Faithful uncertainties in Machine Learning

Luigi Favaro

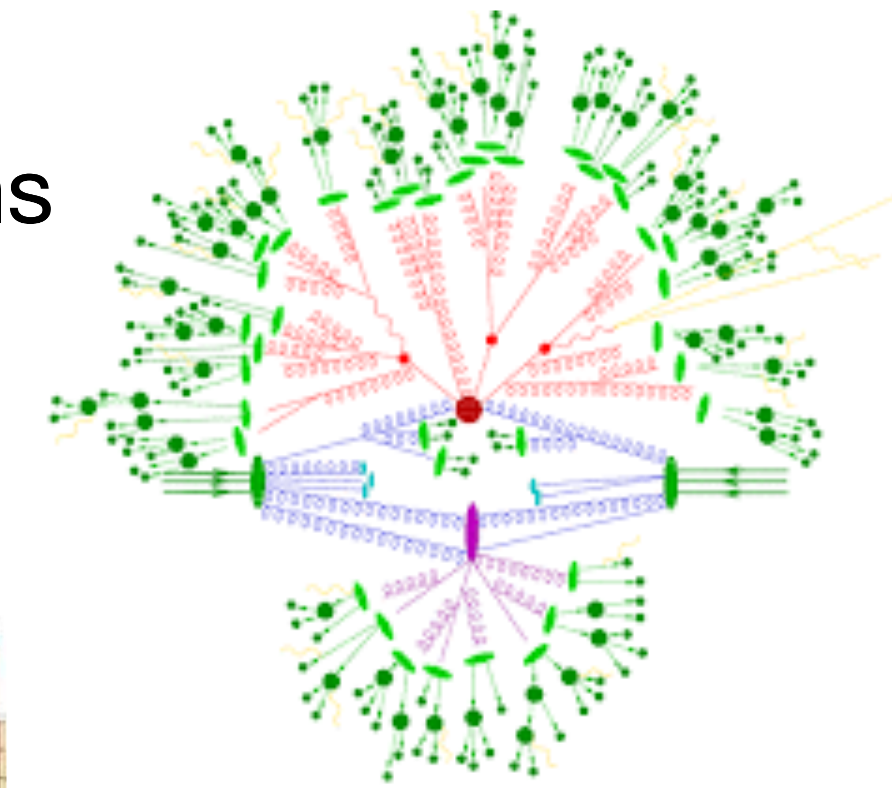
57th Rencontres de Moriond (EW) - La Thuile 23/03/2023

Universität Heidelberg
Zukunft. Seit 1386.

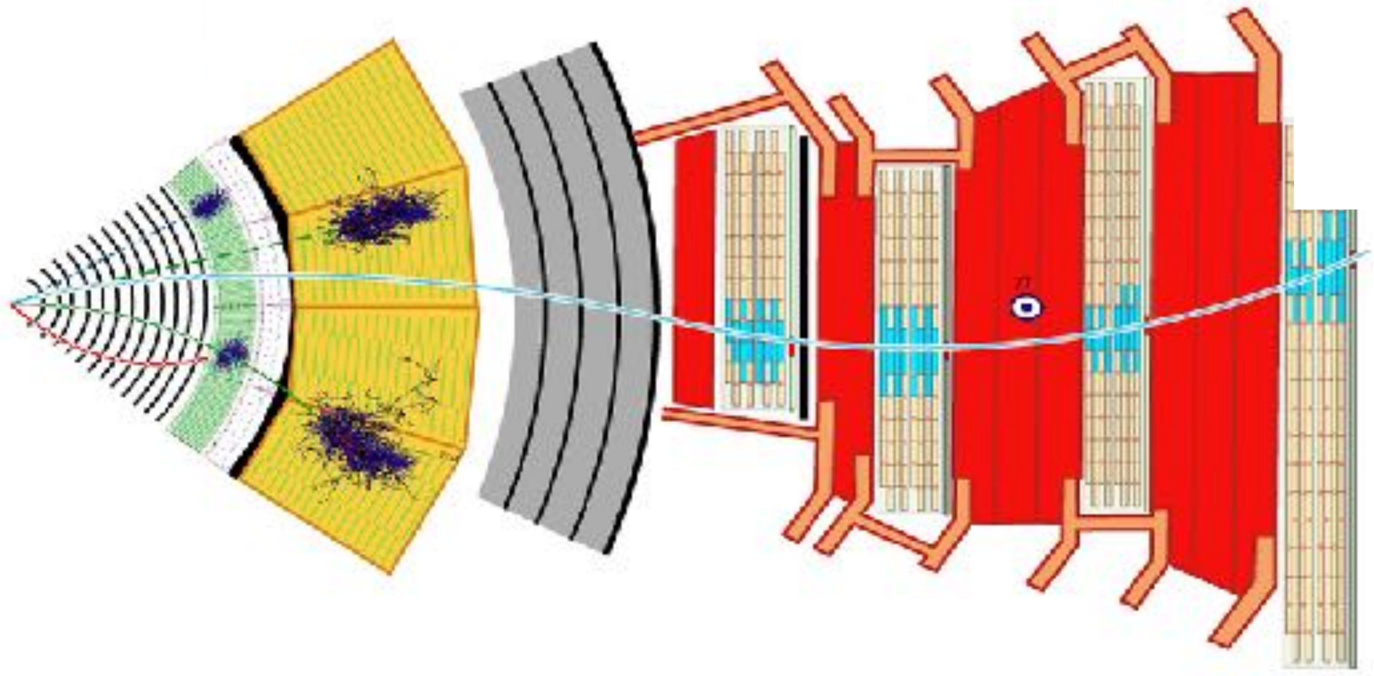
Preliminaries

As you know, analyses are challenging…

Theory predictions

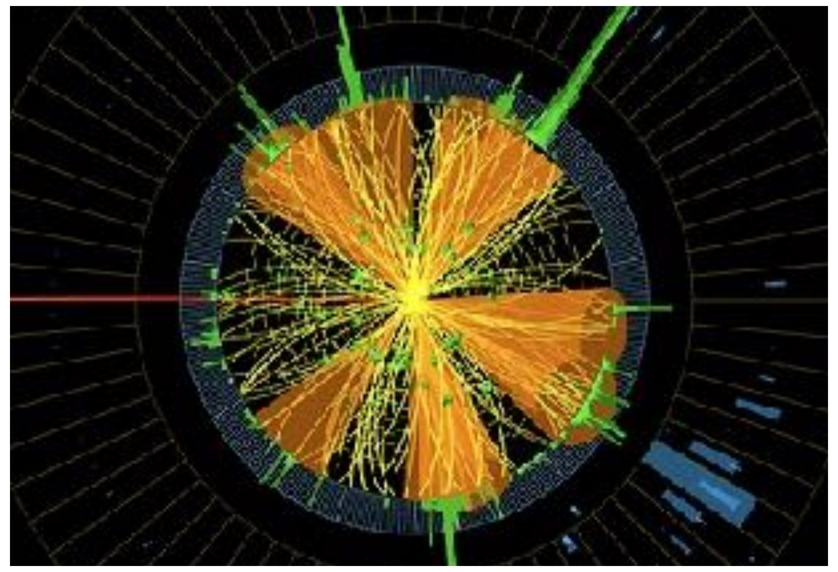


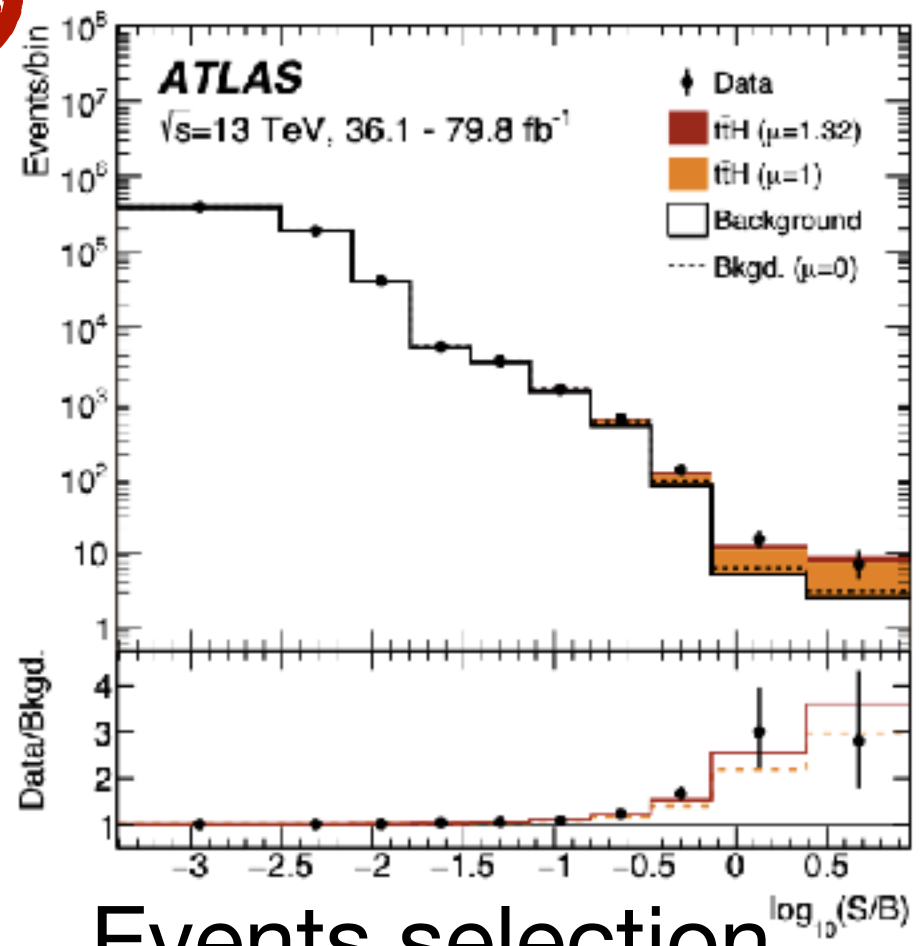
Simulations



Uncertainties?

Reconstruction



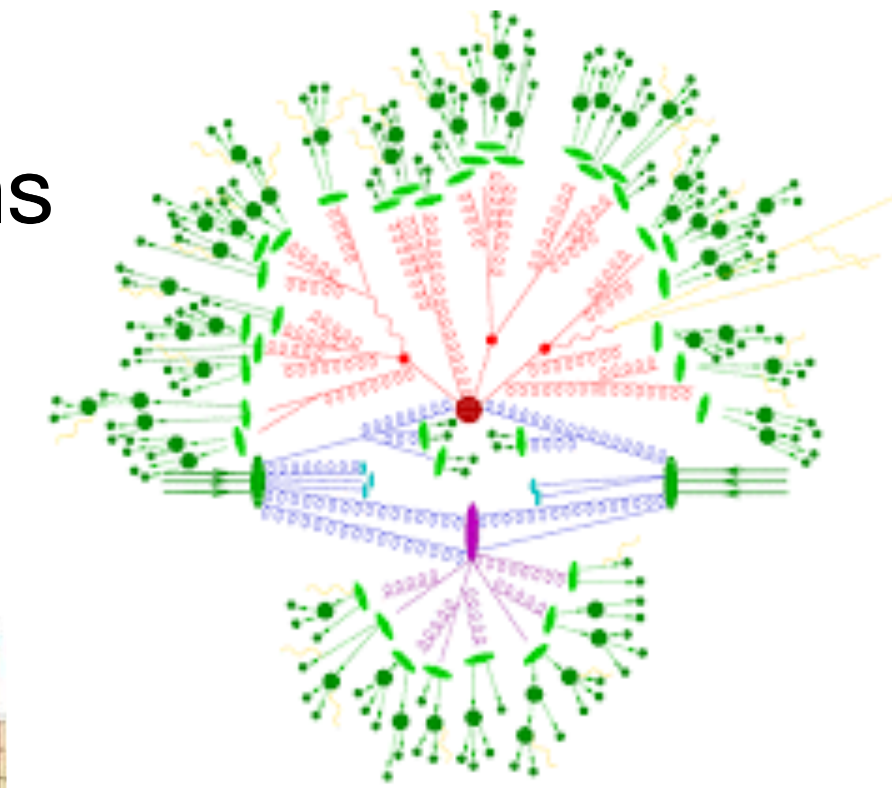


Events selection

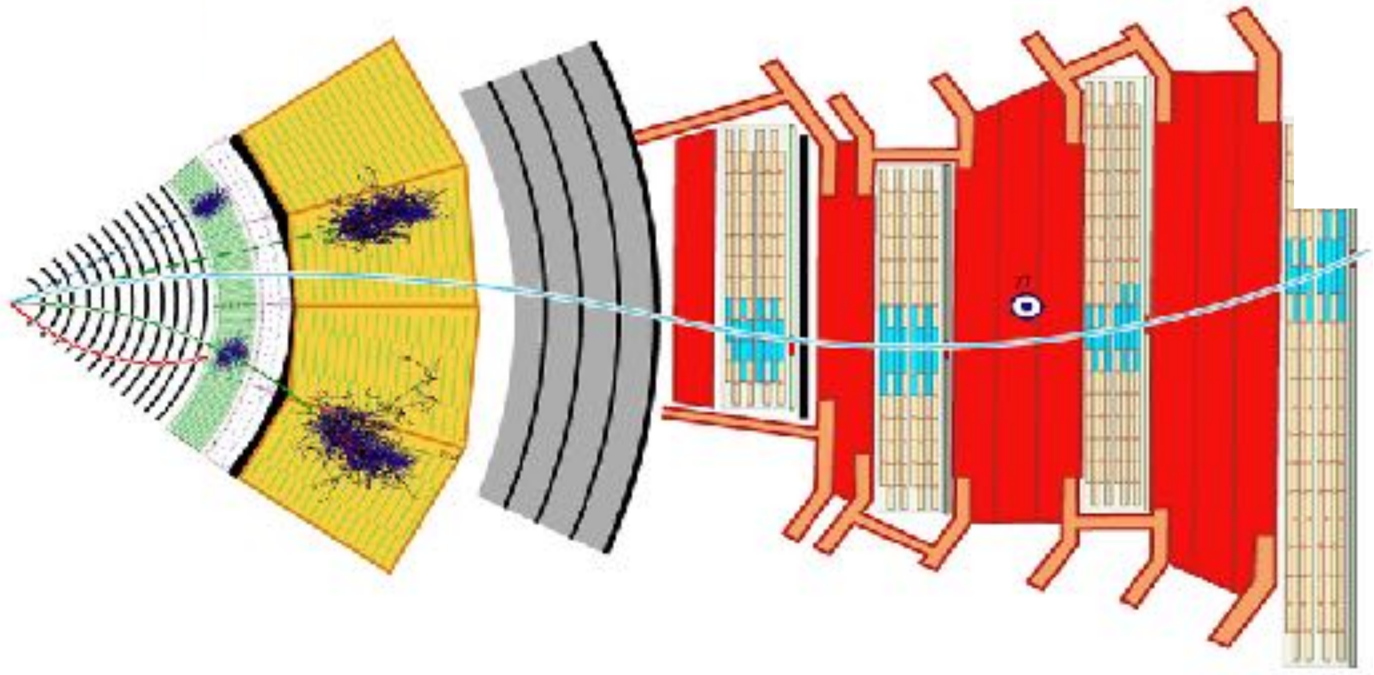
Preliminaries

As you know, analyses are challenging…

Theory predictions

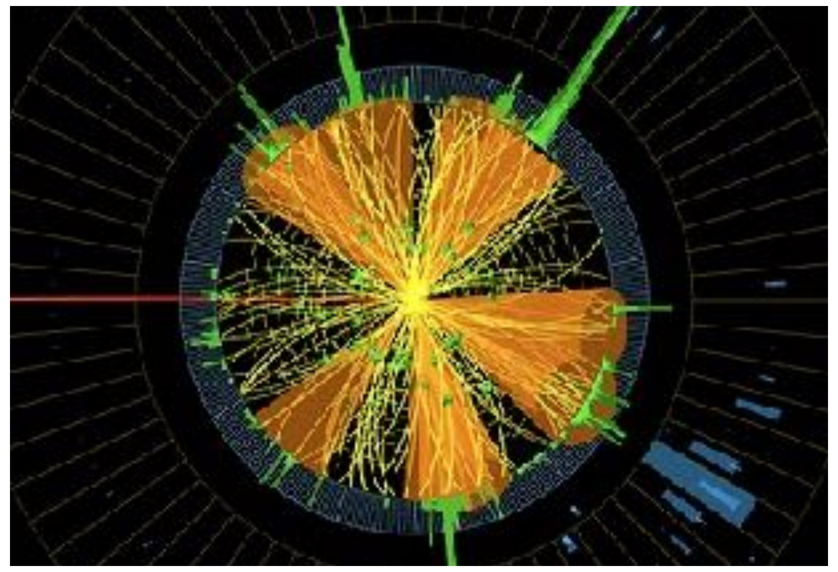


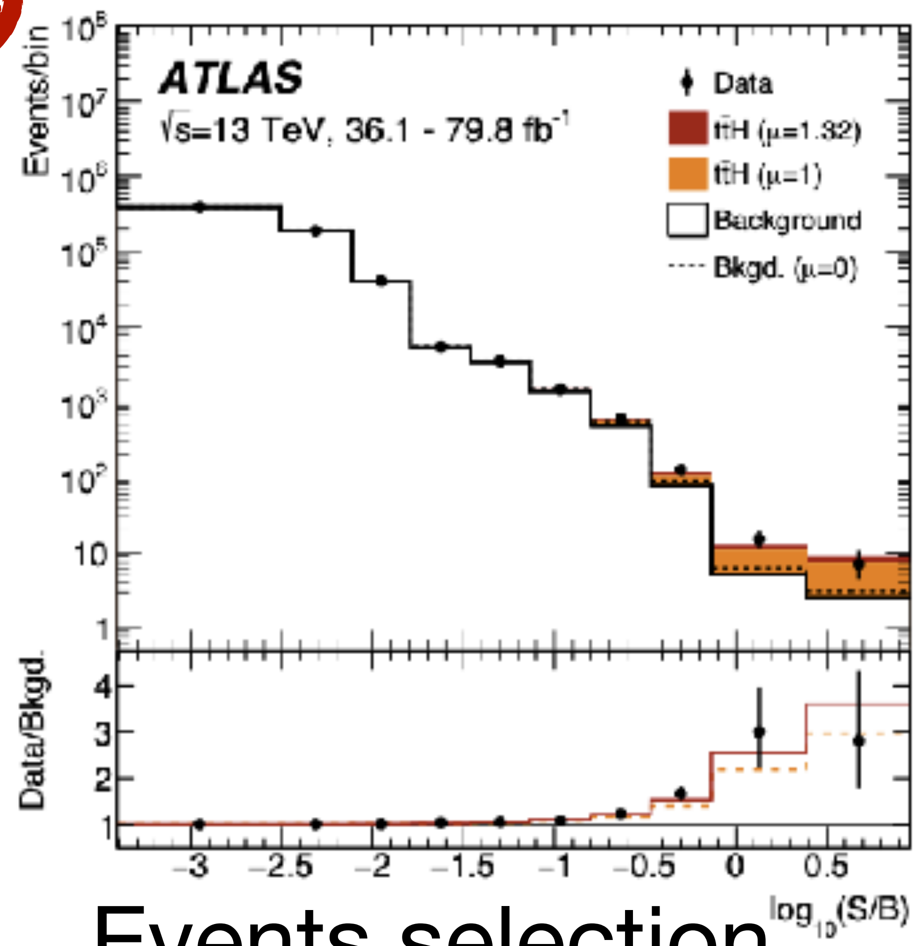
Simulations



Uncertainties?

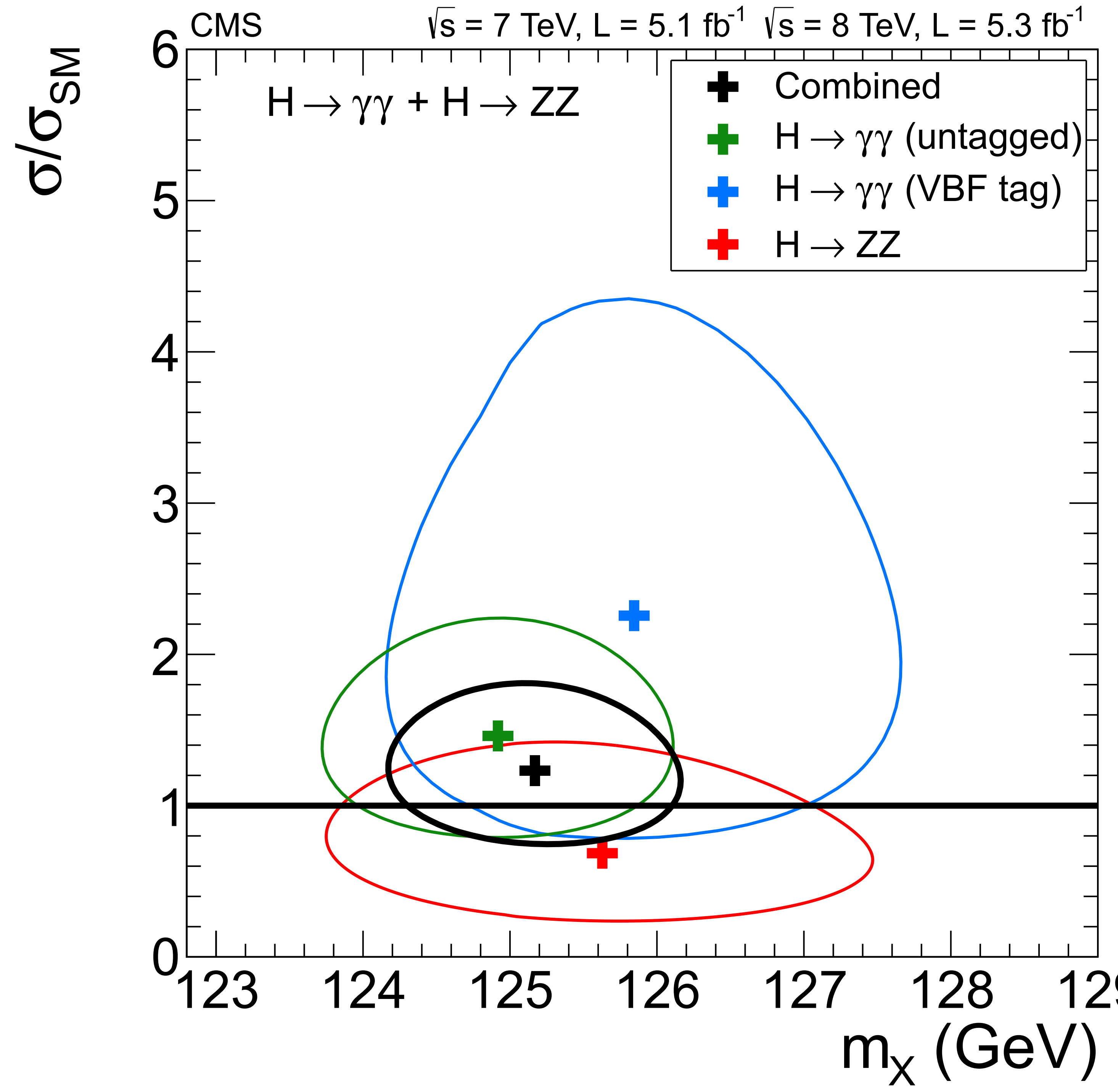
Reconstruction





Events selection

Goal



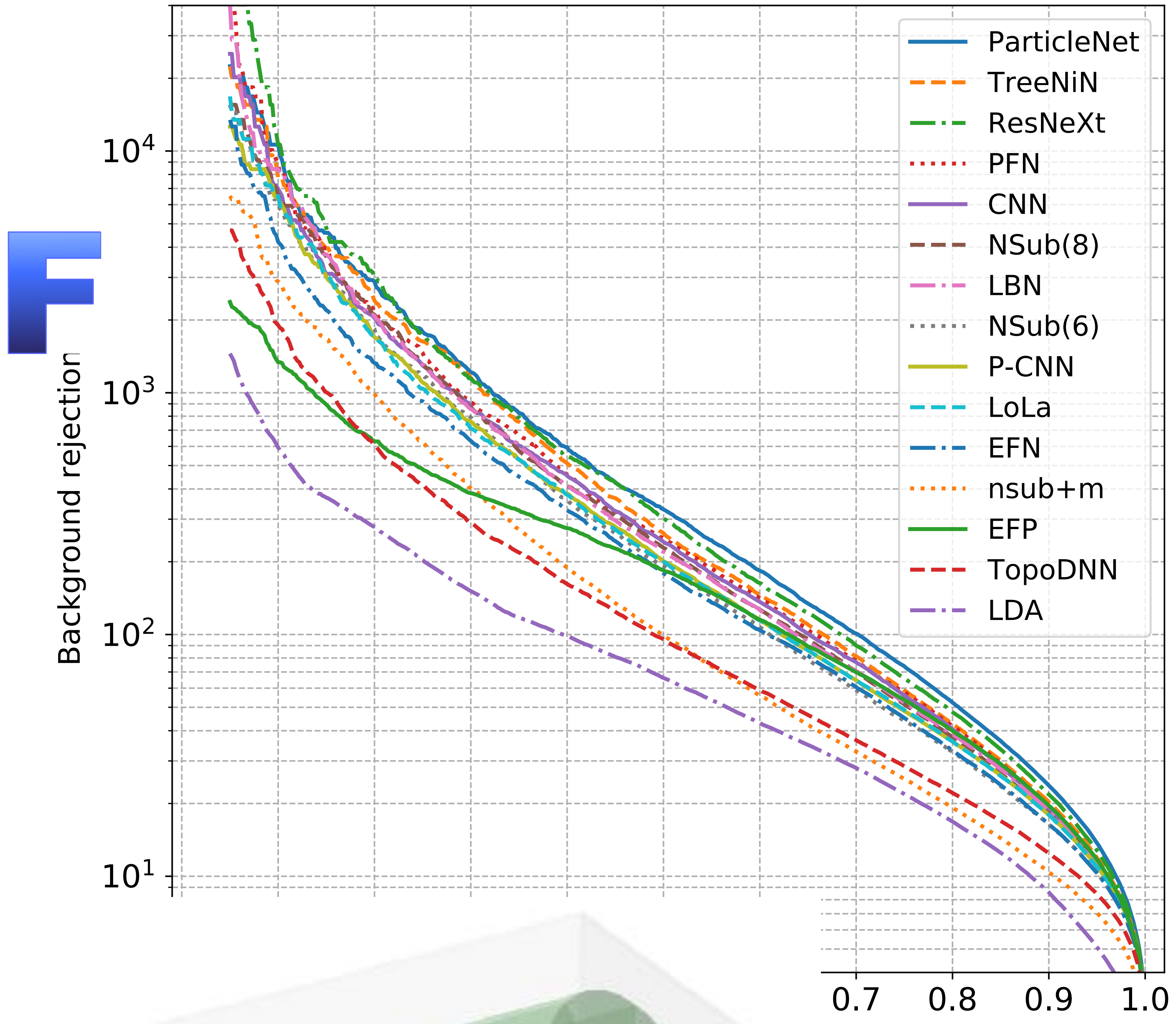
Preliminaries

Machine Learning enters in all this stages:

