Building Robust AI/ML Methods for High Energy Physics

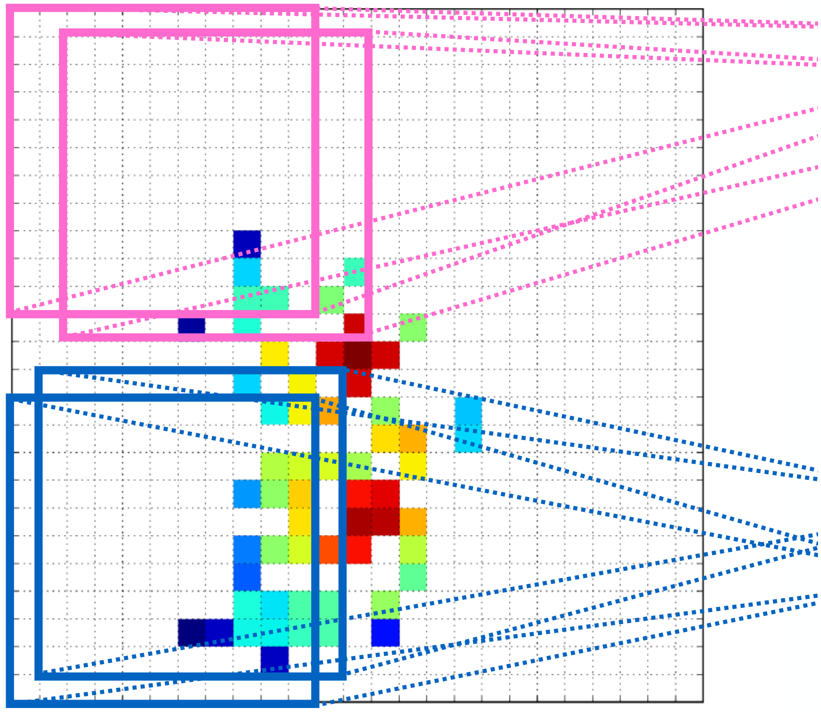
Benjamin Nachman

Lawrence Berkeley National Laboratory

bpnachman.com

bpnachman@lbl.gov

@bpnachman bnachman



BERKELEY LAB

BIDS BERKELEY INSTITUTE FOR DATA SCIENCE

STAMPS-CMU

October 14, 2022

Questions in high energy physics

Theoretical and experimental questions motivate a deep exploration **of the fundamental structure of nature**

Why is the Higgs boson so light?

Hierarchy problem

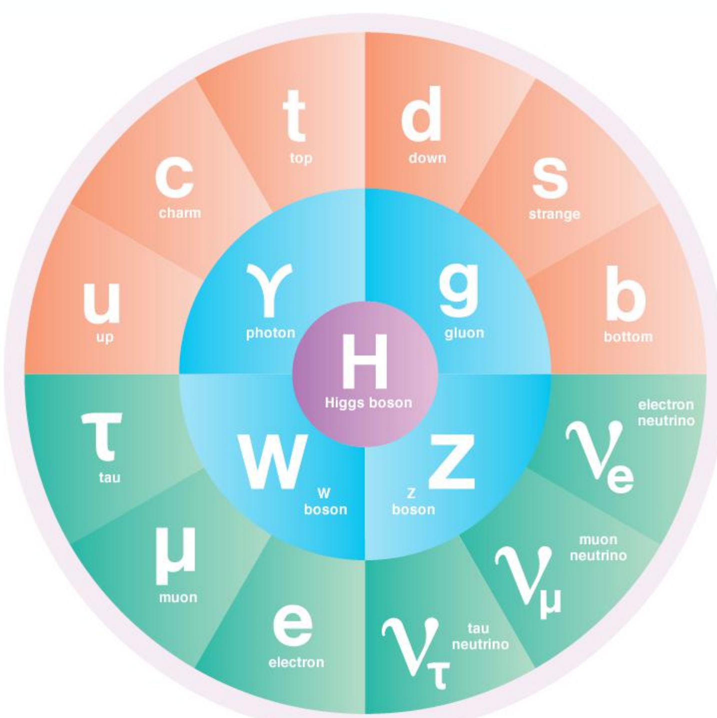


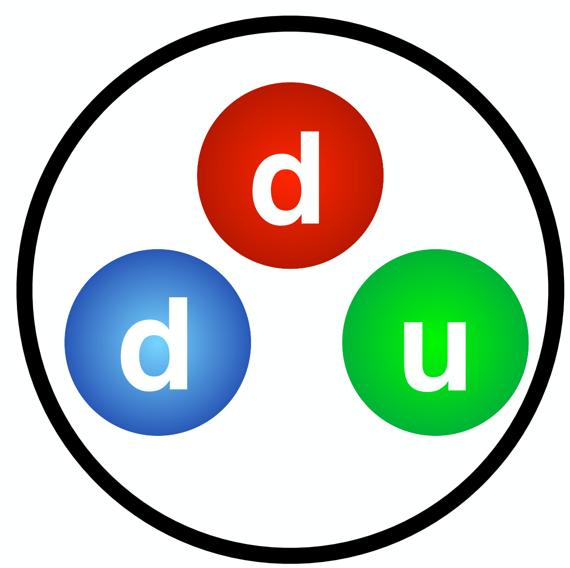
image source: symmetry magazine

See also: quantum gravity

Why do neutrons have no dipole moment?

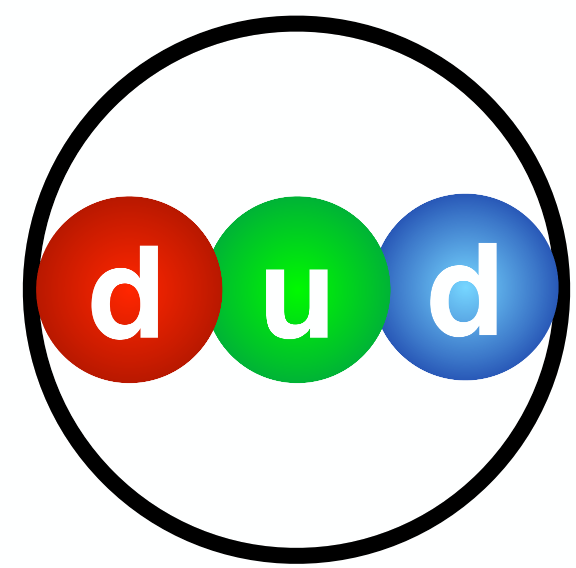
Strong CP

Neutron



>99% of pictures on the internet

Neutron



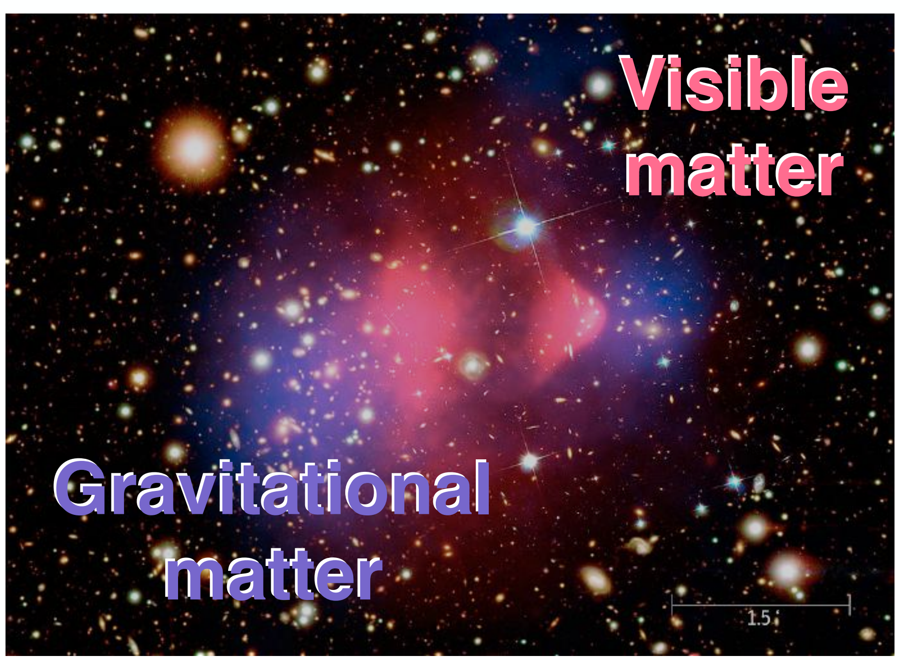
Reality

Questions in high energy physics

Theoretical and **experimental** questions motivate a deep exploration **of the fundamental structure of nature**

What is the extra gravitational matter?

Dark Matter



NASA/CXC/M. Weiss - Chandra X-Ray Observatory: 1E 0657-56

See also: dark energy

Why do neutrinos have a mass?

Flavor puzzles

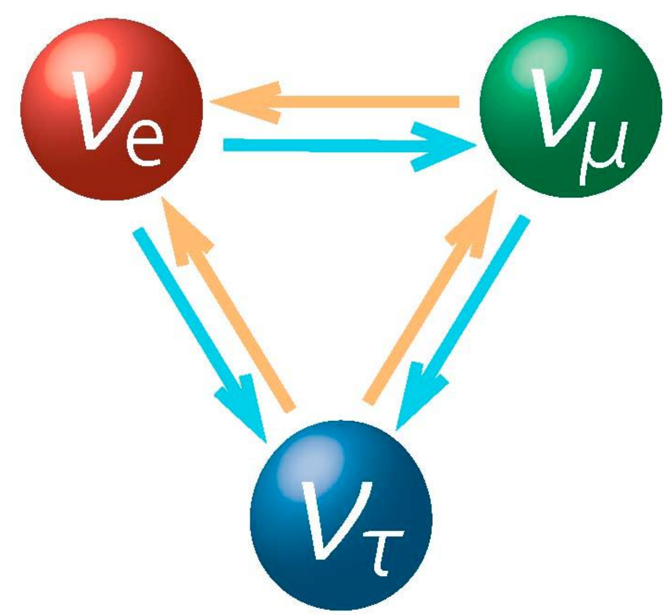


image source: T2K collaboration

See also: Where did all the anti-particles go? (Baryogengesis)

Data analysis in High Energy Physics

Theory of everything	**Nature**
↓	↓
Physics simulators	Experiment
↓	↓
Detector-level observables	Detector-level observables
↓	↓
Pattern recognition ⟷	Pattern recognition

Data analysis in High Energy Physics

+ Machine Learning

Theory of everything

Fast simulation / phase space

Parameter estimation / unfolding

Physics simulators

Detector-level observables

Pattern recognition

Nature

Experiment

Online processing & quality control

Detector-level observables

Pattern recognition

Data curation

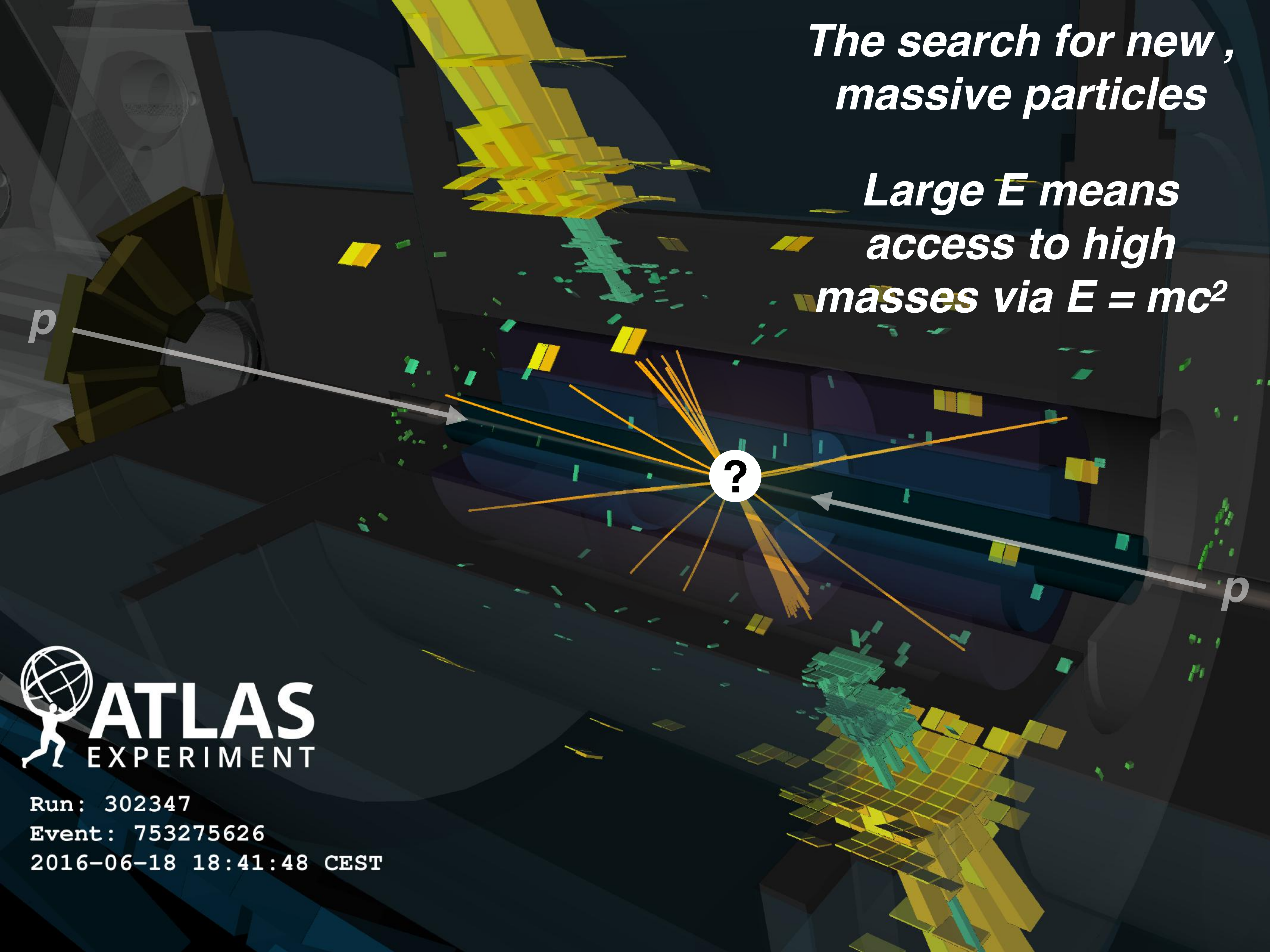
calibration
clustering
tracking
noise mitigation
particle identification
...

Classification to enhance sensitivity

“signal” versus “background”

