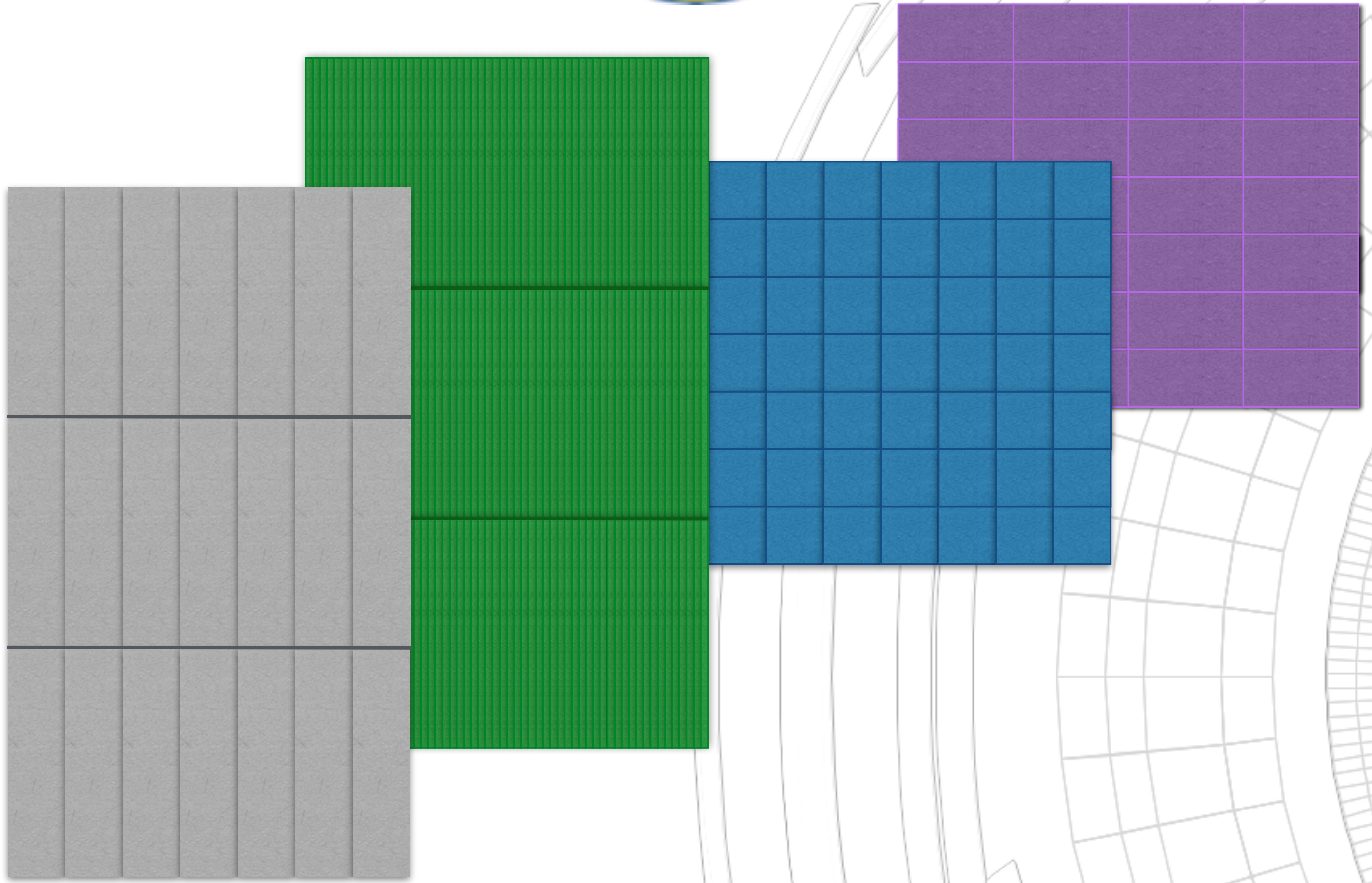
Uncertainty Mitigation with ML And A Cautionary Tale

Aishik Ghosh, Ben Nachman, Daniel Whiteson

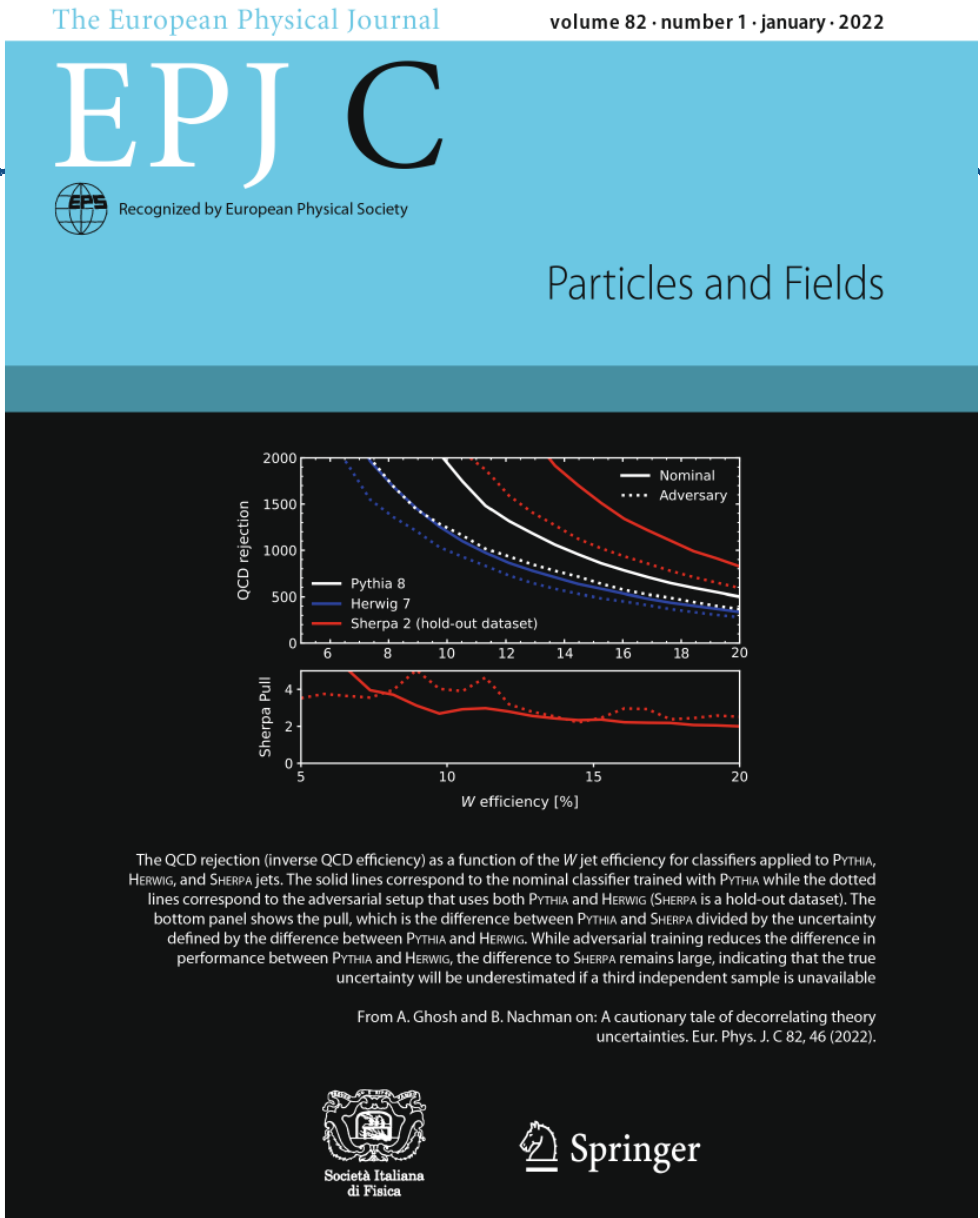
IML Workshop

11 May 2022



Speaking at an ML workshop, **one instance where you shouldn't use ML methods**:

theory uncertainty mitigation



The European Physical Journal

volume 82 · number 1 · january · 2022

EPJ C

Recognized by European Physical Society

Particles and Fields

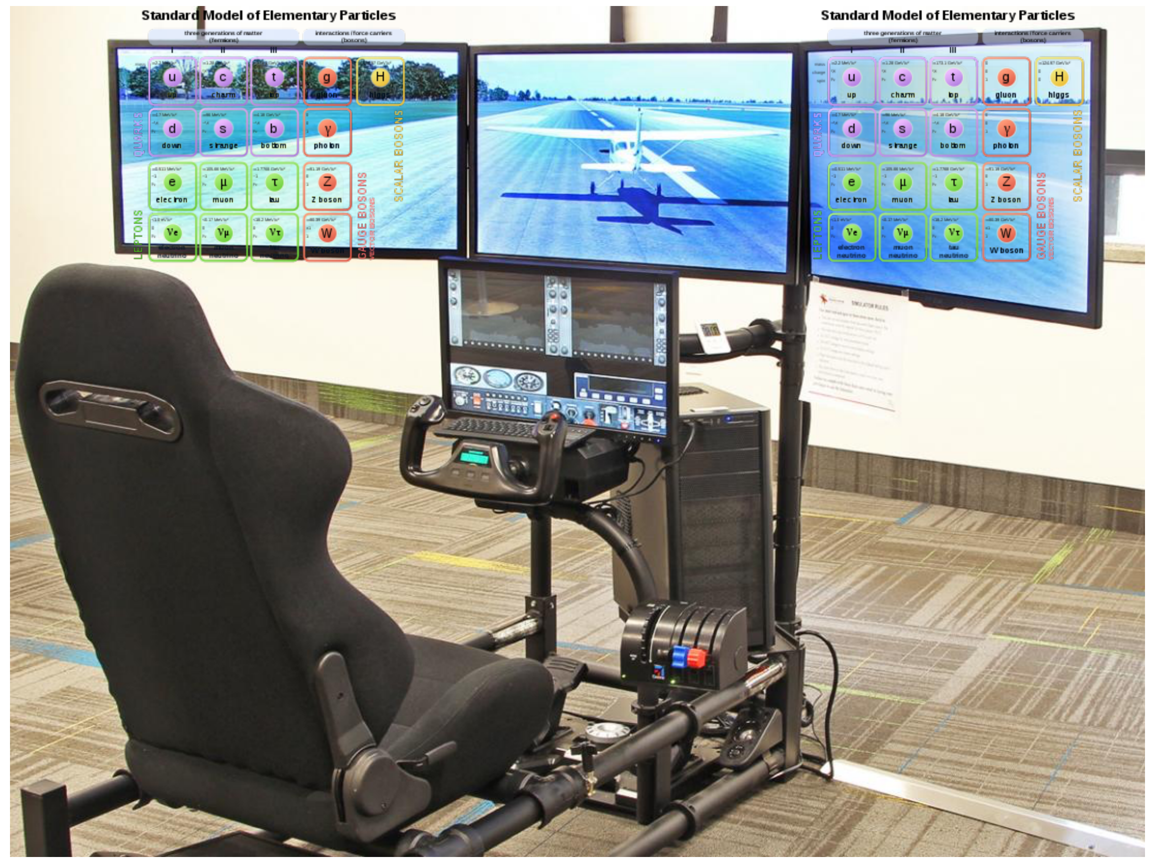
The QCD rejection (inverse QCD efficiency) as a function of the *W* jet efficiency for classifiers applied to Pythia, Herwig, and Sherpa jets. The solid lines correspond to the nominal classifier trained with Pythia while the dotted lines correspond to the adversarial setup that uses both Pythia and Herwig (Sherpa is a hold-out dataset). The bottom panel shows the pull, which is the difference between Pythia and Sherpa divided by the uncertainty defined by the difference between Pythia and Herwig. While adversarial training reduces the difference in performance between Pythia and Herwig, the difference to Sherpa remains large, indicating that the true uncertainty will be underestimated if a third independent sample is unavailable

From A. Ghosh and B. Nachman on: A cautionary tale of decorrelating theory uncertainties. Eur. Phys. J. C 82, 46 (2022).

Società Italiana di Fisica

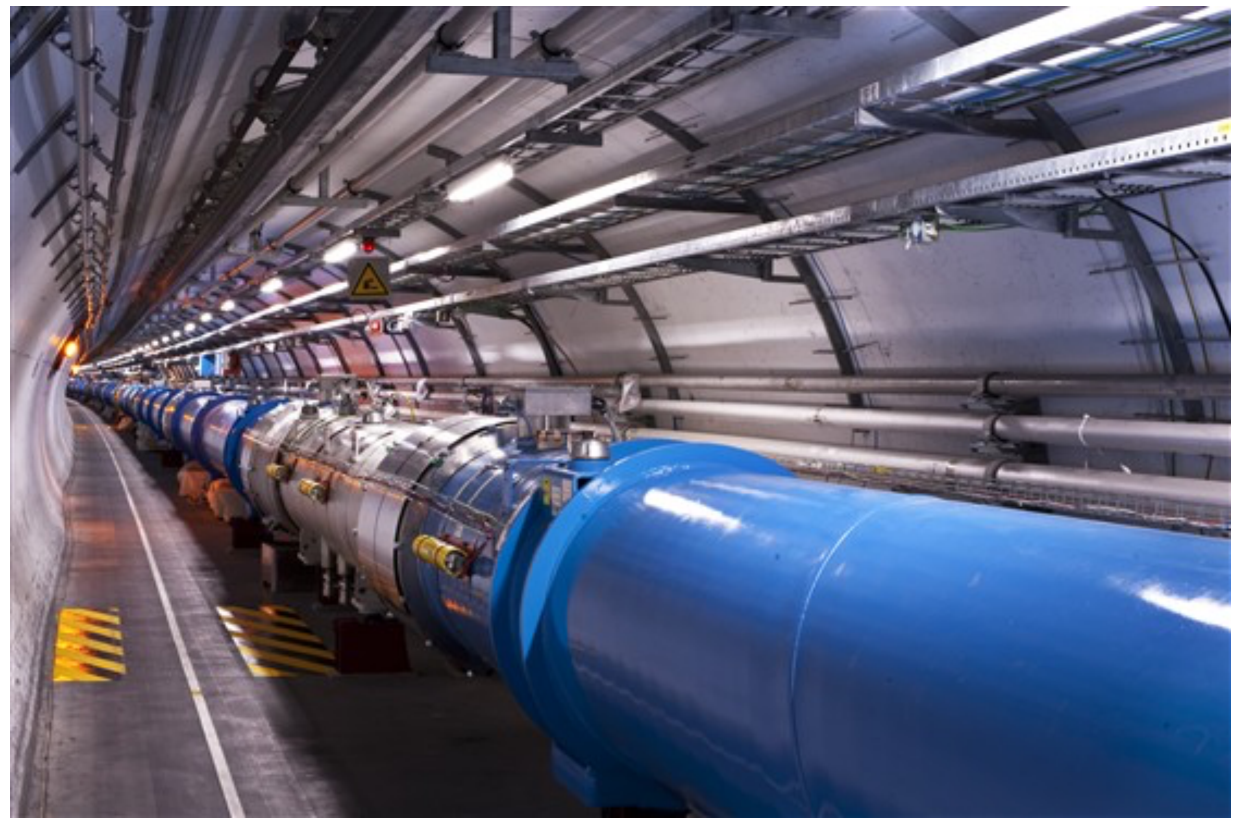
Springer

Biases and systematic uncertainties


Standard Model of Elementary Particles
Standard Model of Elementary Particles

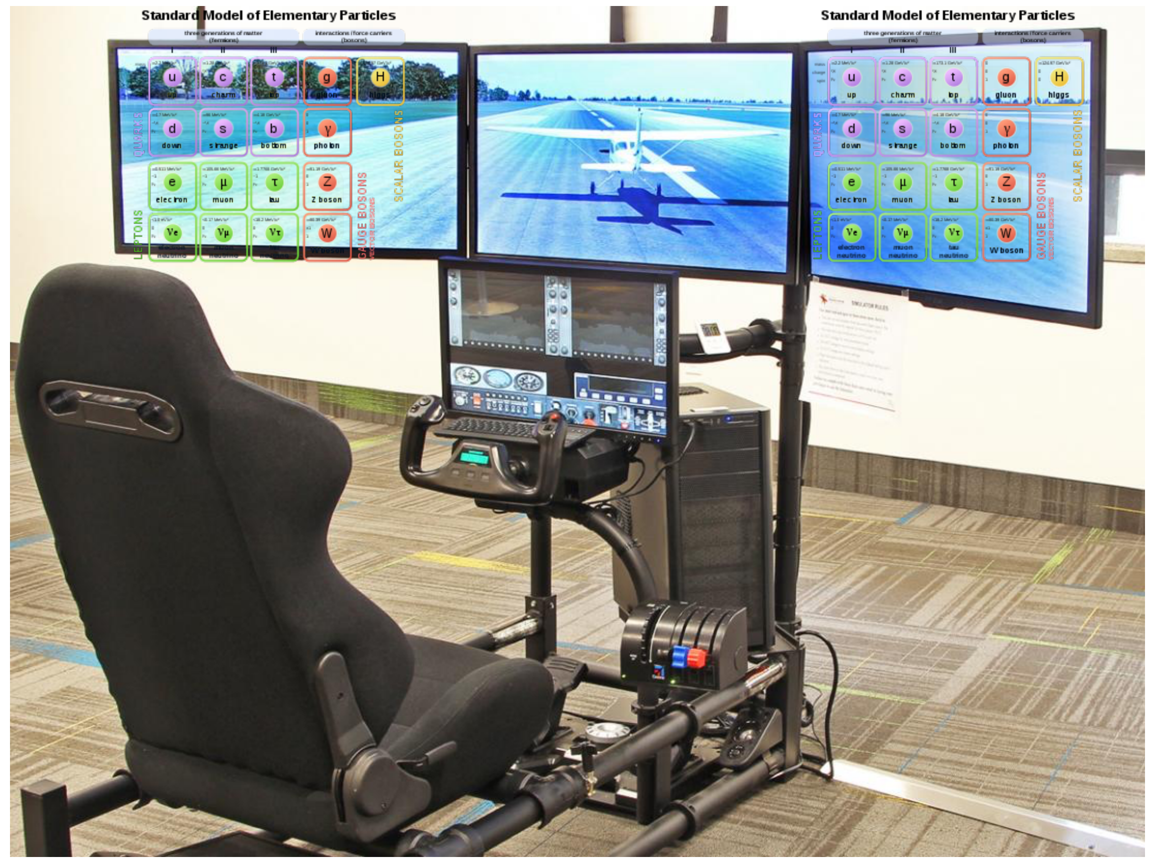
Training Data: Simulations

ML learns any biases in training data:
Known systematic imperfections in physics simulators
→ <u>systematic uncertainties</u>



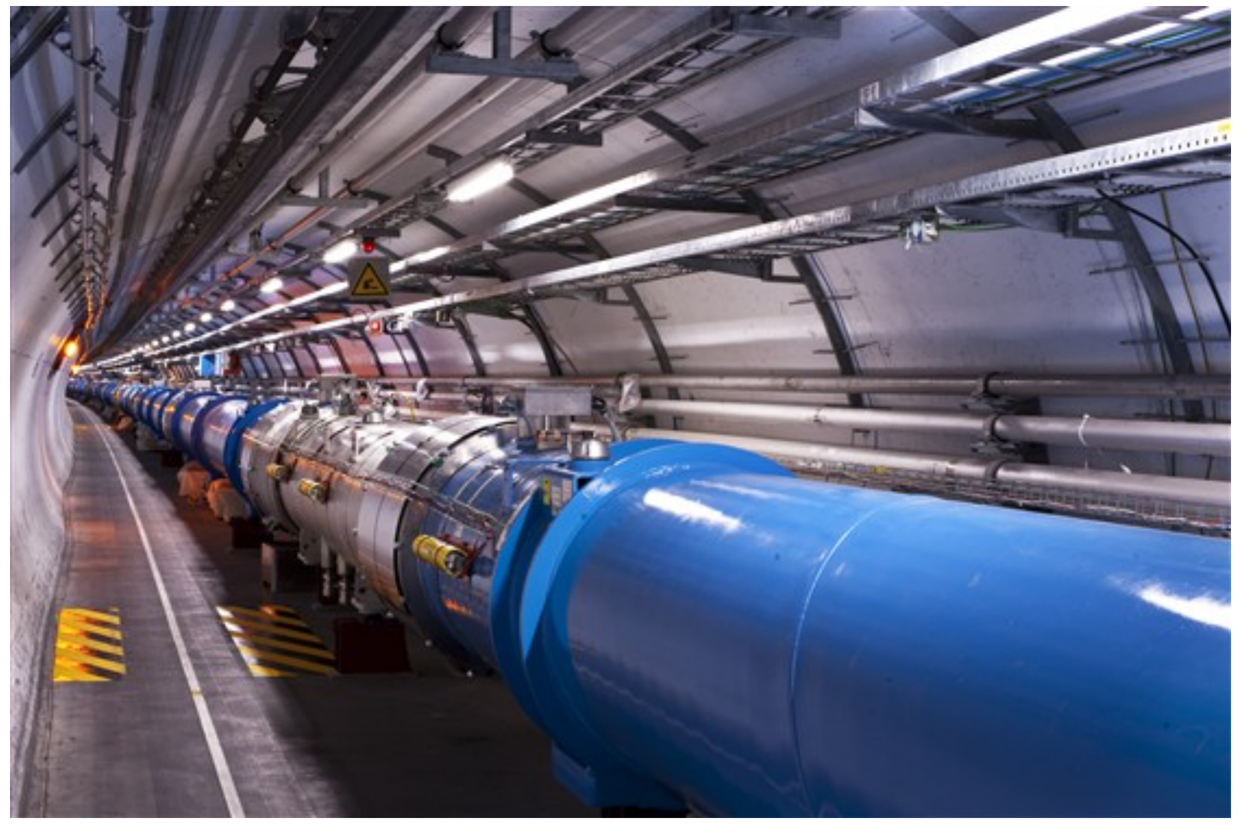
Application: Unlabelled
data from LHC

Biases and systematic uncertainties



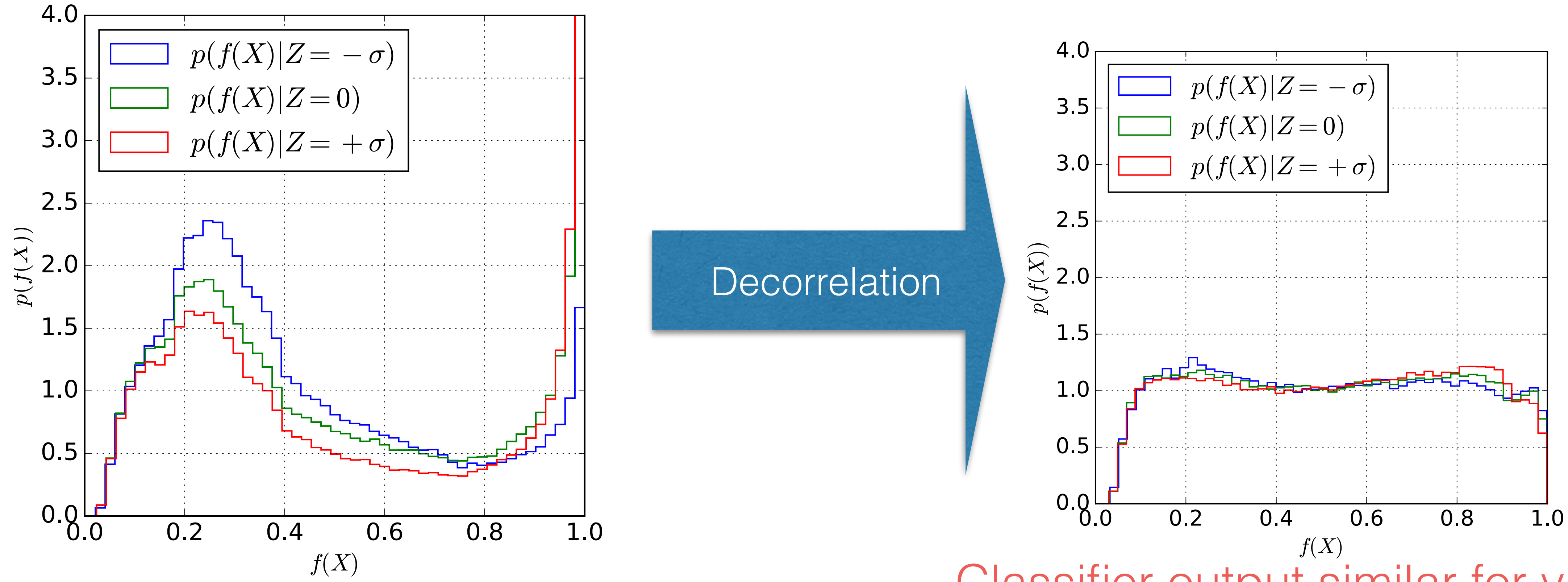
Training Data: Simulations

ML learns any biases in training data:
Known systematic imperfections in physics simulators
→ systematic uncertainties



Application: Unlabelled data from LHC

Most popular solution: Penalise network for having a biased output, eg. with adversarial decorrelation

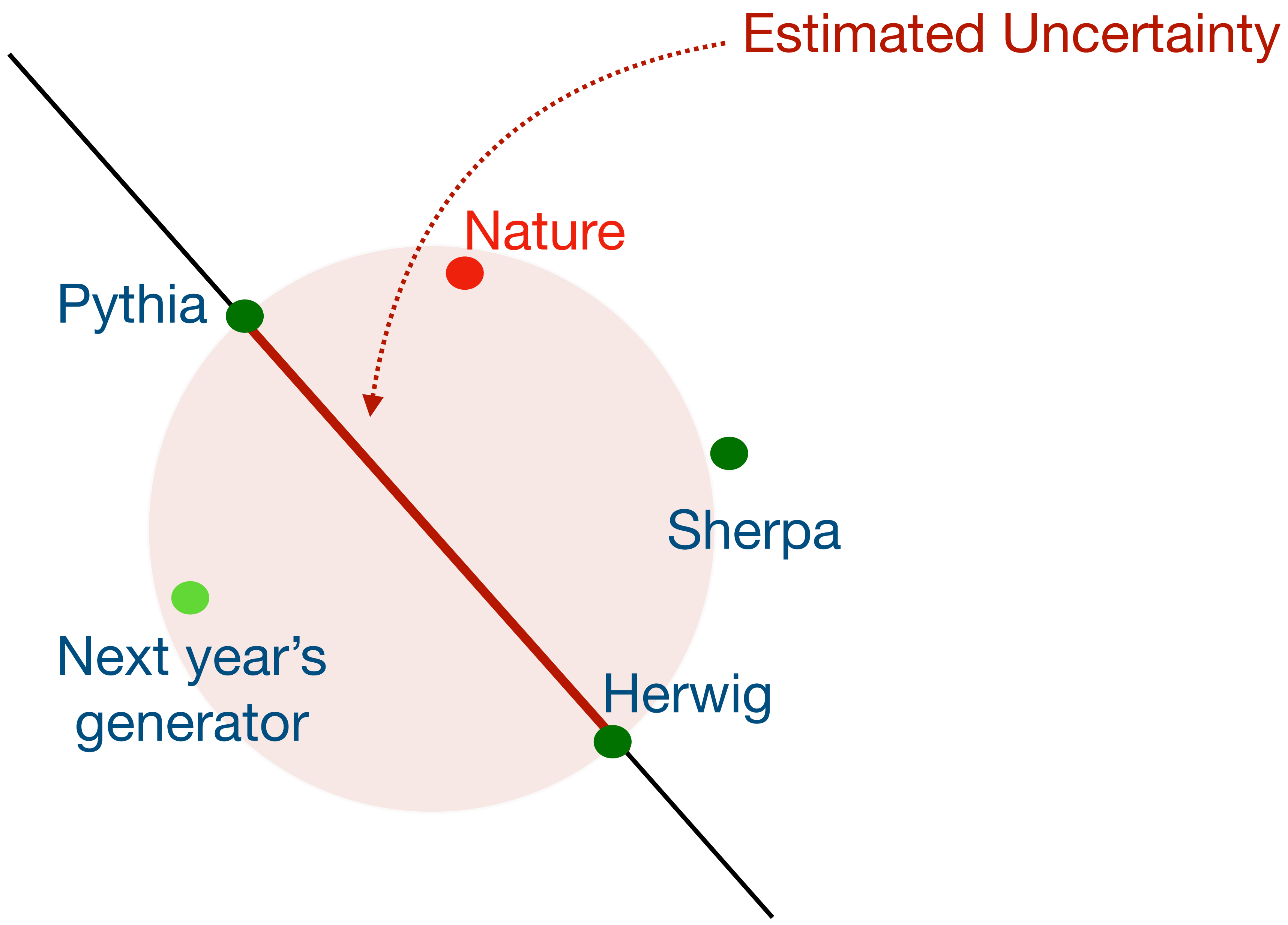


Classifier output similar for various Z

Louppe et al

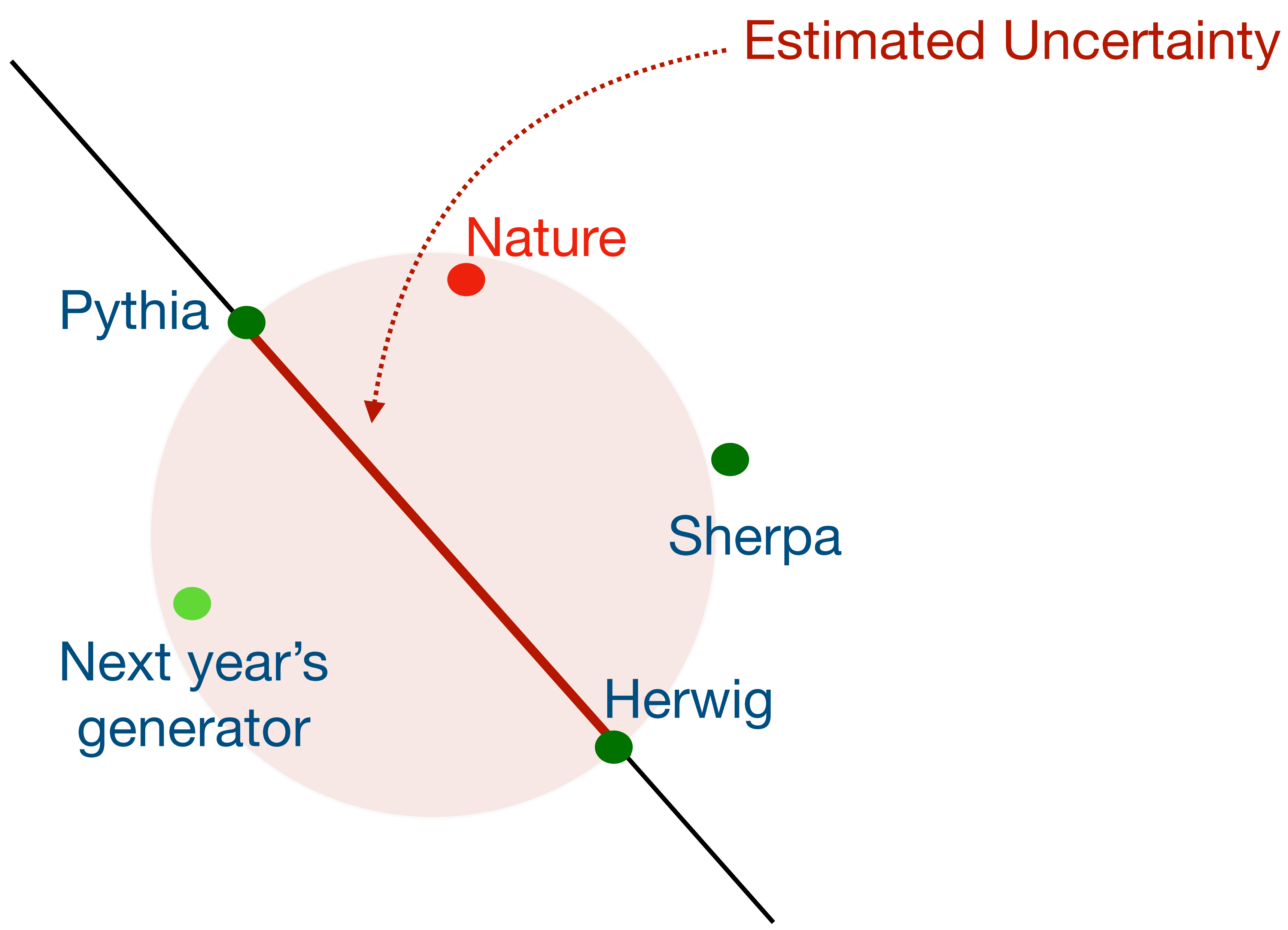
Intuition for what might go wrong with decorrelation for two-point uncertainties

Without Decorrelation

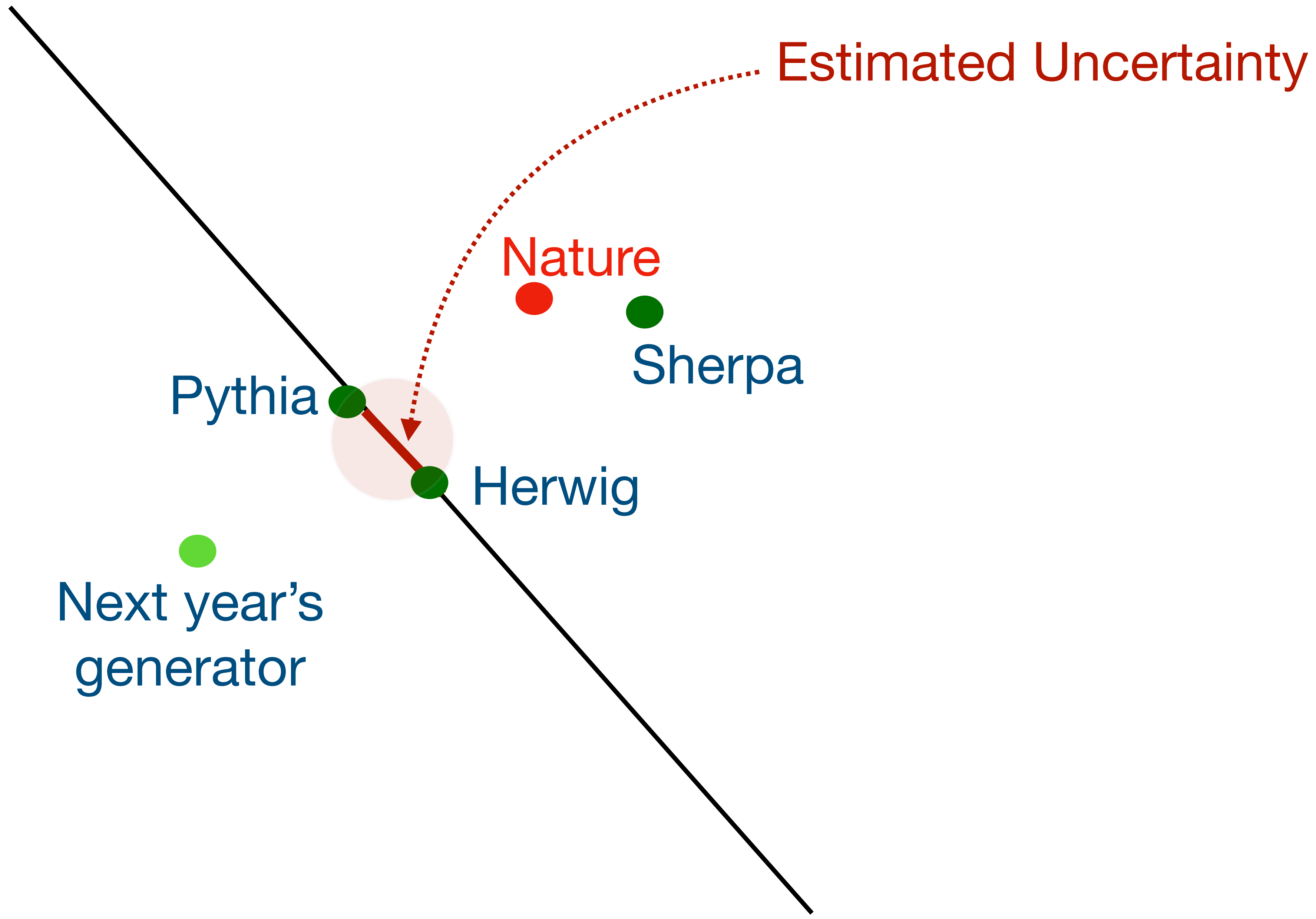


Intuition for what might go wrong with decorrelation for two-point uncertainties

Without Decorrelation



With Decorrelation



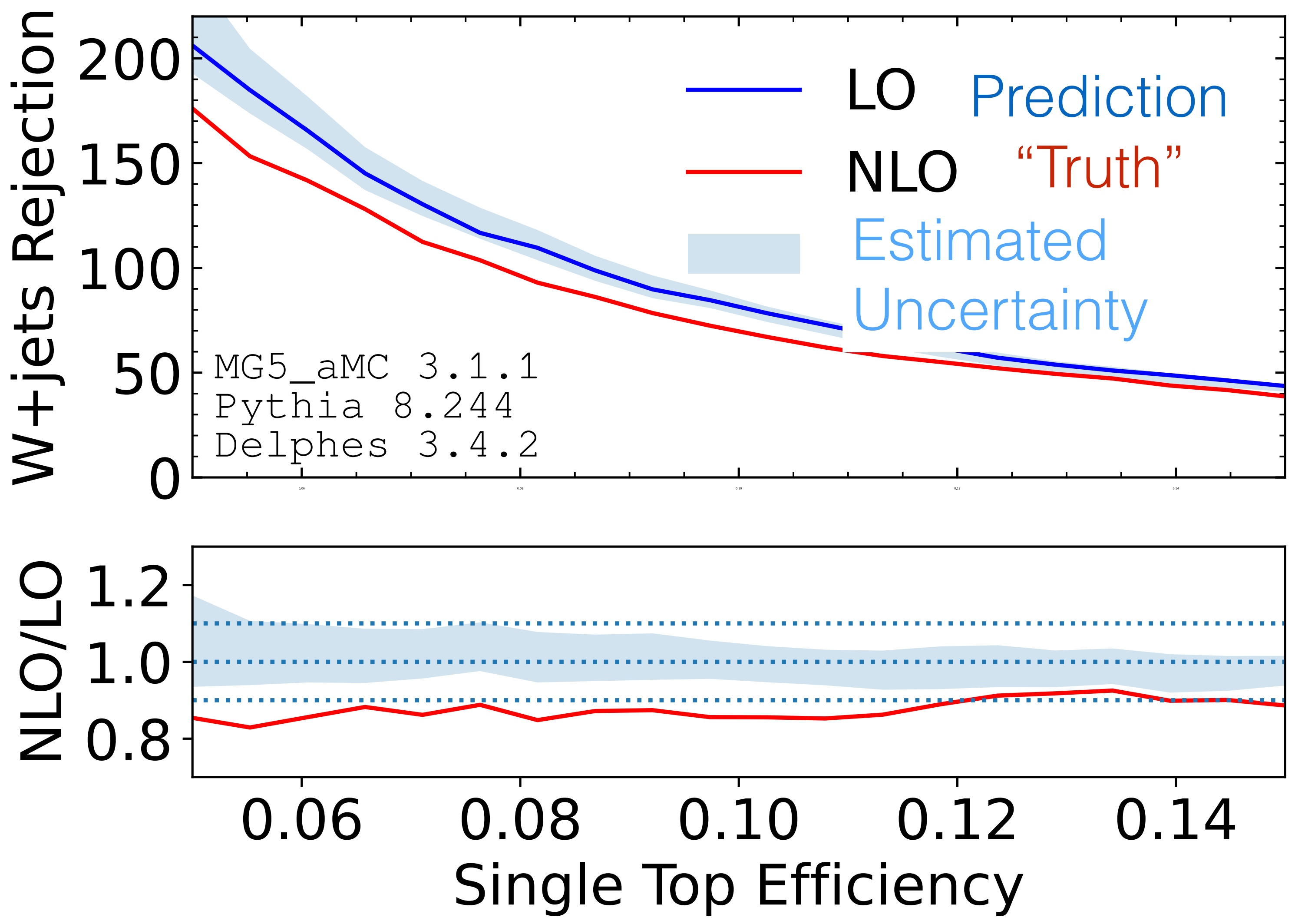
Decorrelation shrinks difference between Herwig & Pythia, but not to nature.
It does not generalise to full phase space!