$f_\theta(x) \simeq g(x)$ $\quad\quad$ $p_\theta(x) \in [0,1]$ $\quad\quad$ $p_\theta(x) \simeq p_{data}(x)$

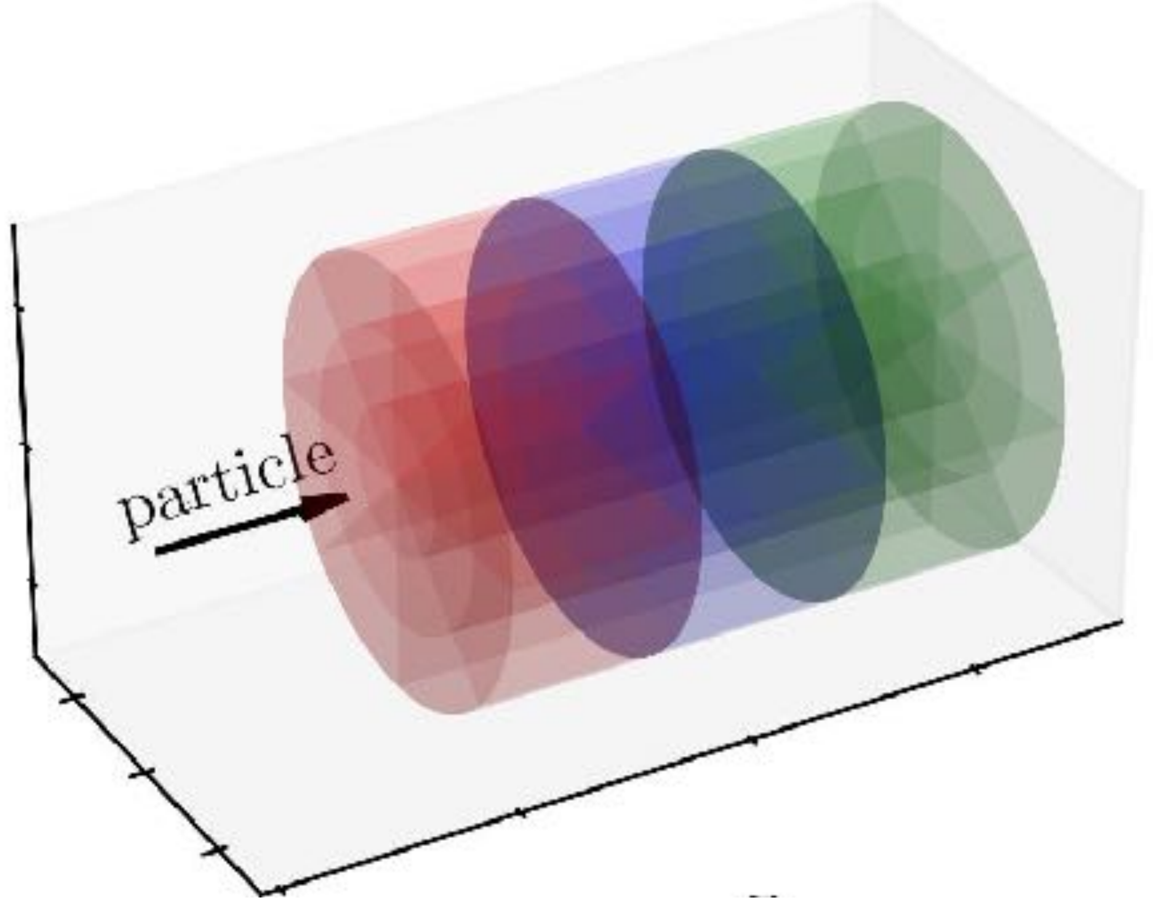
Regression $\quad$ **Classification** $\quad$ **Generation**

Highly expressive, non-linear parameterizations



Train by minimizing a loss function with Stochastic Gradient Descent

Neural Networks are fitting functions



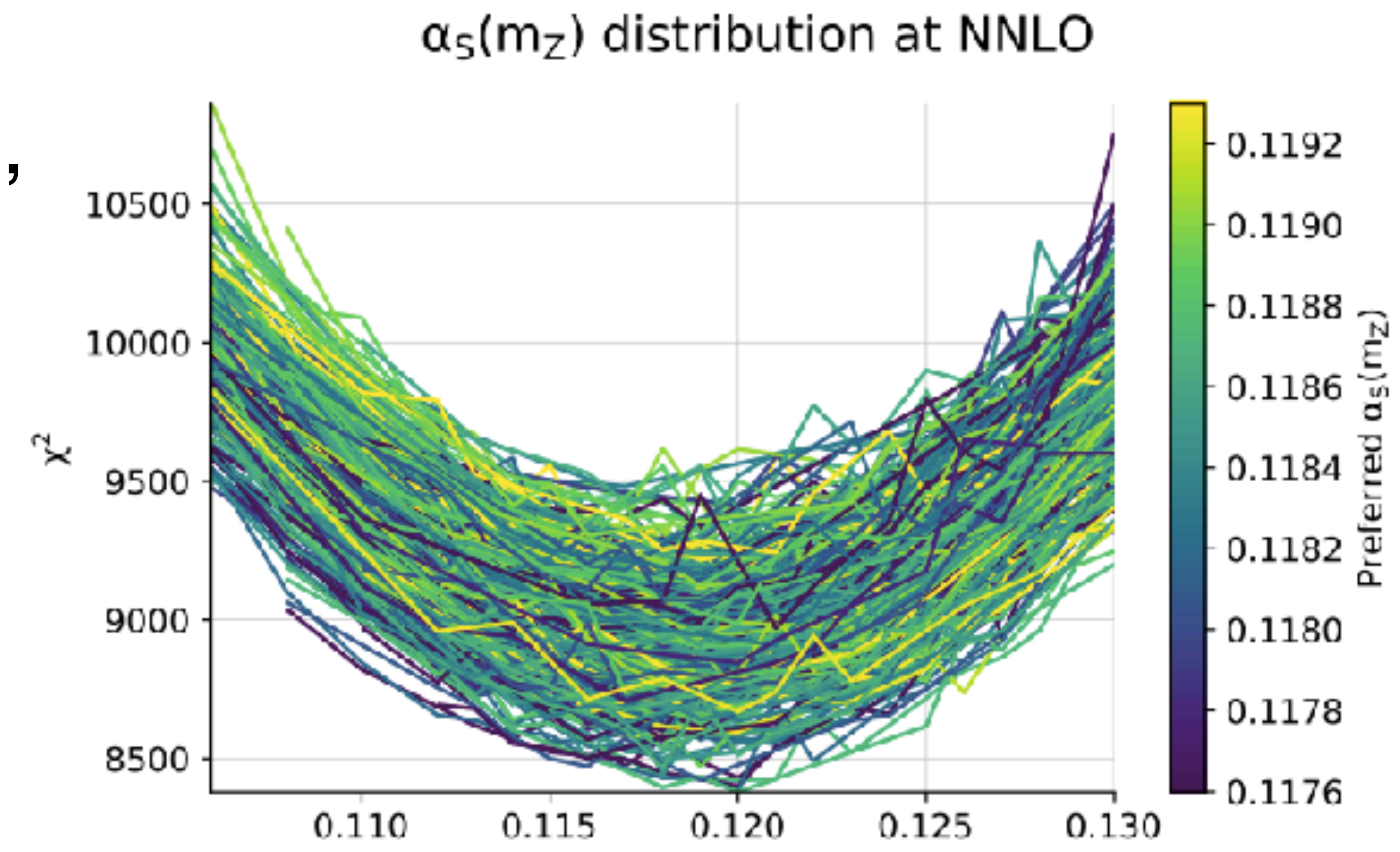
No matter the output, an error bar is needed…

Preliminaries

Model uncertainties with Machine Learning

Extract uncertainties of the model, e.g. limited training data

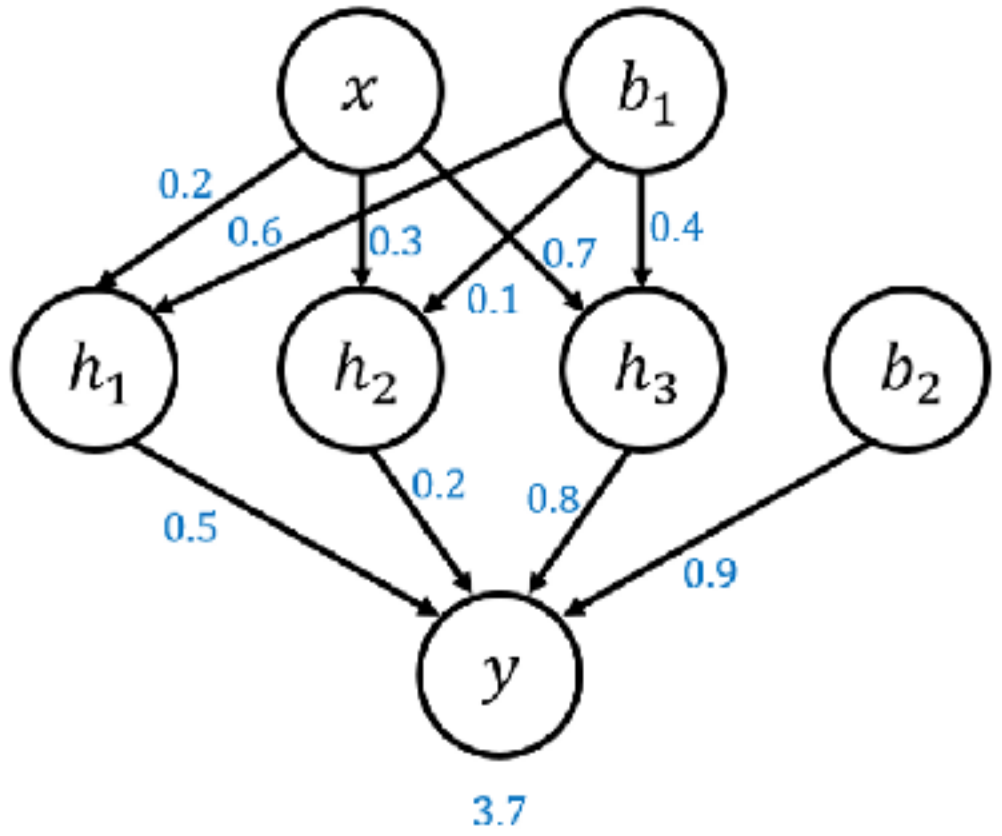
$$\sigma_{H\to\tau\tau} = 2.94 \pm 0.21 \text{ (stat) } {}^{+0.37}_{-0.32} \text{ (syst)}$$

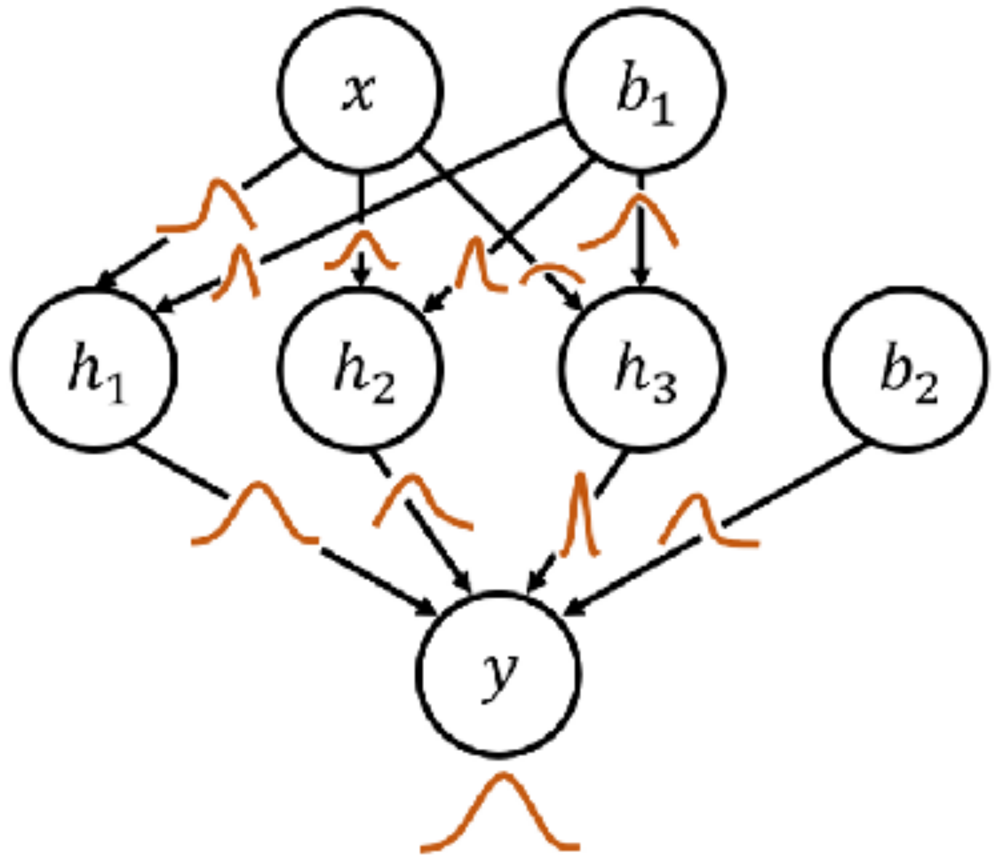


NNPDF

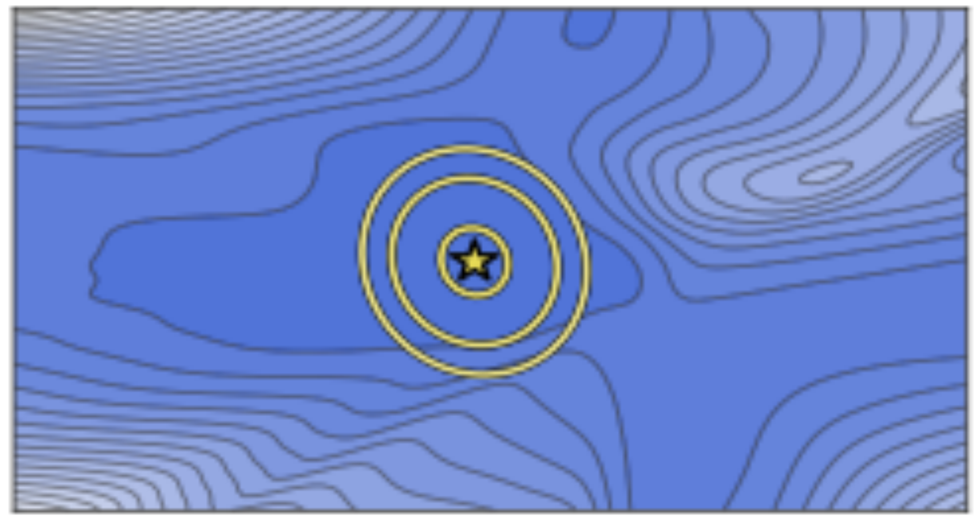
Estimating uncertainties in Machine Learning:

- Ensemble of networks
- Laplace approximation
- **Bayesian Neural Networks**





(b) Laplace Approximation



Preliminaries - BNN

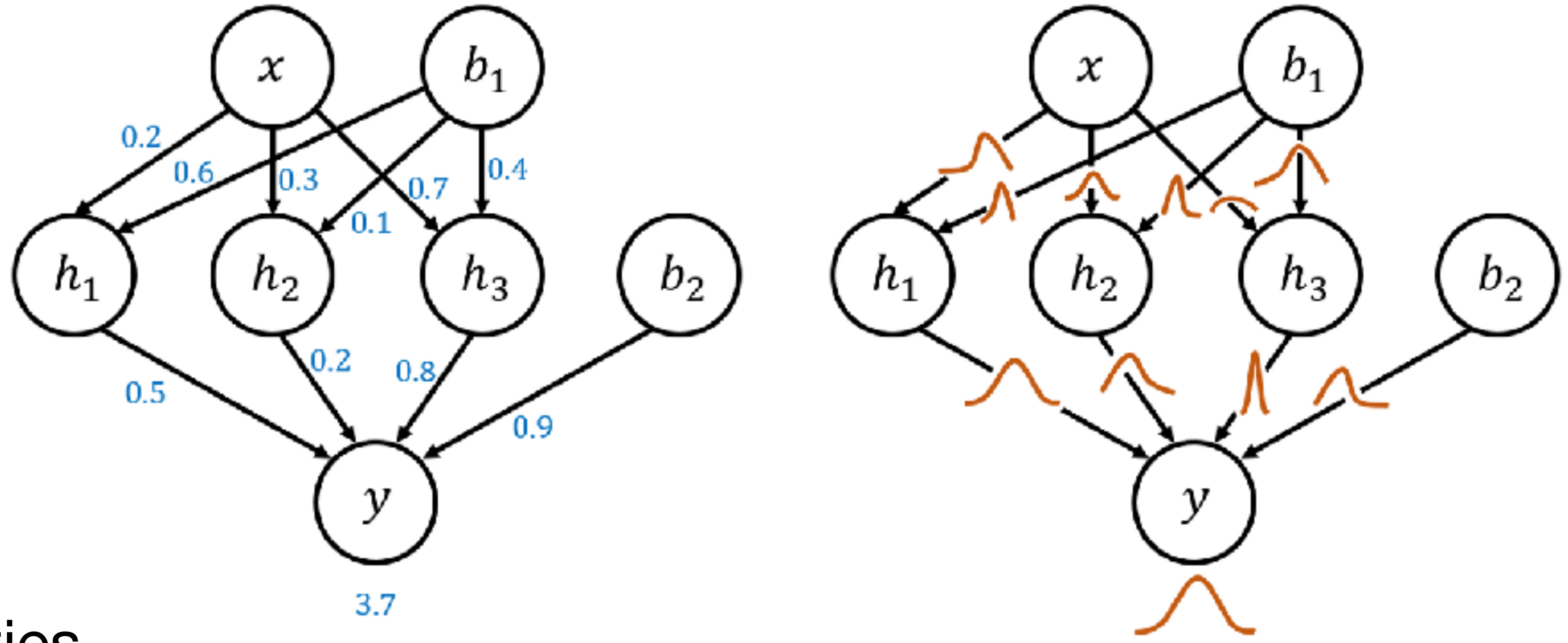
Single value per weight $\longrightarrow$ distribution

Approximate $p(\theta|x)$ then learned during training

Variational approximation:

$$p(\theta|x) \approx q_\phi(x) = \prod_i \mathcal{N}(\mu_i, \sigma_i^2)$$

Evaluation: sample from $p(\theta|x)$ and estimate uncertainties



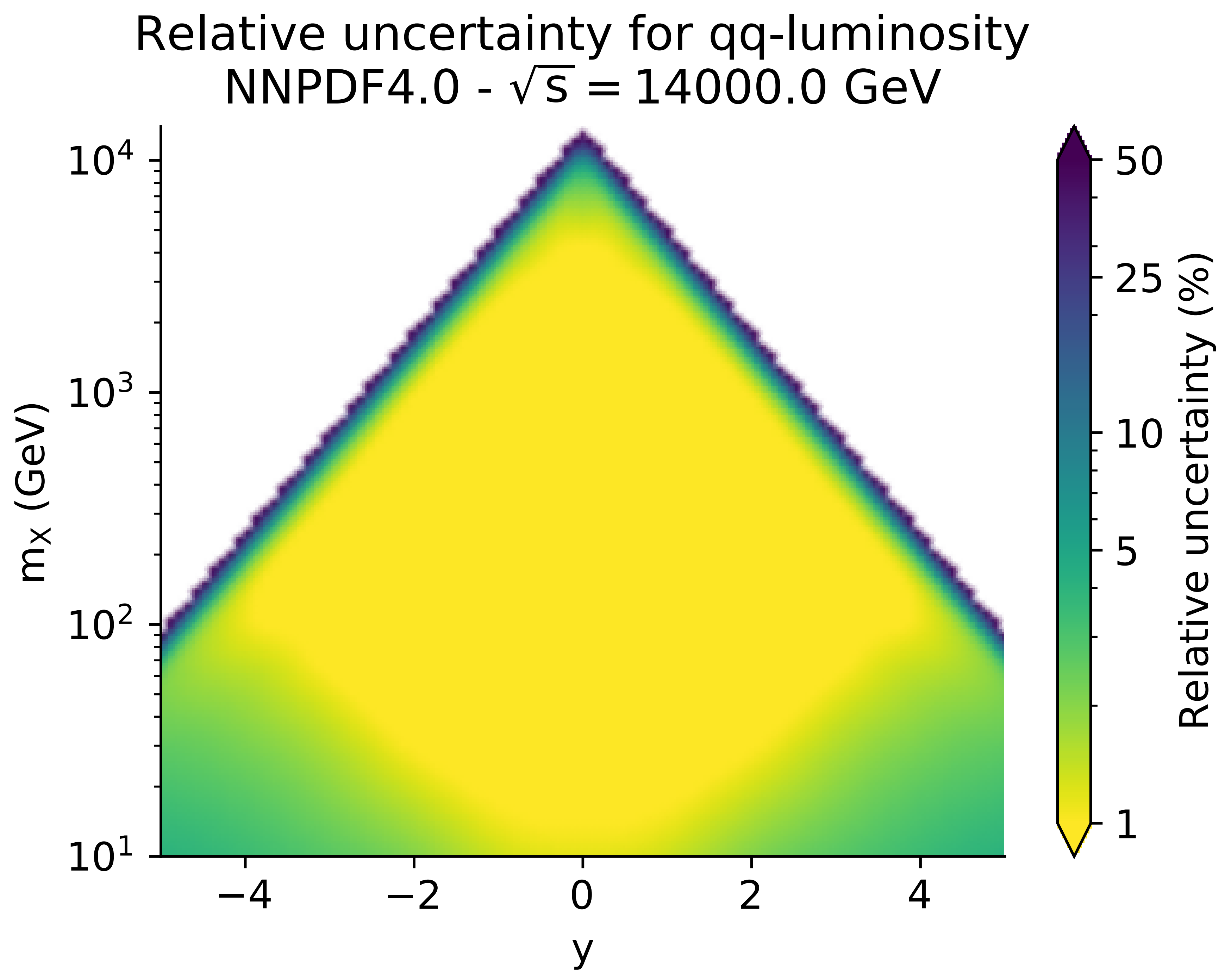
Pros:

- active learning
- fast posterior sampling
- uncertainties estimation from NN
- built-in regularization

