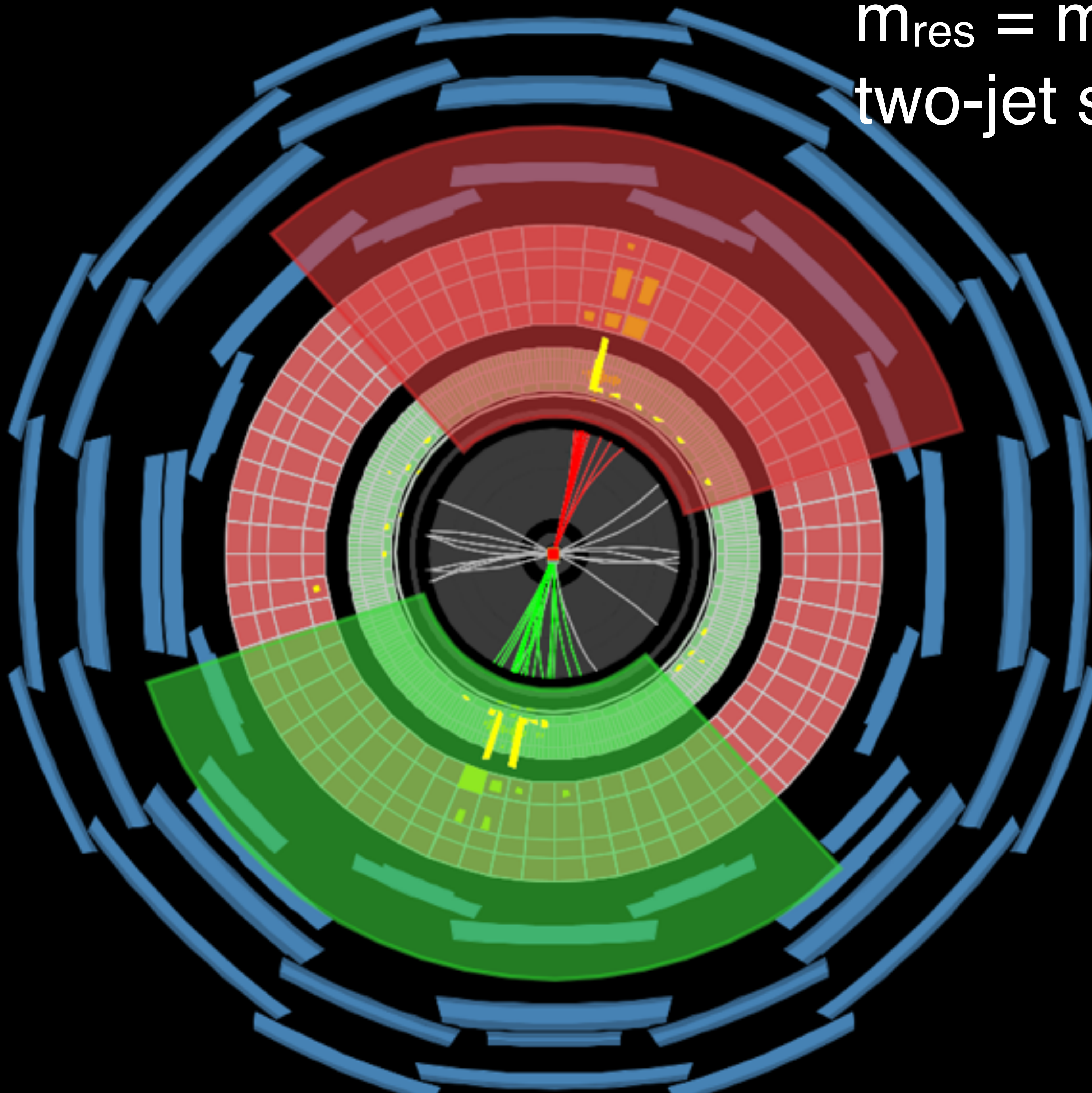


N.B. it is okay that there is some signal in the sidebands!

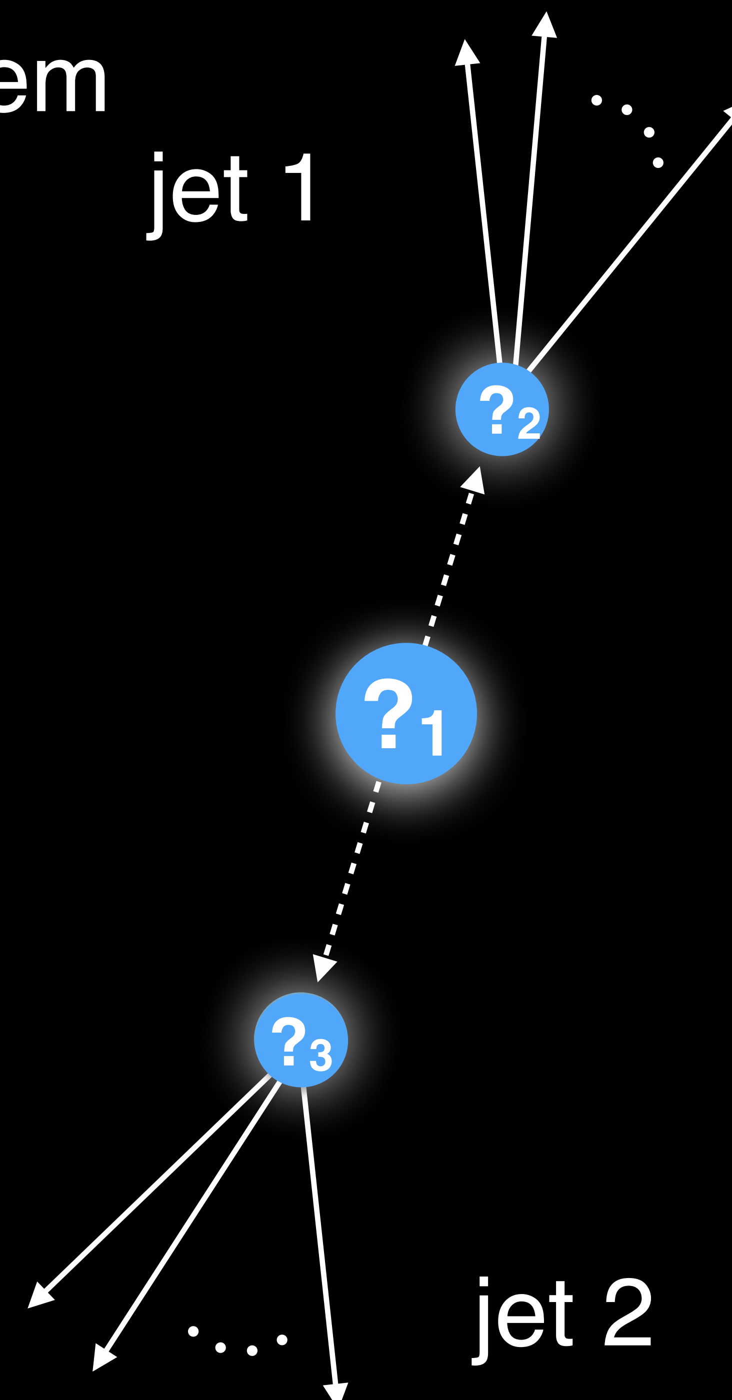
Other features can then be used to train CWoLa.