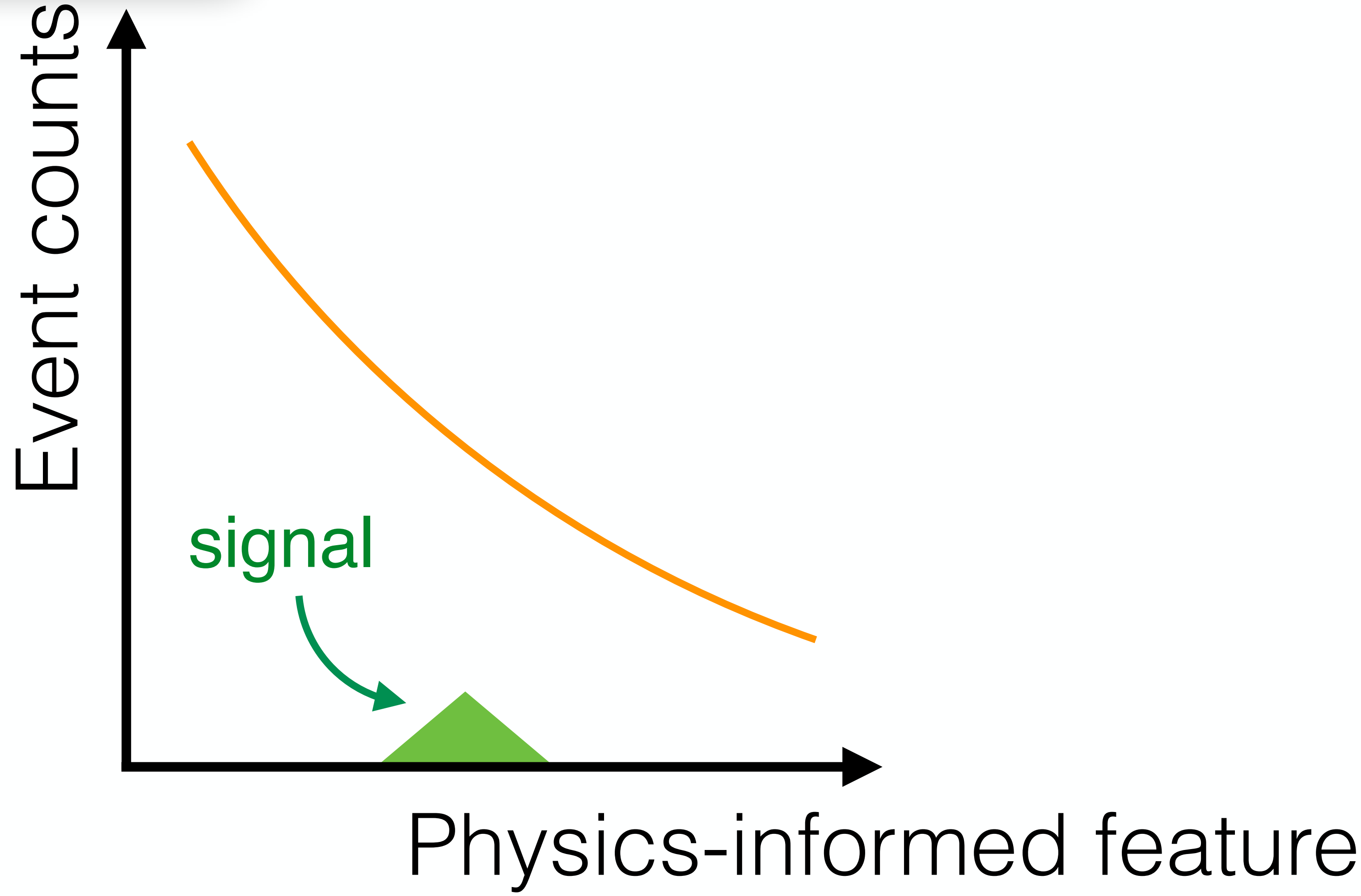
The search for new , massive particles

Large E means access to high masses via $E = mc^2$



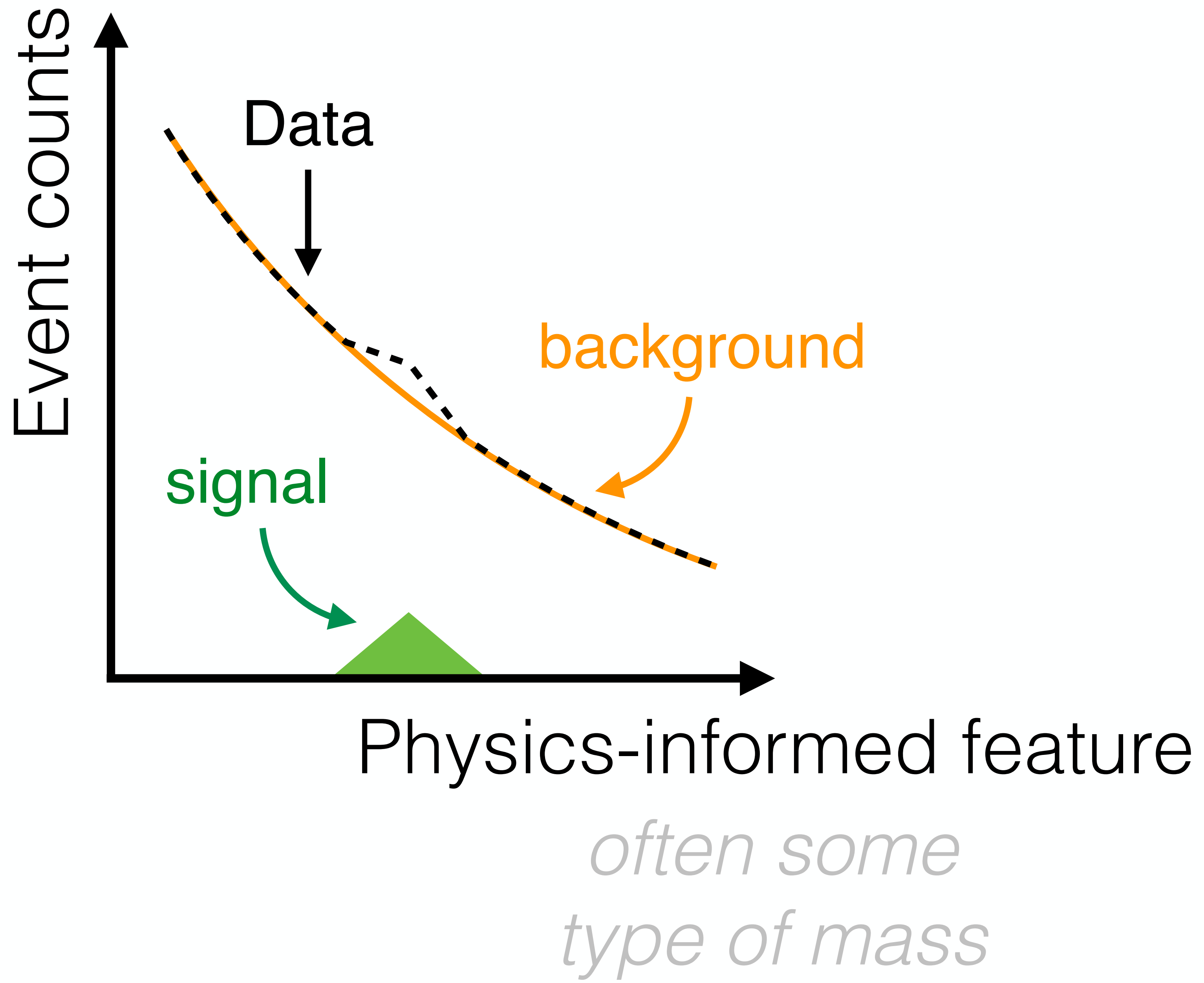
A simplified HEP analysis

High Energy Physics

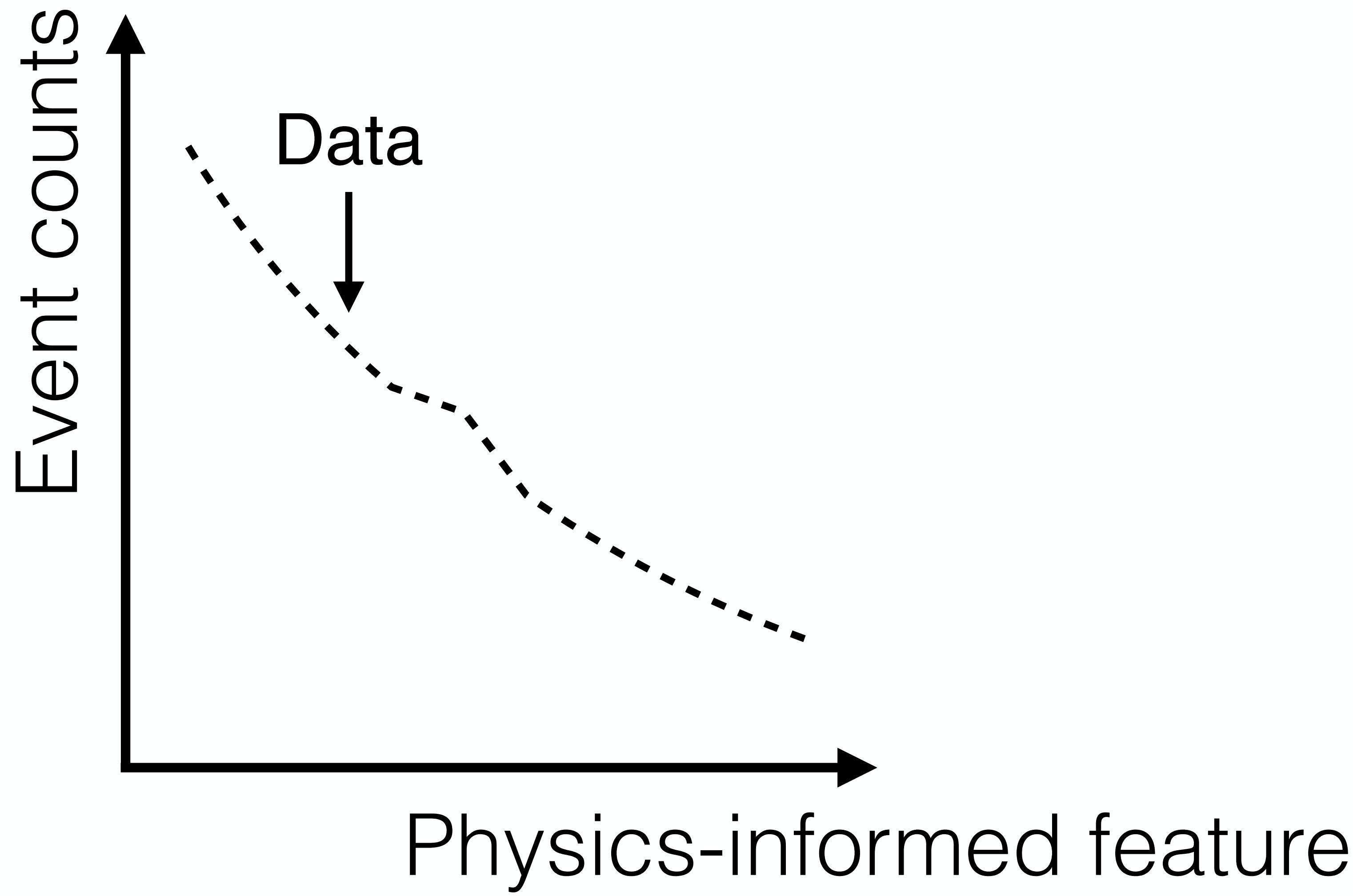


often some type of mass

A simplified HEP analysis

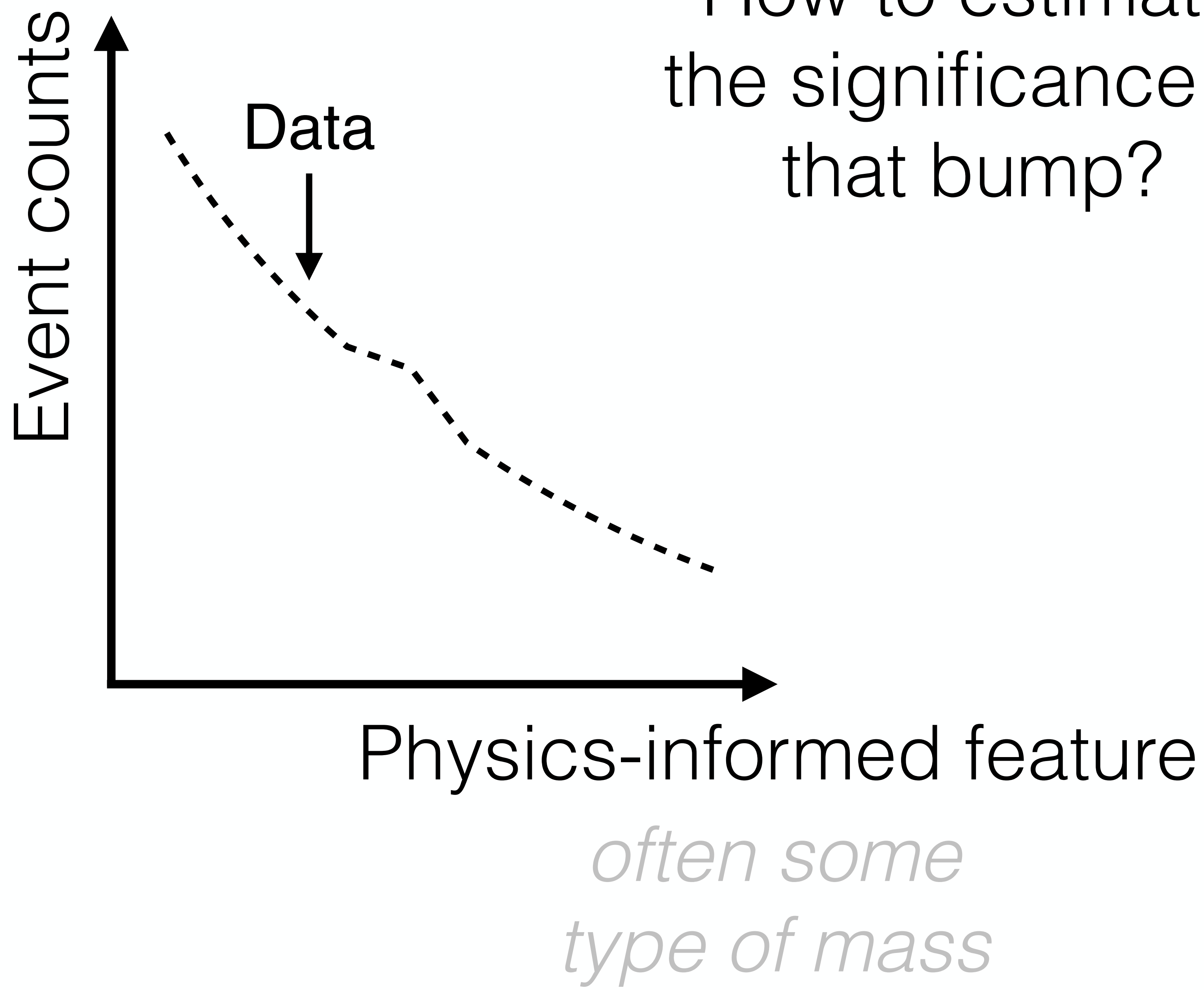


A simplified HEP analysis



often some type of mass

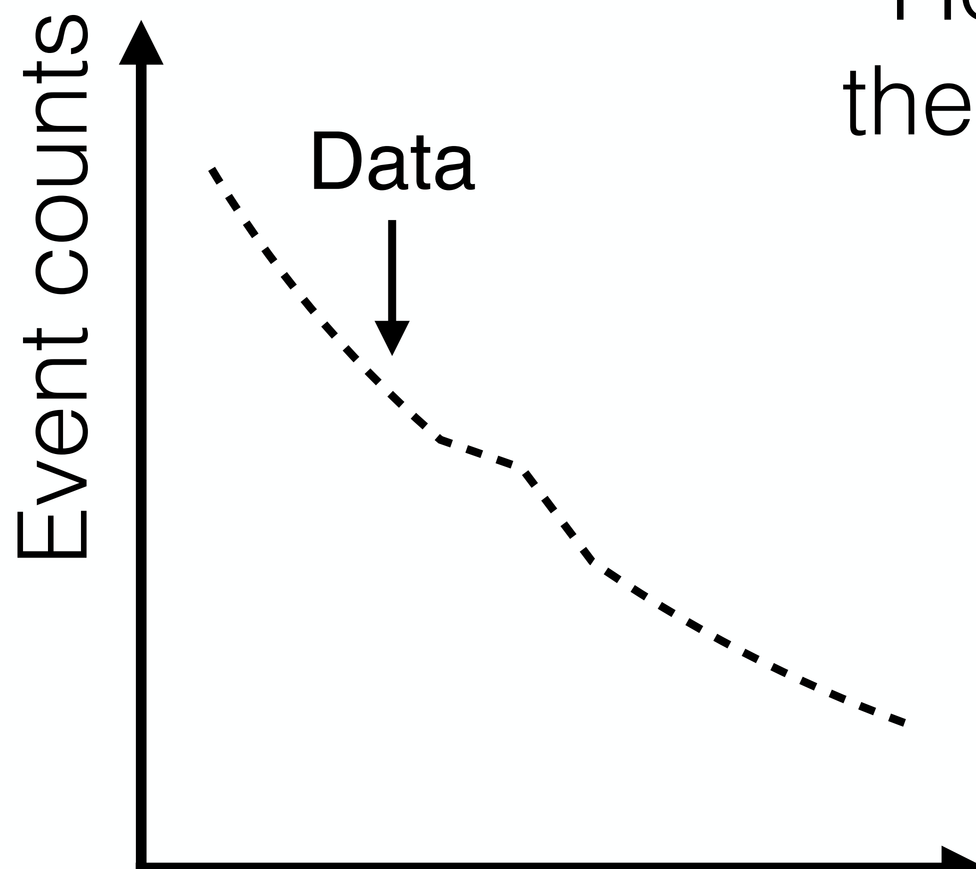
A simplified HEP analysis



How to estimate the significance of that bump?

A simplified HEP analysis

How to estimate
the significance of
that bump?

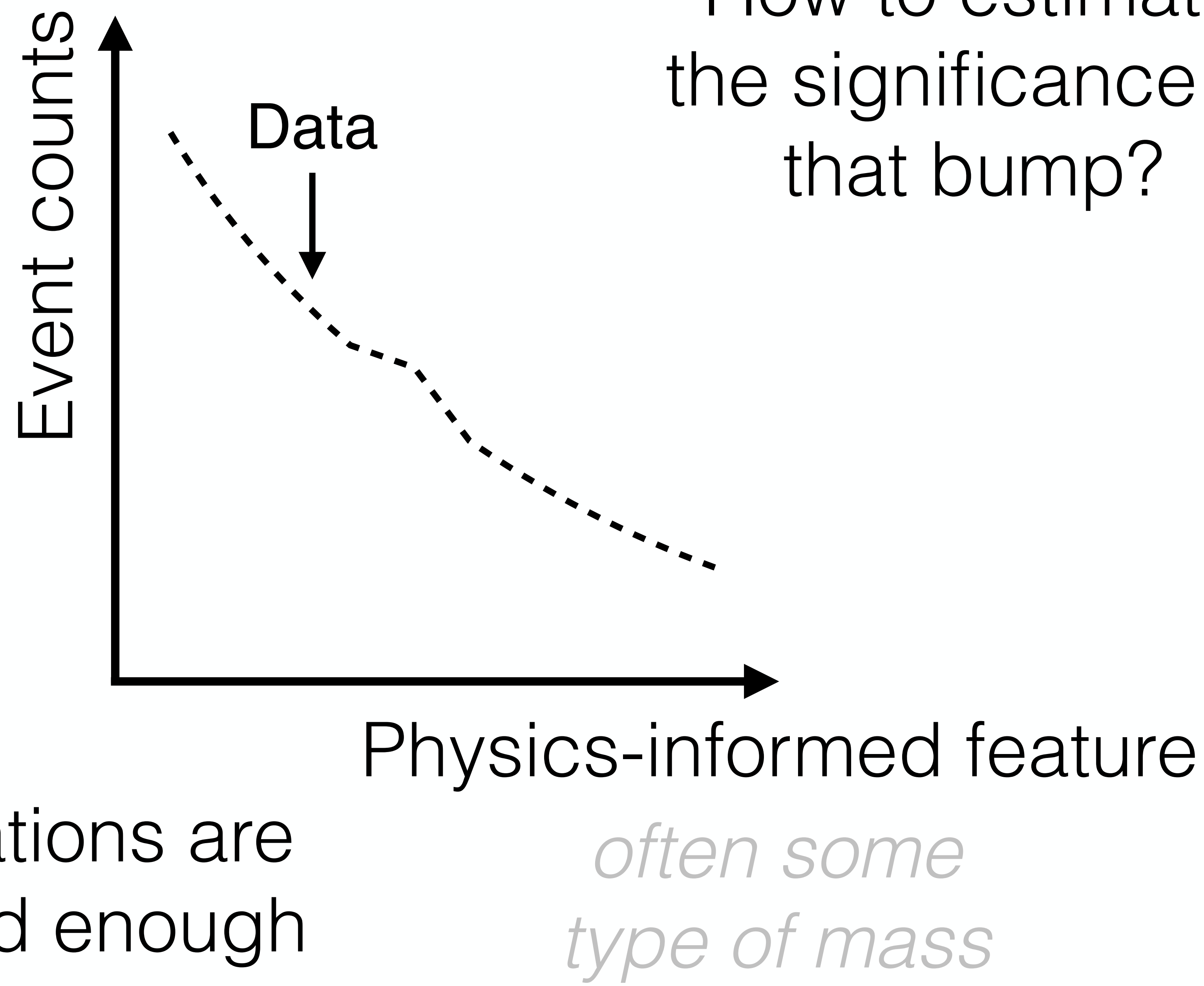


Physics-informed feature

often some type of mass

…often our simulations are good, but not good enough for a precise estimate.

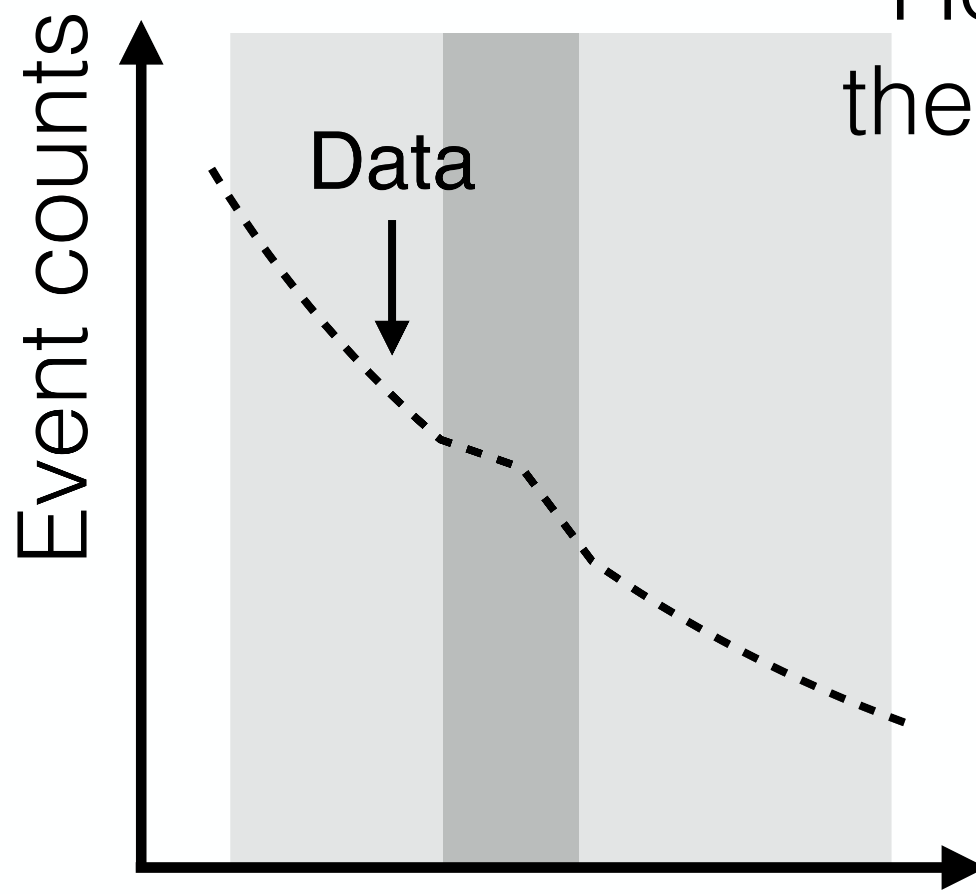
A simplified HEP analysis



How to estimate the significance of that bump?

…often our simulations are good, but not good enough for a precise estimate.
Solution: **learn from data!**

A simplified HEP analysis



How to estimate the significance of that bump?

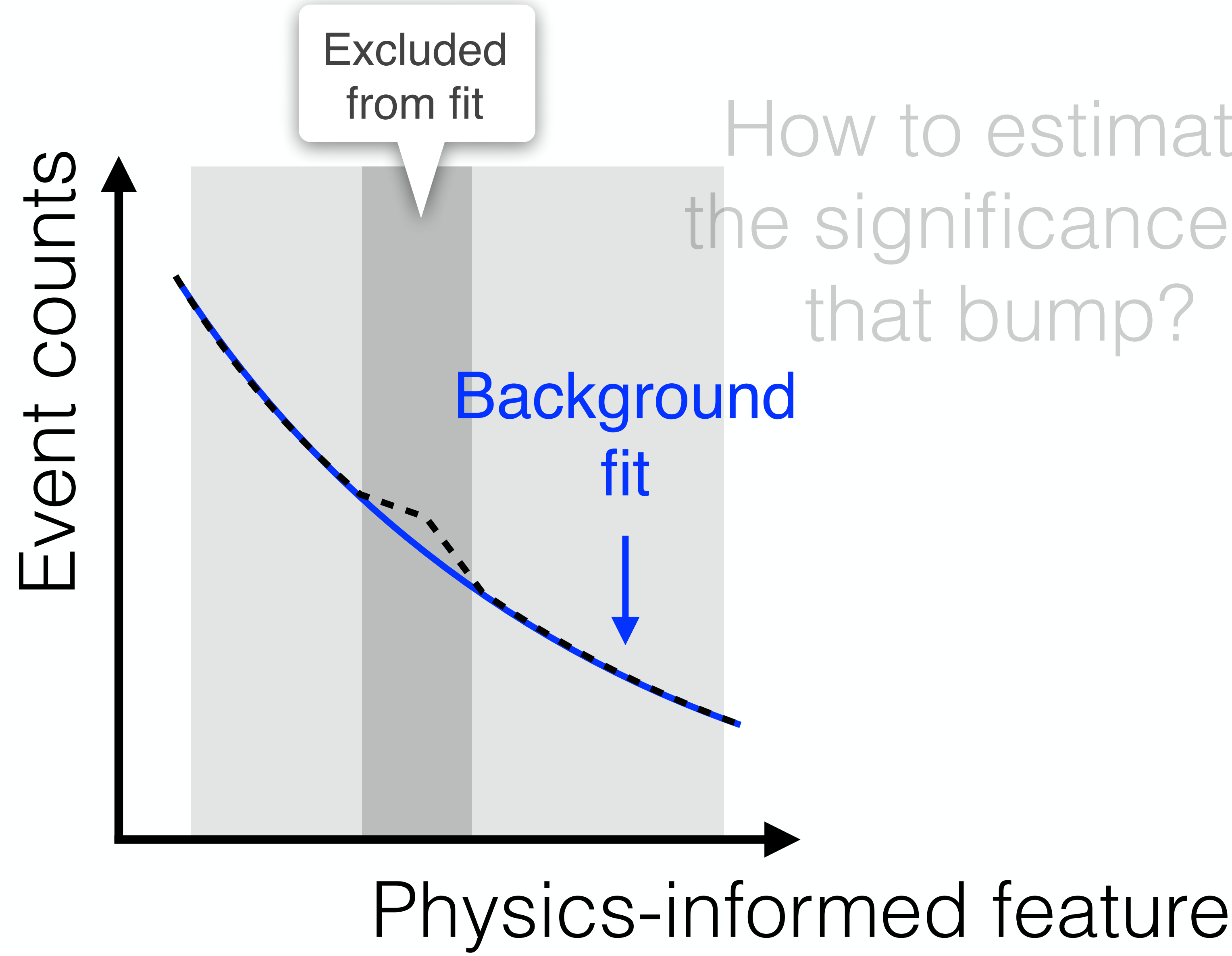
Physics-informed feature

often some type of mass

…often our simulations are good, but not good enough for a precise estimate.