Typically in ATLAS we cannot afford to have a third simulator for this cross-check

Also the case for continuous uncertainties:
Factorisation Scale Uncertainty

ROC curve (higher is better)

No adversary



Prediction

“Truth”

Estimated
Uncertainty

Also the case for continuous uncertainties: Factorisation Scale Uncertainty

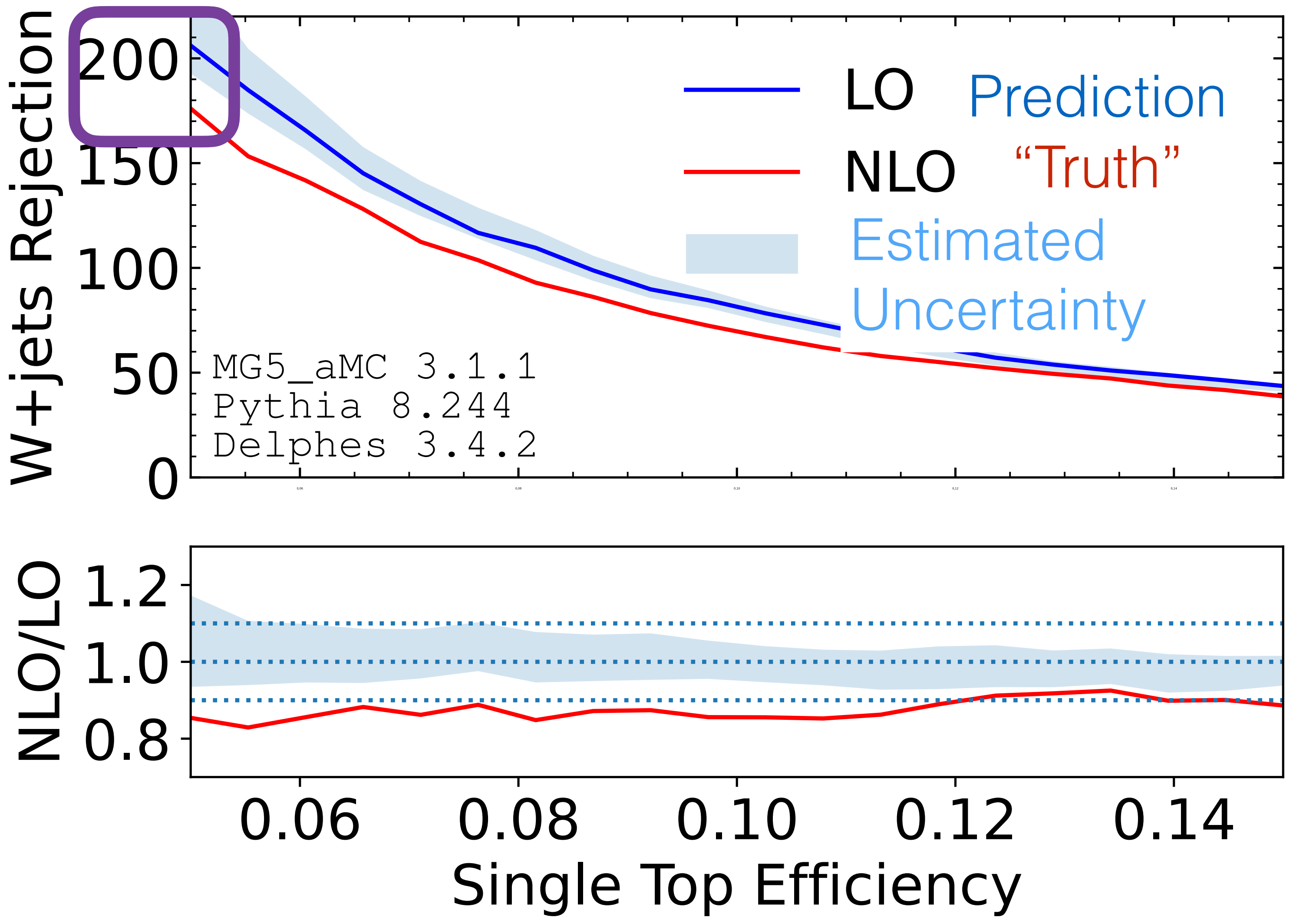
Adversary successfully **sacrifices separation power** in order to reduce difference in performance between factorisation scale variations

Cross-check with higher order physics calculations (NLO) reveals **uncertainty severely underestimated** by decorrelation approach

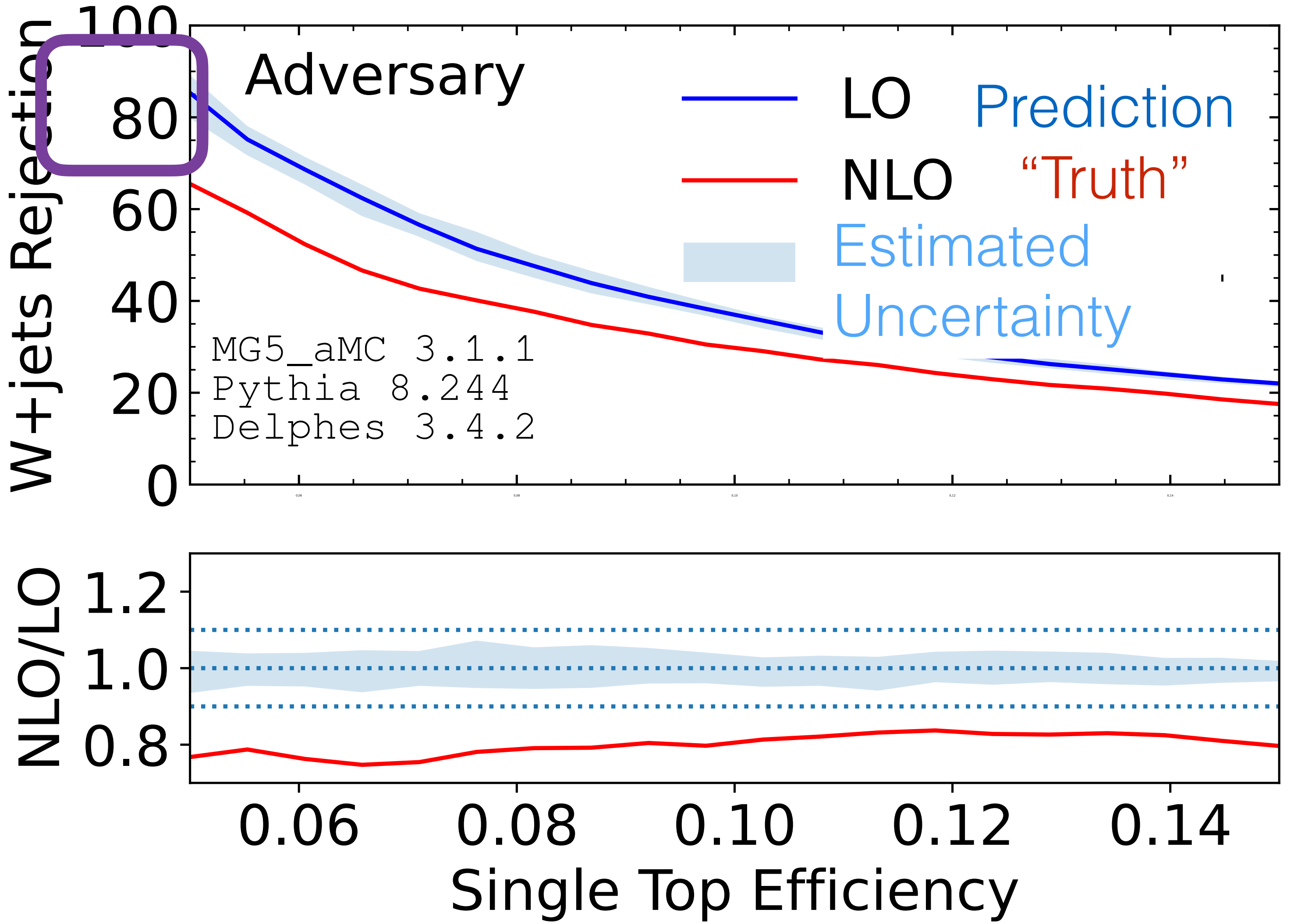
In an typical LHC analysis, a cross-check with higher-order usually unavailable

ROC curve (higher is better)

No adversary



With adversary



Also the case for continuous uncertainties: Factorisation Scale Uncertainty

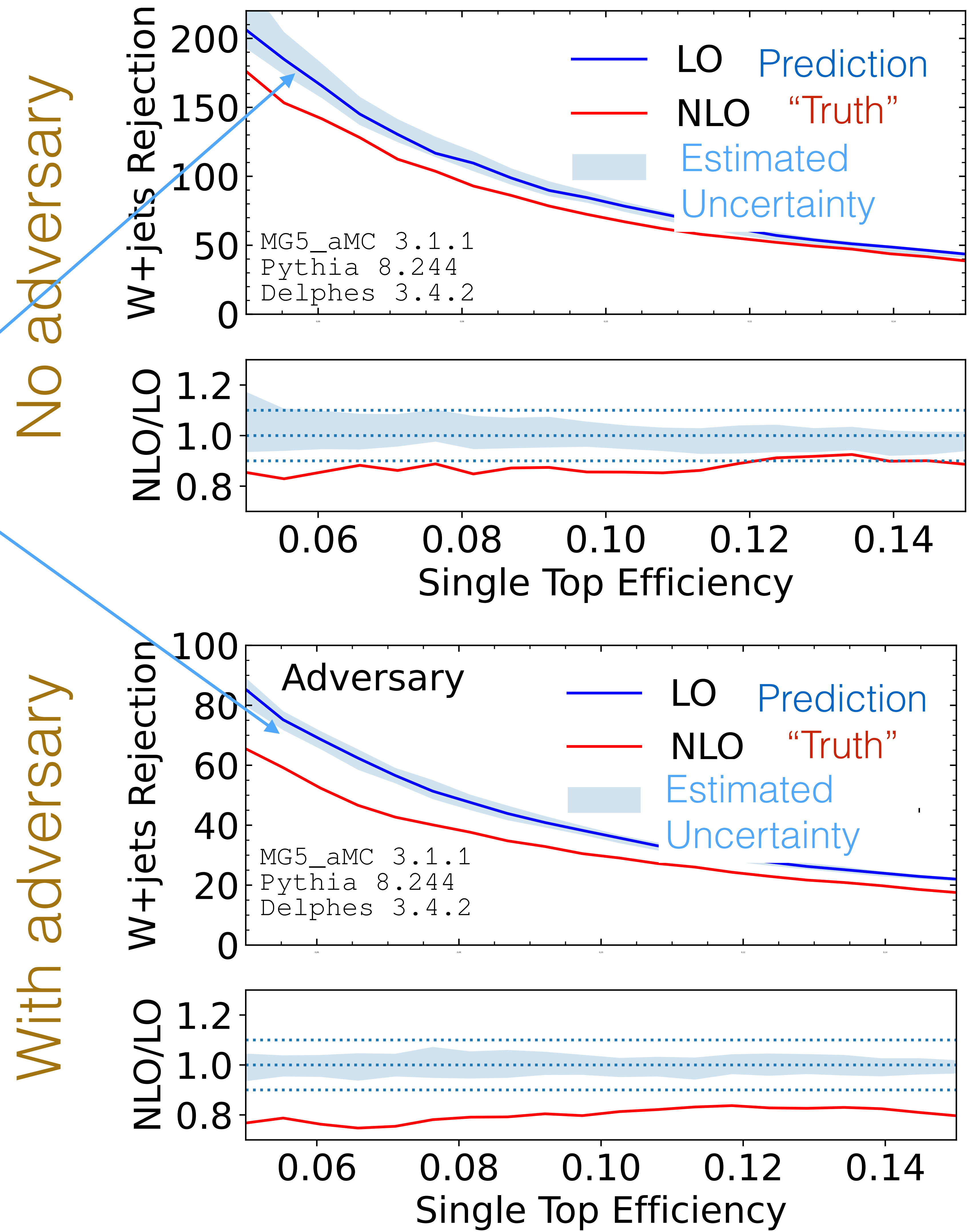
Adversary successfully **sacrifices separation power** in order to reduce difference in performance between factorisation scale variations

Cross-check with higher order physics calculations (NLO) reveals **uncertainty severely underestimated** by decorrelation approach

In an typical LHC analysis, a cross-check with higher-order usually unavailable

Decorrelation:
Only the error bars shrink, not the actual distance to NLO

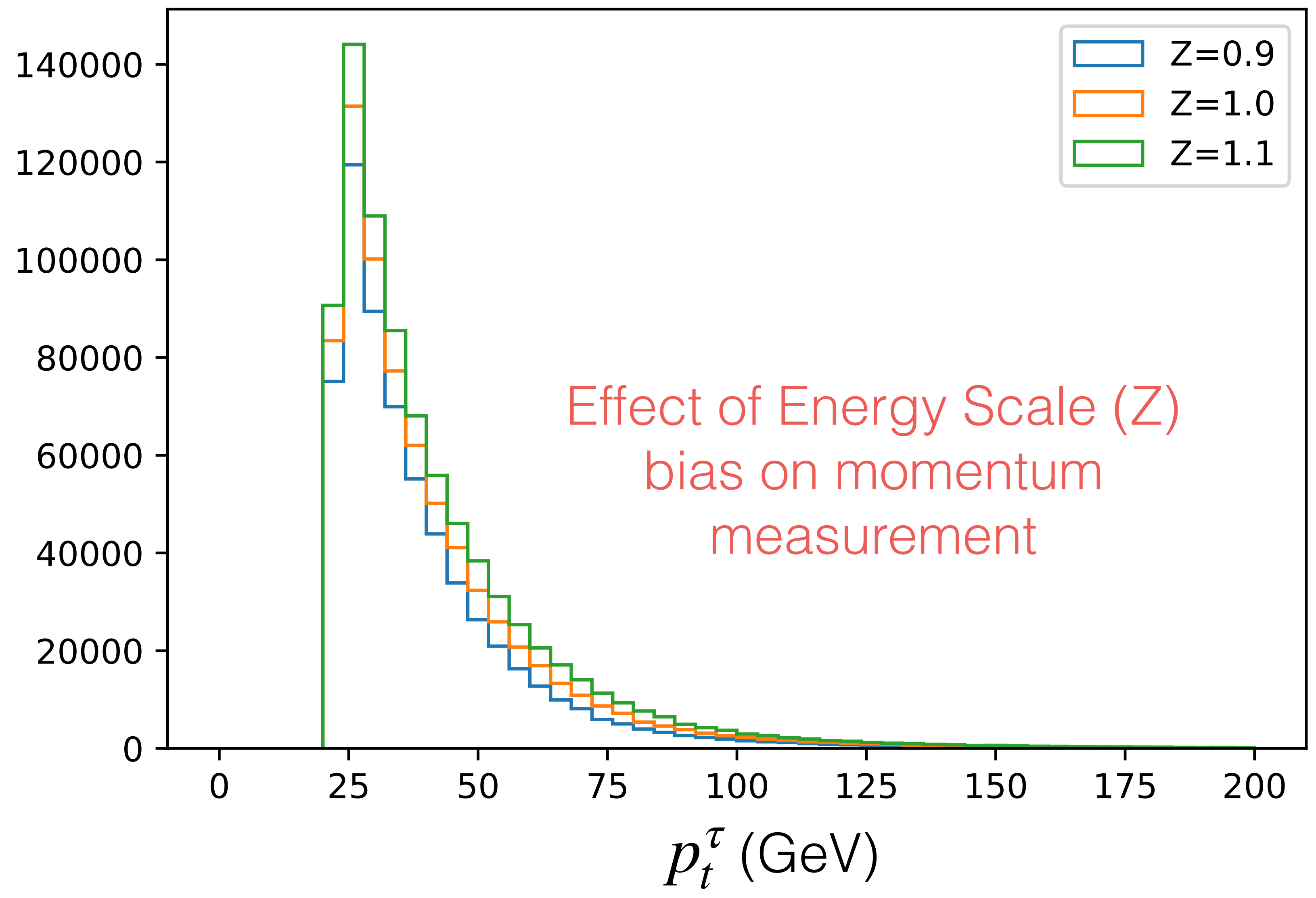
ROC curve (higher is better)



Now, where you actually can do uncertainty mitigation with ML

Experimental uncertainties: Eg. Calibration of a detector, we can produce precise simulations at each possible value of the bias

$H \rightarrow \tau\tau$, HiggsML



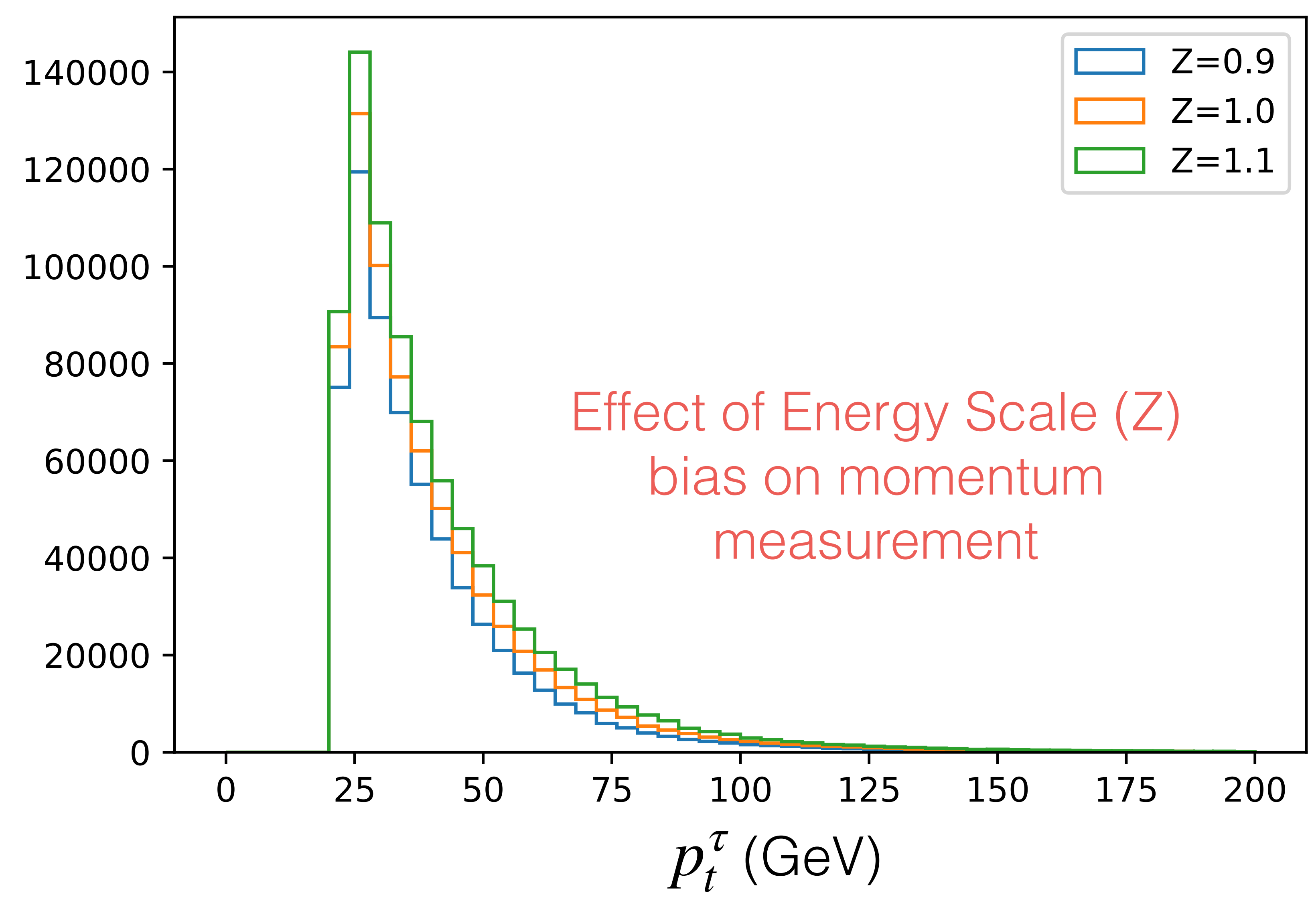
Effect of Energy Scale (Z) bias on momentum measurement

See details

Now, where you actually can do uncertainty mitigation with ML

Experimental uncertainties: Eg. Calibration of a detector, we can produce precise simulations at each possible value of the bias

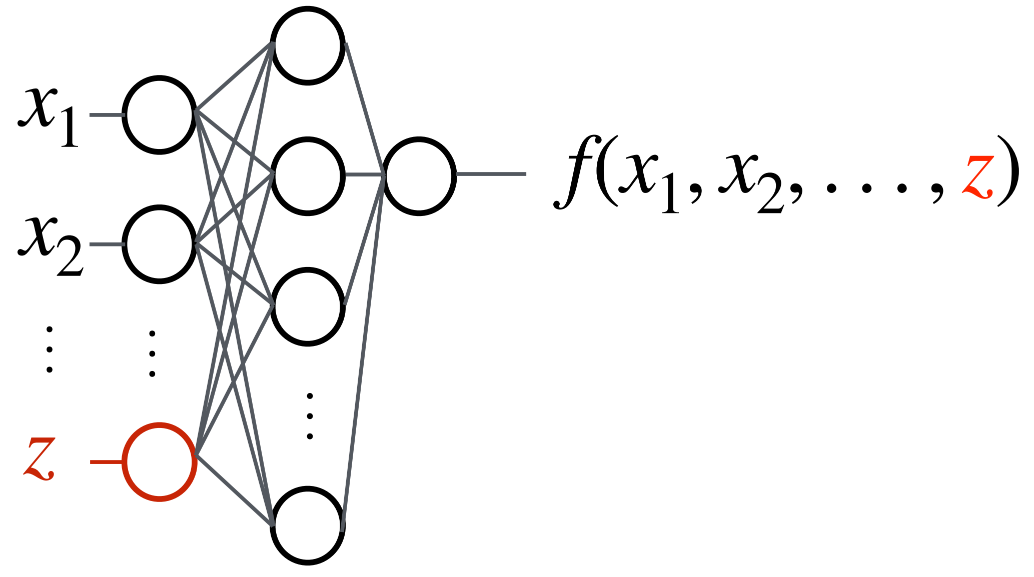
$H \to \tau\tau$, HiggsML



Effect of Energy Scale (Z) bias on momentum measurement

You can train on datasets from various values of bias

For these, we compare different bias mitigation techniques and **show the benefit of uncertainty aware networks to optimally account for additional information about the bias**



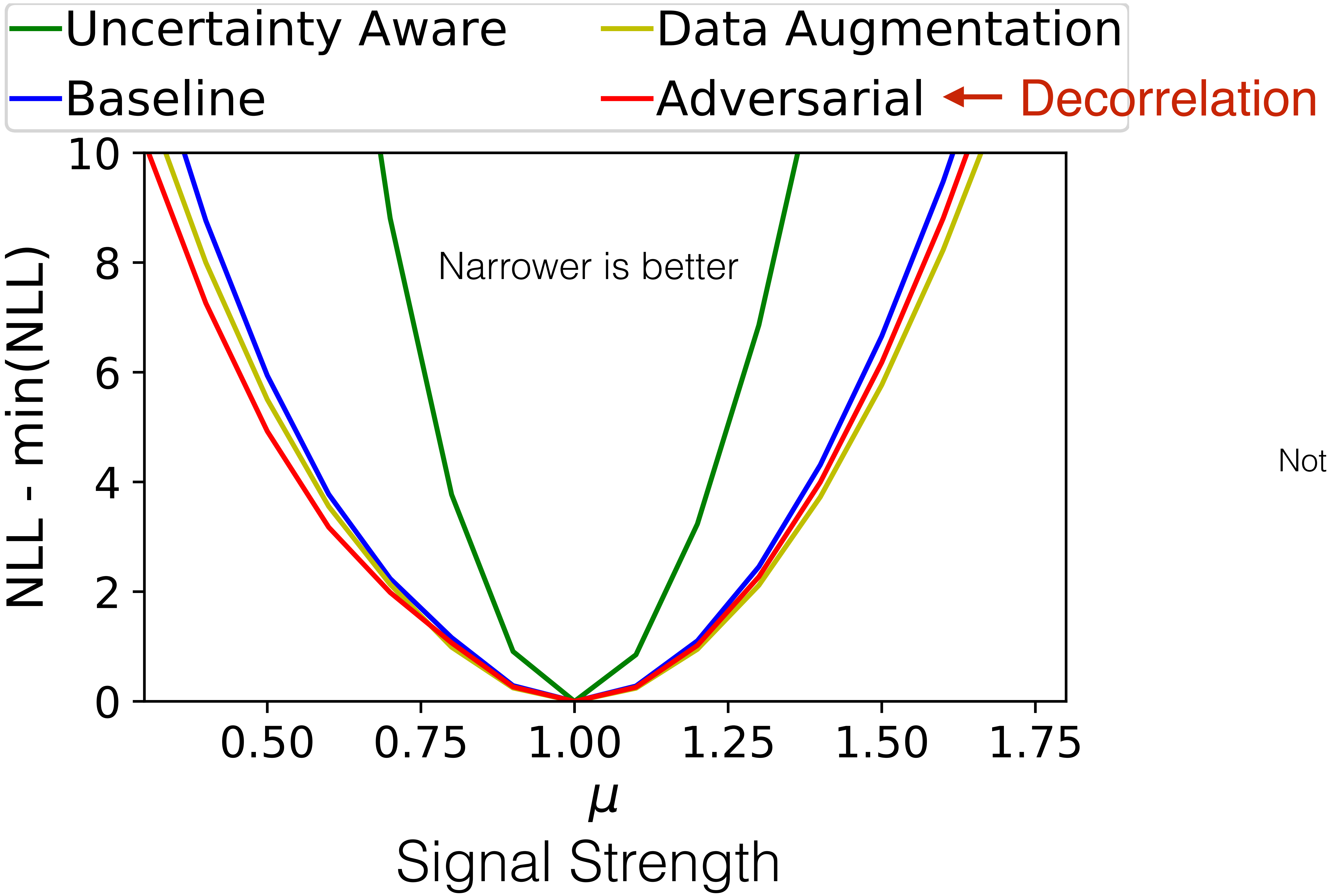
"Adaptive risk minimisation"

See details

Published: Ghosh et al. PhysRevD.104.056026 (2021)

Uncertainty-aware learning **outperforms any other method**, including decorrelation

Straightforward application to full ATLAS/CMS analysis today!



Not compared to inference aware techniques

Uncertainty-Aware classifier is much narrower ⇒ smallest [statistical + systematic] uncertainty on measurement

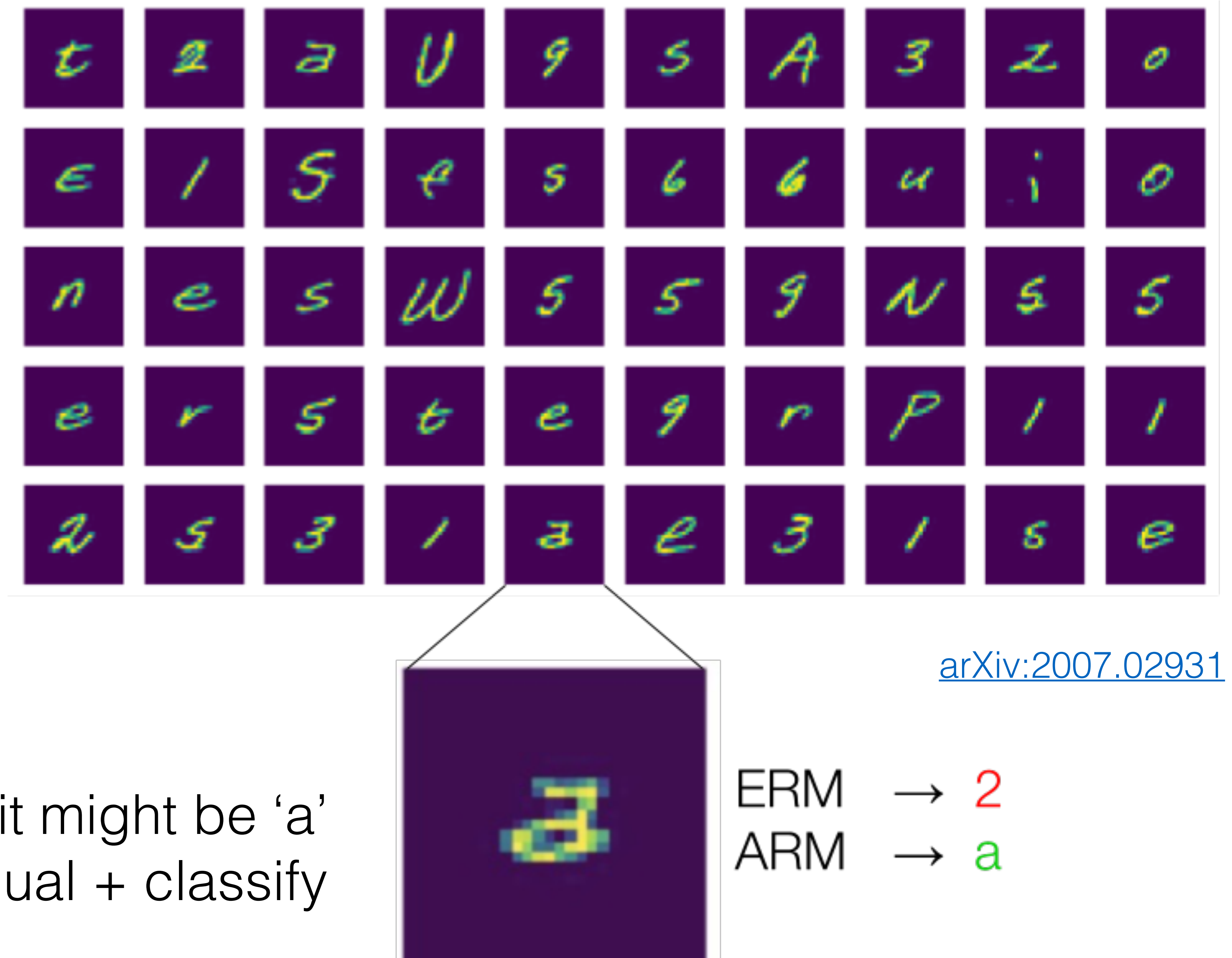
Backup

Related work in ML community: Adaptive risk minimisation

At training time data comes from various values of bias (different handwriting from different people)

At application time all of the data comes from the same bias (same person's handwriting)

If you can infer patterns about the application handwriting, you can get a better final prediction

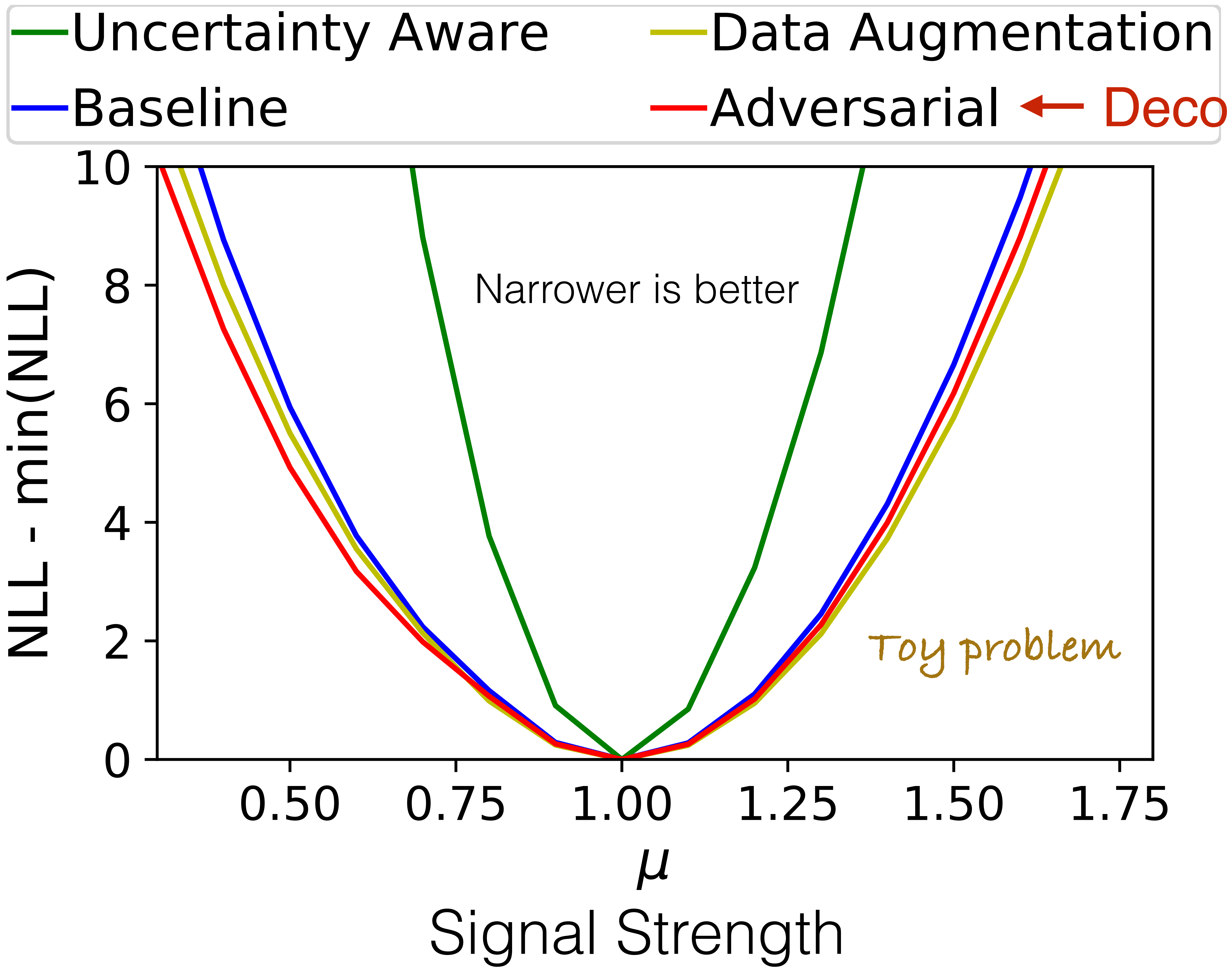


arXiv:2007.02931

For my handwriting this is '2', for yours it might be 'a'
ARM: Adapt to the individual + classify

Published: Ghosh et al. PhysRevD.104.056026 (2021)

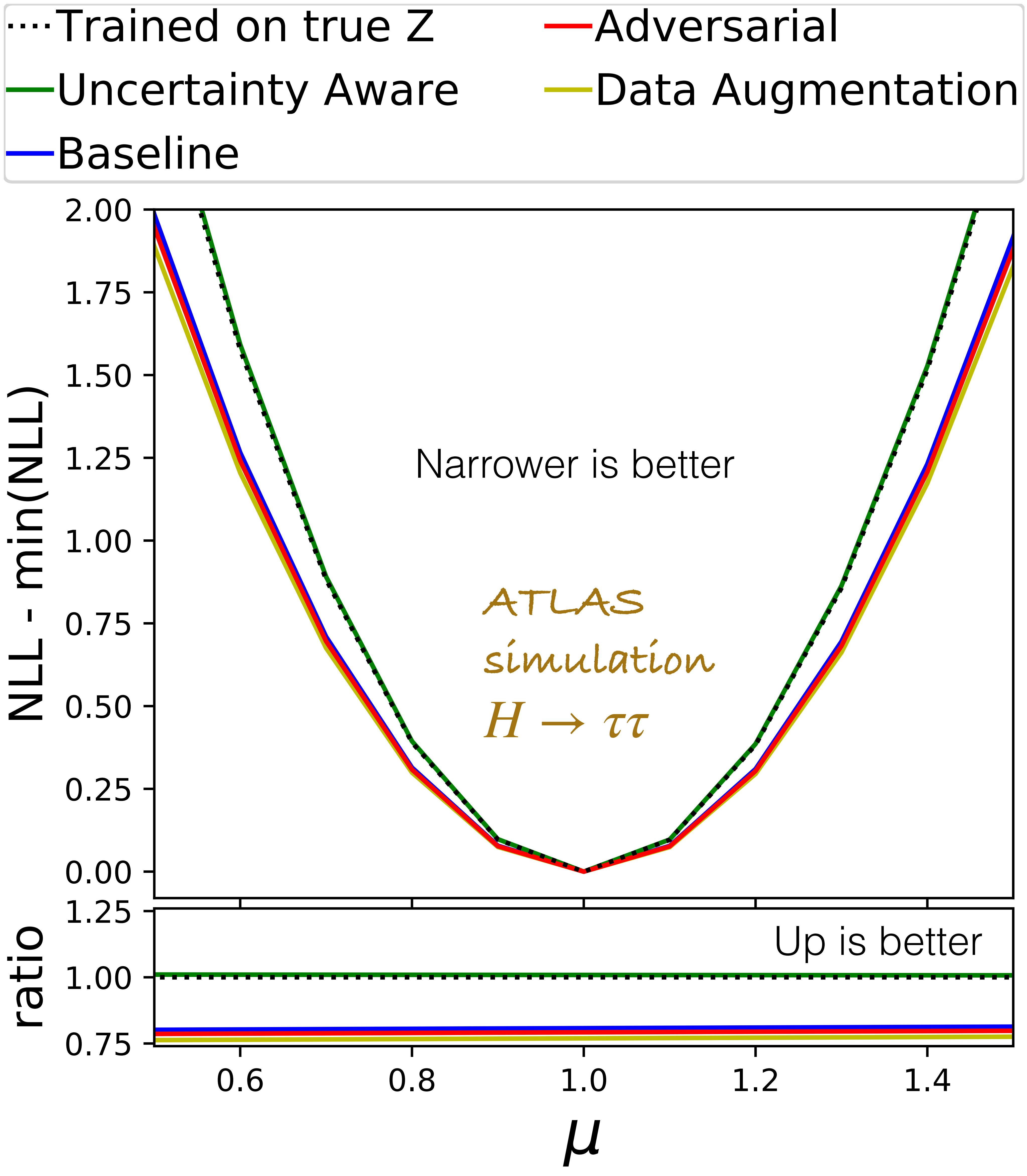
They **outperform any other method**, including decorrelation



← Decorrelation

Narrower is better

Toy problem



Narrower is better

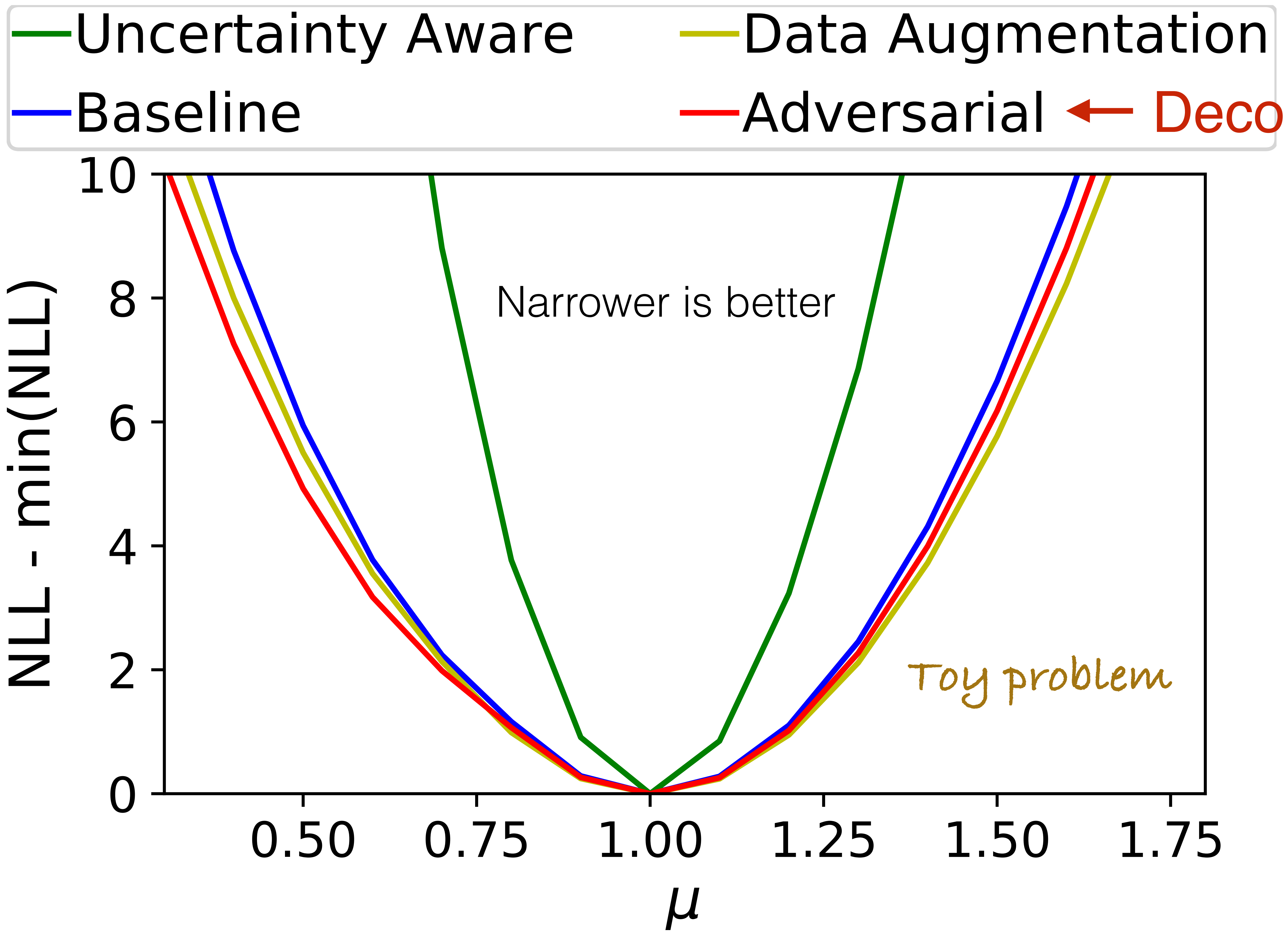
ATLAS simulation $H \rightarrow \tau\tau$

Up is better

Uncertainty-Aware classifier is much narrower ⇒ smallest [statistical + systematic] uncertainty on measurement

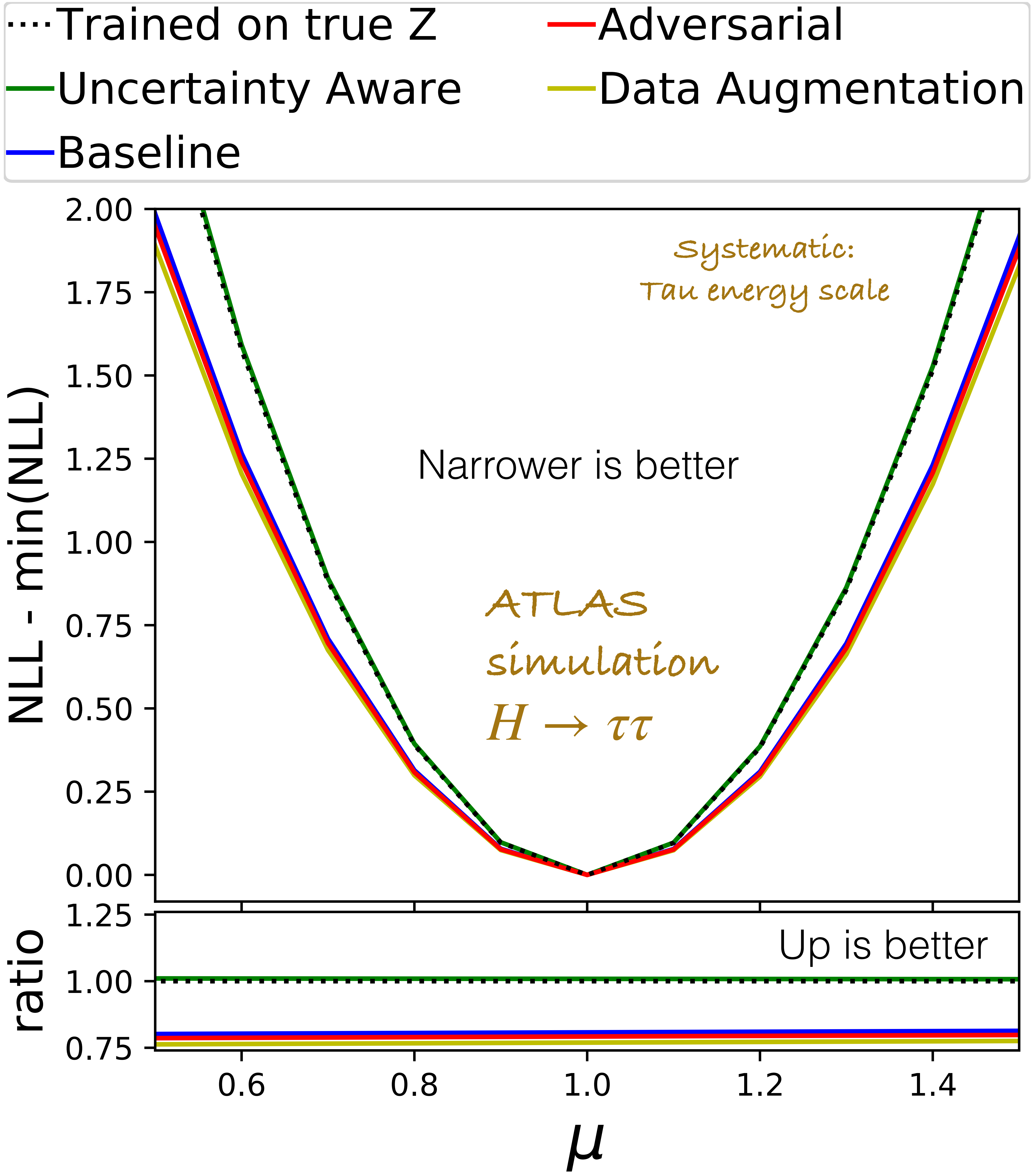
Published: Ghosh et al. PhysRevD.104.056026 (2021)