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

Example: two-jet search

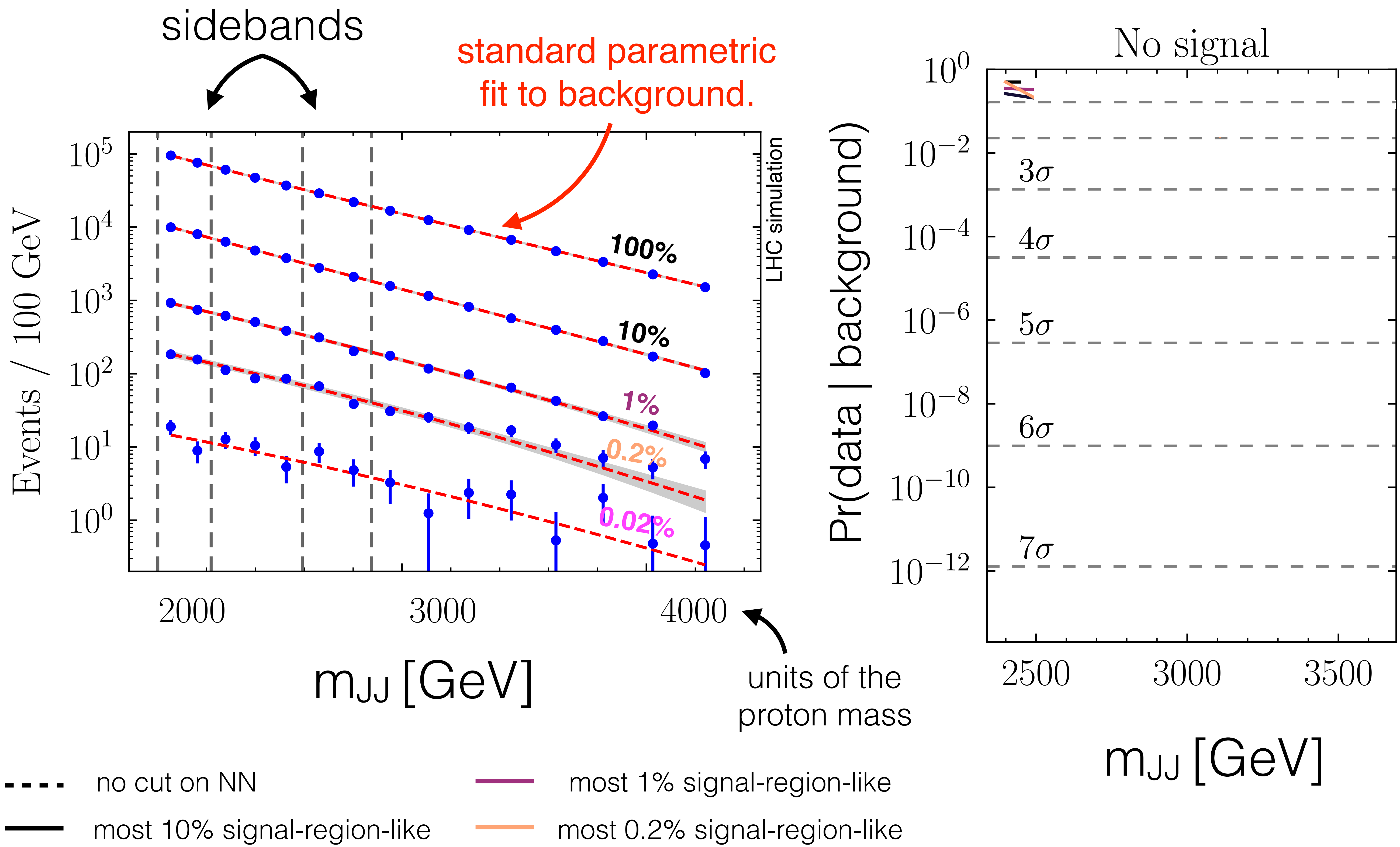


m_{res} = mass of two-jet system

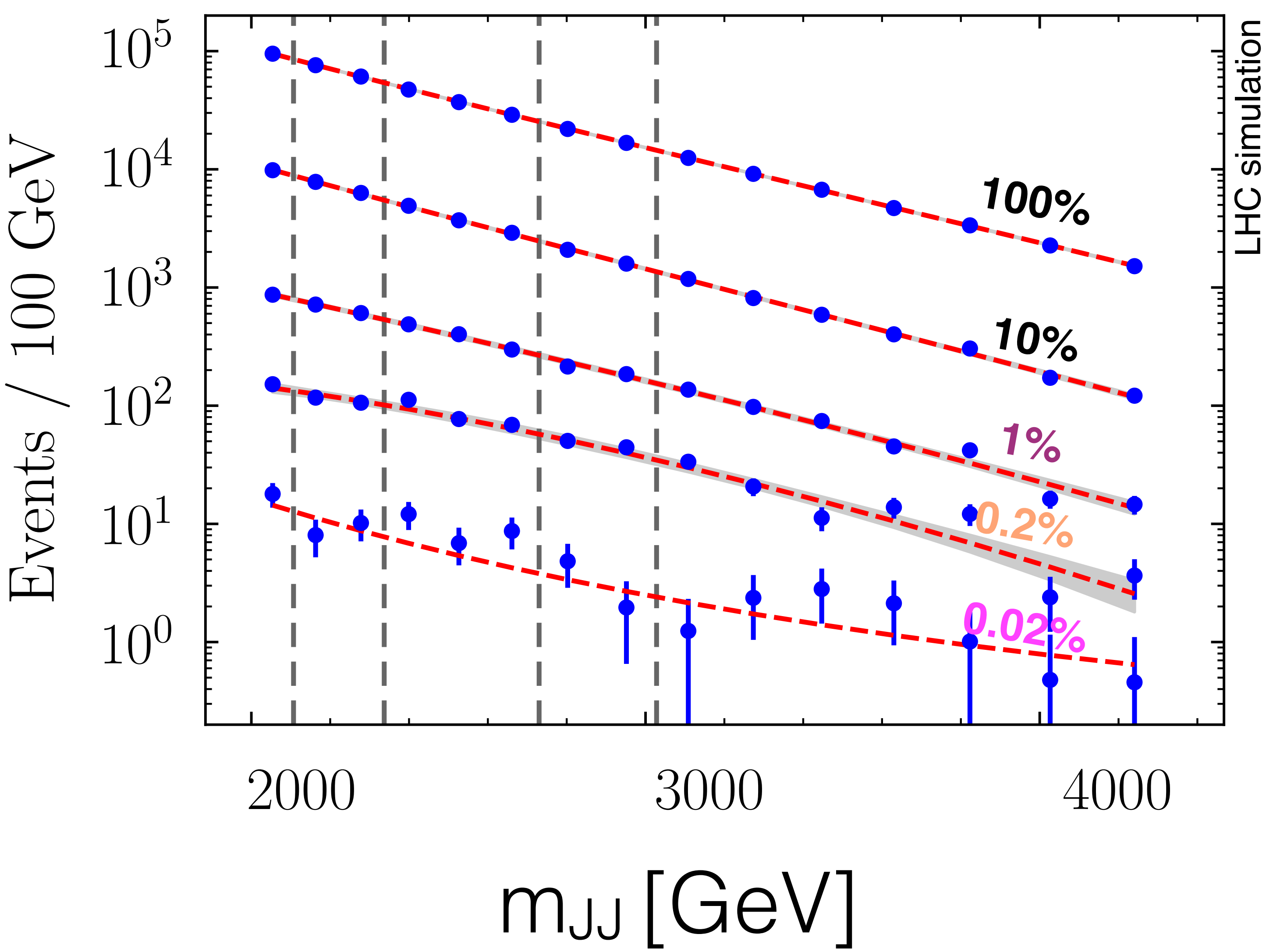


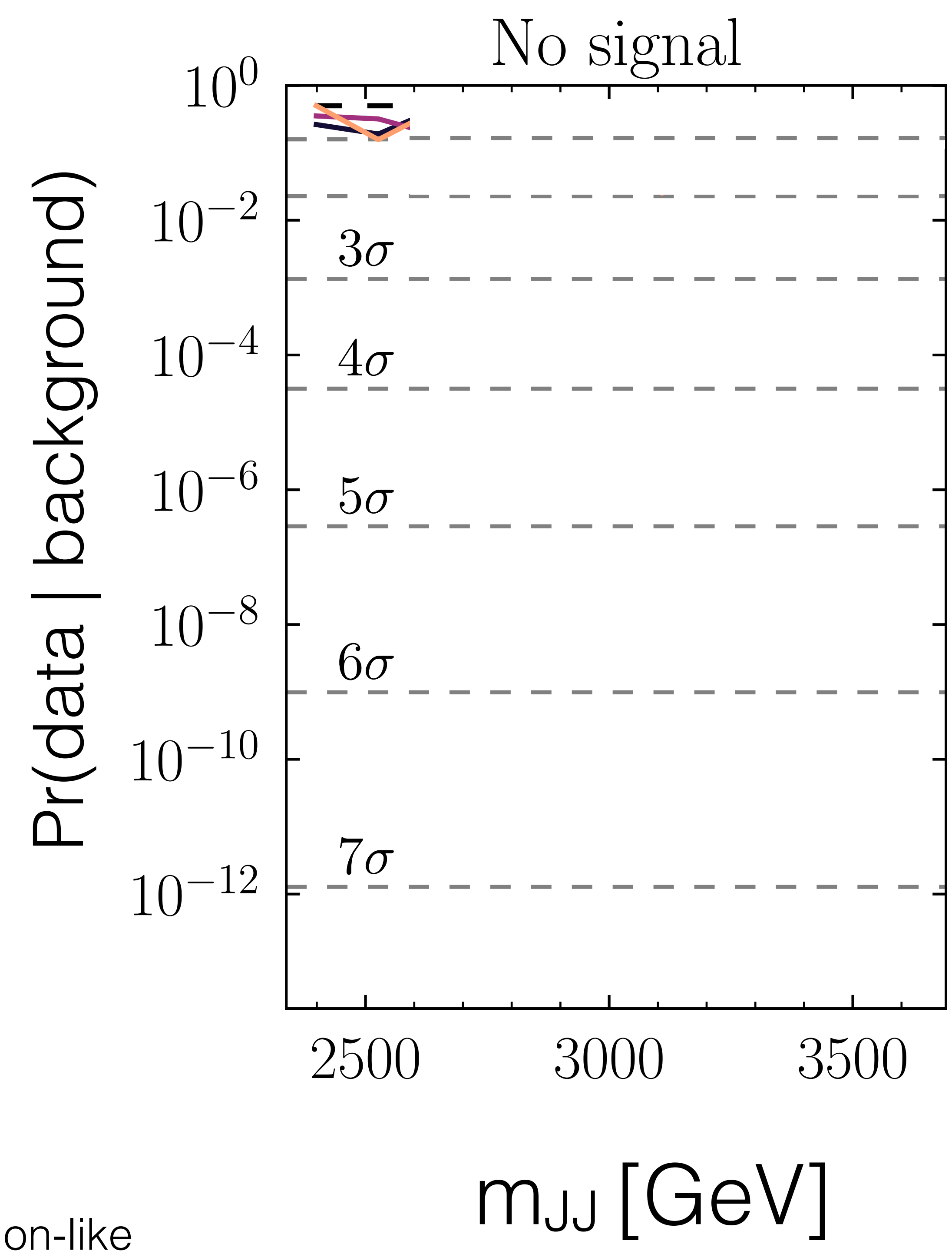
y = many features of the two jets

Example: two-jet search



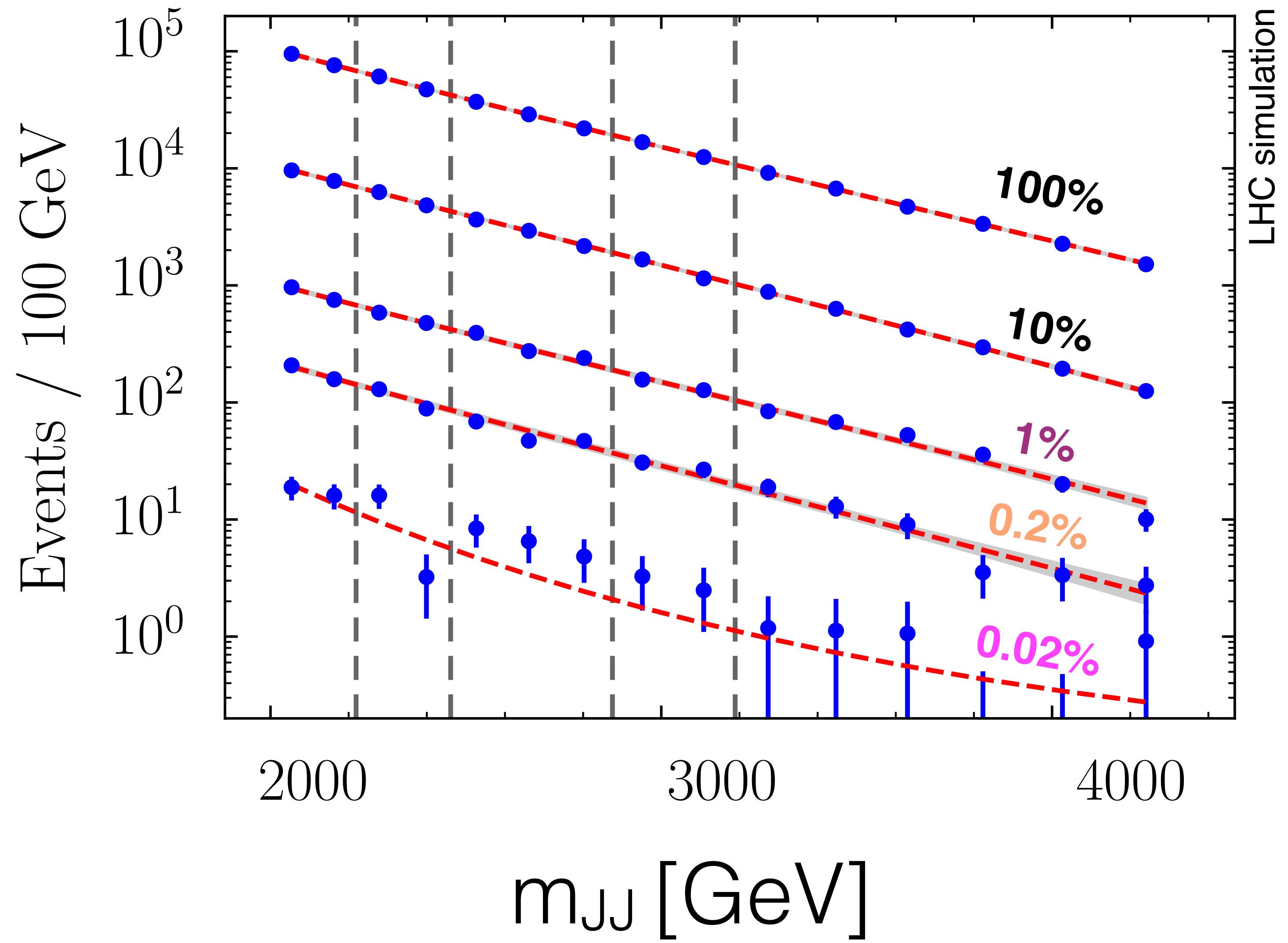
Example: two-jet search

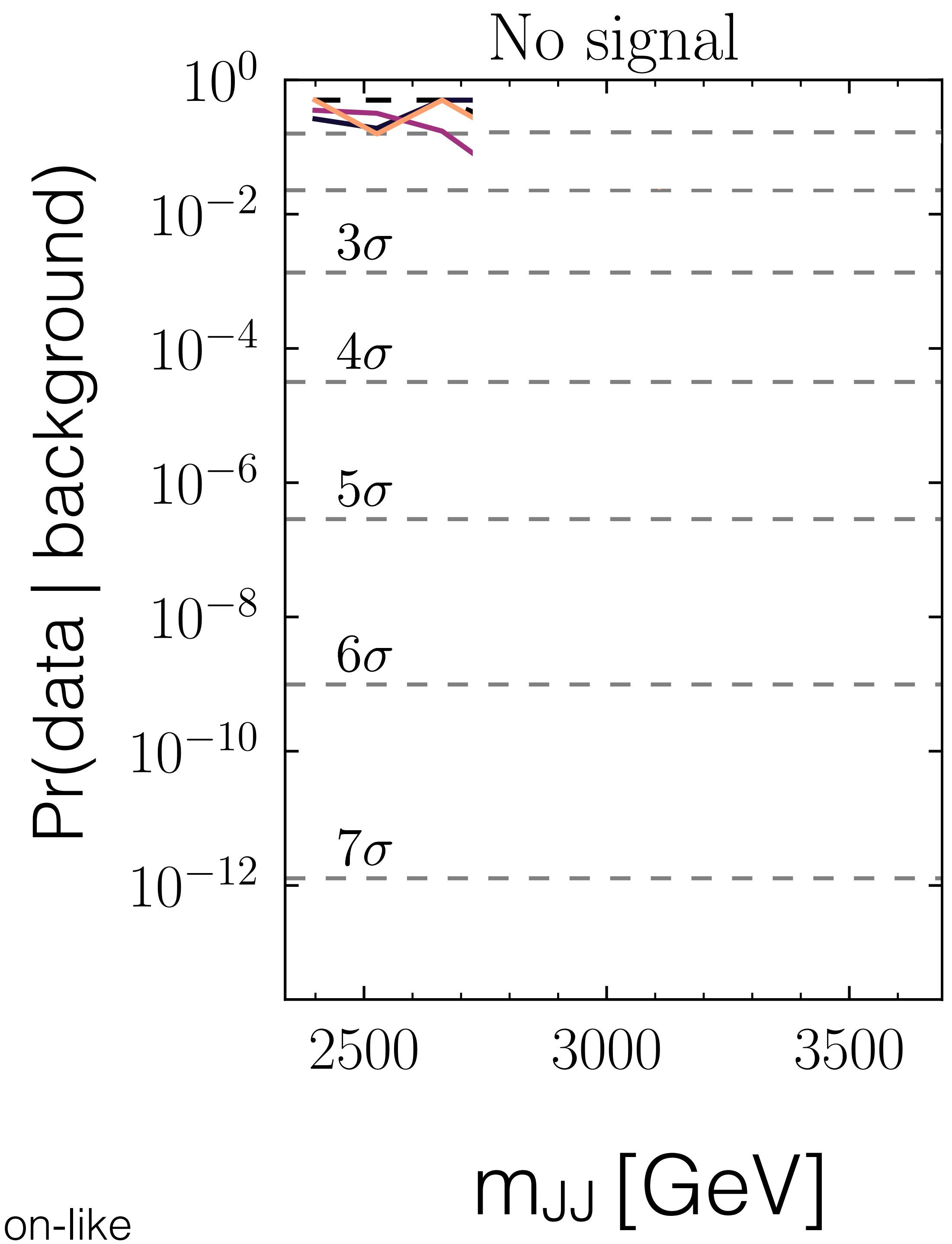




no cut on NN
most 1% signal-region-like
most 10% signal-region-like
most 0.2% signal-region-like

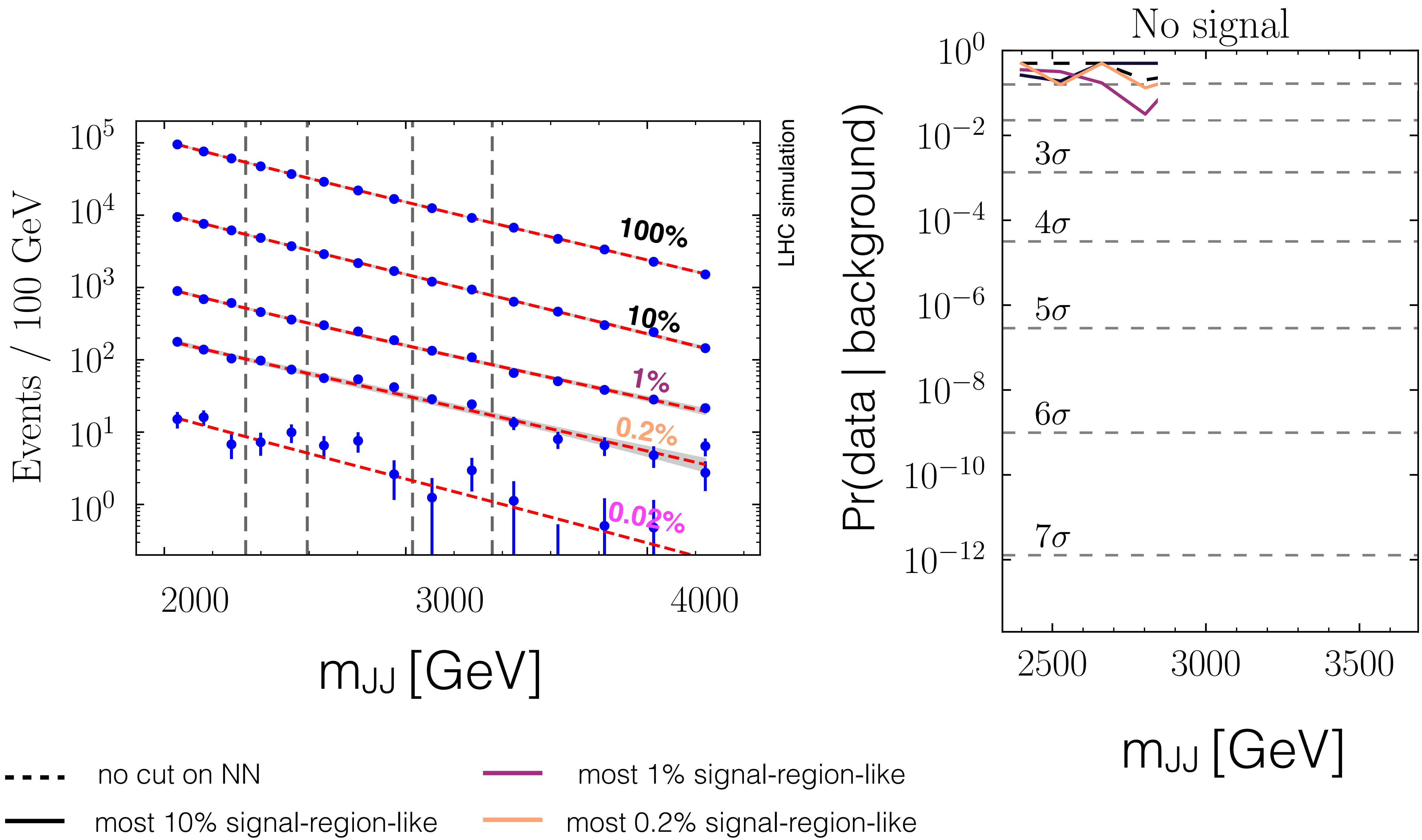
Example: two-jet search



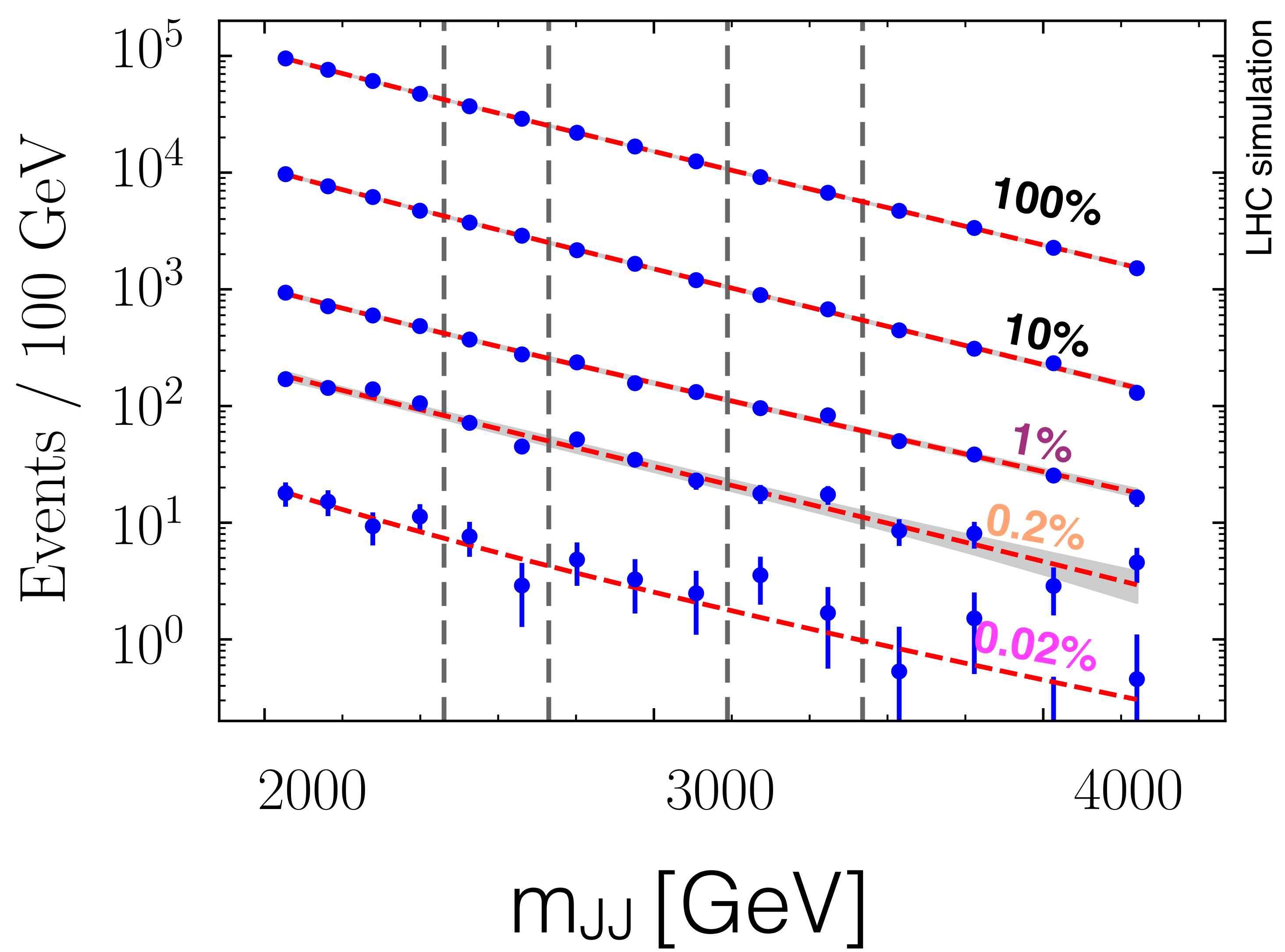


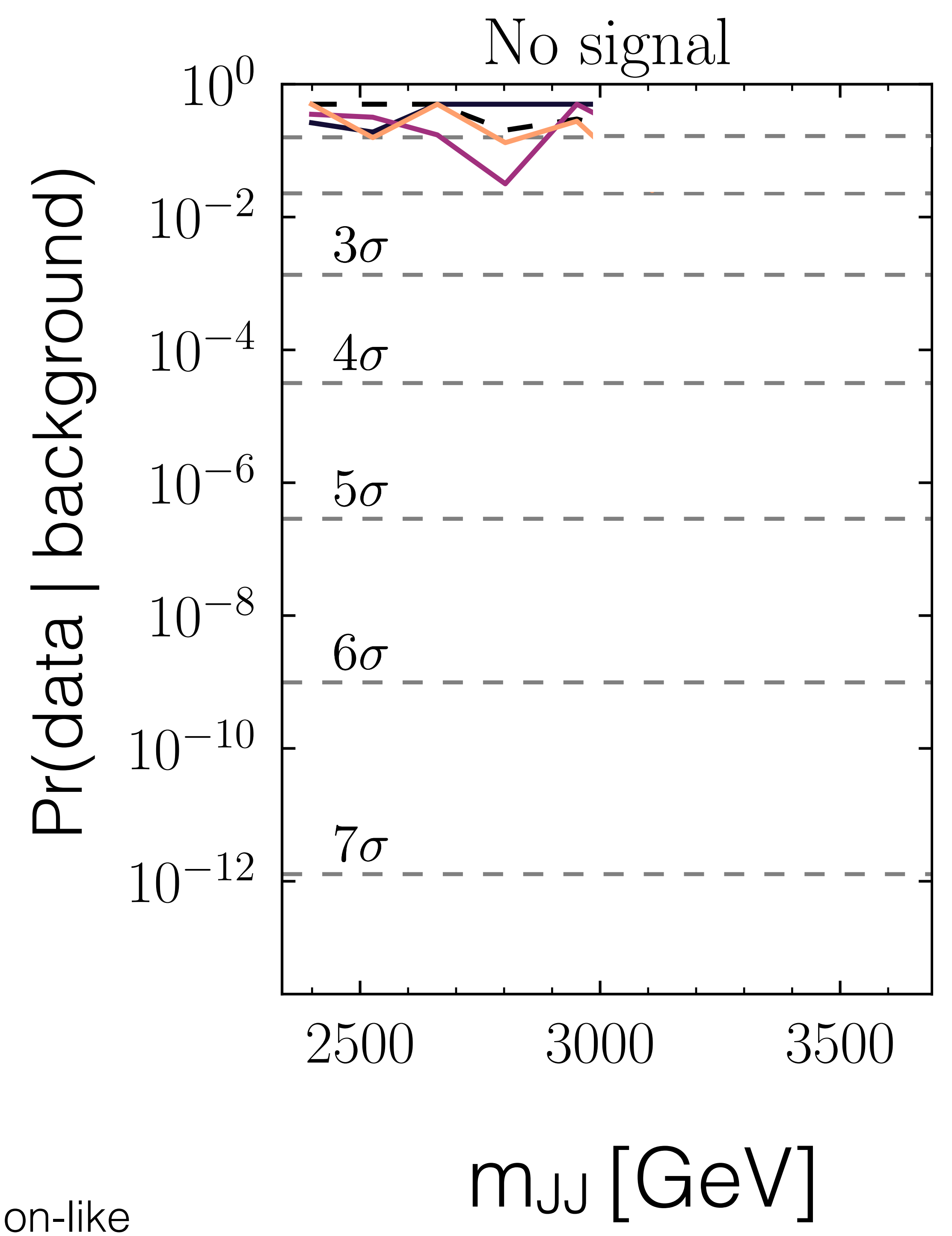
no cut on NN
most 10% signal-region-like
most 1% signal-region-like
most 0.2% signal-region-like