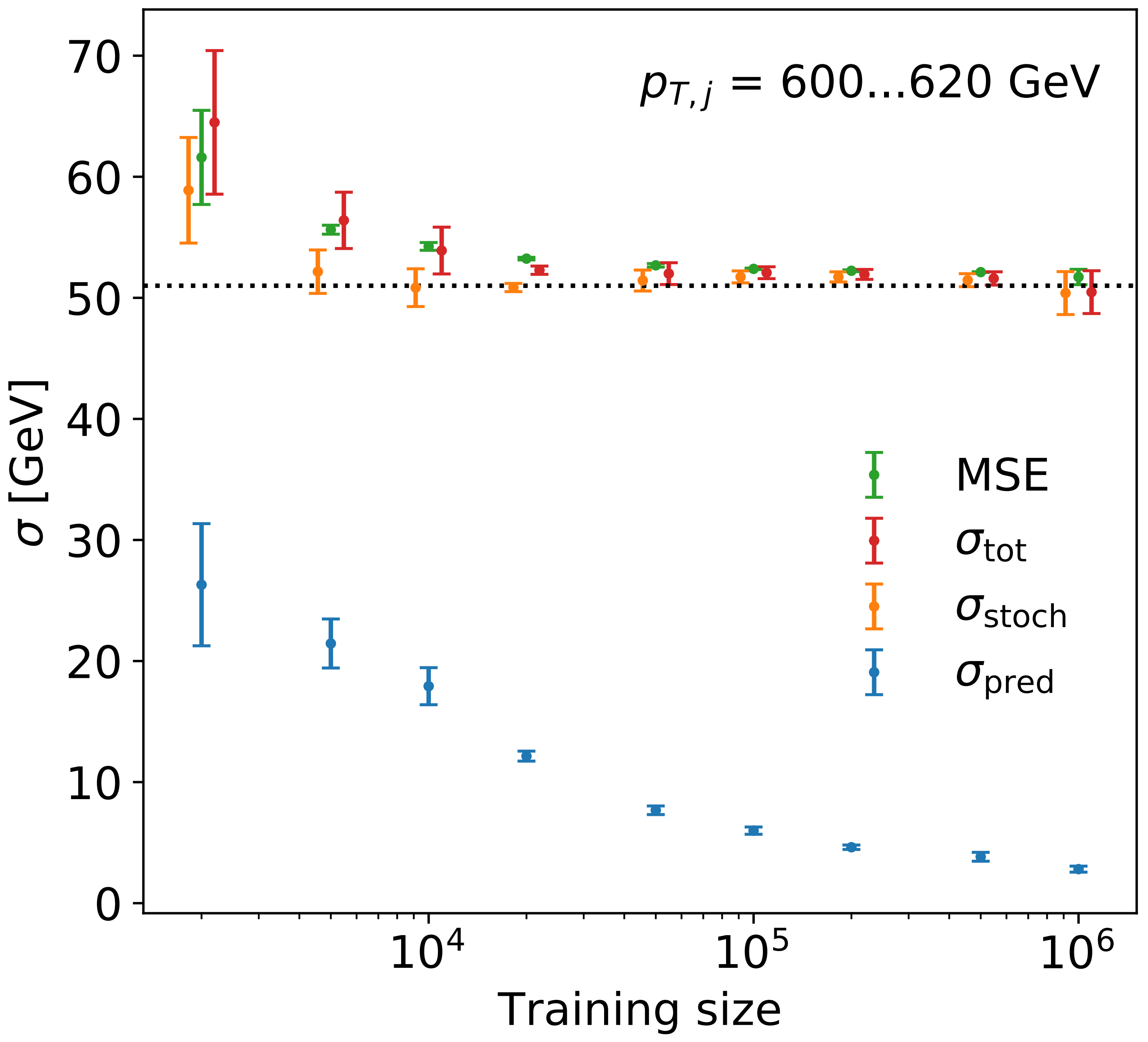
Cons:

- twice the training time
- twice # parameters
- local in loss landscape

Regression



Relative uncertainty for qq-luminosity
NNPDF4.0 - √s = 14000.0 GeV
m_X (GeV)
y
Relative uncertainty (%)
50
25
10
5
1



p_{T,j} = 600...620 GeV
σ [GeV]
Training size
MSE
σ_tot
σ_stoch
σ_pred

Perfect regression example

NNPDF collaboration, arXiv:2109.02653

Leading example: **NNPDF**

Neural Networks

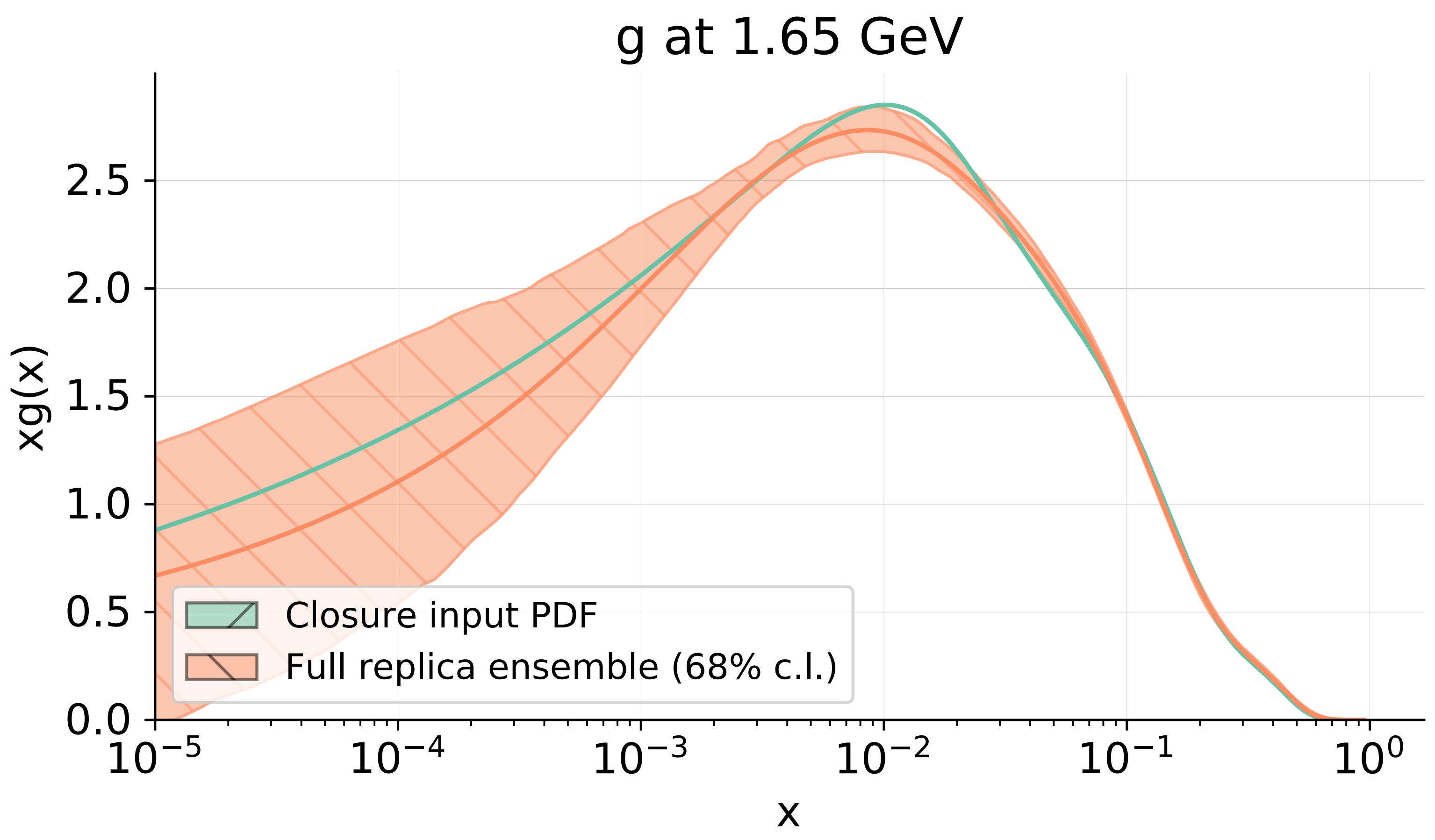
PDF fits

Closure test of the fit:

Test fitting procedure on data with known statistical properties

Future test, validate uncertainties in subsets of data

- extrapolation (pre-HERA and pre-LHC)
- interpolation (NNPDF3.1 - 4.0)



	pre-HERA	pre-LHC
	χ^2_{exp} $(\chi^2_{exp+pdf})$	
HERA	*27.20* (**1.23**)	1.22
DY (Tevatron)	*5.52* (**1.02**)	0.99
DY (LHC)	*18.91* (**1.31**)	*2.63* (**1.58**)
Others	*20.01* (**1.06**)	*1.30* (**0.87**)
	2.69 (**0.98**)	*2.12* (**1.10**)
	19.48 (**1.16**)	*2.10* (**1.15**)

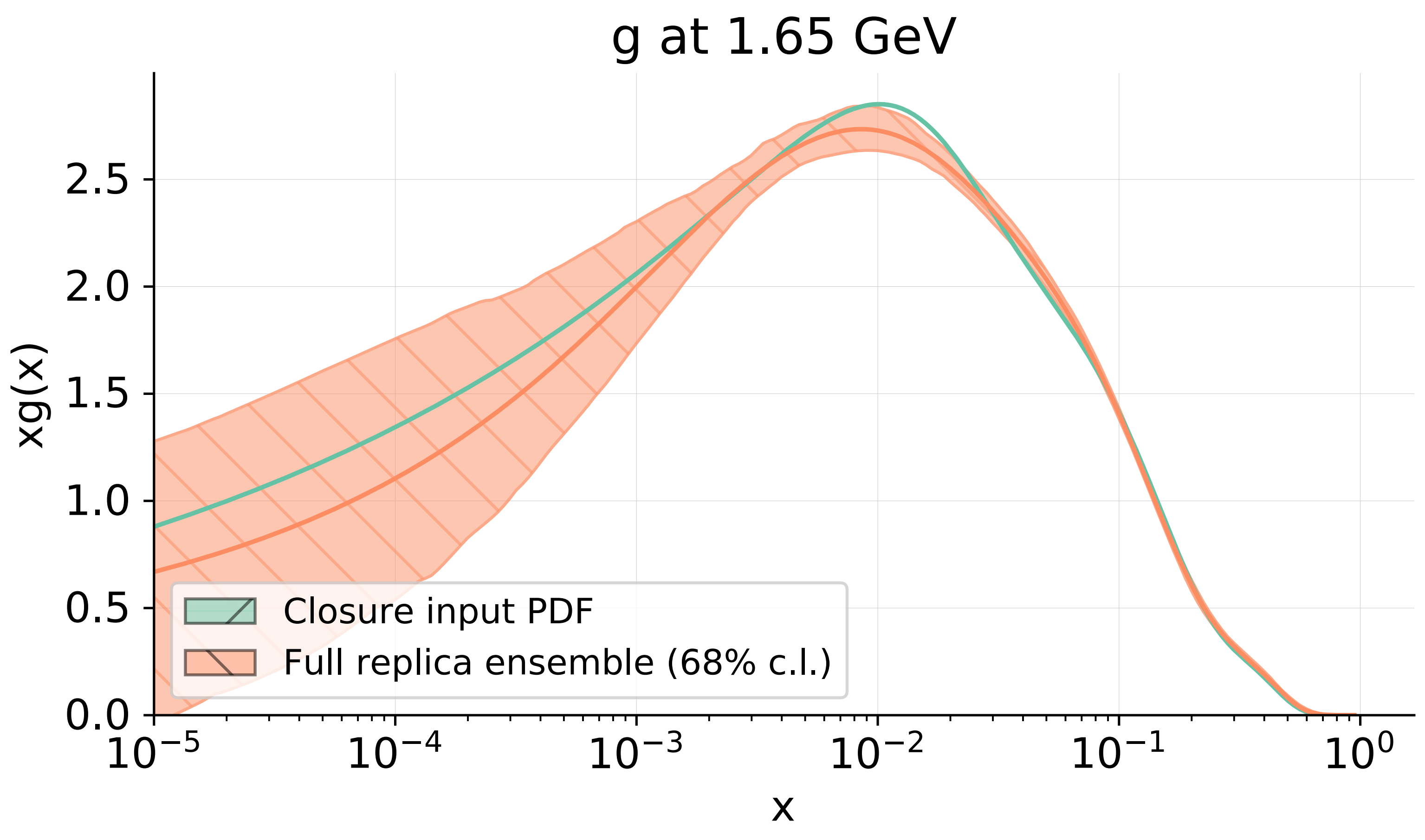
Perfect regression example

NNPDF collaboration, arXiv:2109.02653

Leading example: **NNPDF** ← Neural Networks ← PDF fits

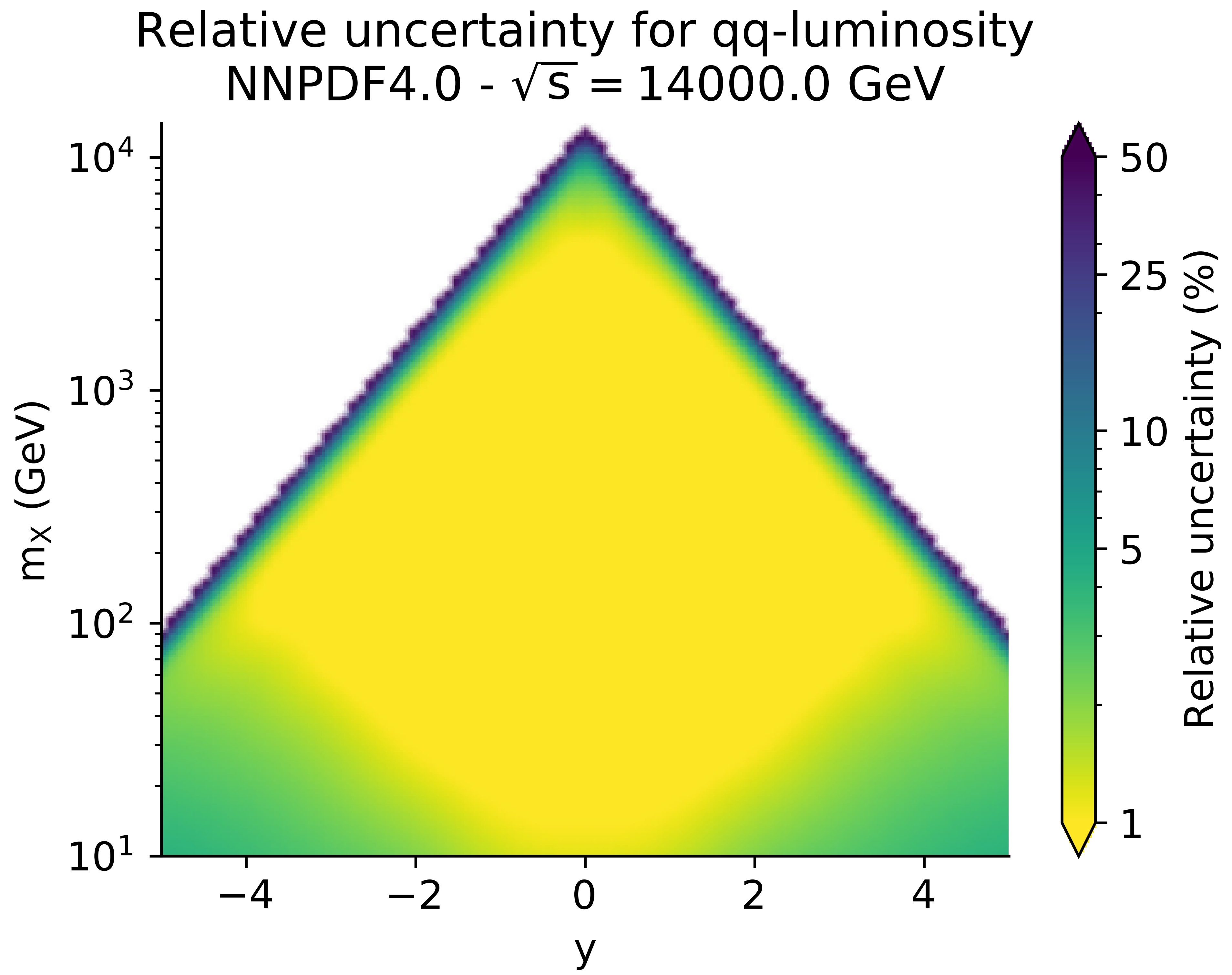
Closure test of the fit:
Test fitting procedure on data with known statistical properties

Future test, validate uncertainties in subsets of data
- extrapolation (pre-HERA and pre-LHC)
- interpolation (NNPDF3.1 - 4.0)



χ^2_{exp} ($\chi^2_{exp+pdf}$)

	pre-HERA	pre-LHC
HERA	*27.20* (**1.23**)	1.22
DY (Tevatron)	*5.52* (**1.02**)	0.99
DY (LHC)	*18.91* (**1.31**)	*2.63* (**1.58**)
Others	*20.01* (**1.06**)	*1.30* (**0.87**)
Others	*2.69* (**0.98**)	*2.12* (**1.10**)
Others	*19.48* (**1.16**)	*2.10* (**1.15**)



Path to % uncertainties on PDFs!

ML(E) for Jet energy reco

Gambhir R., Nachman B., Thaler J., arXiv:2205.03413
Kasieczka G., Plehn T. et al., arXiv:2003.11099

Use a Neural Network to perform an analyses

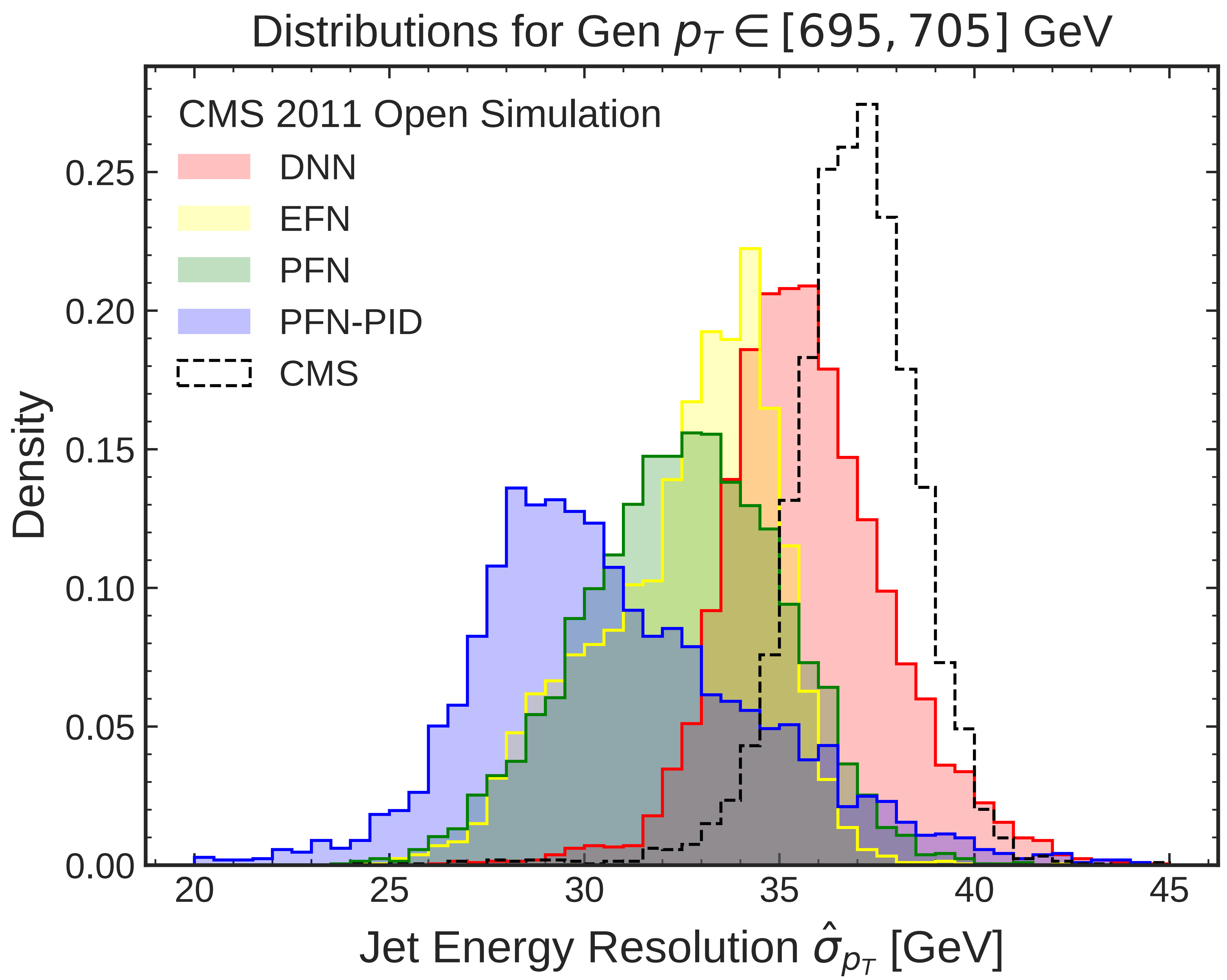
Input X: Jet observables POI Y: $p_{T,GEN}$

Use a set of Neural Networks $T(x)$ to approximate:

- conditional likelihood ratio
- prior-independent MLE estimate
- Gaussian uncertainties

$$T(x) = \frac{p(x\,|\,y)}{p(x)} \quad \hat{y}(x) = \underset{y}{\operatorname{argmax}}\ T(x,y) \quad [\hat{\sigma}_y]_{ij} = \frac{\partial^2 T(x,y)}{\partial y_i \partial y_j}$$

Improve Jet Energy Resolution!



ML(E) for Jet energy reco

Gambhir R., Nachman B., Thaler J., arXiv:2205.03413
Kasieczka G., Plehn T. et al., arXiv:2003.11099

Use a Neural Network to perform an analyses

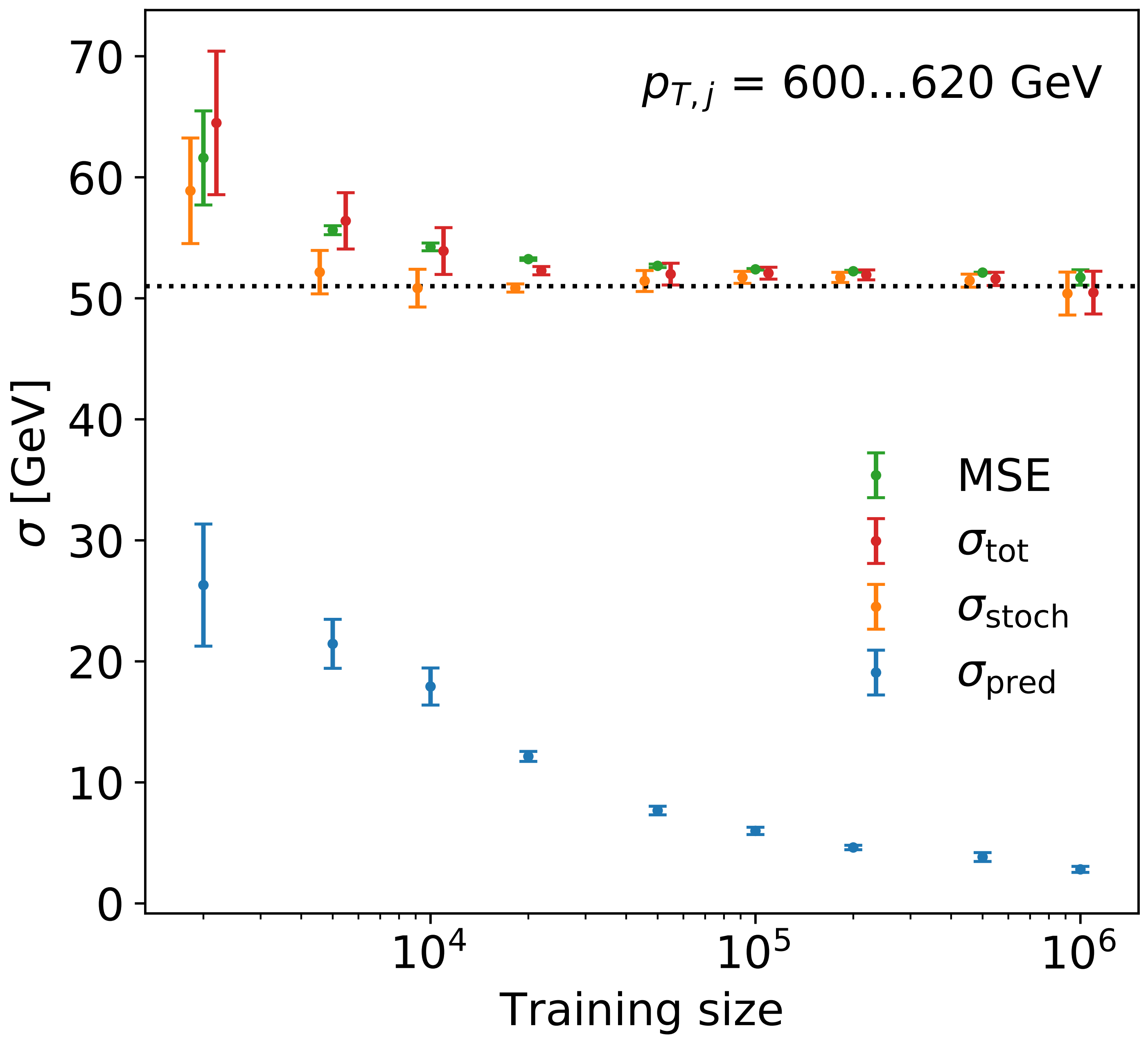
Input X: Jet observables POI Z: $p_{T,GEN}$