They **outperform any other method**, including decorrelation



← Decorrelation

Signal Strength

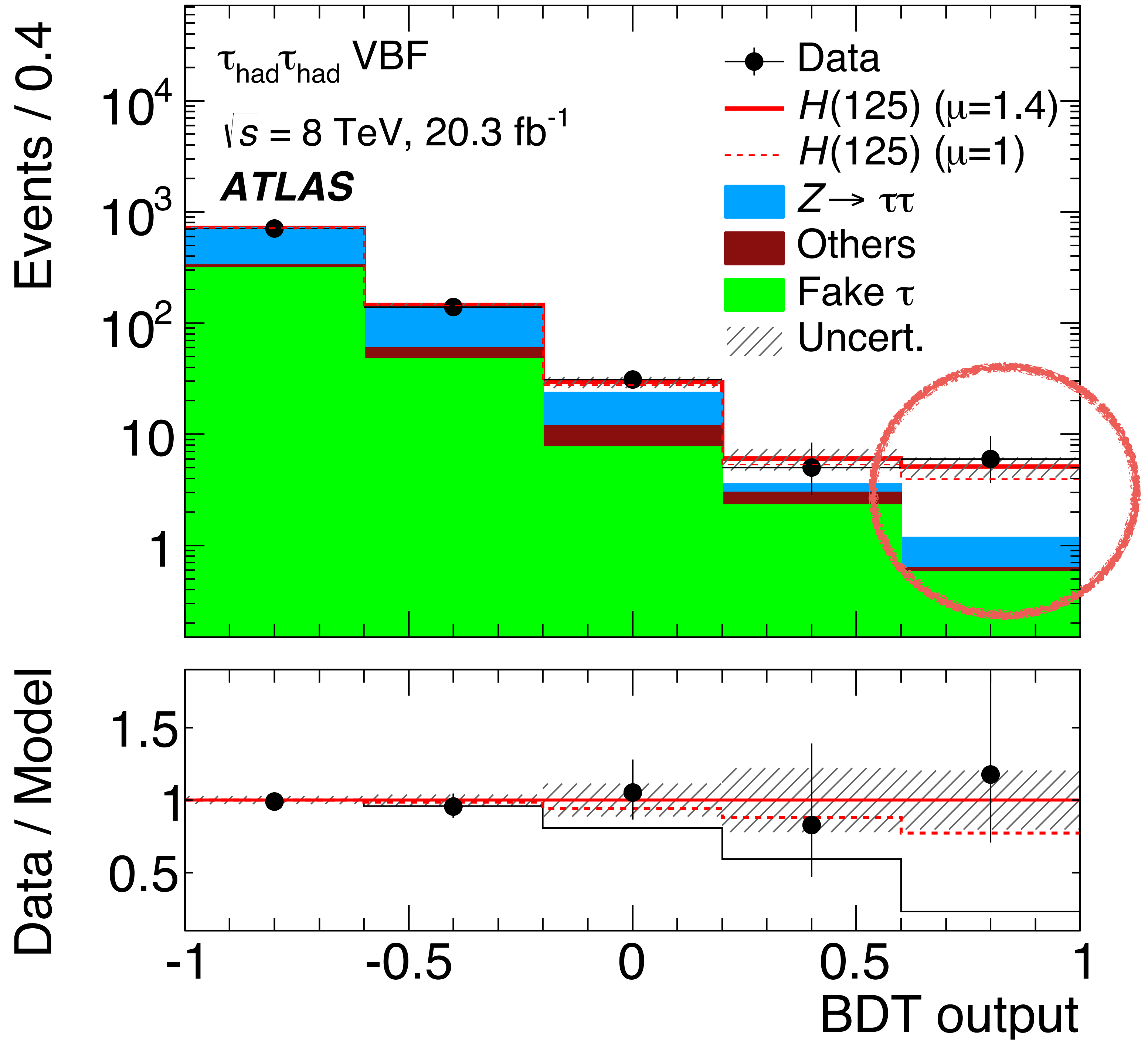


Uncertainty-Aware classifier is much narrower ⇒ smallest [statistical + systematic] uncertainty on measurement

Most common use of ML in HEP

Train ML classifier:

- Signal (example Higgs Bosons) vs background
- Output is “optimal” observable to measure theory parameters using a maximum likelihood fit over several bins of histogram



Compare various simulations to data to find best fit

1501.04943

Similar ideas: 1905.10384, 1305.7248, 1907.11674, epjconf_chep2018_06024

Decorrelating classifier from Z

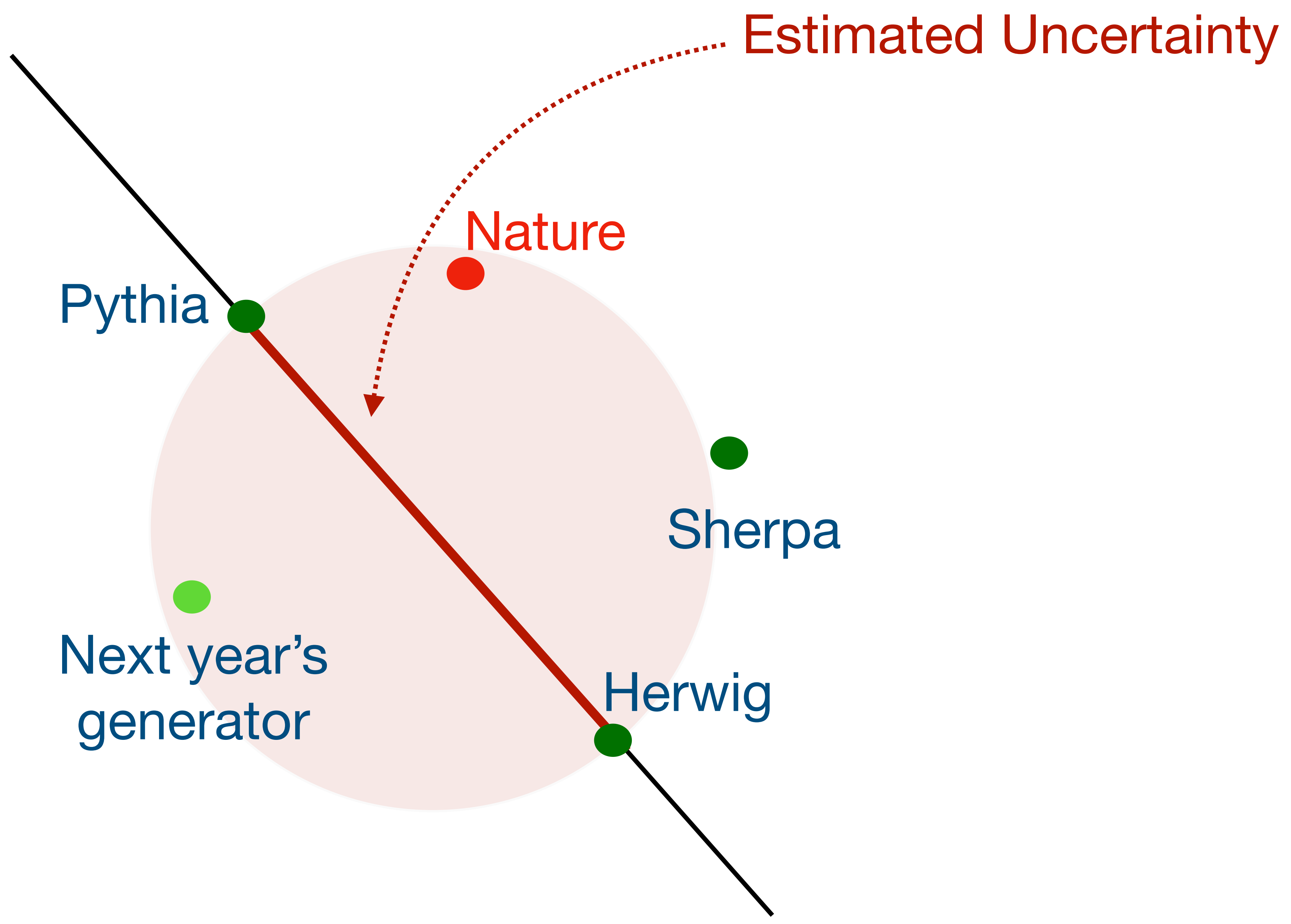
We sacrifice separation power for an unbiased classifier, expecting this reduces systematic uncertainty on final result

Great idea for original use case described in paper, but has since become a popular for all kinds of systematic uncertainties

We question the appropriateness of these techniques for theoretical uncertainties

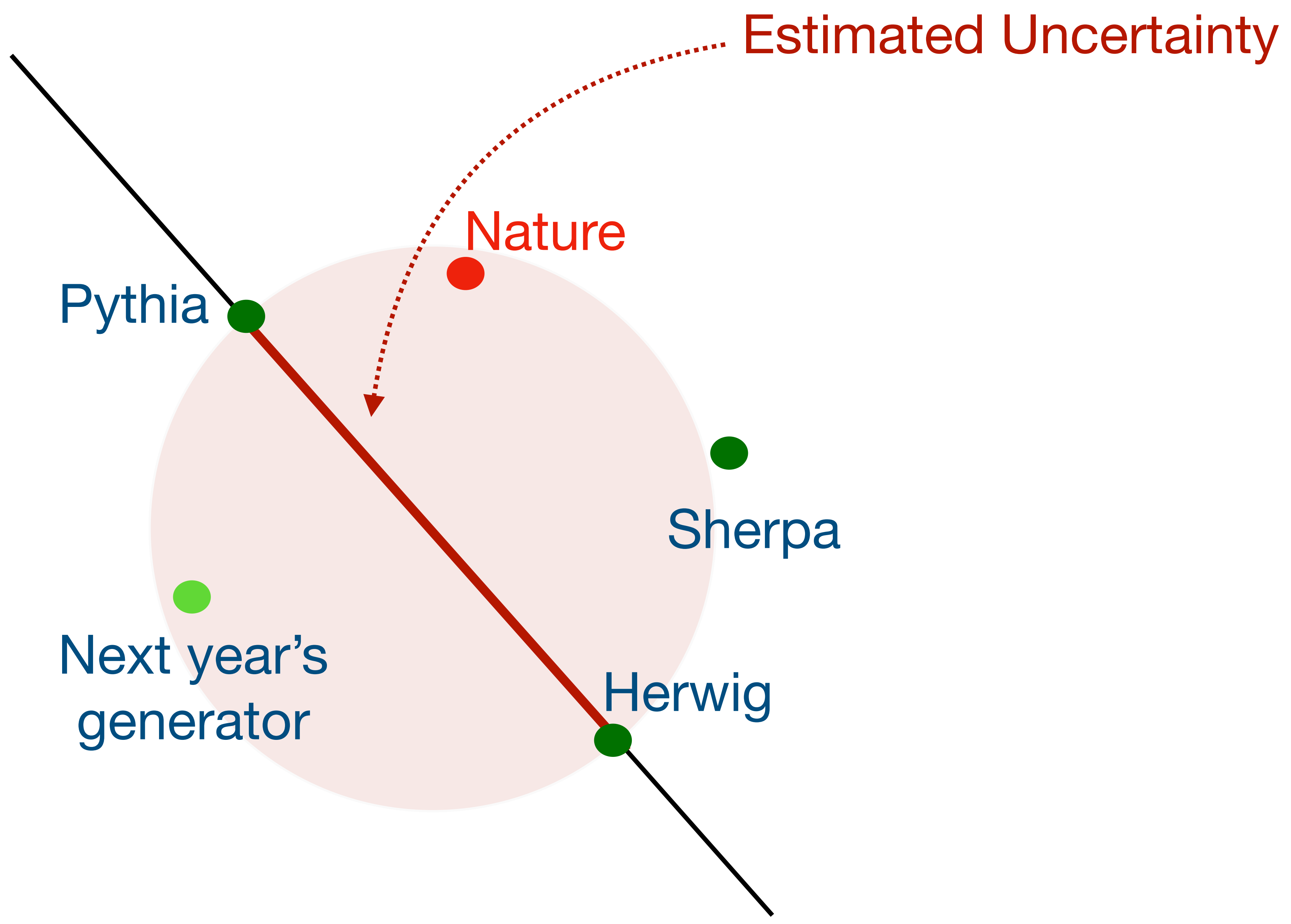
What are ‘theory uncertainties’ in HEP ?

Theory uncertainties often describe our lack of understanding / ability to simulate



What are ‘theory uncertainties’ in HEP ?

Theory uncertainties often describe our lack of understanding / ability to simulate



Eg. Hadronisation:

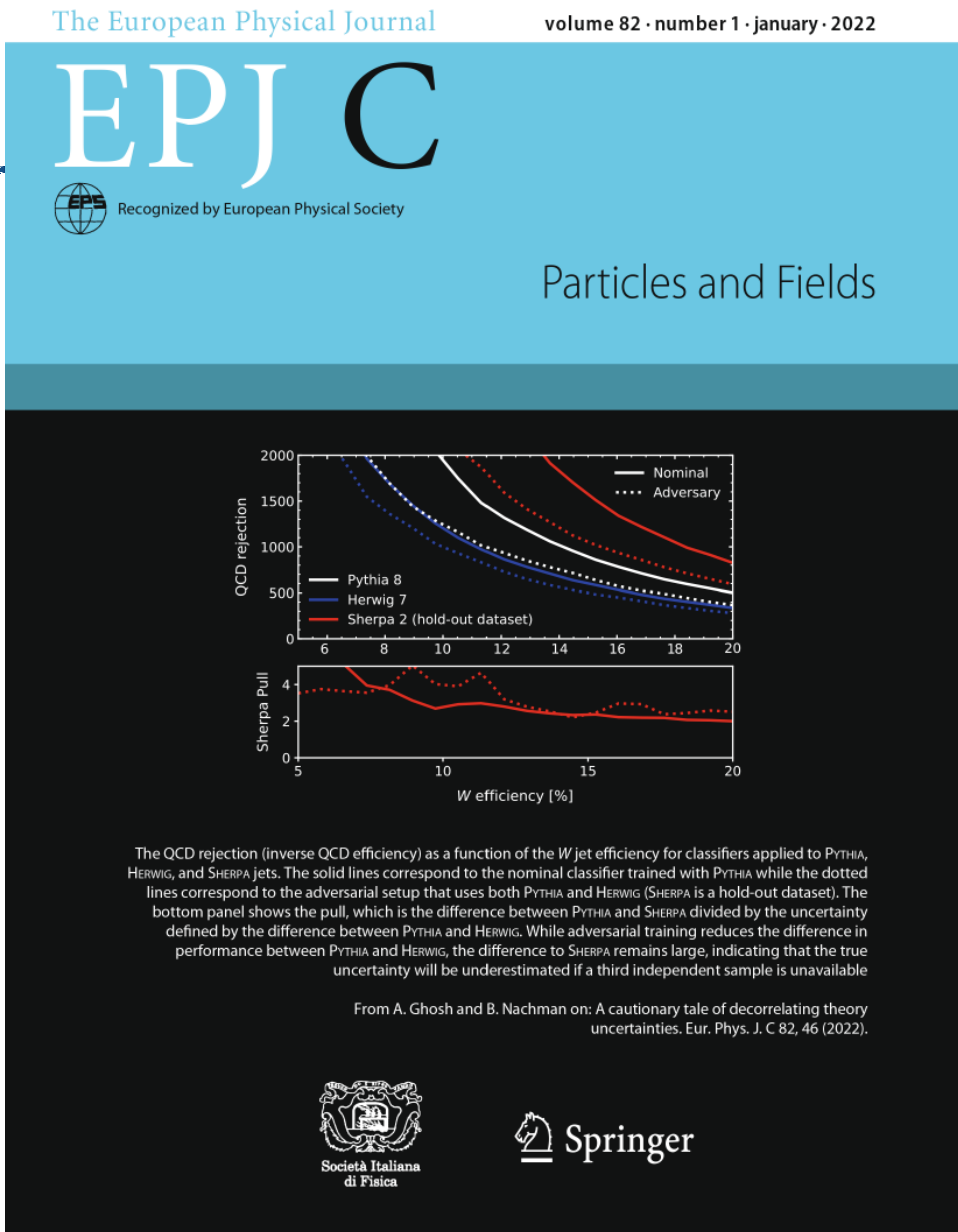
- Few different packages to simulate it
- None are correct!
- Use difference in performance of your data analysis algorithm on Pythia simulator vs Herwig simulator ad-hoc estimate of uncertainty
    - These are just 2 random points in unexplored theory space (usually we can afford to have only 2 points)

Implications beyond HEP…?

Fact that of all my ML-for-physics work, this is the one that made it to a journal cover says something about how conservative larger community still is…?

Ghosh, A., Nachman, B. Eur. Phys. J. C 82, 46 (2022)

EPJC highlight article written about our work



The European Physical Journal volume 82 · number 1 · january · 2022

EPJ C

Recognized by European Physical Society

Particles and Fields

The QCD rejection (inverse QCD efficiency) as a function of the *W* jet efficiency for classifiers applied to Pythia, Herwig, and Sherpa jets. The solid lines correspond to the nominal classifier trained with Pythia while the dotted lines correspond to the adversarial setup that uses both Pythia and Herwig (Sherpa is a hold-out dataset). The bottom panel shows the pull, which is the difference between Pythia and Sherpa divided by the uncertainty defined by the difference between Pythia and Herwig. While adversarial training reduces the difference in performance between Pythia and Herwig, the difference to Sherpa remains large, indicating that the true uncertainty will be underestimated if a third independent sample is unavailable

From A. Ghosh and B. Nachman on: A cautionary tale of decorrelating theory uncertainties. Eur. Phys. J. C 82, 46 (2022).

Società Italiana di Fisica

Springer

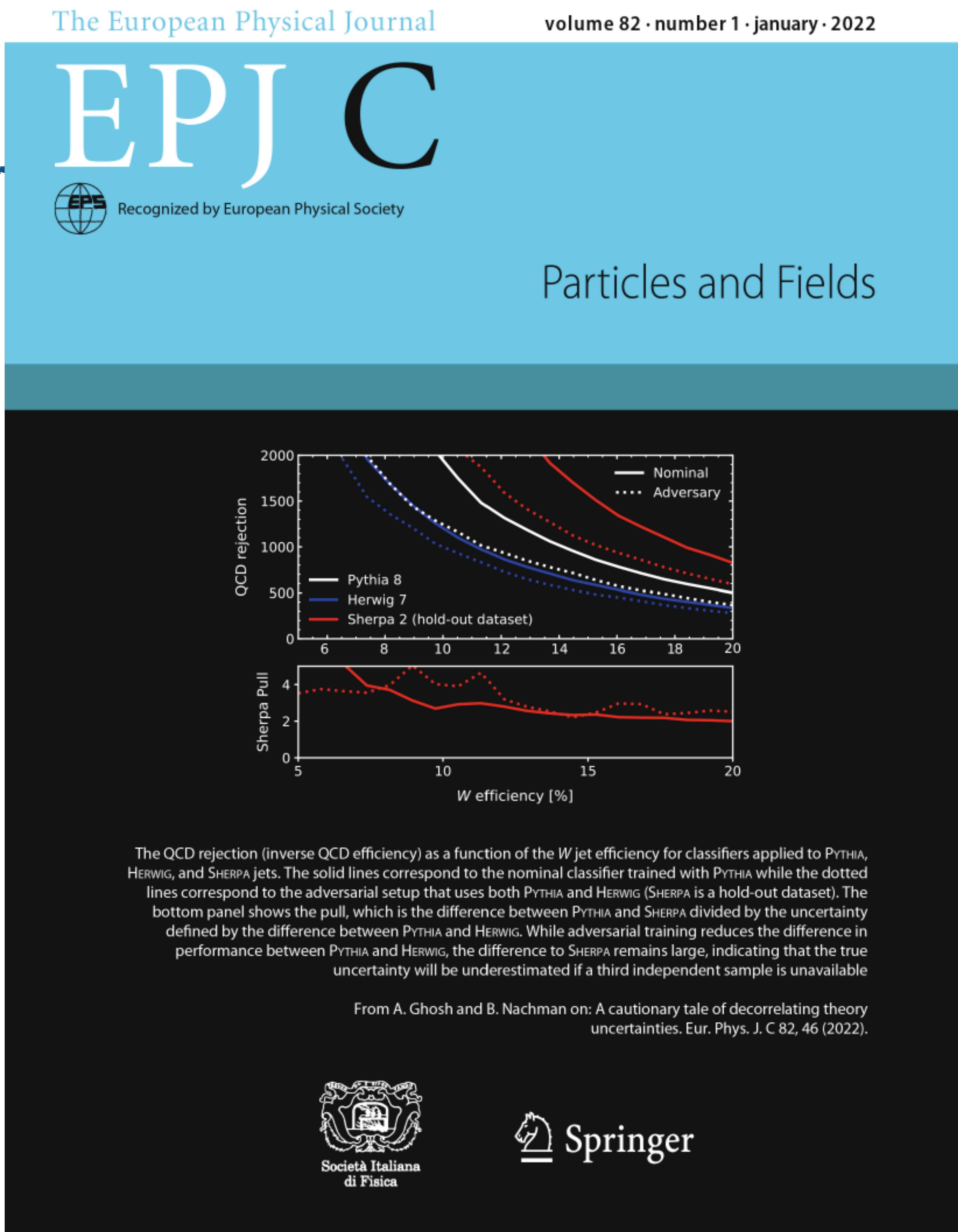
Implications beyond HEP…?

Fact that of all my ML-for-physics work, this is the one that made it to a journal cover says something about how conservative larger community still is…?

Implications for decorrelating biases in gender / race / age …? What are the unintended consequences?

Ghosh, A., Nachman, B. Eur. Phys. J. C 82, 46 (2022)

EPJC highlight article written about our work



The European Physical Journal volume 82 · number 1 · january · 2022

EPJ C

Recognized by European Physical Society

Particles and Fields

The QCD rejection (inverse QCD efficiency) as a function of the *W* jet efficiency for classifiers applied to Pythia, Herwig, and Sherpa jets. The solid lines correspond to the nominal classifier trained with Pythia while the dotted lines correspond to the adversarial setup that uses both Pythia and Herwig (Sherpa is a hold-out dataset). The bottom panel shows the pull, which is the difference between Pythia and Sherpa divided by the uncertainty defined by the difference between Pythia and Herwig. While adversarial training reduces the difference in performance between Pythia and Herwig, the difference to Sherpa remains large, indicating that the true uncertainty will be underestimated if a third independent sample is unavailable

From A. Ghosh and B. Nachman on: A cautionary tale of decorrelating theory uncertainties. Eur. Phys. J. C 82, 46 (2022).

Società Italiana di Fisica

Springer

Outlook for uncertainties and ML

- It is tempting to apply ML decorrelation to reduce uncertainties
- When source of uncertainty well understood: Uncertainty-Aware Networks do a better job (but decorrelation may be simpler)
- When source of uncertainty not well understood: caution must be taken before applying any domain adaptation techniques

Outlook for uncertainties and ML

- It is tempting to apply ML decorrelation to reduce uncertainties

- When source of uncertainty well understood: Uncertainty-Aware Networks do a better job (but decorrelation may be simpler)

- 🙈 When source of uncertainty not well understood: caution must be taken before applying any domain adaptation techniques

Systematic Uncertainties

Imagine a metal ruler calibrated at room temperature but used at near 0 K

Experimental physics example: Calibration of some energy scale

Image: https://www.shutterstock.com/image-photo/measuring-stick-snow-ruler-shows-amount-1896983614

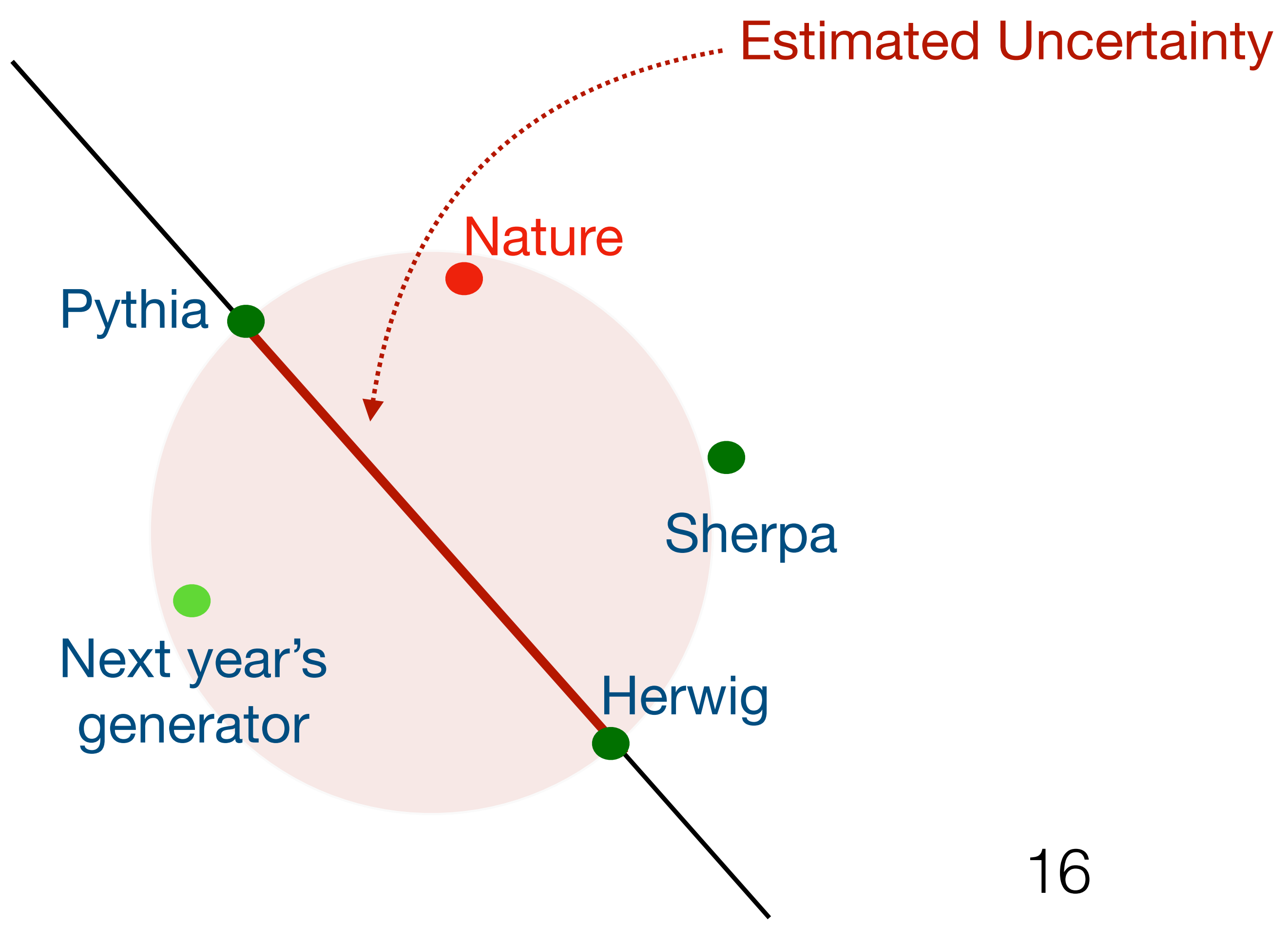
Systematic Uncertainties

Imagine a metal ruler calibrated at room temperature but used at near 0 K

Experimental physics example: Calibration of some energy scale

Image: https://www.shutterstock.com/image-photo/measuring-stick-snow-ruler-shows-amount-1896983614

Theory example: Fragmentation modelling not yet precise ➞ every generator models it a bit differently and simulates something slightly different



Systematic Uncertainties

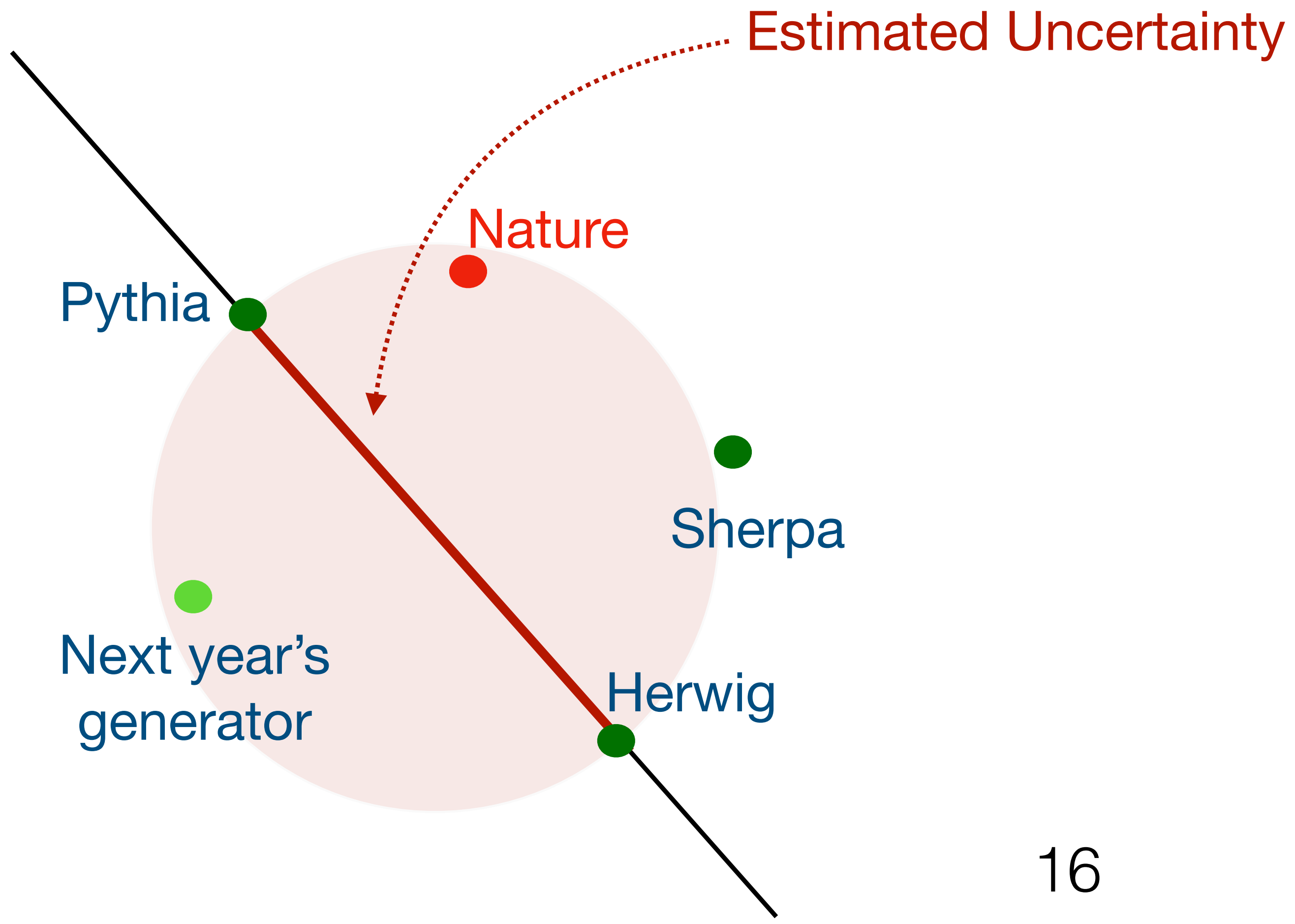
Imagine a metal ruler calibrated at room temperature but used at near 0 K

Experimental physics example: Calibration of some energy scale

Image: https://www.shutterstock.com/image-photo/measuring-stick-snow-ruler-shows-amount-1896983614

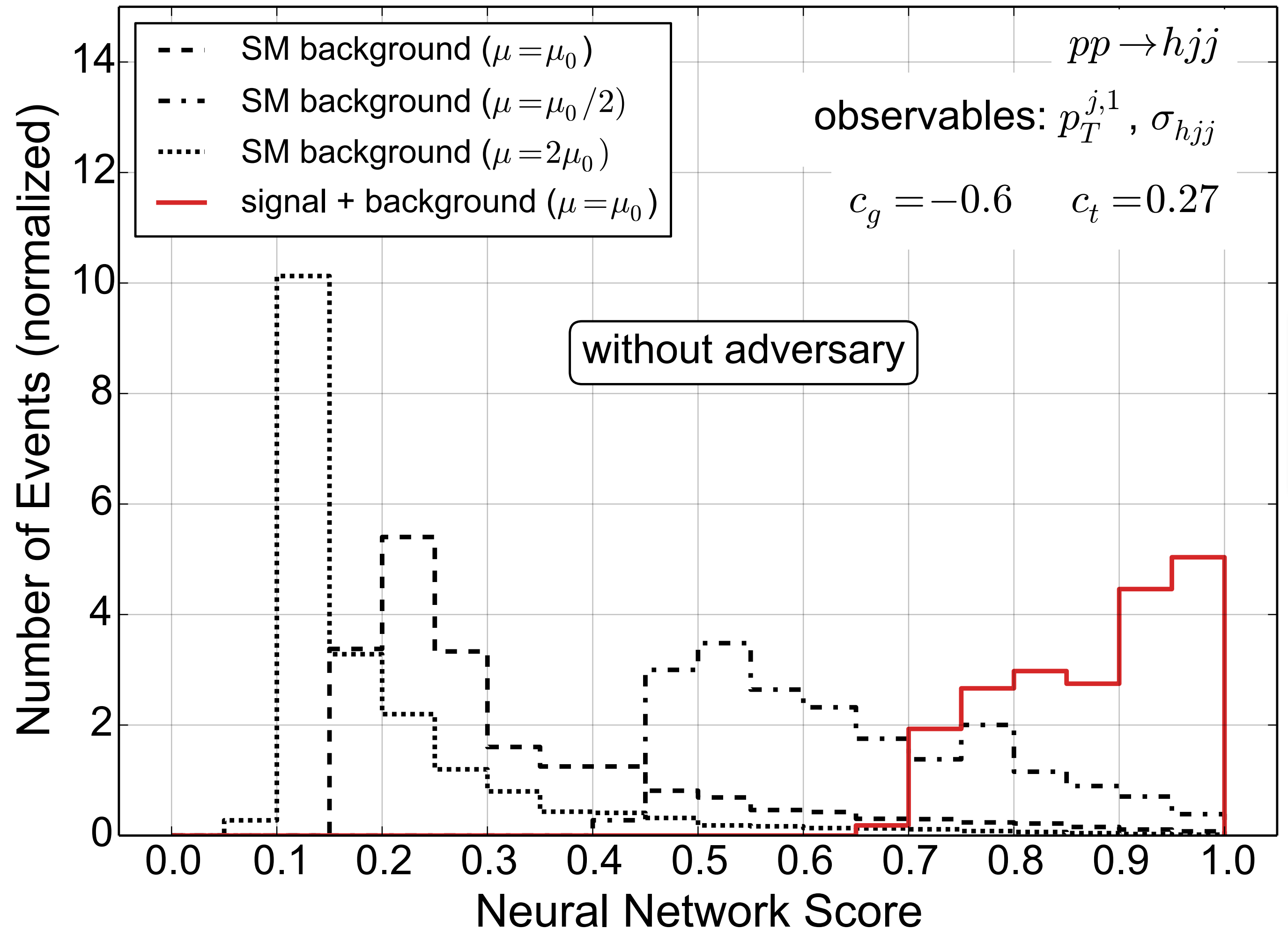
- Different from model uncertainties → A more powerful ML model won't help reduce these uncertainties
- Different from data uncertainties → More training data won't help reduce these uncertainties

Theory example: Fragmentation modelling not yet precise → every generator models it a bit differently and simulates something slightly different

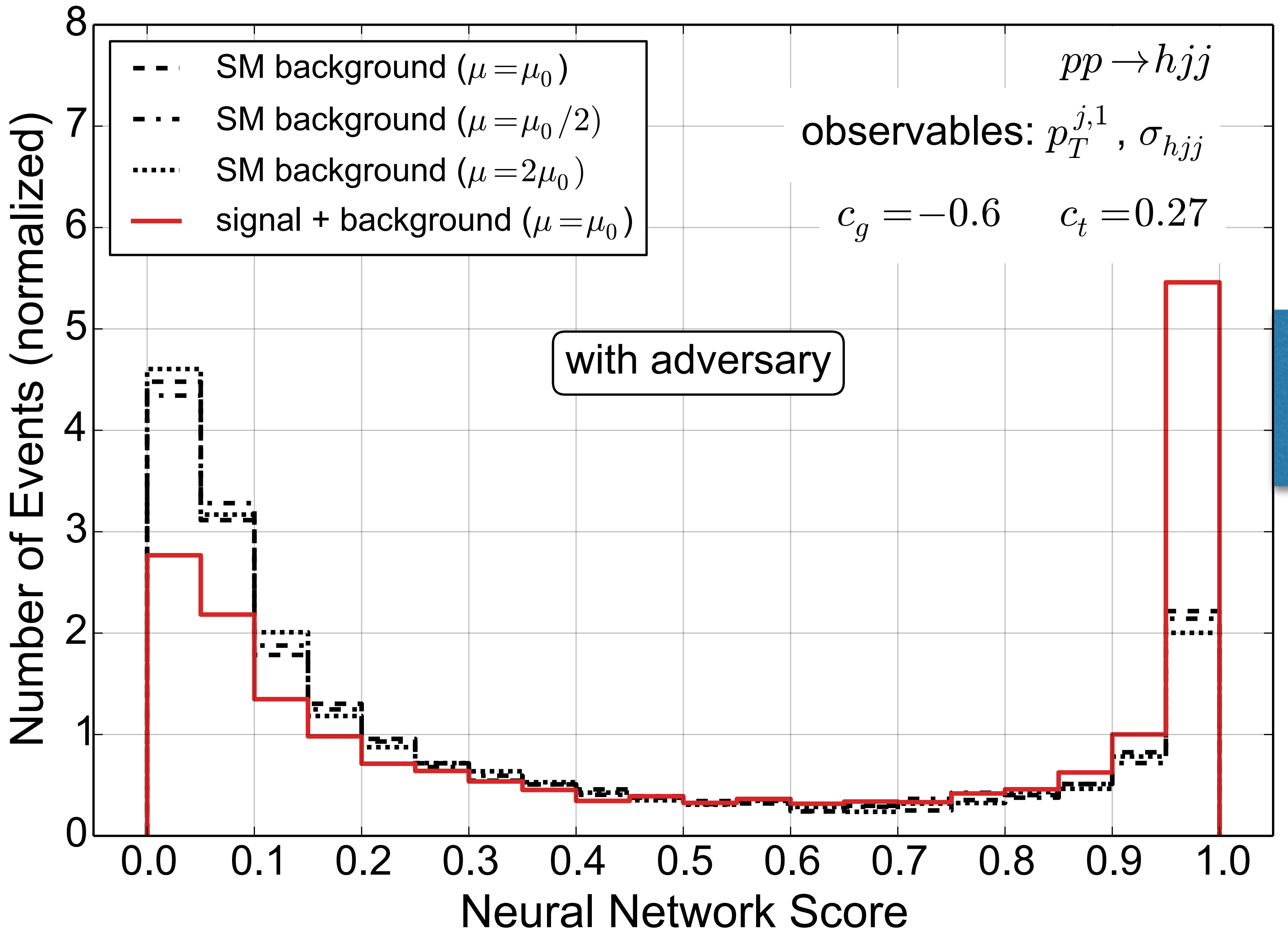


Prior work for theory uncertainties

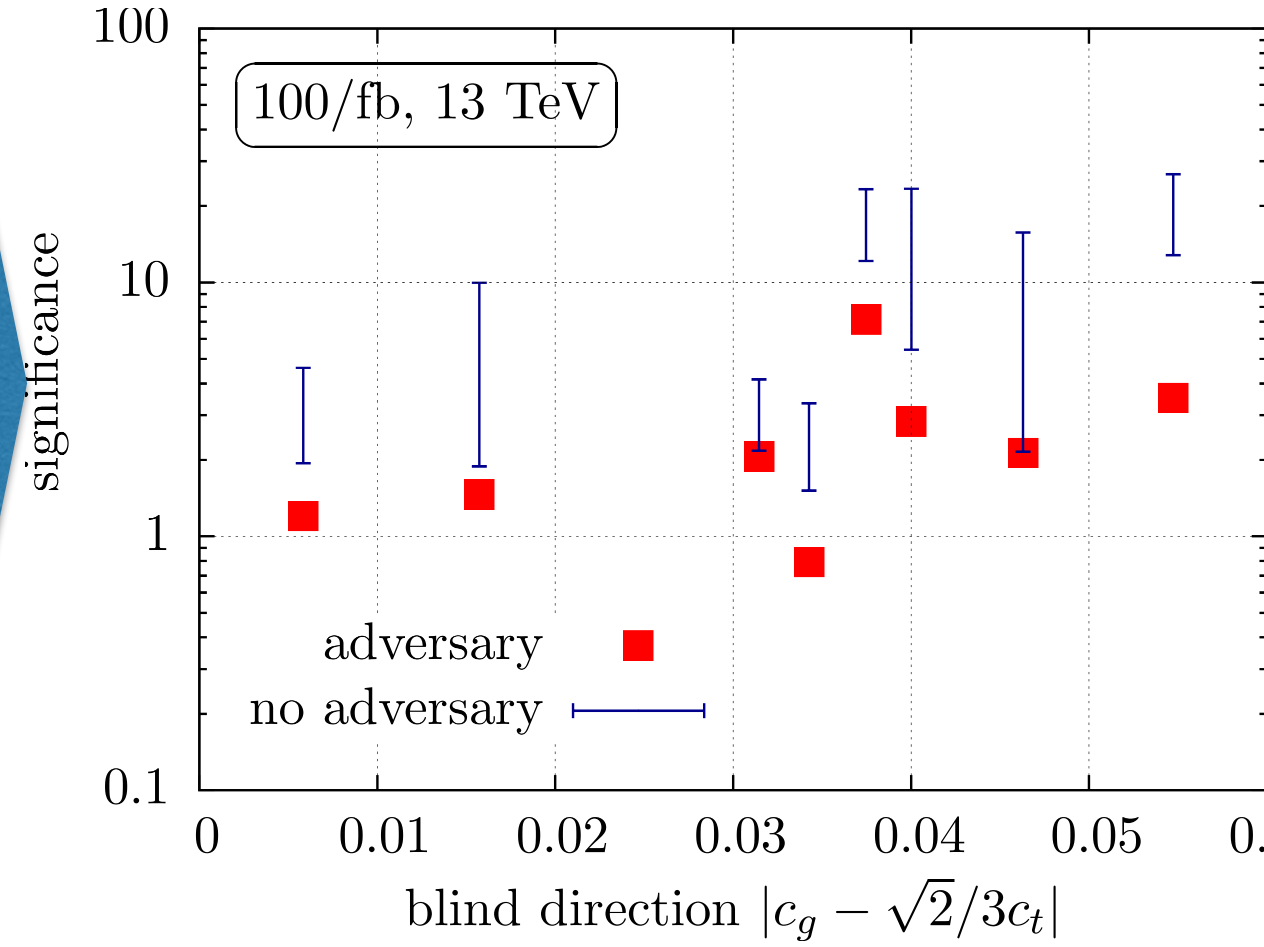
arXiv:1807.08763



Adversarial Training

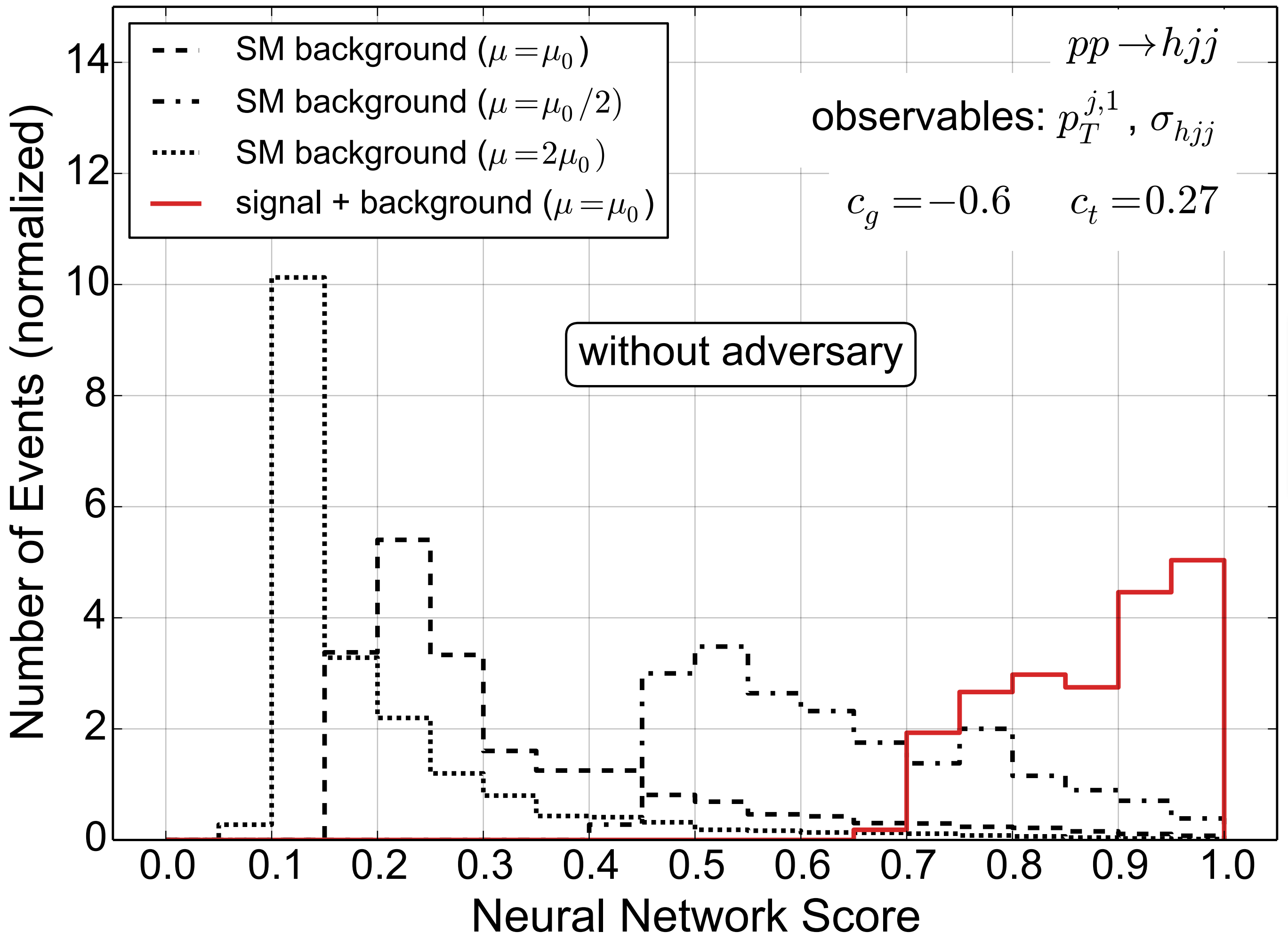


“Smaller errors”

