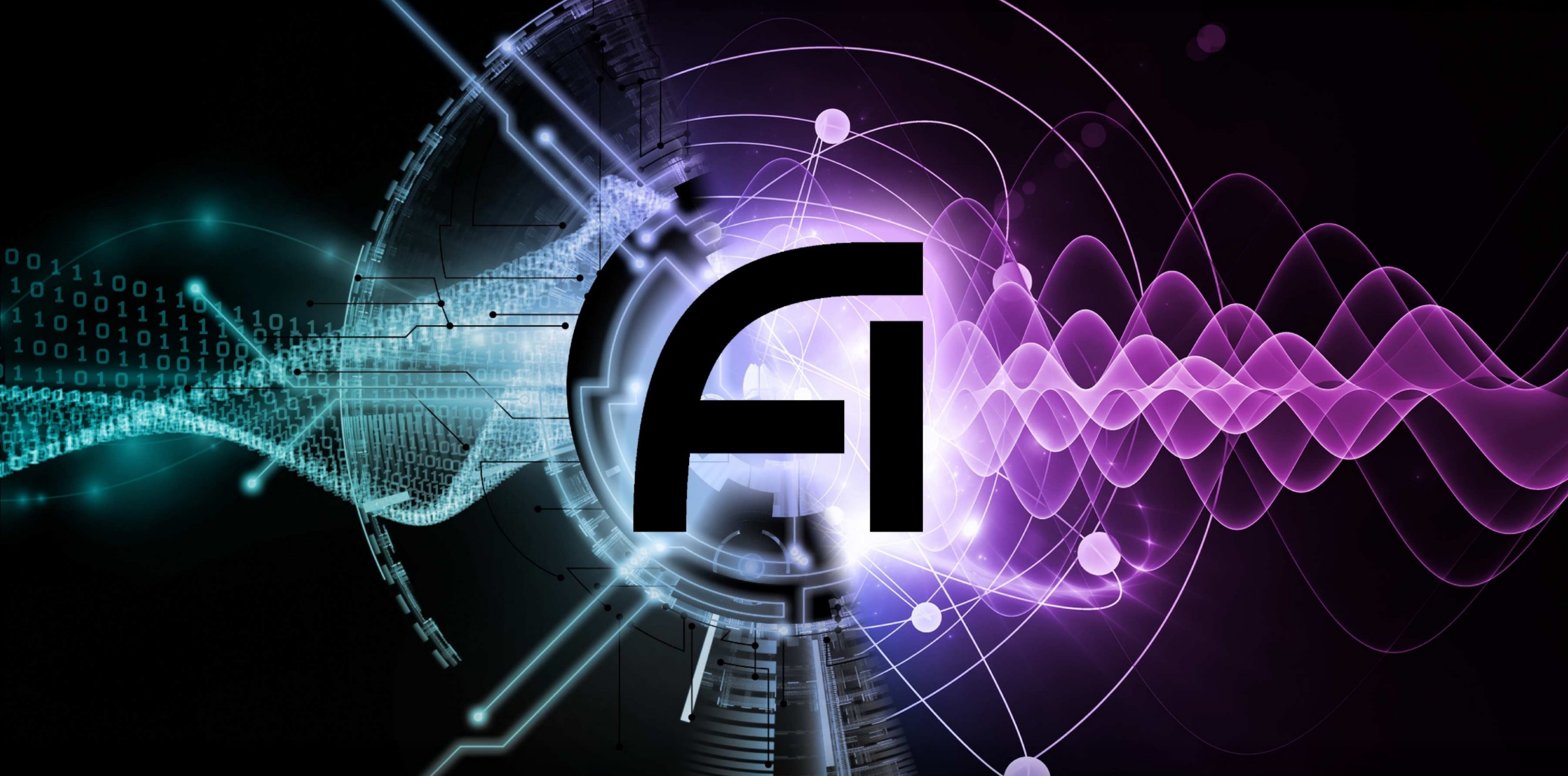
Learning Uncertainties the Frequentist Way