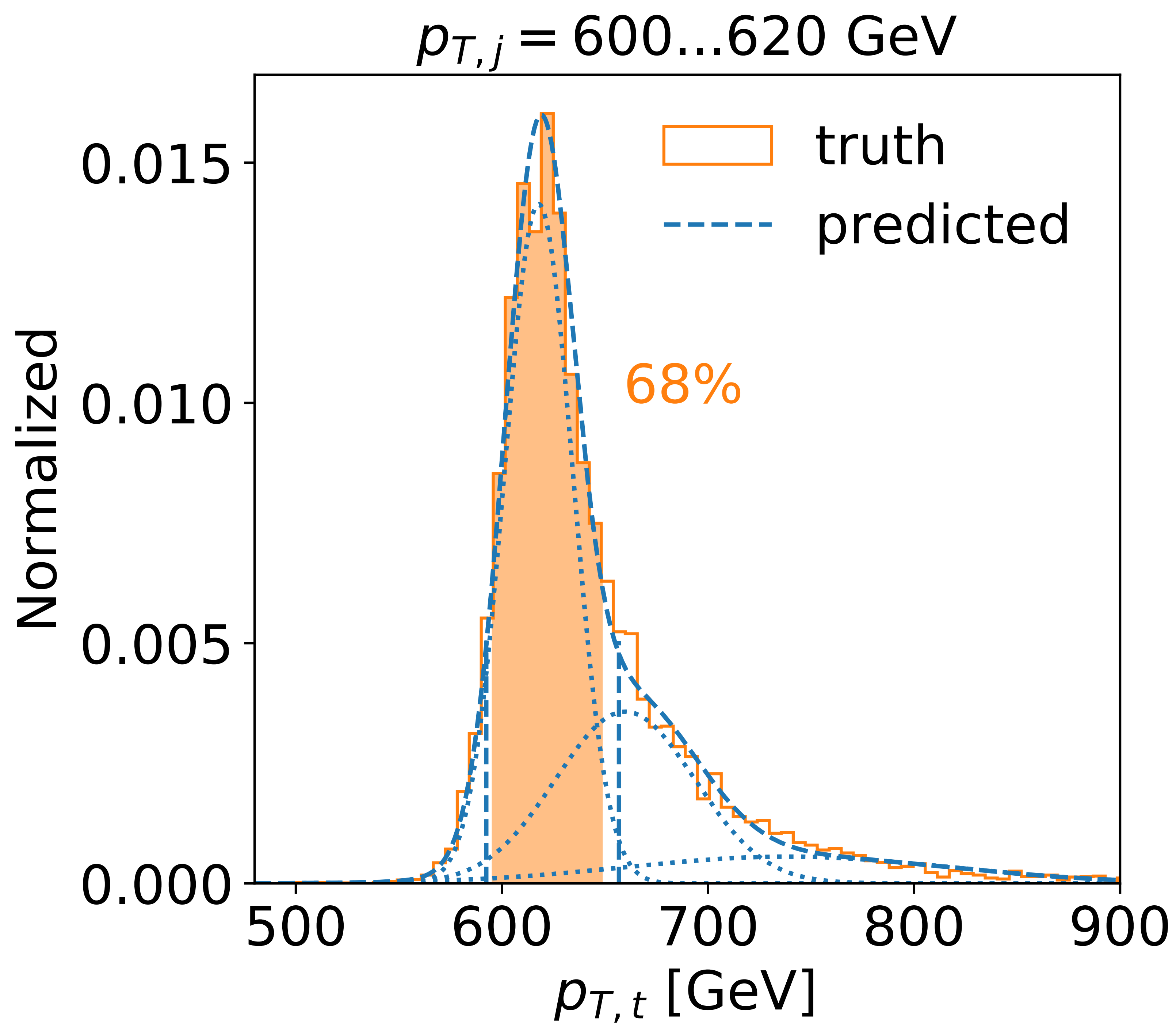
Use a set of Neural Networks $T(x)$ to approximate:

- conditional likelihood ratio
- prior-independent MLE estimate
- Gaussian uncertainties

Possible with BNNs as well:

- non-Gaussian uncertainties
- prior (in)dependence





ML(E) for Jet energy reco

Gambhir R., Nachman B., Thaler J., arXiv:2205.03413
Kasieczka G., Plehn T. et al., arXiv:2003.11099

Use a Neural Network to perform an analyses

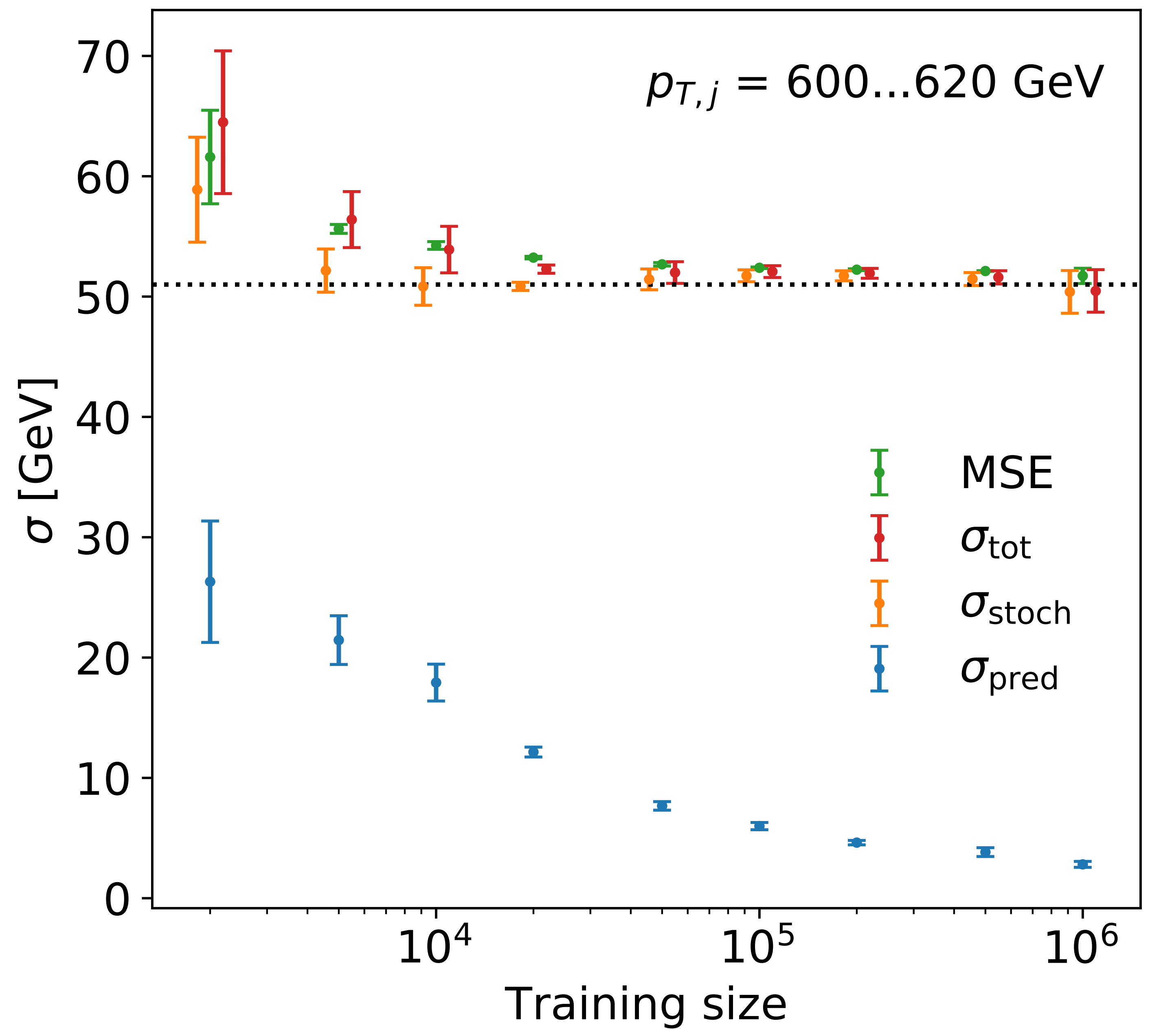
Input X: Jet observables POI Z: $p_{T,GEN}$

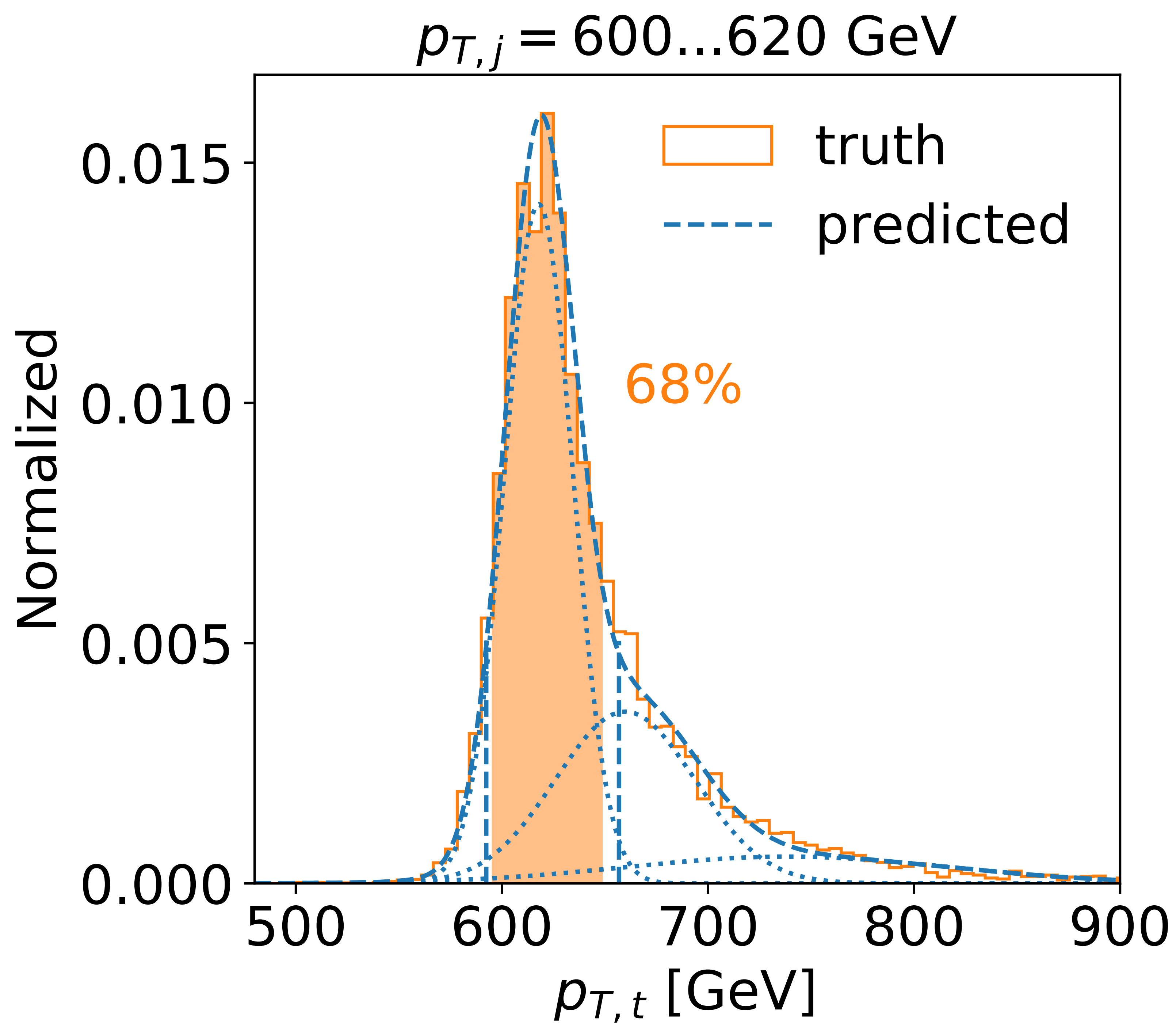
Use a set of Neural Networks $T(x)$ to approximate:

- conditional likelihood ratio
- prior-independent MLE estimate
- Gaussian uncertainties

Possible with BNNs as well:

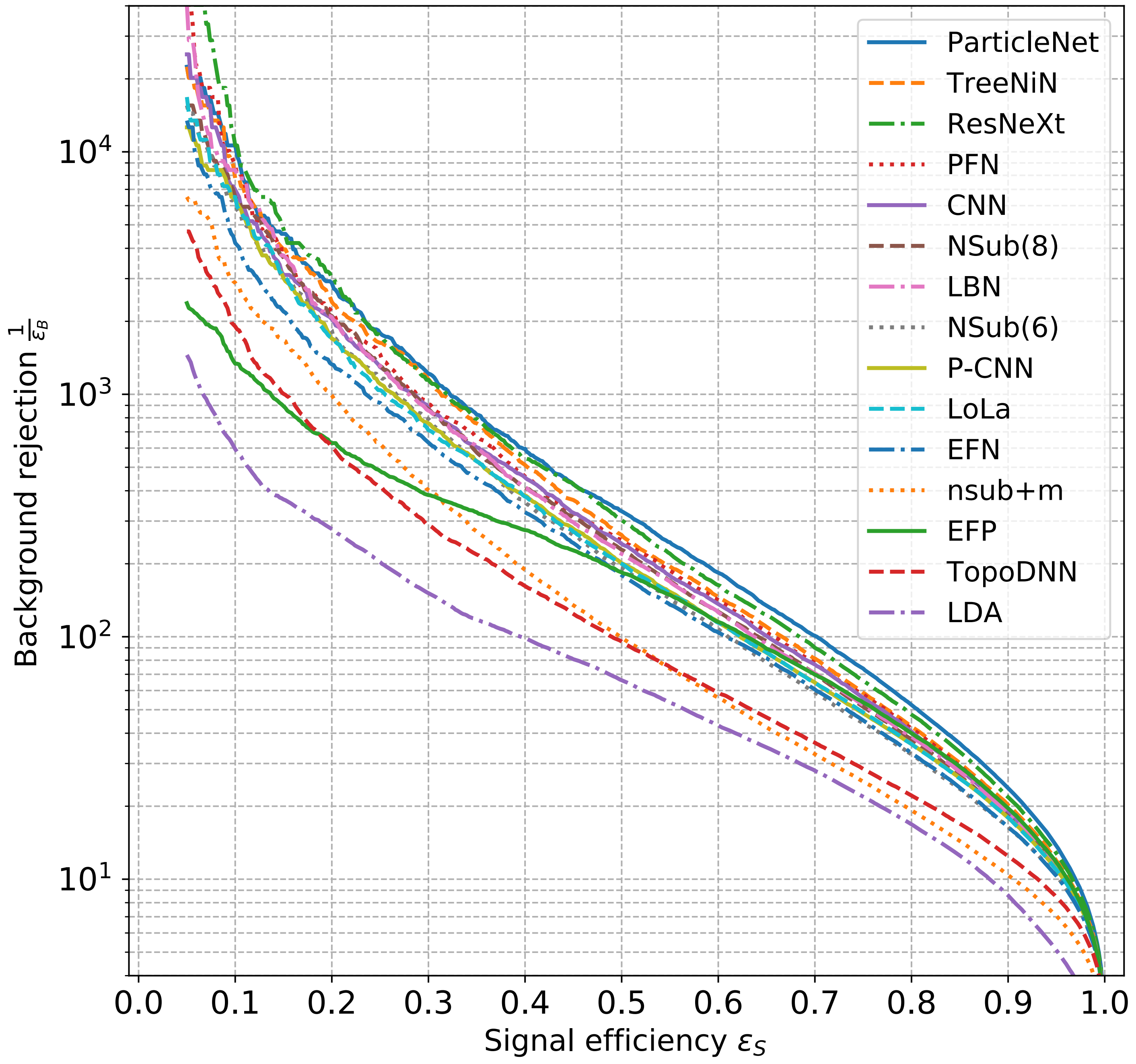
- non-Gaussian uncertainties
- prior (in)dependence

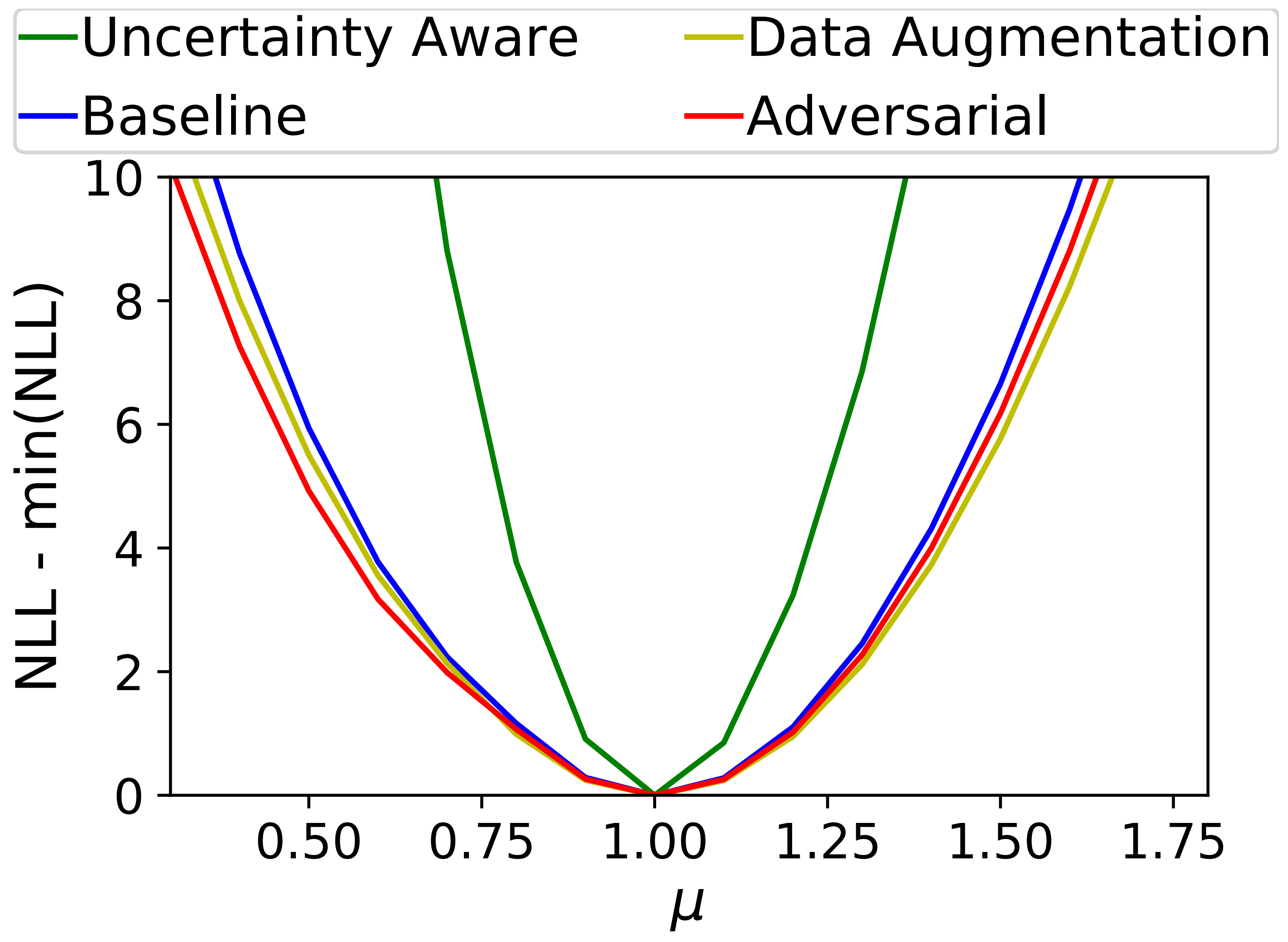




Decompose uncertainties in reducible and irreducible

Classification





Classifiers

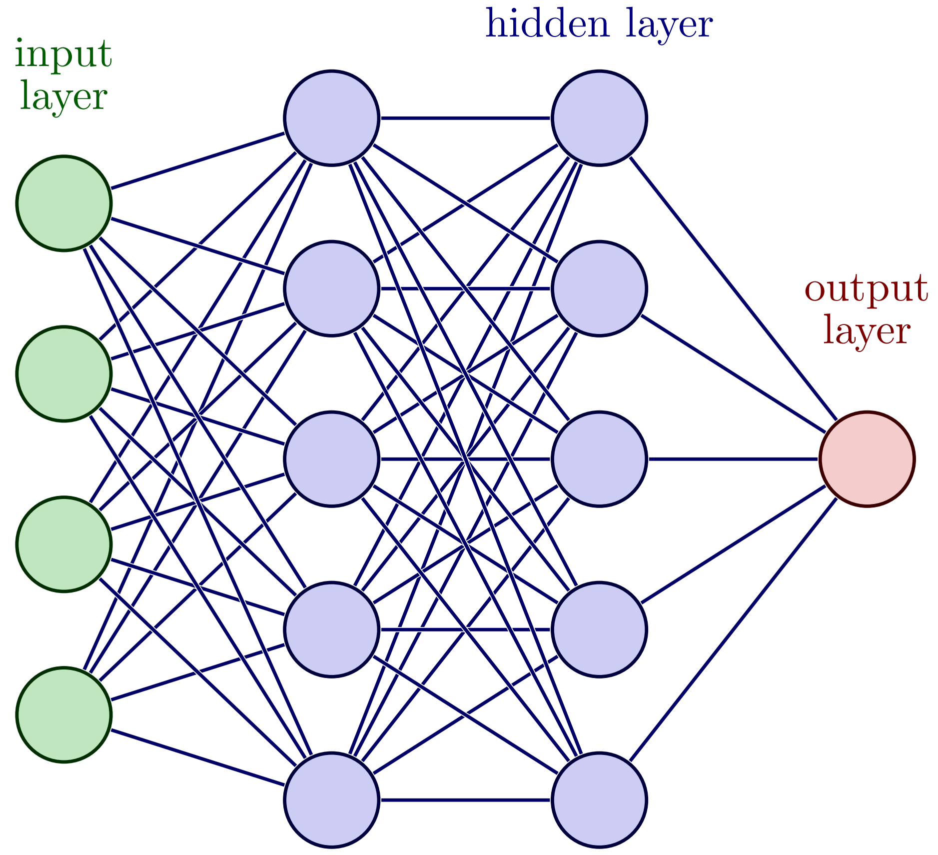
Landscape of Top taggers, arXiv:1902.09914

Extract summary statistic from input observables

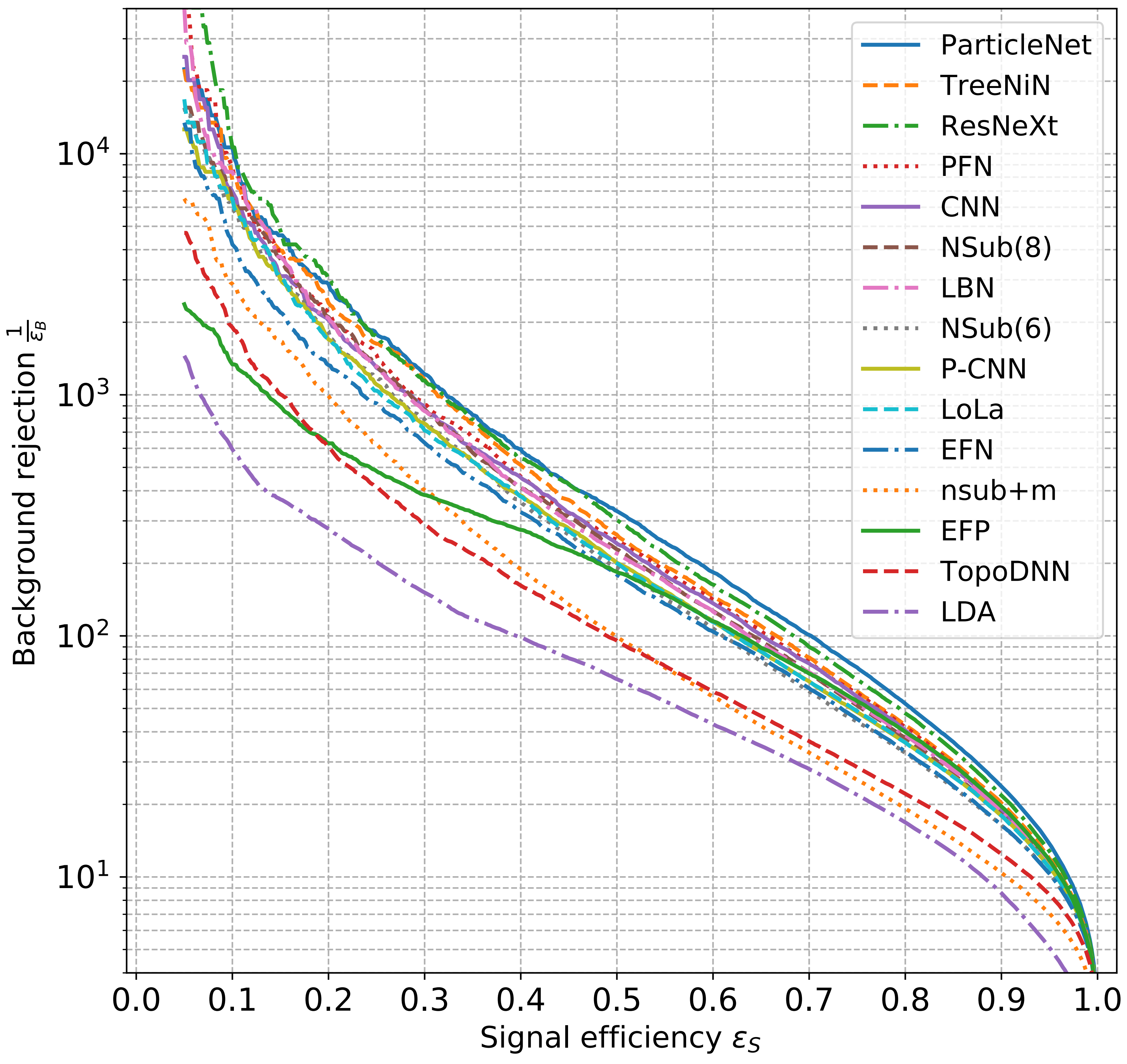
optimal summary statistic for 2-mixture models:

- likelihood ratio $p_s(x)/p_b(x)$

Used everywhere at LHC: “the present” for performance



$\longrightarrow \quad p_\theta(x) \in [0,1]$



Moving forward:

dependence on Nuisance Parameters (NPs), uncertainty aware, and resilience are pivotal concepts

Studying systematics

Learning to Pivot, arXiv:1611.01046
DisCo Fever, arXiv:2001.05310

Classifier output will depend on NPs, how to reduce this effect?

Introduce NPs information during training:

- Augment data with NPs
- Penalizing loss function
- Adversarial loss

Hinders optimality of the classifier

$$dCorr^2(X,Y) = \frac{dCov^2(X,Y)}{dCov(X,Y)\,dCov(Y,Y)}$$

$$\mathscr{L} = L - \lambda L_{Adv}$$

$$s(x) = \frac{\langle p(x|z,S)\rangle_{p_z}}{\langle p(x|z,S)\rangle_{p_z} + \langle p(x|z,B)\rangle_{p_z}}$$

Simulation for each value of z

Studying systematics

Learning to Pivot, arXiv:1611.01046
DisCo Fever, arXiv:2001.05310

Classifier output will depend on NPs, how to reduce this effect?