Example: two-jet search



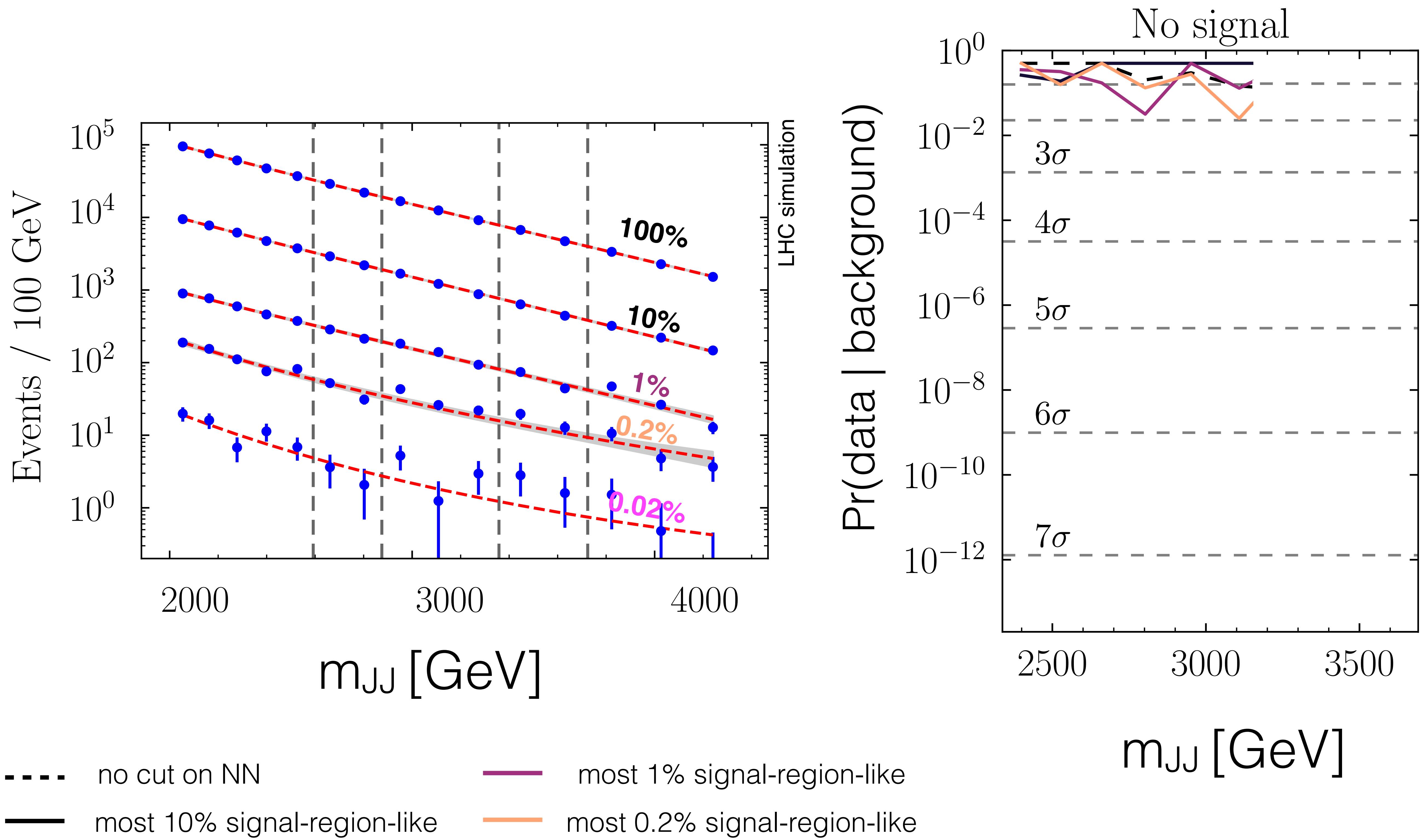
Example: two-jet search



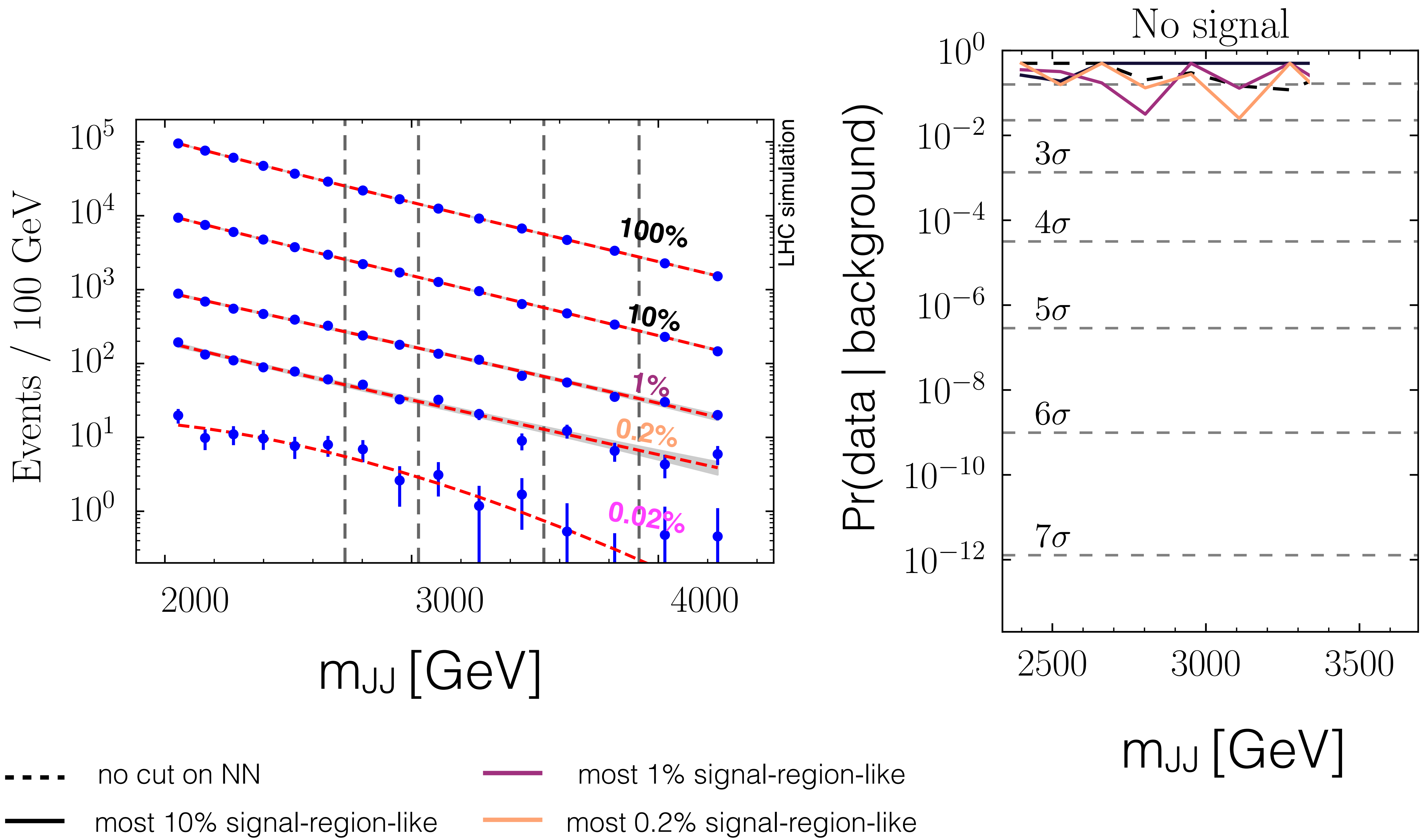


- - - no cut on NN
— most 10% signal-region-like
— most 1% signal-region-like
— most 0.2% signal-region-like

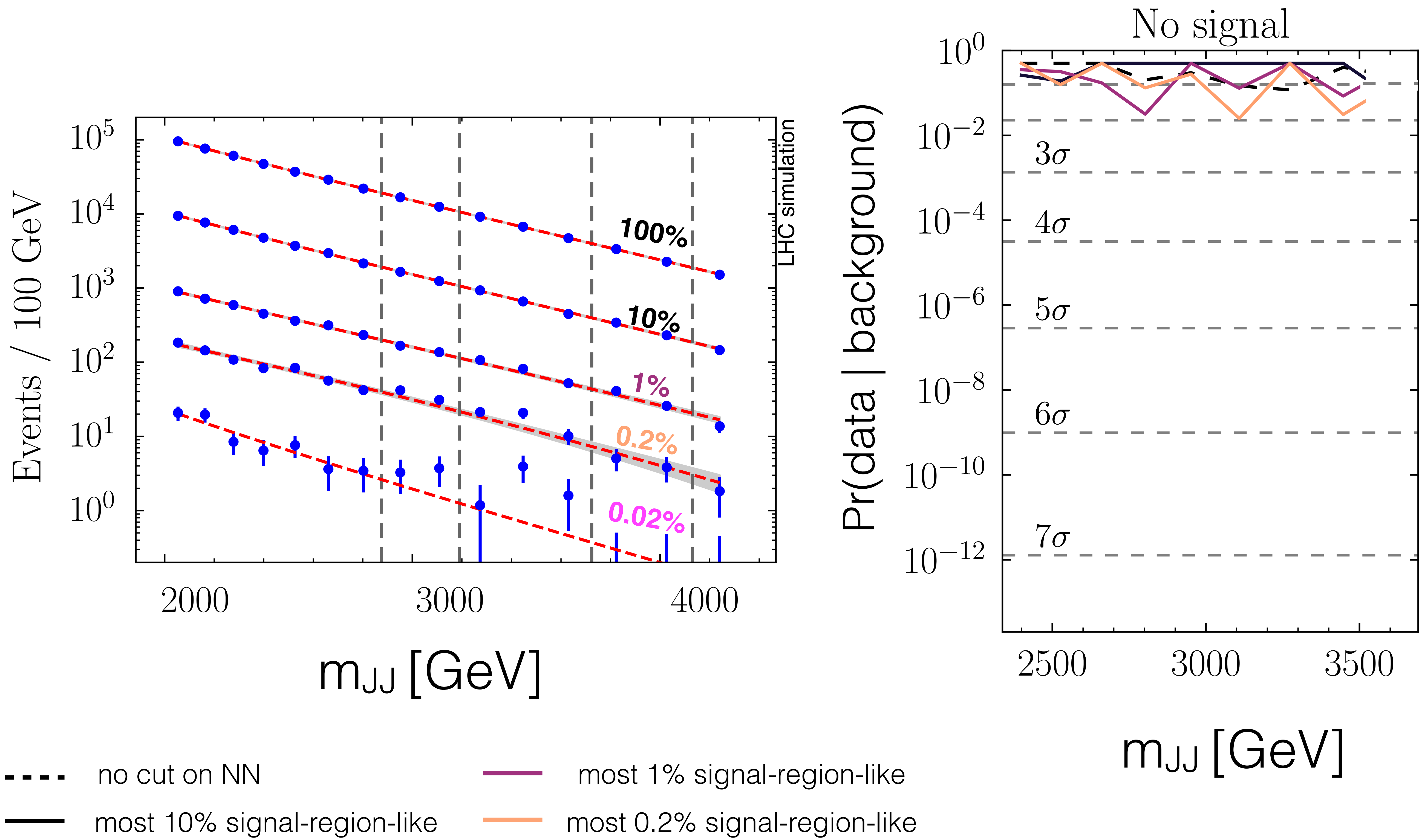
Example: two-jet search



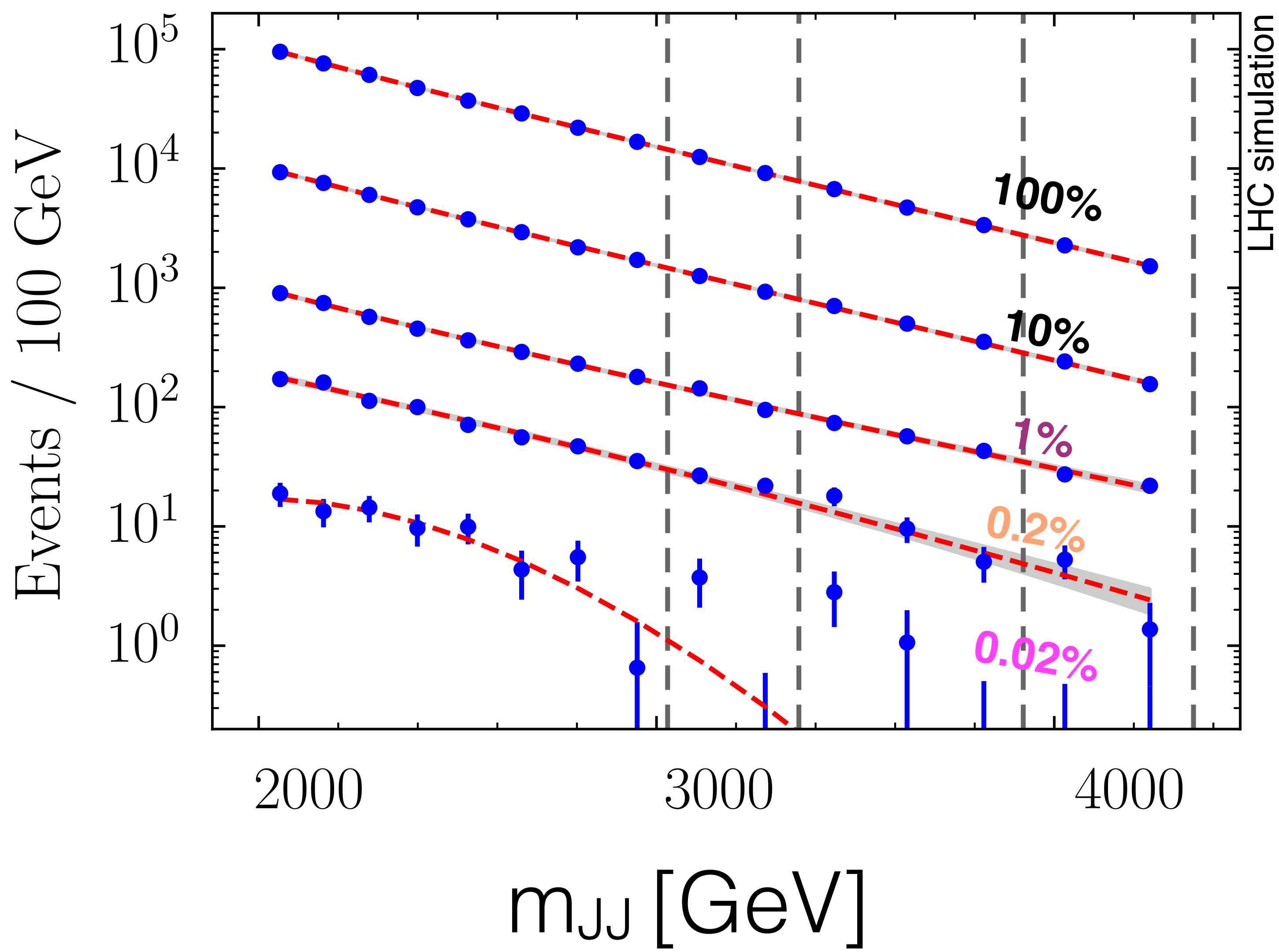
Example: two-jet search

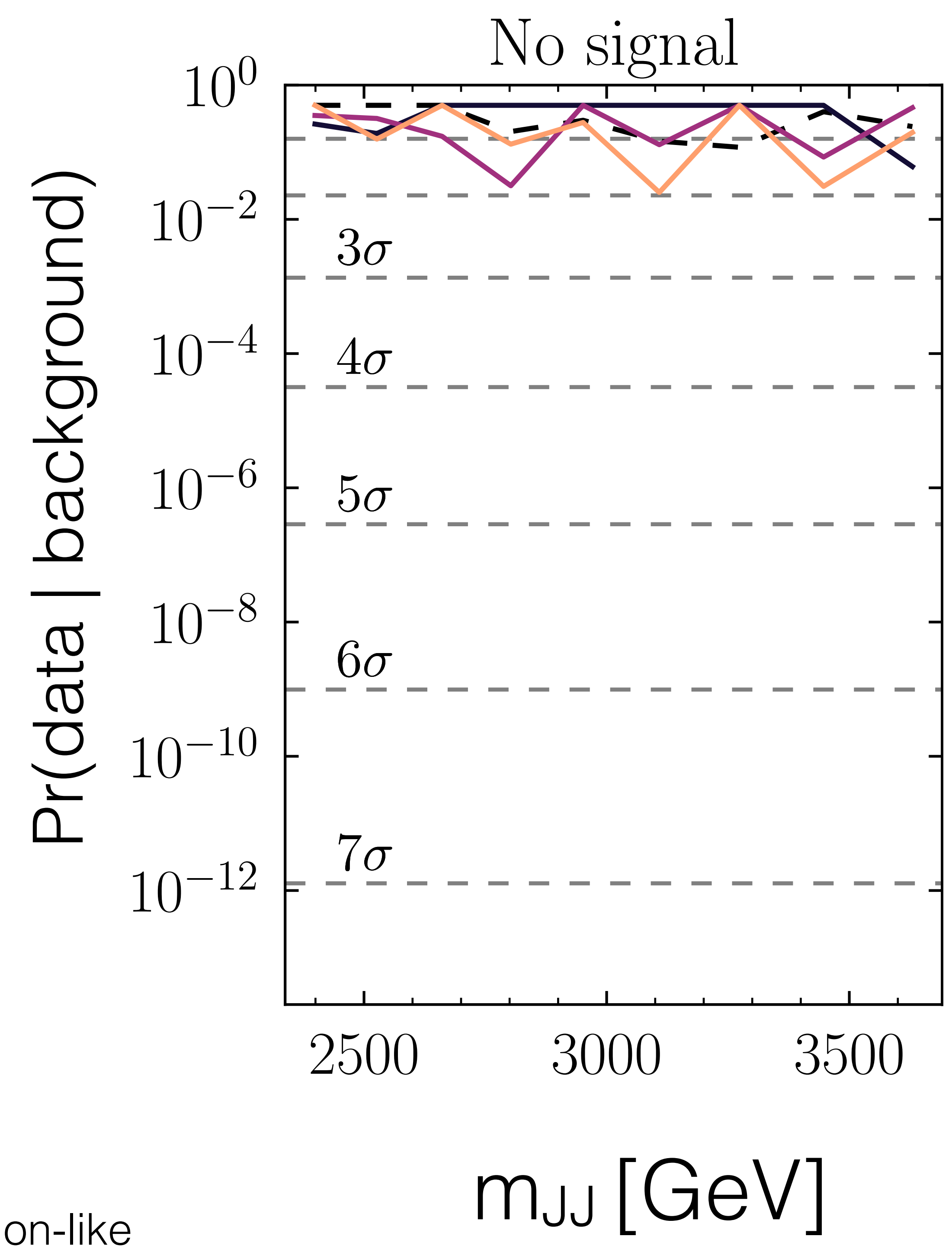


Example: two-jet search



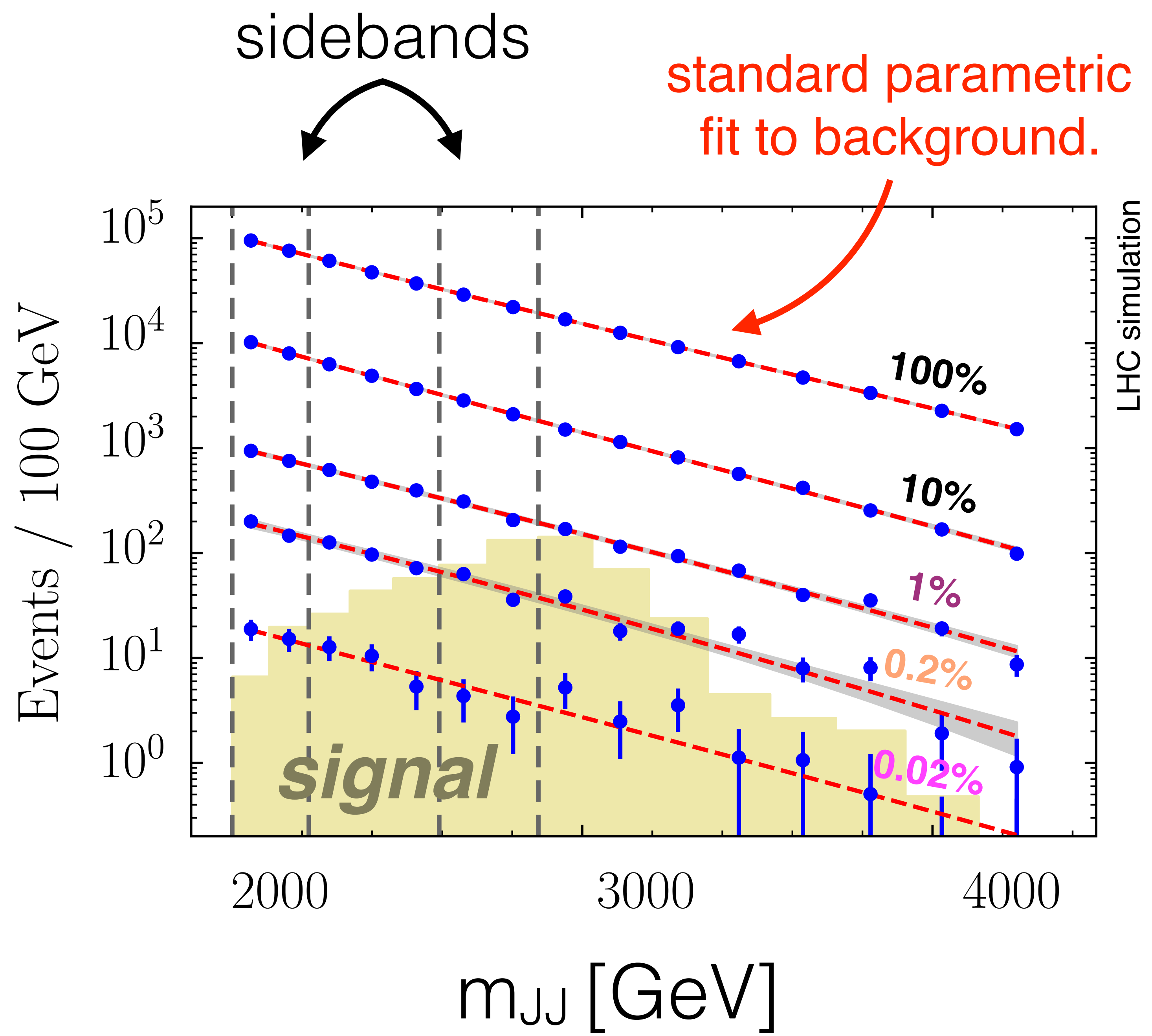
Example: two-jet search

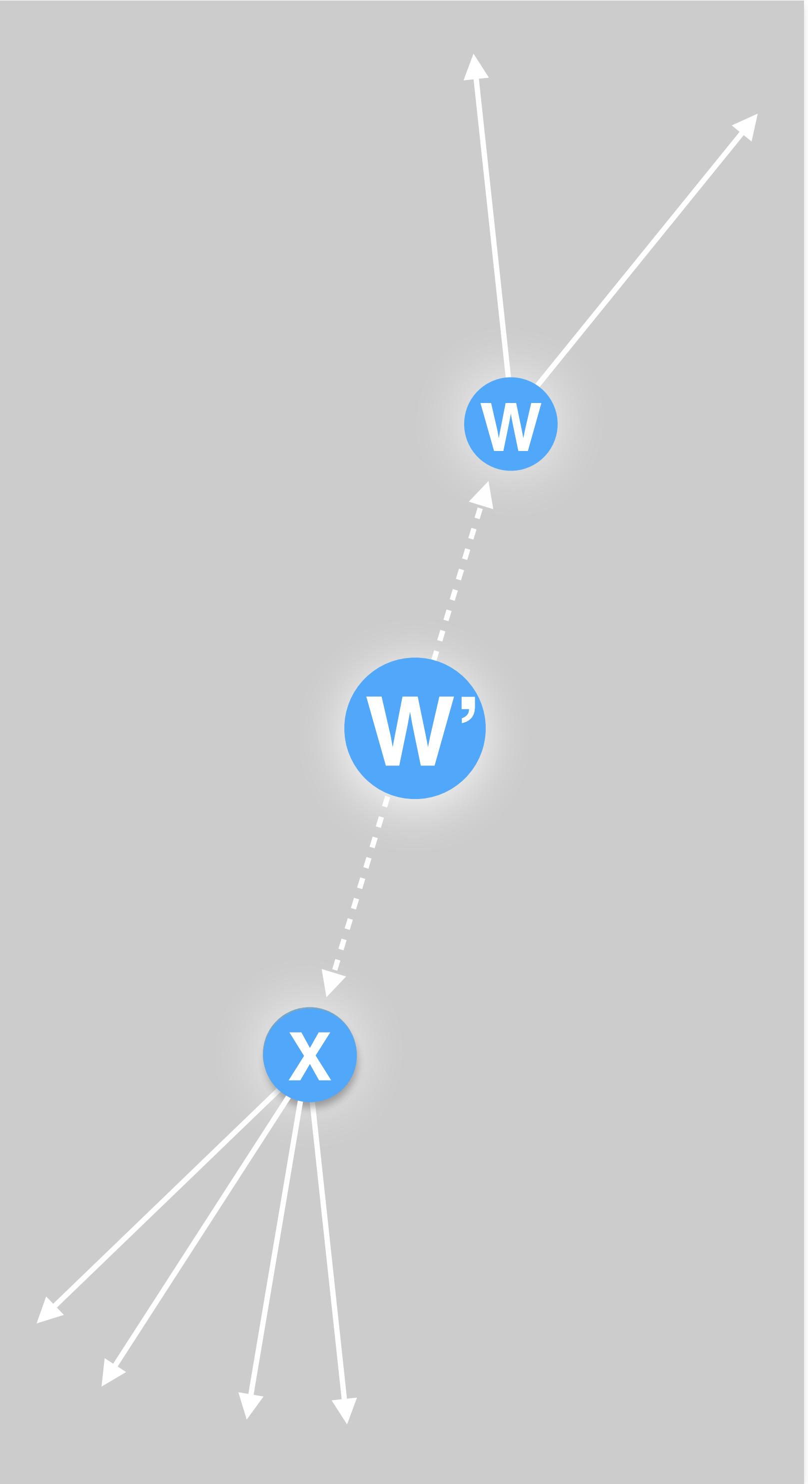




no cut on NN
most 10% signal-region-like
most 1% signal-region-like
most 0.2% signal-region-like