Solution: **learn from data!**

A simplified HEP analysis

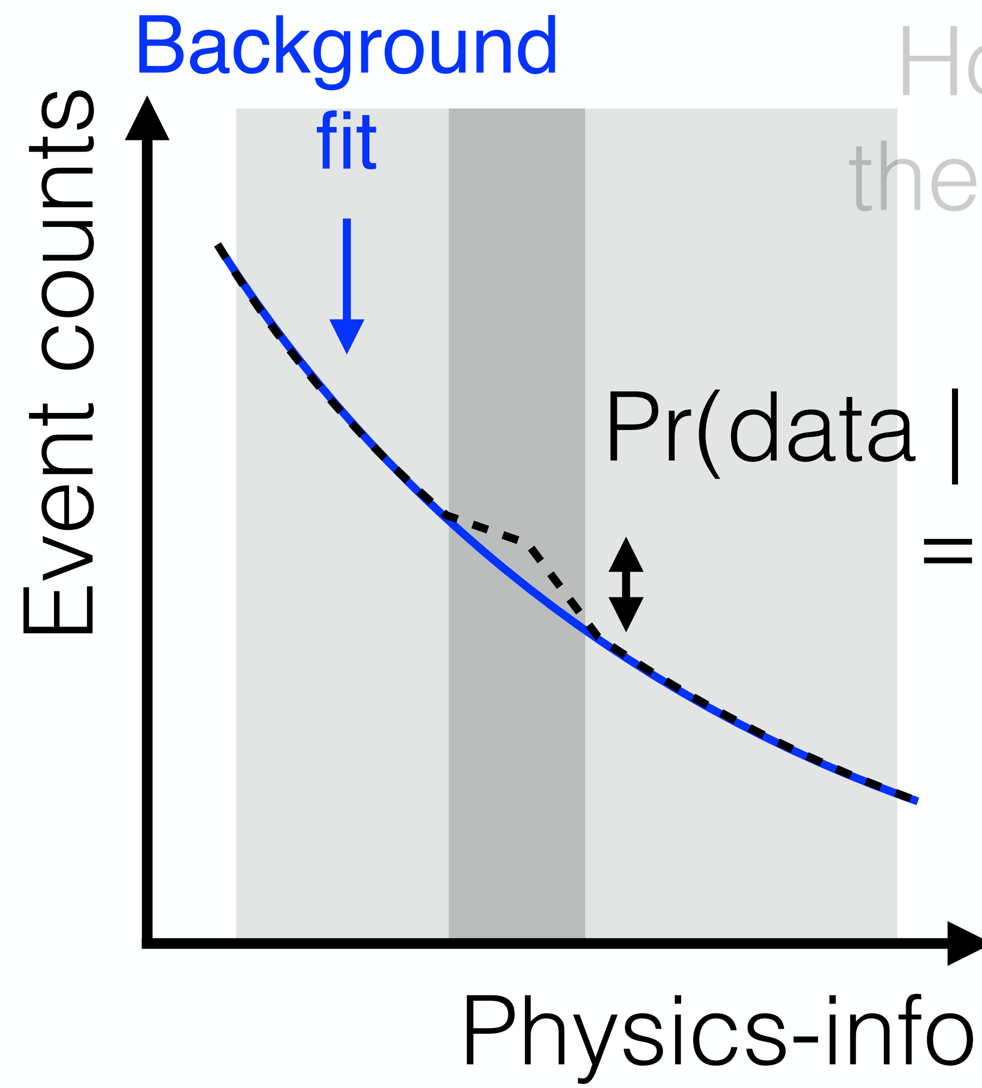


How to estimate the significance of that bump?

often some type of mass

…often our simulations are good, but not good enough for a precise estimate.
Solution: **learn from data!**

A simplified HEP analysis



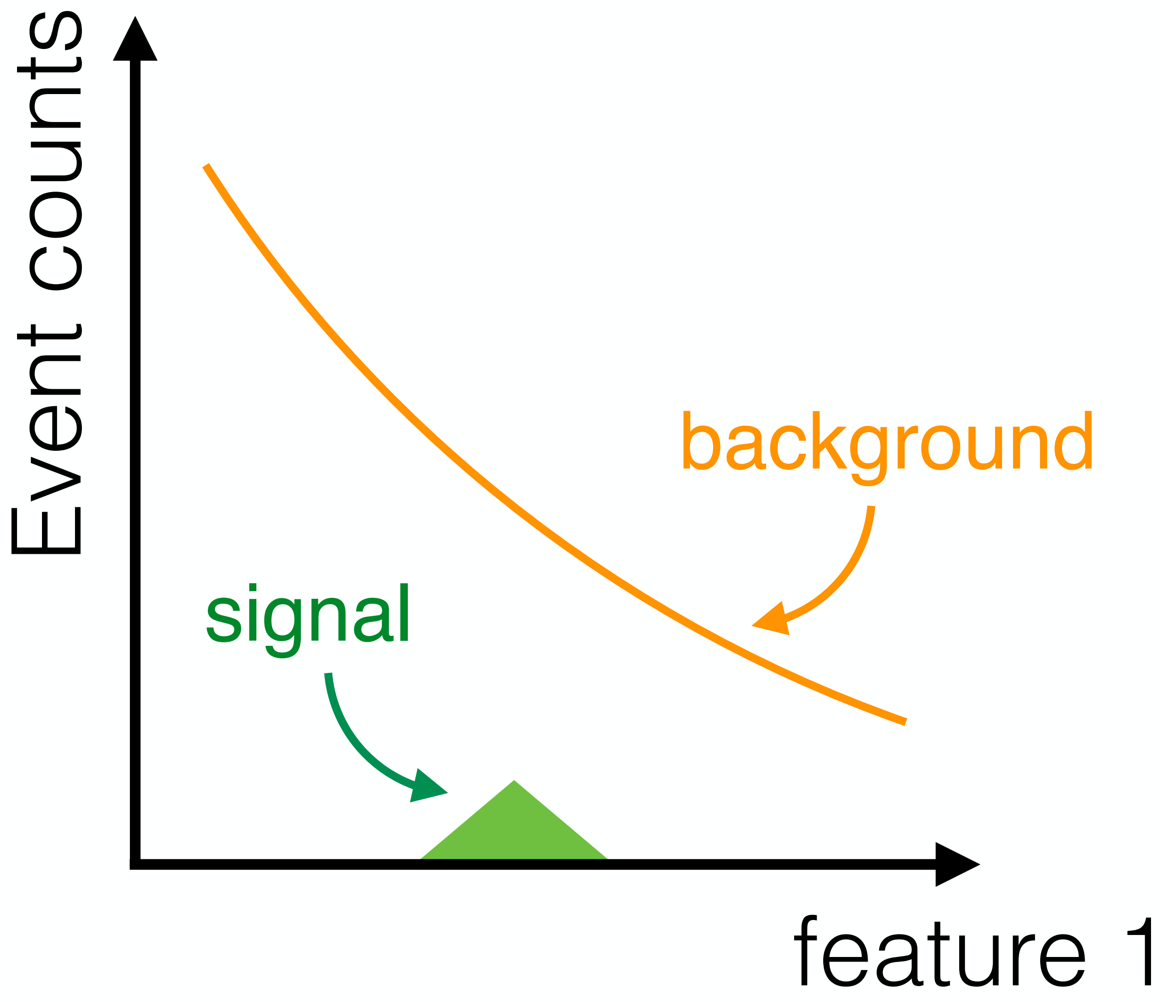
How to estimate the significance of that bump?

Pr(data | background) << 1
= **Discovery**

often some type of mass

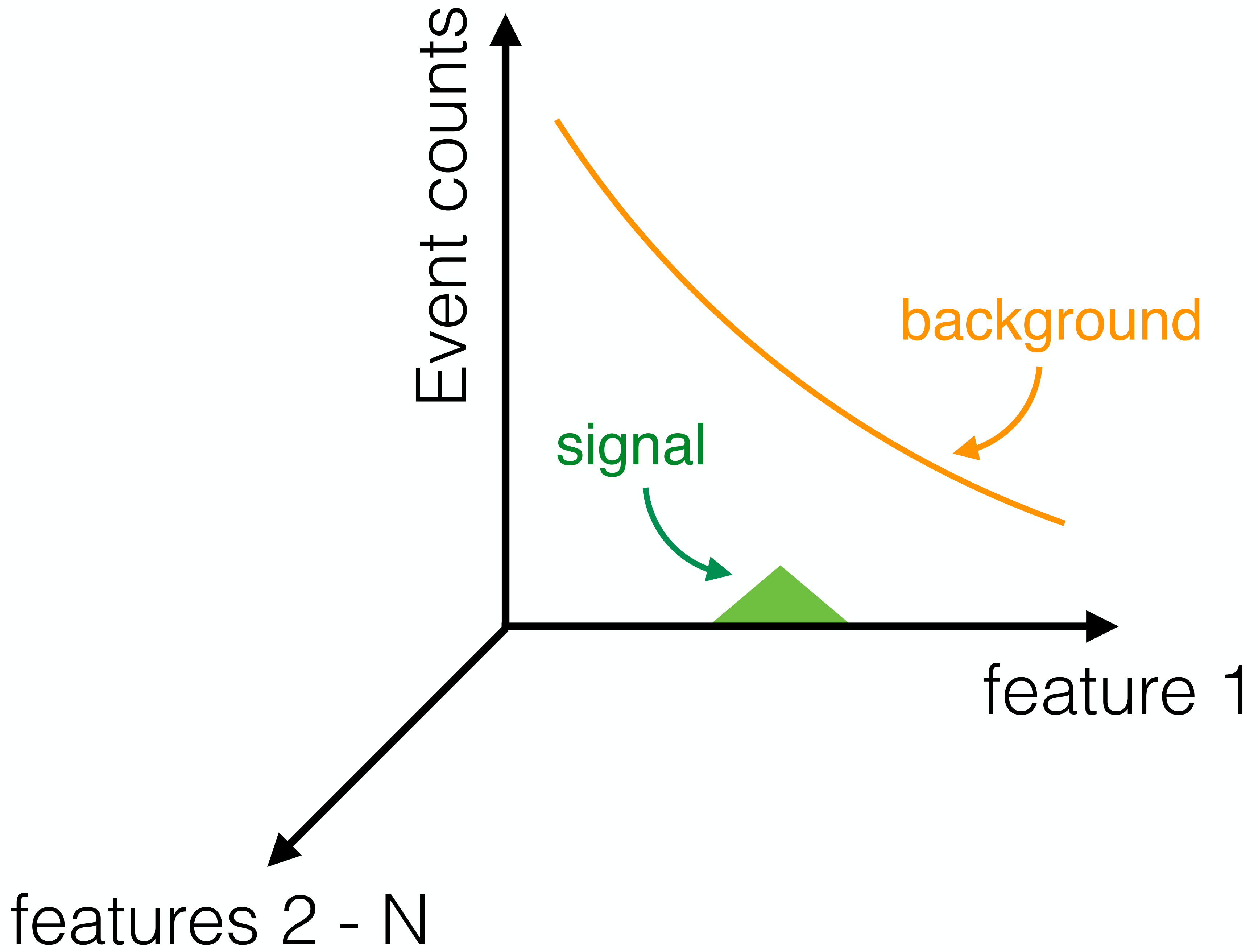
…often our simulations are good, but not good enough for a precise estimate.
Solution: **learn from data!**