Introduce NPs information during training:

- Augment data with NPs
- Penalizing loss function
- Adversarial loss

Hinders optimality of the classifier

$$s(x) = \frac{\langle p(x|z,S)\rangle_{p_z}}{\langle p(x|z,S)\rangle_{p_z} + \langle p(x|z,B)\rangle_{p_z}}$$

$$dCorr^2(X,Y) = \frac{dCov^2(X,Y)}{dCov(X,Y)\,dCov(Y,Y)}$$

Simulation for each value of z

$$\mathscr{L} = L - \lambda L_{Adv}$$

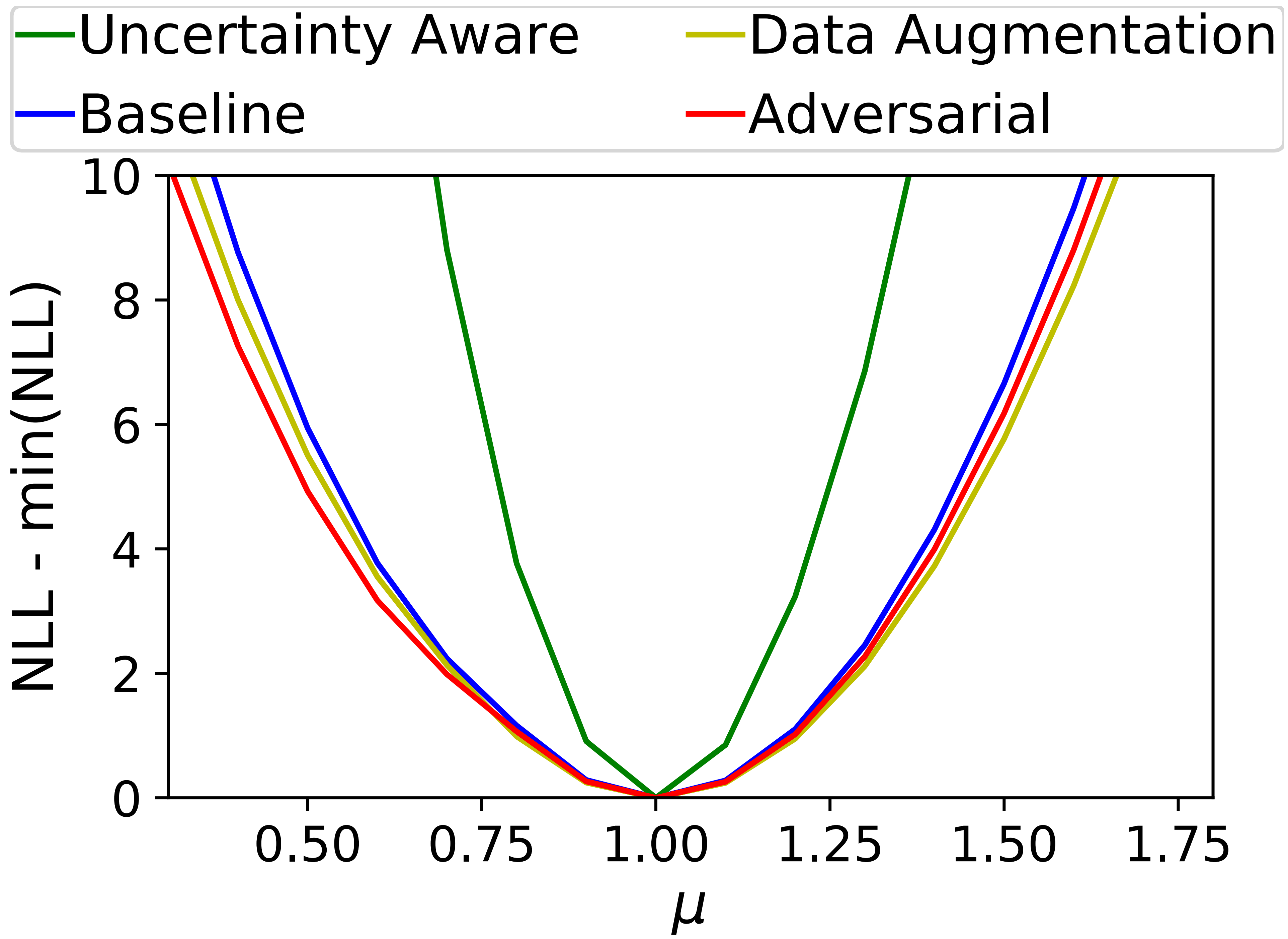
Cannot de-correlate if NPs are aligned with features

Example: # of constituents in Quark/Gluon tagging

Uncertainty-aware classifier

Ghosh A., Nachman B., Whiteson D., arXiv:2105.08742
INFERNO, arXiv:1806.04743

- Introduce z as a network parameter: $f_\theta(x_1, x_2, \ldots, z)$
- NP can be varied during evaluation
- Tested on an analytic likelihood example
- Classifier is aware of possible values of z
- Final evaluation on Profiled Likelihood



$\mu = 1$ data at: $z = \frac{\pi}{2}$

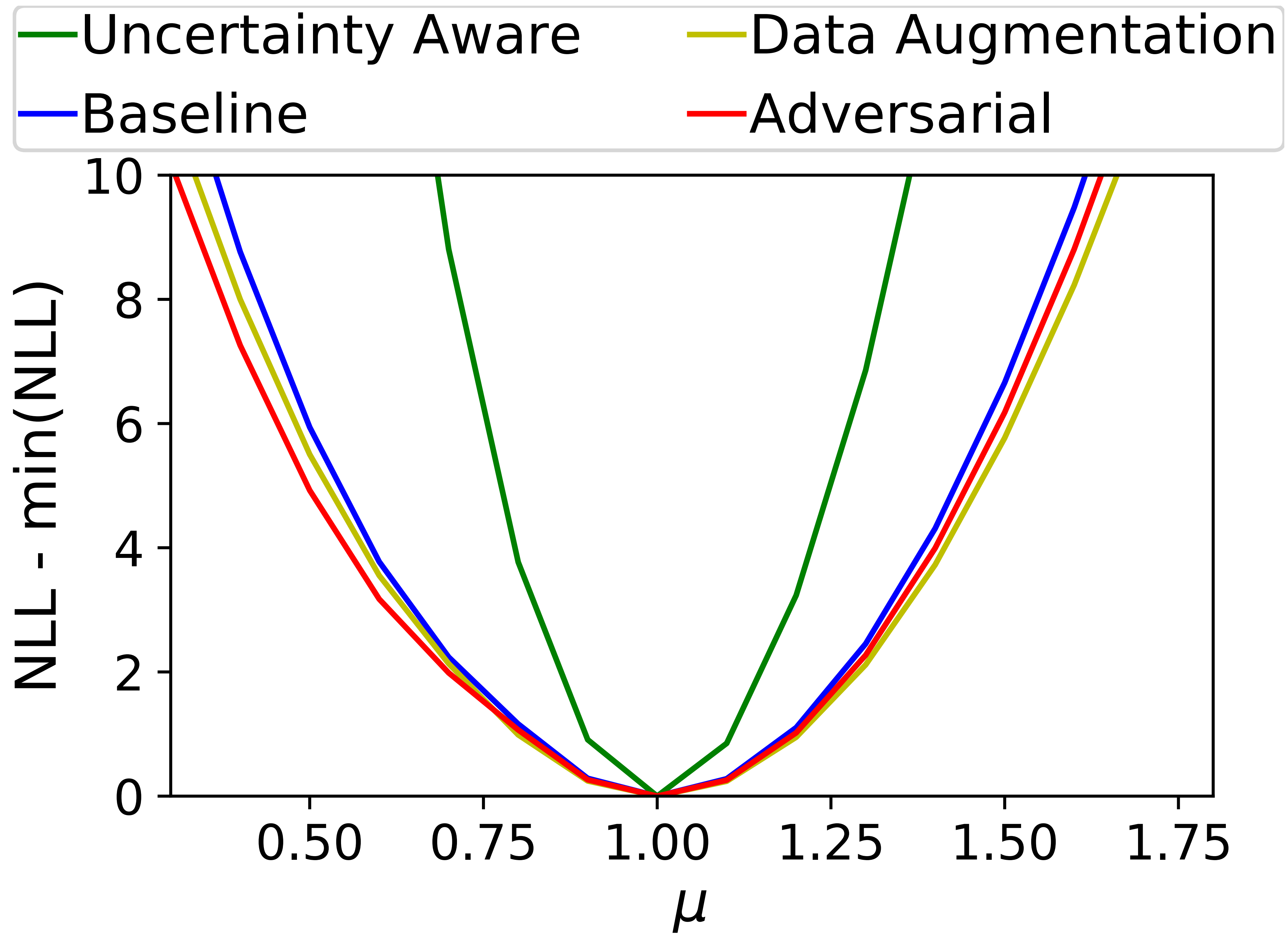
$z = \frac{\pi}{4}$ (left), uncertainty-aware (right)

Uncertainty-aware classifier

Ghosh A., Nachman B., Whiteson D., arXiv:2105.08742
INFERNO, arXiv:1806.04743

- Introduce z as a network parameter: $f_\theta(x_1, x_2, \ldots, z)$
- NP can be varied during evaluation
- Tested on an analytic likelihood example
- Classifier is aware of possible values of z
- Final evaluation on Profiled Likelihood

Uncertainty-aware classifier retains optimal performance



$\mu = 1$ data at: $z = \frac{\pi}{2}$

$z = \frac{\pi}{4}$ (left), uncertainty-aware (right)

Performance vs Resilience

Butter A., Dillon B., Plehn T., Vogel L., arXiv:2212.10493

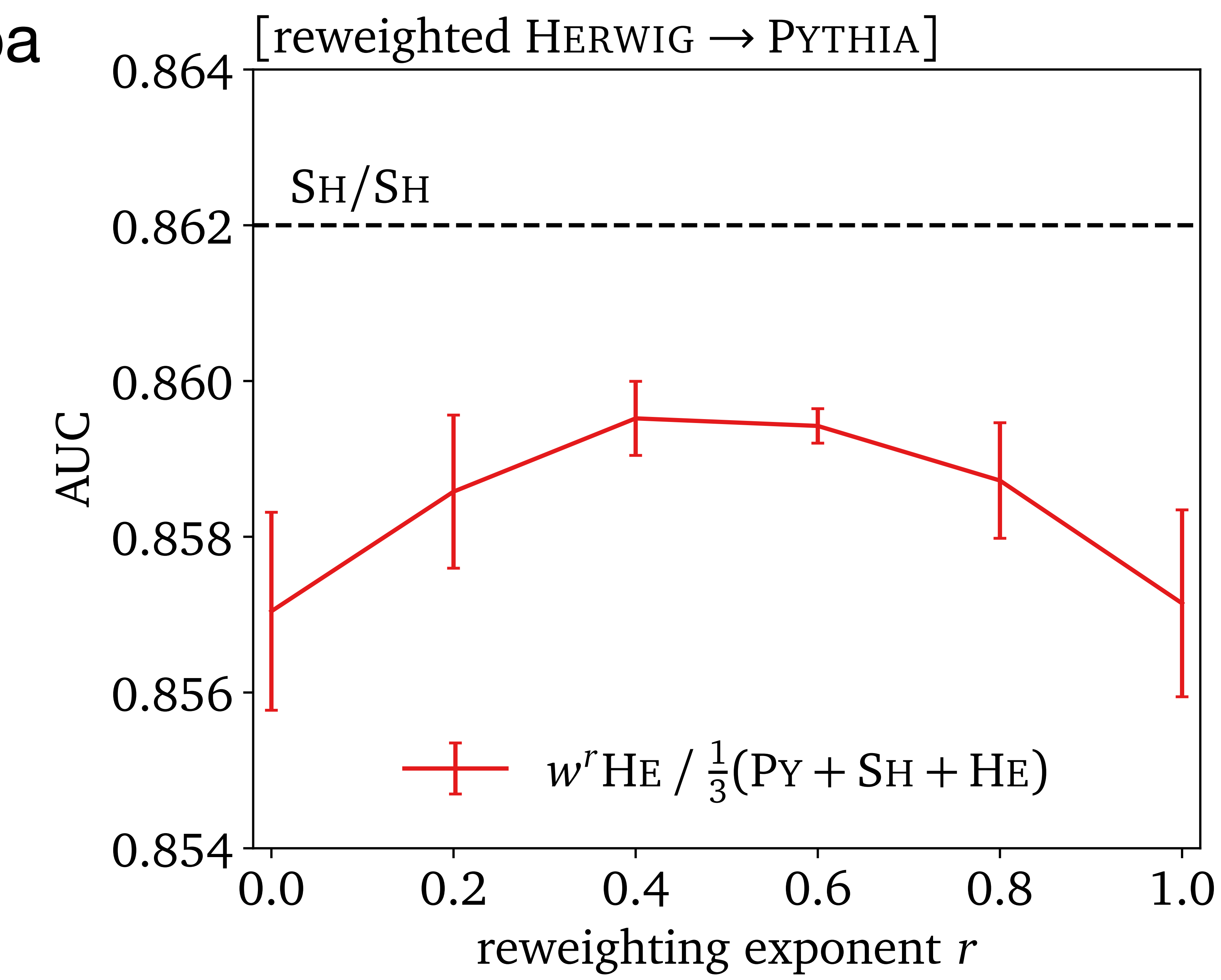
Study resilience vs performance, training on simulations

Model: Bayesian classifier **Data:** train on Pythia/Herwig, test on Sherpa

Introduce a reweighing parameter r:

$$r = 0 \longrightarrow \text{Herwig} \qquad r = 1 \longrightarrow \text{Pythia}$$

$$\mathscr{L} = -\frac{1}{M}\sum_{i=1}^{M} w(x_i)^r \log p(y_i|x_i,\omega) + KL$$

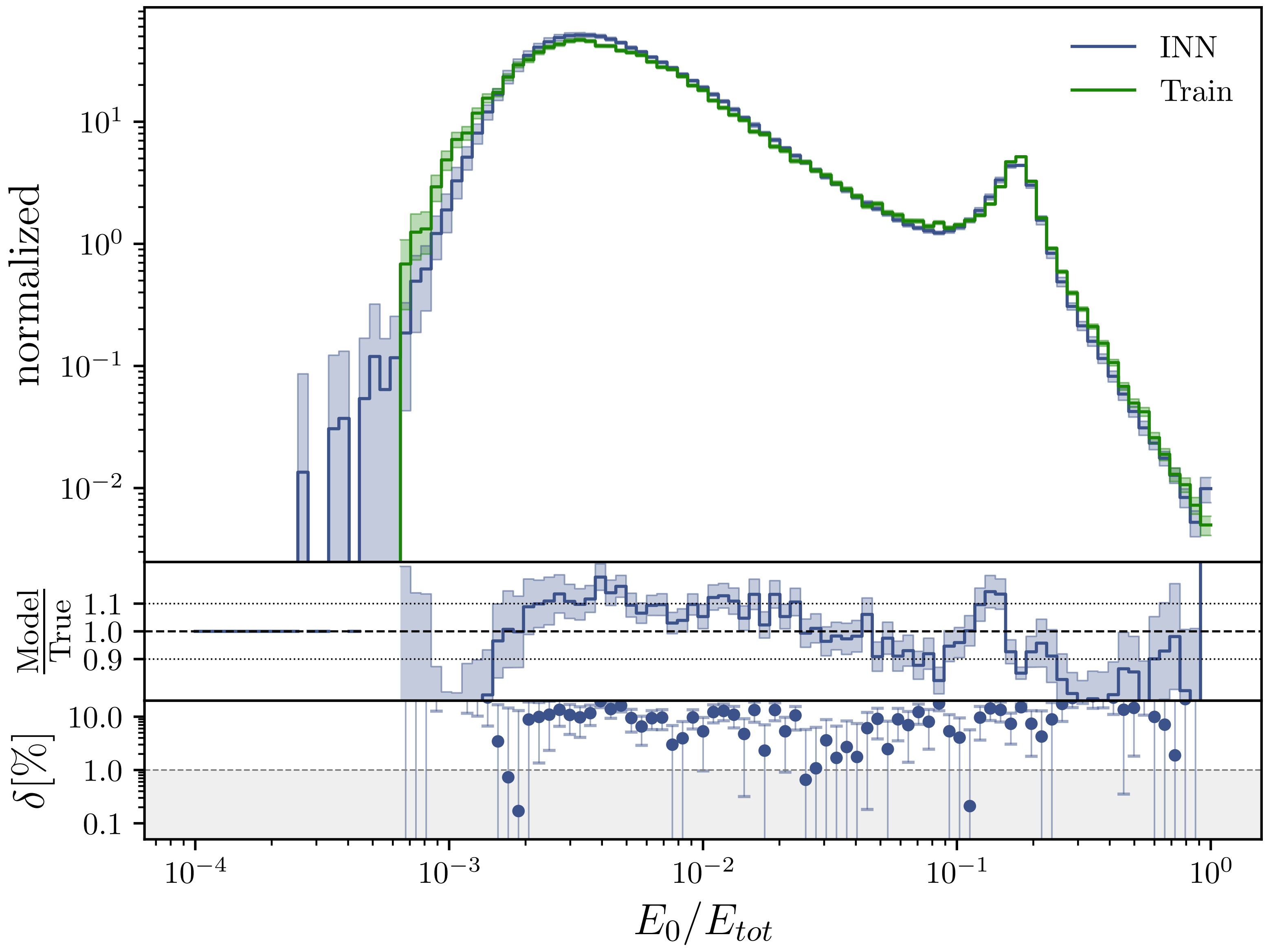


Study uncertainties and performance as a function of r:

- Optimal value of r on the calibration dataset (sherpa), estimate uncertainties
- Check calibration of the interpolated training

Generation

“A Physics conference in the Alps in cartoon style”
- stability.ai



“The” problem in ML

The power of ML has been shown outside Physics in the last years

ML community effort:

Natural Language Processing (NLP)

Images generation

DALL·E 2

stability.ai

Crayon

Implications in Particle Physics:

- event generation
- detector simulation
- unfolding
- anomaly detection

Generative Models

“A Physics conference in the Alps in cartoon style”
- stability.ai

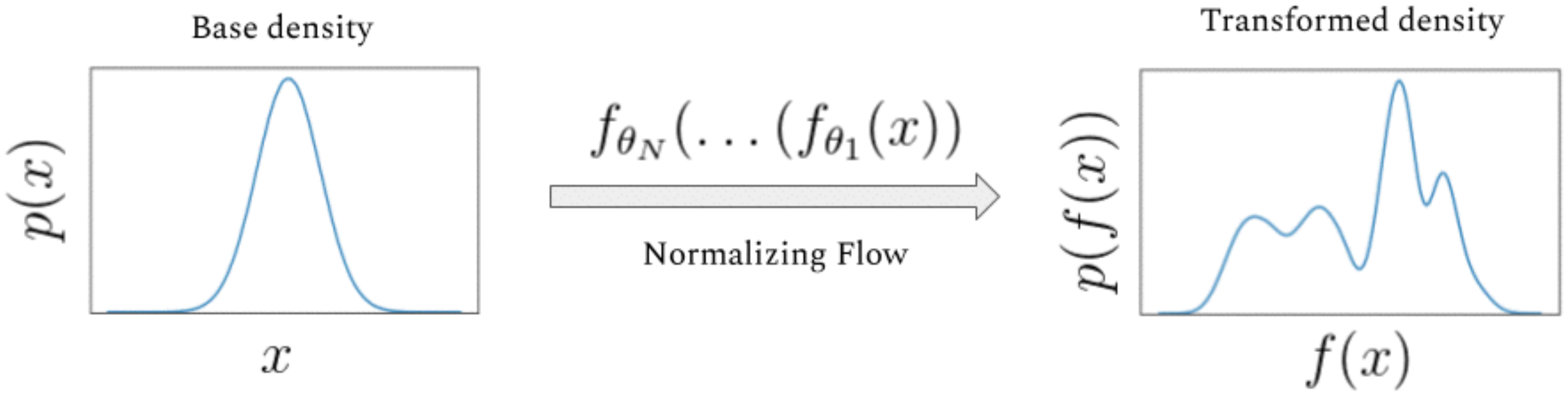
Normalizing Flows

Favaro L., Plehn T., Krause C., Shih D., arXiv:23??.SOON
CATHODE, arXiv:2109:00546

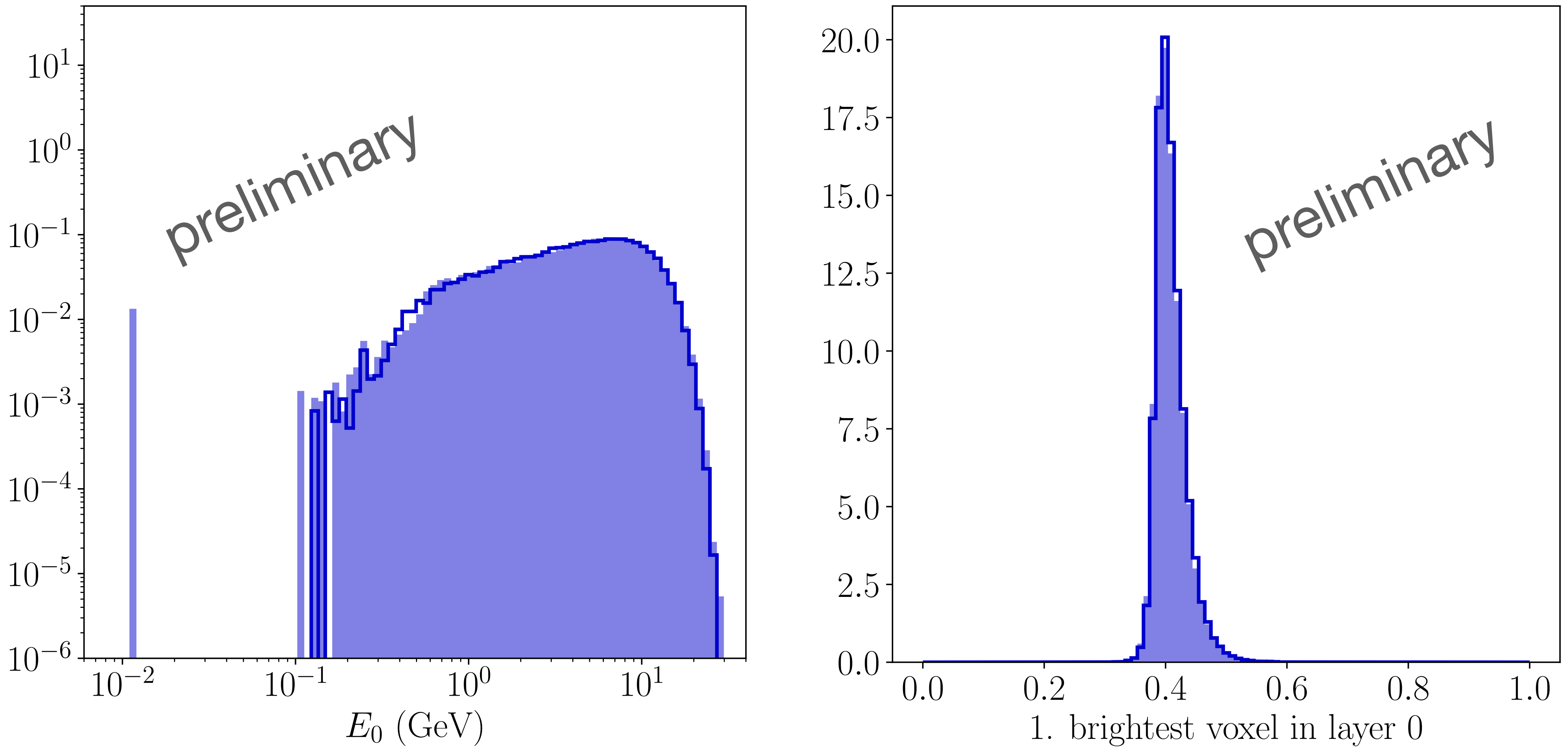
Path to % precision in learning probability distributions?