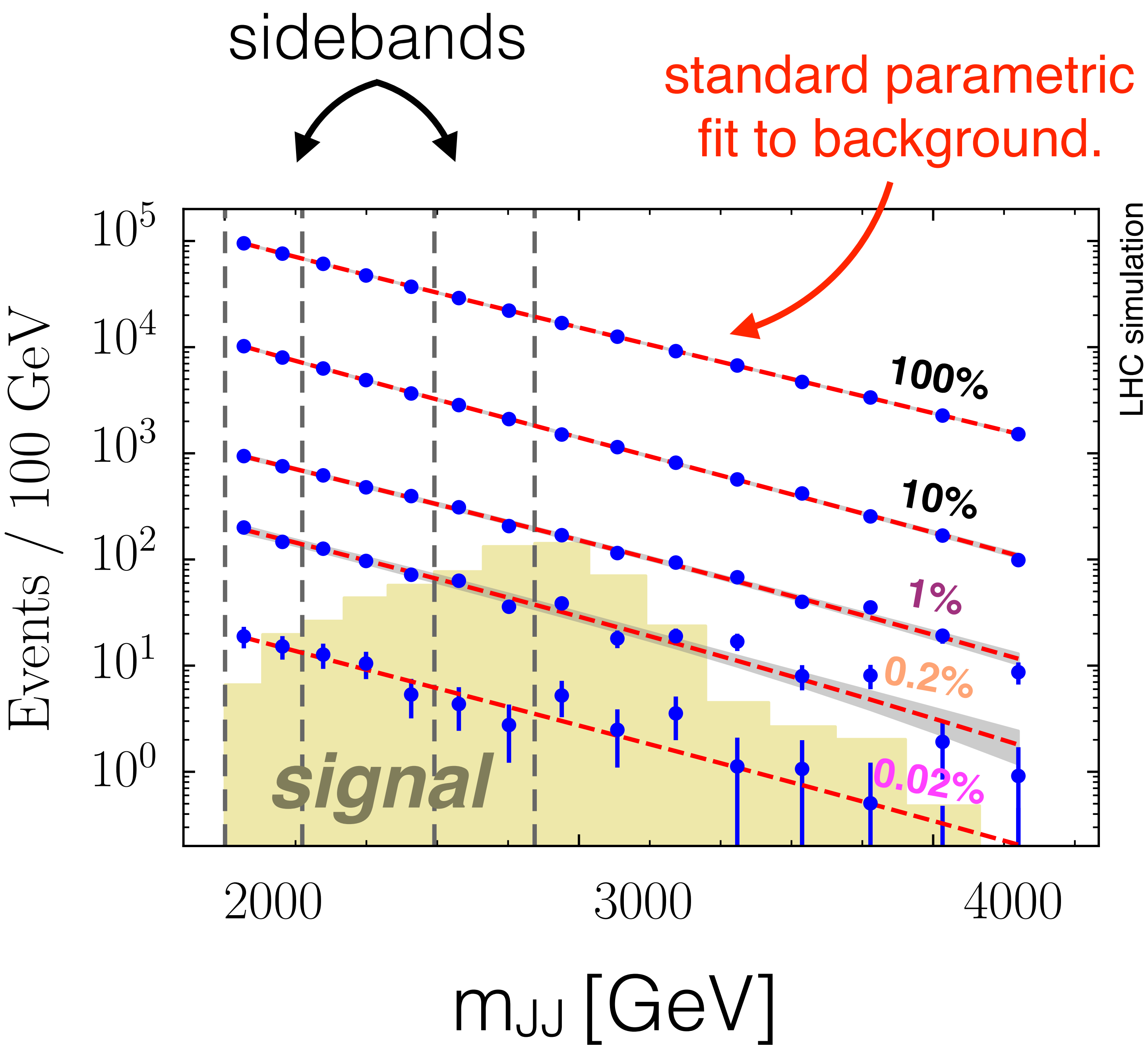
…and when there is a signal?

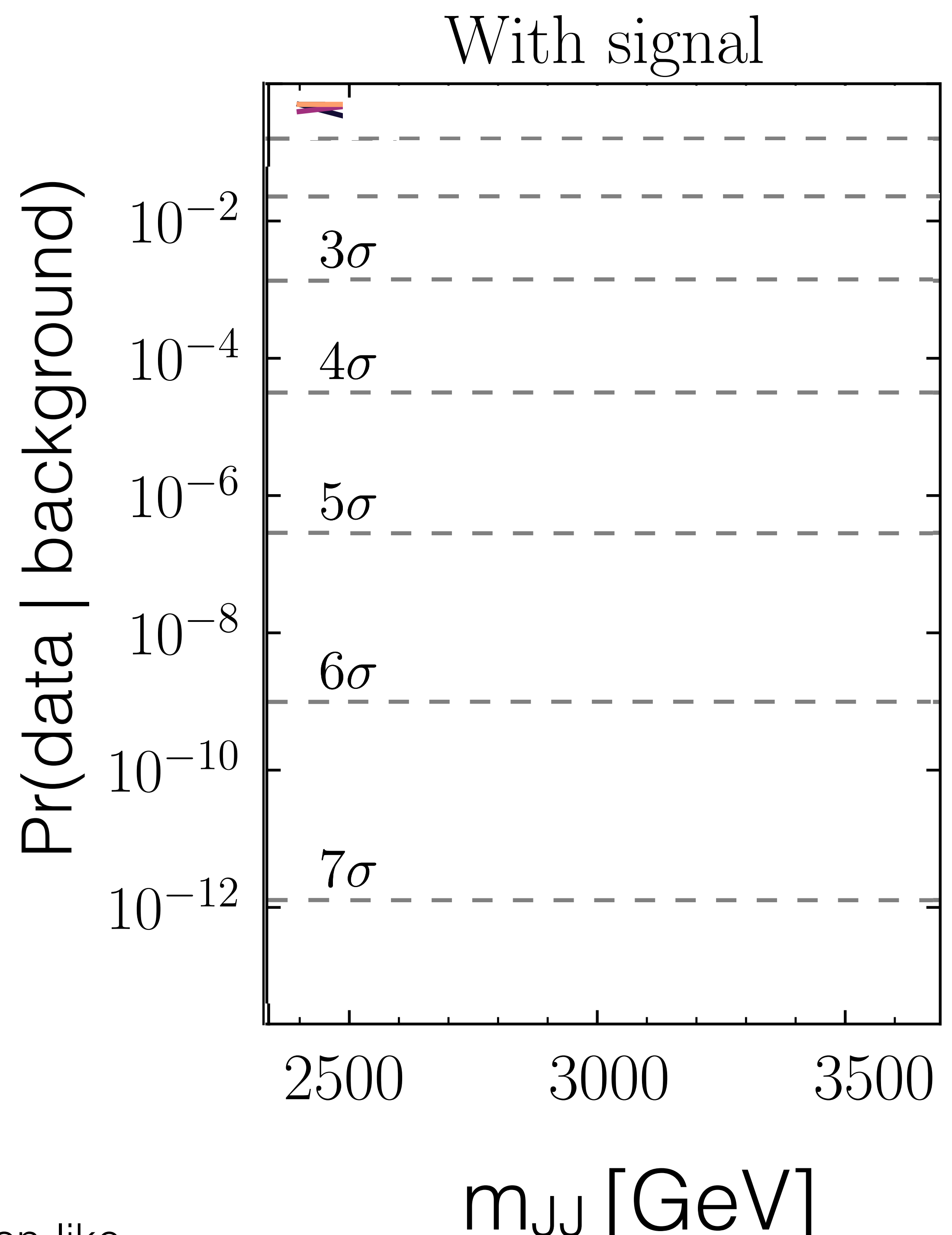




- - - - no cut on NN
— most 10% signal-region-like
— most 1% signal-region-like
— most 0.2% signal-region-like

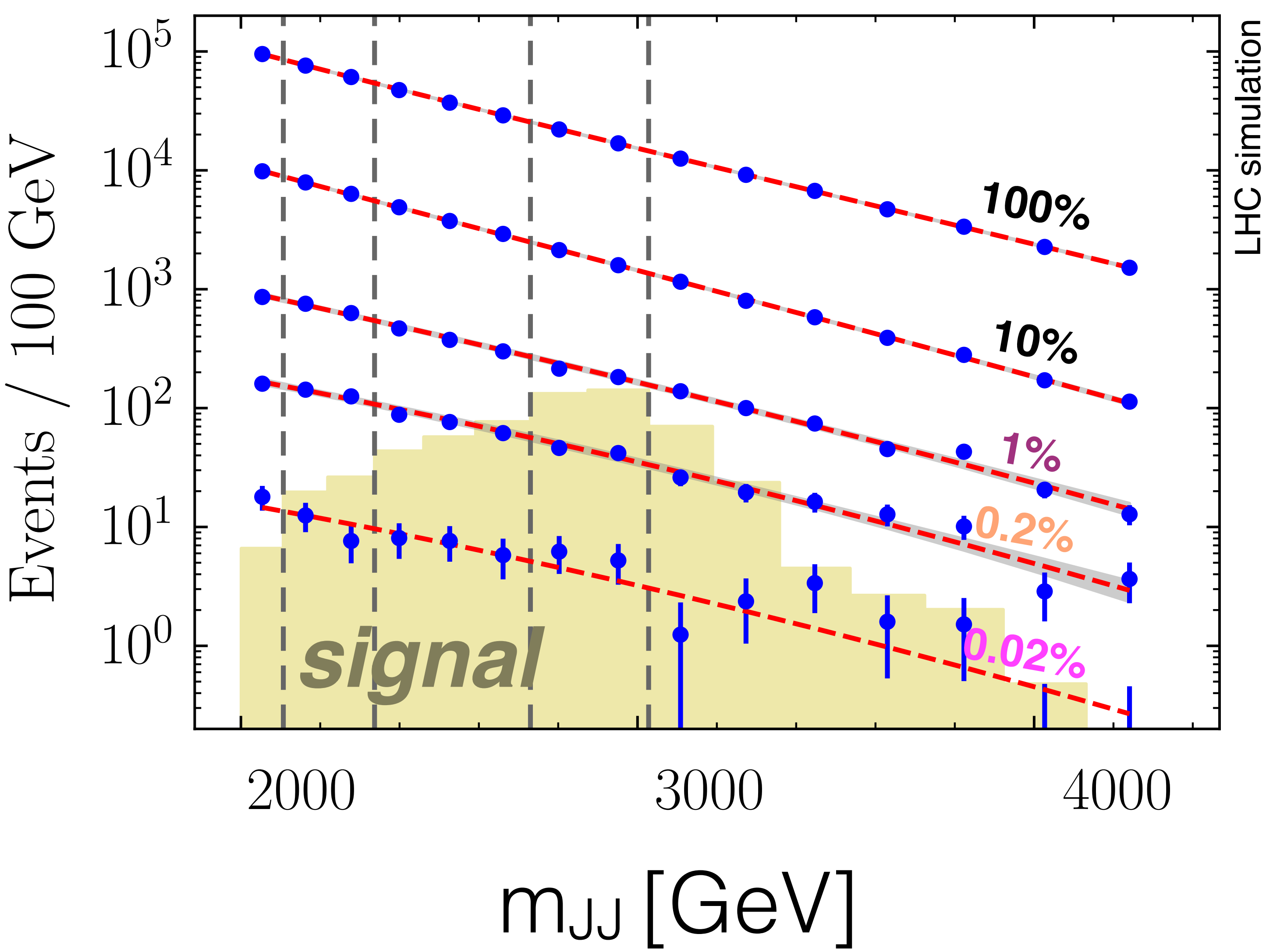
…and when there is a signal?

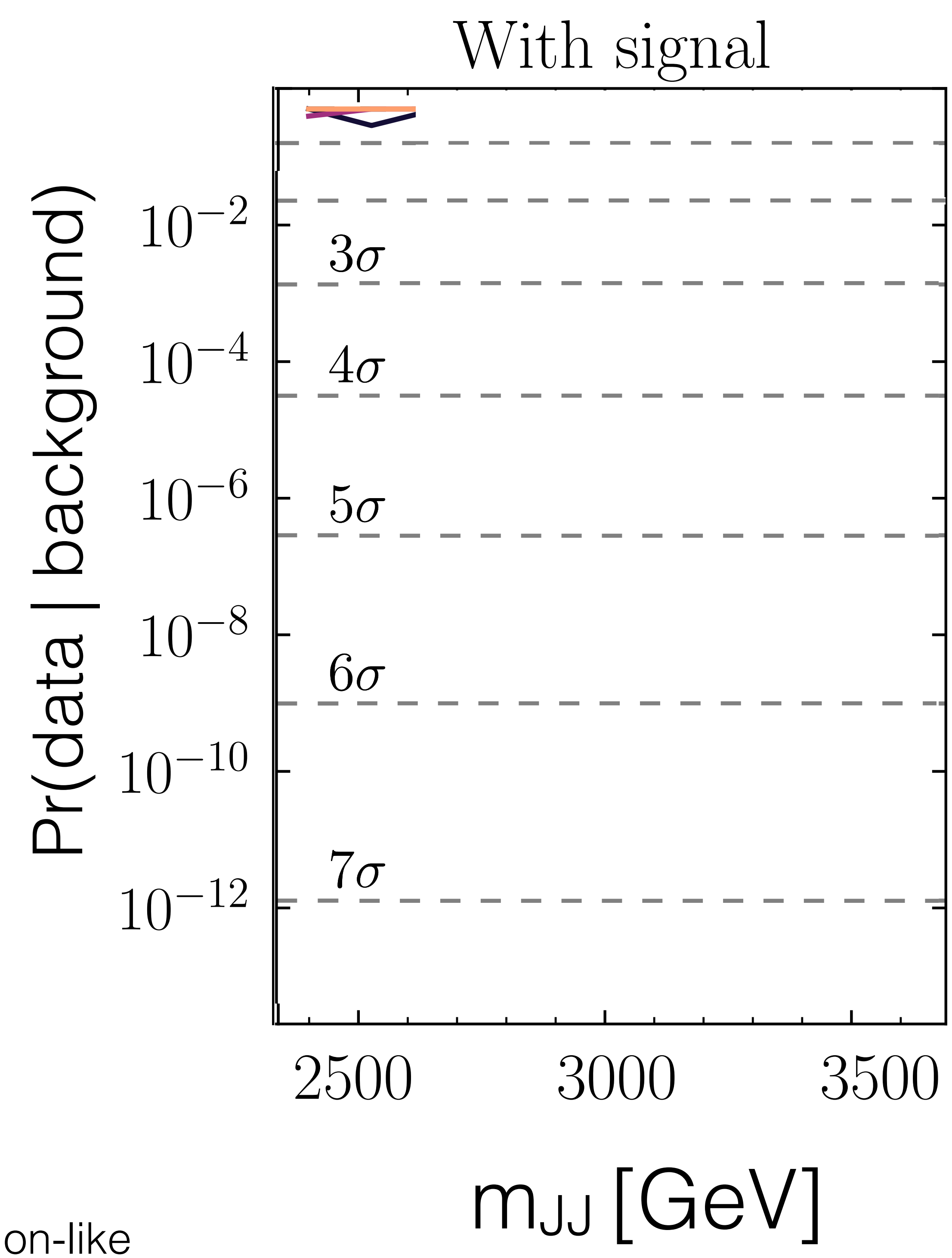




no cut on NN
most 1% signal-region-like
most 10% signal-region-like
most 0.2% signal-region-like

…and when there is a signal?