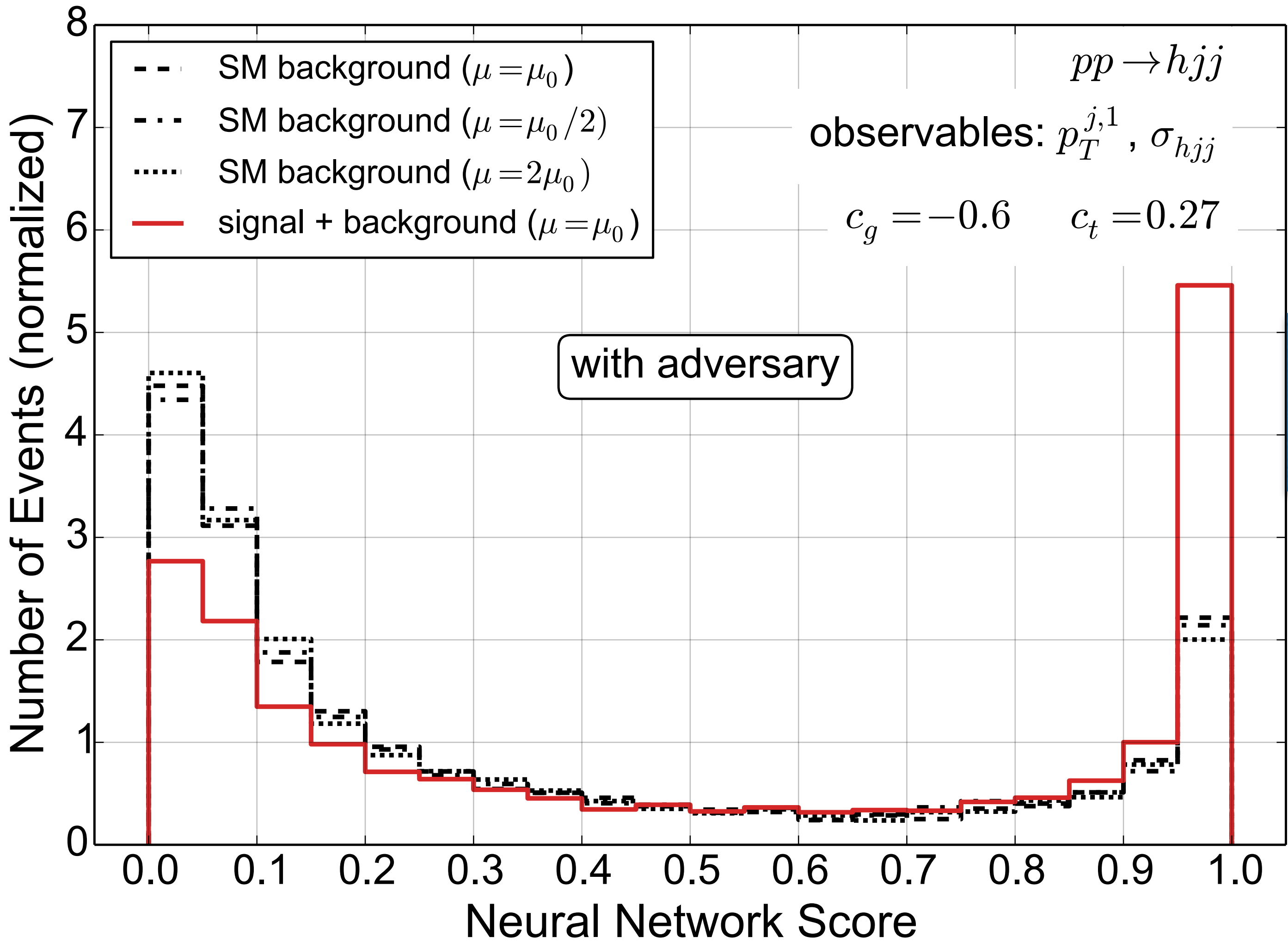


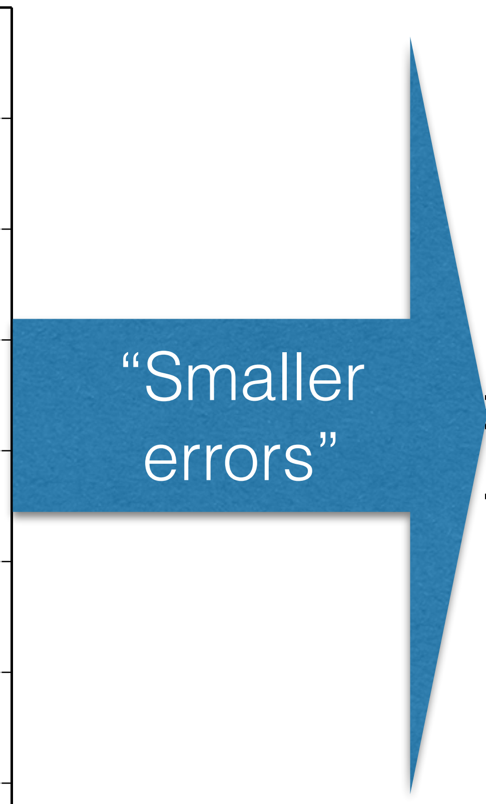
Prior work for theory uncertainties

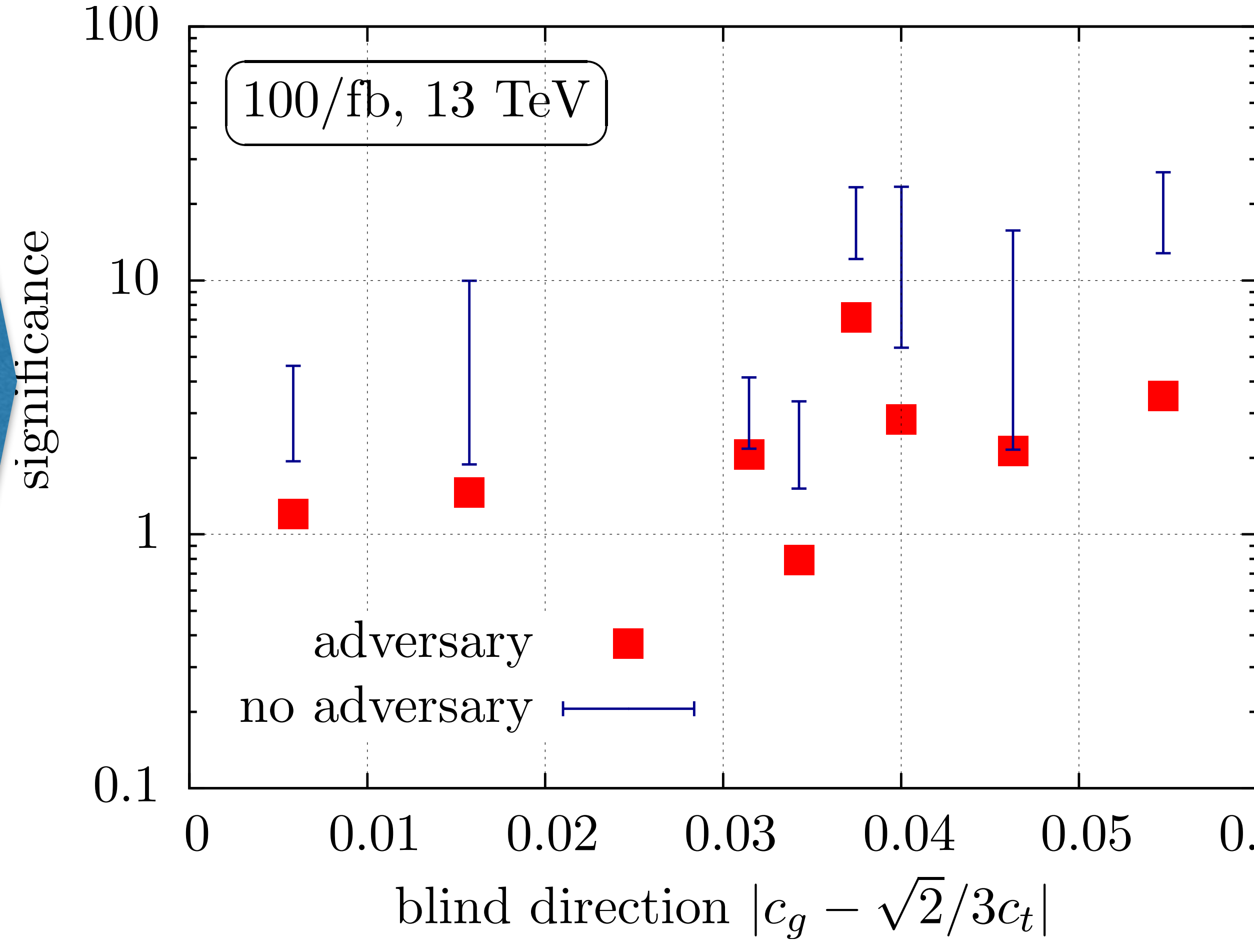
arXiv:1807.08763

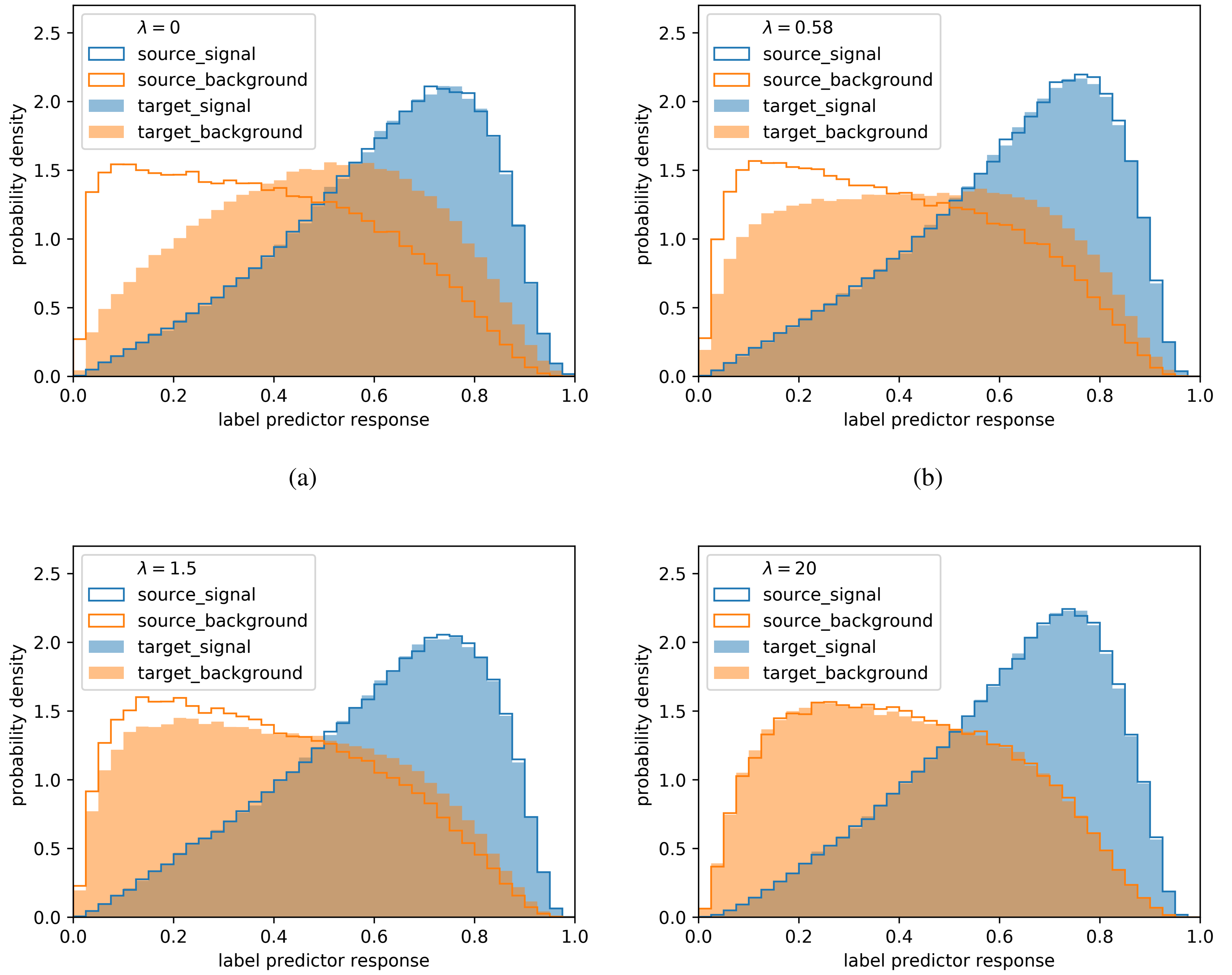


Adversarial Training




“Smaller errors”





(a)

(b)

Suggestion is to tune the extent of decorrelation vs separation power based on application case

arXiv:2005.00568

Case Study 2: Continuous uncertainty (Higher-order corrections)

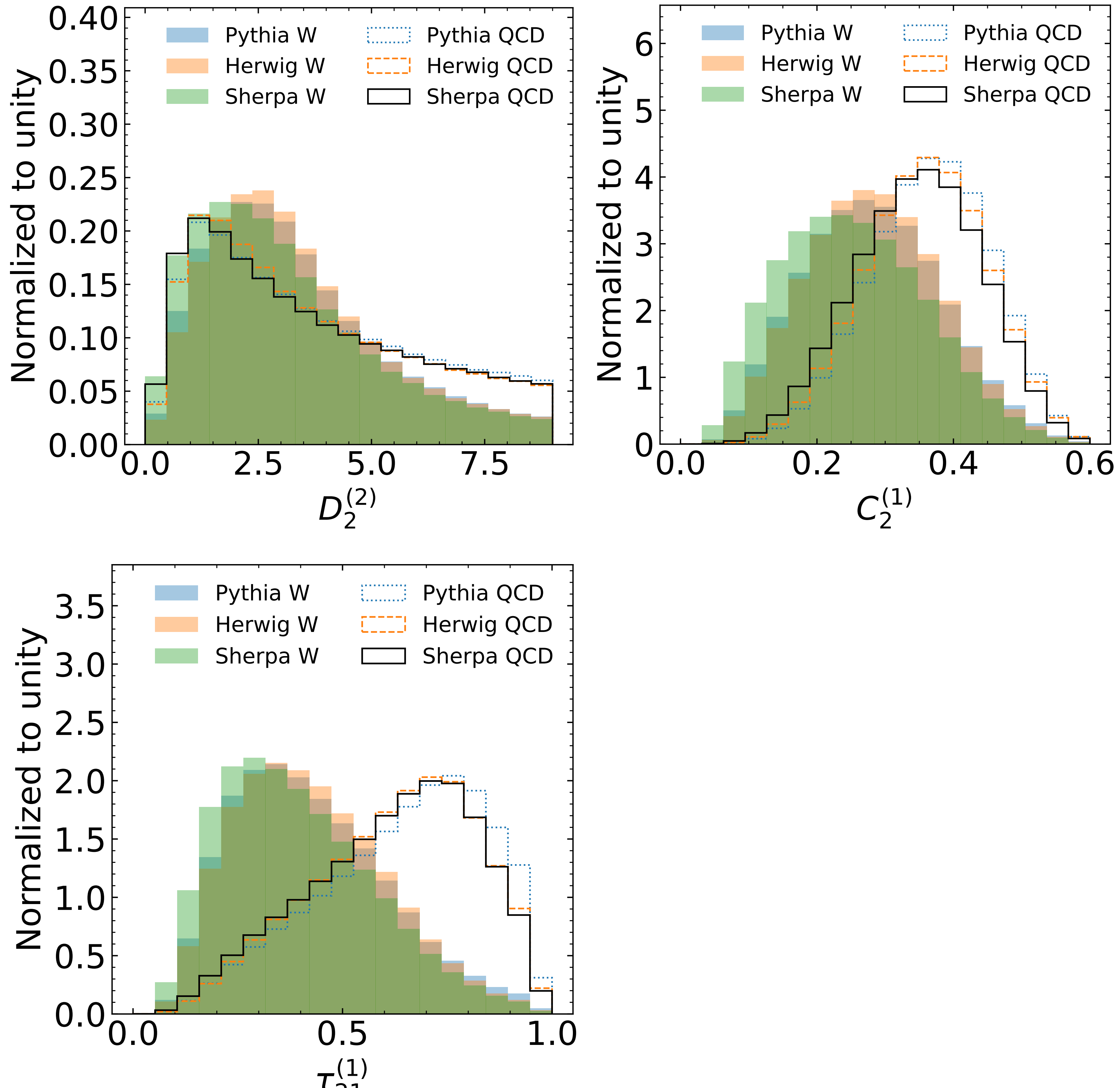
- Uncertainty from truncating order of perturbative calculation (QFT) is estimated by varying scales (renormalization scale, factorisation scale) and looking at the change in result
  - For example NLO + scale variations to estimate uncertainty for NNLO
  - Scale usually varied between 1/2 to 2 to estimate uncertainty - no deep physics reason for it
- We focus on factorisation scale - dictates separation bw long and short distance physics

Case Study 1: Two-point uncertainty (fragmentation modelling)

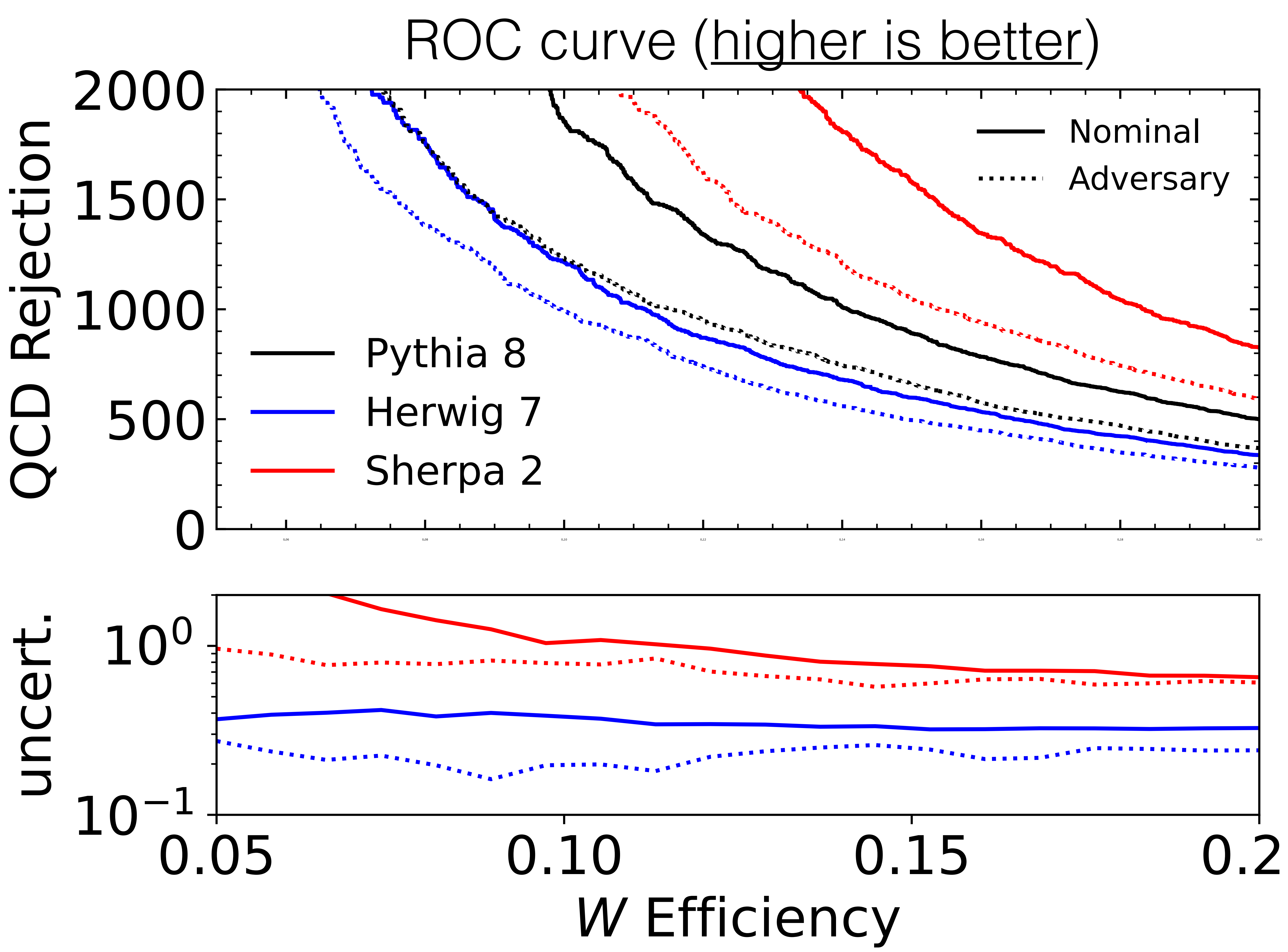
Goal: W jets vs QCD jets

Decorrelation: Reduce difference in performance on Herwig vs Pythia

Cross-check: Test uncertainty estimate from {Herwig vs Pythia} using Sherpa



Case Study 1: Two-point uncertainty - Result



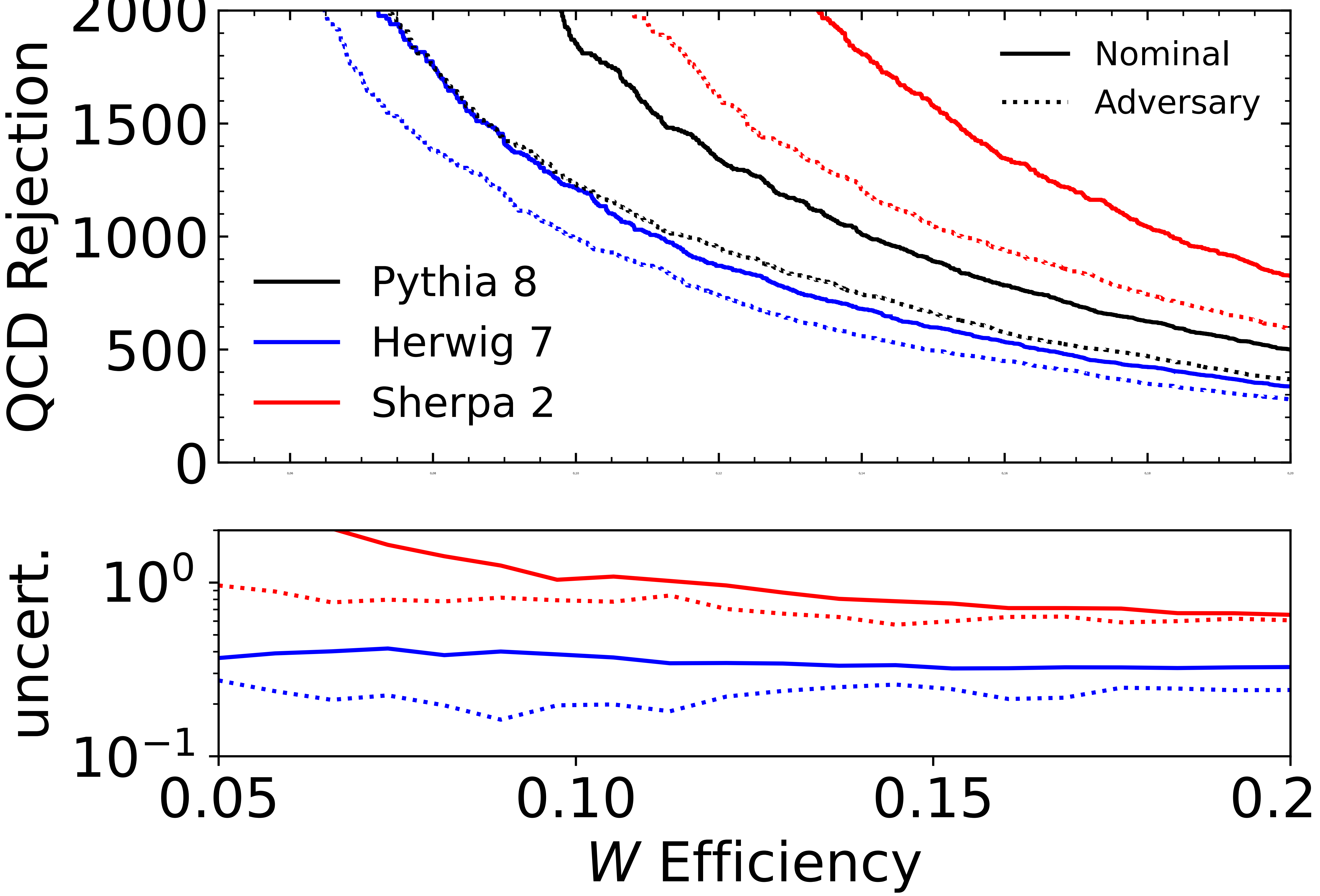
Case Study 1: Two-point uncertainty - Result

Adversary successfully sacrifices separation power in order to reduce difference in performance between Herwig and Pythia

Cross-check with Sherpa reveals uncertainty severely underestimated by usual Herwig vs Pythia comparison

In an typical LHC analysis, a cross-check with third generator rarely performed, similar to prior work suggesting decorrelation for theory uncertainties

ROC curve (higher is better)

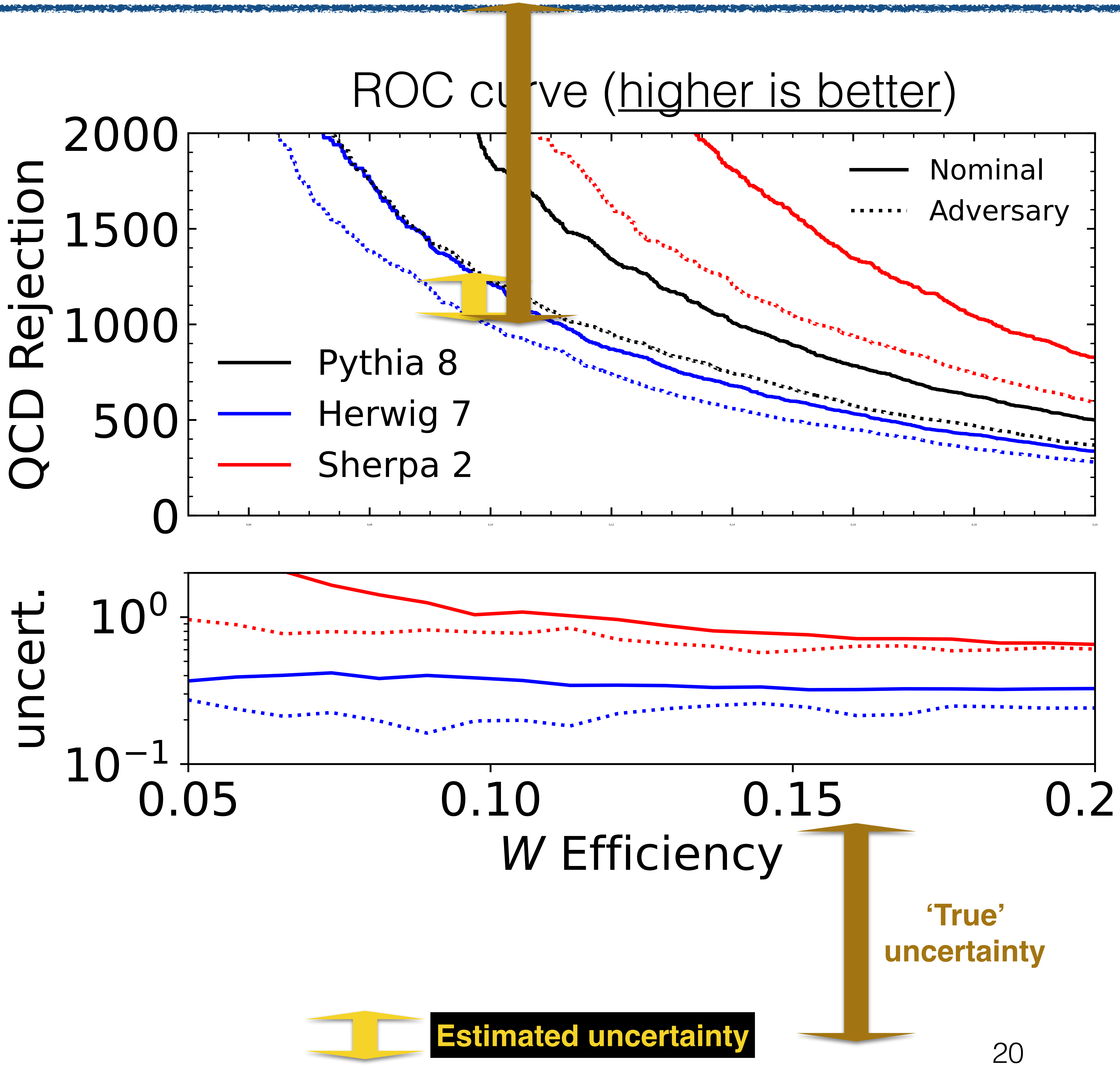


Case Study 1: Two-point uncertainty - Result

Adversary successfully sacrifices separation power in order to reduce difference in performance between Herwig and Pythia

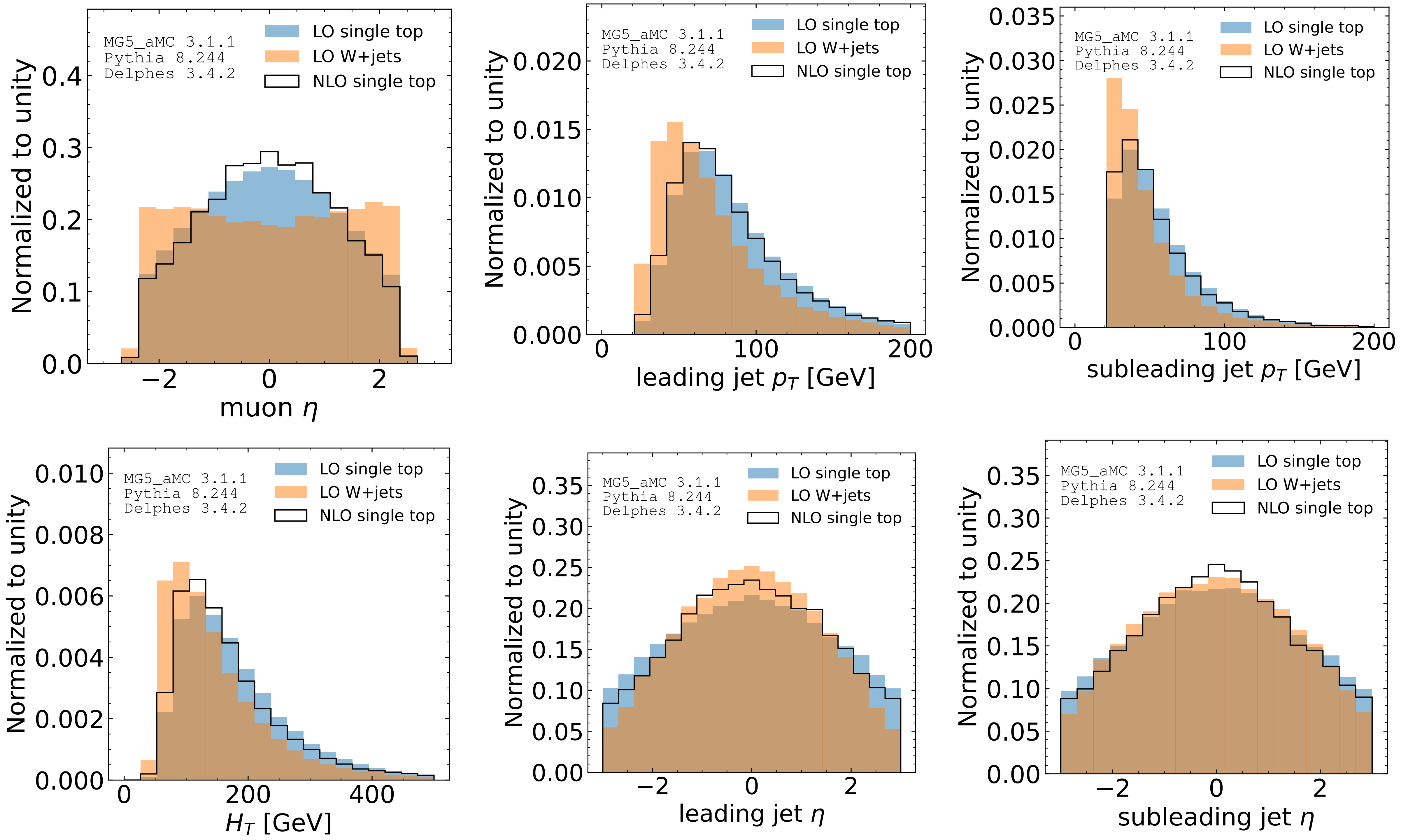
Cross-check with Sherpa reveals uncertainty severely underestimated by usual Herwig vs Pythia comparison

In an typical LHC analysis, a cross-check with third generator rarely performed, similar to prior work suggesting decorrelation for theory uncertainties

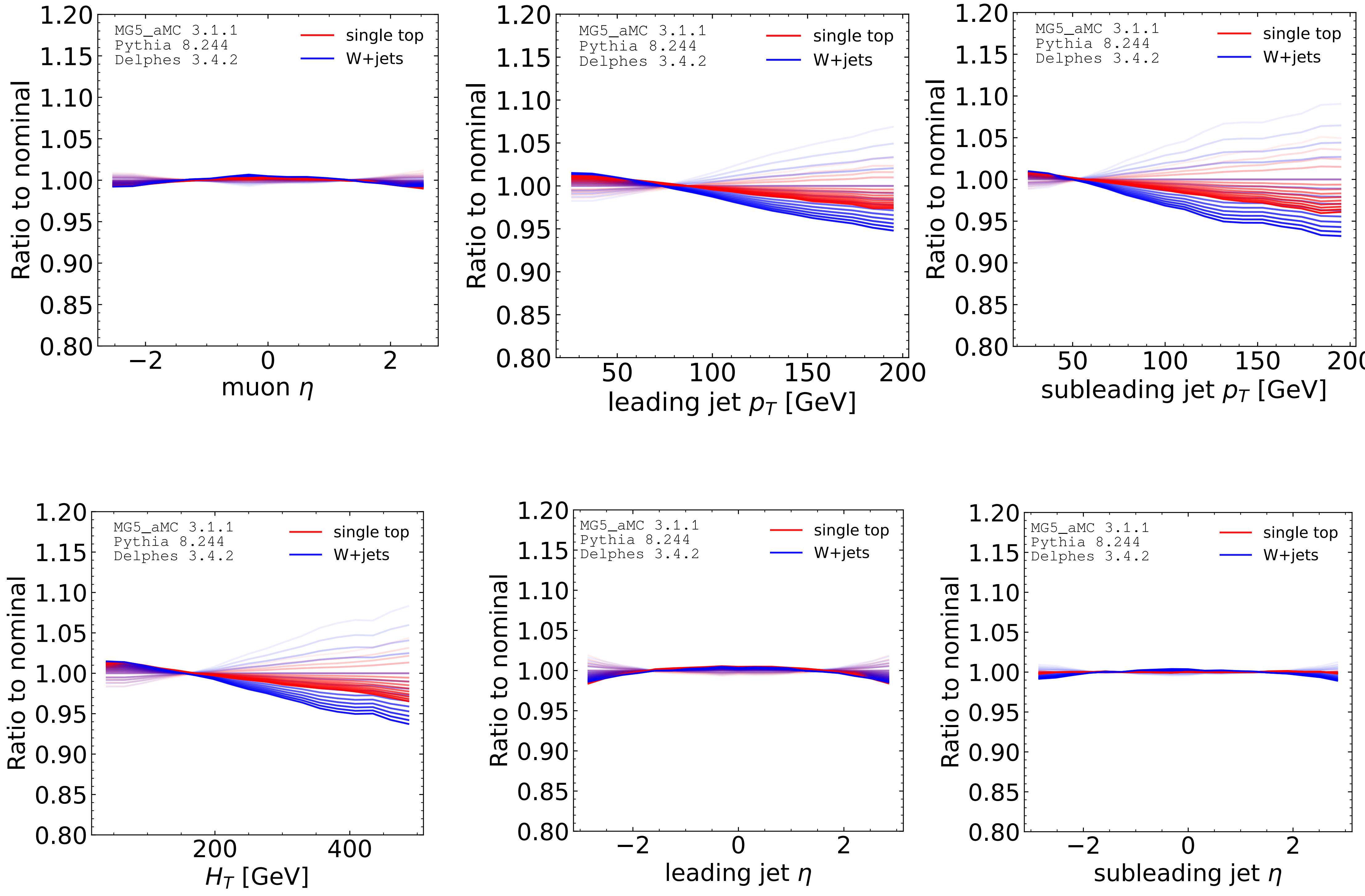


Case Study 2: Continuous uncertainty - Problem Setup

Goal: Single top vs W+Jets
Decorrelation: Reduce difference in performance on scale variations at LO
Cross-check: Test uncertainty estimate from {scale variations at LO} using NLO



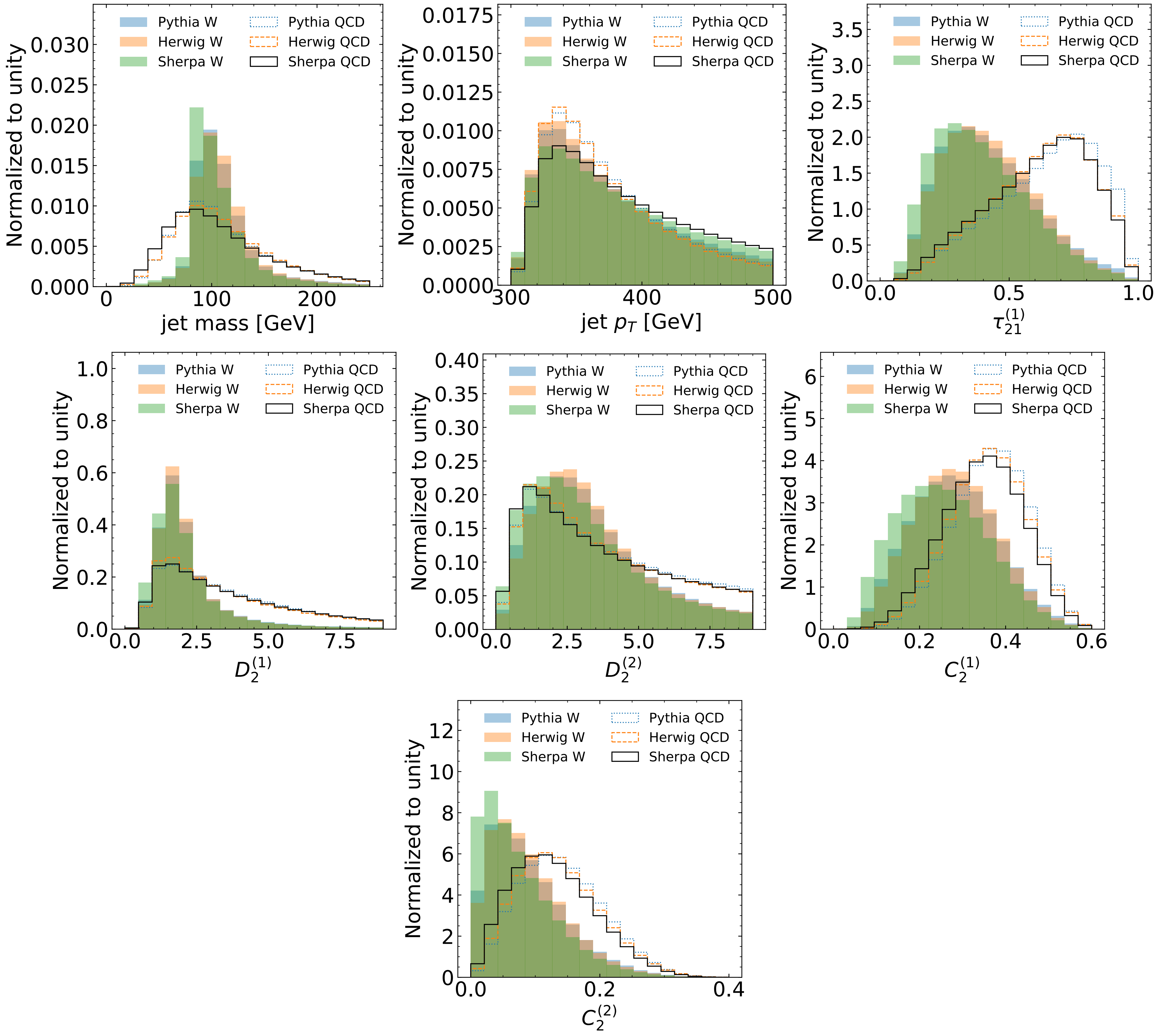
NLO vs LO



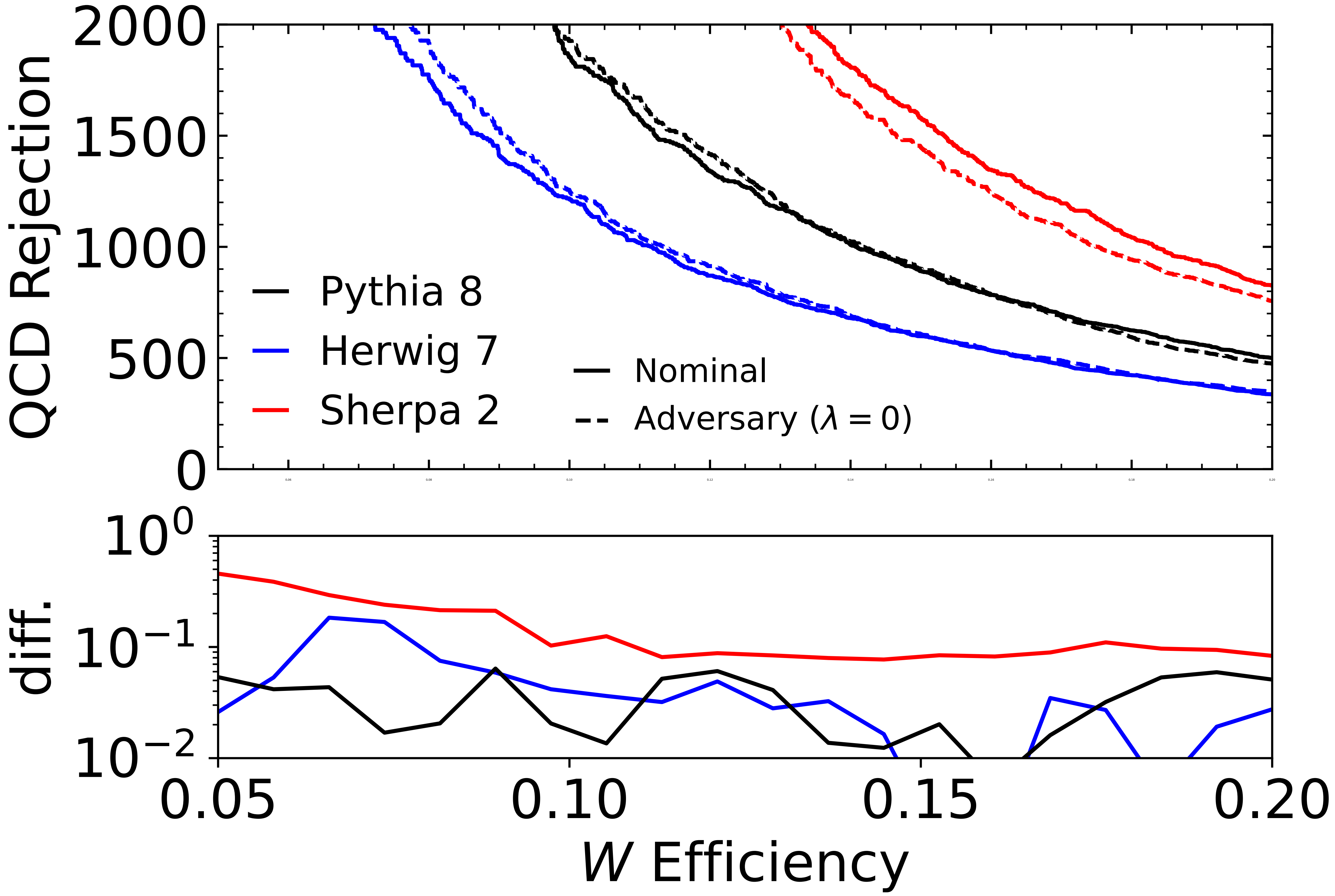
Factorisation scale variations going from 1/2 to 2