Learn a bijective transformation parametrized by a NN:

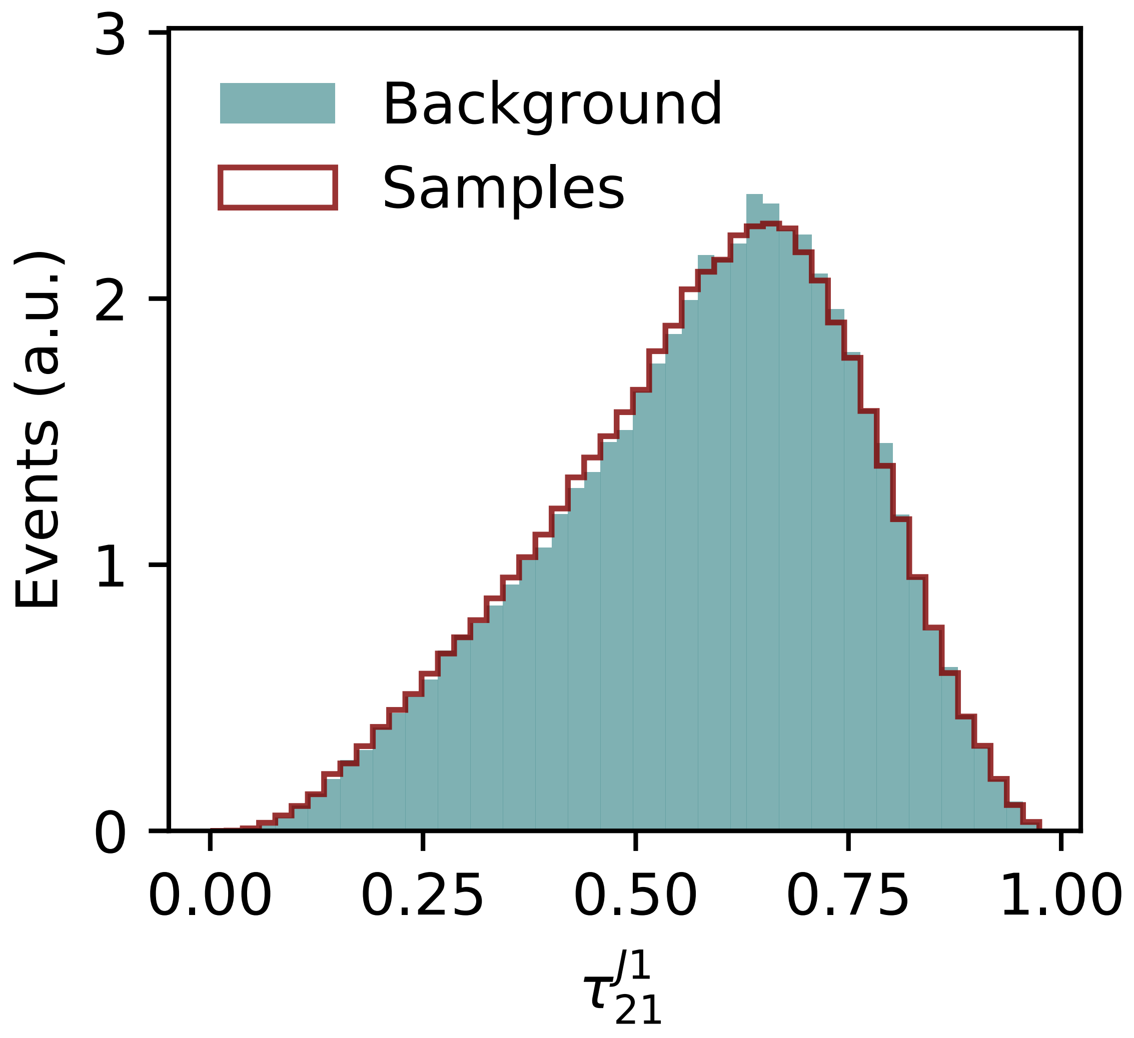
$$p_X(x) = p_Z(f_\theta^{-1}(x)) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$



e^+ calorimeter simulation



Anomaly detection (CATHODE)



Generative networks

Task: learning $p_{data}(x)$ with (complex) NN models $\longrightarrow$ natural objective: minimize $\mathscr{L} = -\log p_\theta(x)$

Questions:

- Are there missing features?
- Is the density correctly estimated?
- $p_{true} \neq p_{data}$ are we overtraining?
- Do we gain statistics in generation?

Generative networks

Task: learning $p_{data}(x)$ with (complex) NN models $\longrightarrow$ natural objective: minimize $\mathscr{L} = -\log p_\theta(x)$

Questions:

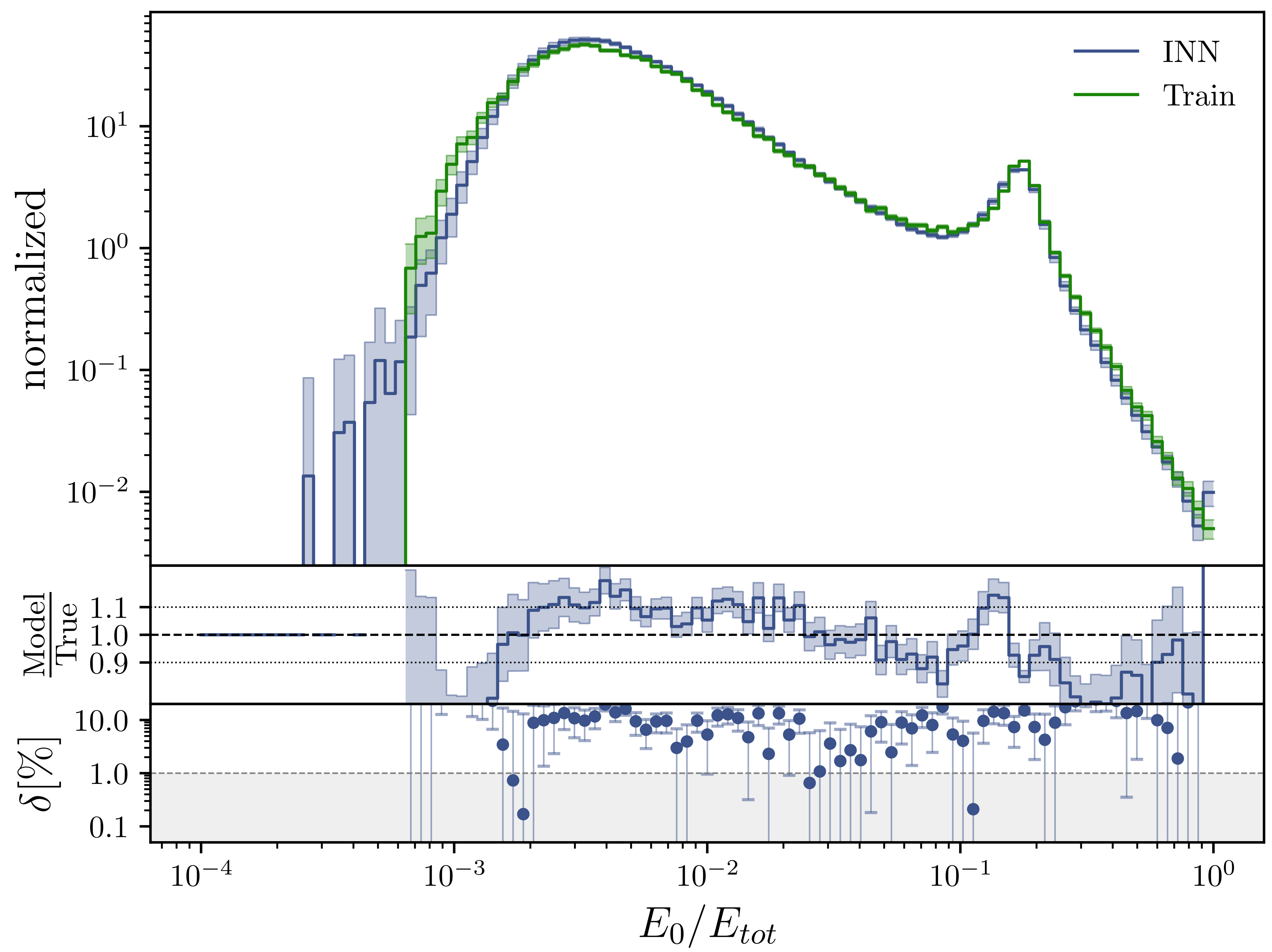
- Are there missing features?
- Is the density correctly estimated?
- $p_{true} \neq p_{data}$ are we overtraining?
- Do we gain statistics in generation?

Controlling Generative Networks is essential

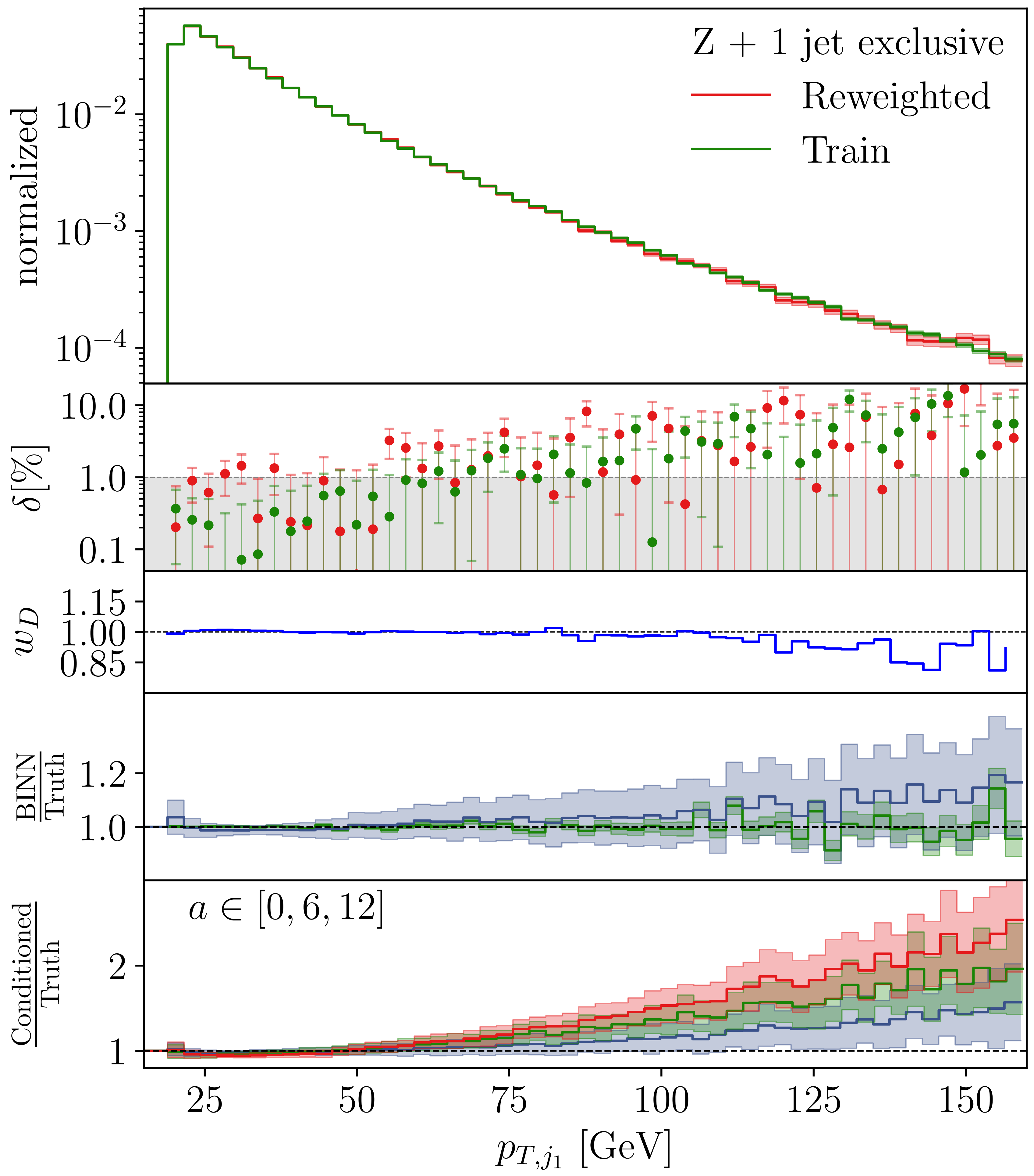
Density estimation - BNN

Generative Networks for Precision Enthusiasts, arXiv:2110.13632
Favaro L., Plehn T., Krause C., Shih D., arXiv:23??.SOON

Model uncertainties in Generative networks:
- control uncertainties on the density estimate



Calorimeter simulation



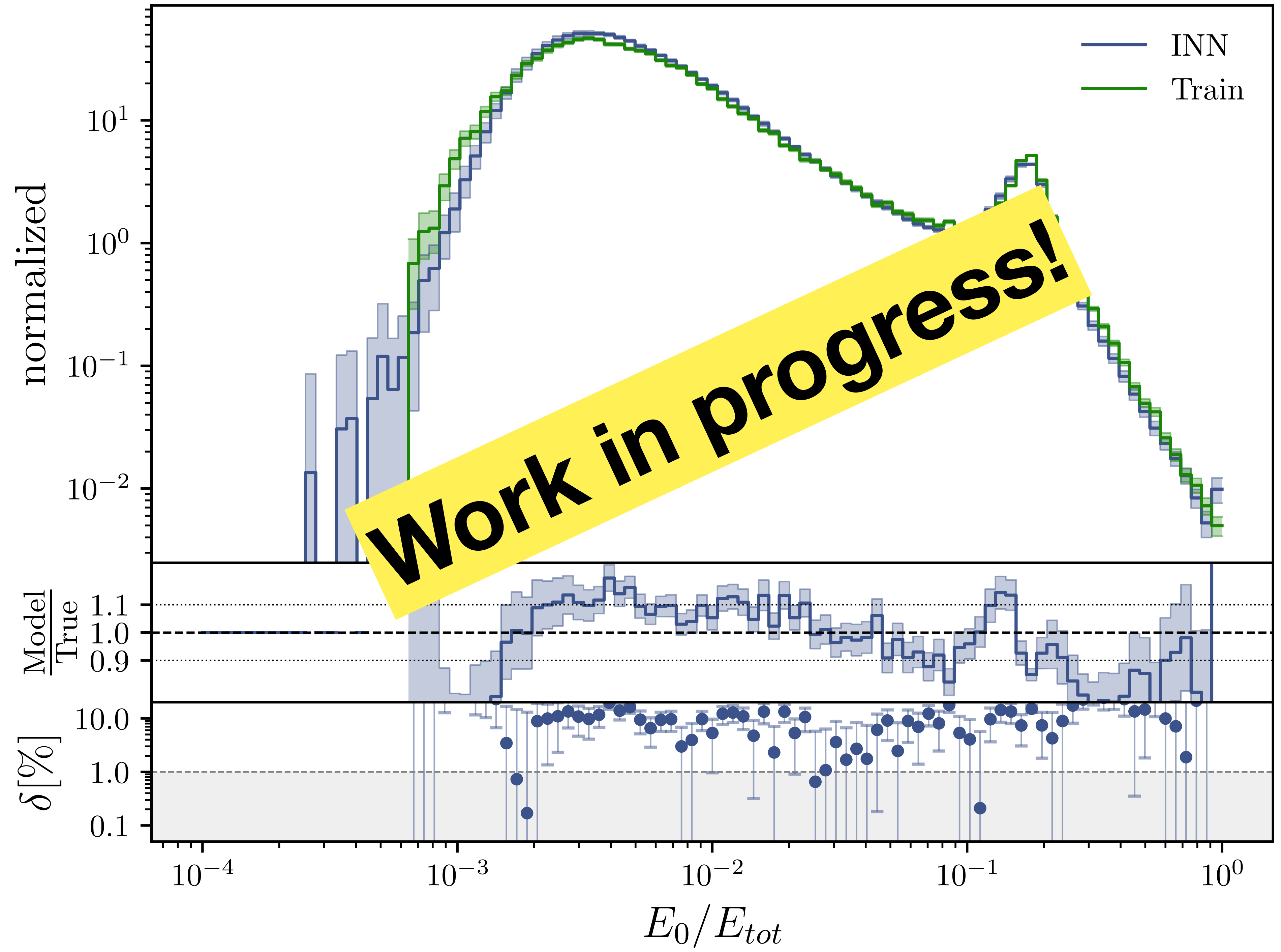
Event generation

Density estimation - BNN

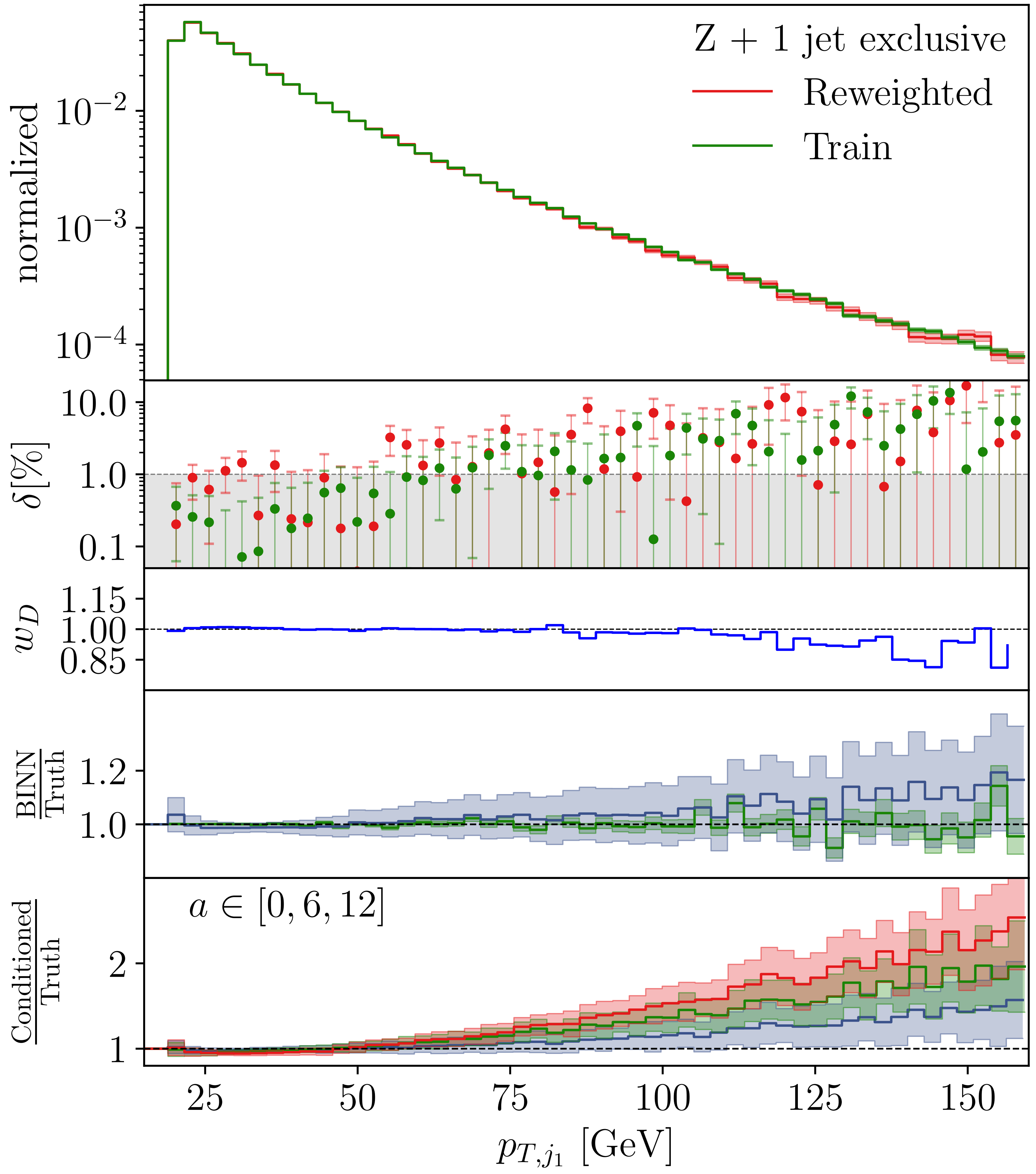
Generative Networks for Precision Enthusiasts, arXiv:2110.13632
Favaro L., Plehn T., Krause C., Shih D., arXiv:23??.SOON

Model uncertainties in Generative networks:

- control uncertainties on the density estimate



Calorimeter simulation



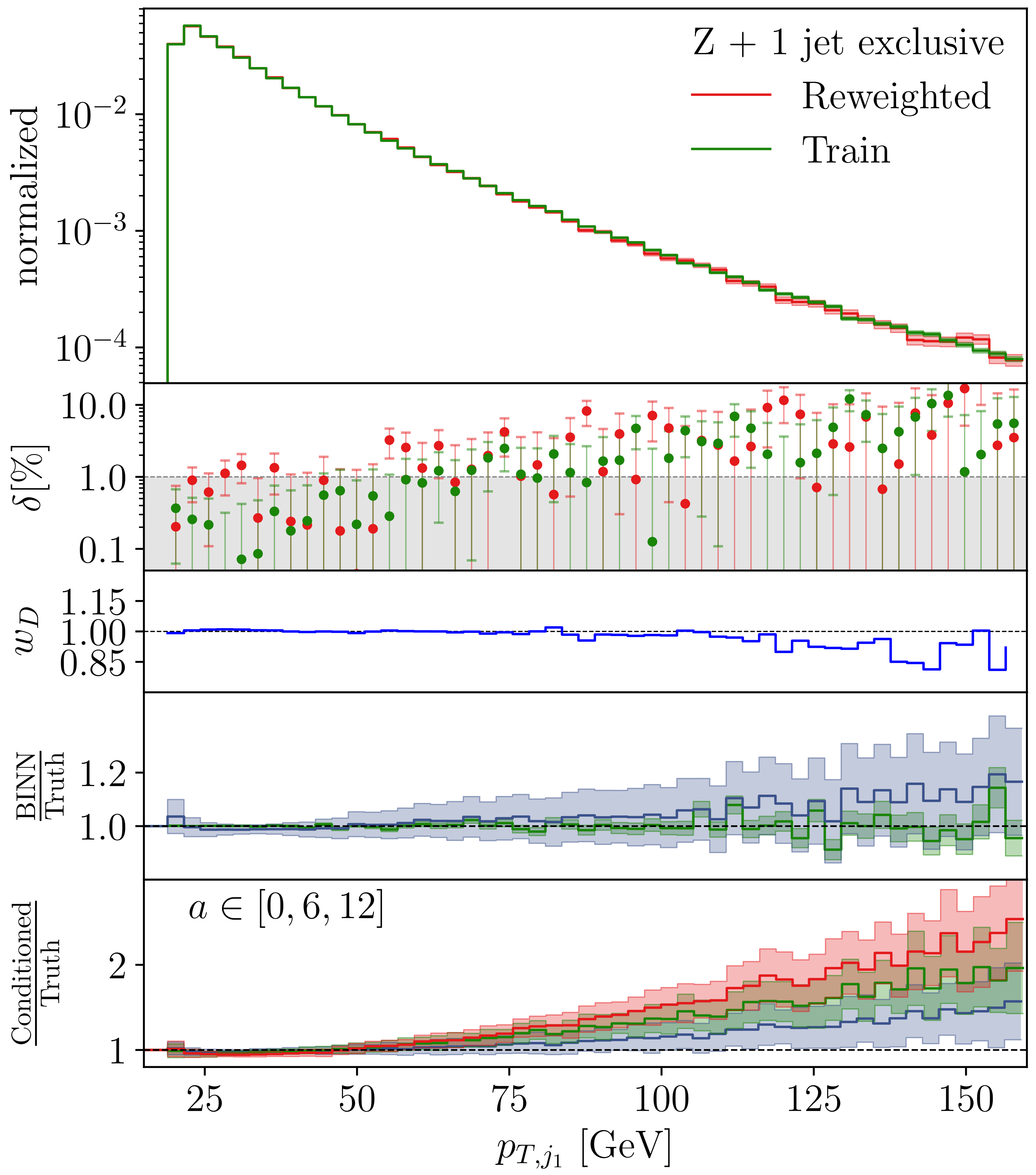
Event generation

Density estimation - BNN

Generative Networks for Precision Enthusiasts, arXiv:2110.13632
Favaro L., Plehn T., Krause C., Shih D., arXiv:23??.SOON

- Model uncertainties from NPs adding a conditional training
- Generative model for $Z \to \mu\mu + jets$
- Introduce additional noise in the tails with a
- Control uncertainties with a Bayesian Normalizing Flow

Learn a conditional probability distribution $p_\theta(x \,|\, a)$



Event generation

Learning from the model

Generative Networks for Precision Enthusiasts, arXiv:2110.13632
DCTRGAN, arXiv:2009.03796

Well known trick in Physics (and only in Physics): reweigthing

Idea of a GAN: train a discriminator to classify samples

define weights from classifier output: $w(x) = \dfrac{D(x)}{1 - D(x)}$

- Reweight according to $w(x)$:

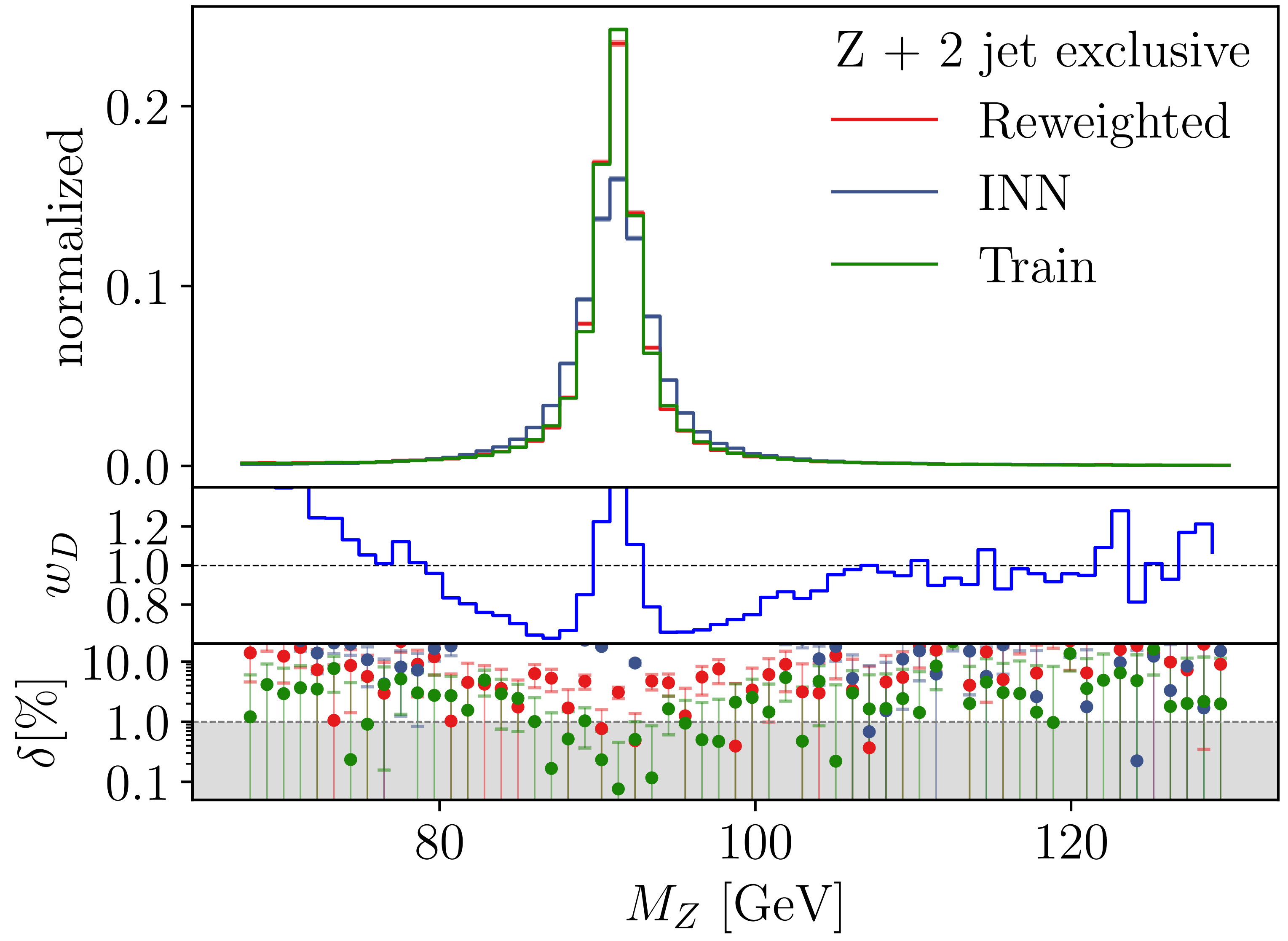
works… but end up with **weighted samples**