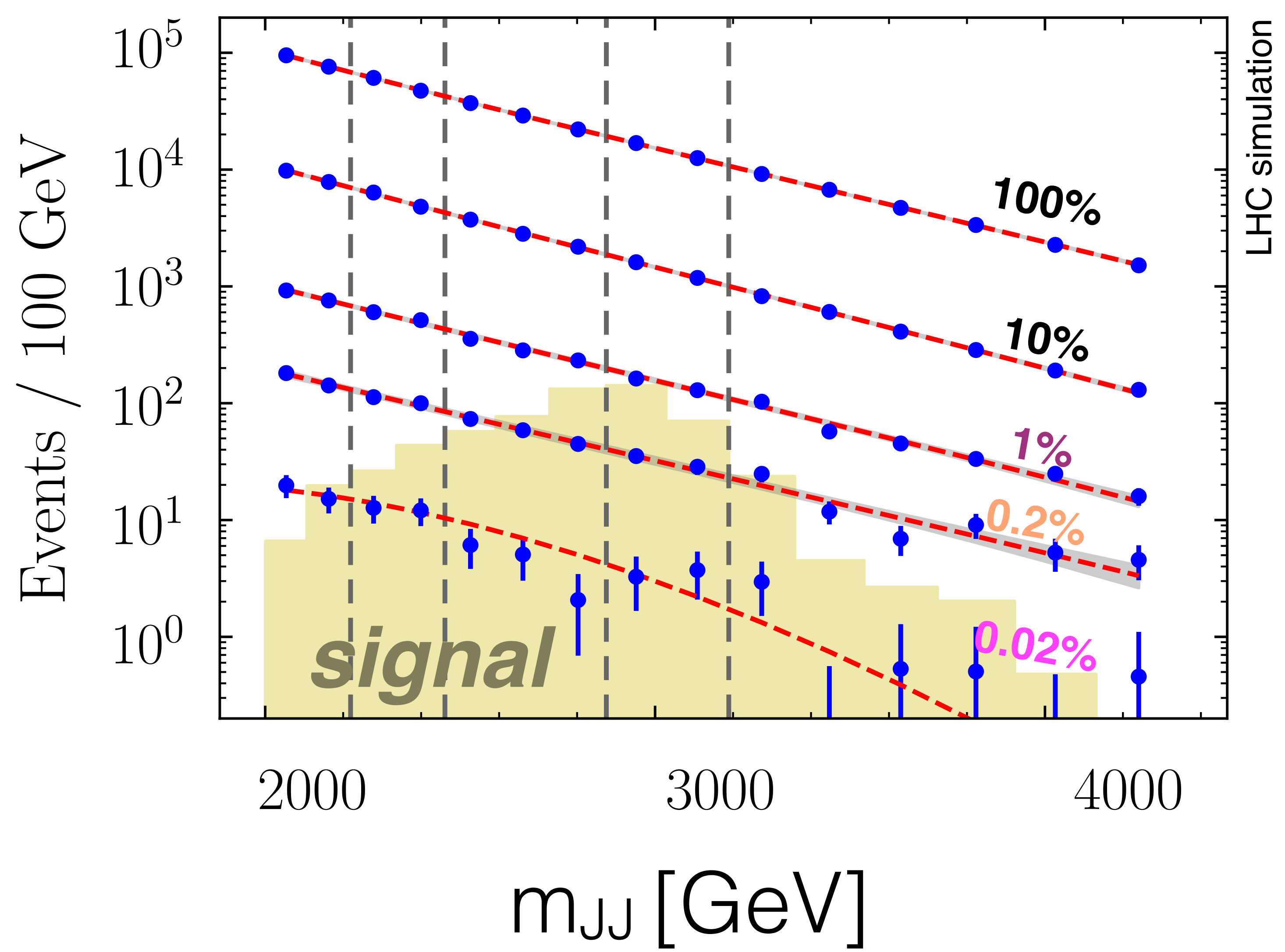
no cut on NN

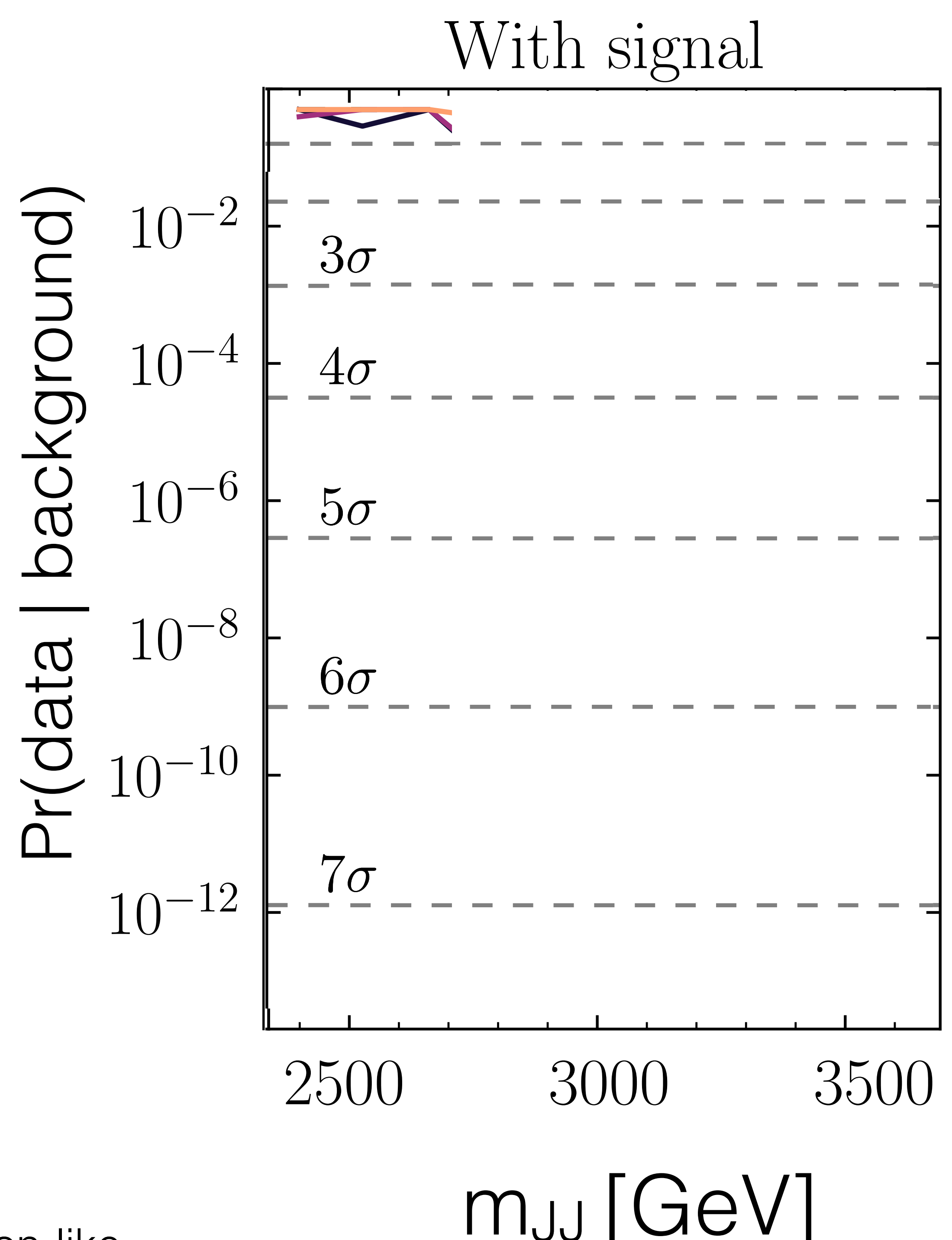
most 10% signal-region-like

most 1% signal-region-like

most 0.2% signal-region-like

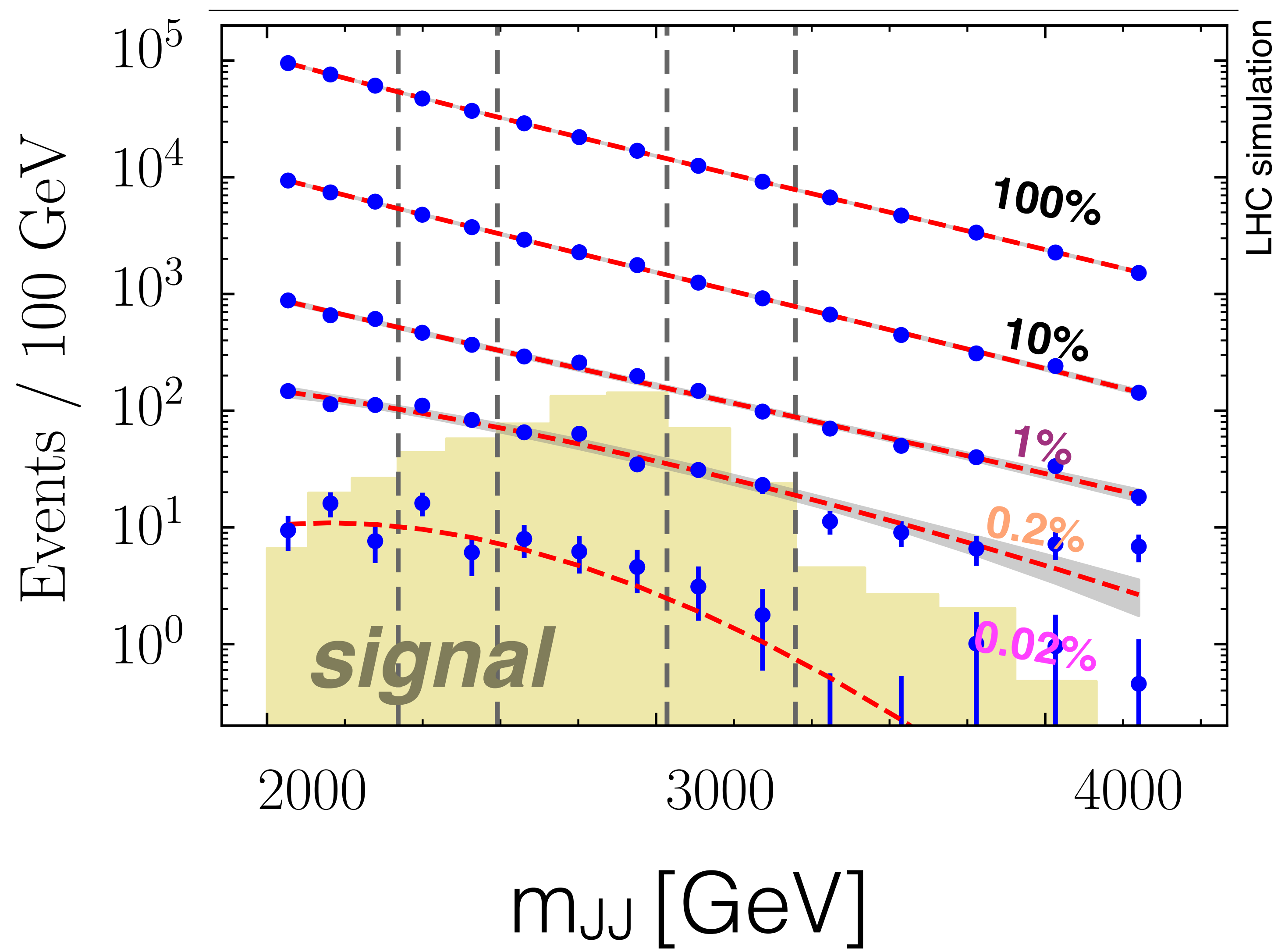
…and when there is a signal?

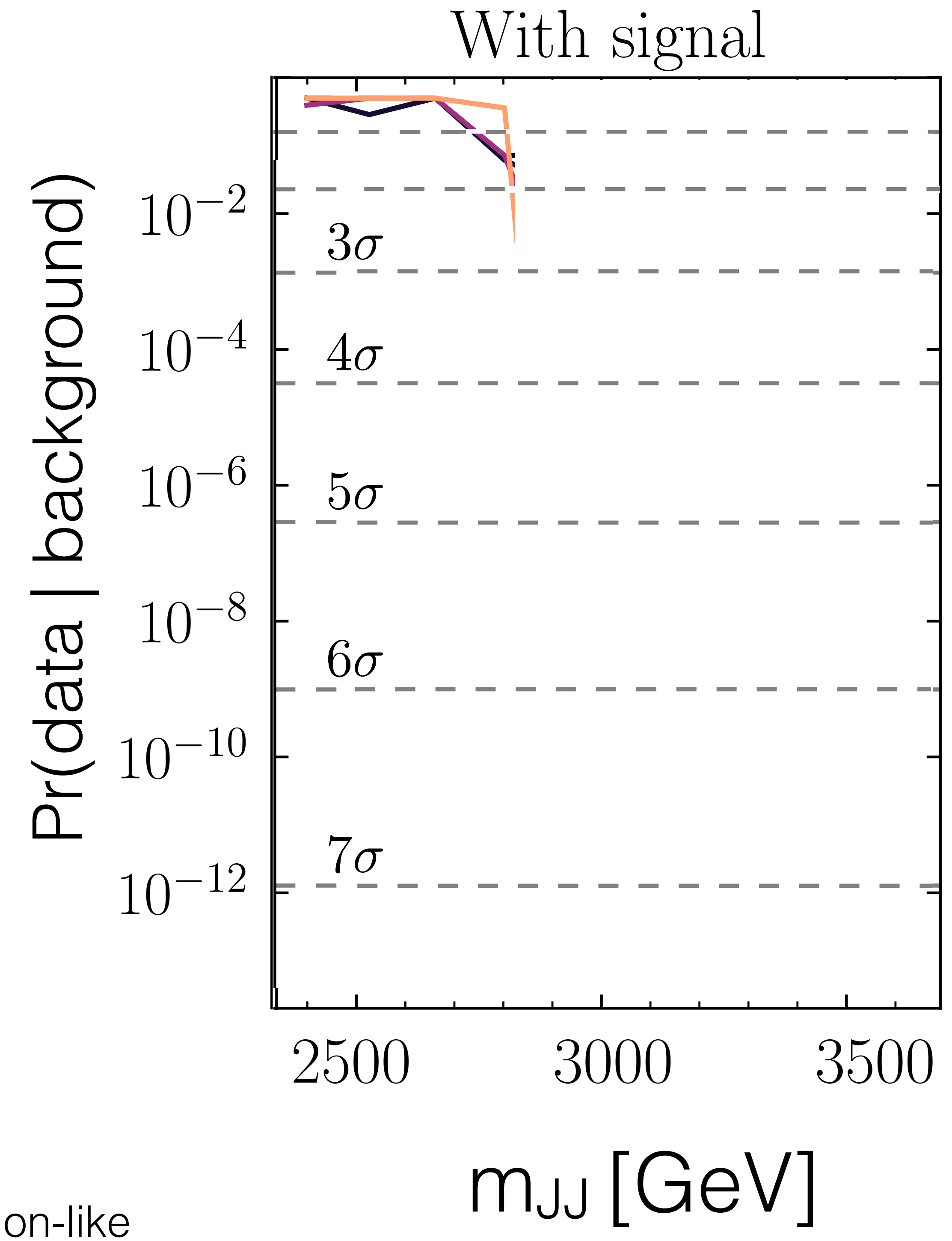




- - - no cut on NN
— most 10% signal-region-like
— most 1% signal-region-like
— most 0.2% signal-region-like

…and when there is a signal?