Appendix - Case Study 1: Two-point Uncertainty

All input observables

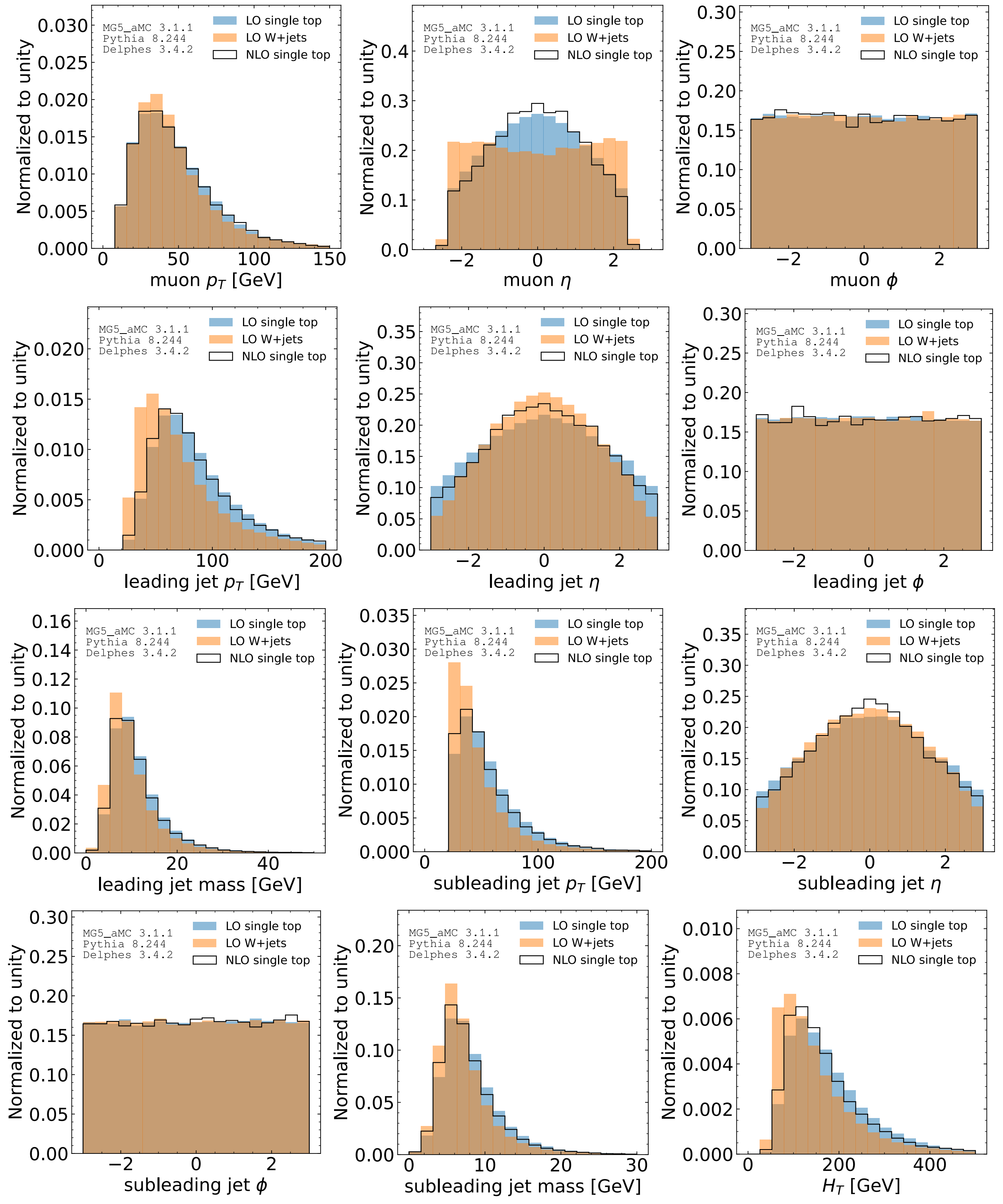


Decorrelation parameter $\lambda = 0$ (Effectively data augmentation)

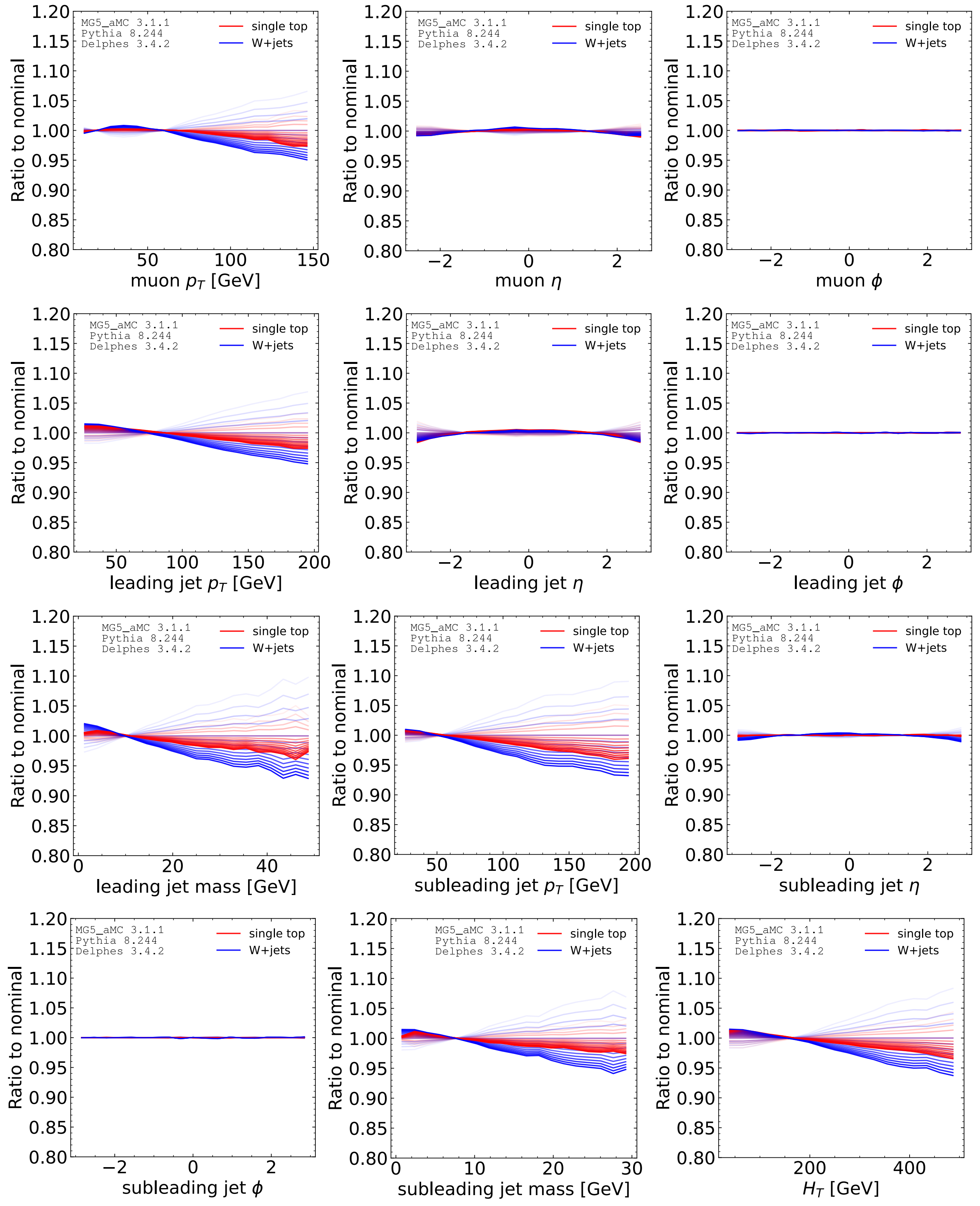


Appendix - Case Study 2: Continuous Uncertainty

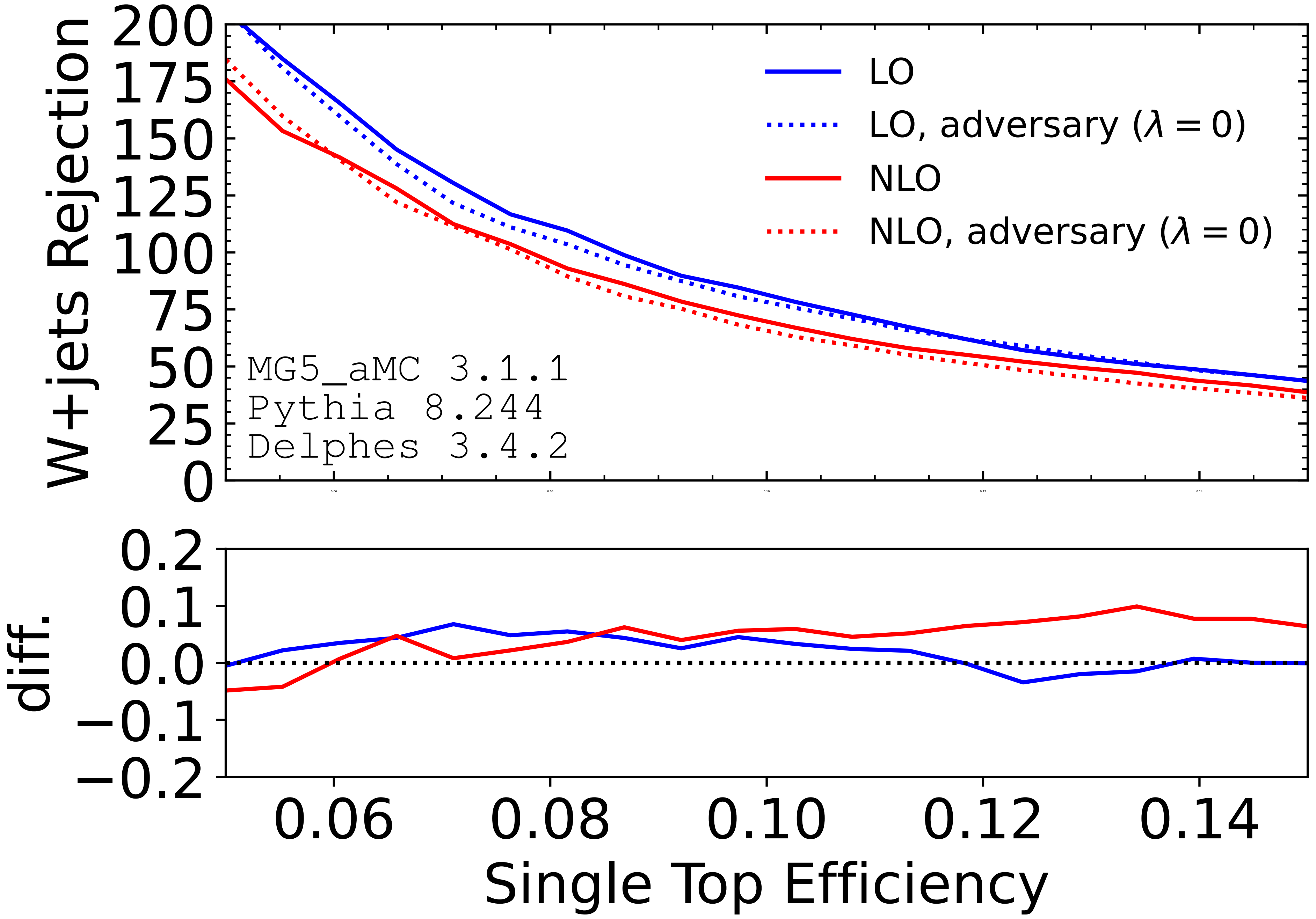
All input observables



Scale variations at LO



Decorrelation parameter $\lambda = 0$ (Effectively data augmentation)

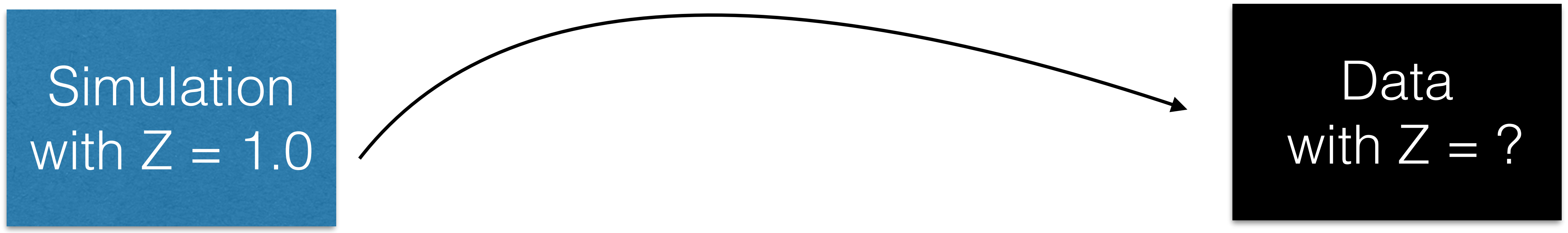


Nominal Classifier and Data Augmentation

- Baseline solution has been to train a classifier on nominal data (Z=1) and just account for uncertainties in measurement – which may be large. Full profile likelihood or shift Z and look at impact.

Nominal Classifier and Data Augmentation

- Baseline solution has been to train a classifier on nominal data (Z=1) and just account for uncertainties in measurement – which may be large. Full profile likelihood or shift Z and look at impact.

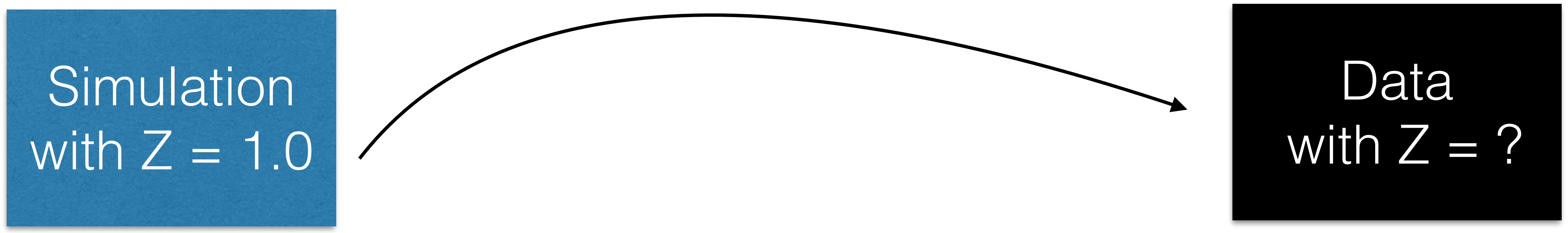


Simulation with Z = 1.0

Data with Z = ?

Nominal Classifier and Data Augmentation

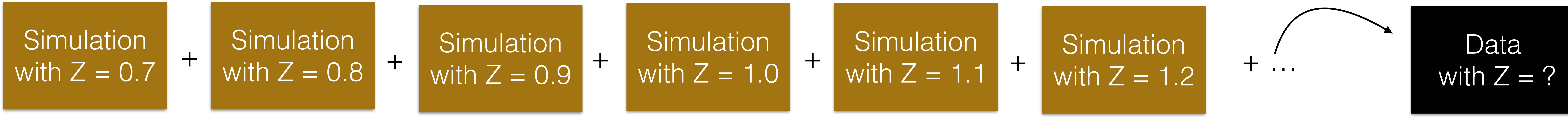
- Baseline solution has been to train a classifier on nominal data (Z=1) and just account for uncertainties in measurement – which may be large. Full profile likelihood or shift Z and look at impact.



Simulation with Z = 1.0

Data with Z = ?

- One way to attack the problem is “Data Augmentation”: Train classifier on simulated data generated with various values of Z, hope that it learns a robust decision function

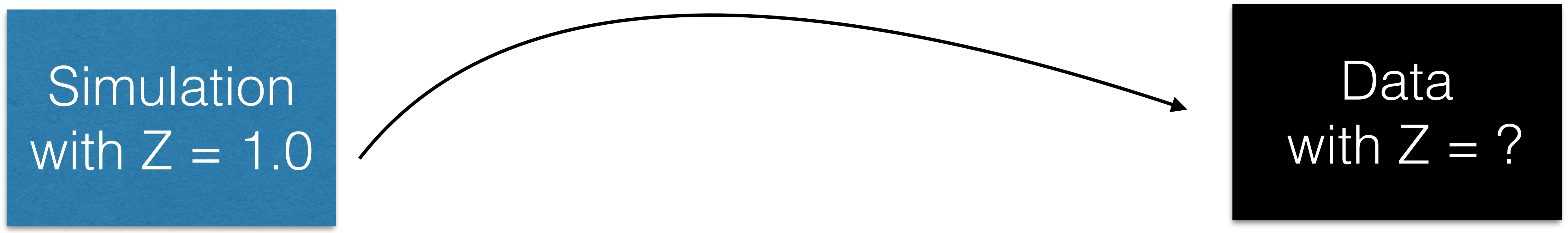


Simulation with Z = 0.7 + Simulation with Z = 0.8 + Simulation with Z = 0.9 + Simulation with Z = 1.0 + Simulation with Z = 1.1 + Simulation with Z = 1.2 + …

Data with Z = ?

Nominal Classifier and Data Augmentation

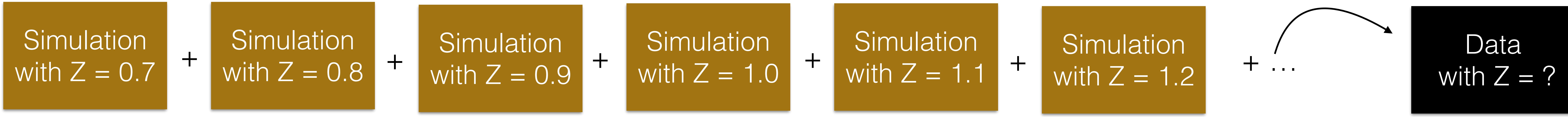
- Baseline solution has been to train a classifier on nominal data (Z=1) and just account for uncertainties in measurement – which may be large. Full profile likelihood or shift Z and look at impact.



Simulation with Z = 1.0

Data with Z = ?

- One way to attack the problem is "Data Augmentation": Train classifier on simulated data generated with various values of Z, hope that it learns a robust decision function



Simulation with Z = 0.7 + Simulation with Z = 0.8 + Simulation with Z = 0.9 + Simulation with Z = 1.0 + Simulation with Z = 1.1 + Simulation with Z = 1.2 + …

Data with Z = ?

The classifier will learn some general characteristics, but will not be "optimal" for any particular value of Z

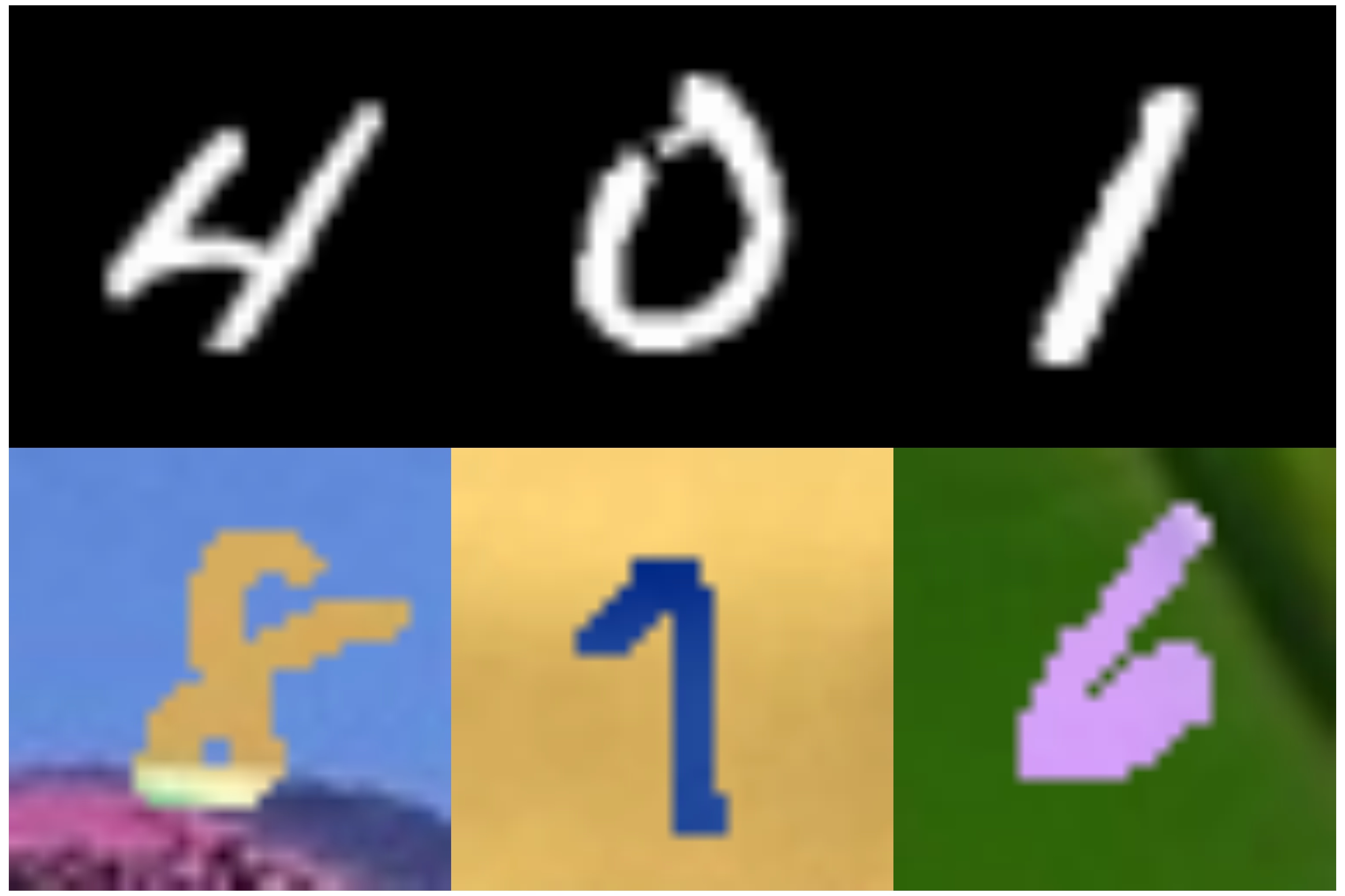
"Optimal": For us means classifier trained at the true value of Z

Connection to domain adaptation: Adversarial Decorrelation

MNIST

Source
(Simulation)

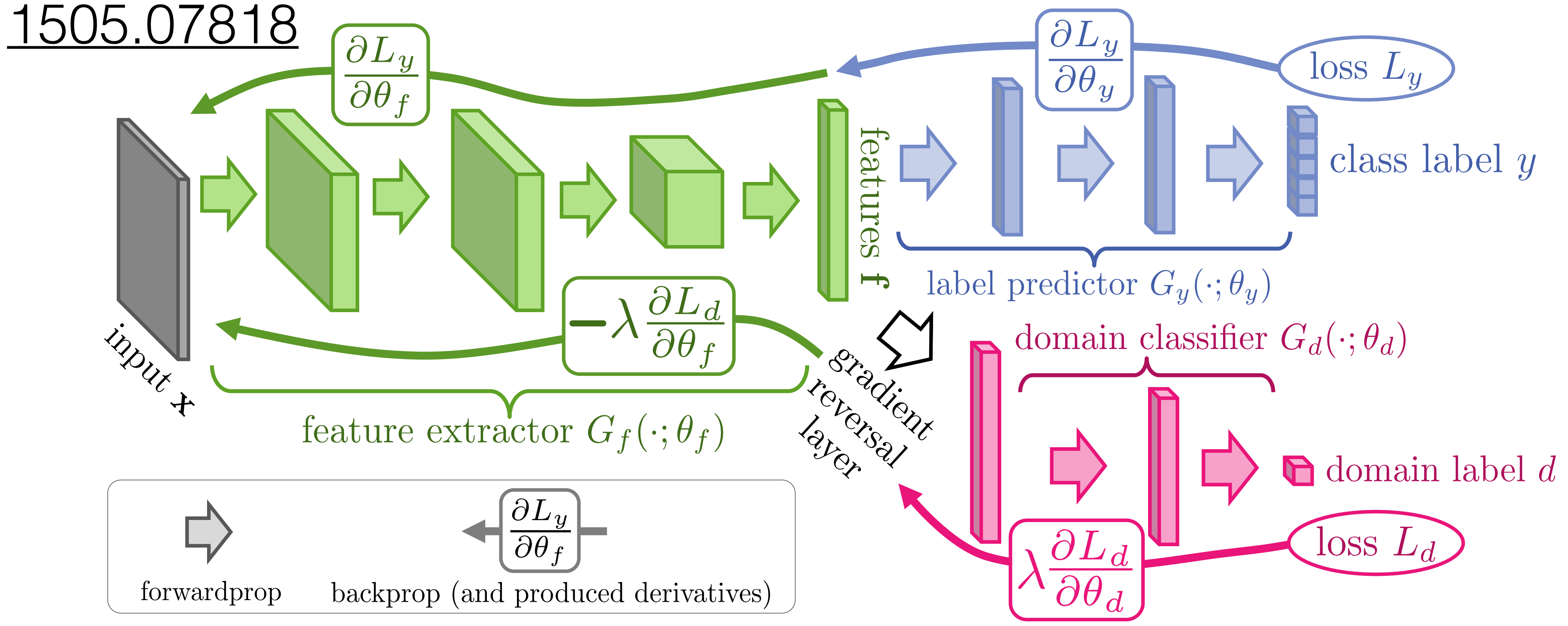
Target
(Data)



MNIST-M

Connection to domain adaptation: Adversarial Decorrelation

1505.07818

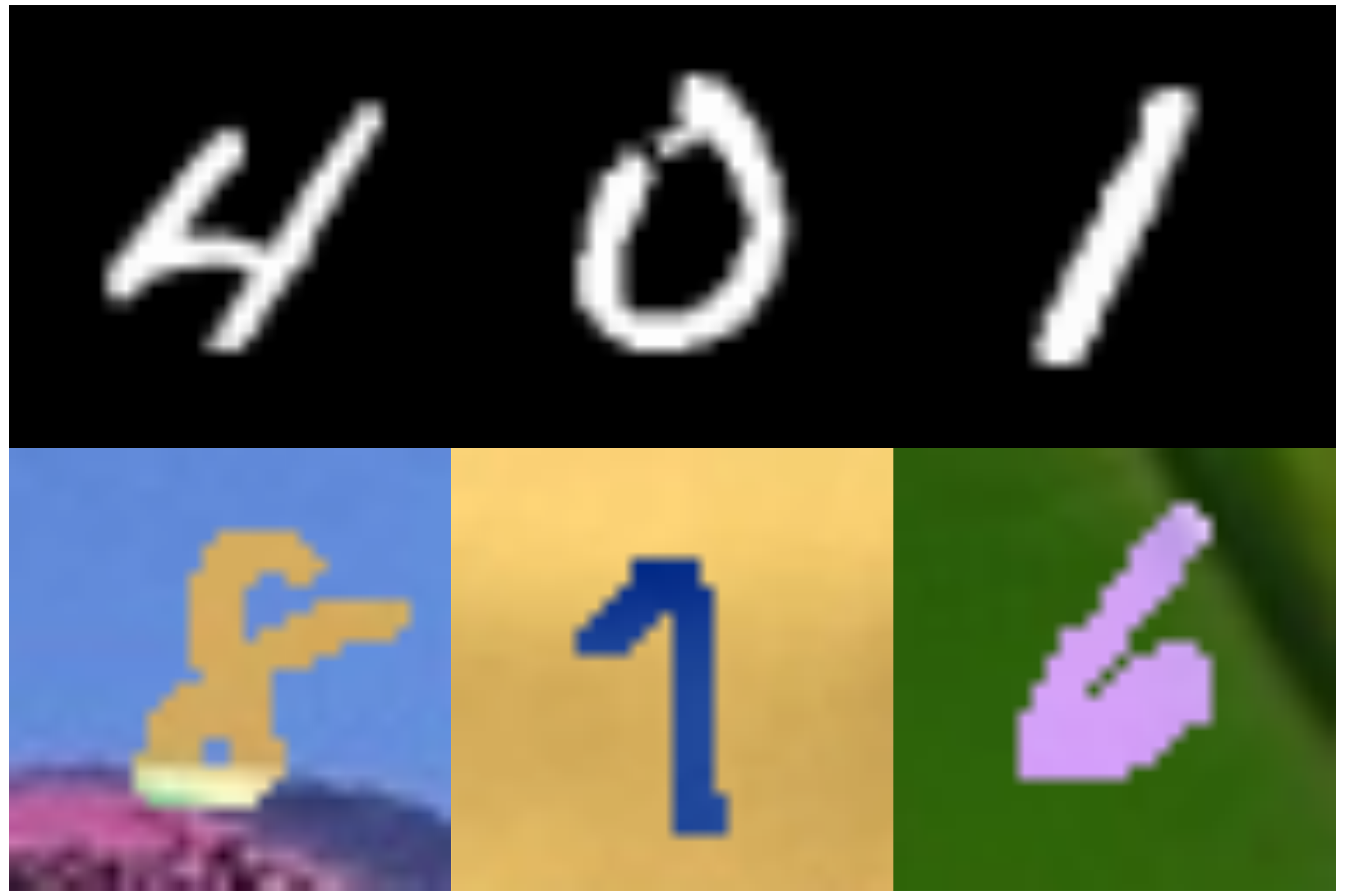


$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
input $\mathbf{x}$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
loss L_d
forwardprop
backprop (and produced derivatives)

SOURCE
(Simulation)

TARGET
(Data)

MNIST



MNIST-M

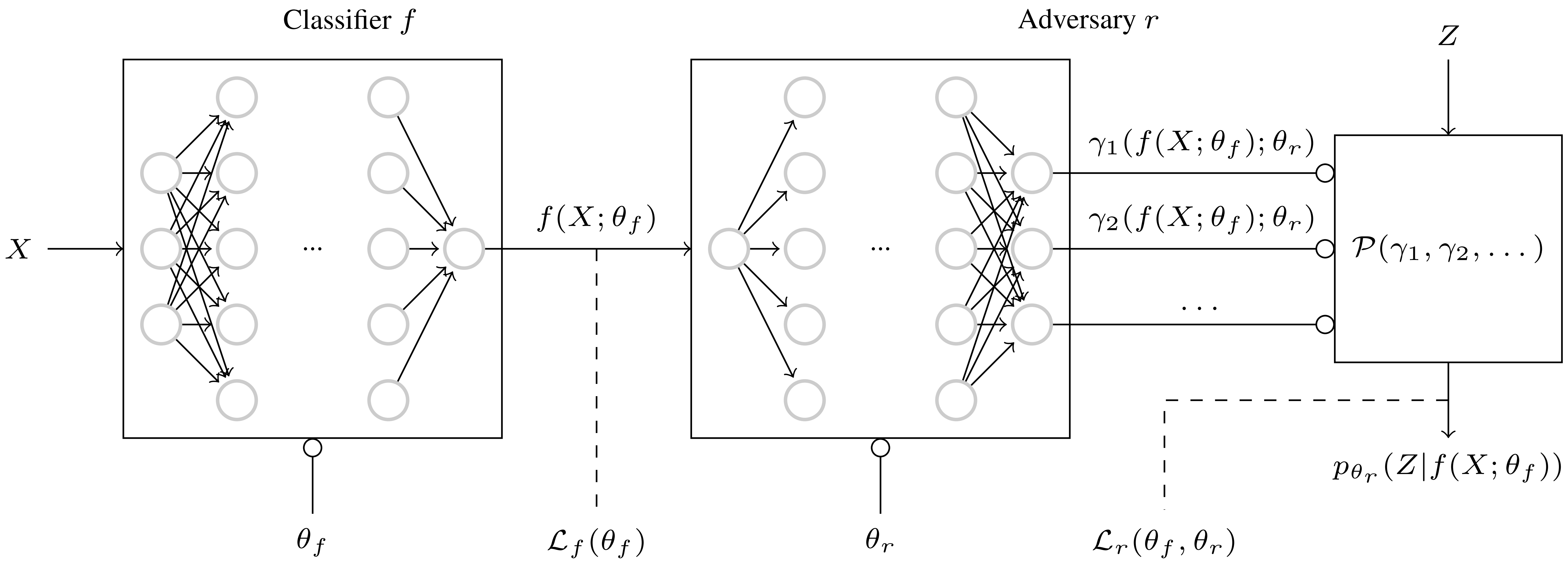
Learn only the relevant, transferable features from source, ignore background / colours

Uses a second network (adversary) to force invariance to background / colours

1611.01046
Similar ideas: 1905.10384, 1305.7248, 1907.11674, epjconf_chep2018_06024

Adversarial decorrelation for physics

Eg. Pivot Adversarial Training to make classifier output invariant to nuisance parameter ‘Z’ (source of uncertainty)



Classifier f
Adversary r
Z
X
$f(X;\theta_f)$
$\gamma_1(f(X;\theta_f);\theta_r)$
$\gamma_2(f(X;\theta_f);\theta_r)$
$\dots$
$\mathcal{P}(\gamma_1,\gamma_2,\dots)$
$p_{\theta_r}(Z|f(X;\theta_f))$
θ_f
$\mathcal{L}_f(\theta_f)$
θ_r
$\mathcal{L}_r(\theta_f,\theta_r)$

$$L_{Classifier} = L_{Classification} - \lambda \cdot L_{Adversary}$$

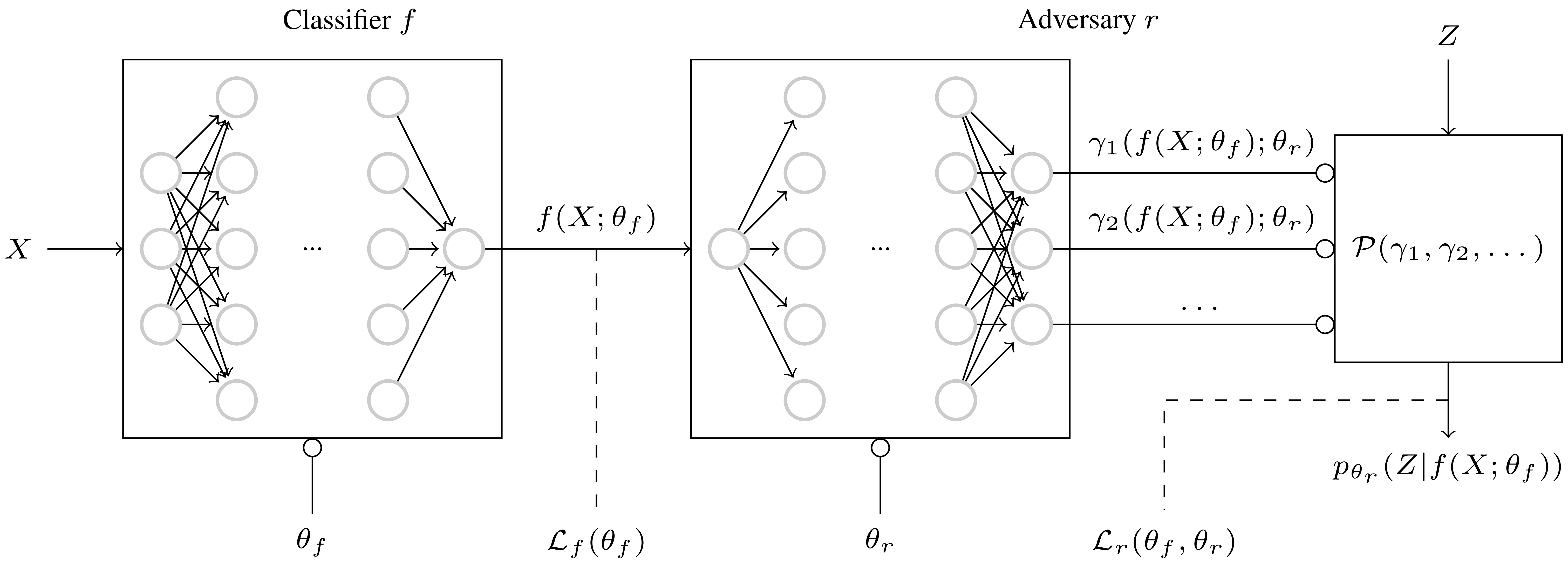
Similar to a GAN, two networks trained against each other:

- Adversary learns correlation of classifier output with Z
- Classifier tries to fool adversary + maximise separation power
    - λ parameter to weight the two objectives

1611.01046
Similar ideas: 1905.10384, 1305.7248, 1907.11674, epjconf_chep2018_06024

Adversarial decorrelation for physics

Eg. Pivot Adversarial Training to make classifier output invariant to nuisance parameter ‘Z’ (source of uncertainty)



Classifier f
Adversary r
Z
X
$f(X;\theta_f)$
$\gamma_1(f(X;\theta_f);\theta_r)$
$\gamma_2(f(X;\theta_f);\theta_r)$
$\dots$
$\mathcal{P}(\gamma_1,\gamma_2,\dots)$
$p_{\theta_r}(Z|f(X;\theta_f))$
θ_f
$\mathcal{L}_f(\theta_f)$
θ_r
$\mathcal{L}_r(\theta_f,\theta_r)$

To fool the adversary, classifier output should be decorrelated to Z

$$L_{Classifier} = L_{Classification} - \lambda \cdot L_{Adversary}$$

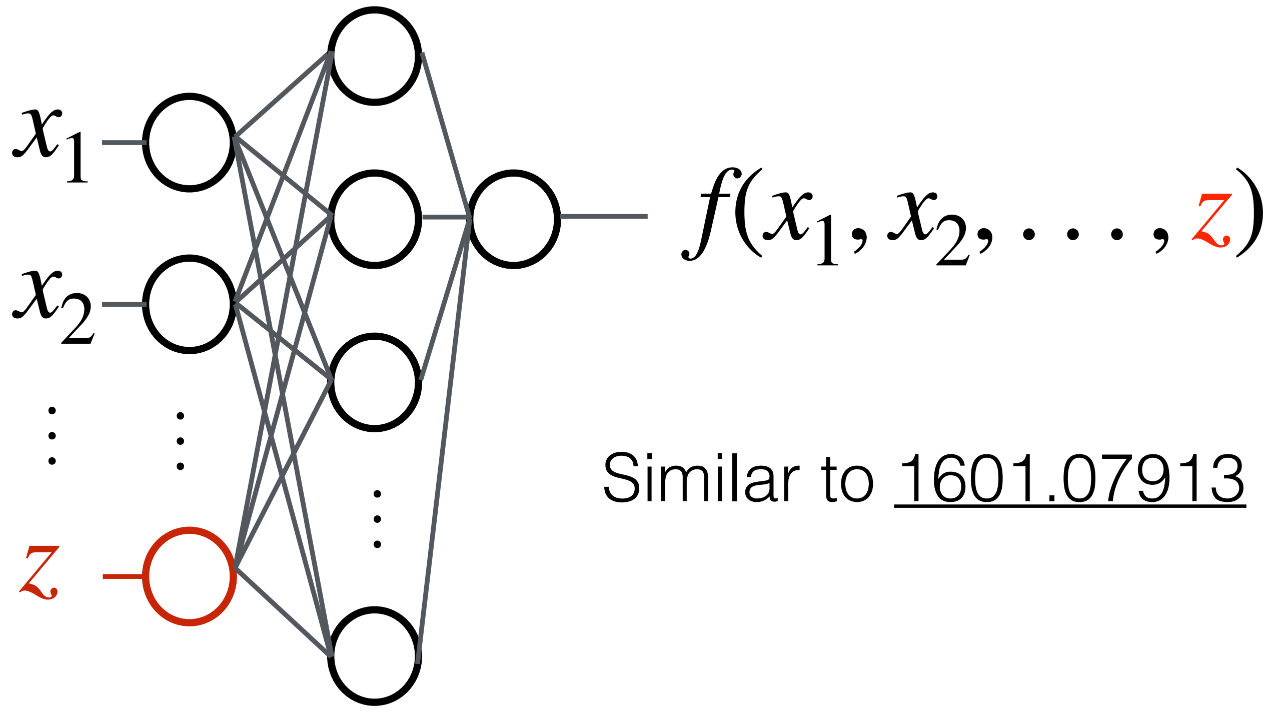
Similar to a GAN, two networks trained against each other:

- Adversary learns correlation of classifier output with Z
- Classifier tries to fool adversary + maximise separation power
    - λ parameter to weight the two objectives