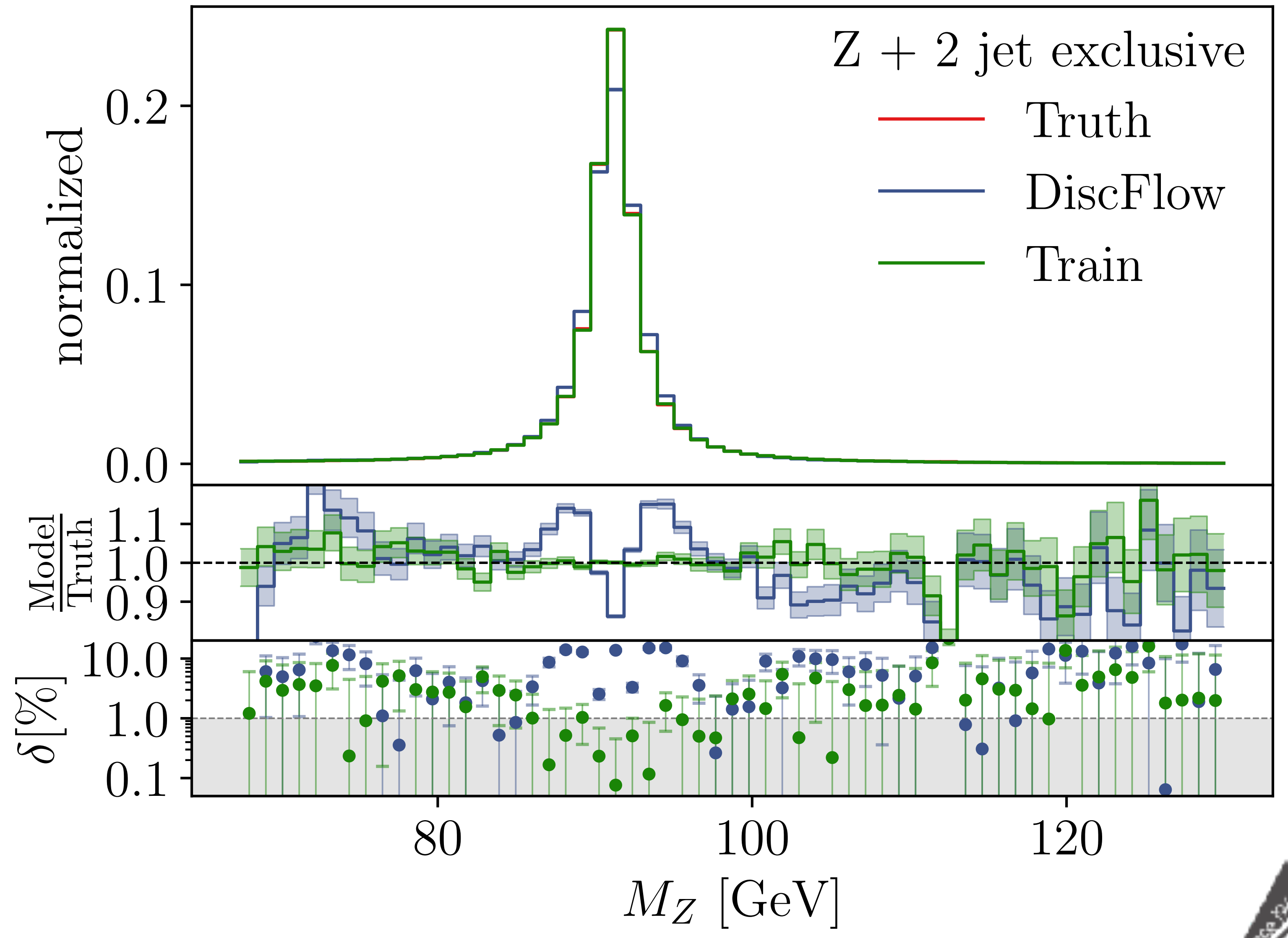
- Add this information to training $\longrightarrow$ improve NN generator

cool idea… not fully understood (yet)

Example: Events generation, $Z \rightarrow \mu\mu + jets$
DiscFlow (CLS + NF) training

Improve (unweighted) mass peak generation





Learning from the model

Generative Networks for Precision Enthusiasts, arXiv:2110.13632
DCTRGAN, arXiv:2009.03796

Well known trick in Physics (and only in Physics): reweigthing

Idea of a GAN: train a discriminator to classify samples

define weights from classifier output: $w(x) = \dfrac{D(x)}{1 - D(x)}$

- Reweight according to $w(x)$:

works… but end up with **weighted samples**

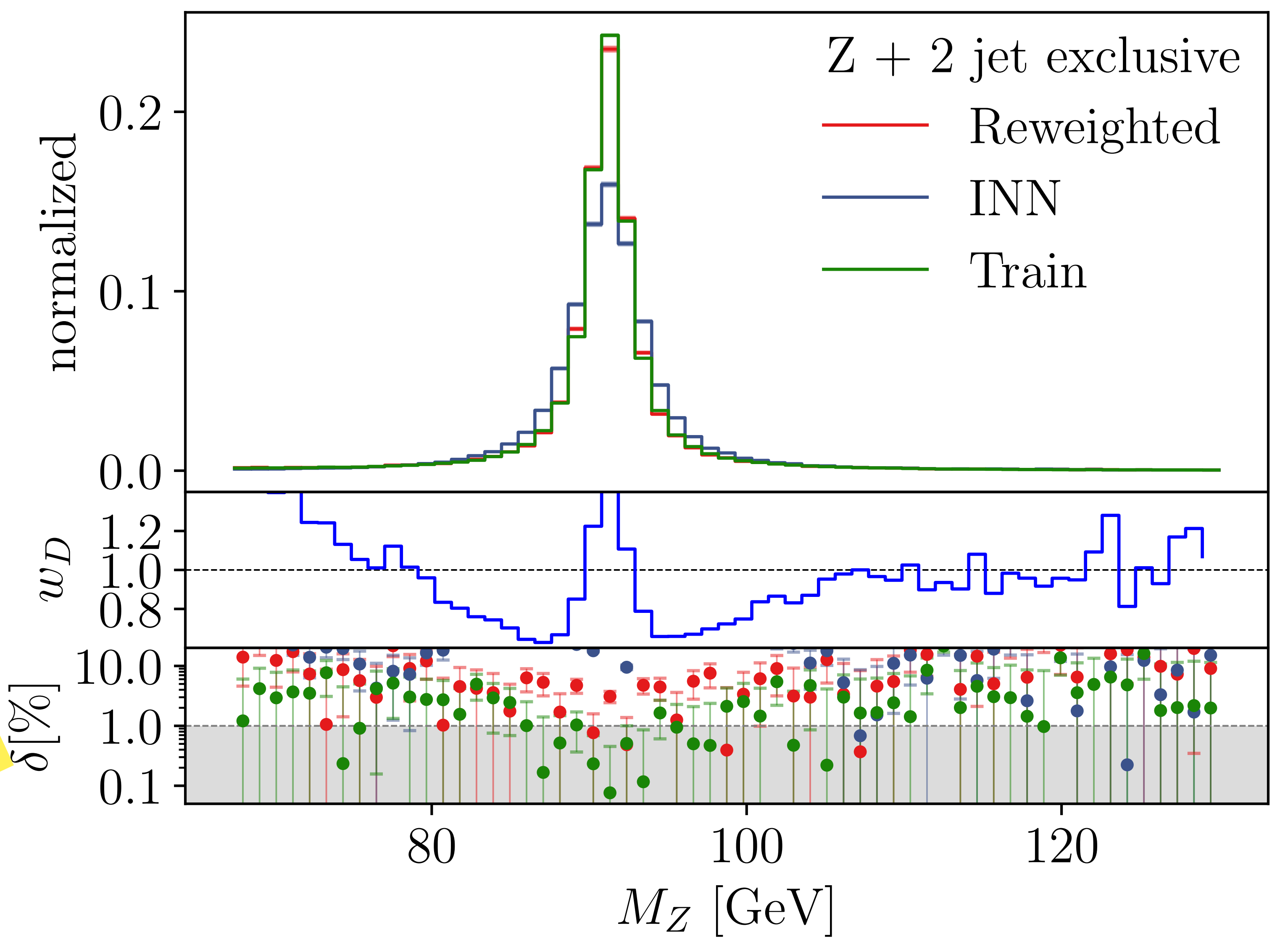
- Add this information to training $\longrightarrow$ improve [illegible] rator

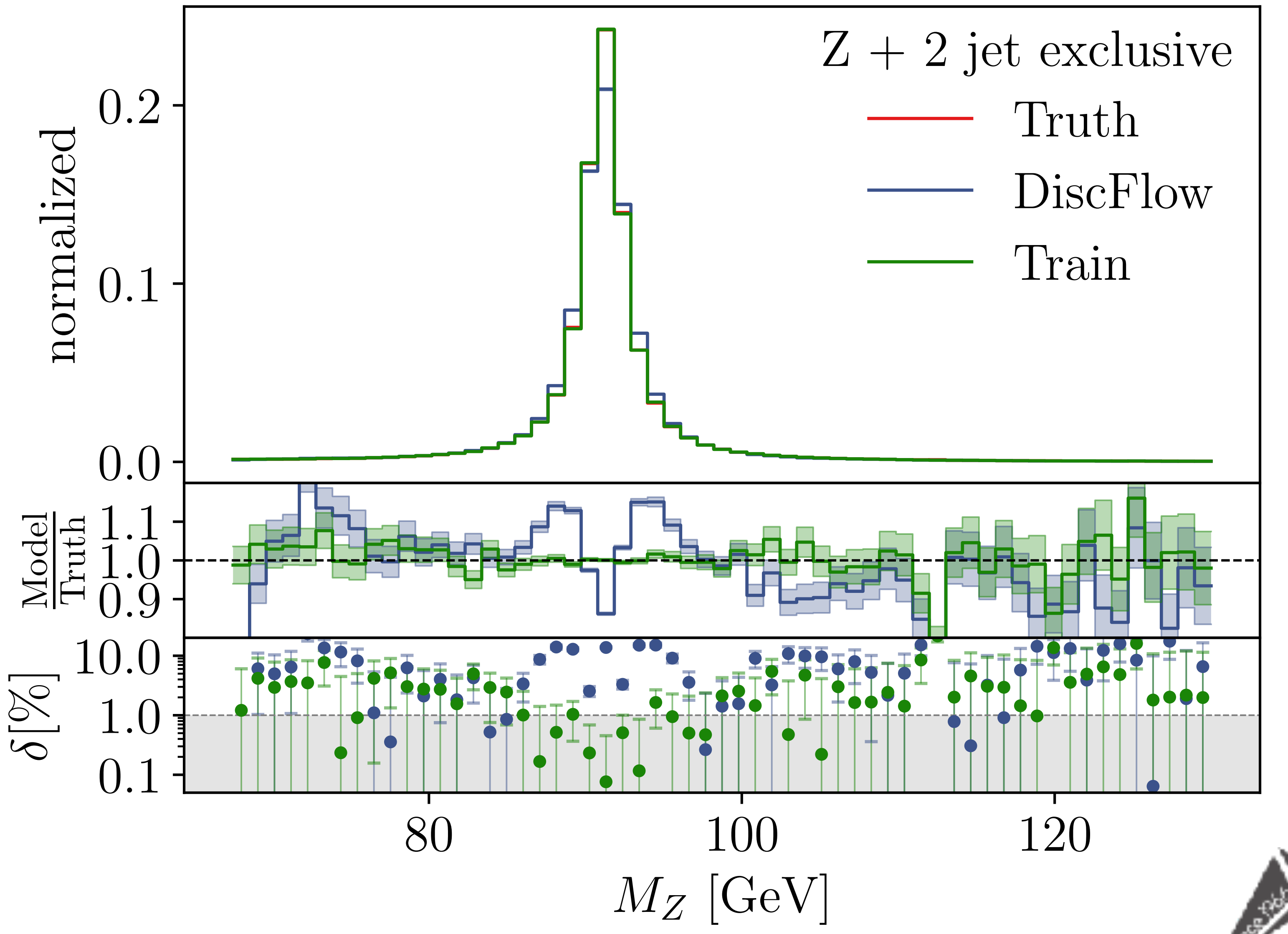
cool idea… not fully understood (yet)

Soon on detector simulation!

Example: Events generation, $Z \to \mu\mu + jets$
DiscFlow (CLS + NF) training

Improve (unweighted) mass peak generation





Summary

Estimating model uncertainties:

• deep ensembles:
reliable, with an awful scaling

• Laplace approximation:
local, applicable to trained models

• Bayesian Neural Networks:
local, active learning, drop-in in most networks

Regression:

• closure tests

• prior independent analysis

• boosted training

• calibration

Classification:

• Optimality

• Effect of nuisance parameters

• Preserving optimality

• performance vs resilience

Generation:

• study reproduction of pd

• optimal metric

• uncertainties from the network

• “active reweighting”

Summary

Estimating model uncertainties:

• deep ensembles:
reliable, with an awful scaling

• Laplace approximation:
local, applicable to trained models

• Bayesian Neural Networks:
local, active learning, drop-in in most networks

Regression:

• closure tests

• prior independent analysis

• boosted training

• calibration

Fast moving field with a lot of new ideas… Stay tuned!

Classification:

• Optimality

• Effect of nuisance parameters

• Preserving optimality

• performance vs resilience

Generation:

• study reproduction of pd

• optimal metric

• uncertainties from the network

• “active reweighting”

Backup

ML uncertainties lingo

Uncertainties

- aleatoric systematic: additional noise in the single measurement
- aleatoric stochastic: stochastic noise in the dataset
- epistemic (systematic): how much the model is uncertain of its prediction (fidelity)

statistical $\longrightarrow$ aleatoric stochastic

systematic $\nearrow$ aleatoric systematic

systematic $\searrow$ epistemic

Studying systematics

Learning to Pivot, arXiv:1611.01046
DisCo Fever, arXiv:2001.05310

Classifier output will depend on NPs, how to reduce this effect?

Introduce NPs information during training:

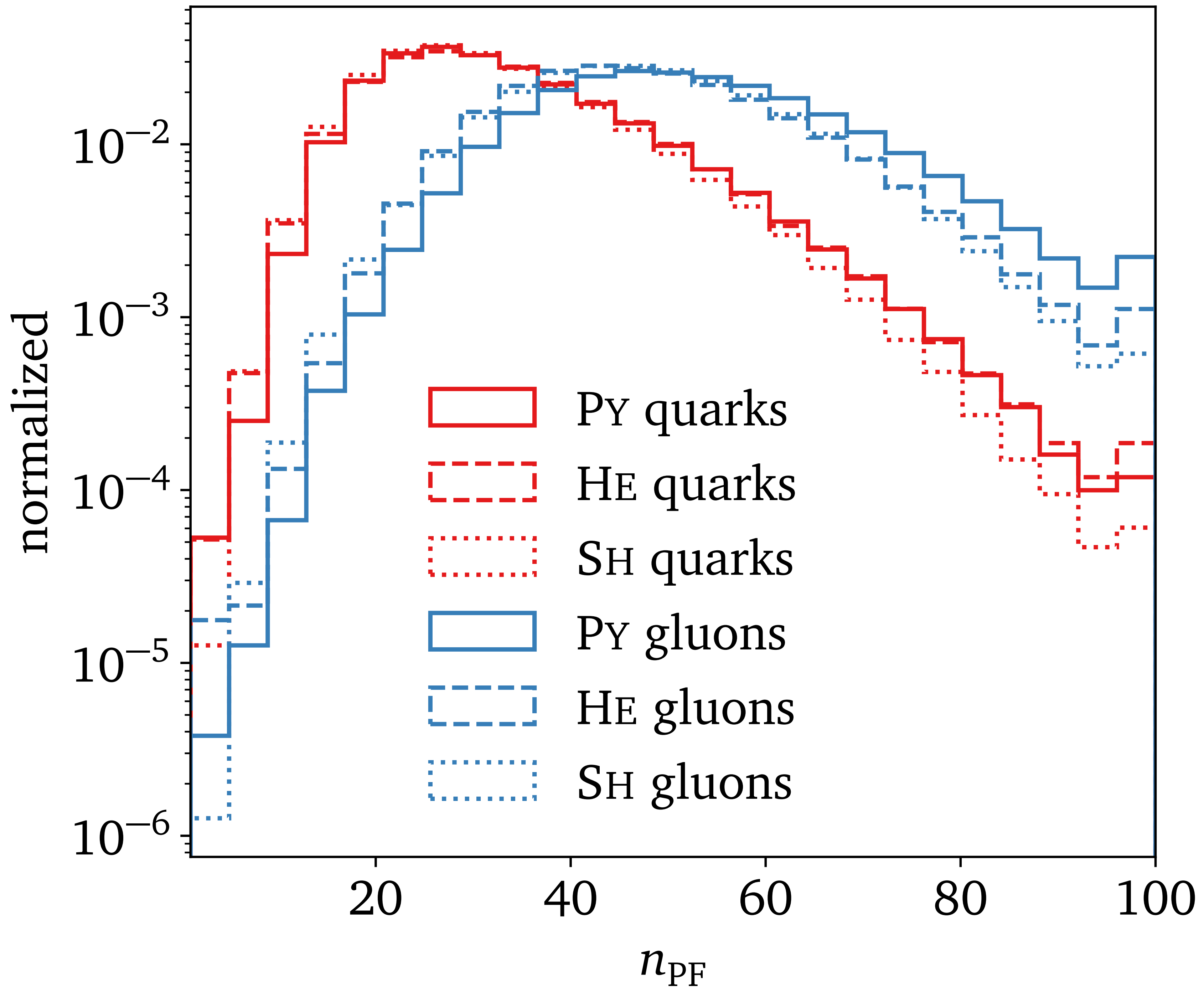
- Augment data with NPs
- Penalizing loss function
- Adversarial loss

Hinders optimality of the classifier

$$dCorr^2(X,Y) = \frac{dCov^2(X,Y)}{dCov(X,Y)\,dCov(Y,Y)}$$

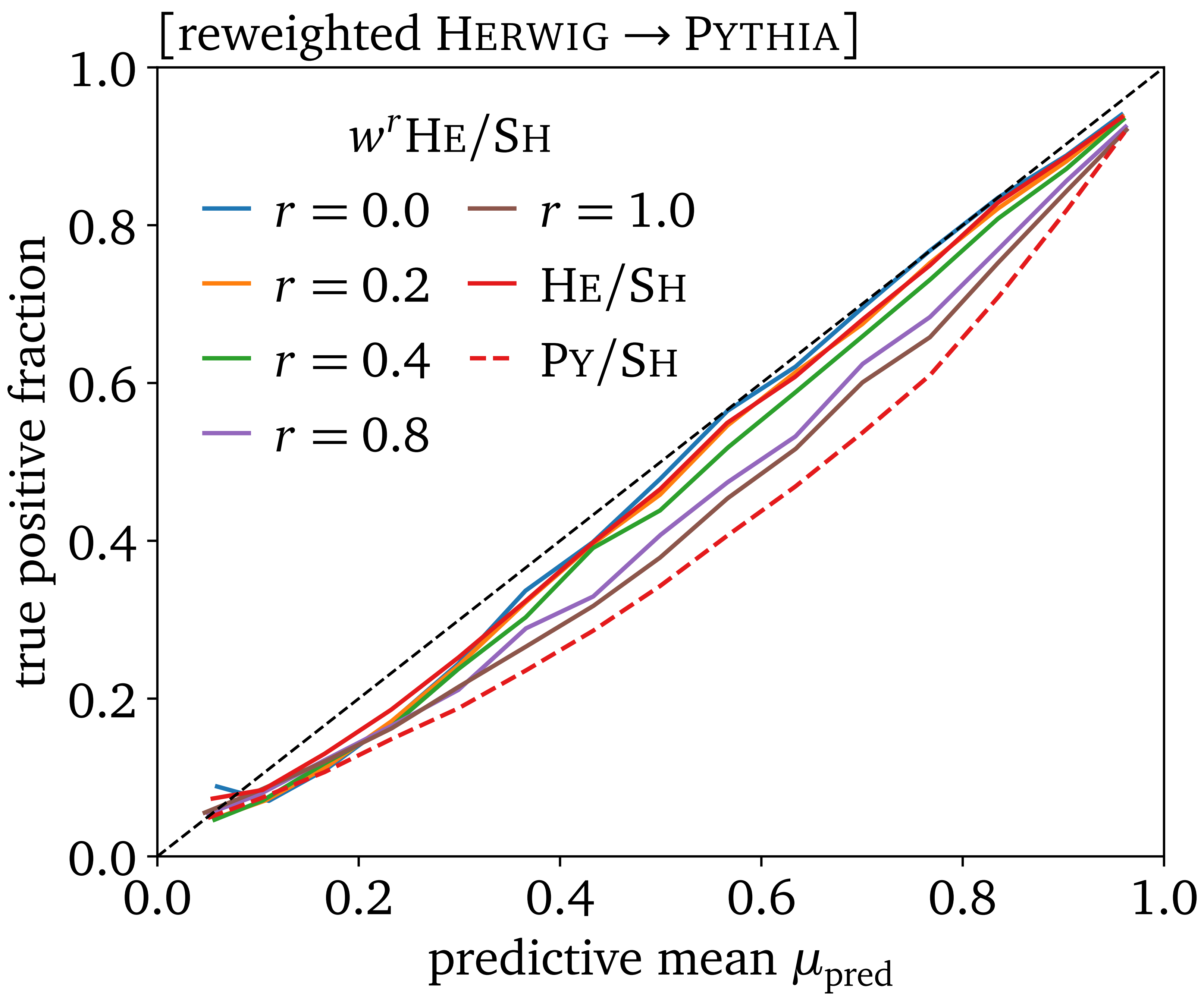
$$\mathscr{L} = L - \lambda L_{Adv}$$

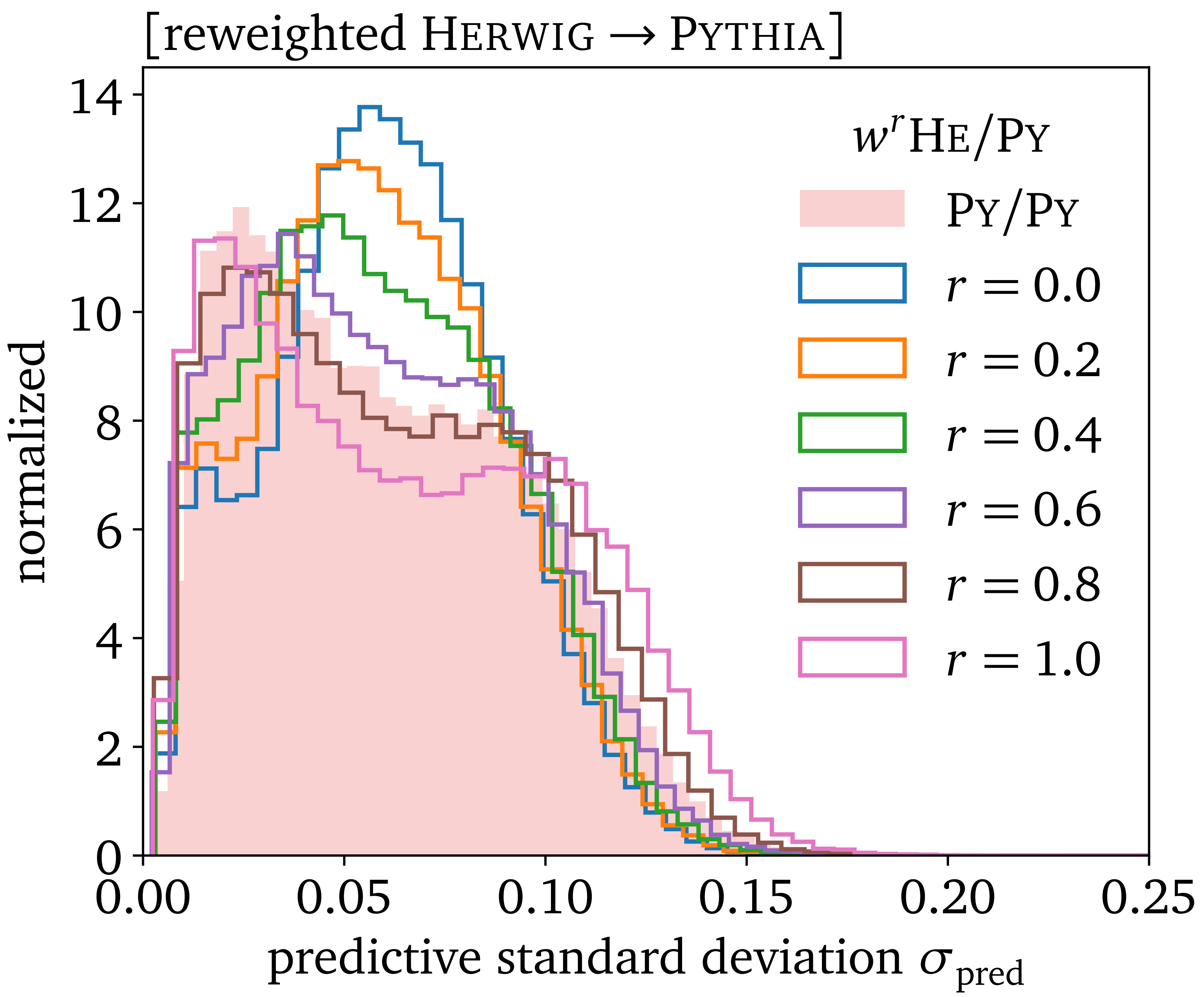
Cannot de-correlate if NPs are aligned with features