A simplified HEP analysis



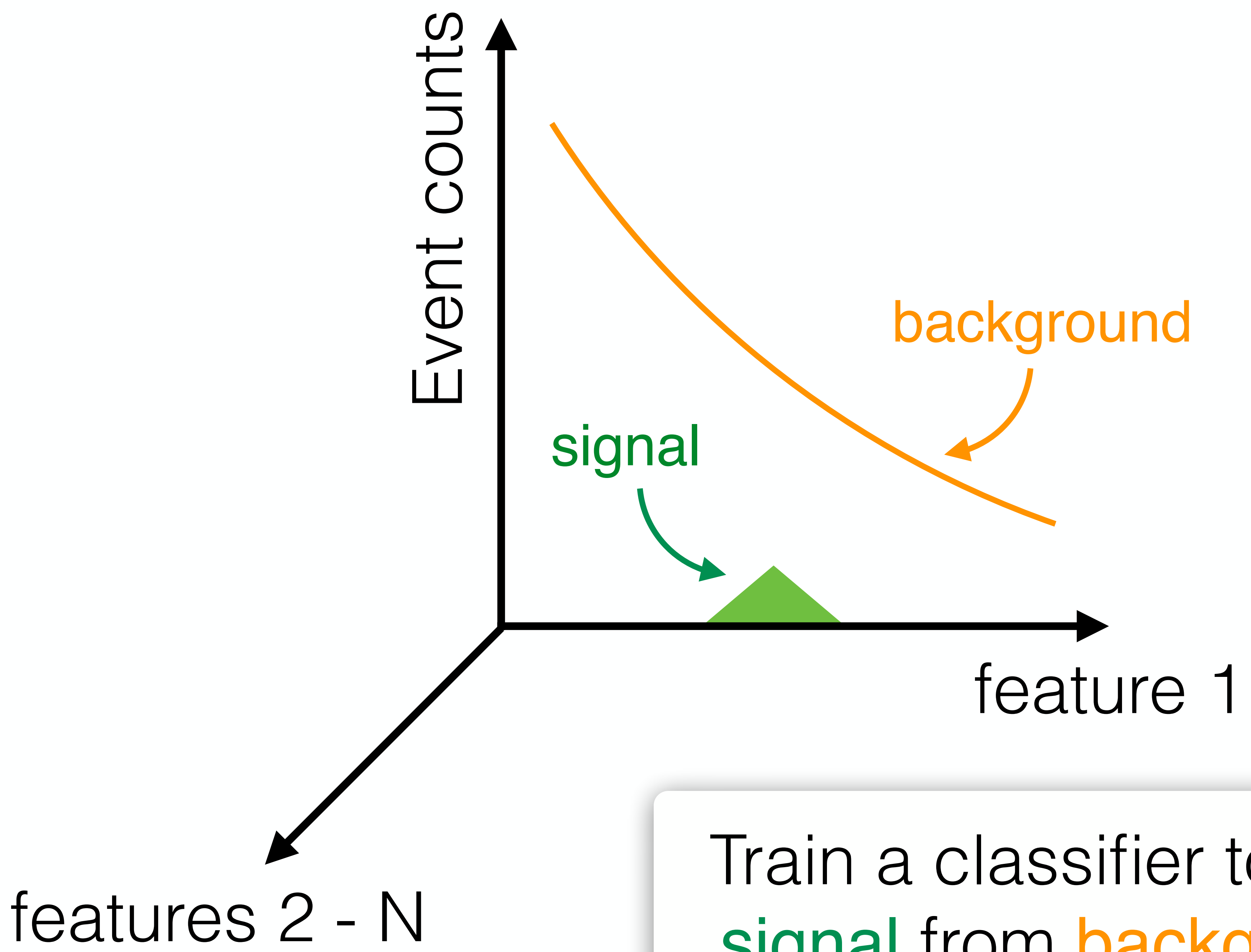
A simplified HEP analysis

+ Machine Learning



A simplified HEP analysis

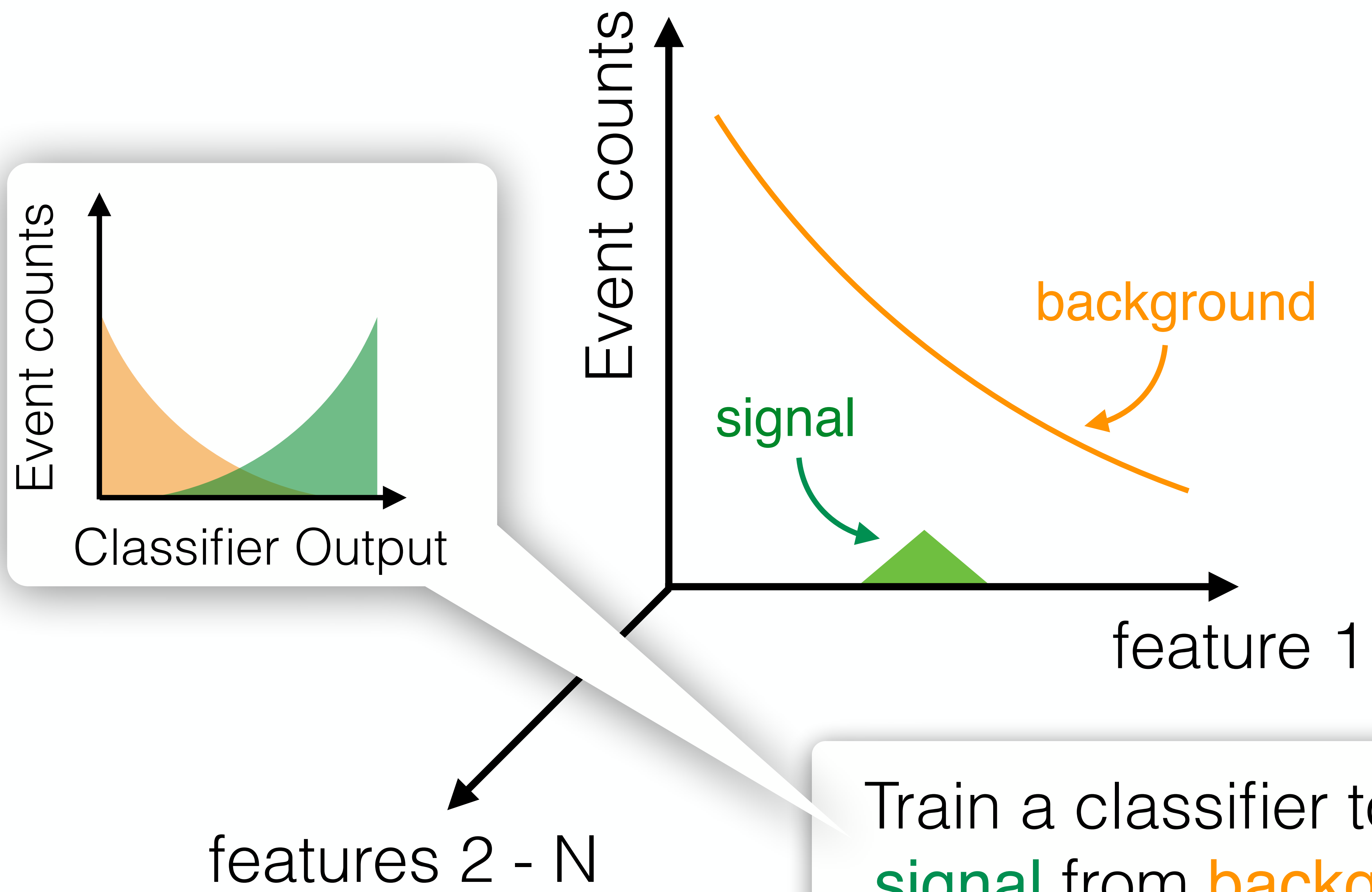
+ Machine Learning



Train a classifier to distinguish signal from background using features 2 - N

A simplified HEP analysis

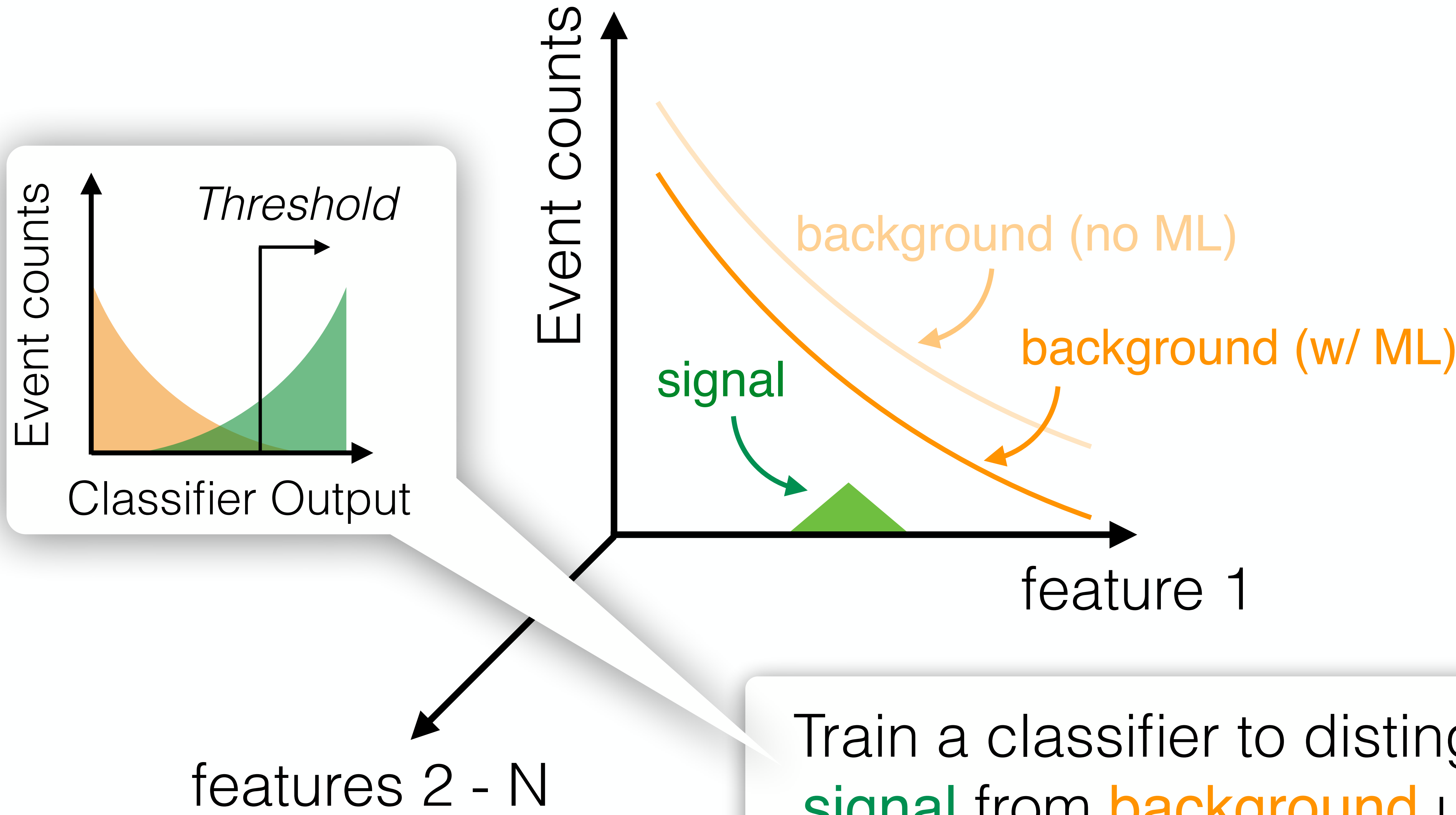
+ Machine Learning



Train a classifier to distinguish signal from background using features 2 - N

A simplified HEP analysis

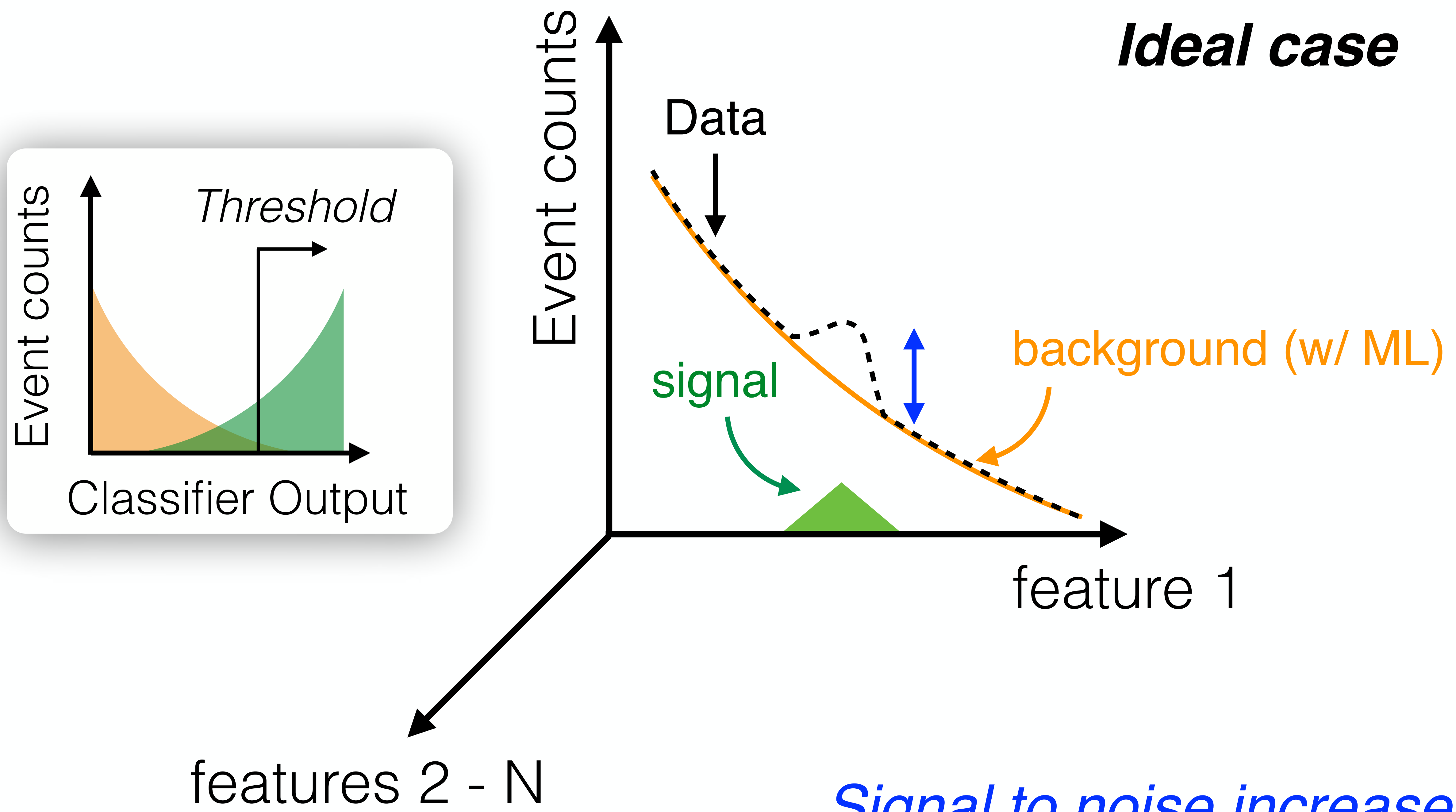
+ Machine Learning



Train a classifier to distinguish signal from background using features 2 - N

A simplified HEP analysis

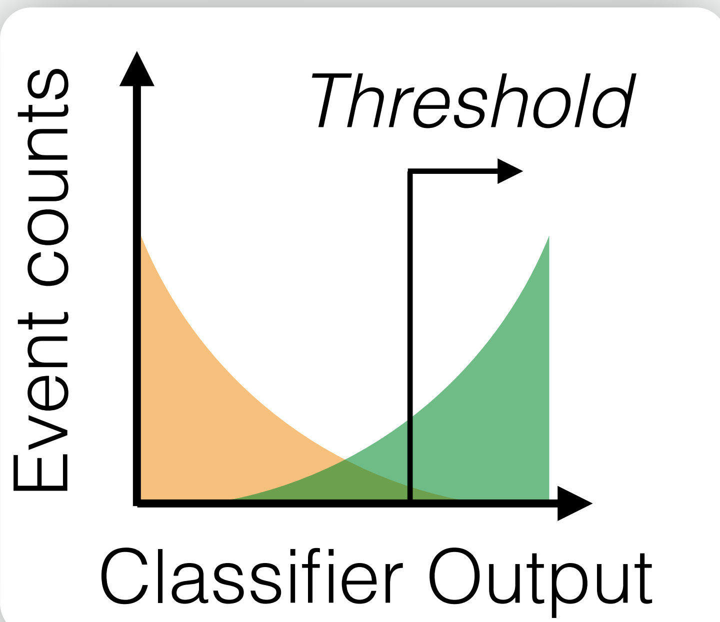
+ Machine Learning

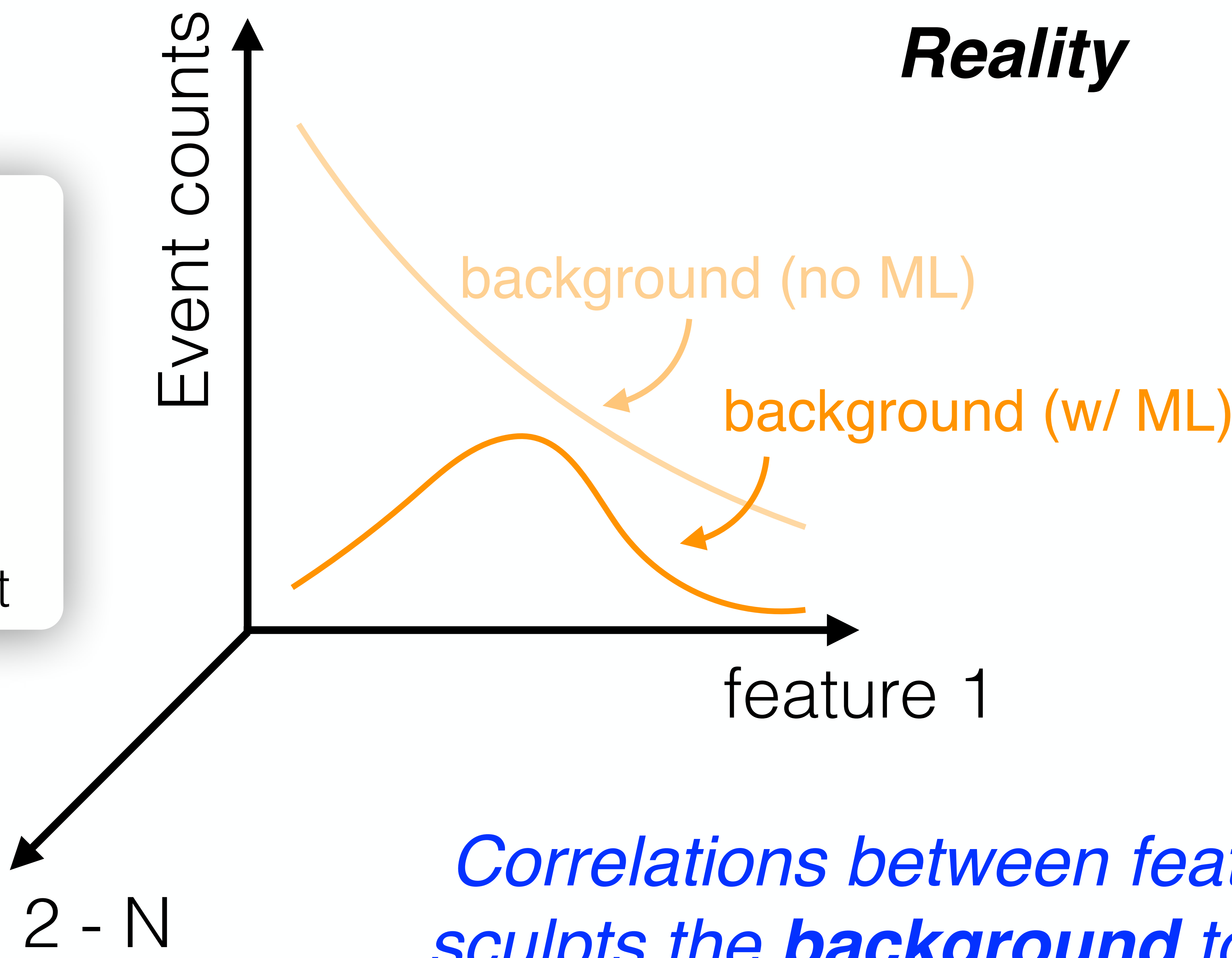


Signal to noise increases with a cut on the classifier output

A simplified HEP analysis

+ Machine Learning

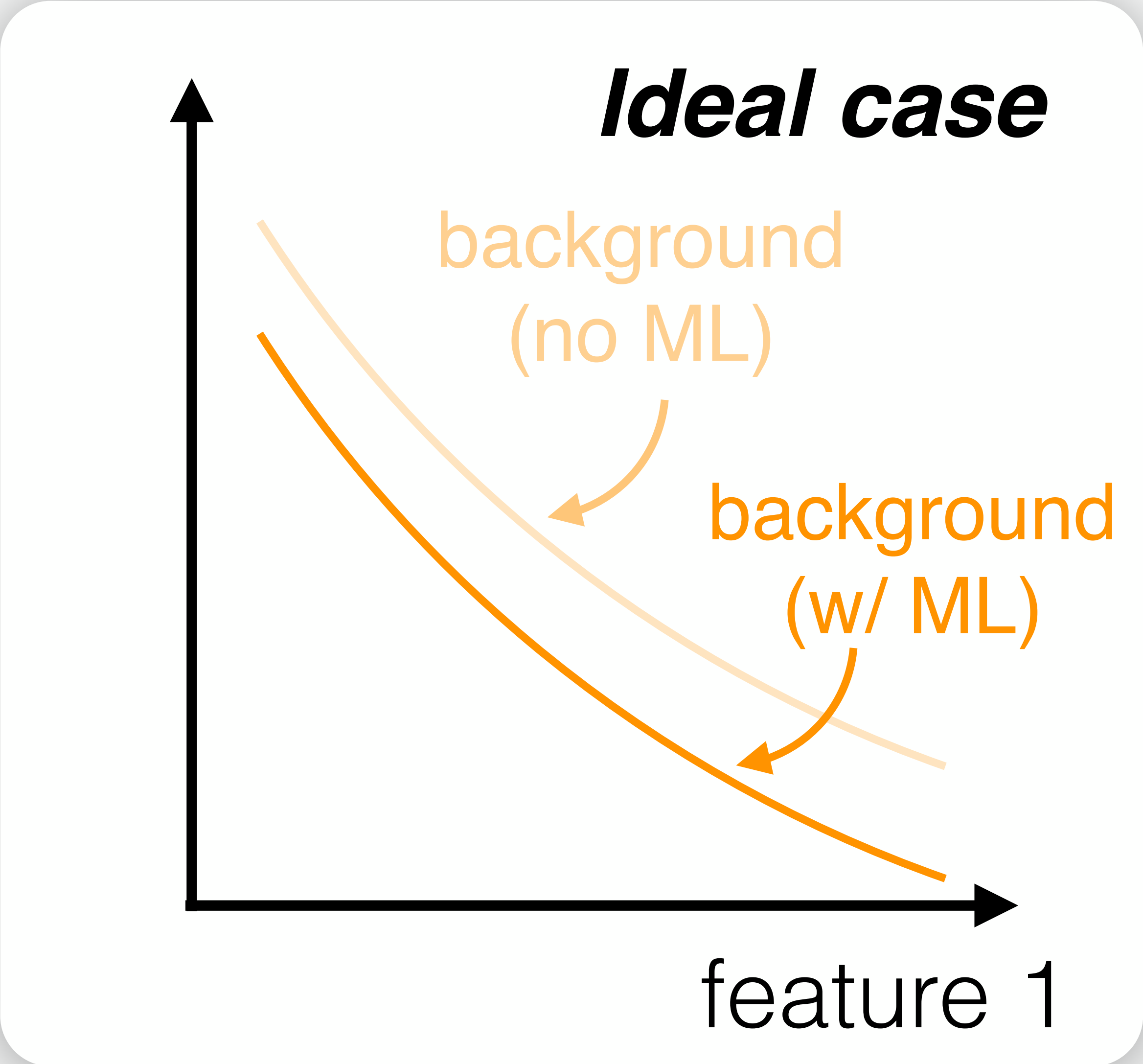


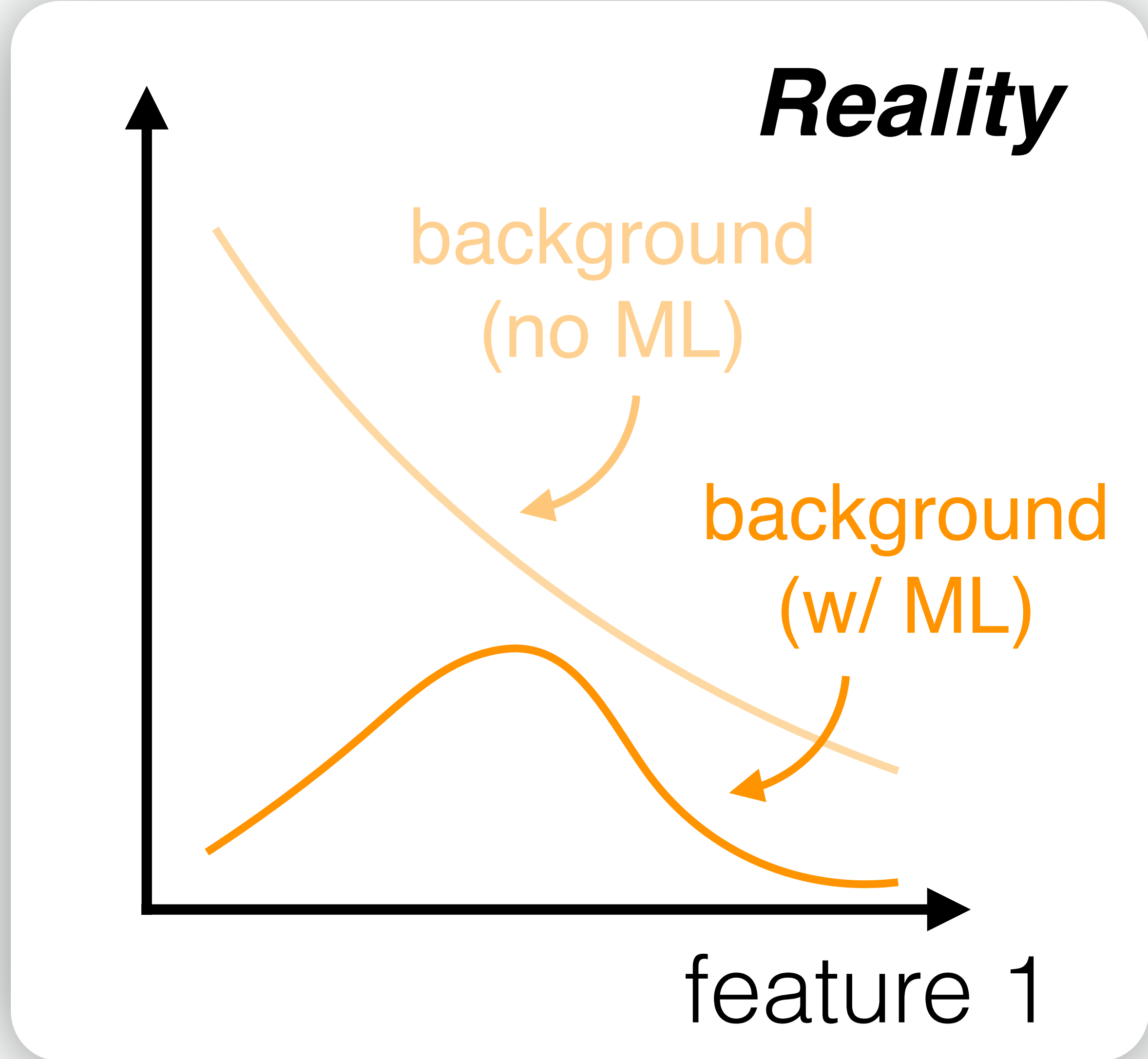


Correlations between features sculpts the ***background*** *to look like the signal → **not fittable!***

Learning robust classifiers

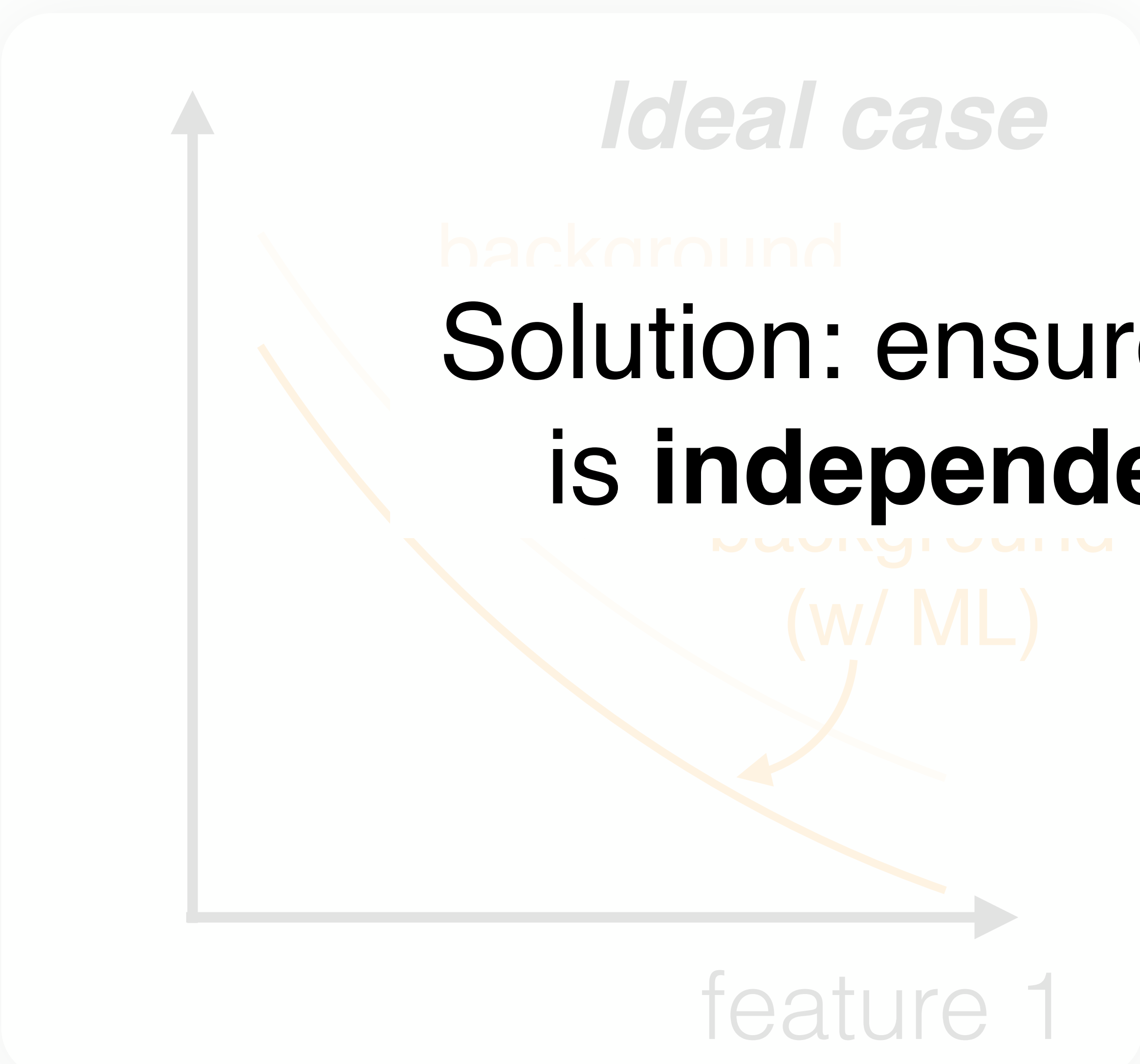
How can we learn a classifier that does not sculpt a bump in the background?

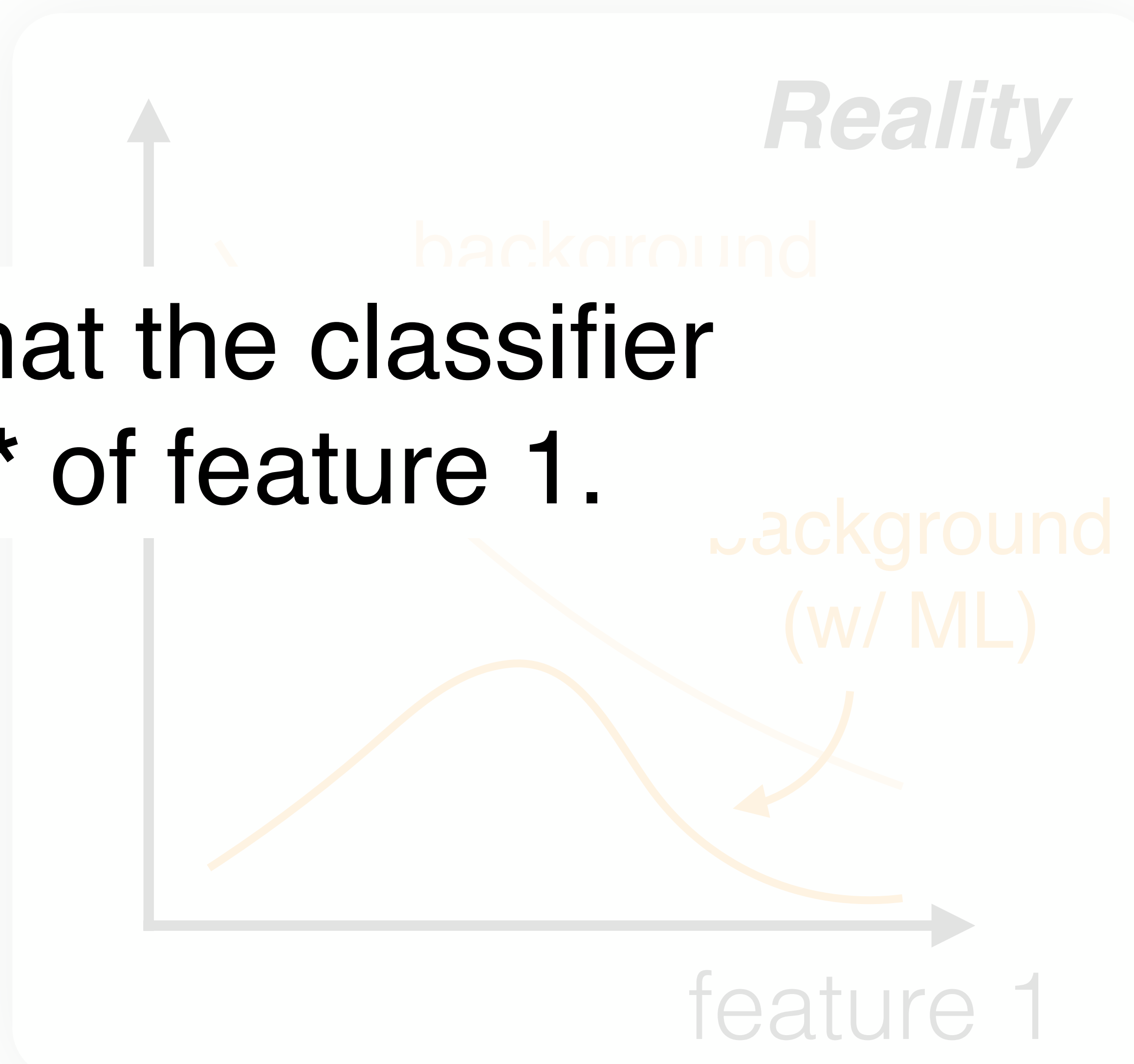




Learning robust classifiers

How can we learn a classifier that does not sculpt a bump in the background?