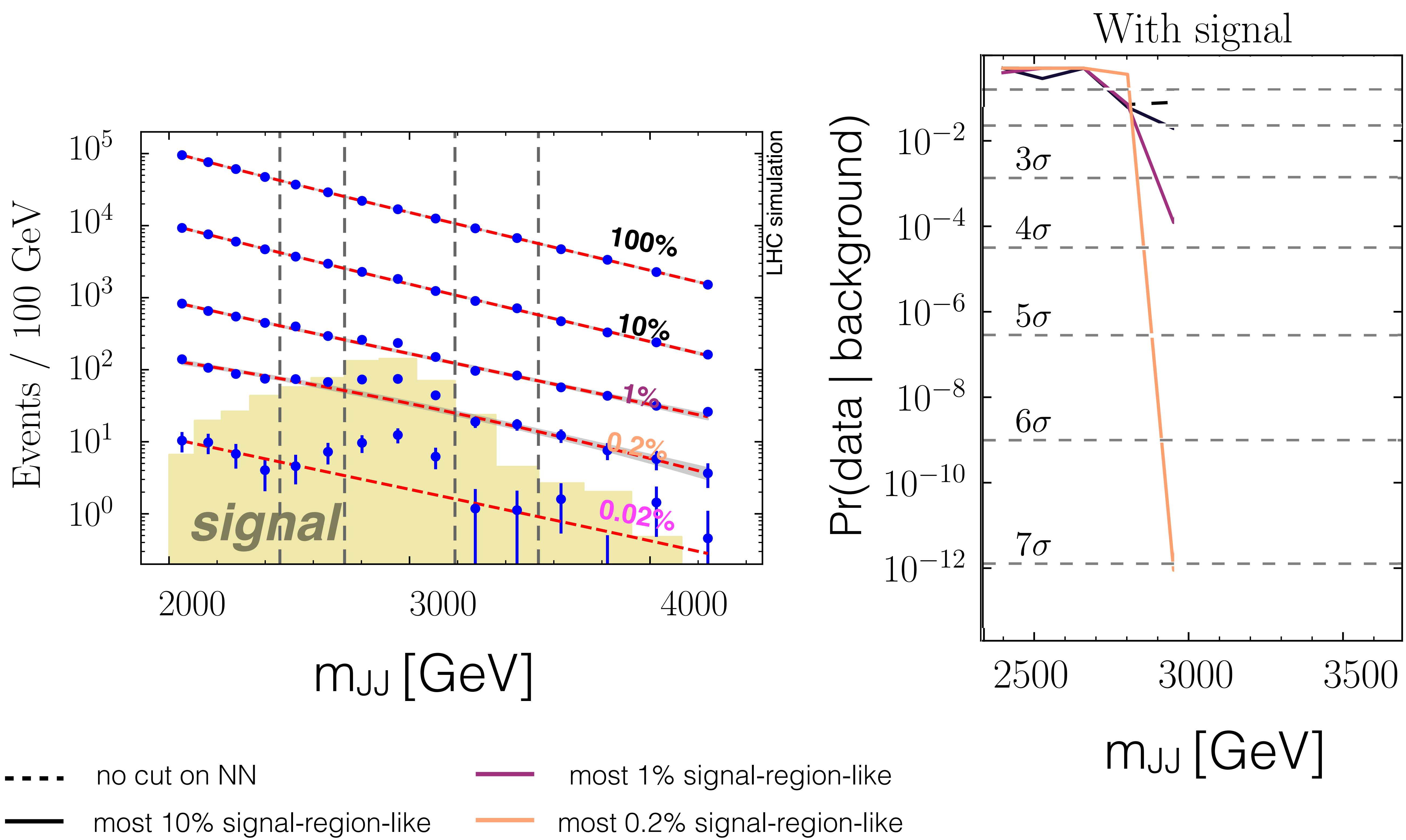
no cut on NN

most 10% signal-region-like

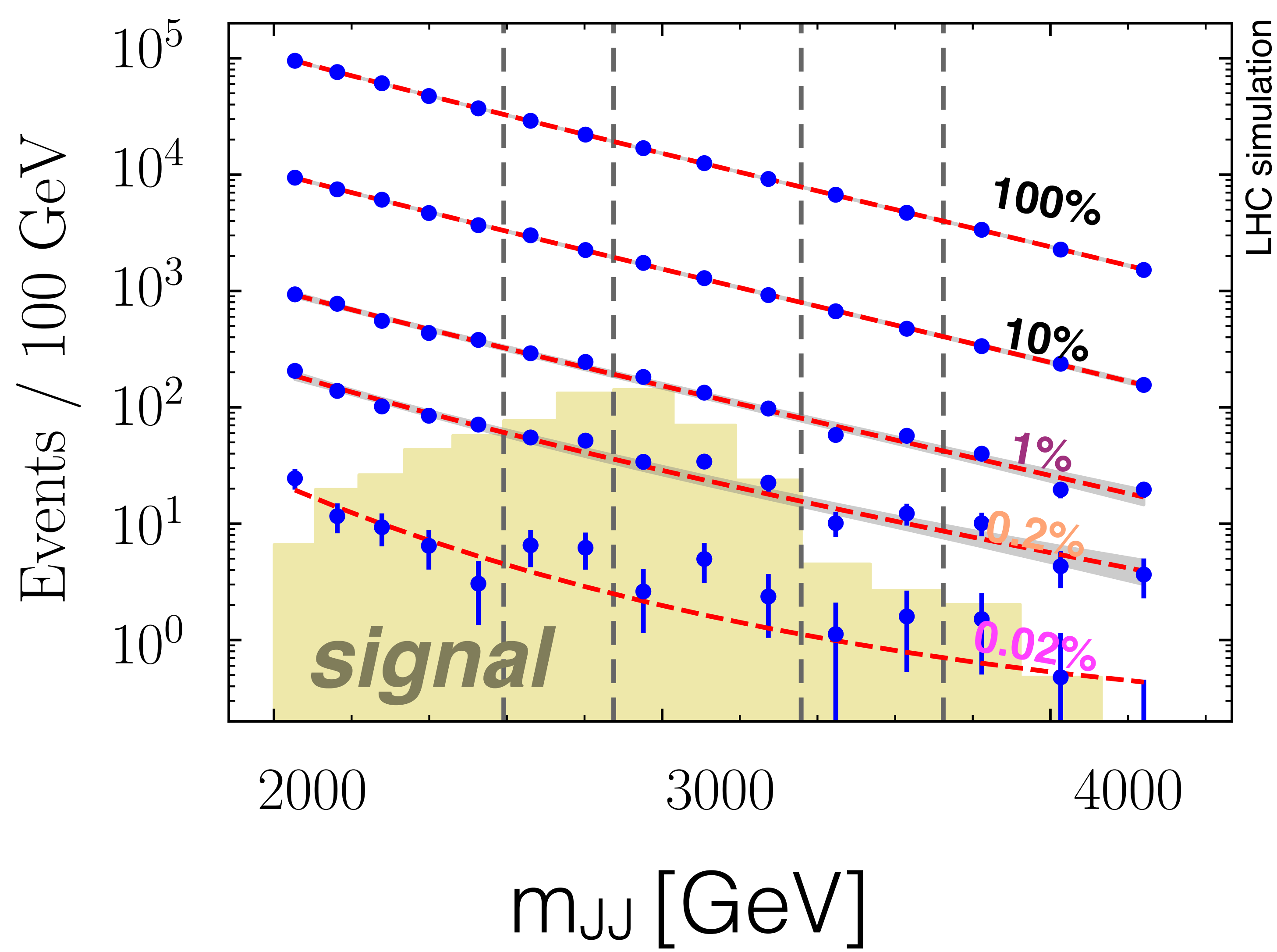
most 1% signal-region-like

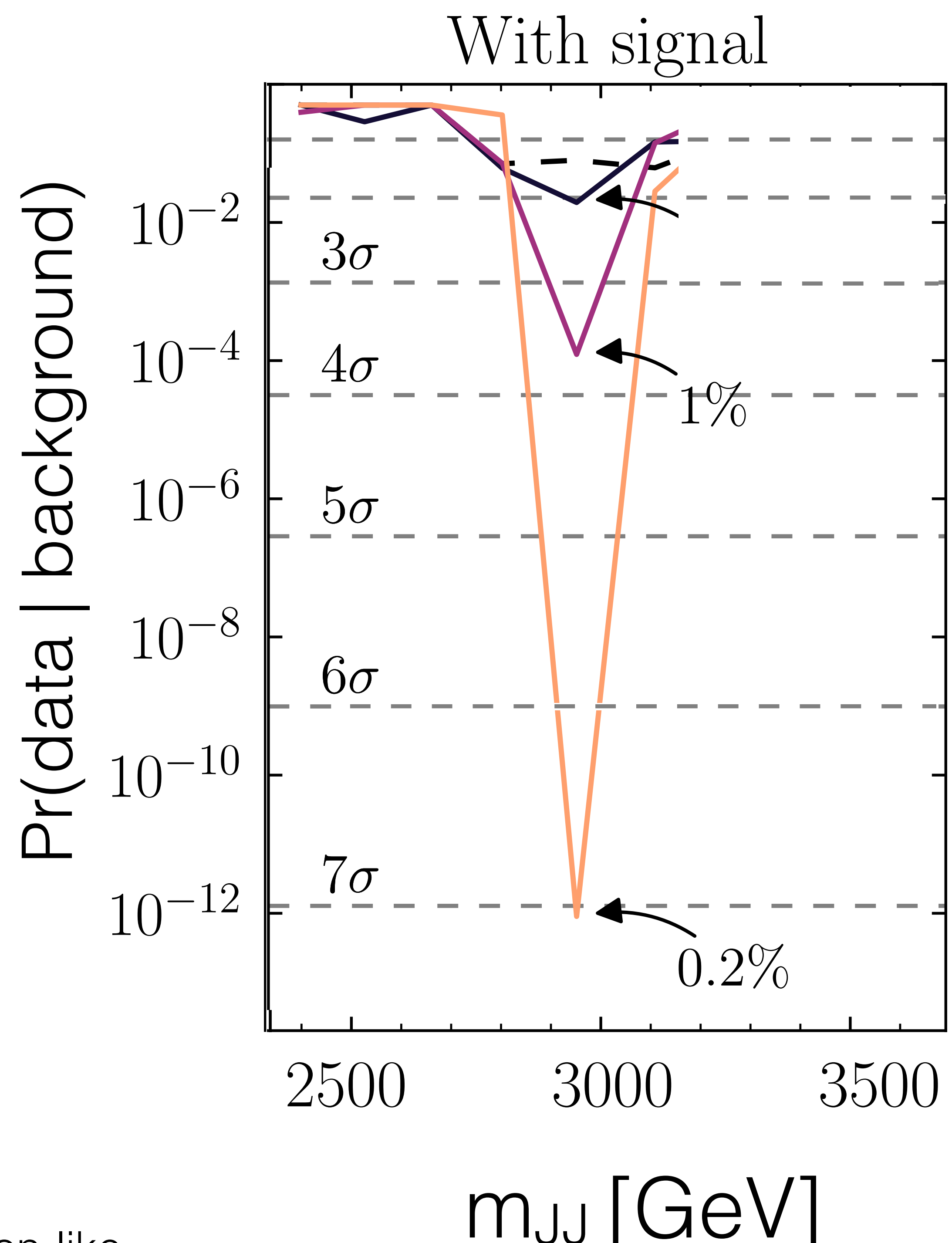
most 0.2% signal-region-like

…and when there is a signal?



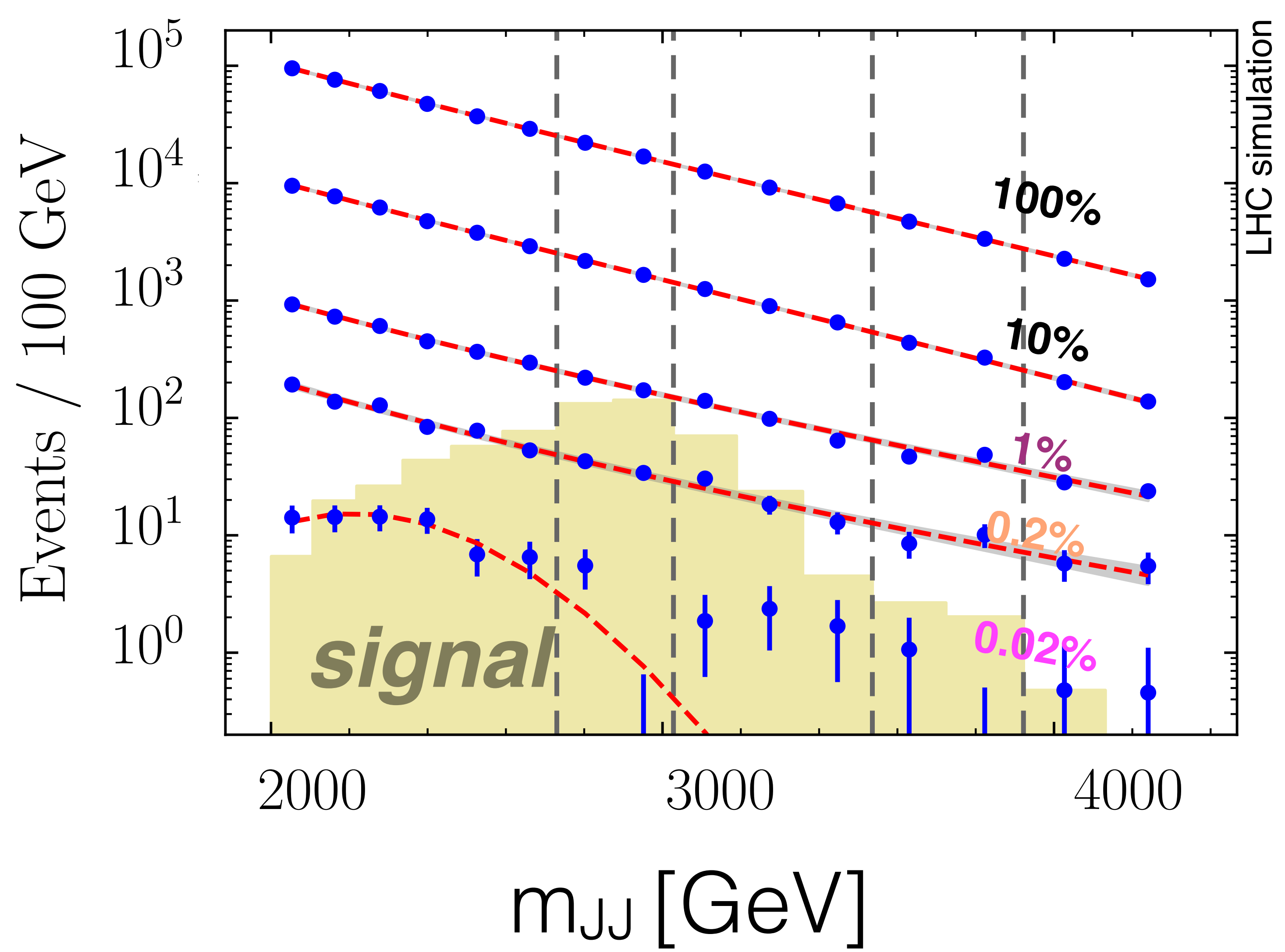
…and when there is a signal?

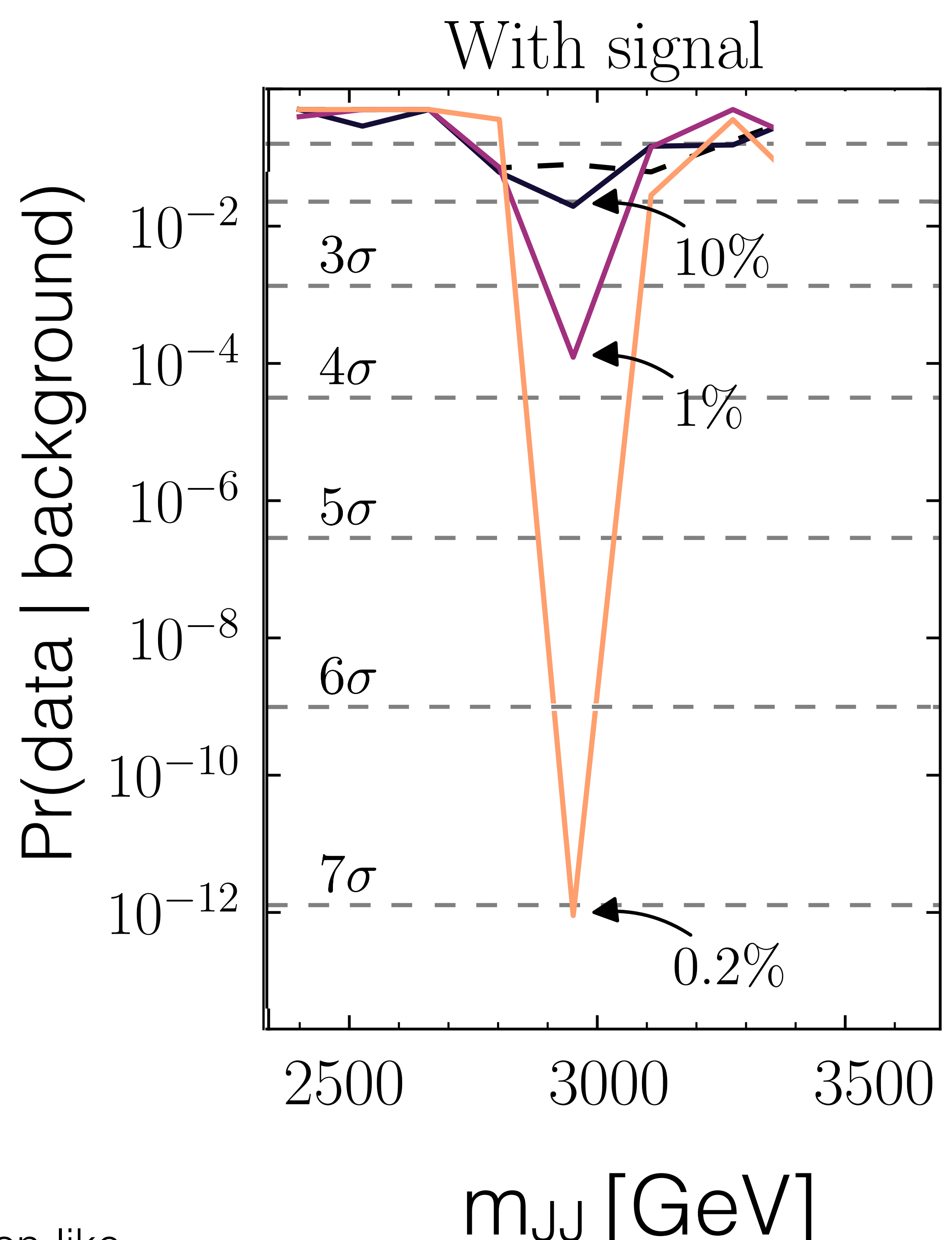




- - - no cut on NN
— most 10% signal-region-like
— most 1% signal-region-like
— most 0.2% signal-region-like

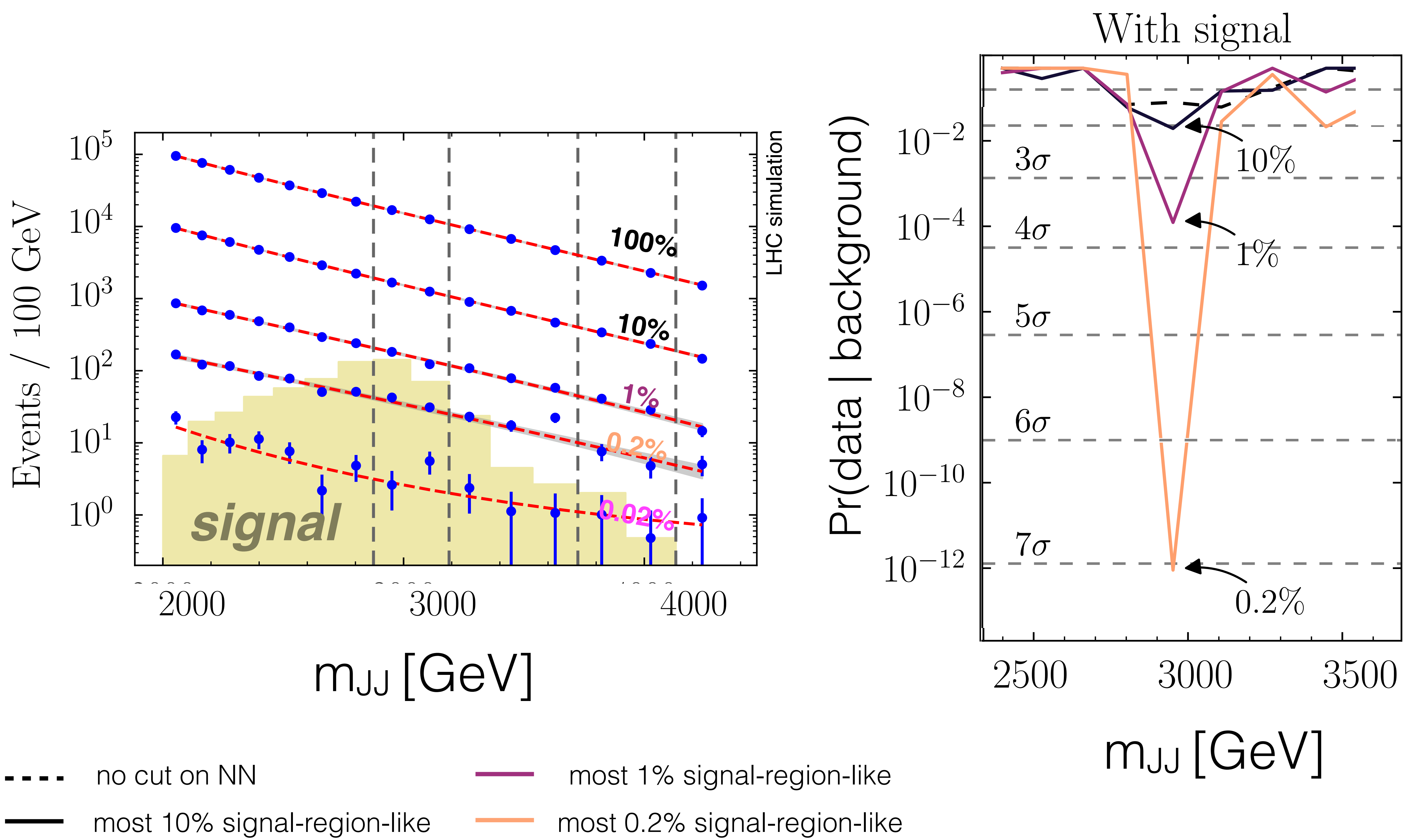
…and when there is a signal?



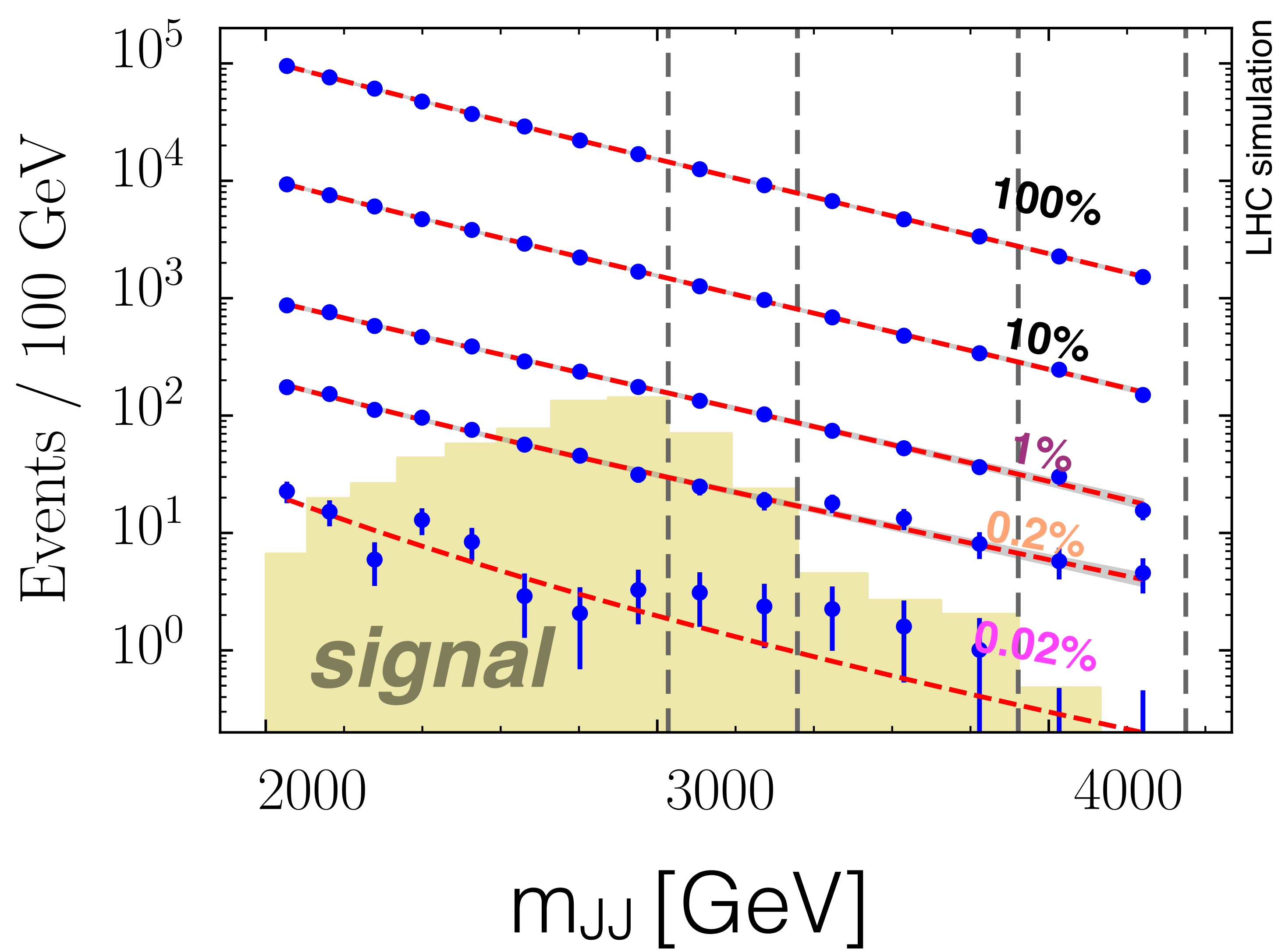


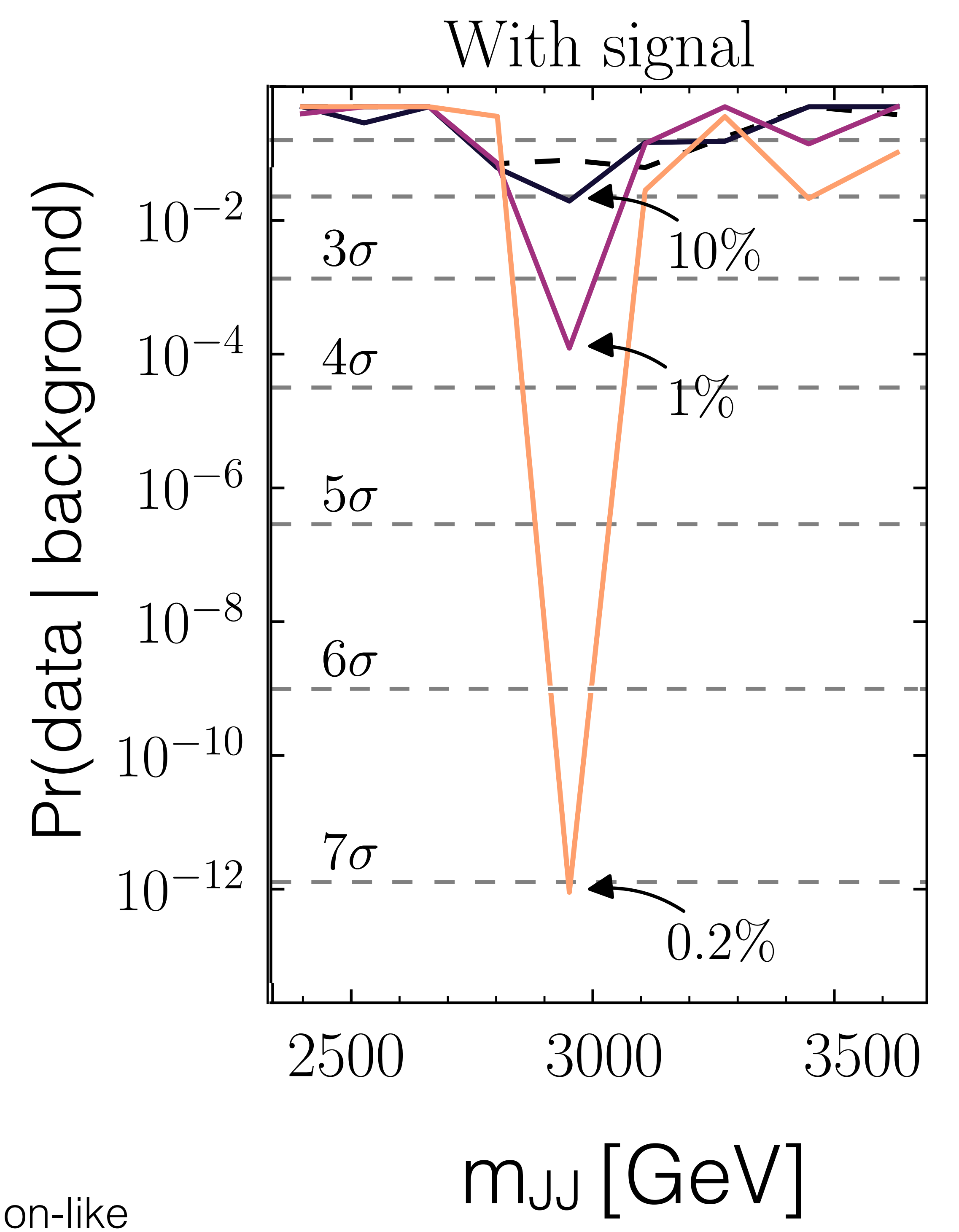
- - - no cut on NN
— most 10% signal-region-like
— most 1% signal-region-like
— most 0.2% signal-region-like

…and when there is a signal?