We advocate for the opposite

PhysRevD.104.056026

- Fully parameterise the classifier on Z in a “systematic aware” way

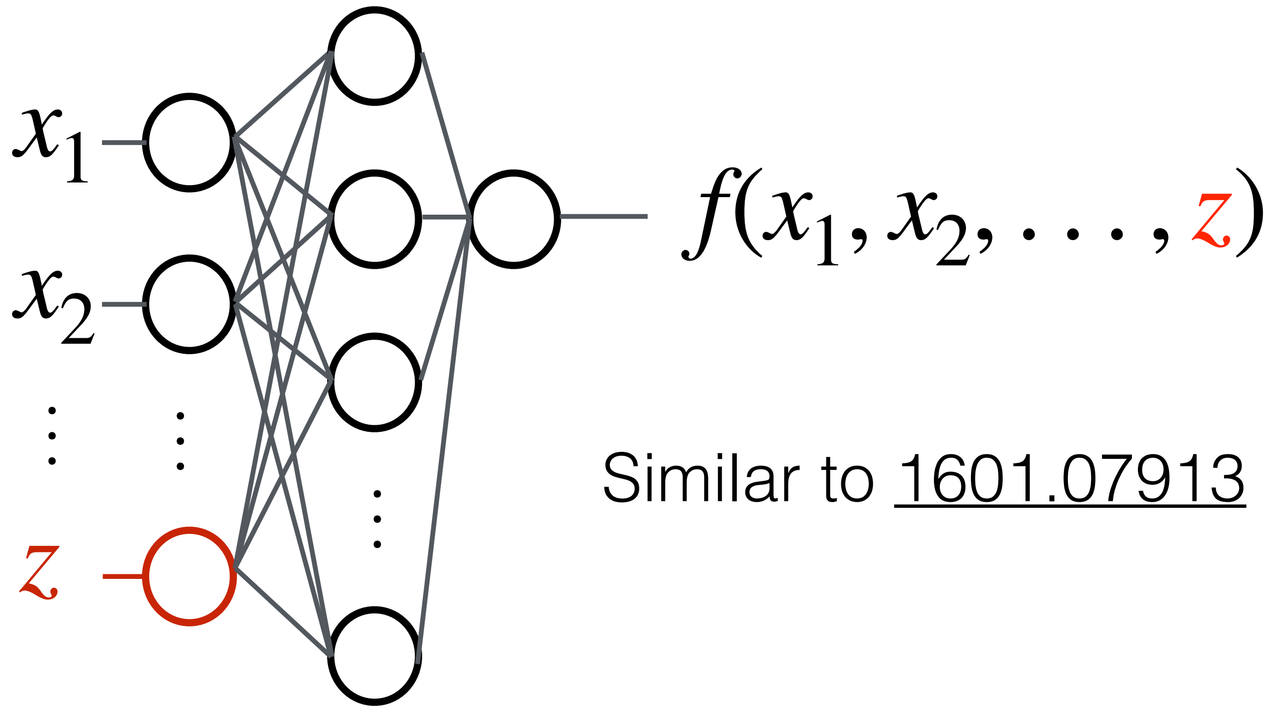


Similar to 1601.07913

We advocate for the opposite

PhysRevD.104.056026

- Fully parameterise the classifier on Z in a “systematic aware” way



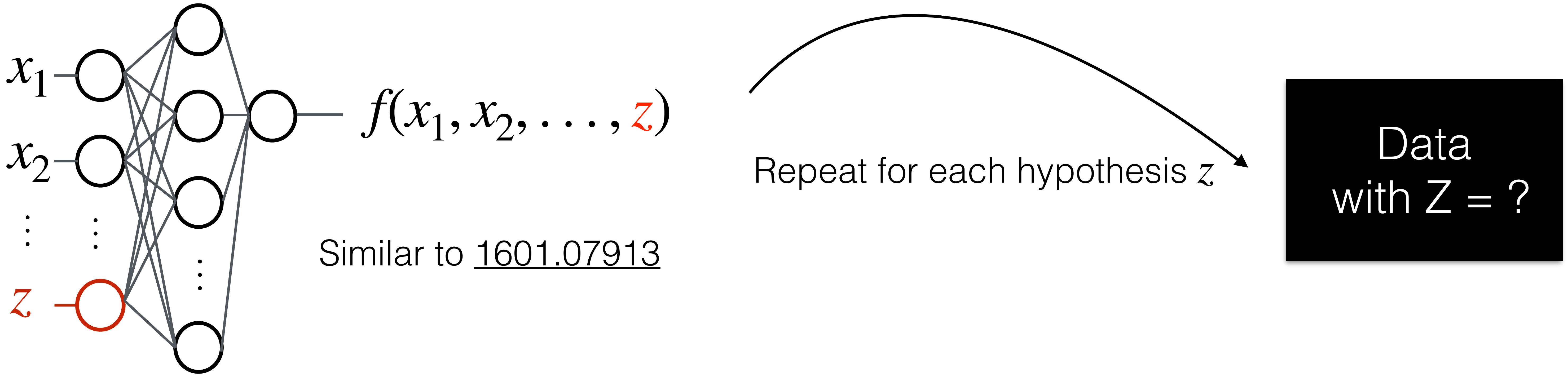
Similar to 1601.07913

- Intuition: Allow the analysis technique to vary with Z
  You always get the best classifier for each value of Z

We advocate for the opposite

PhysRevD.104.056026

- Fully parameterise the classifier on Z in a “systematic aware” way



Similar to 1601.07913

Repeat for each hypothesis z

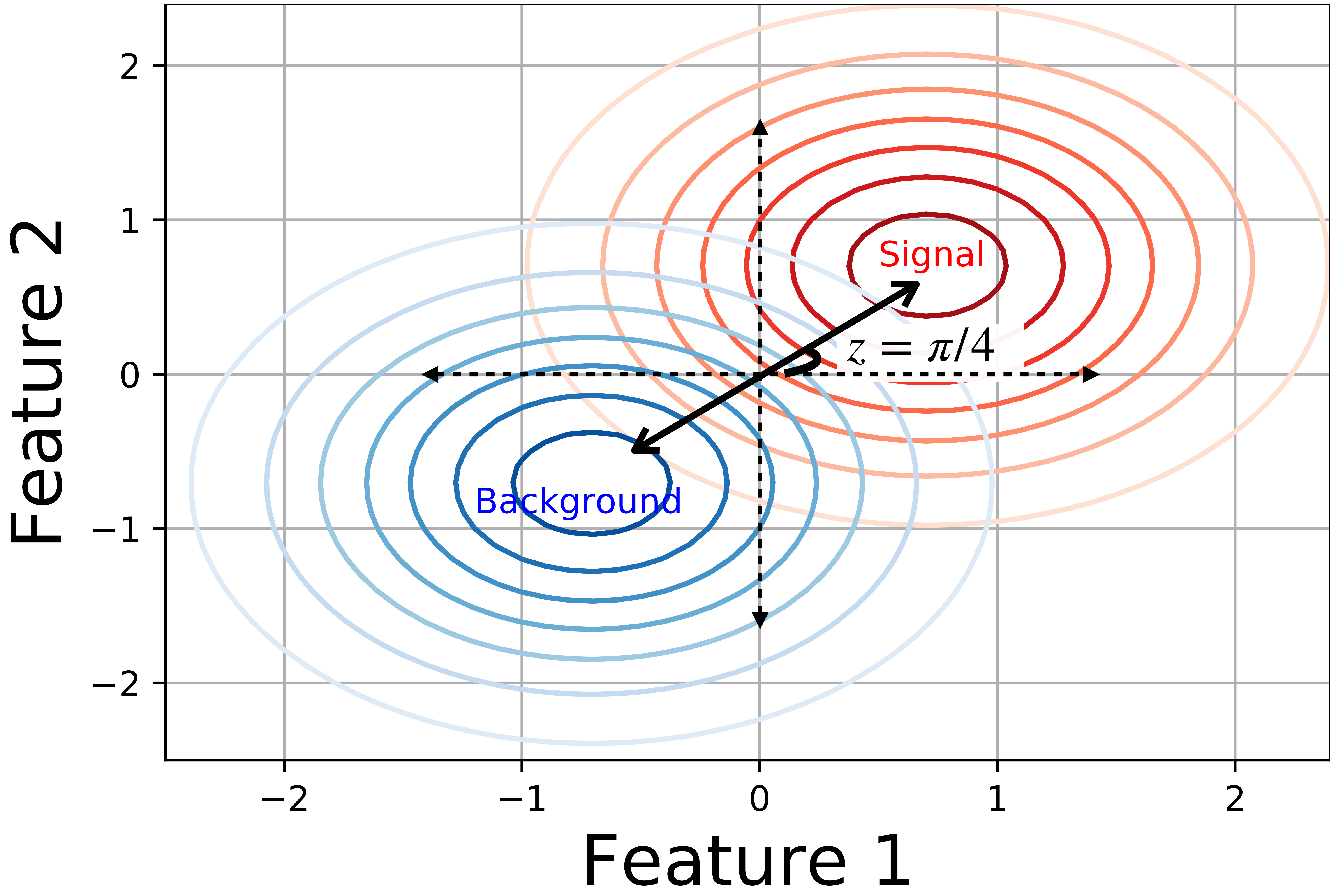
Data
with Z = ?

- Intuition: Allow the analysis technique to vary with Z
  You always get the best classifier for each value of Z

- Use the parameterised classifier response for final likelihood fit to constrain parameters of interest (POI) and nuisance parameters (NP)

In following slides, POI will be the signal strength parameter ‘μ’ and the NP will be denoted ‘Z’

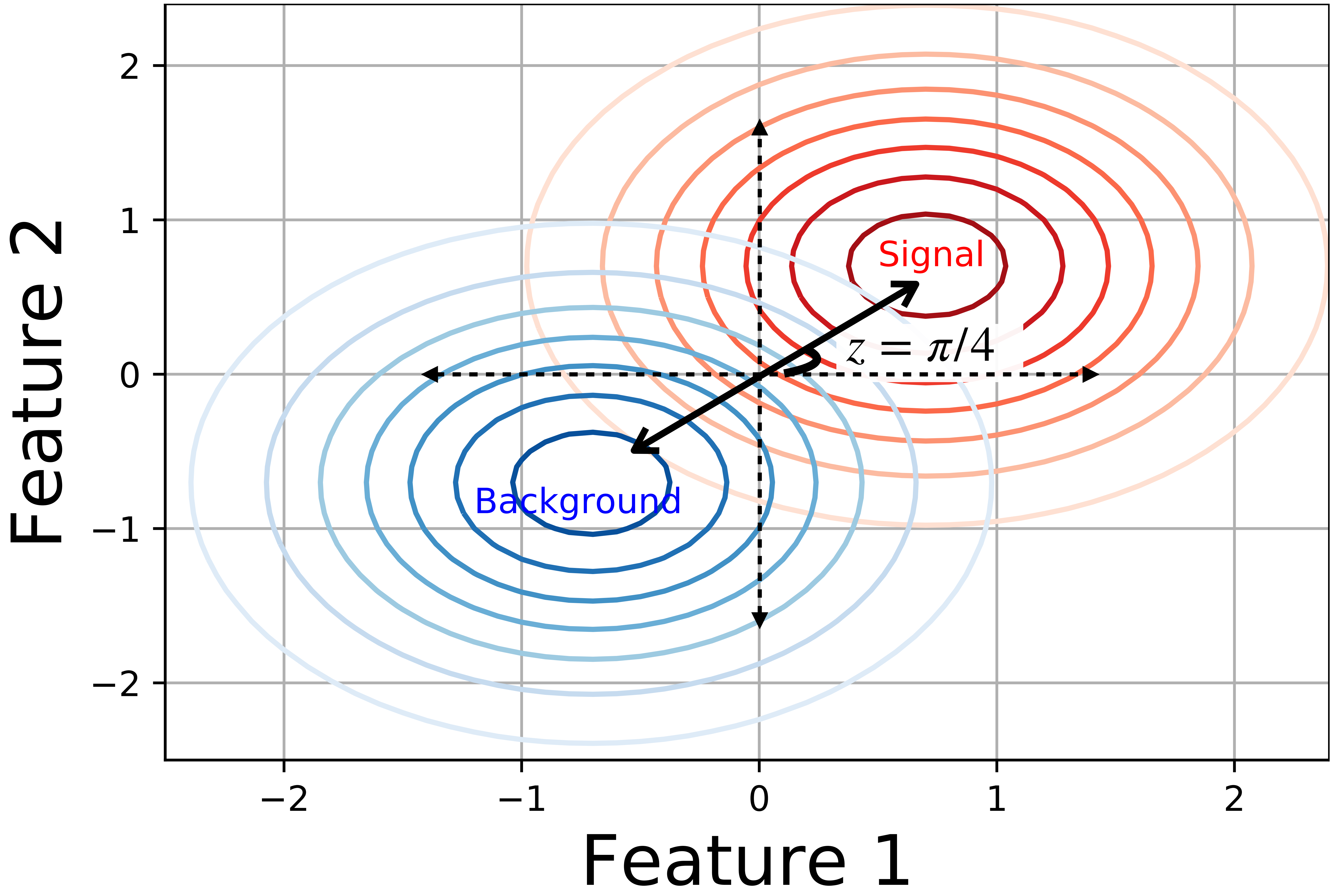
Demonstration on Toy Problem



$$\mu = \frac{N_{s,obs}}{N_{s,exp}}$$

z = Angle

Demonstration on Toy Problem



$$\mu = \frac{N_{s,obs}}{N_{s,exp}}$$

z = Angle

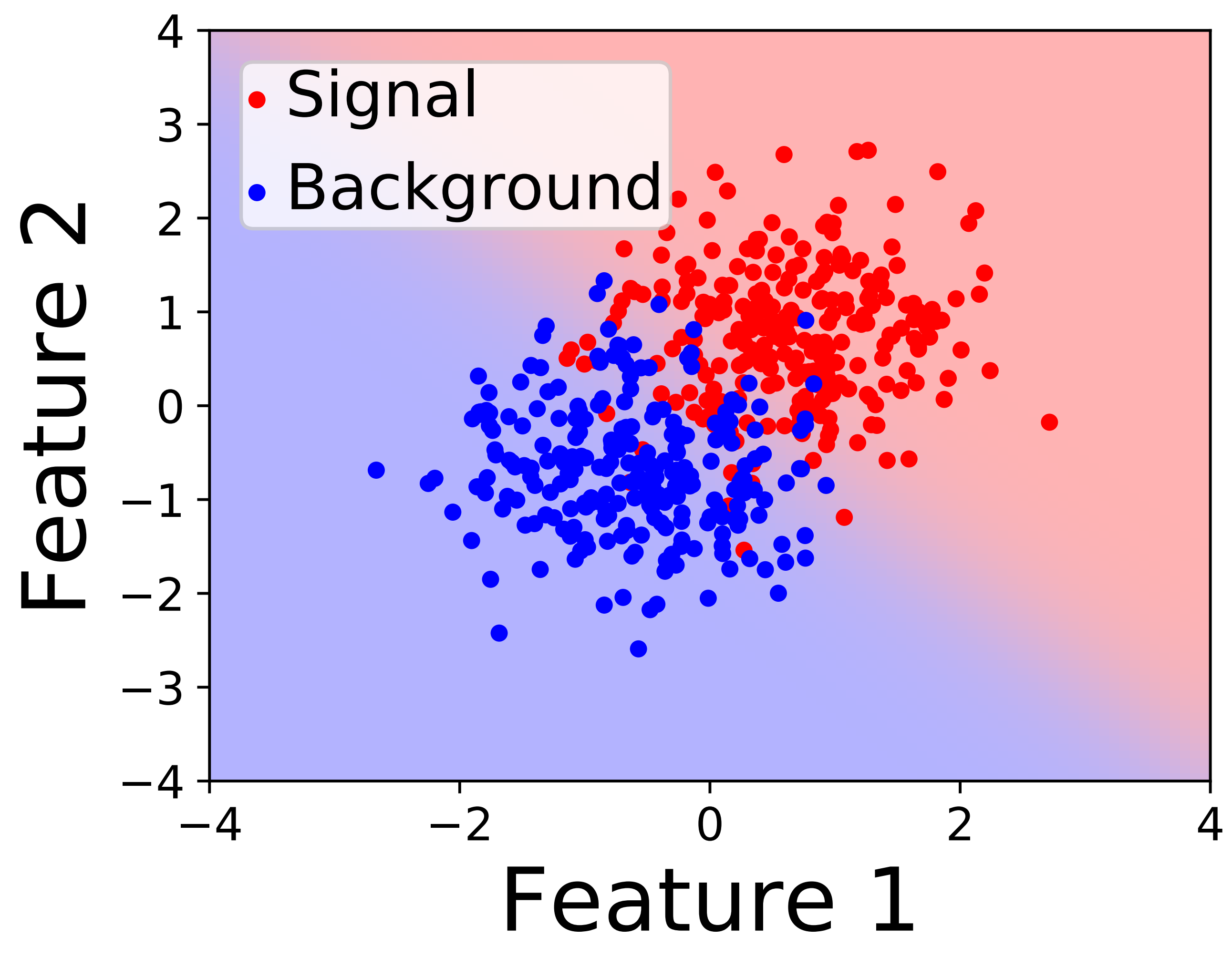
Being invariant to Z would result in a terrible classifier

Nominal and Systematic Up Examples

Baseline Classifier

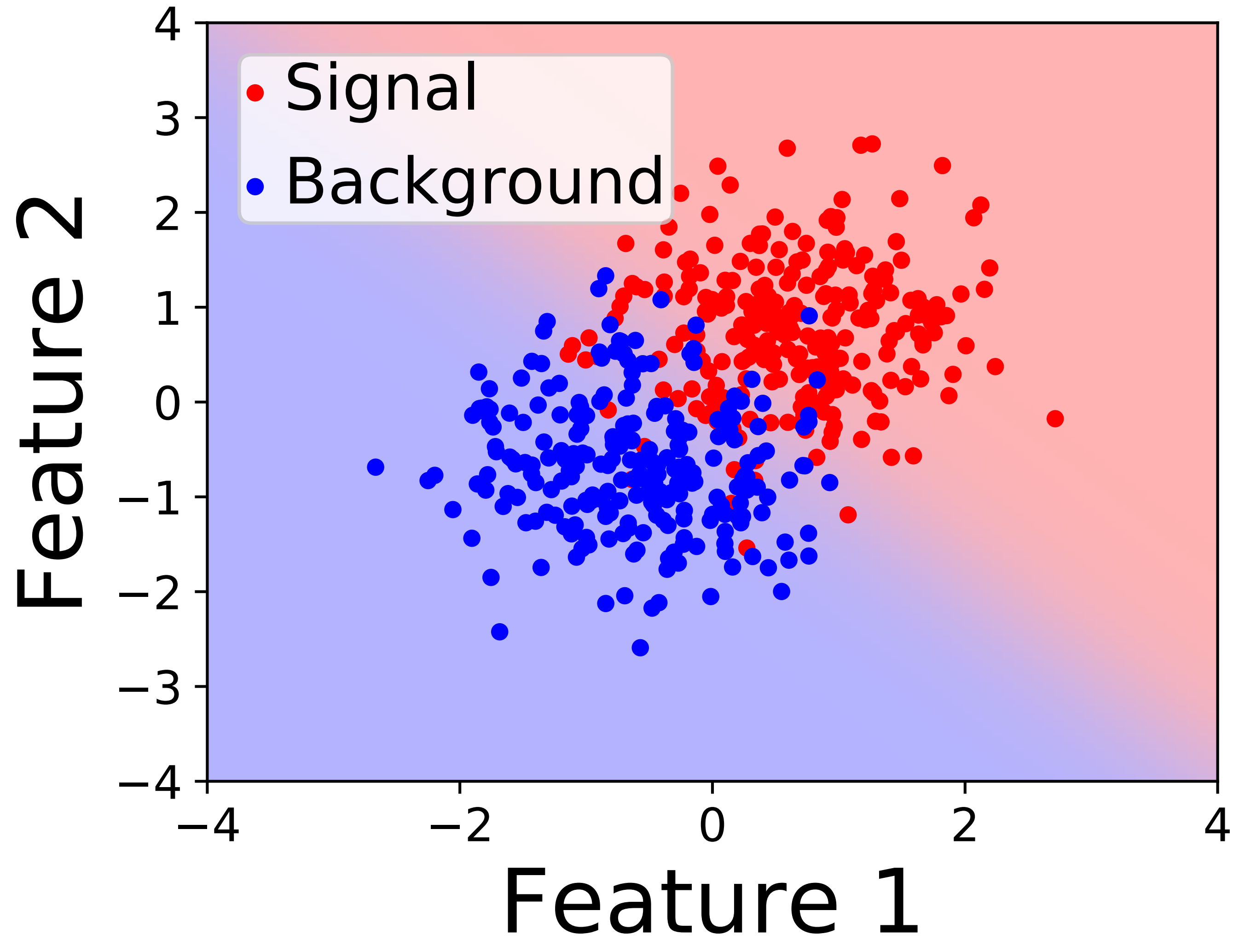
Uncertainty-Aware Classifier

Nominal "Data"



AUC=0.978

Optimal



AUC=0.978

Optimal

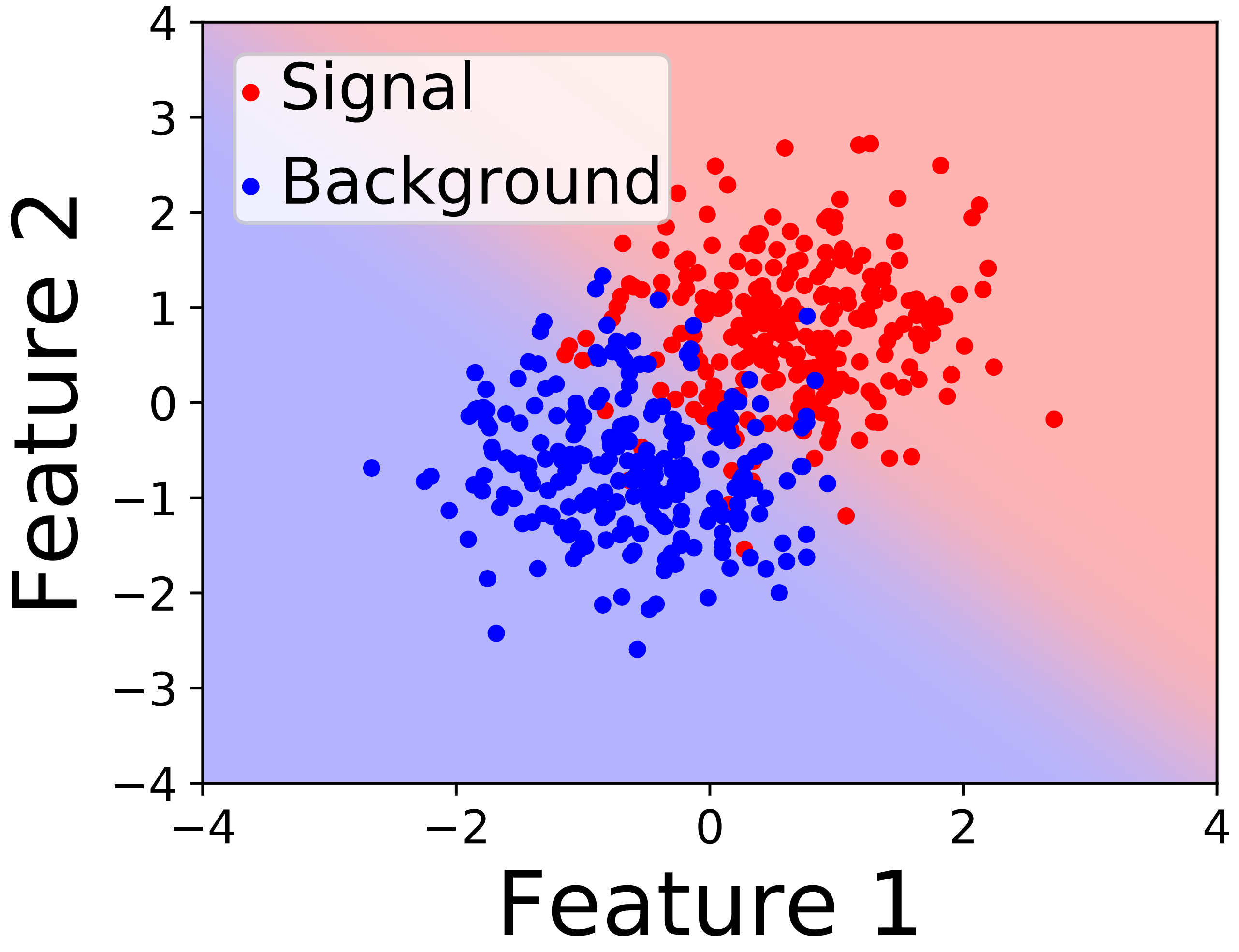
SystUp "Data"

Nominal and Systematic Up Examples

Baseline Classifier

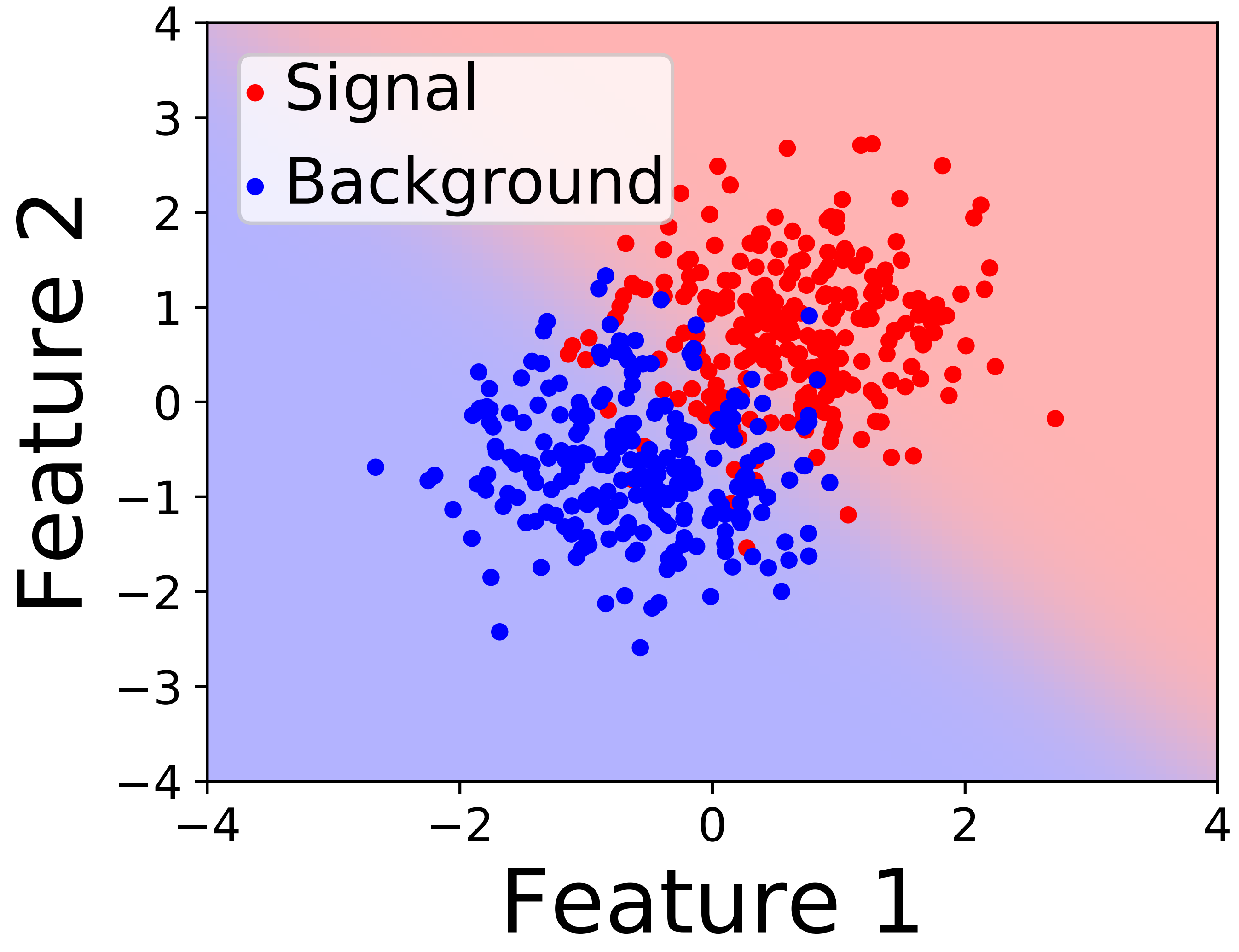
Uncertainty-Aware Classifier

Nominal "Data"



AUC=0.978

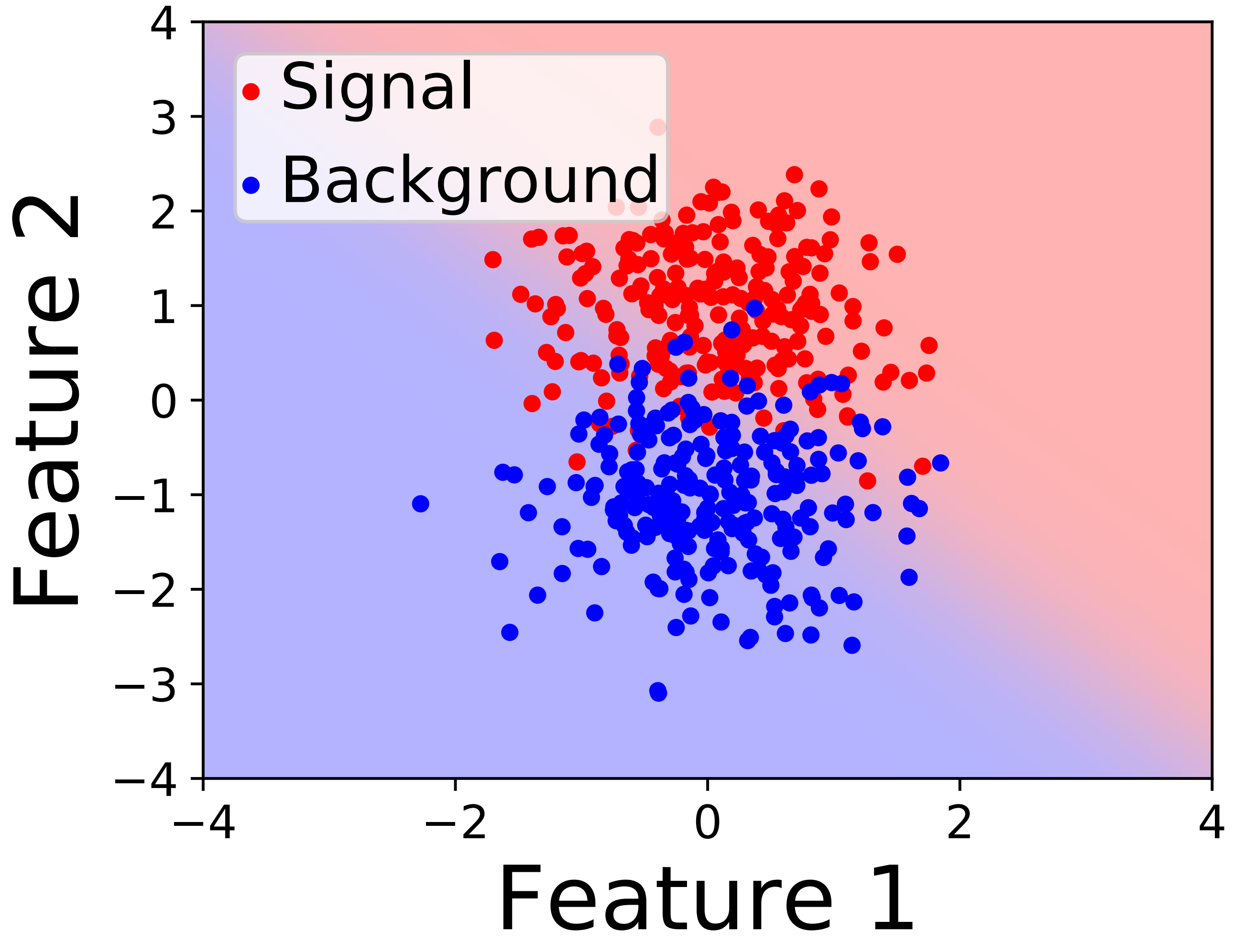
Optimal



AUC=0.978

Optimal

SystUp "Data"



AUC=0.924

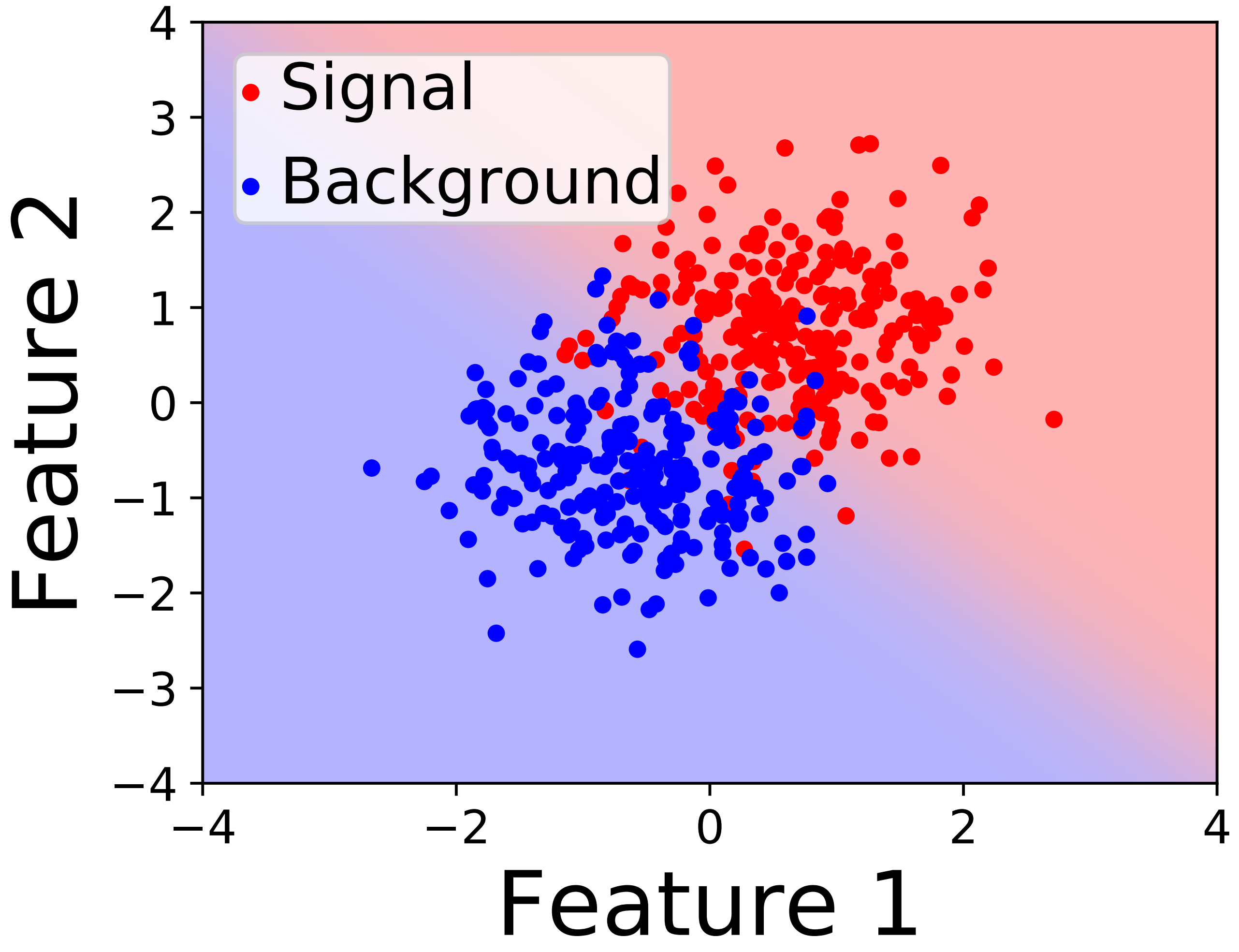
Sub-Optimal

Nominal and Systematic Up Examples

Baseline Classifier

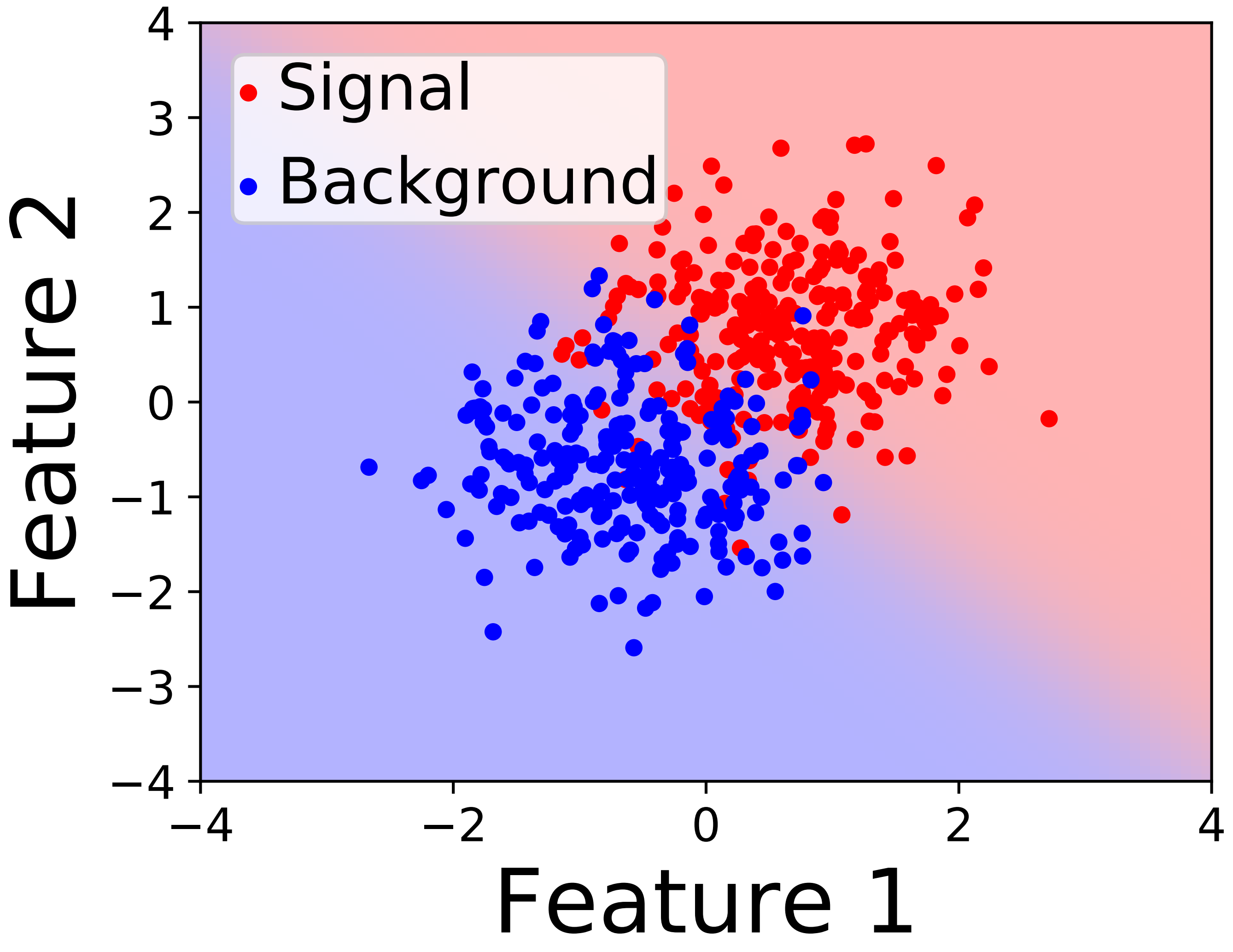
Uncertainty-Aware Classifier

Nominal "Data"



AUC=0.978

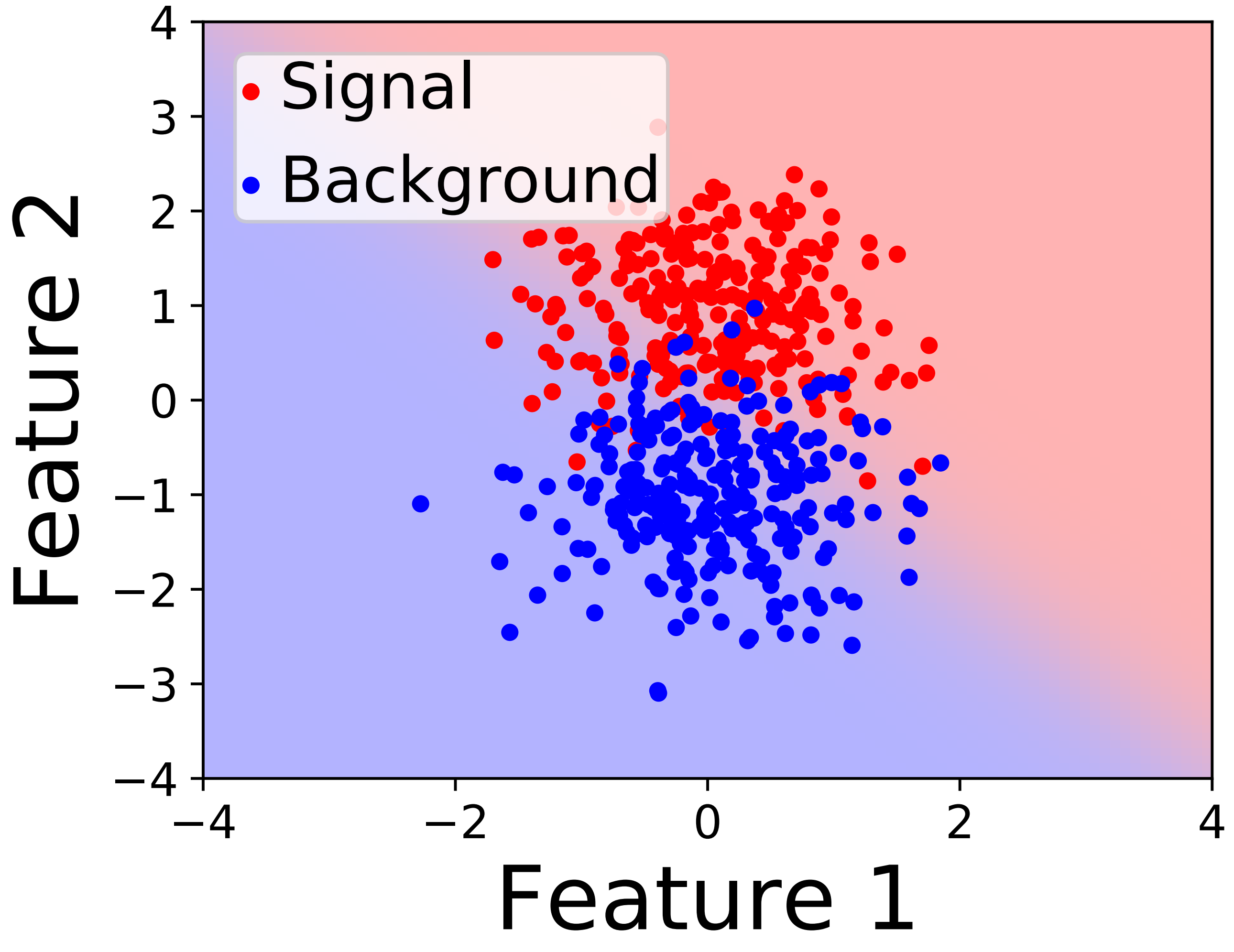
Optimal



AUC=0.978

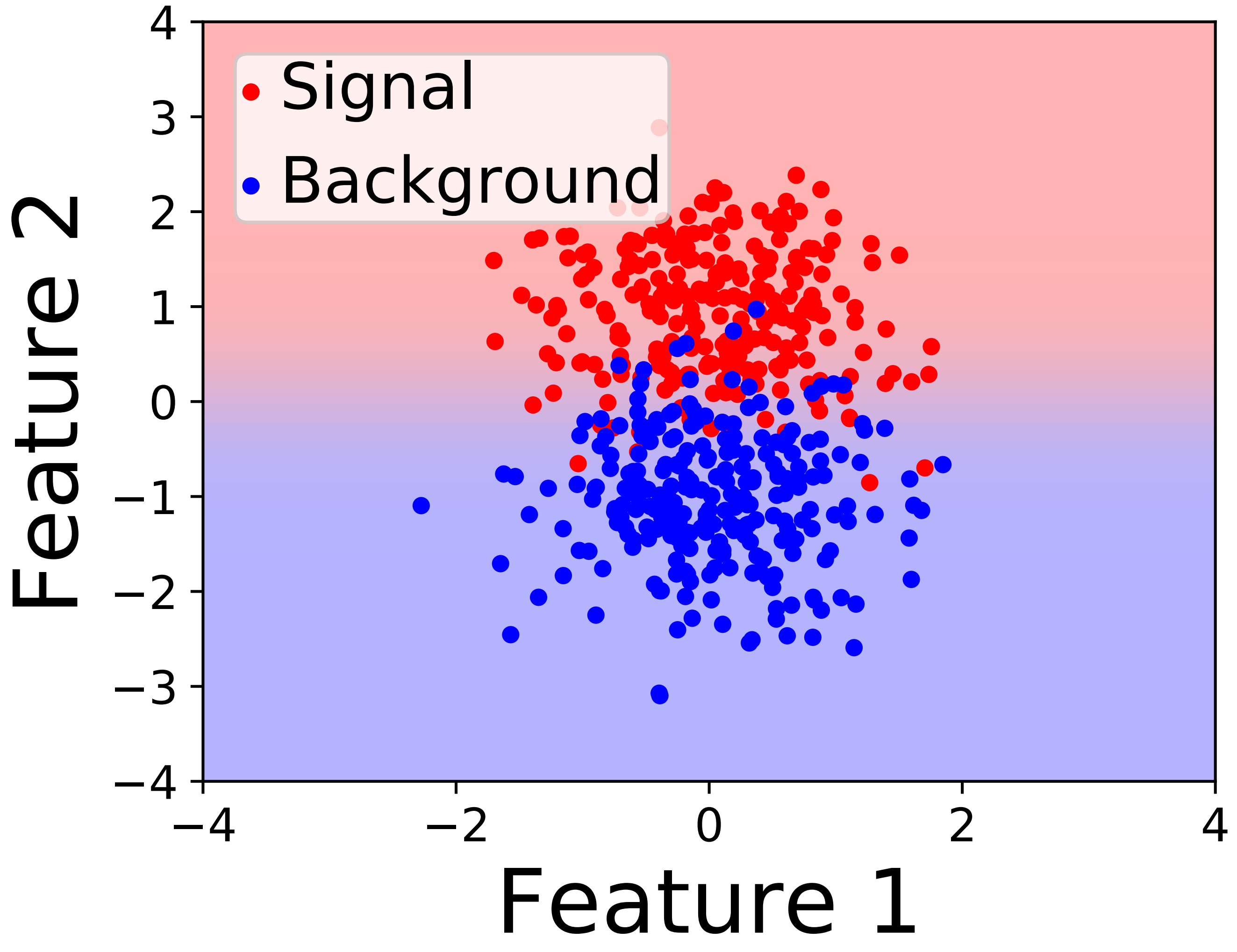
Optimal

SystUp "Data"