Jesse Thaler

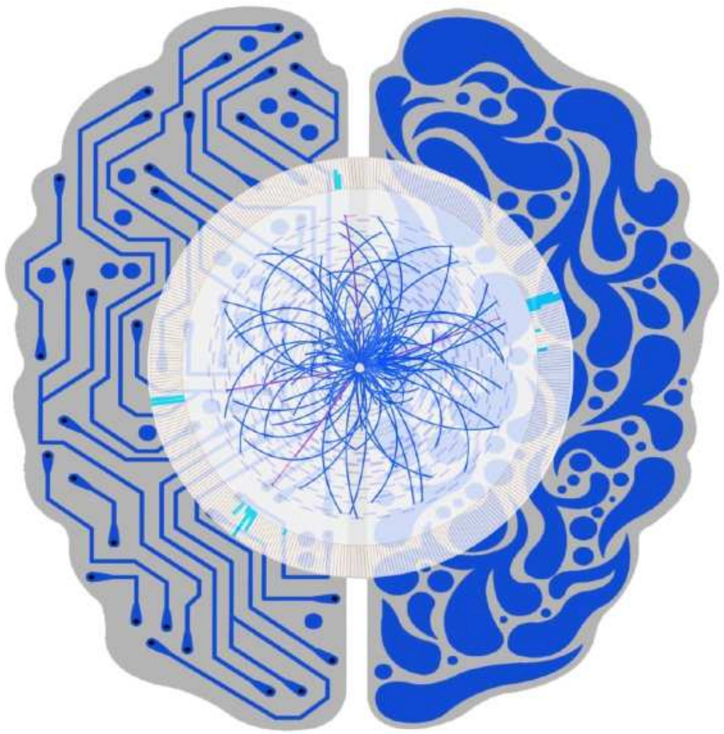
PHYSTAT Seminar — January 25, 2023

The NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI /aɪ-faɪ/ iaifi.org)



Advance physics knowledge — from the smallest building blocks of nature to the largest structures in the universe — and galvanize AI research innovation

The NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI /aɪ-faɪ/ iaifi.org)



Infuse physics intelligence into artificial intelligence

Machine learning that incorporates first principles, best practices, and domain knowledge from fundamental physics

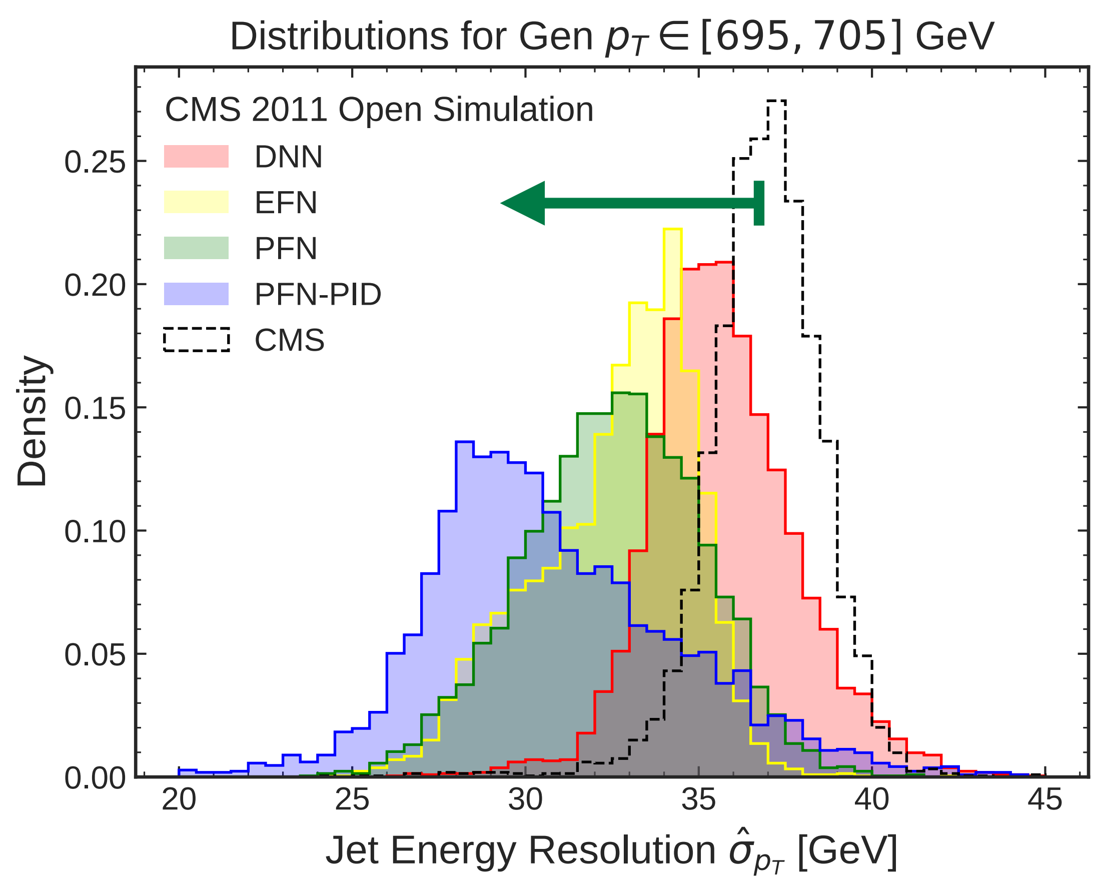
Advance physics knowledge — from the smallest building blocks of nature to the largest structures in the universe — and galvanize AI research innovation