Example: # of constituents in Quark/Gluon tagging

Studying systematics





Uncertainty-aware classifier

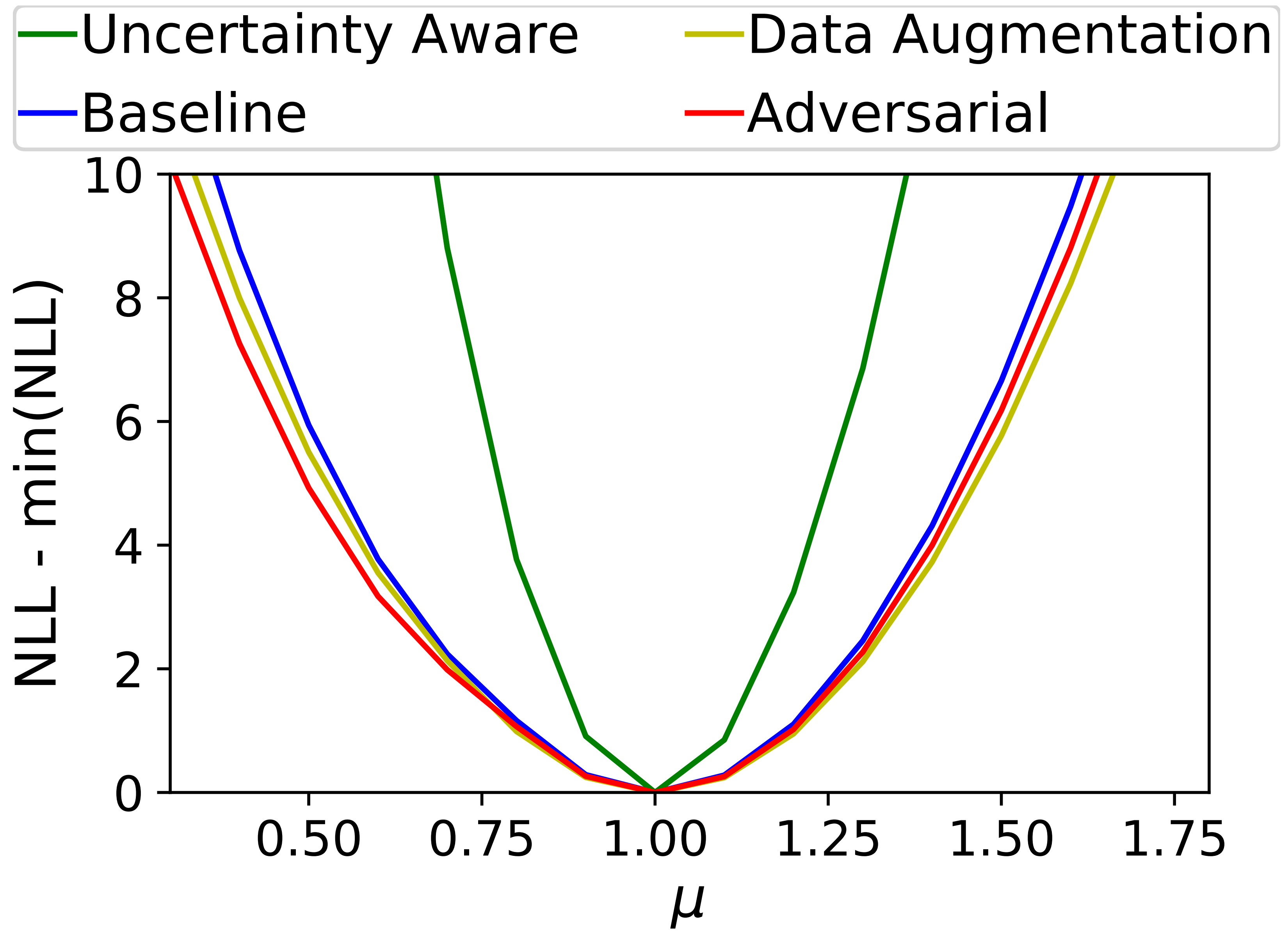
Ghosh A., Nachman B., Whiteson D., arXiv:2105.08742
INFERNO, arXiv:1806.04743

Introduce z as a network parameter: $f(x_1, x_2, \ldots, z)$

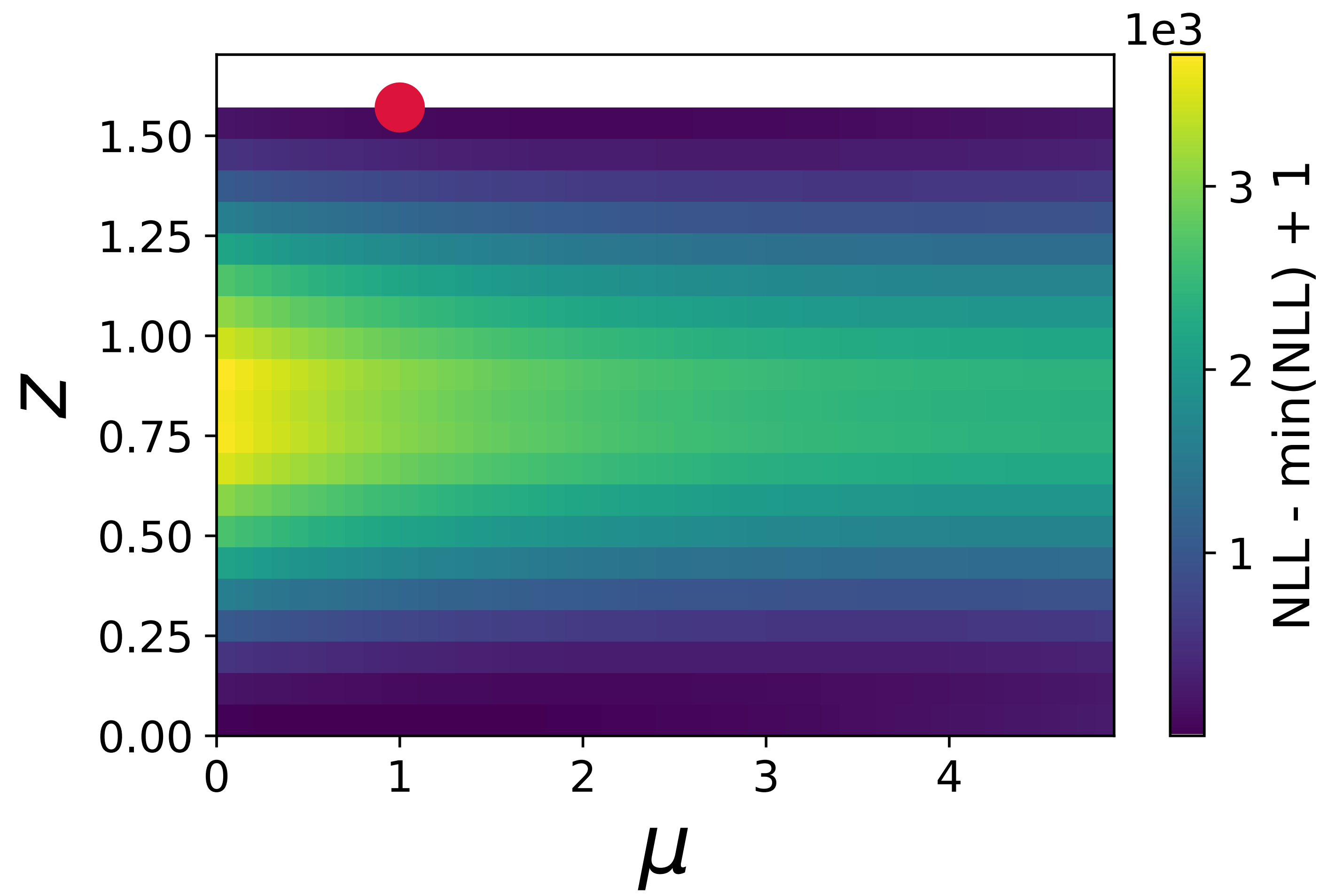
NP can be varied during evaluation

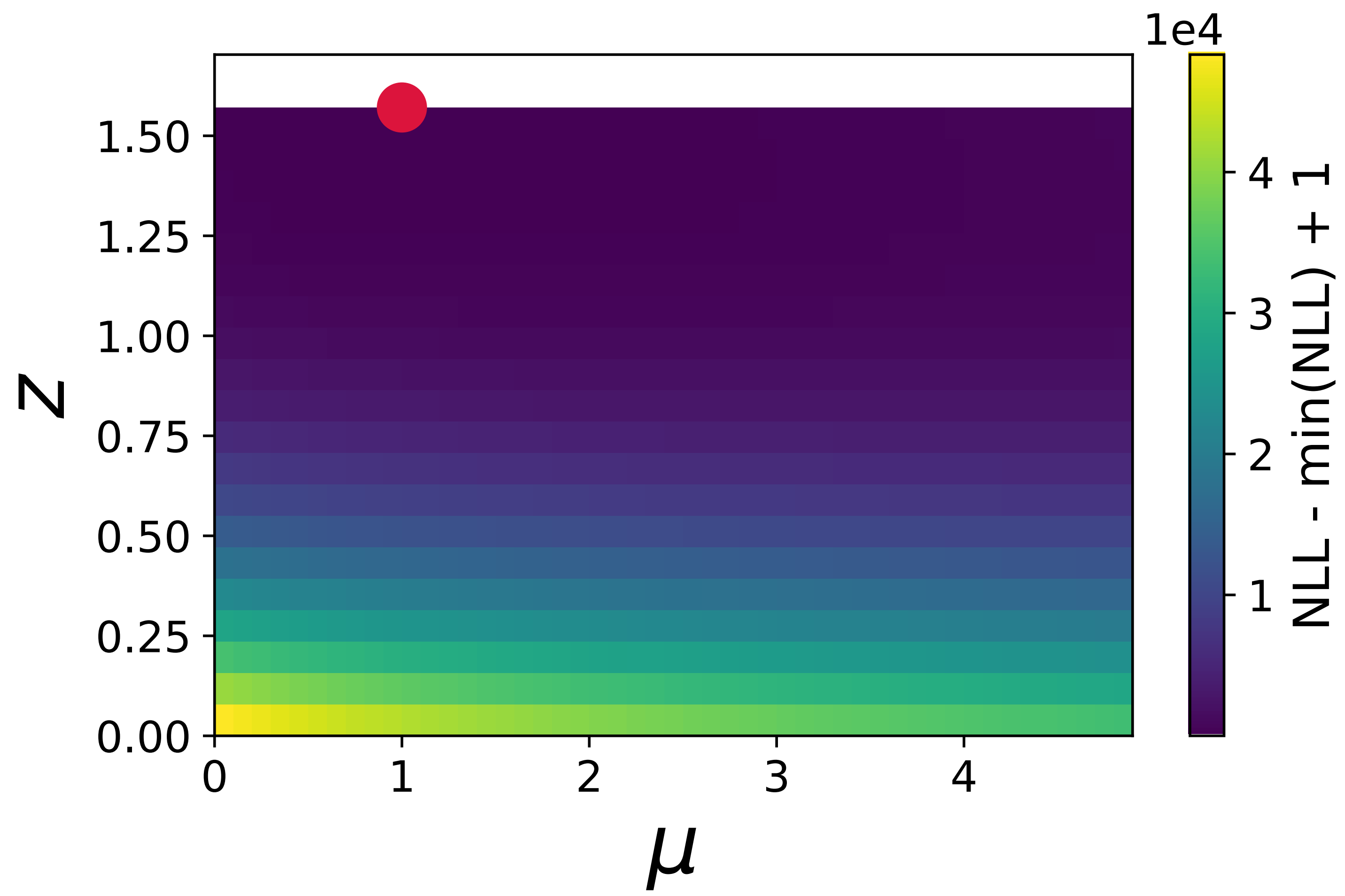
classifier is aware of possible values of z

final evaluation on Profiled Likelihood



Uncertainty-aware classifier gives optimal performance





$\mu = 1$ data at: $z = \dfrac{\pi}{2}$

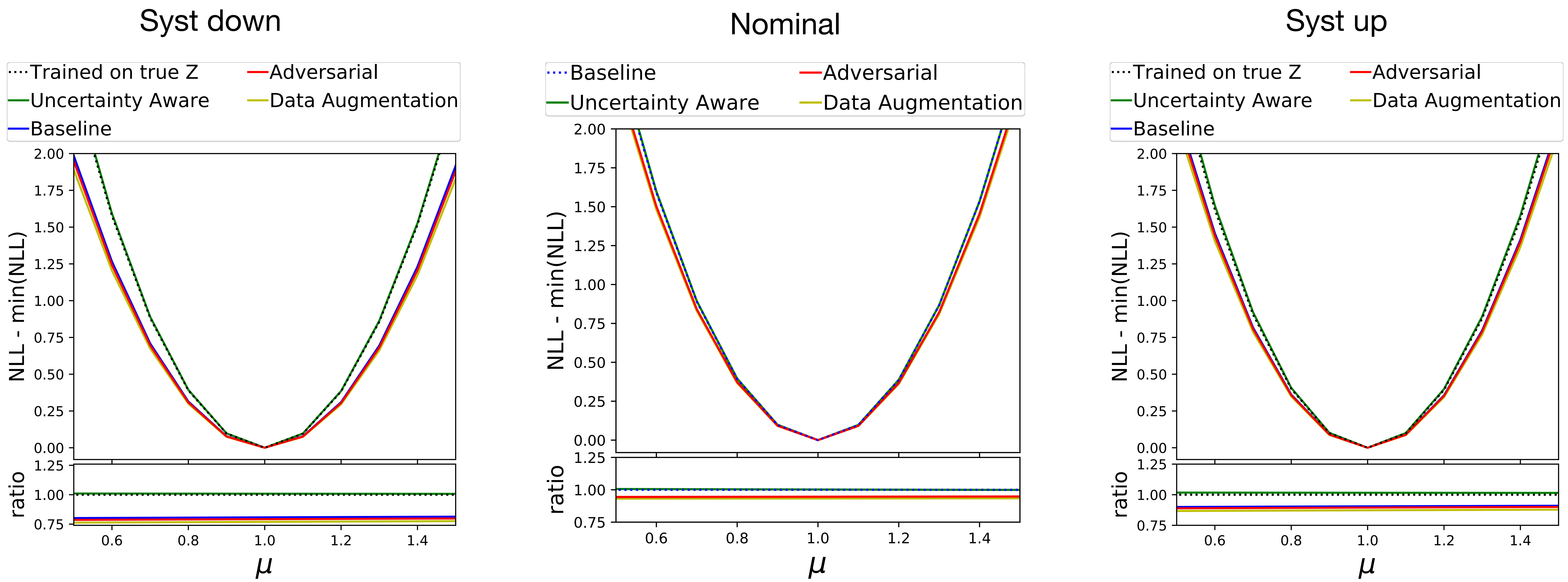
$z = \dfrac{\pi}{4}$ (left), uncertainty-aware (right)

U-aware classifier, $H \to \tau\tau$

Ghosh A., Nachman B., Whiteson D., arXiv:2105.08742

Physics example: Higgs decay to $\tau\tau$

z energy scale of τ_{had} main nuisance parameter



Looking for the optimal metric

CaloFlow, arXiv2106.05285
L2LFlows, arXiv:2302.11594

Data space metrics:

look directly at low-level input distribution

curse of dimensionality, sensitive to non-physical features

High-level observables:

select physically meaningful features

features selection, biased

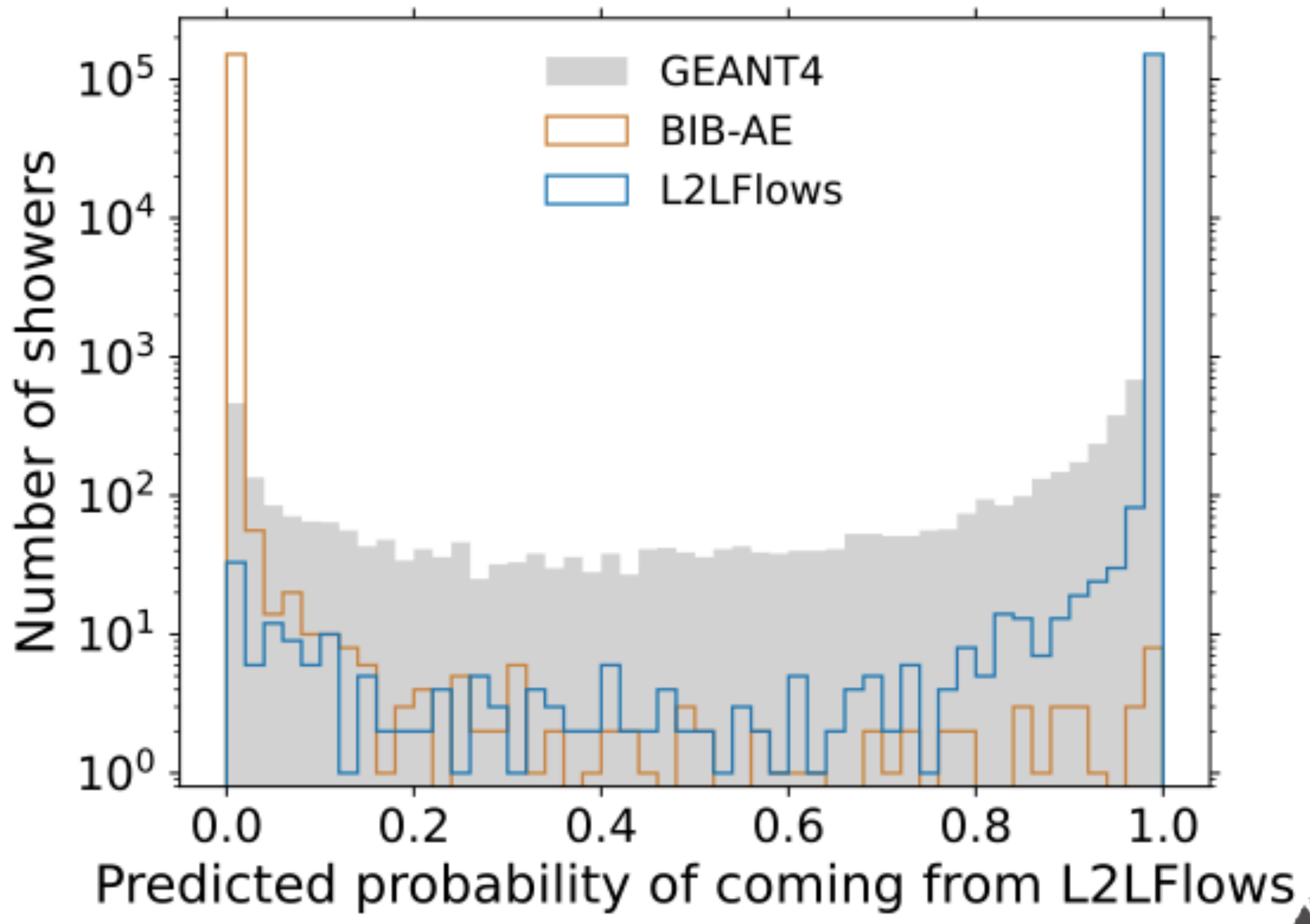
How to check for a faithful reproduction of $p_{data}(x)$?

- train a classifier between generated and true samples
- can the model fool the classifier?

Yes → $p_{gen} = p_{data}$

No → study classifier output

Alternative?



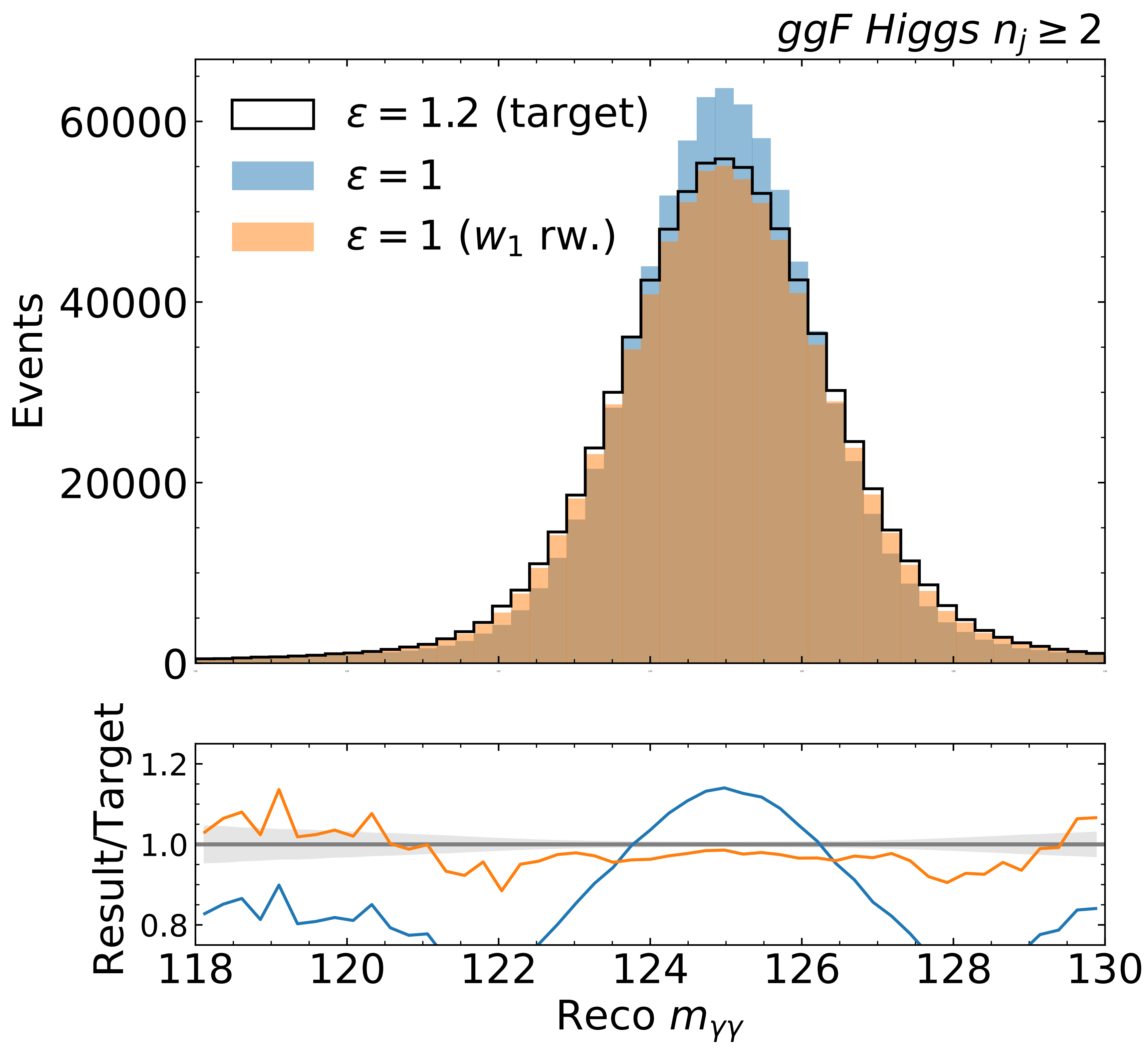
Unbinned Profiled Unfolding

Chan J., Nachman B., arXiv:2302.05390

Train two reweighting functions:

w_0 reweights from unfolded distribution to simulation

$w_1 = \dfrac{p_\theta(R\,|\,T)}{p_{\theta_0}(R\,|\,T)}$ takes into account experimental NPs ^{w_0}



Recent paper on un-binned profiled unfolding

Summary

Estimating model uncertainties:

- deep ensembles: faithful and reliable, with an awful scaling
- Laplace approximation: local, applicable to trained models
- Bayesian Neural Networks: local, active learning, drop-in in most networks

Summary

validation in a controlled environment

MLE with Neural Networks for POIs estimation

given reliable and unbiased data

Regression (e.g. NNPDF):

- closure tests
- prior independent analysis
- Bayesian regression
- boosted training
- calibration

Summary

Classification:

- Optimality — optimal summary statistic for 2-mixture models
- Effect of nuisance parameters — NPs are under control and optimality is preserved
- Preserving optimality
- performance vs resilience — study optimal performance/resilience in simulations vs data

Summary

Controlling generative networks is not easy

open problem: optimal metric

get an error on the density estimate

use weights to improve generator

Generation:

- study reproduction of pd
- optimal metric
- uncertainties from the network
- “active reweighting”