AUC=0.924

Sub-Optimal



AUC=0.978

Optimal

Syst-Aware Classifier is able to rotate its decision function based on Z while the Baseline Classifier decision function remains frozen

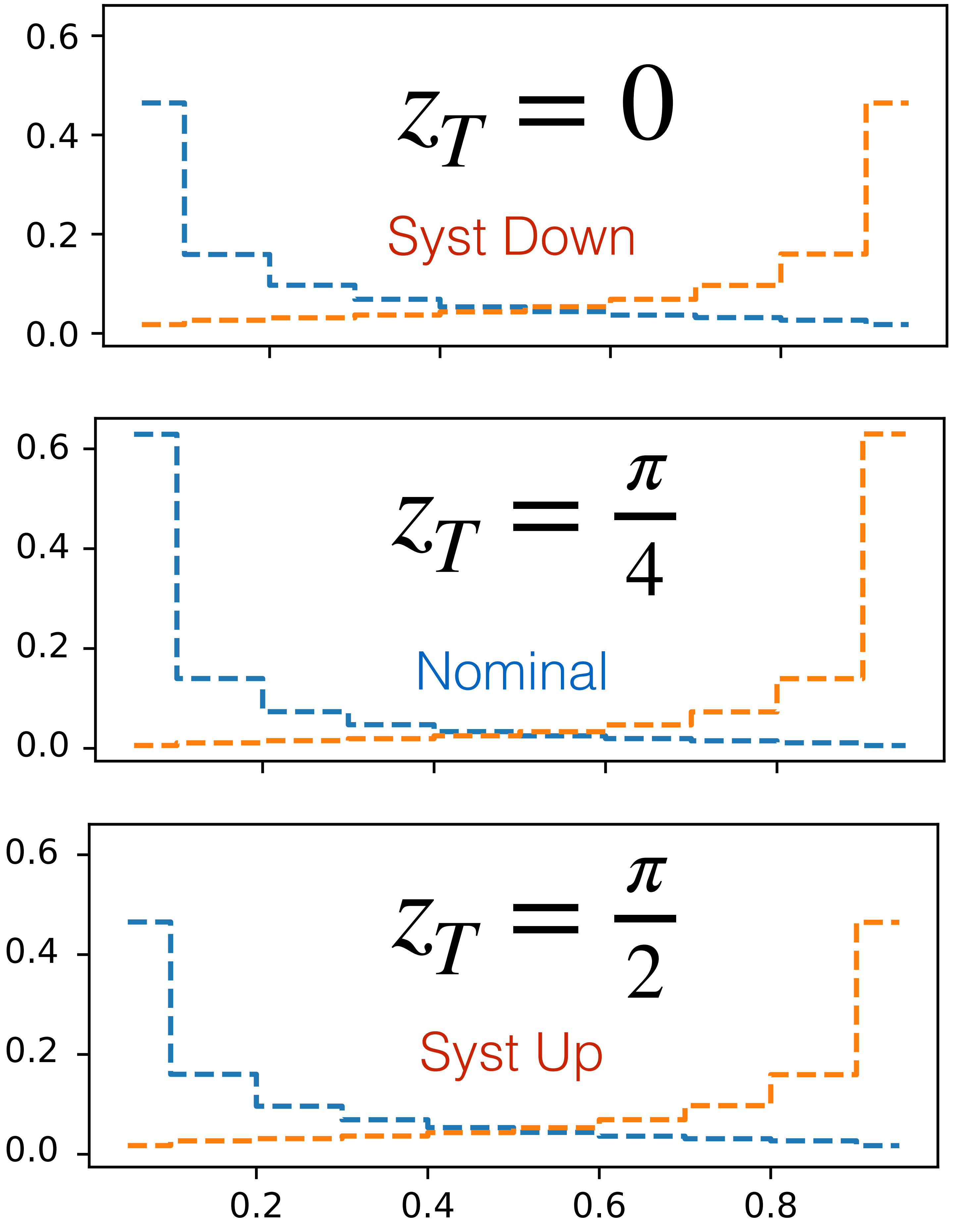
But in a real measurement we don’t know true Z a priori, would this still help?

But in a real measurement we don’t know true Z a priori, would this still help?

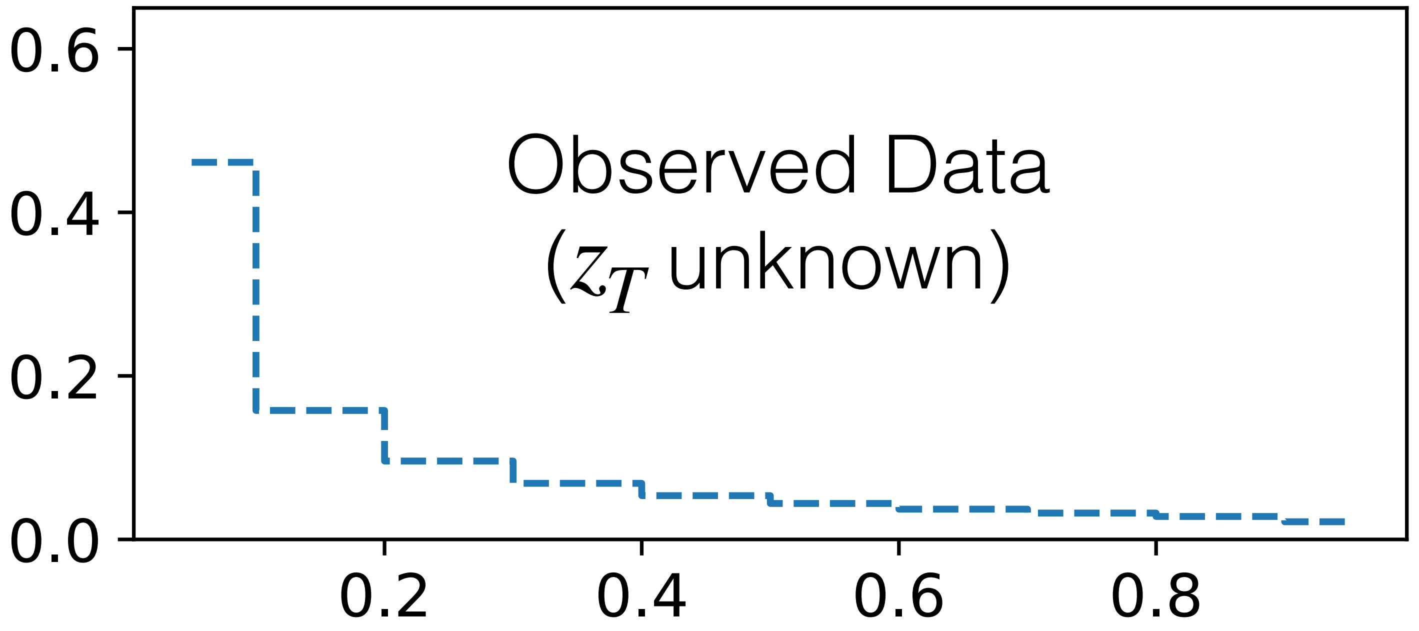
Let’s see what we’ll need to do..

Scan the 2D Likelihood space in Z vs μ

Template **Baseline Classifier** Score Histograms for various Z

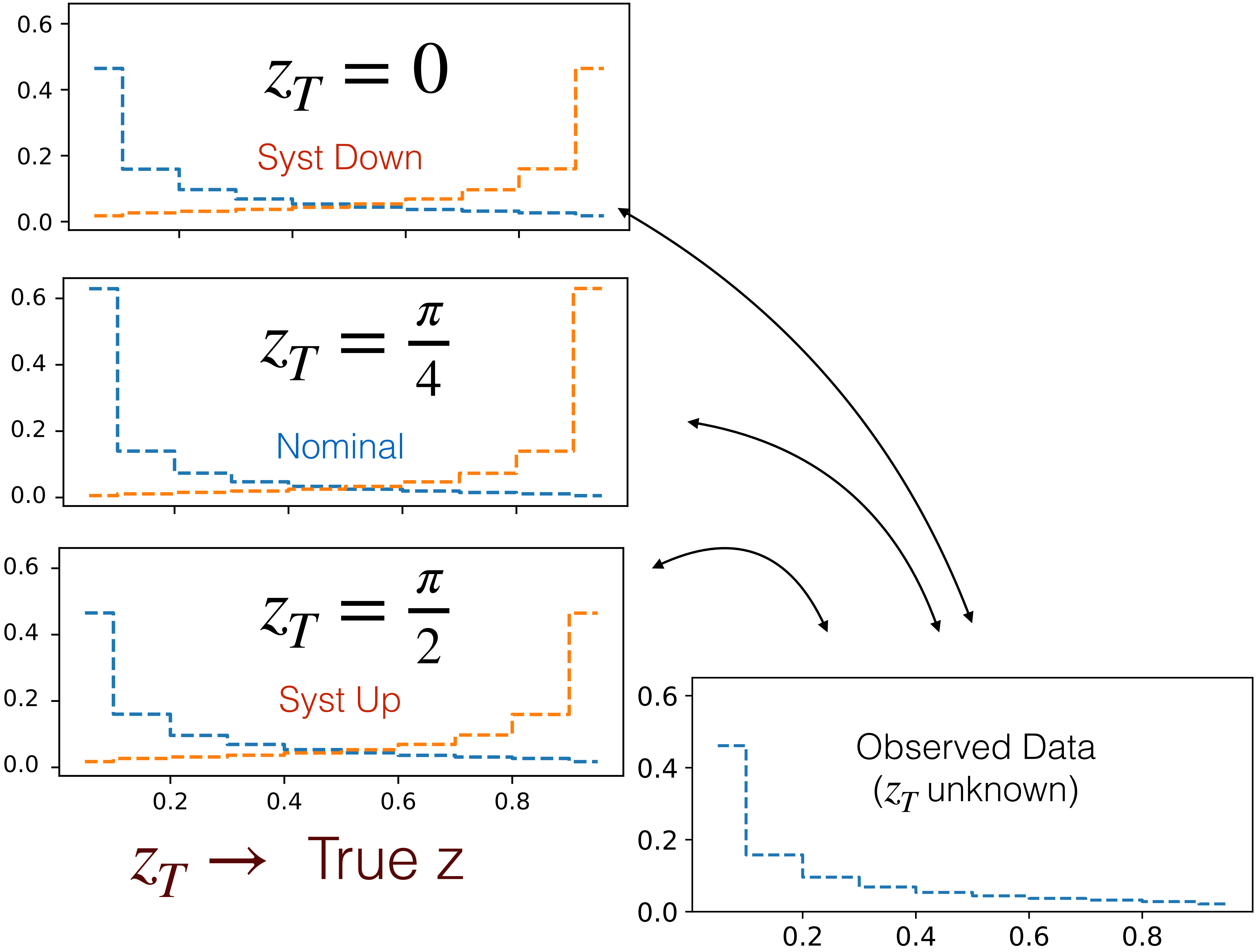


$z_T \rightarrow$ True z



Scan the 2D Likelihood space in Z vs μ

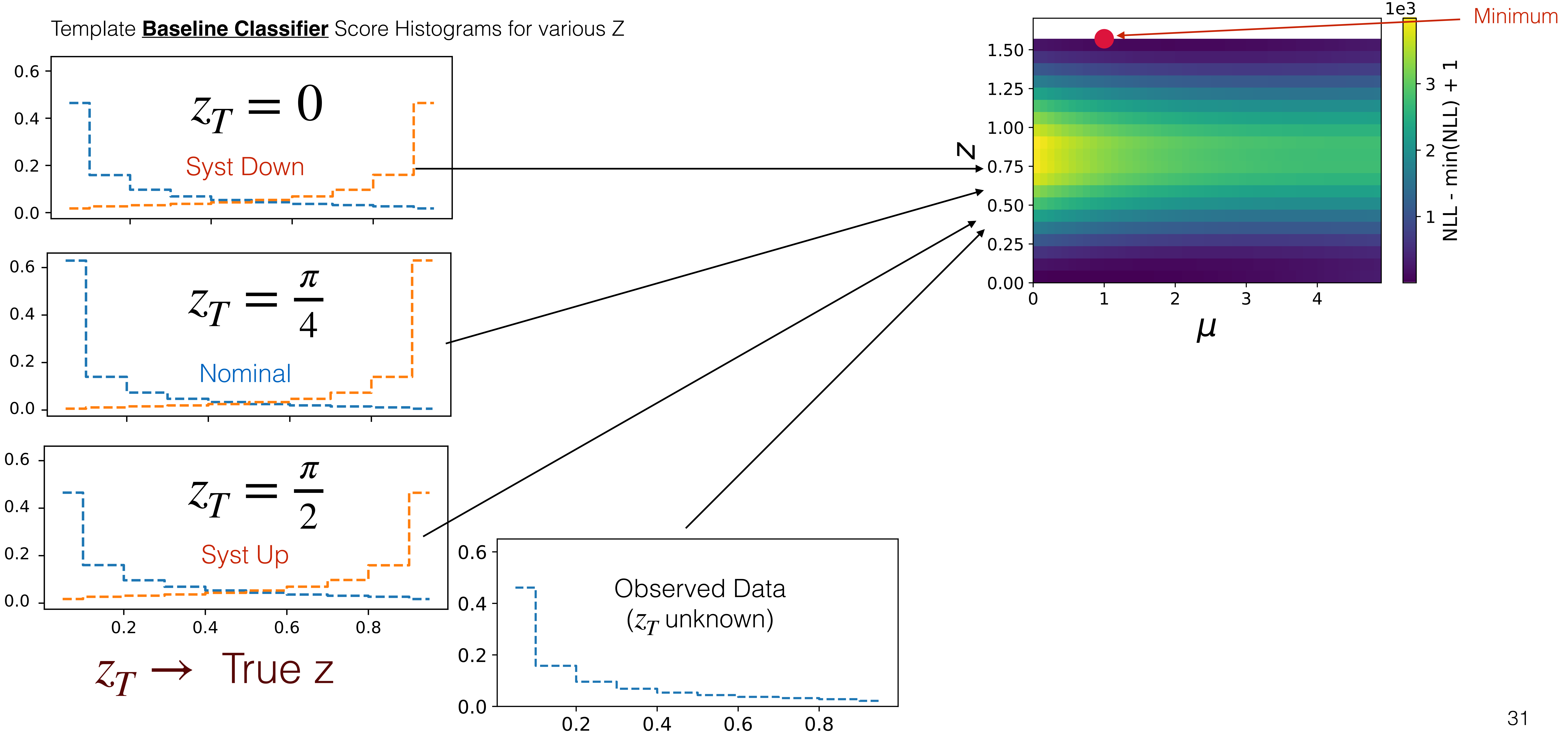
Template **Baseline Classifier** Score Histograms for various Z



z_T → True z

Scan the 2D Likelihood space in Z vs μ

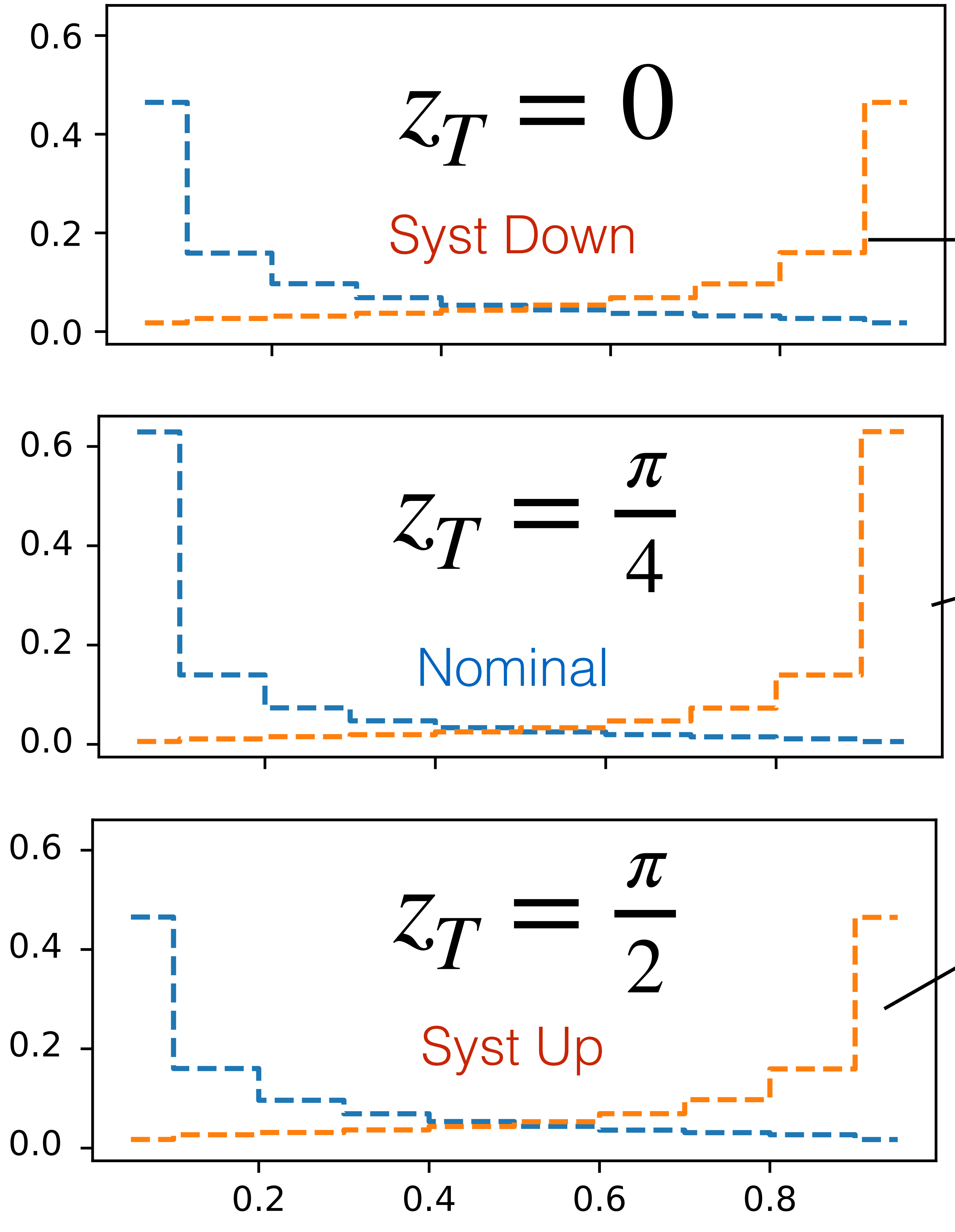
Template **Baseline Classifier** Score Histograms for various Z

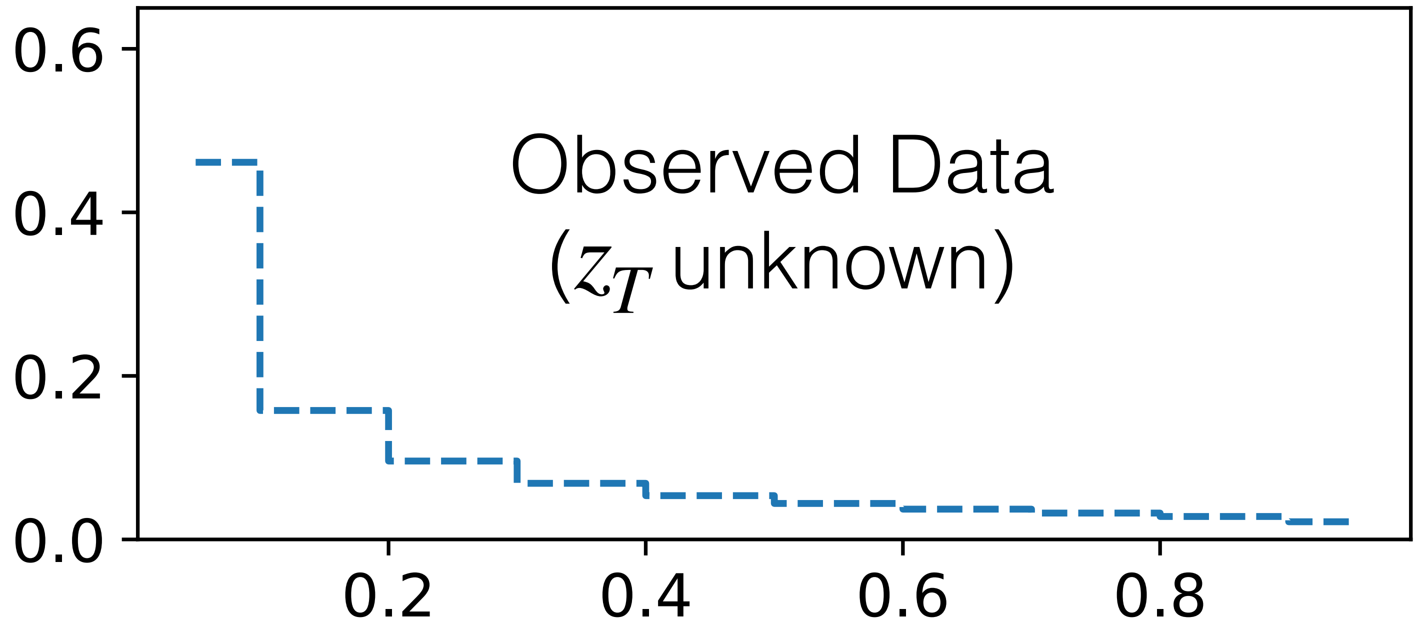


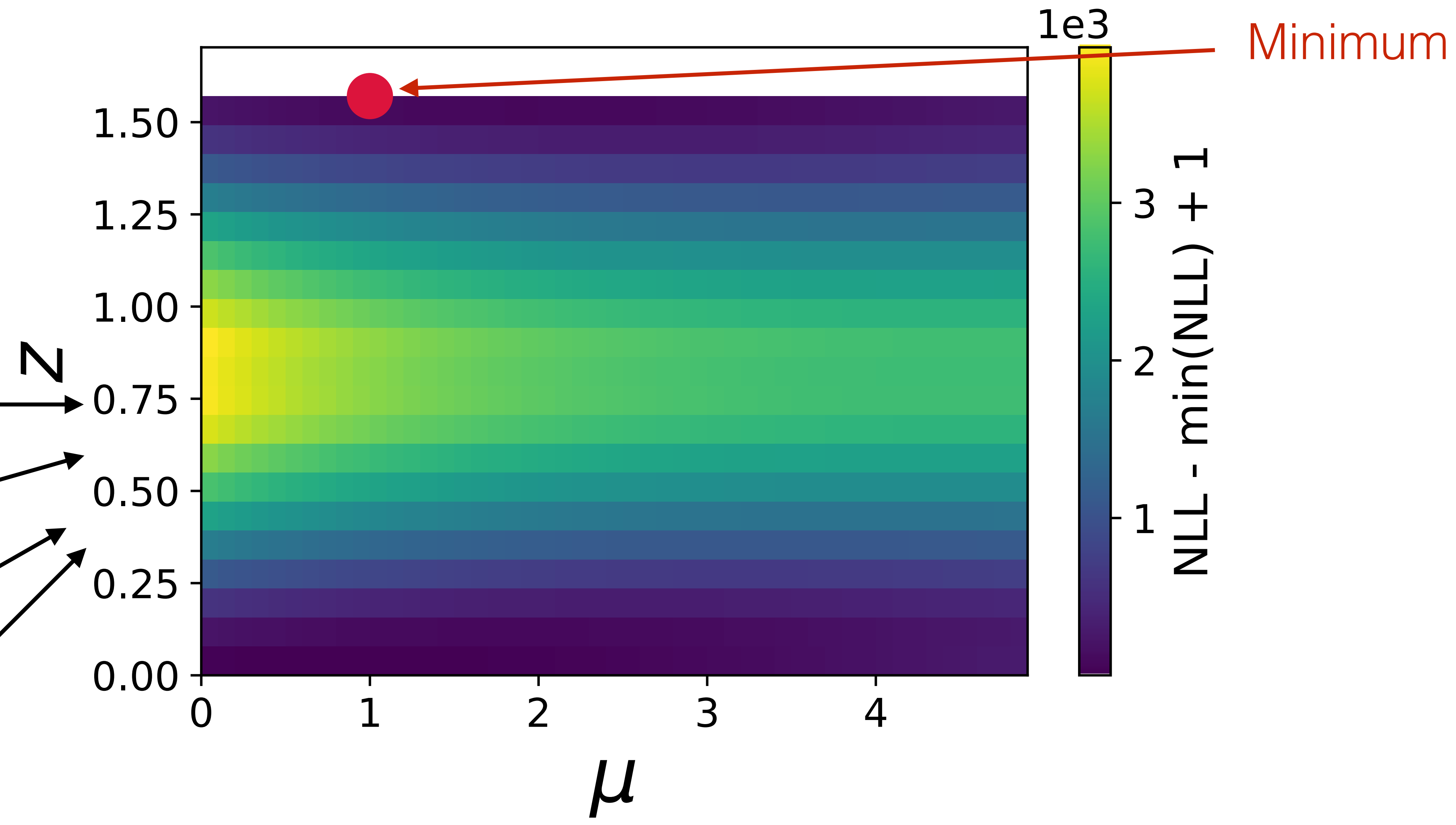
$z_T \rightarrow$ True z

Scan the 2D Likelihood space in Z vs μ

Template **Baseline Classifier** Score Histograms for various Z







Likelihood statistical component = Poisson per histogram bin
Likelihood systematic component = Gaussian (1, 0.5) as prior on Z
Full Likelihood = statistical + systematic

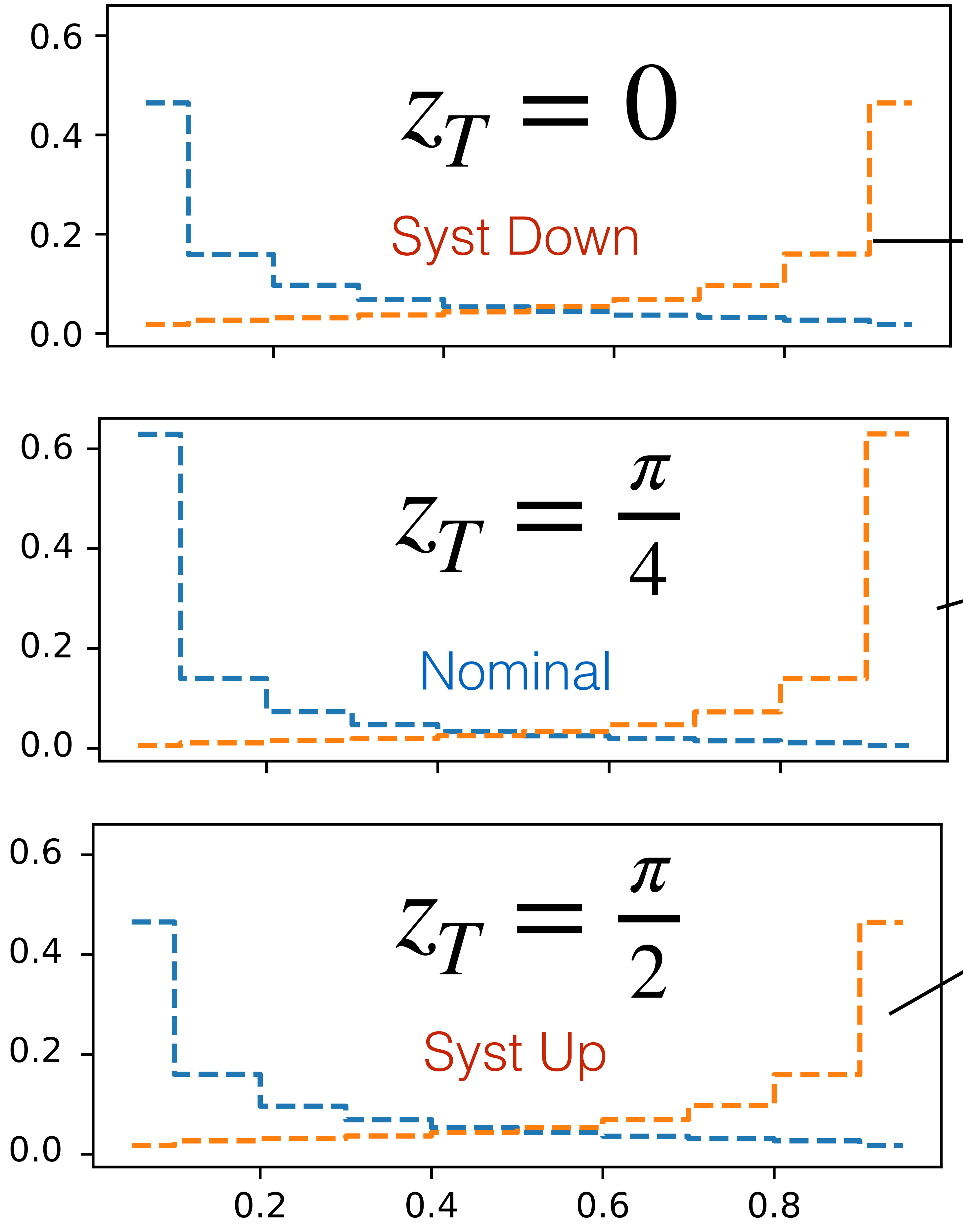
$$-\log\mathcal{L}(\mu, z|\{x_i\}) = -\sum_{j=1}^{n_{\mathrm{bins}}}\left[N_j\cdot\log\left(\mu s_j+b_j\right)-\mu s_j-b_j-\log(\Gamma(N_i))\right] + \left(\frac{z-z_0}{\sqrt{2}\sigma_z}\right)^2,$$

$z_T \rightarrow$ True z

But could be done unbinned/KDE too

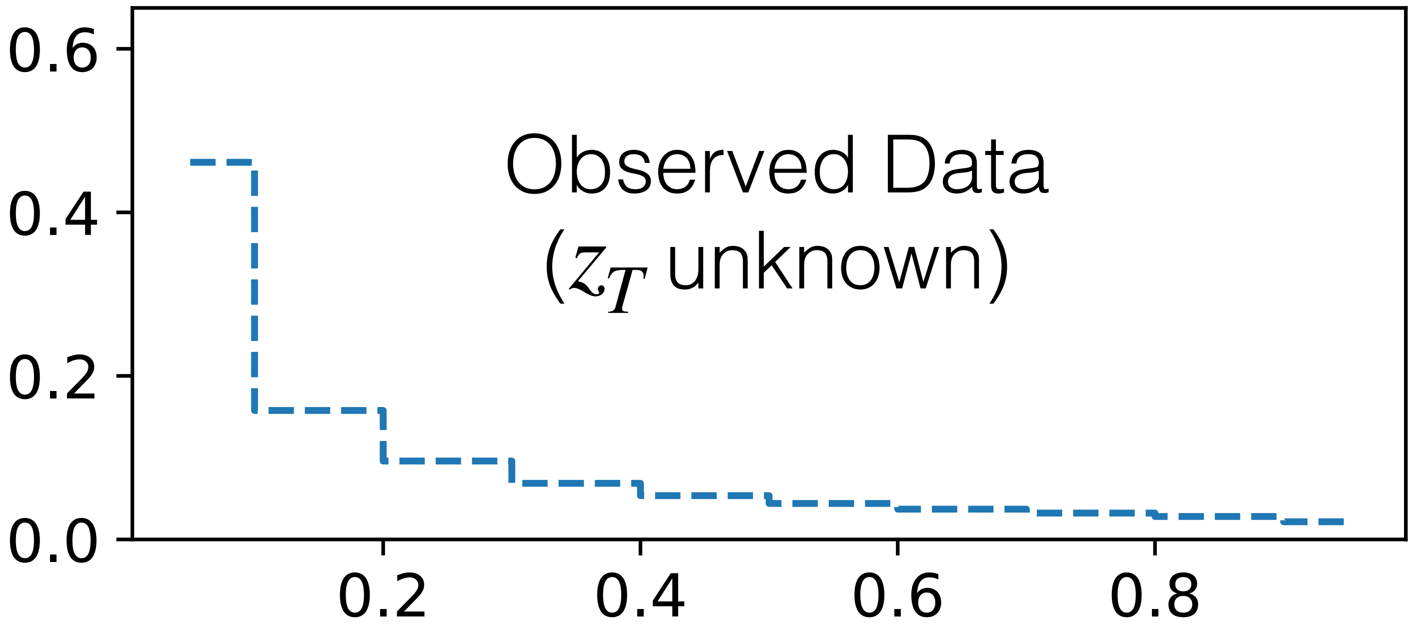
Scan the 2D Likelihood space in Z vs μ

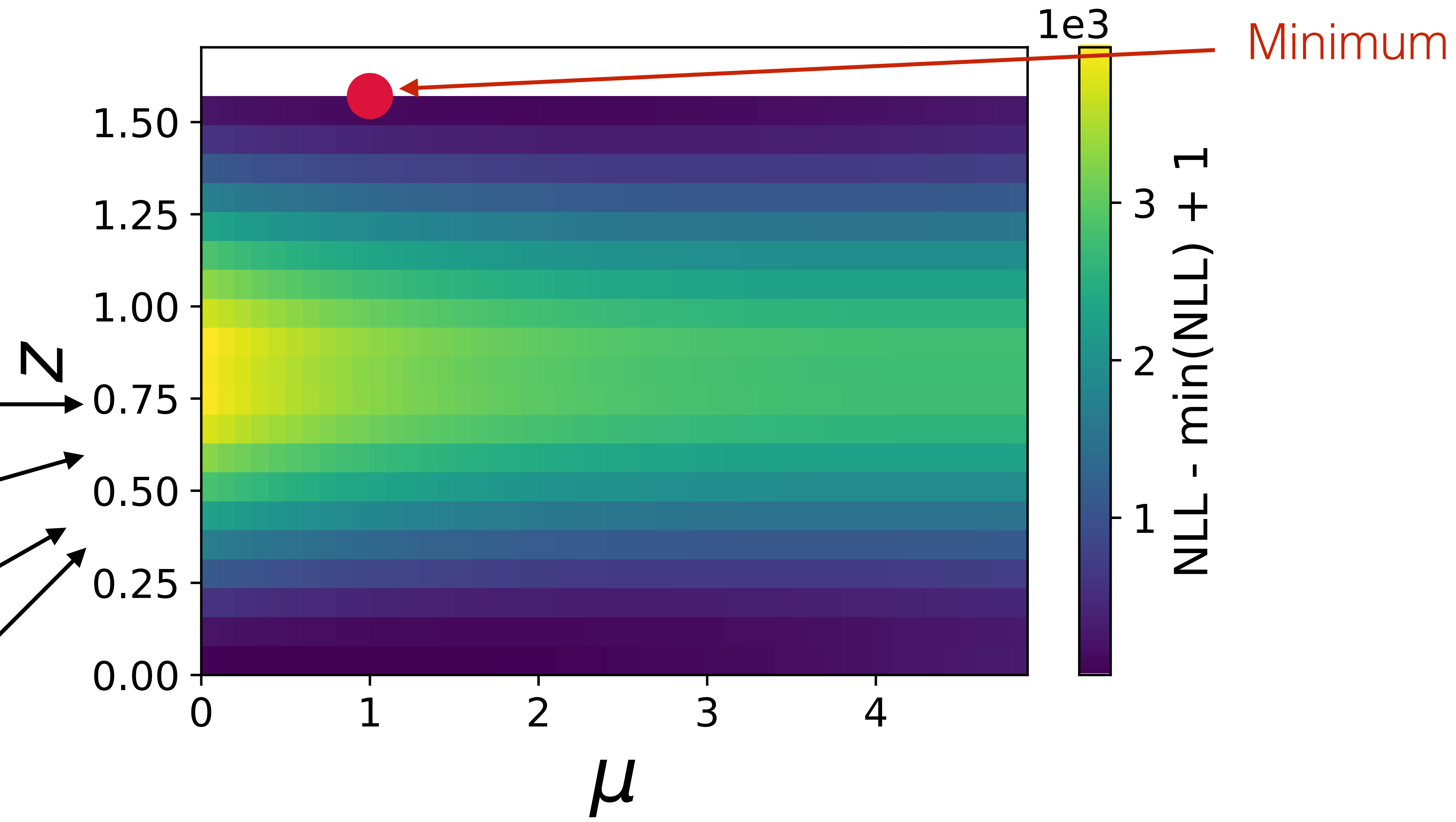
Template **Baseline Classifier** Score Histograms for various Z



$z_T \rightarrow$ True z

But could be done unbinned/KDE too



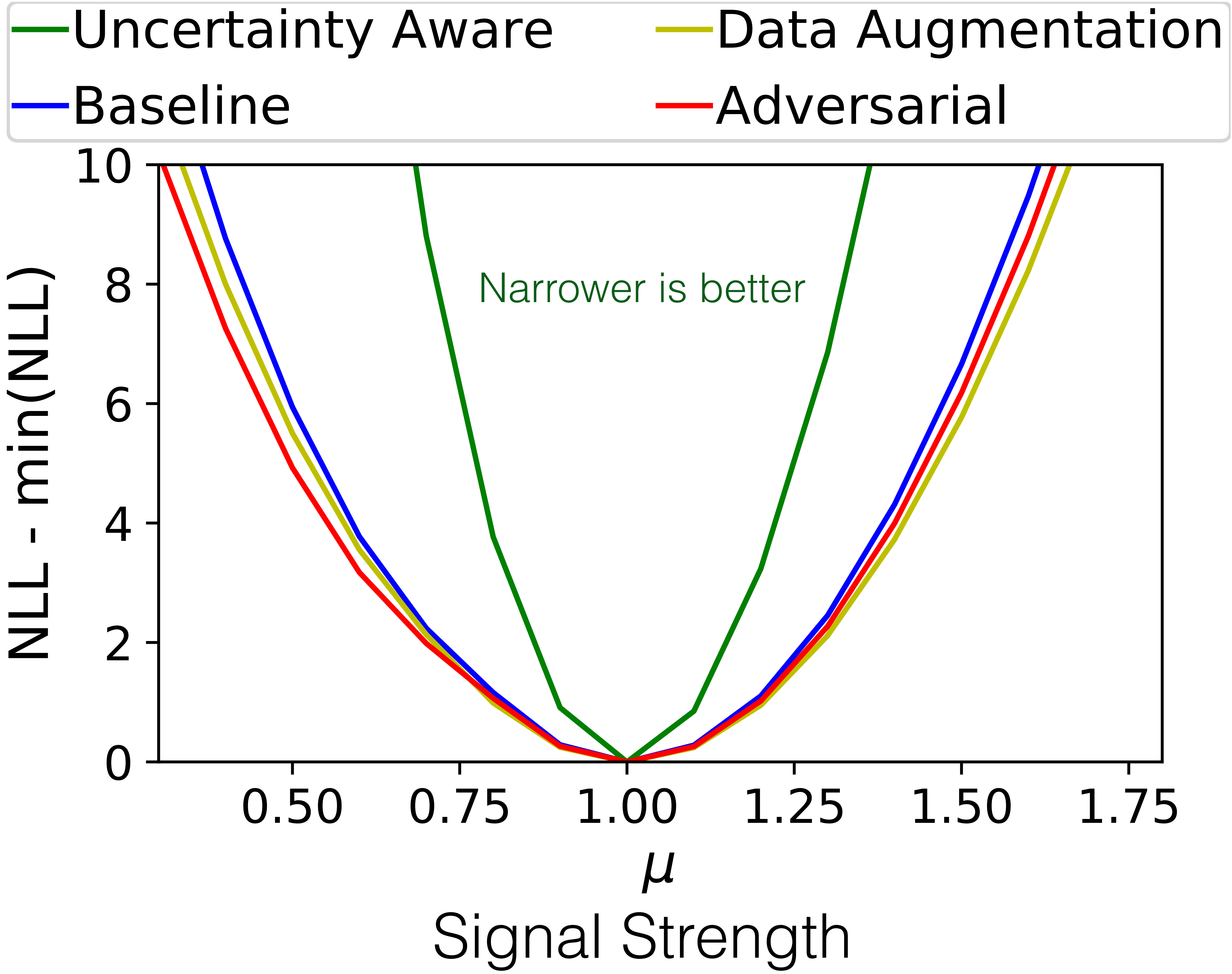


Likelihood statistical component = Poisson per histogram bin
Likelihood systematic component = Gaussian (1, 0.5) as prior on Z
Full Likelihood = statistical + systematic

$$-\log\mathcal{L}(\mu, z|\{x_i\}) = -\sum_{j=1}^{n_{\mathrm{bins}}}\left[N_j\cdot\log\left(\mu s_j+b_j\right)-\mu s_j-b_j-\log(\Gamma(N_i))\right] + \left(\frac{z-z_0}{\sqrt{2}\sigma_z}\right)^2,$$

Next step: profile over Z dimension (take the bin with maximum likelihood in each column)

Profile away Z - Example at $(\mu, Z)_{\text{True}} = (1, 1.57)$



Narrower is better: We can exclude wrong values of μ with greater confidence.