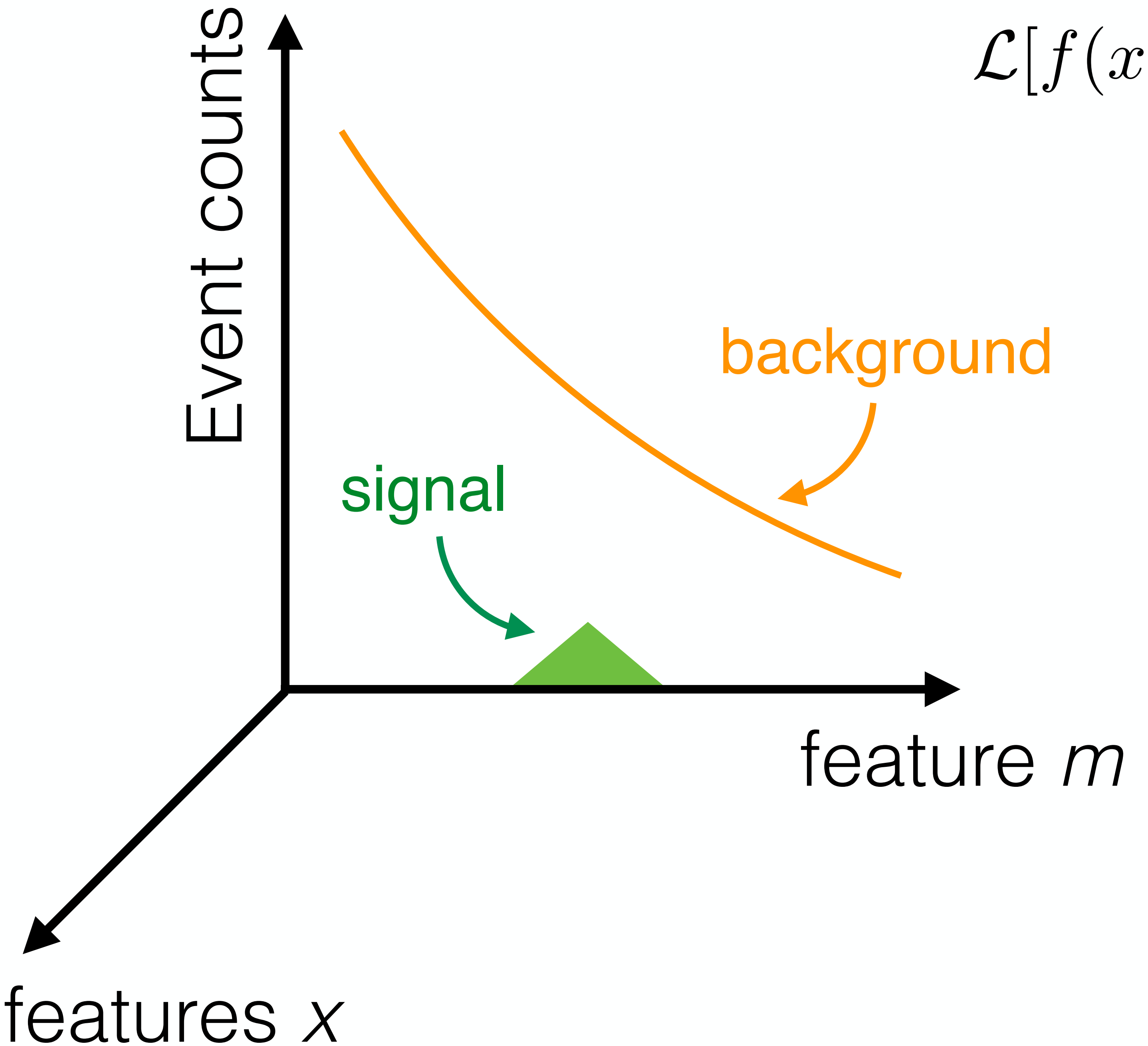


Solution: ensure that the classifier is **independent*** of feature 1.

**This is actually sufficient but unnecessary. There are many dependencies (e.g. linear) that would not sculpt bumps.*

Enforcing Independence

Train e.g. a neural network

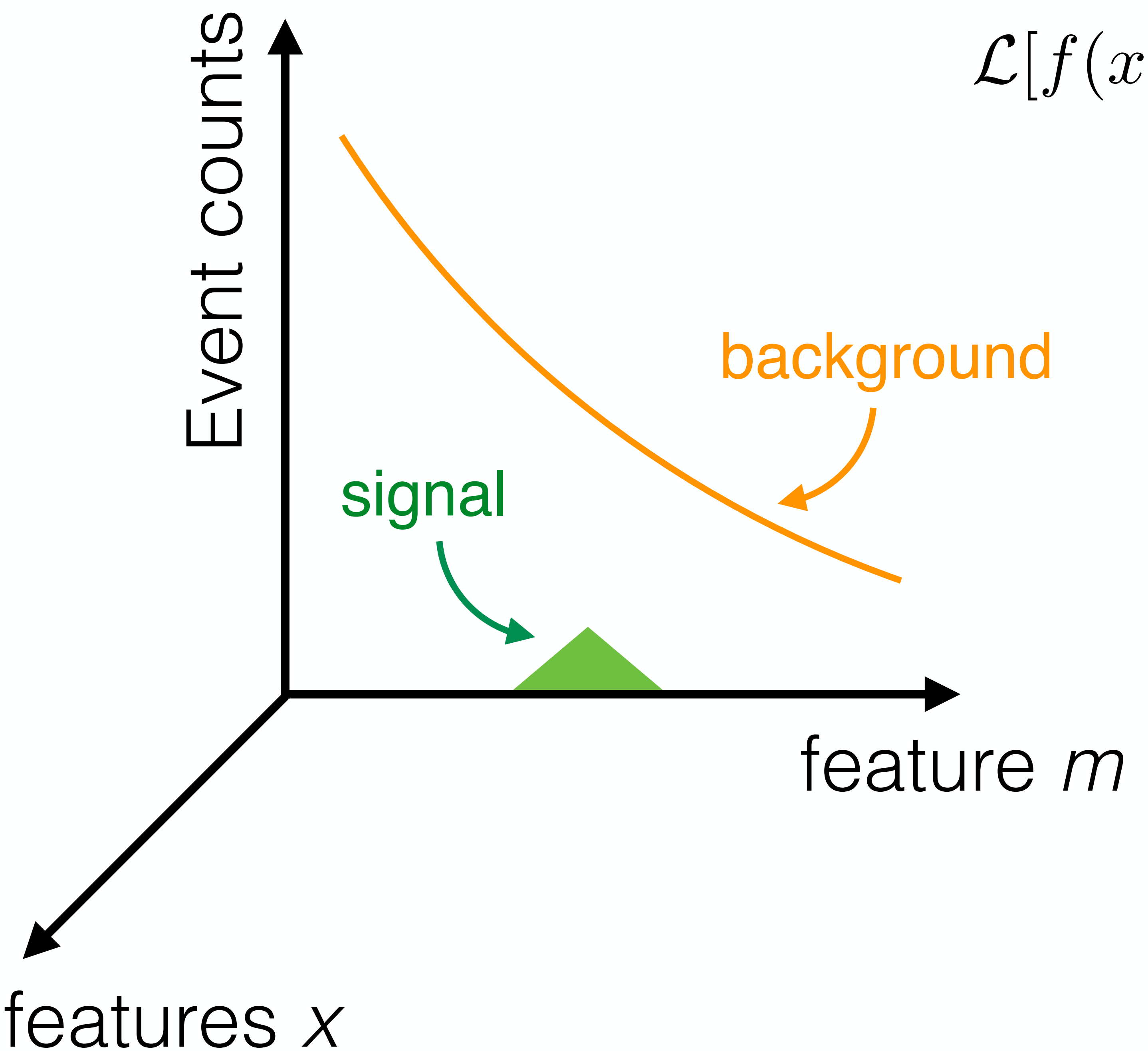


$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i), 1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i), 0)$$

$L_{classifier}$ is the usual classifier loss, e.g. cross entropy or mean squared error.

Enforcing Independence

Train e.g. a neural network with a **custom loss functional**



$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i), 1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i), 0) + \lambda \sum_{i\in b} L_{\text{decor}}(f(x_i), m_i)$$

$L_{classifier}$ is the usual classifier loss, e.g. cross entropy or mean squared error.

L_{decor} is large when *f(x)* and *m* are "correlated"

Enforcing Independence

Train e.g. a neural network with a **custom loss functional**

$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i), 1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i), 0) \\ + \lambda \sum_{i\in b} L_{\text{decor}}(f(x_i), m_i)$$

Recent proposals:

Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn *m* from *f(x)*.

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between *m* and *f(x)*.

Mode Decorrelation: L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m*.

Enforcing Independence

Train e.g. a neural network with a **custom loss functional**

$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i), 1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i), 0) + \lambda \sum_{i\in b} L_{\text{decor}}(f(x_i), m_i)$$

Recent proposals:

Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn *m* from *f(x)*.

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between *m* and *f(x)*.

Mode Decorrelation: L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m*.

Enforcing Independence

Train e.g. a neural network with a **custom loss functional**

$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i), 1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i), 0)$$
$$+\lambda \sum_{i\in b} L_{\text{decor}}(f(x_i), m_i)$$

Recent proposals:

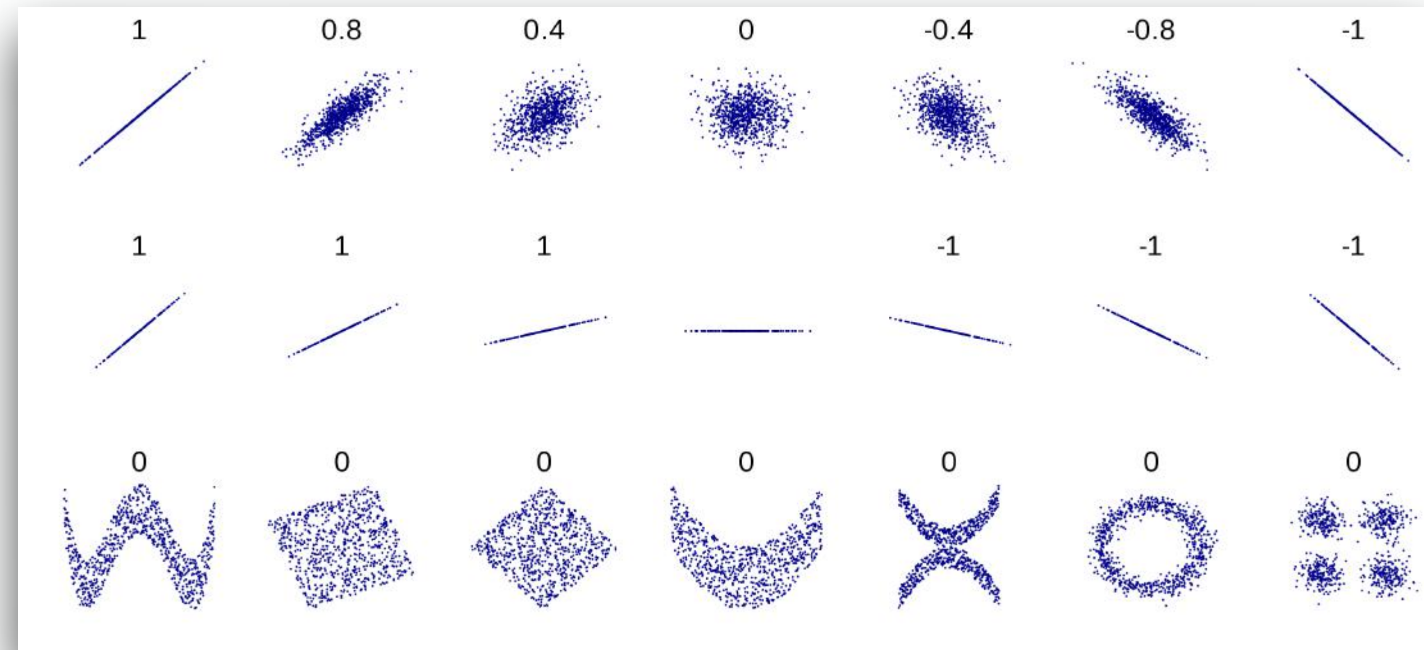
Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn *m* from *f(x)*.

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between *m* and *f(x)*.

Mode Decorrelation: L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m*.

Enforcing Independence

Pearson Correlation



Distance Correlation

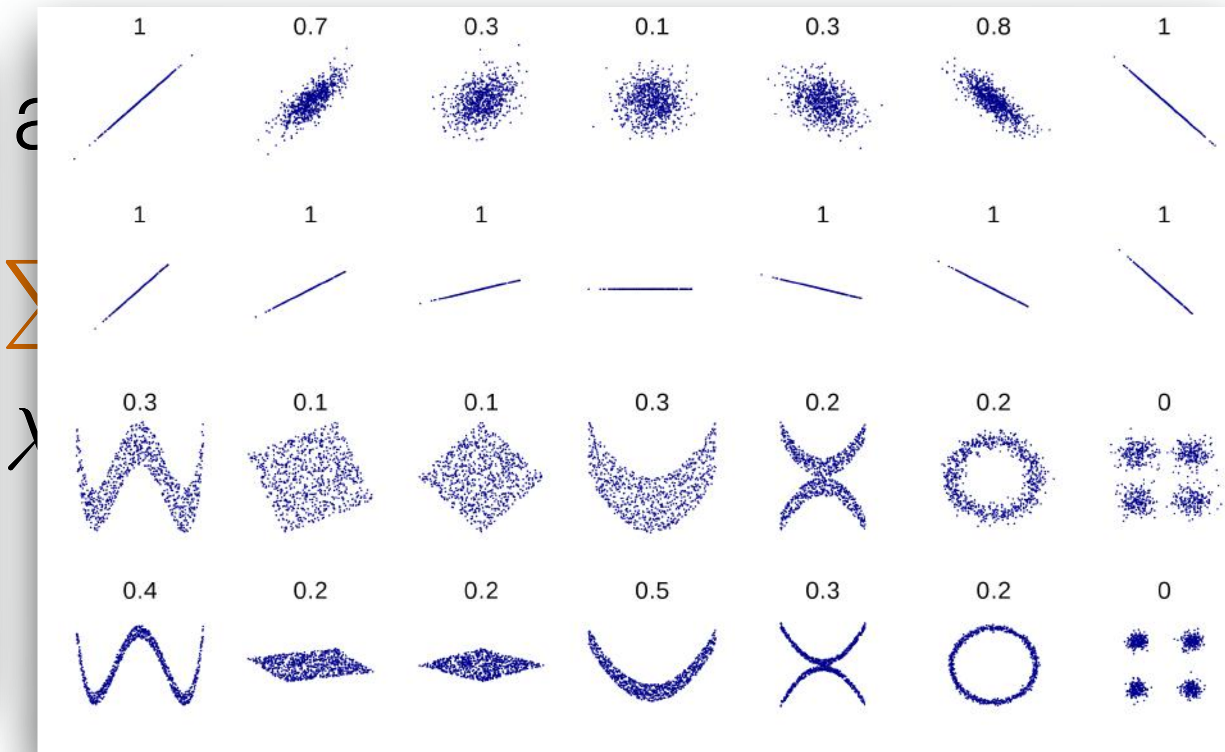


Image credit: Denis Boigelot

Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn *m* from *f(x)*.

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between *m* and *f(x)*.

Mode Decorrelation: L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m*.

Enforcing Independence

Train e.g. a neural network with a **custom loss functional**

$$\mathcal{L}[f(x)] = \sum_{i\in s} L_{\text{classifier}}(f(x_i), 1) + \sum_{i\in b} L_{\text{classifier}}(f(x_i), 0) + \lambda \sum_{i\in b} L_{\text{decor}}(f(x_i), m_i)$$

Recent proposals:

Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn *m* from *f(x)*.

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between *m* and *f(x)*.

Mode Decorrelation: L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m*.

Adversaries

Adversaries: L_{decor} is the loss of **a 2nd NN** (adversary) that tries to learn m from $f(x)$.

Pros: Very flexible and m can be multidimensional

Cons: Hard to train (minimax problem) & many parameters

G. Louppe, M. Kagan, K. Cranmer, 1611.01046; C. Shimmin et al., 1703.03507

Distance Correlation

Distance Correlation: L_{decor} is **distance correlation** (generalizes Pearson correlation) between m and $f(x)$.

Pros: Convex (easier to train) and no free parameters

Cons: Memory intensive to compute distance correlation

G. Kasieczka and D. Shih, 2001.05310; G. Kasieczka, **BPN**, M. Schwartz, D. Shih, 2007.14400

Mode Decorrelation

Mode Decorrelation (MoDE): L_{decor} is small when the **CDF** of *f(x)* is the same across different values of *m.*

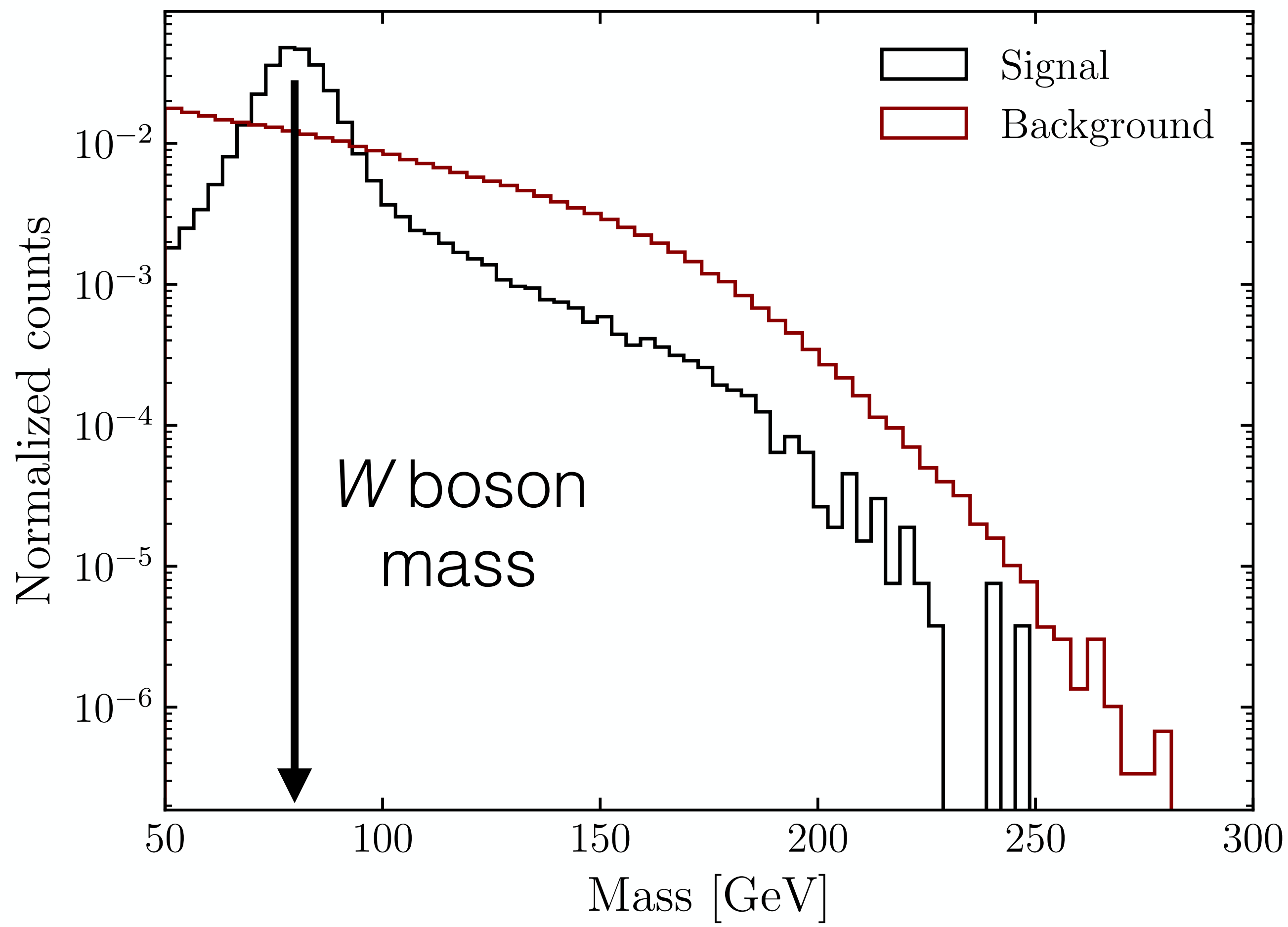
Pros: Readily generalizes beyond independence (can require linear, quadratic (+monotonic), …

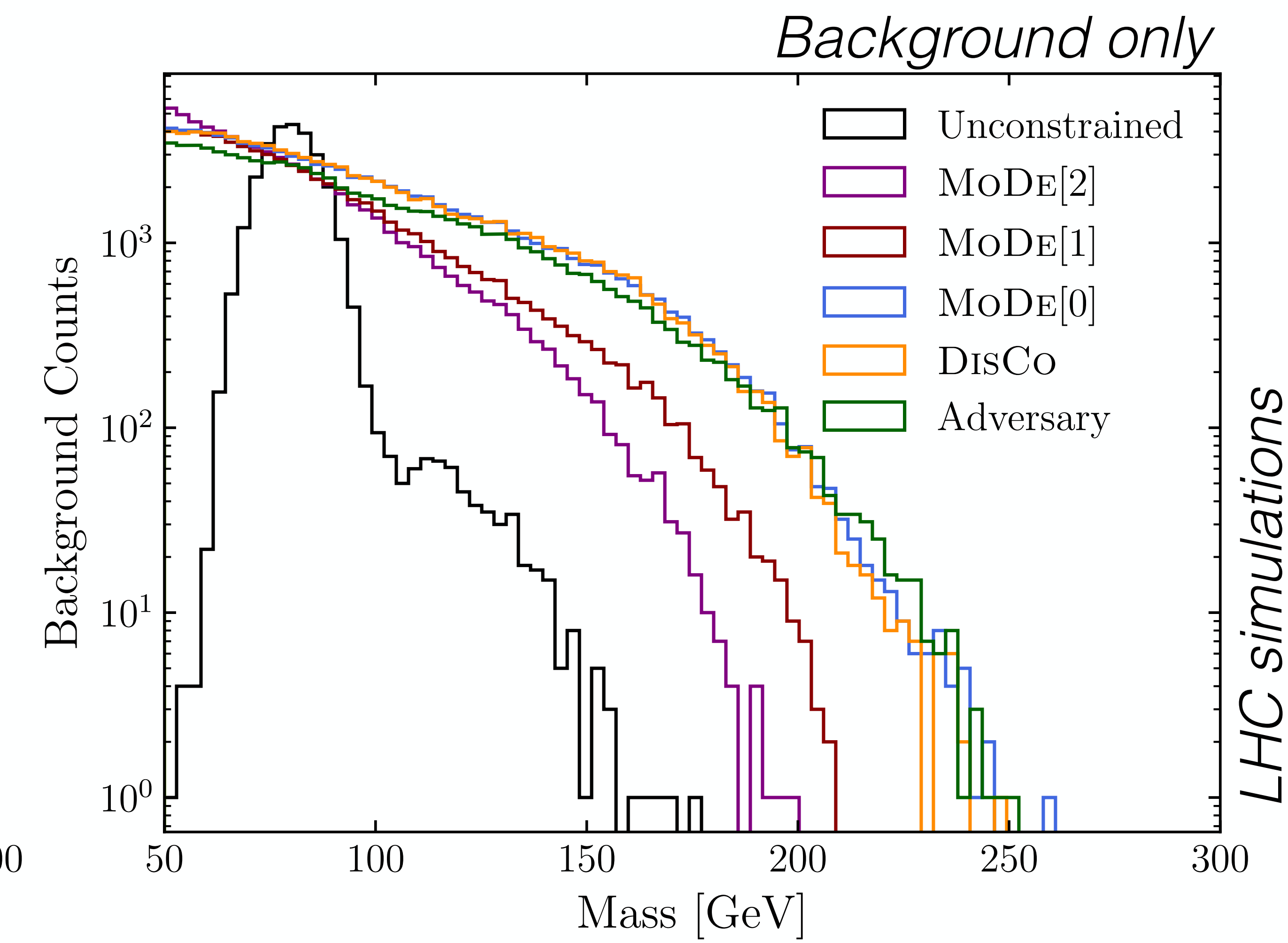
No free parameters and small memory footprint

Cons: In its simplest form, need discrete bins in *m* (does not seem to be fundamental)

O. Kitouni, **BPN**, C. Weisser, M. Williams, 2010.09745

Real world example: the search for Lorentz-boosted *W* bosons at the Large Hadron Collider





MoDe[0] enforces independence, [1] is linear, [2] is monotonic quadratic, …

O. Kitouni, **BPN**, C. Weisser, M. Williams, 2010.09745

Connections

What does **decorrelation** have to do with other areas of science, society and industry?