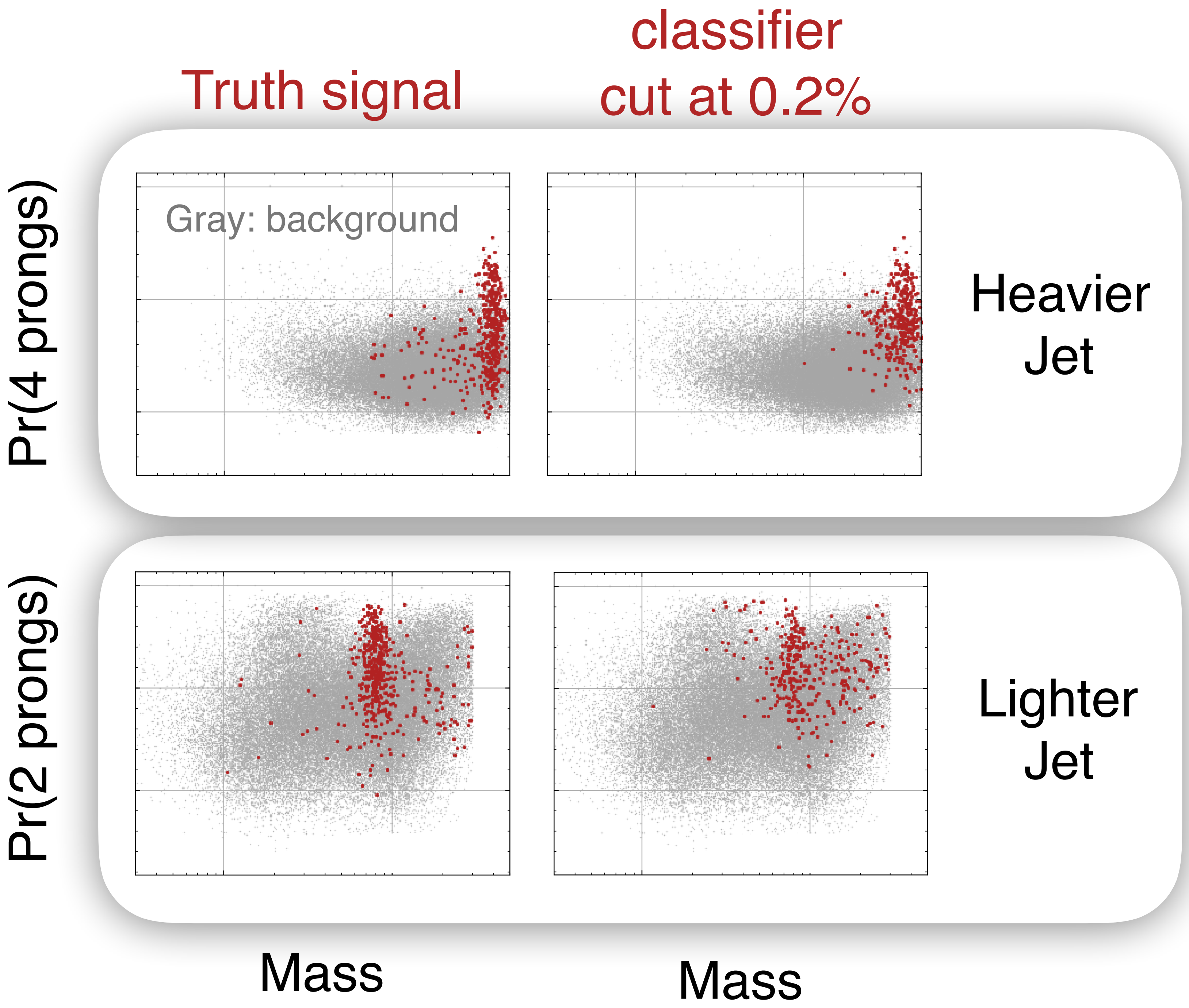
…and when there is a signal?

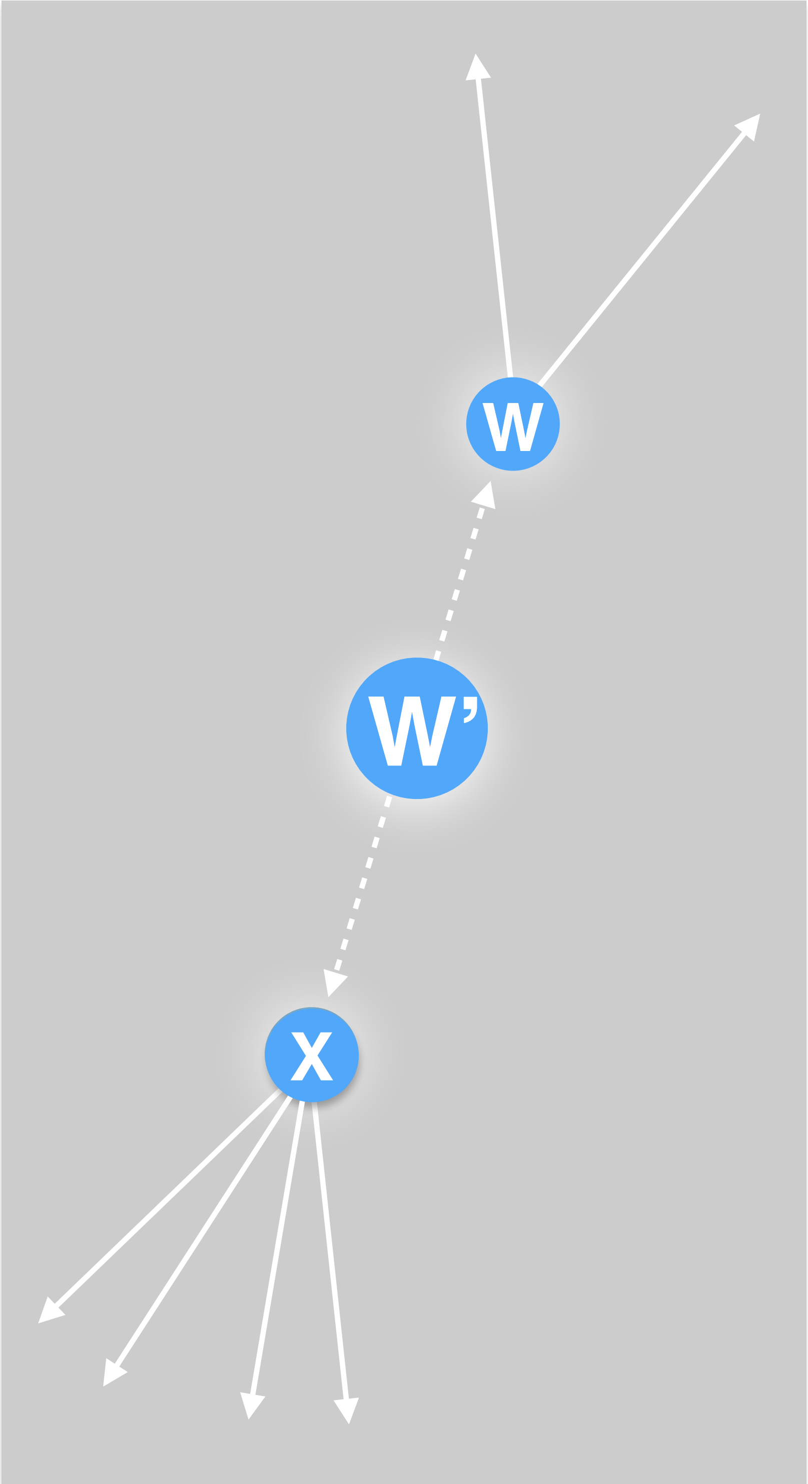




- - - - no cut on NN
—— most 10% signal-region-like
—— most 1% signal-region-like
—— most 0.2% signal-region-like

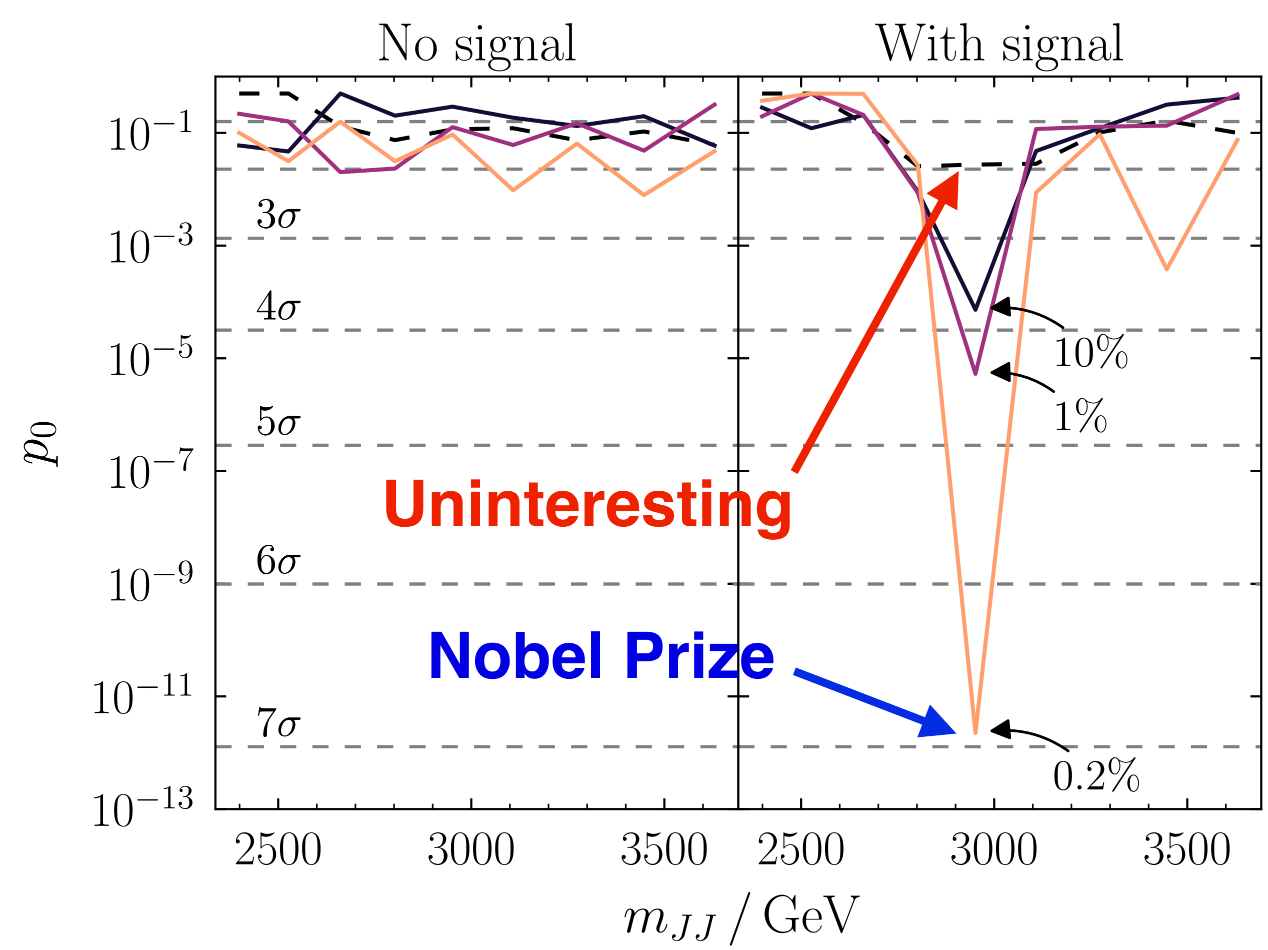
What is the network learning?





Learns to find the signal !

Outlook



Deep learning has a great potential to **enhance**, **accelerate**, and **empower** HEP analyses.

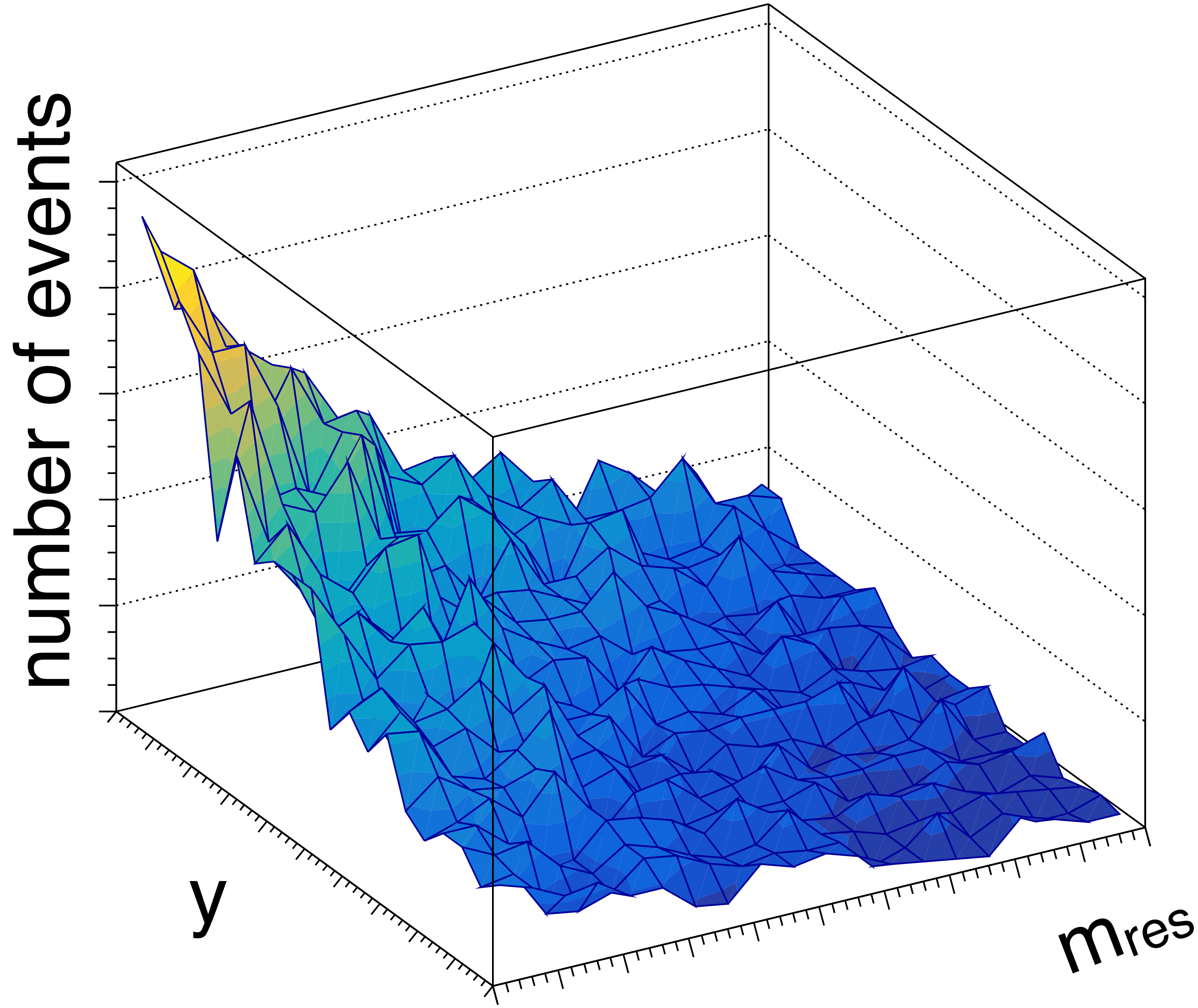
Today, I have told you about an application for anomaly detection at the LHC. I would love to hear your thoughts on the applicability to DM/Neutrinos!!

Backup

Overtraining & Look Elsewhere Effect*

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

i.e. you will sculpt lots of bumps!