The profiled (Negative-Log-) Likelihood curve for Uncertainty-Aware classifier is much narrower ⇒ smallest

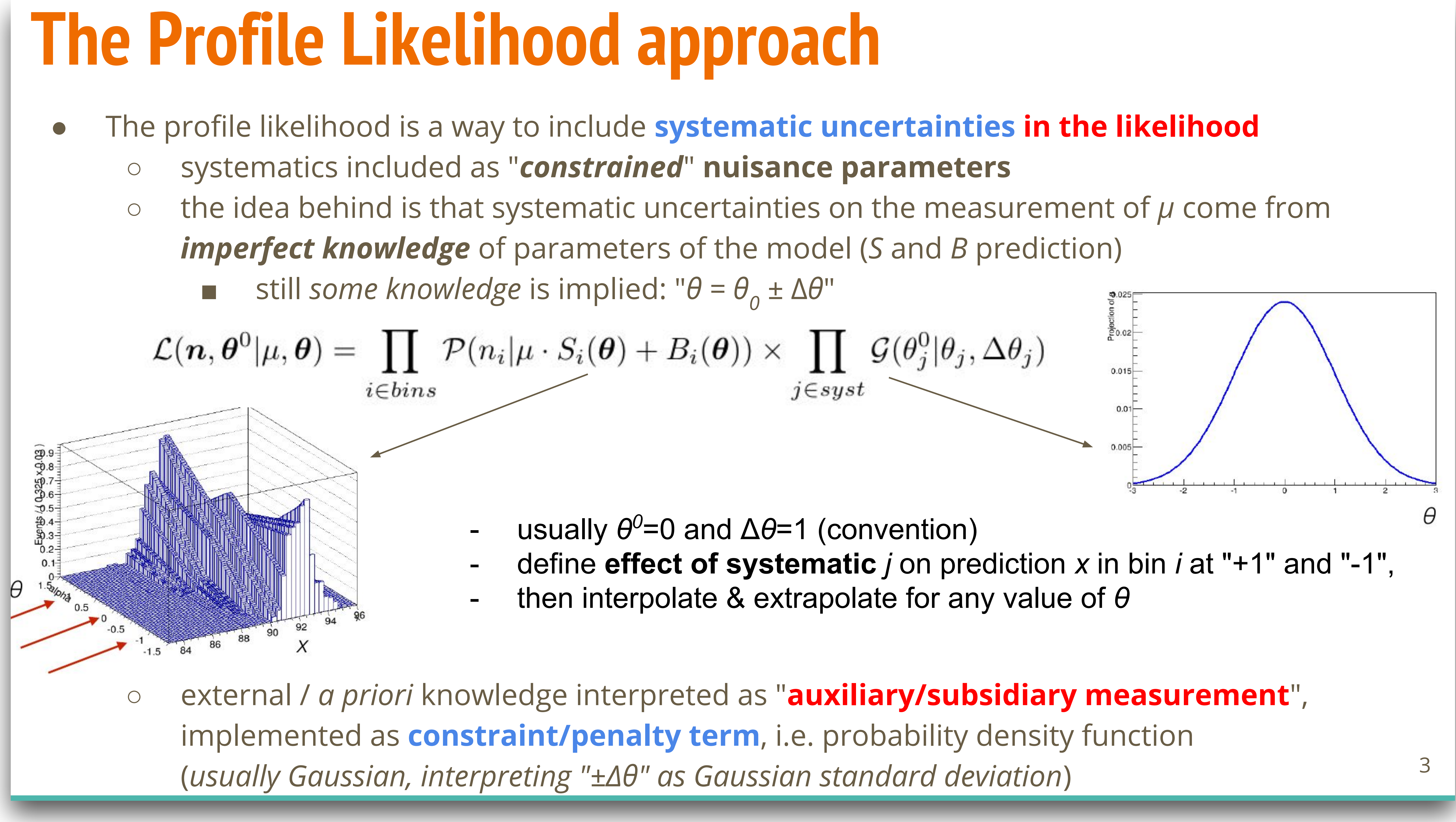
[statistical + systematic] uncertainty on measurement

Profile Likelihood

Standard method of including the systematic uncertainty into the likelihood computation

We simply make the selection/observable a function of z

In principle could also be done in cut-based analysis: make cut a continuous function of z



The Profile Likelihood approach

- The profile likelihood is a way to include **systematic uncertainties in the likelihood**
  - systematics included as "***constrained***" **nuisance parameters**
  - the idea behind is that systematic uncertainties on the measurement of μ come from ***imperfect knowledge*** of parameters of the model (S and B prediction)
    - still *some knowledge* is implied: "$\theta = \theta_0 \pm \Delta\theta$"

$$\mathcal{L}(\boldsymbol{n}, \boldsymbol{\theta}^0 | \mu, \boldsymbol{\theta}) = \prod_{i \in bins} \mathcal{P}(n_i | \mu \cdot S_i(\boldsymbol{\theta}) + B_i(\boldsymbol{\theta})) \times \prod_{j \in syst} \mathcal{G}(\theta_j^0 | \theta_j, \Delta\theta_j)$$

- usually θ^0=0 and $\Delta\theta$=1 (convention)
- define **effect of systematic** *j* on prediction *x* in bin *i* at "+1" and "-1",
- then interpolate & extrapolate for any value of θ

  - external / *a priori* knowledge interpreted as "**auxiliary/subsidiary measurement**", implemented as **constraint/penalty term**, i.e. probability density function (*usually Gaussian, interpreting "±Δθ" as Gaussian standard deviation*)

3

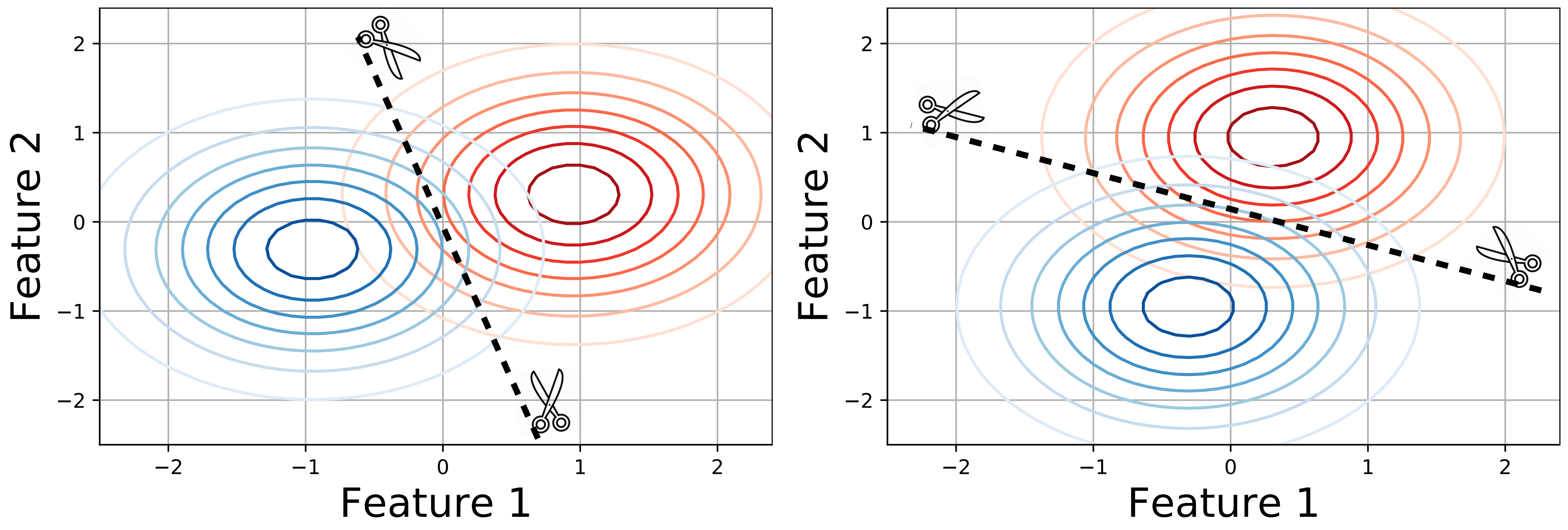
From Michele Pinamonti's talk:
https://indico.cern.ch/event/727396/contributions/3021899/attachments/1657532/2654085/Statistical_methods_at_ATLAS_and_CMS_2.pdf

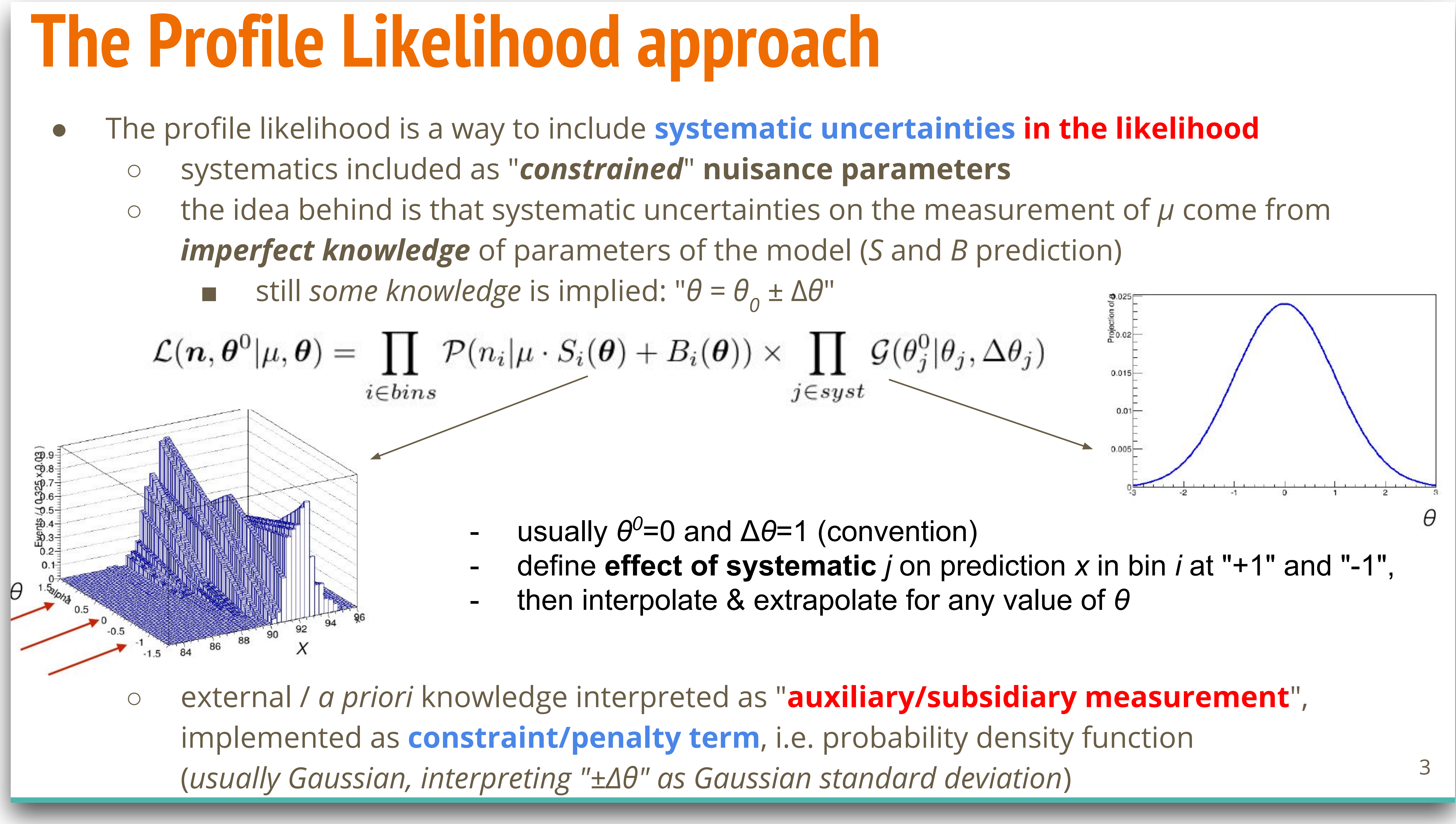
Profile Likelihood

Standard method of including the systematic uncertainty into the likelihood computation

We simply make the selection/observable a function of z

In principle could also be done in cut-based analysis: make cut a continuous function of z





The Profile Likelihood approach

- The profile likelihood is a way to include **systematic uncertainties** **in the likelihood**
  - systematics included as "***constrained***" **nuisance parameters**
  - the idea behind is that systematic uncertainties on the measurement of μ come from ***imperfect knowledge*** of parameters of the model (S and B prediction)
    - still *some knowledge* is implied: "$\theta = \theta_0 \pm \Delta\theta$"

$$\mathcal{L}(\boldsymbol{n}, \boldsymbol{\theta}^0 | \mu, \boldsymbol{\theta}) = \prod_{i \in bins} \mathcal{P}(n_i | \mu \cdot S_i(\boldsymbol{\theta}) + B_i(\boldsymbol{\theta})) \times \prod_{j \in syst} \mathcal{G}(\theta_j^0 | \theta_j, \Delta\theta_j)$$

- usually θ^0=0 and $\Delta\theta$=1 (convention)
- define **effect of systematic** *j* on prediction *x* in bin *i* at "+1" and "-1",
- then interpolate & extrapolate for any value of θ

  - external / *a priori* knowledge interpreted as "**auxiliary/subsidiary measurement**", implemented as **constraint/penalty term**, i.e. probability density function (*usually Gaussian, interpreting "±Δθ" as Gaussian standard deviation*)

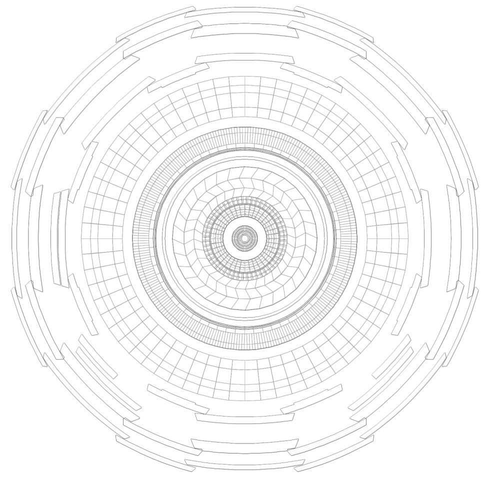
3

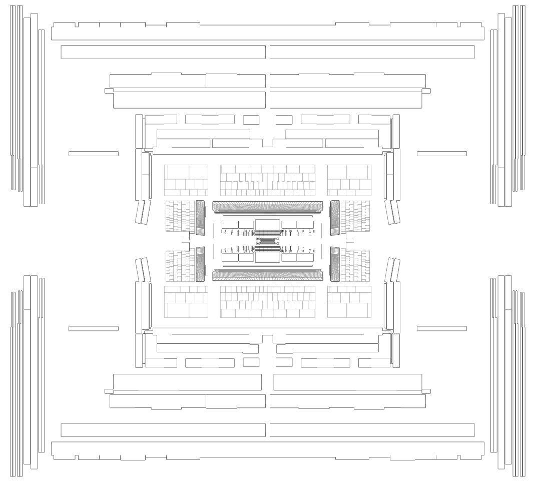
From Michele Pinamonti's talk:
https://indico.cern.ch/event/727396/contributions/3021899/attachments/1657532/2654085/Statistical_methods_at_ATLAS_and_CMS_2.pdf

But in a real measurement we don’t know true Z a priori, would this still help?

But in a real measurement we don’t know true Z a priori, would this still help?

Yes!





But in a real measurement we don’t know true Z a priori, would this still help?

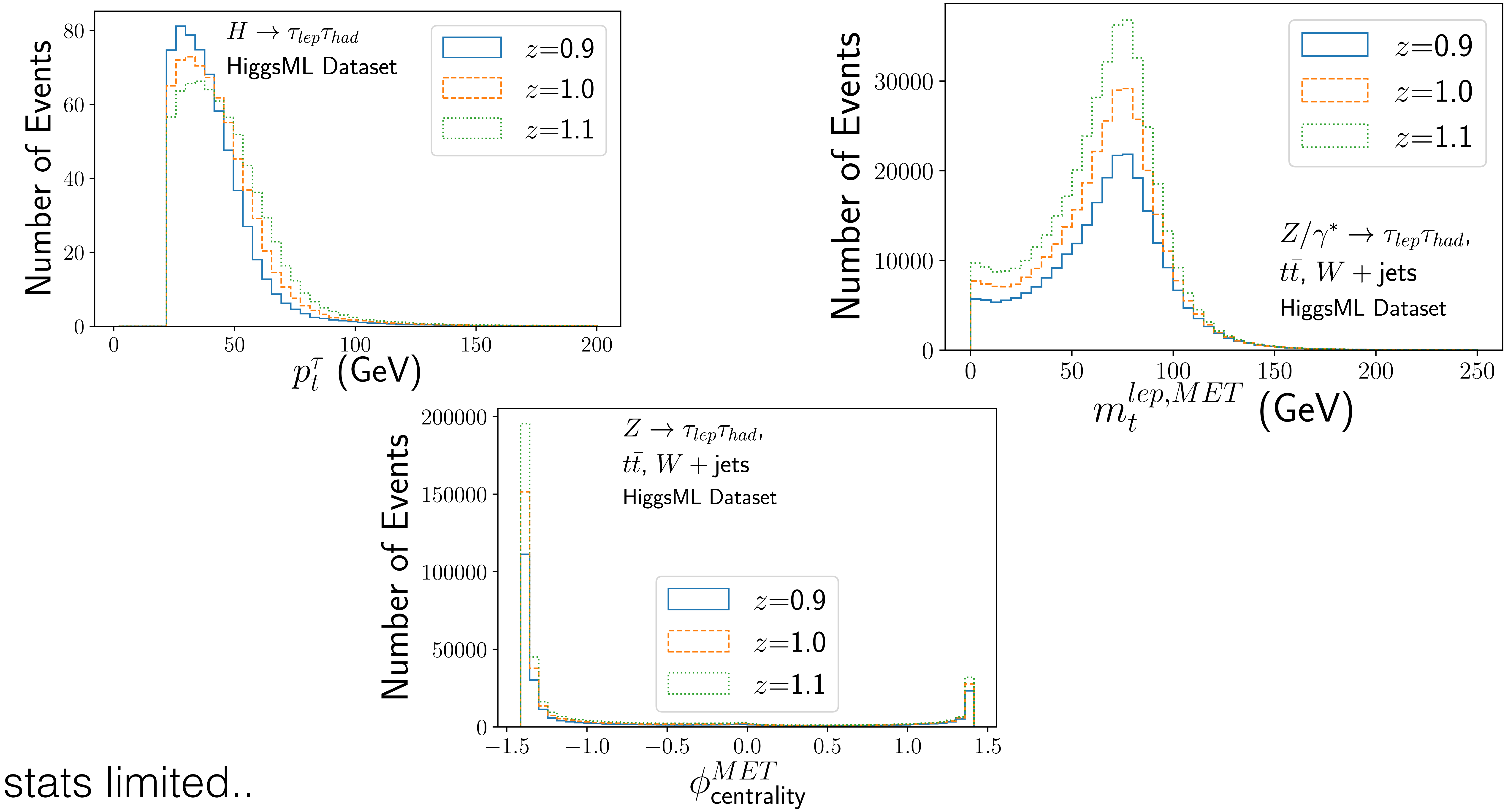
Yes!

Okay, it works on your handcrafted toy problem.
What about a real physics dataset?

HiggsML Public Dataset with Tau Energy Scale (TES) as Z

Higgs challenge
the HiggsML challenge
May to September 2014
When High Energy Physics meets Machine Learning
info to participate and compete : https://www.kaggle.com/c/higgs-boson
ATLAS EXPERIMENT
inria
kaggle
Google

Parameter of Interest is Higgs signal strength μ, and TES is the nuisance parameter Z



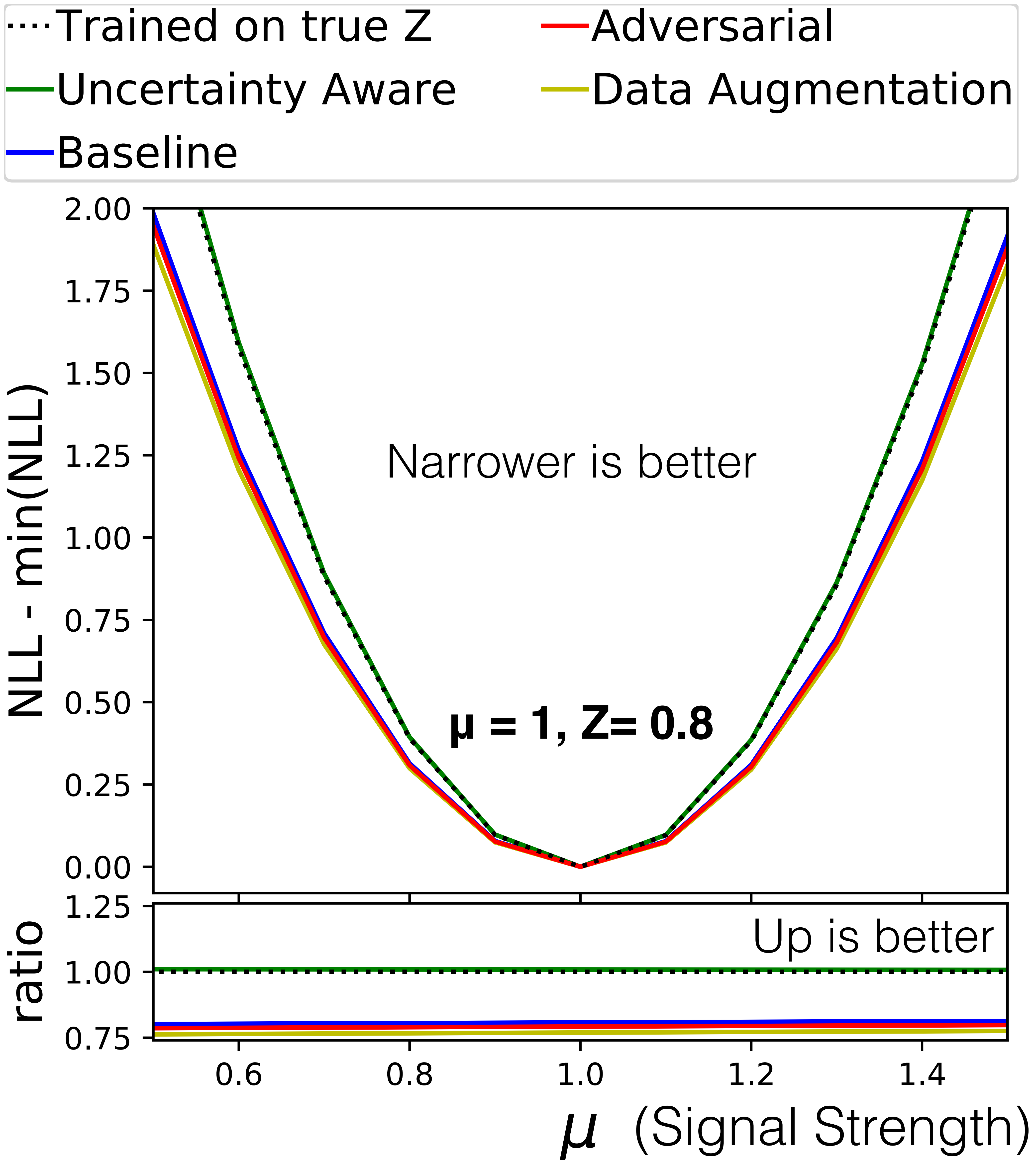
We later realised dataset isn't ideal, stats limited..

Test performance for “observed” at Systematic below Nominal

μ = 1, Z= 0.8

(Signal Strength)

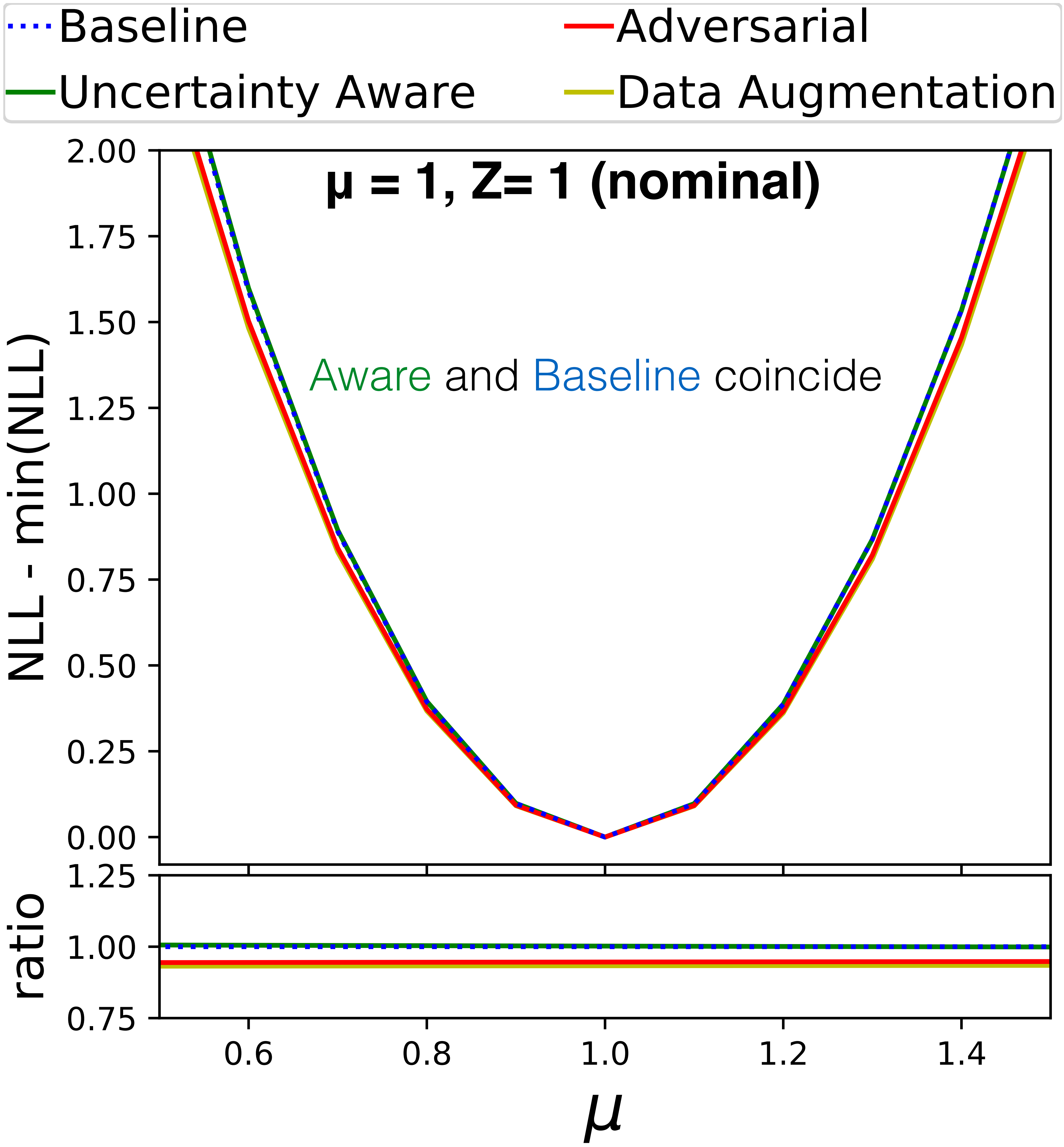
Test performance for “observed” at Systematic below Nominal

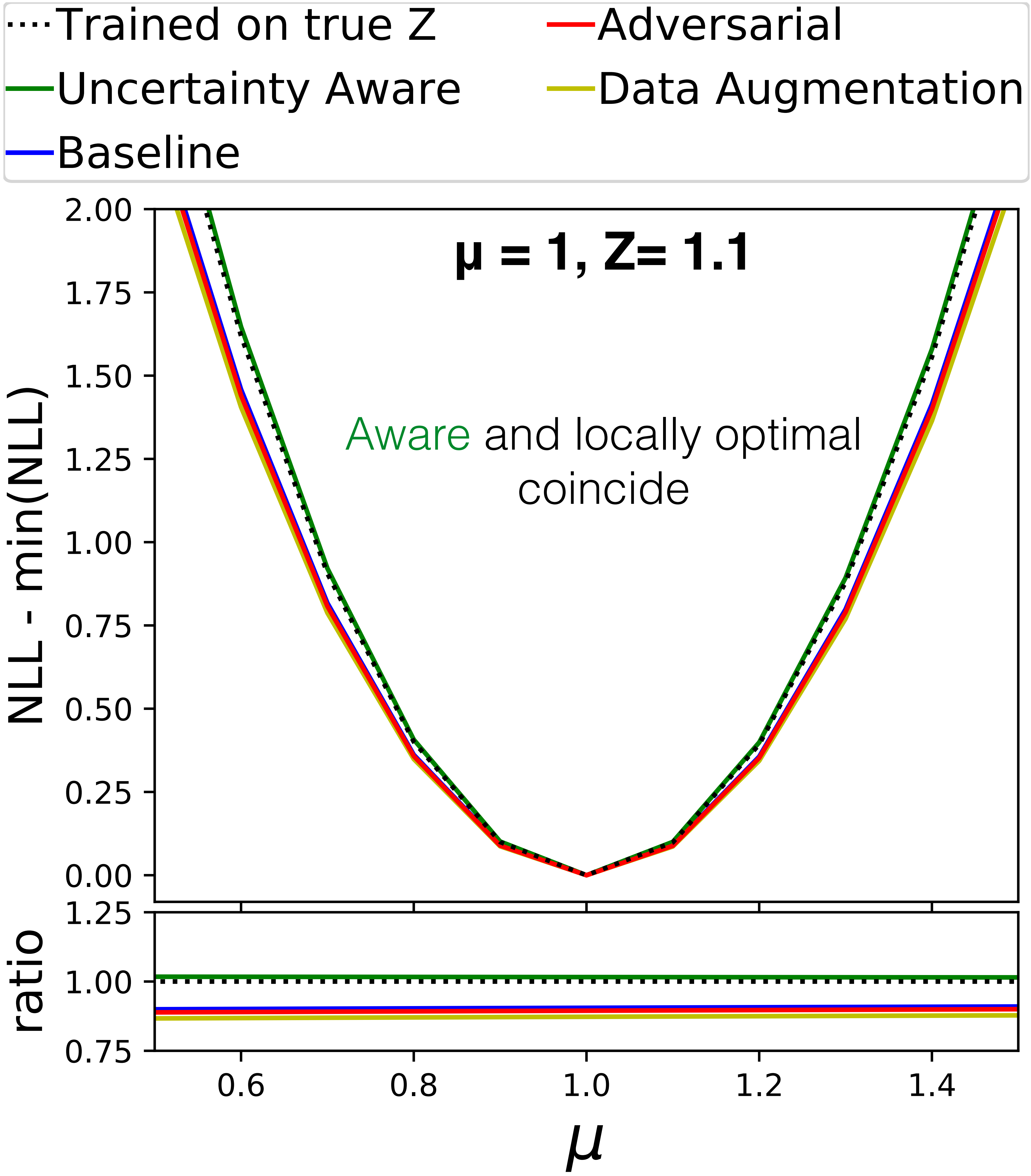


Uncertainty-Aware coincides with classifier trained on true Z

⇒ It is optimal!

Test performance for “observed” datasets at nominal and above nominal Z



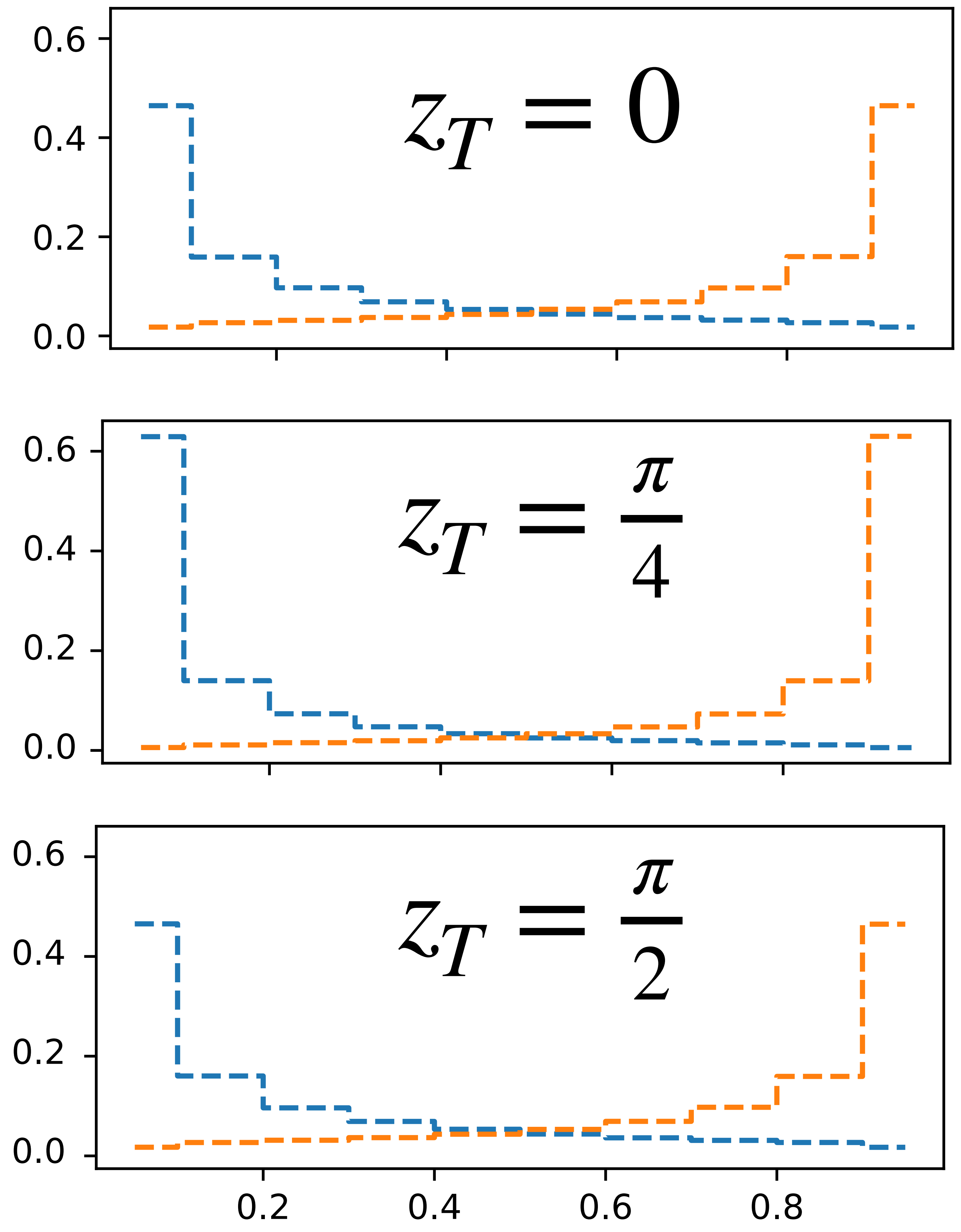


In every case the Aware Classifier is as good as the optimal one, no other technique matches its performance everywhere

Practical advantages of factorising inference

While using histogram (or KDE) templates seems clunky, it has practical advantages:

- More diagnostic tools: look at histograms, test for over-constraining of z
- Study impact of/profile over untrained nuisance parameters
- No worries about calibration of NN



Auxiliary measurement of Z instead of prior

Simplistic auxiliary measurement of z_T

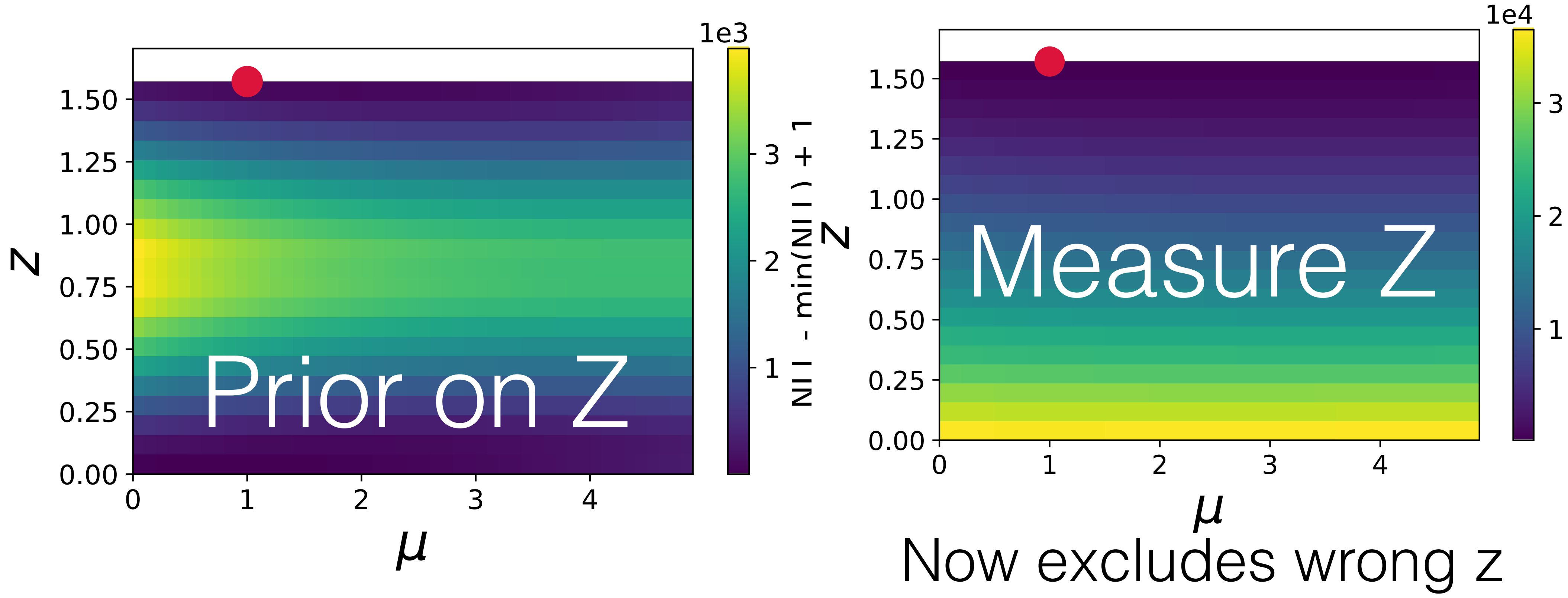
No need to re-train any network, change only in likelihood computation step

All methods provide improved limits on μ if Z is tightly constrained

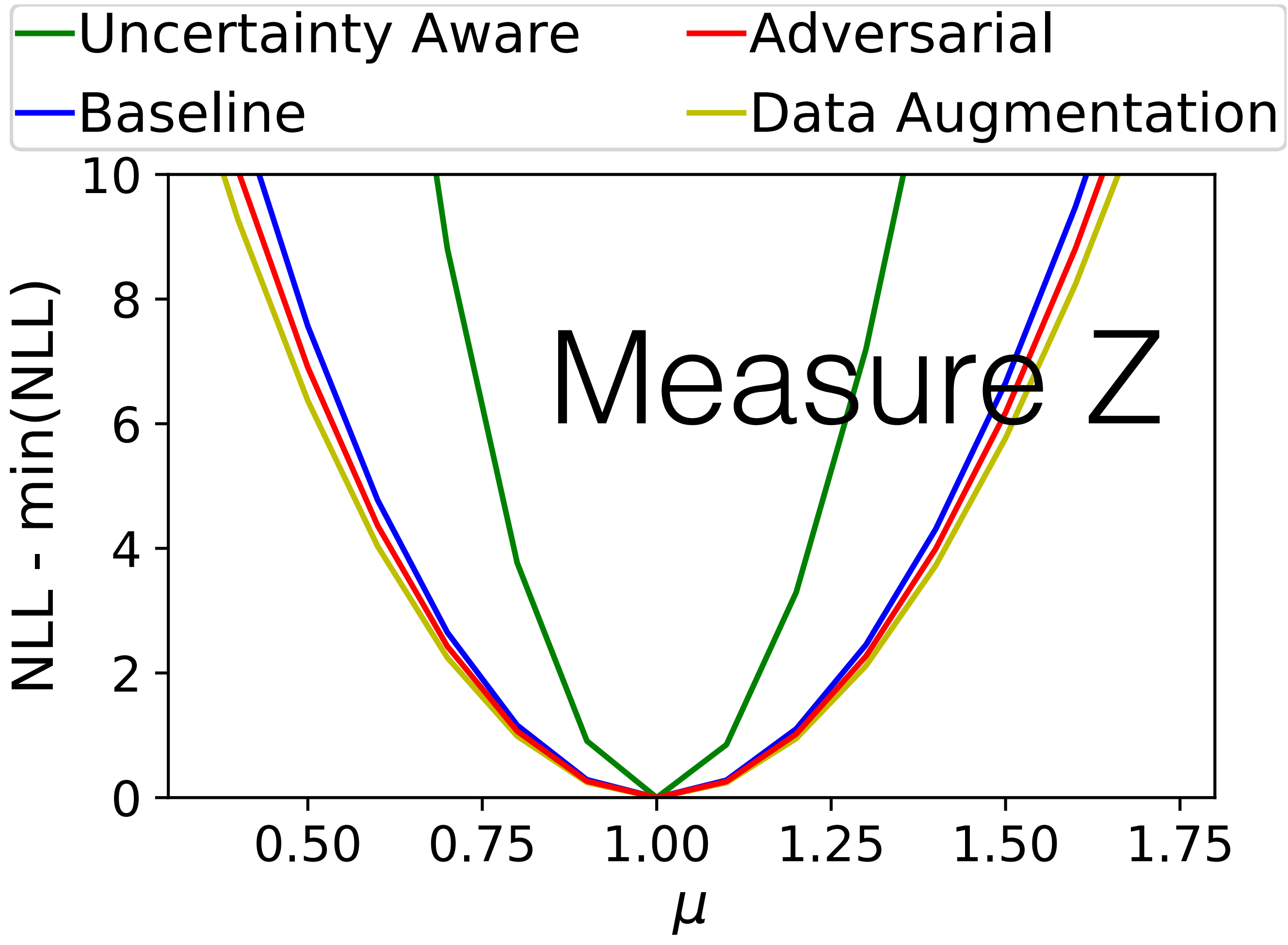
Aware classifier still best one to use

$$
\begin{aligned}
&-\log \mathcal{L}(\mu, z|\{x_i\}) \\
&\quad = -\sum_{j=1}^{n_{\text{bins}}} \left[N_j \cdot \log\left(\mu s_j + b_j\right) - \mu s_j - b_j - \log(\Gamma(N_i)) \right] \\
&\quad\quad - \sum_{k=1}^{m_{\text{bins}}} \left[N_k^{aux} \cdot \log\left(a_{\text{k}}^z\right) - a_{\text{k}}^z - \log(\Gamma(N_k^{aux})) \right],
\end{aligned}
$$

Baseline classifier trained on nominal



Now excludes wrong z confidently



(b) Data generated with $z = \frac{\pi}{2}$.

Test performance for “observed” datasets at $\mu = 2$

Trained on true Z
Adversarial
Uncertainty Aware
Data Augmentation
Baseline
Baseline
Adversarial
Uncertainty Aware
Data Augmentation
Trained on true Z
Adversarial
Uncertainty Aware
Data Augmentation
Baseline
NLL - min(NLL)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
ratio
1.25
1.00
0.75
1.6
1.8
2.0
2.2
2.4
μ

In every case the Aware Classifier is as good as the optimal one, no other technique matches its performance everywhere

Take home message

- Training a uncertainty aware classifier and profiling over the nuisance parameter provides performance similar to a locally optimal classifier

- This prescription can also handle auxiliary measurements of the nuisance parameter straightforwardly by combining the likelihoods

- Not a black-box procedure: Can also study impact of untrained systematics on sensitivity

- Solution scales to real physics dataset, easy to integrate into ATLAS/CMS chain

PhysRevD.104.056026