This is solving the same problem as **fairness**.

e.g. you train a classifier to screen CVs of job candidates and you want it to not indirectly learn age, race, ethnicity, gender, …

Can tools from HEP be applied more broadly? For example, when we have continuous categories and/or monotonicity (and not independence) requirements?

A related topic: prior dependence

A common task in HEP is **calibration**: learning a map from measured quantities to true (pre-detector) values.

A related topic: prior dependence

A common task in HEP is **calibration**: learning a map from measured quantities to true (pre-detector) values.

For example, a particle of a given energy hits our detector and registers measurements in a number of sensors

A related topic: prior dependence

A common task in HEP is **calibration**: learning a map from measured quantities to true (pre-detector) values.

For example, a particle of a given energy hits our detector and registers measurements in a number of sensors

A good calibration is **universal**: it should be unbiased even if the true values in the training dataset follow a different spectrum than the test set.

A related topic: prior dependence

A common task in HEP is **calibration**: learning a map from measured quantities to true (pre-detector) values.

For example, a particle of a given energy hits our detector and registers measurements in a number of sensors

A good calibration is **universal**: it should be unbiased even if the true values in the training dataset follow a different spectrum than the test set.

e.g. the particle energy is uniform during training, but exponential for certain running conditions.

Why standard methods are bad

A common task in HEP is **calibration**: learning a map from measured quantities to true (pre-detector) values.

For example, a particle of a given energy hits our detector and registers measurements in a number of sensors

Your first instinct here might have been to train a classifier to estimate the true value given measured values using simulated data.

Why standard methods are bad

A common task in HEP is **calibration**: learning a map from measured quantities to true (pre-detector) values.

For example, a particle of a given energy hits our detector and registers measurements in a number of sensors

Your first instinct here might have been to train a classifier to estimate the true value given measured values using simulated data.

Claim: this is prior dependent !

What goes wrong?

Suppose you have some features x and you want to predict y.

detector energy *true energy*

One way to do this is to find an f that minimizes the mean squared error (MSE):

$$f = \mathrm{argmin}_g \sum_i (g(x_i) - y_i)^2$$

Then, f(x) = E[y|x].

Why is this a problem?

What goes wrong?

Suppose you have some features x and you want to predict y.

detector energy *true energy*

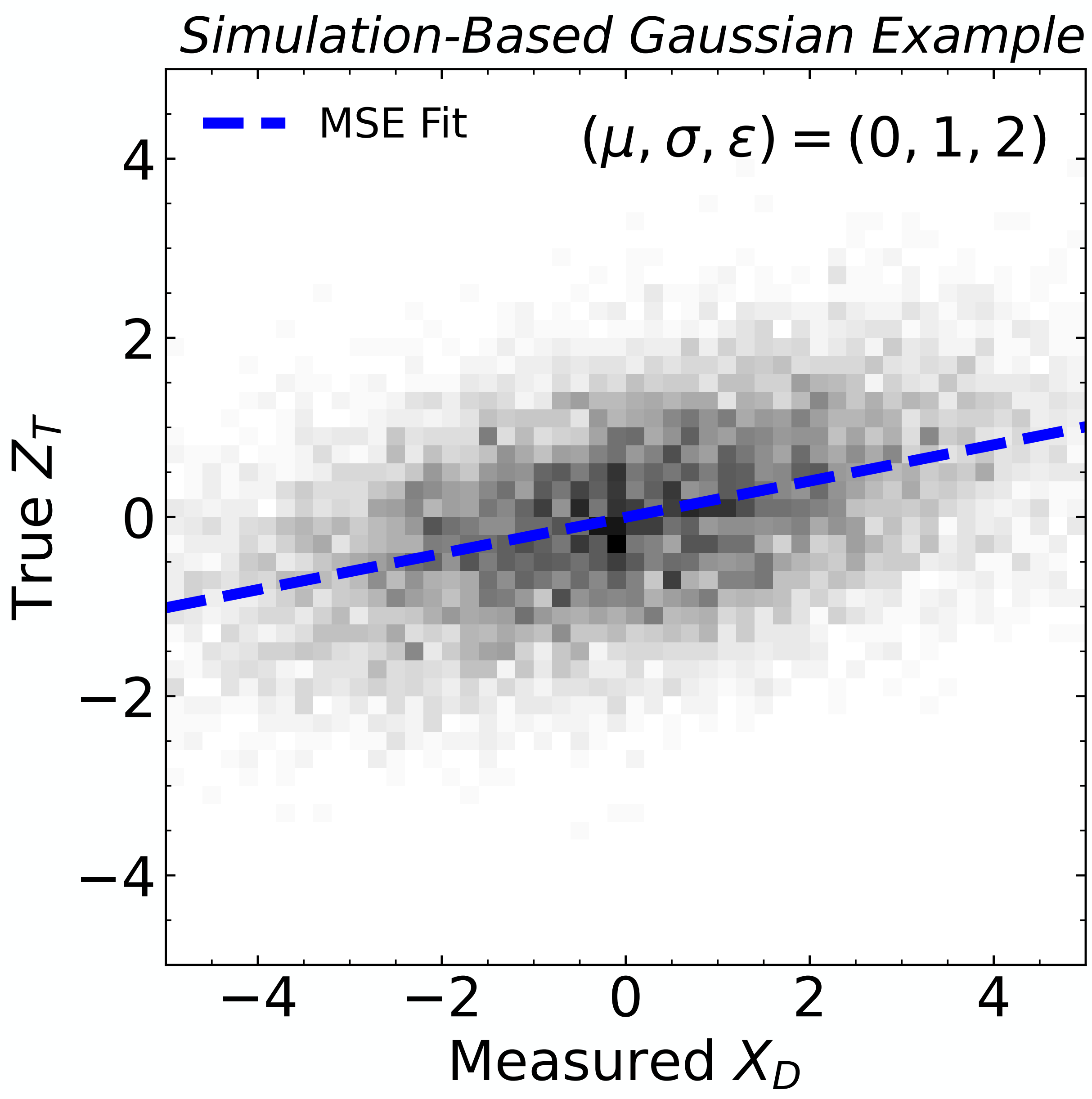
$$f(x) = E[y|x] = \int \mathrm{d}y\, y\, p(y|x)$$

$$E[f(x)|y] = \int \mathrm{d}x\, \mathrm{d}y'\, y'\, p_{\mathrm{train}}(y'|x)\, p_{\mathrm{test}}(x|y)$$

this need not be y even if $p_{train} = p_{test}$ (!)

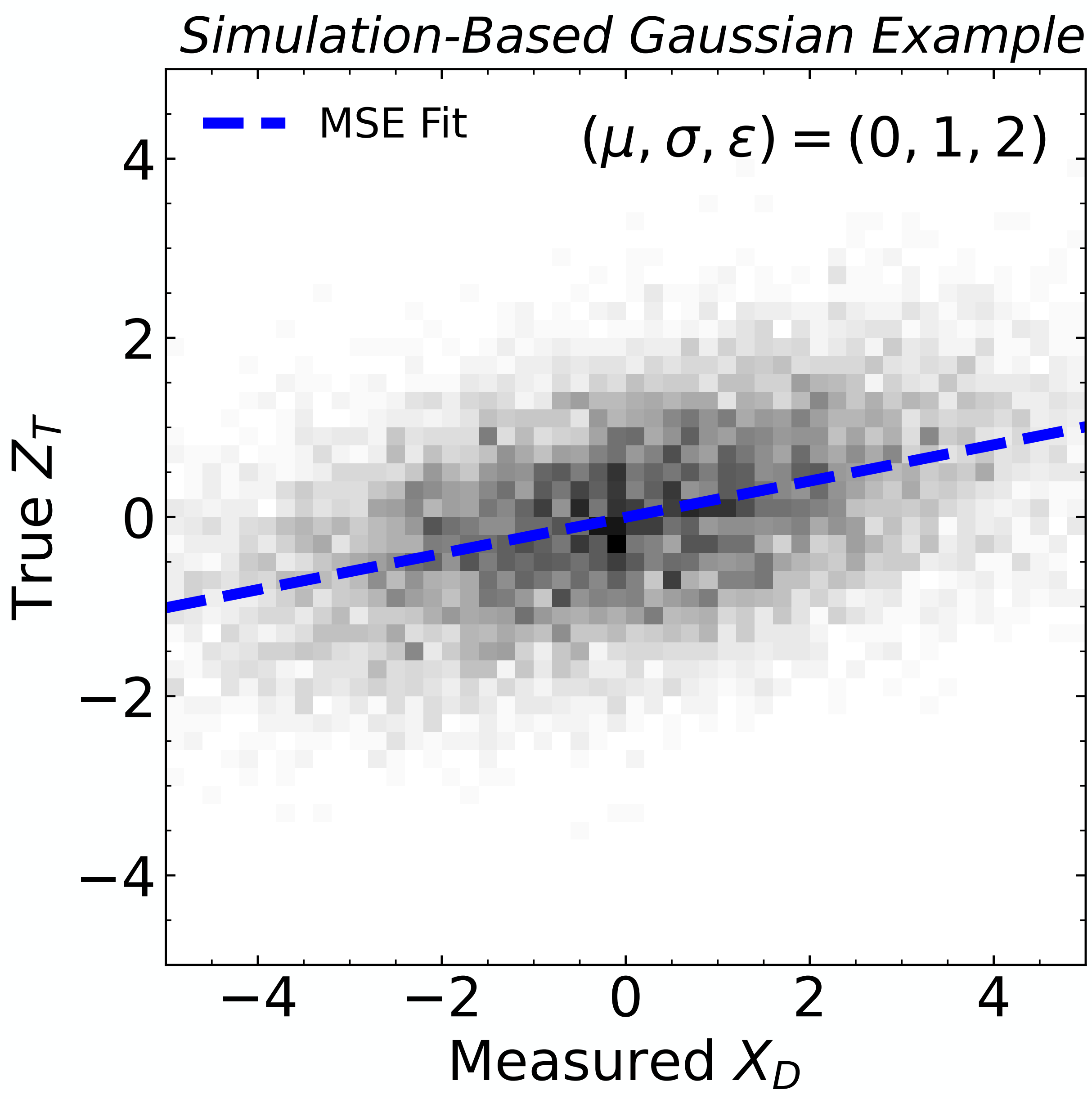
Why is this a problem?

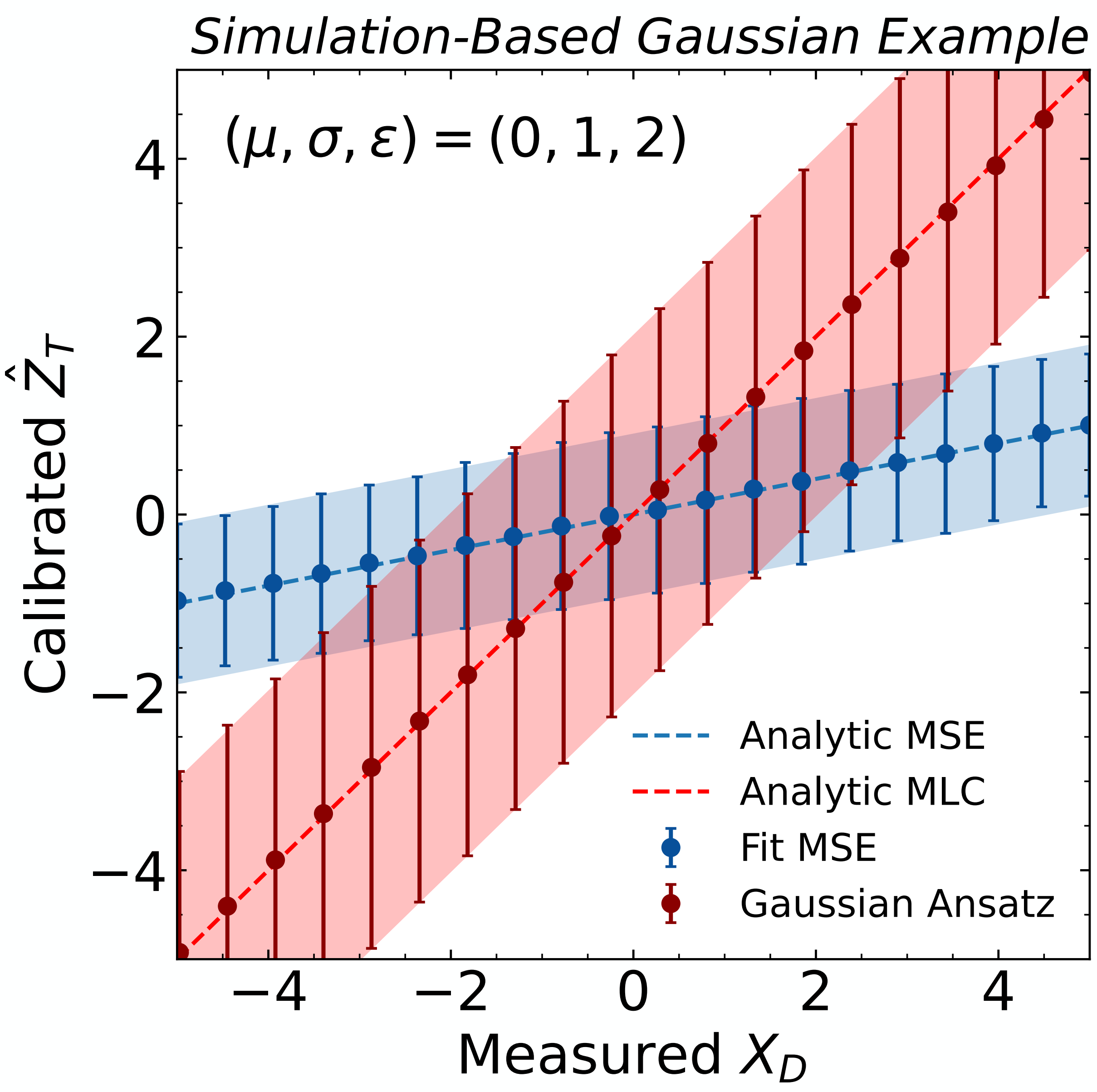
Gaussian Example



R. Gambhir, **BPN**, J. Thaler, 2205.05084

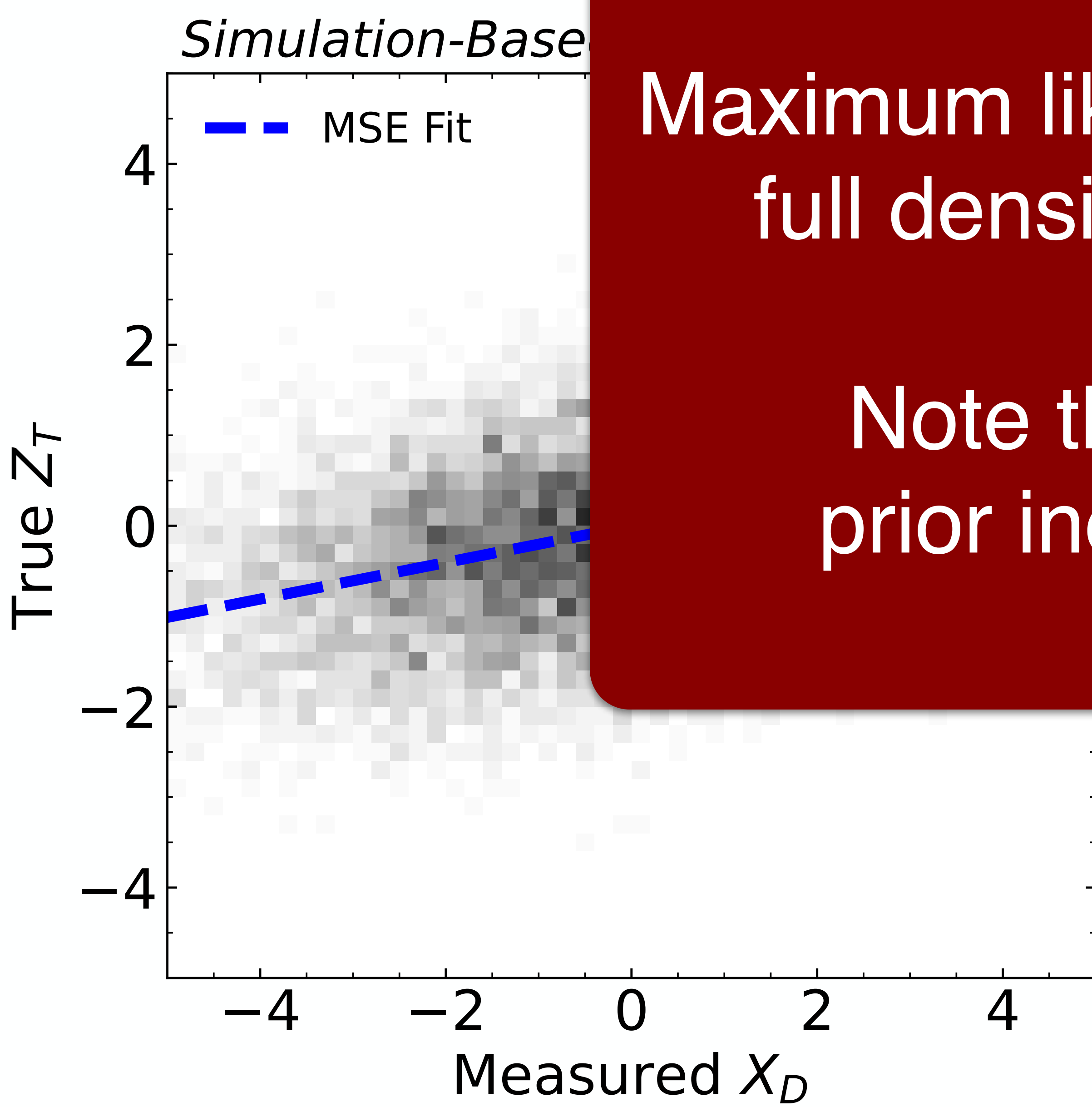
Gaussian Example





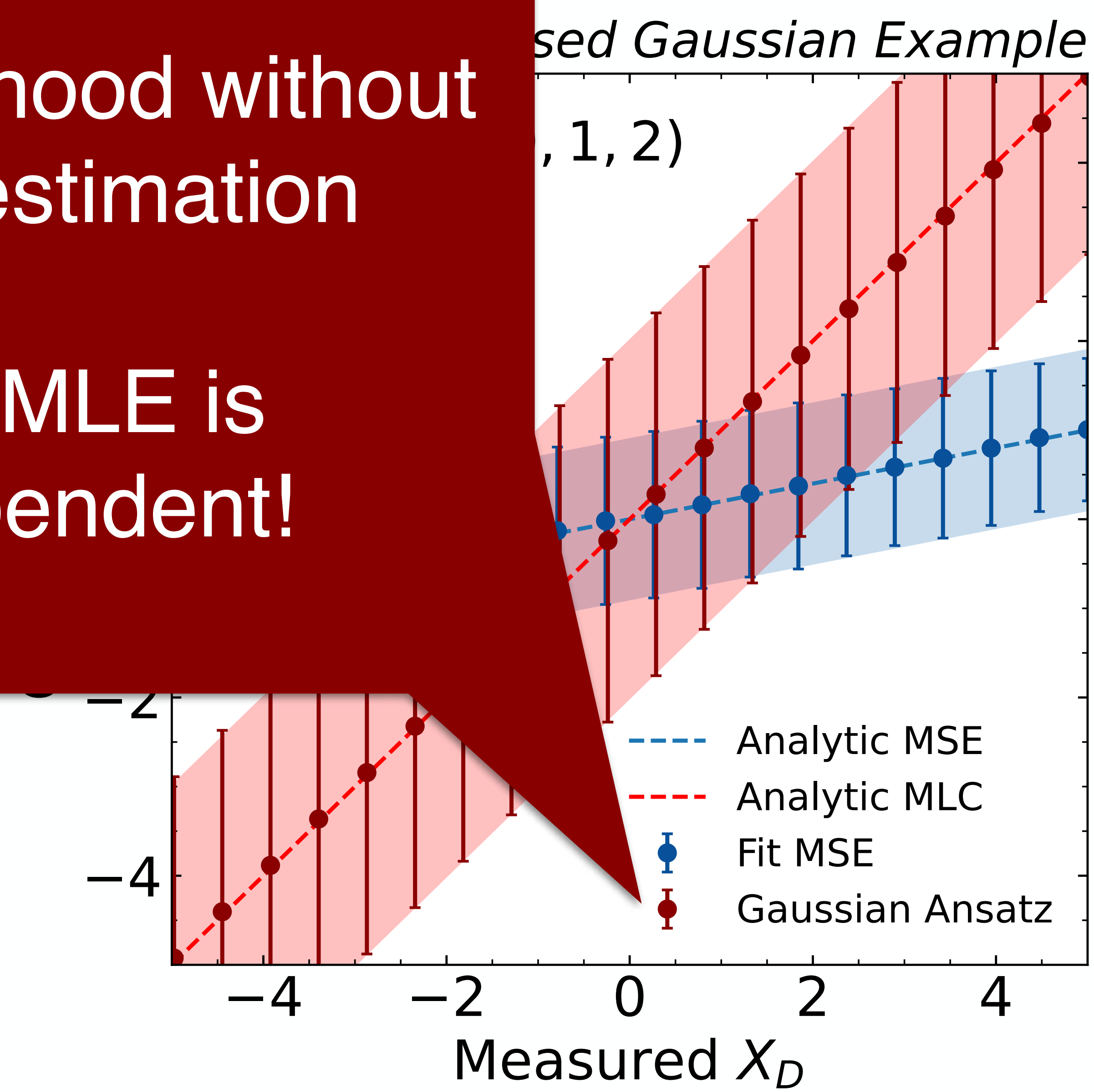
R. Gambhir, **BPN**, J. Thaler, 2205.05084

Gaussian Example



Maximum likelihood without full density estimation

Note that MLE is prior independent!