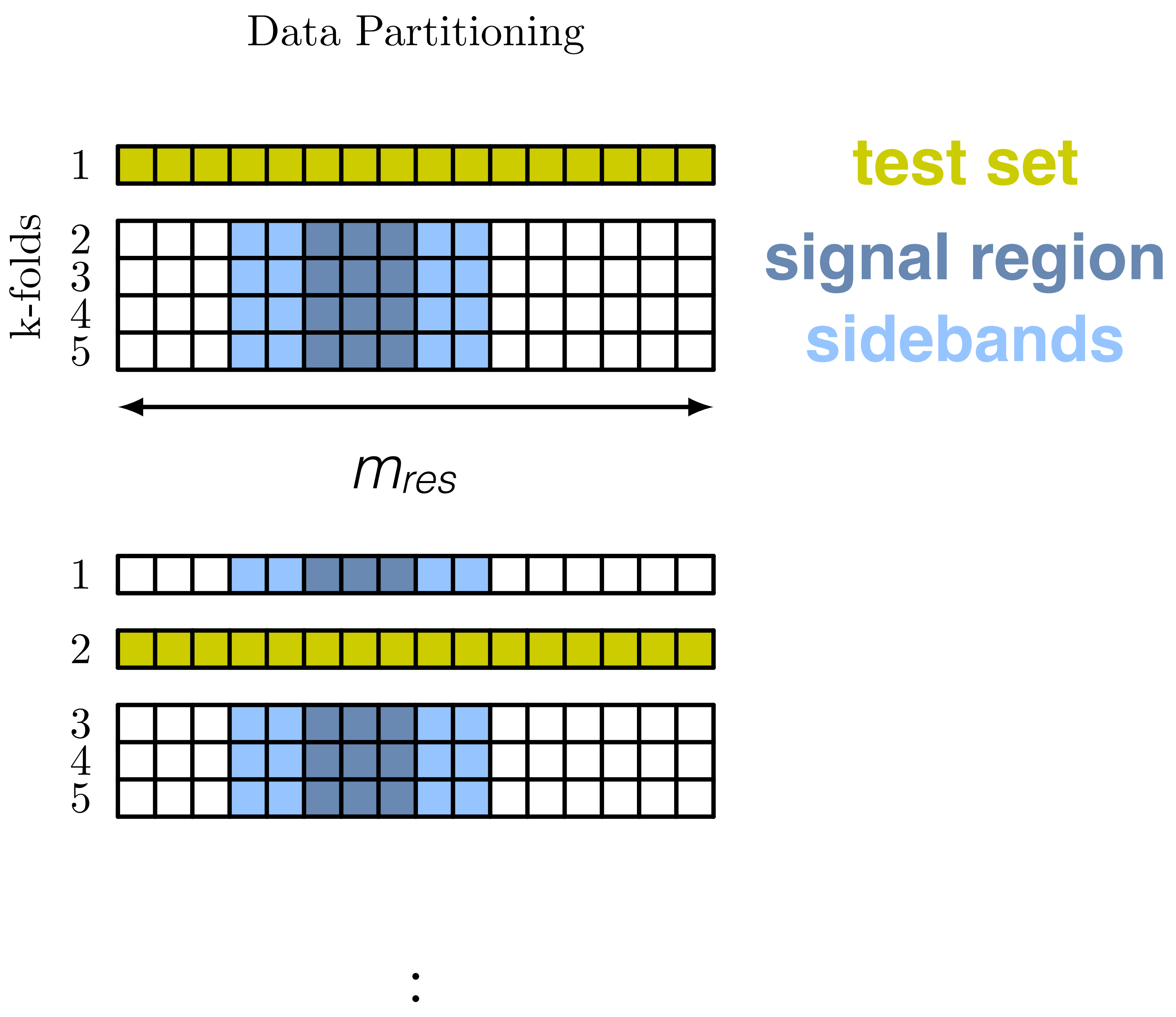


*you may know this as the multiple comparisons problem

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

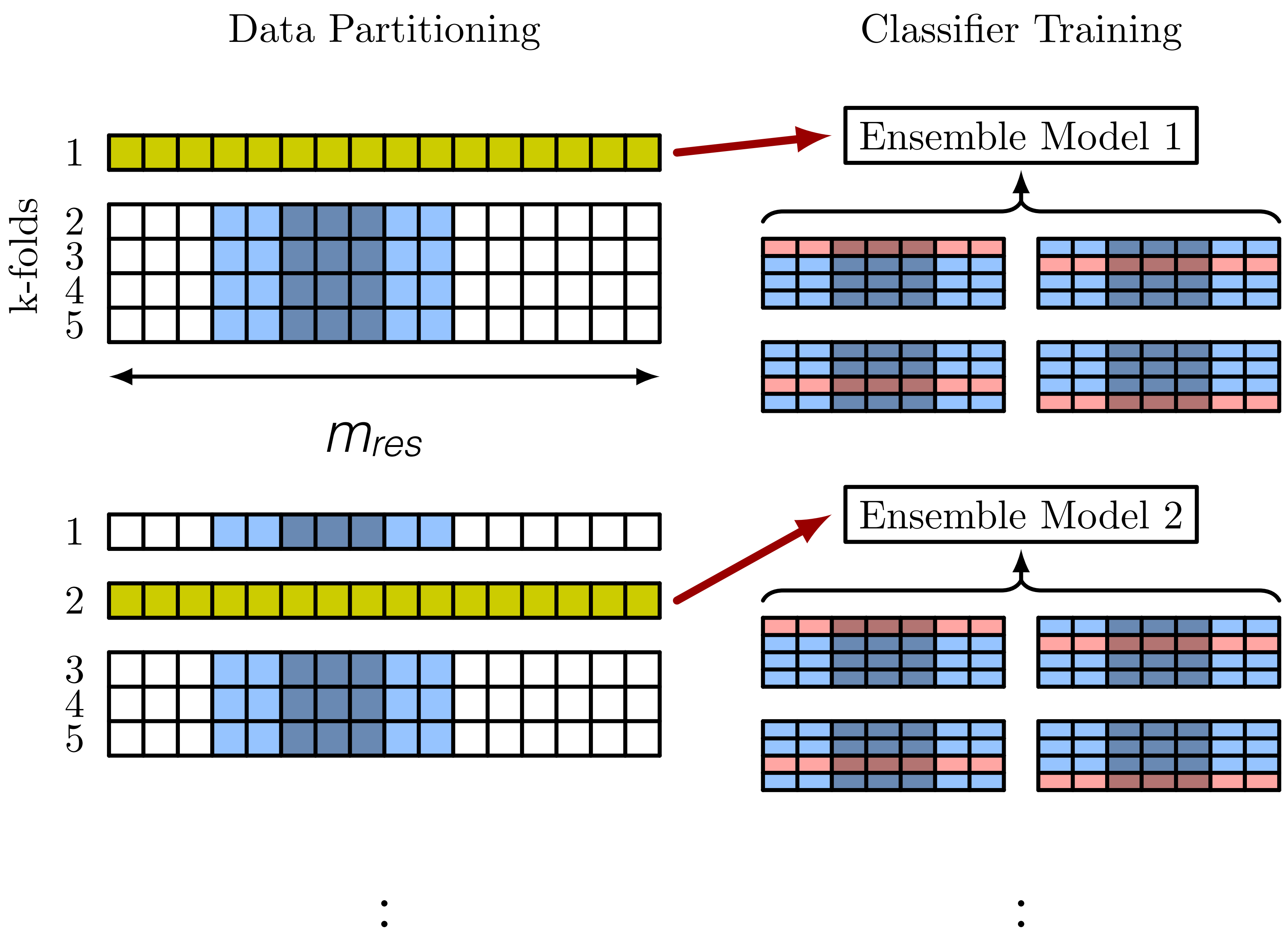
Nested cross-training

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



Nested cross-training

(2) Train CWoLa classifiers.



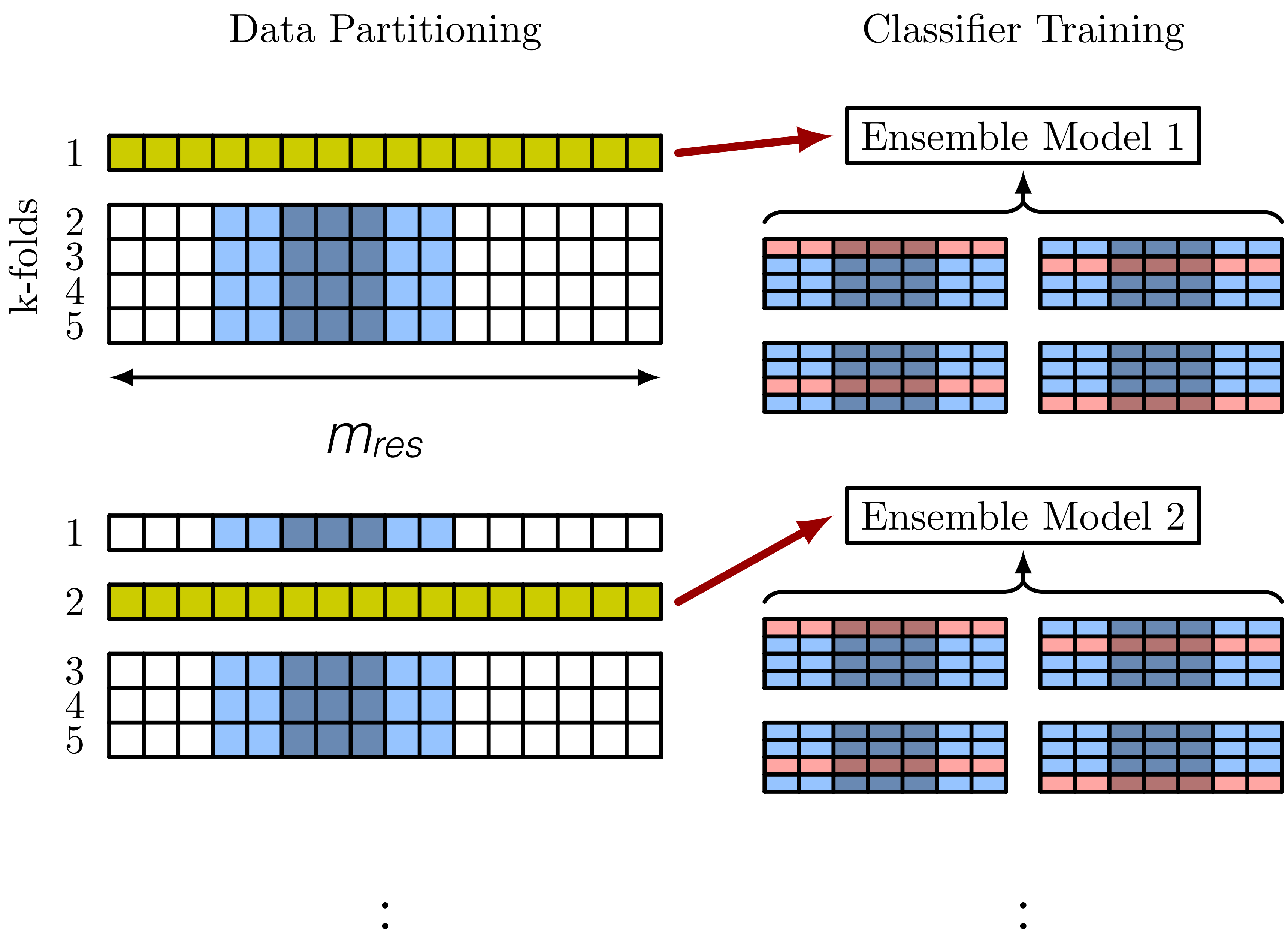
Train signal versus sideband k-1 times

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.

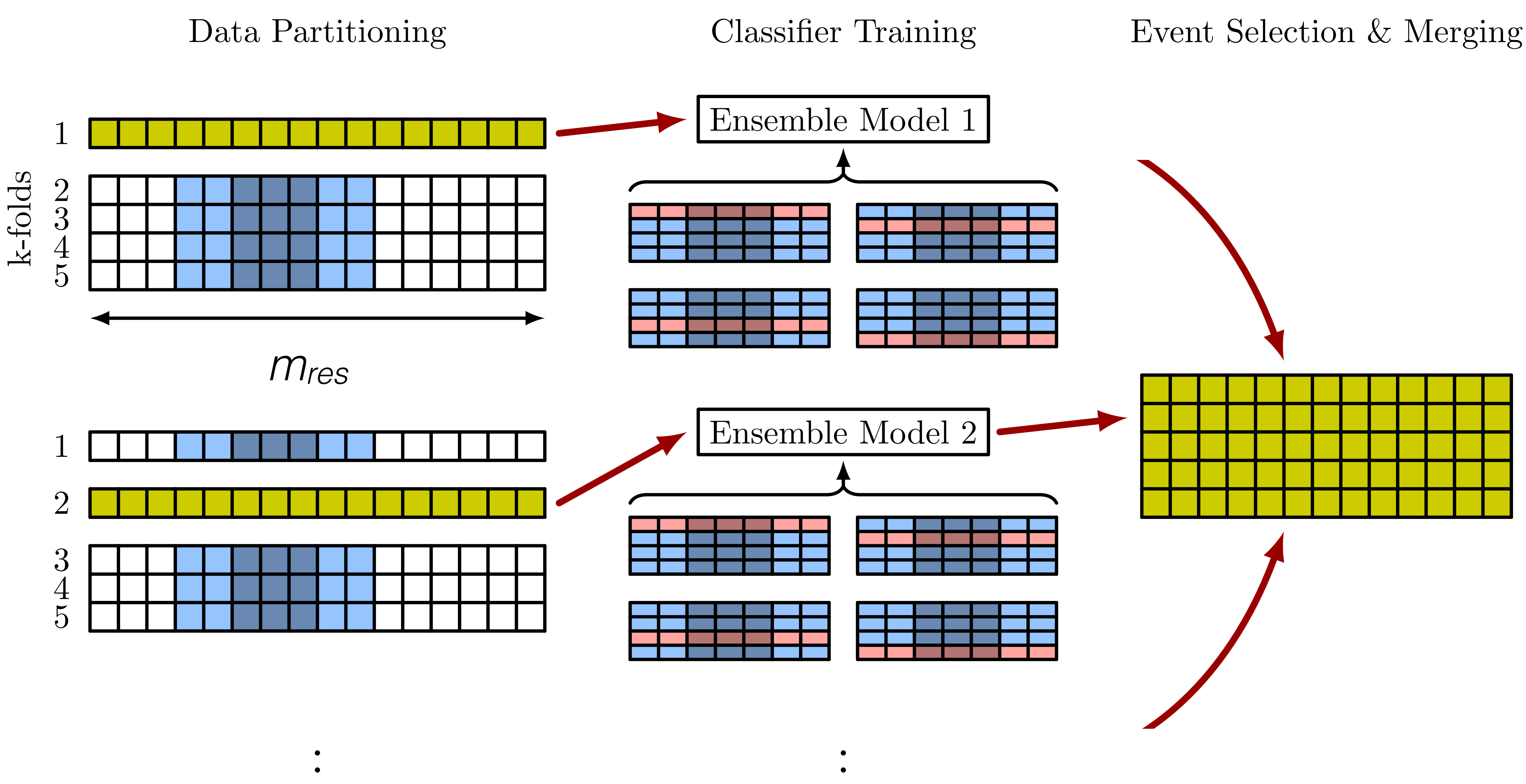


The $\boxed{\text{Ensemble Model}}$ is just the average of the four networks.

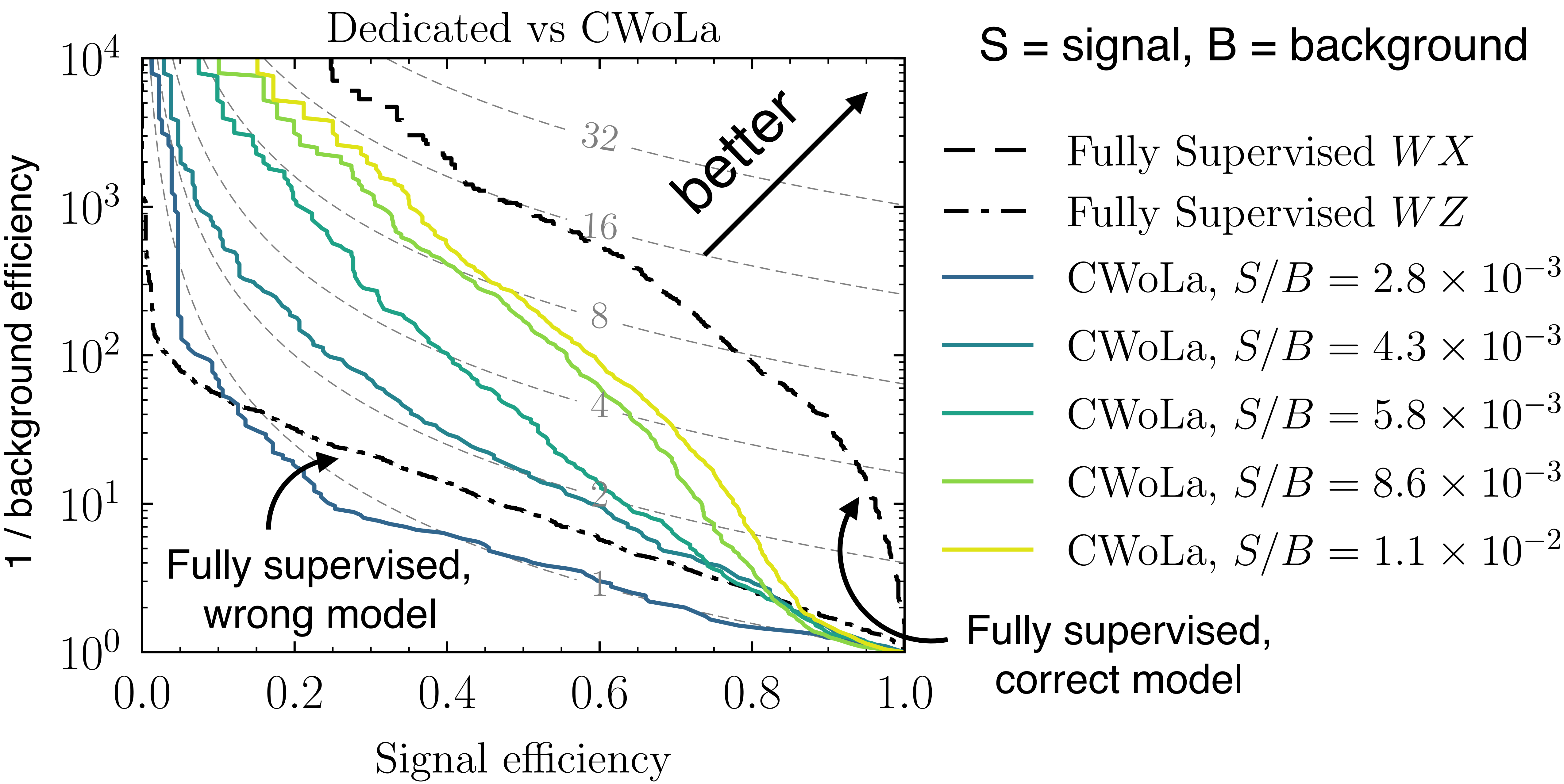
Data fluctuations will cancel destructively while signal interferes constructively.

Nested cross-training

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



CWoLa hunting vs. Full Supervision

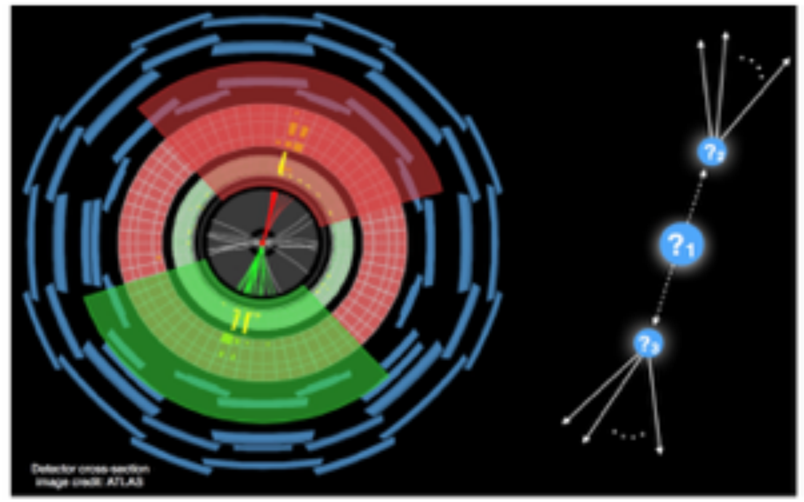


If you know what you are looking for, you should look for it. If you don't know, then CWoLa hunting may be able to catch it!

LHC Olympics 2020

- Overview
- Timetable
- Participant List
- LHCOlympics2020
- Slack channel

LHCOlympics2020



Despite an impressive and extensive effort by the LHC collaborations, there is currently no convincing evidence for new particles produced in high-energy collisions. At the same time, there has been a growing interest in machine learning techniques to enhance potential signals using all of the available information.

In the spirit of the first LHC Olympics (circa 2005-2006) [1st, 2nd, 3rd, 4th], we are organizing the 2020 LHC Olympics. Our goal is to ensure that the LHC search program is sufficiently well-rounded to capture "all" rare and complex signals. The final state for this olympics will be focused (generic dijet events) but the observable phase space and potential BSM parameter space(s) are large: all hadrons in the event can be used for learning (be it "cuts", supervised machine learning, or unsupervised machine learning).

For setting up, developing, and validating your methods, we provide background events and a few benchmark signal models. You can download these from [ZENODO LINK]. To help get you started, we have also prepared simple python scripts to read in the data and do some basic processing.

The final test will happen 2 weeks before the ML4Jets2020 workshop. We will release new datasets where the "background" will be similar to but not identical to the one in the development set (as is true in real data!). Each of these datasets will have signal injected somewhere and it is up to you to see if you can find (a) find a signal (b) what is the mass, and (c) what is the cross section. To keep the scope limited, all signals will be of the form X -> hadrons, where X is a new massive particle with an O(TeV) mass. The events require at least one R = 1.0 jet with $p_T > 1$ TeV. For each event, we provide a list of all hadrons (p_T, eta, phi, p_T, eta, phi, ...) zero-padded up to 600 hadrons.

We strongly encourage you to publish your original research methods using these datasets (before or after) the unveiling of the results. Anyone who participates will be part of a summary paper to be prepared following the workshop.

Please do not hesitate to ask questions: we will use the ML4Jets slack channel to discuss technical questions related to this challenge.

Good luck!

Gregor Kasieczka, Ben Nachman, and David Shih

indico.cern.ch/e/ML4Jets2020