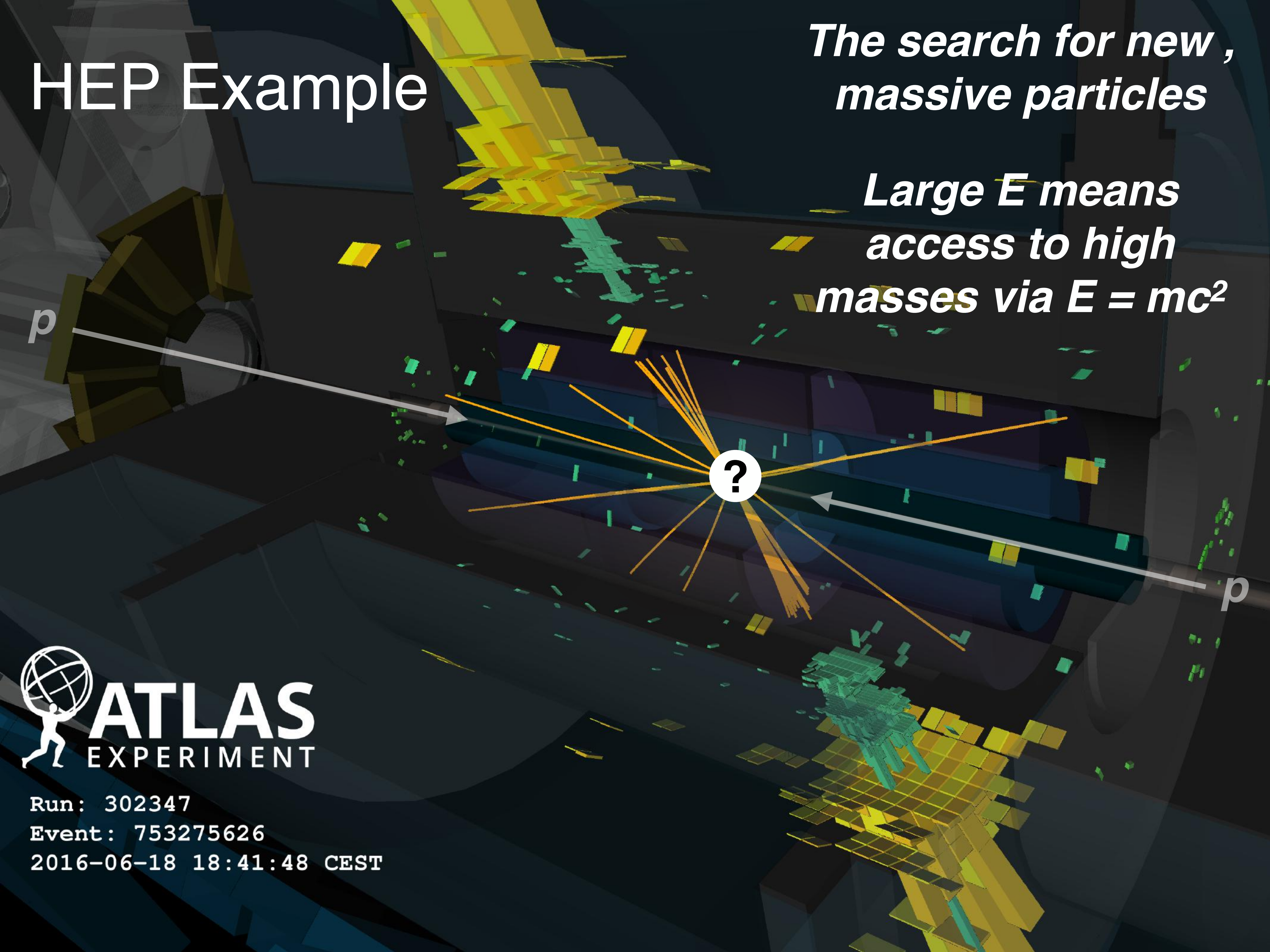


R. Gambhir, **BPN**, J. Thaler, 2205.05084

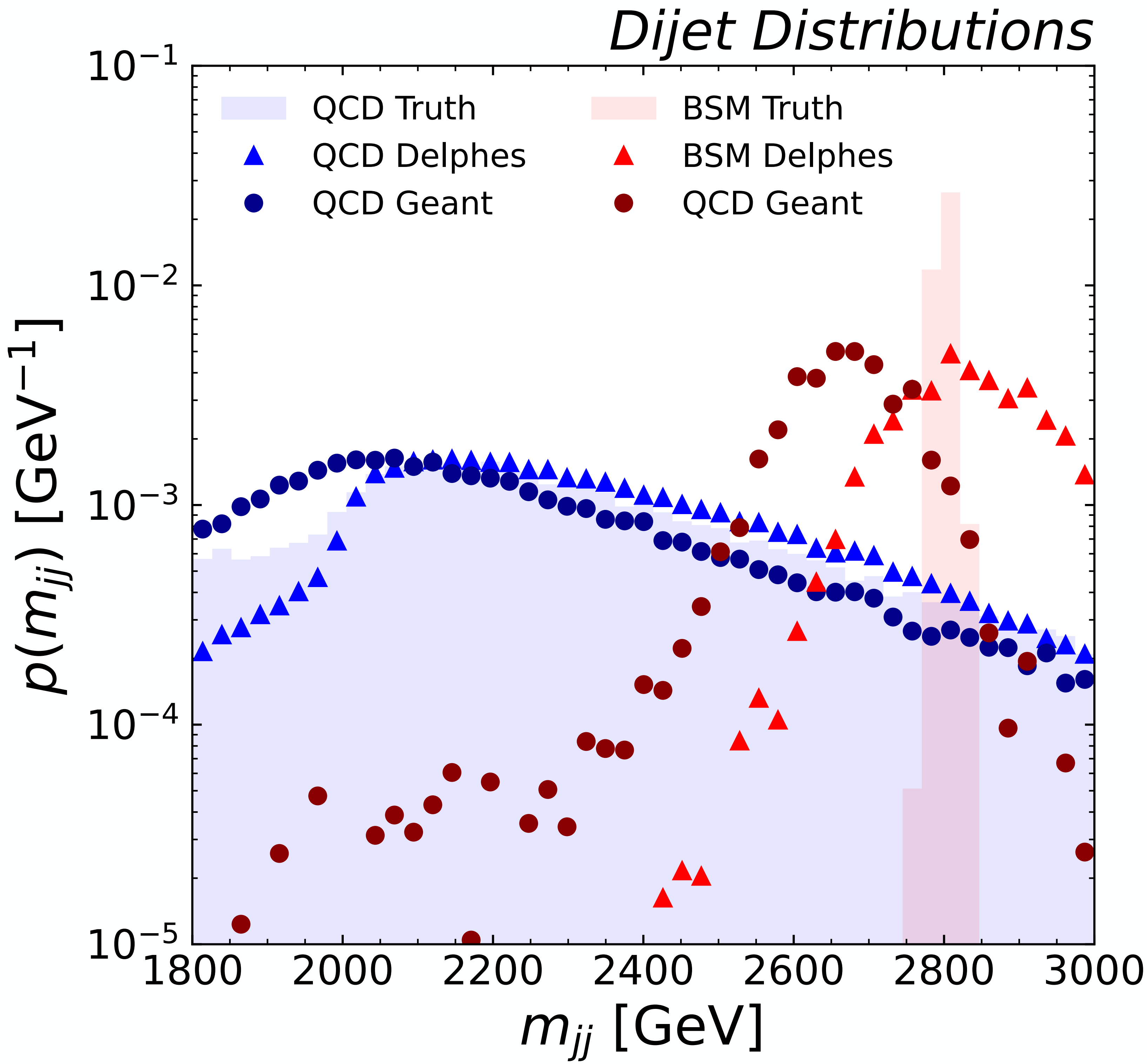
HEP Example

The search for new , massive particles

Large E means access to high masses via $E = mc^2$



HEP Example



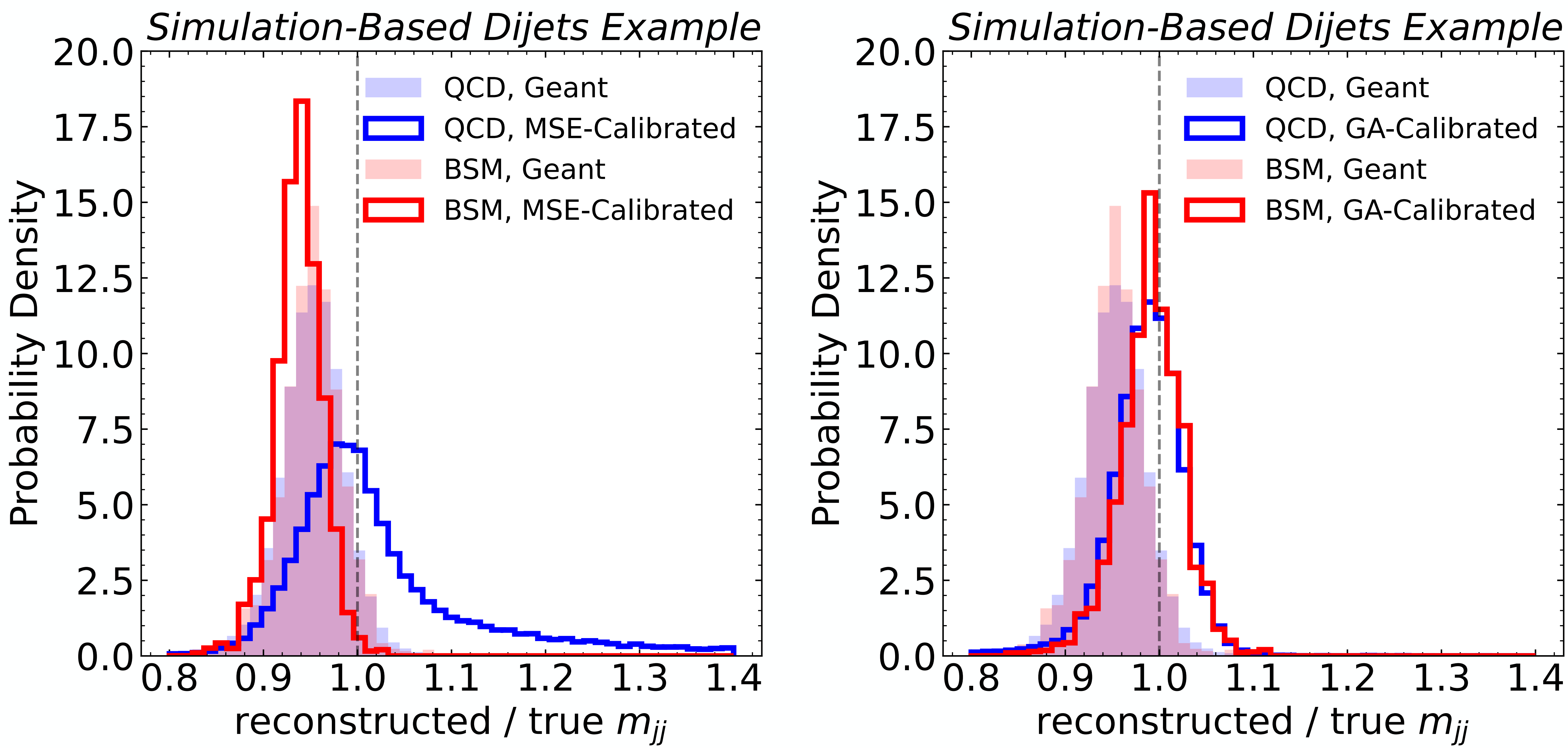
QCD = quarks and gluons

BSM = new physics

Looking for new massive particles that produce jets

R. Gambhir, **BPN**, J. Thaler, 2205.05084

HEP Example



R. Gambhir, **BPN**, J. Thaler, 2205.05084

Connections

What does **prior independence** have to do with other areas of science, society and industry?

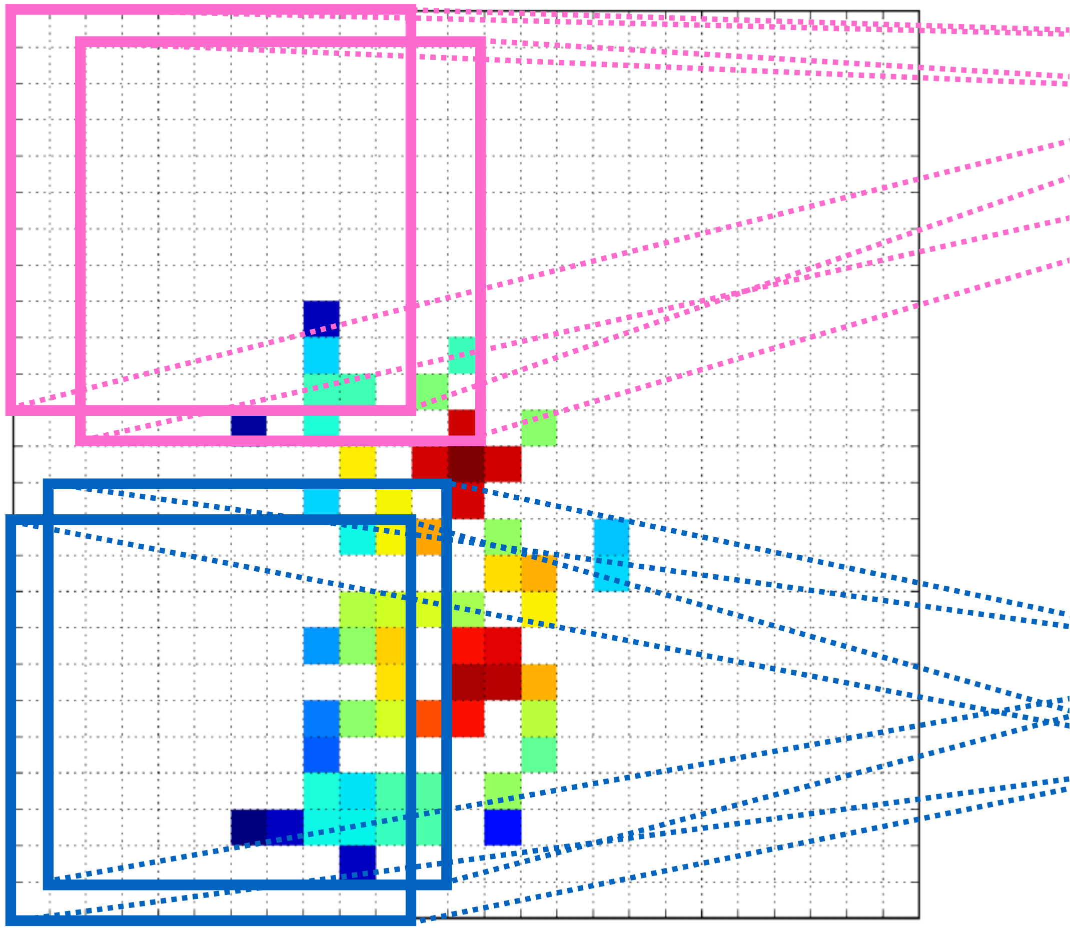
This is also related to **fairness**.

e.g. you expect the scale at the doctor's office to be correct for you on average even if the spread of weights is different than the calibration sample

Can tools from HEP be applied more broadly?

Conclusions and Outlook

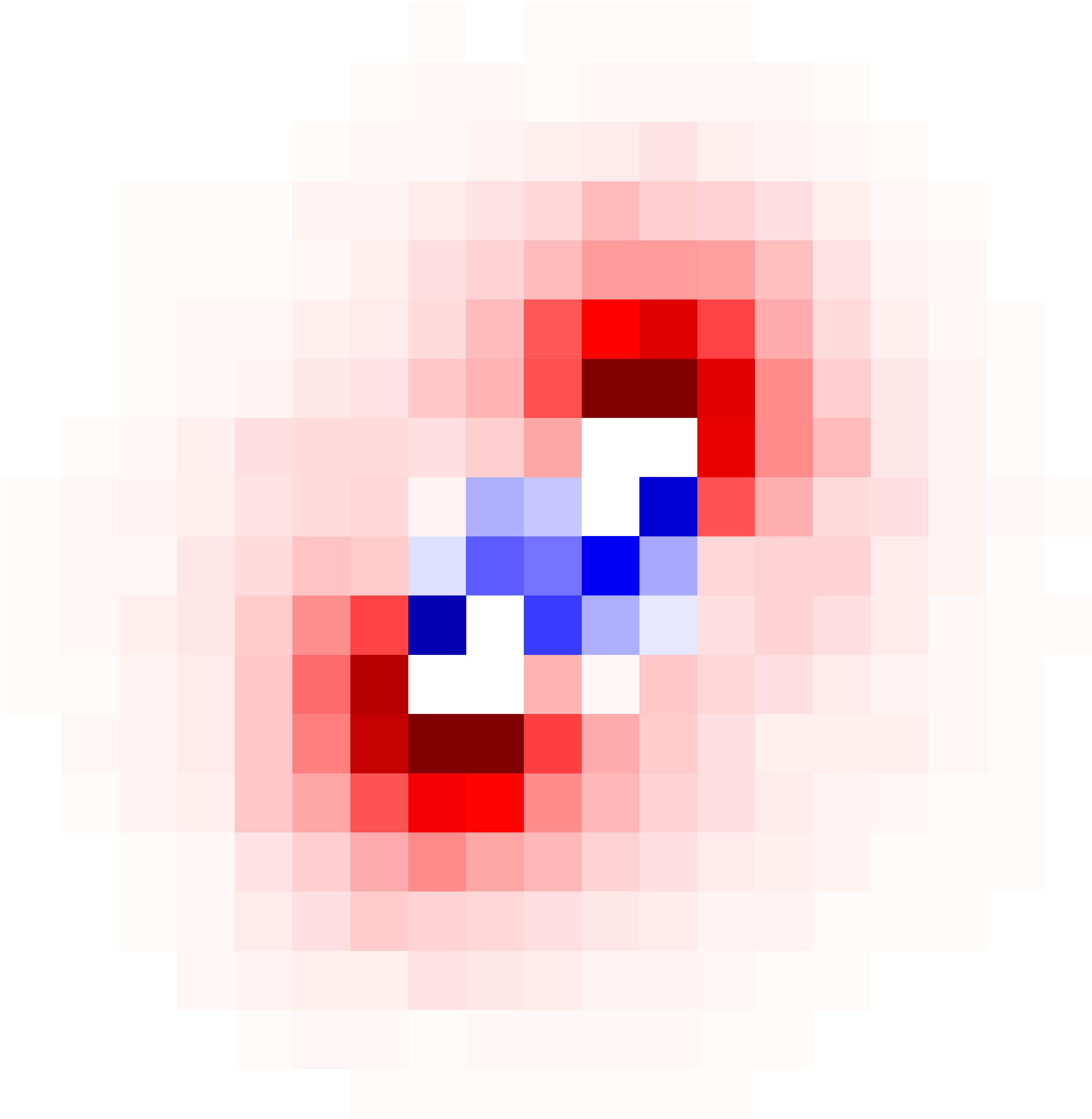
AI/ML has a great potential to **enhance**, **accelerate**, and **empower** HEP analyses



In order to make the best use of these tools, we need to ensure that they are **robust**

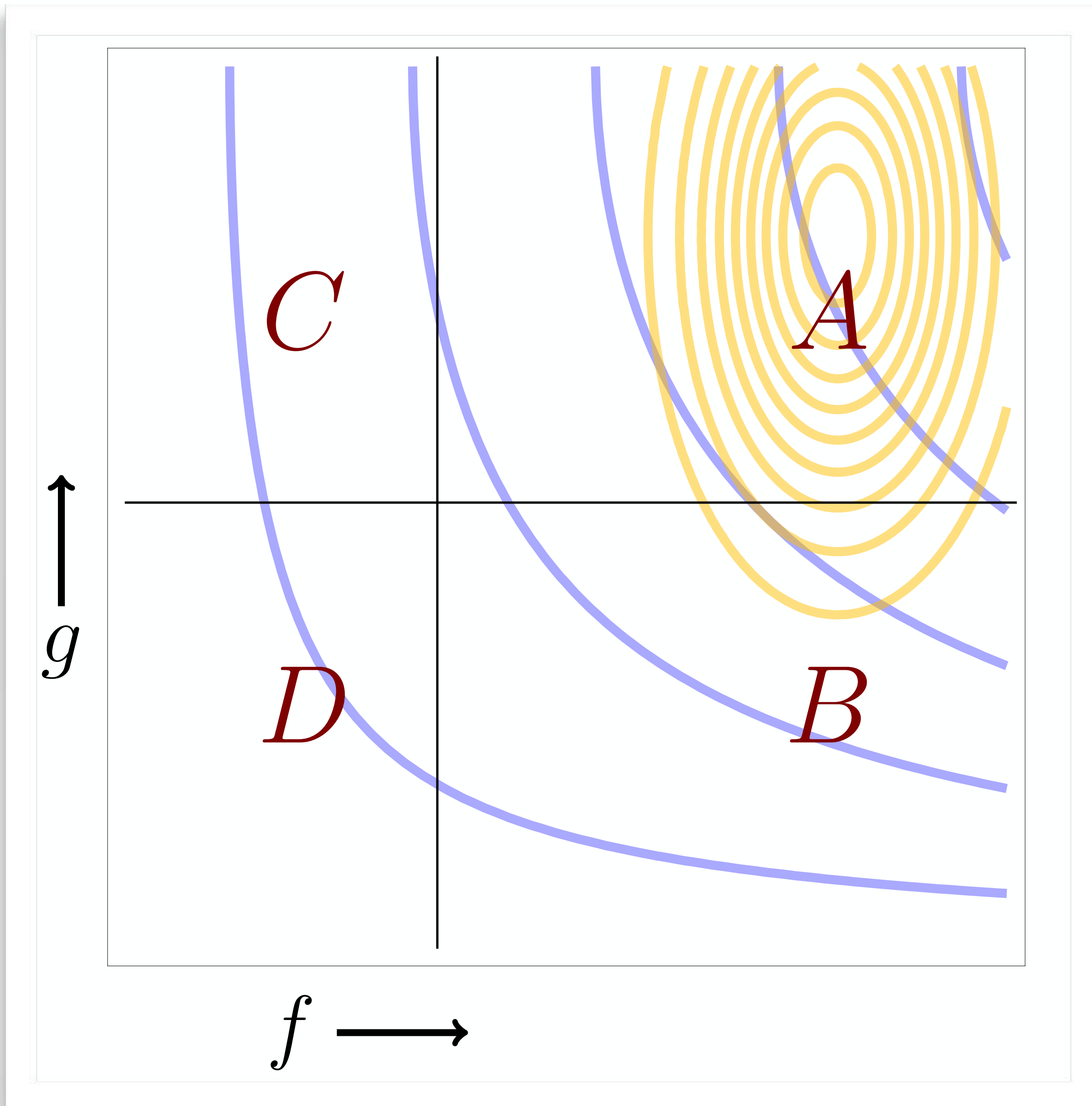
A tool is only as good as its calibration !

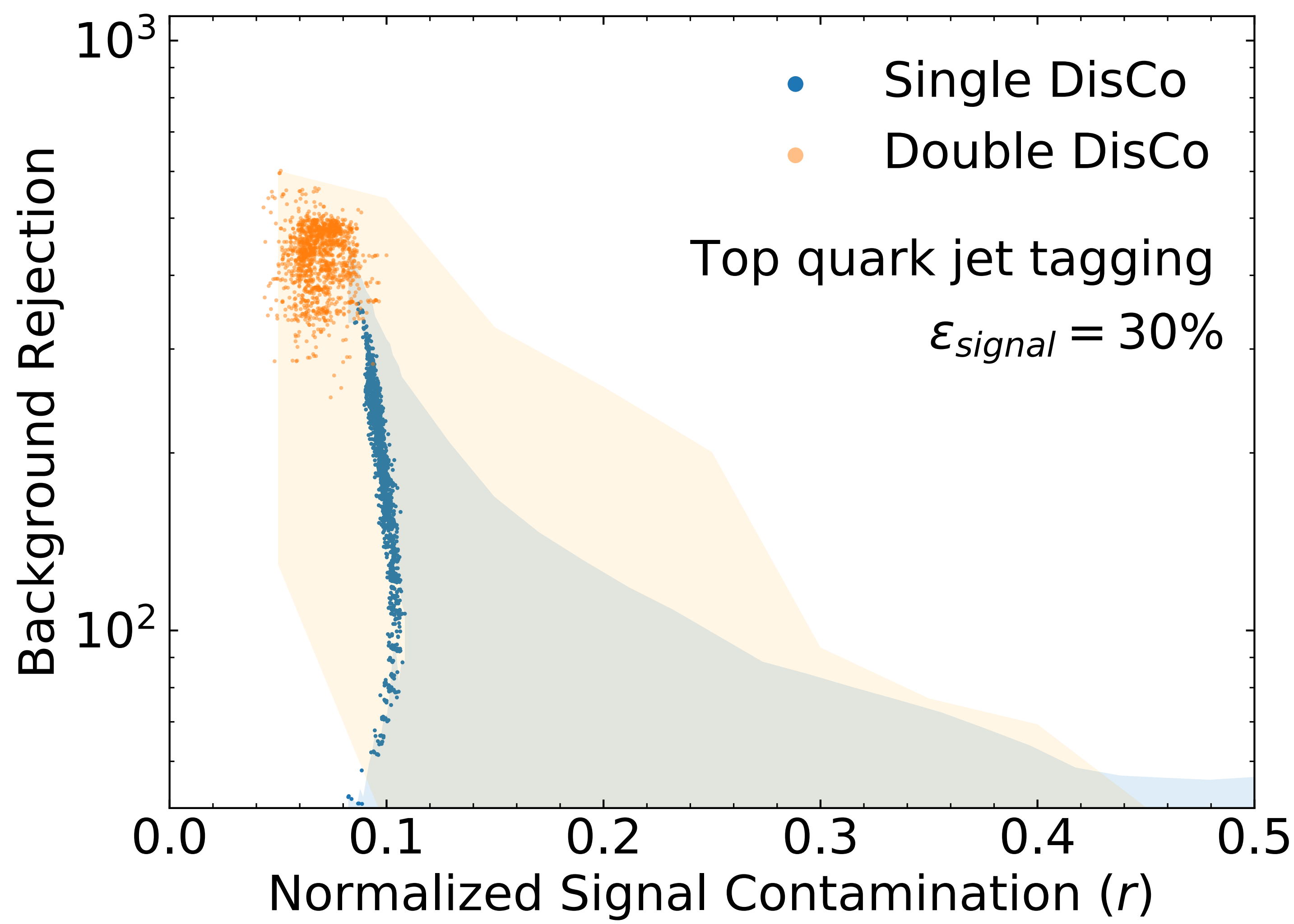
We can build robustness into classifier training and tools developed for HEP may be useful in other contexts.



Fin.

Double DisCo

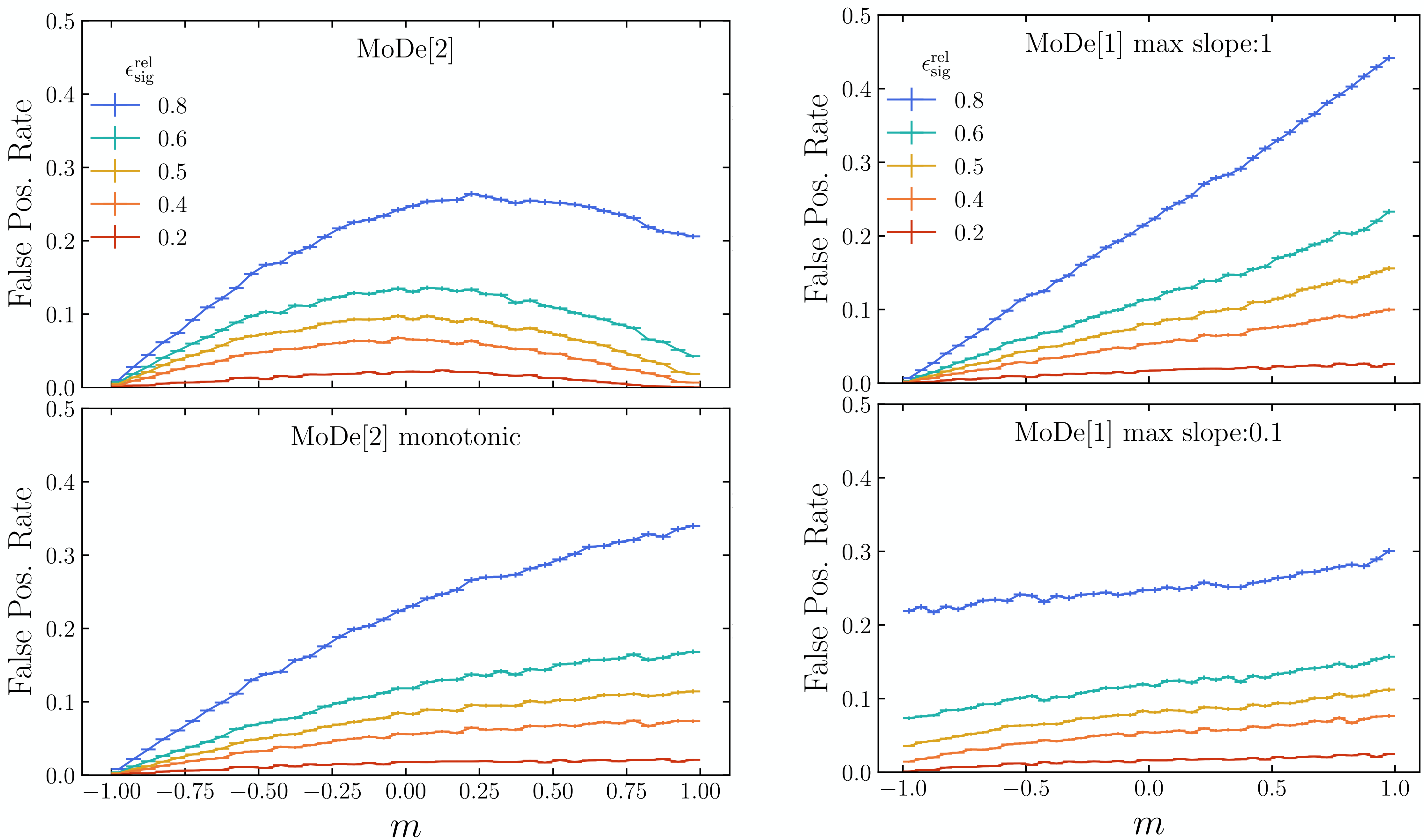




$$\mathcal{L}[f,g] = \mathcal{L}_{\text{classifier}}[f(X),y] + \mathcal{L}_{\text{classifier}}[g(X),y] + \lambda\,\text{dCorr}^2_{y=0}[f(X),g(X)]$$

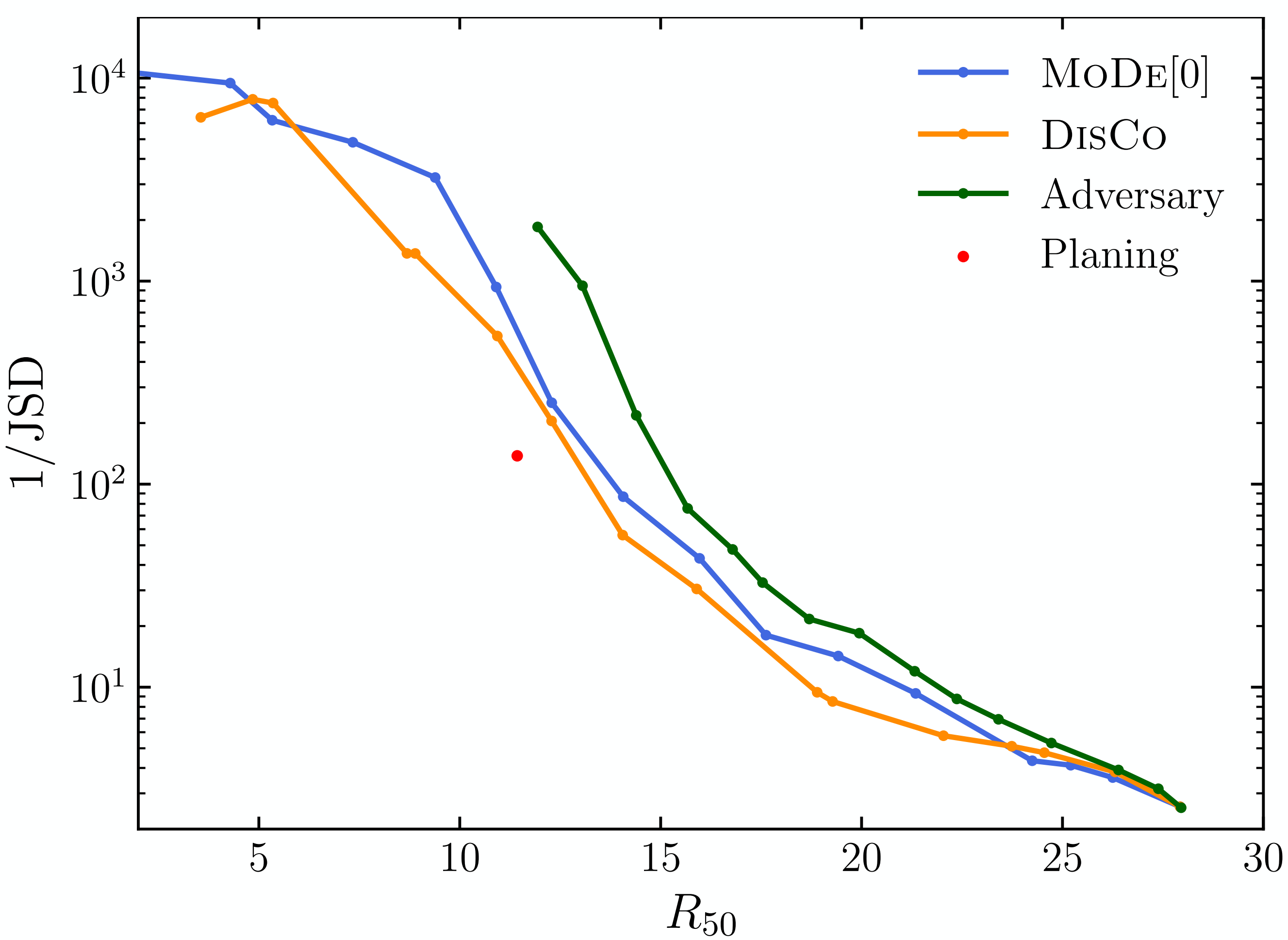
G. Kasieczka, **BPN**, M. Schwartz, D. Shih, 2007.14400

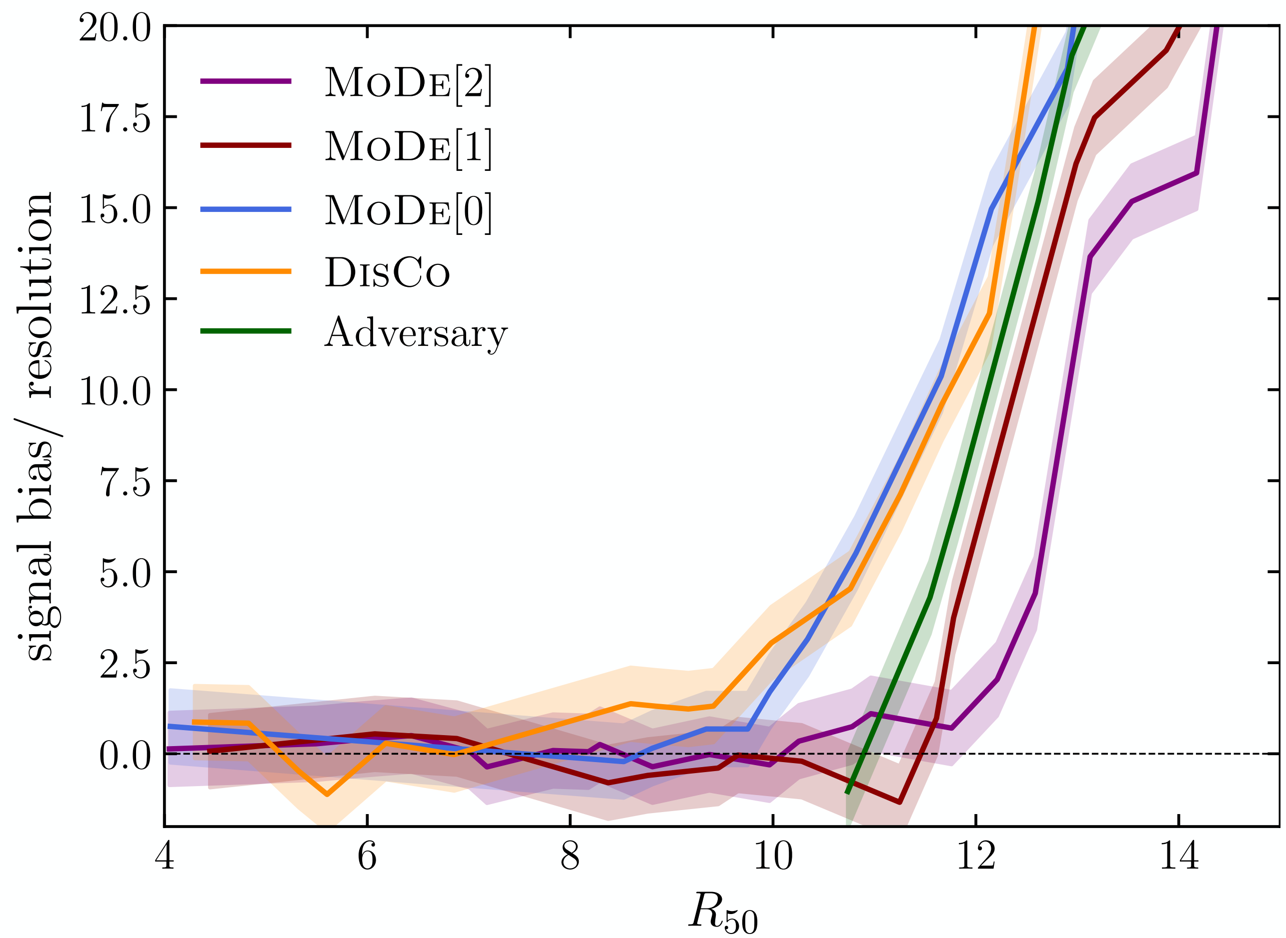
Mode Decomposition (MoDe)



O. Kitouni, **BPN**, C. Weisser, M. Williams, 2010.09745

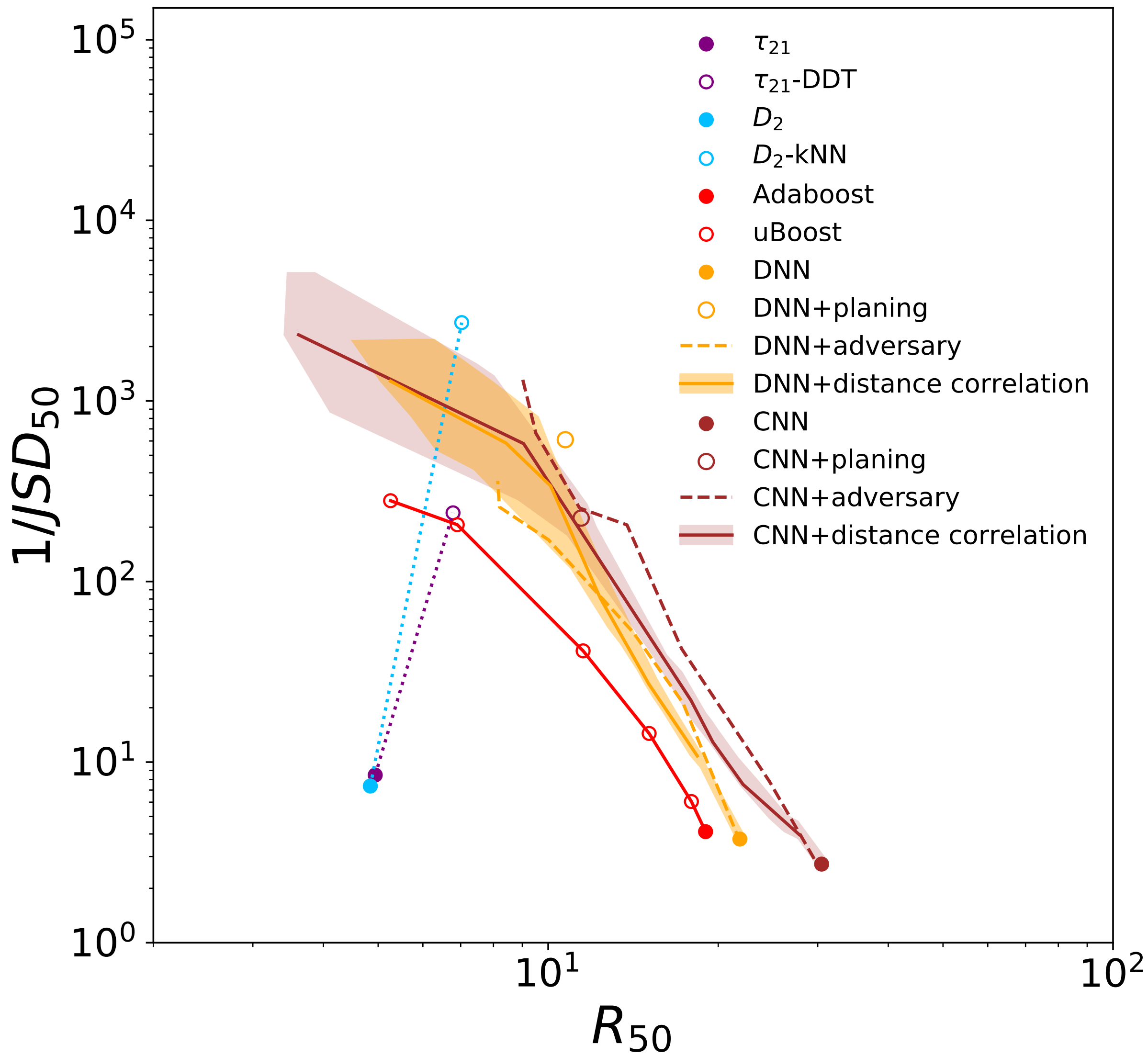
Mode Decomposition (MoDe)

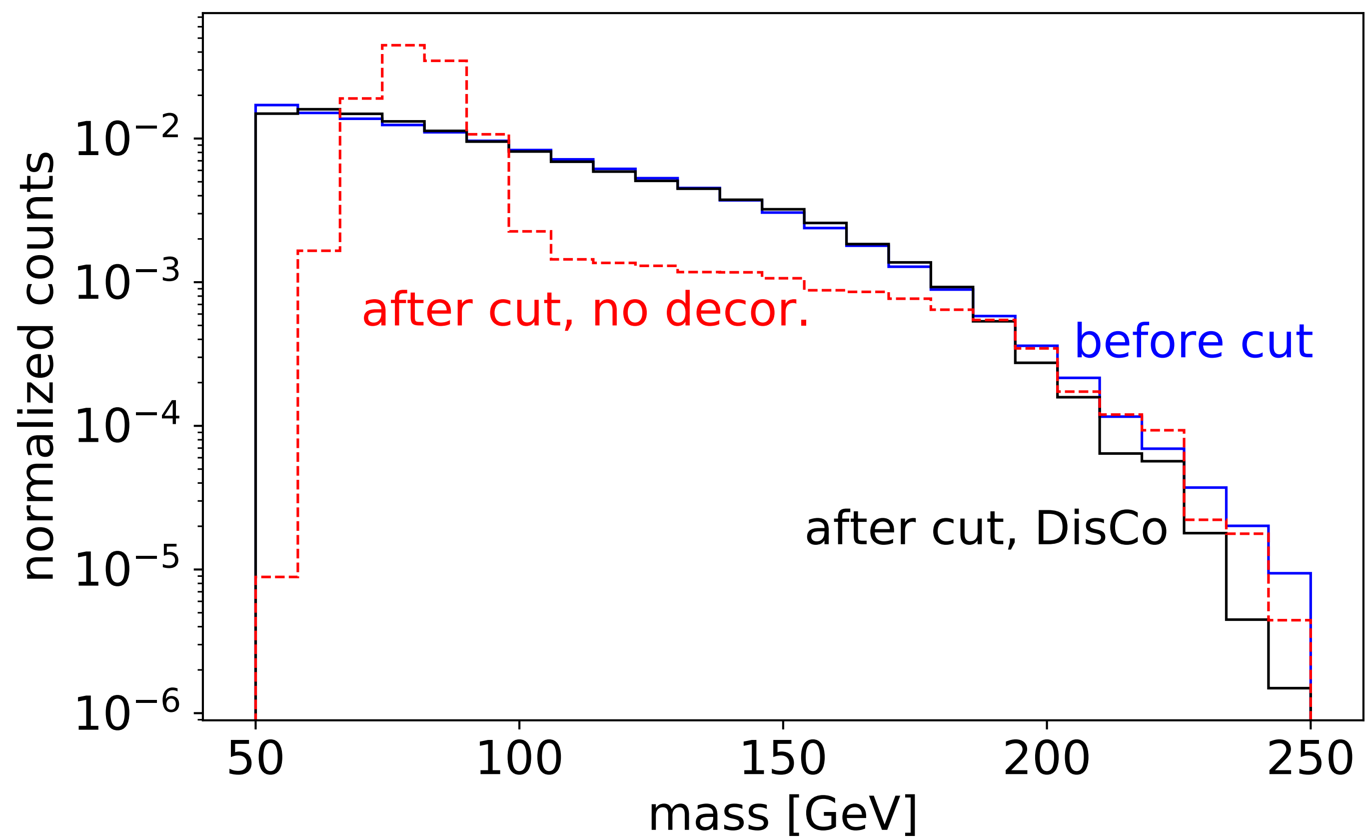




O. Kitouni, **BPN**, C. Weisser, M. Williams, 2010.09745

Distance Correlation (DisCo)





G. Kasieczka and D. Shih, 2001.05310