What is “Physics Intelligence”?

One key aspect:
Making scientific decisions in the presence of uncertainties

Machine Learning to Quantify Uncertainties

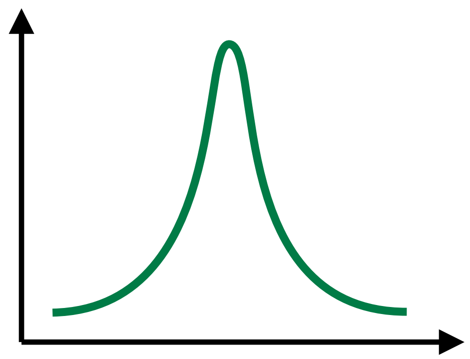
When used correctly, machine learning is a fantastic strategy to incorporate certain kinds of uncertainties



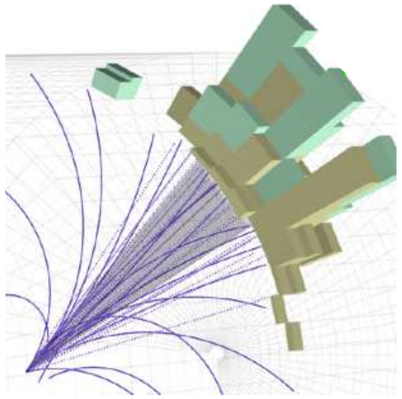
Today's talk: Quantifying and improving experimental "resolution" using our Gaussian Ansatz

[Gambhir, Nachman, JDT, PRL 2022]
[see also Gambhir, Nachman, JDT, PRD 2022]

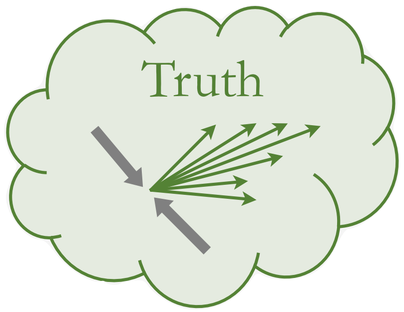
Outline



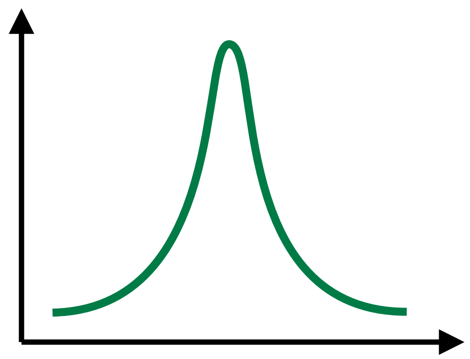
Learning and Uncertainties



Correlation for Calibration



The Next Frontier for UQ in HEP/ML



Learning and Uncertainties

Correlation for Calibration

The Next Frontier for UQ in HEP/ML

Disclaimer

I am a statistics novice, and I am still learning how to speak the language

For the purpose of this talk:

Bayesian Inference:	Making scientific decisions with a probabilistic interpretation (casino); Requires choice of priors
Frequentist Inference:	Making scientific decisions without reference to priors; I'm still amazed this is possible!

I wish I had a formal education in these topics…

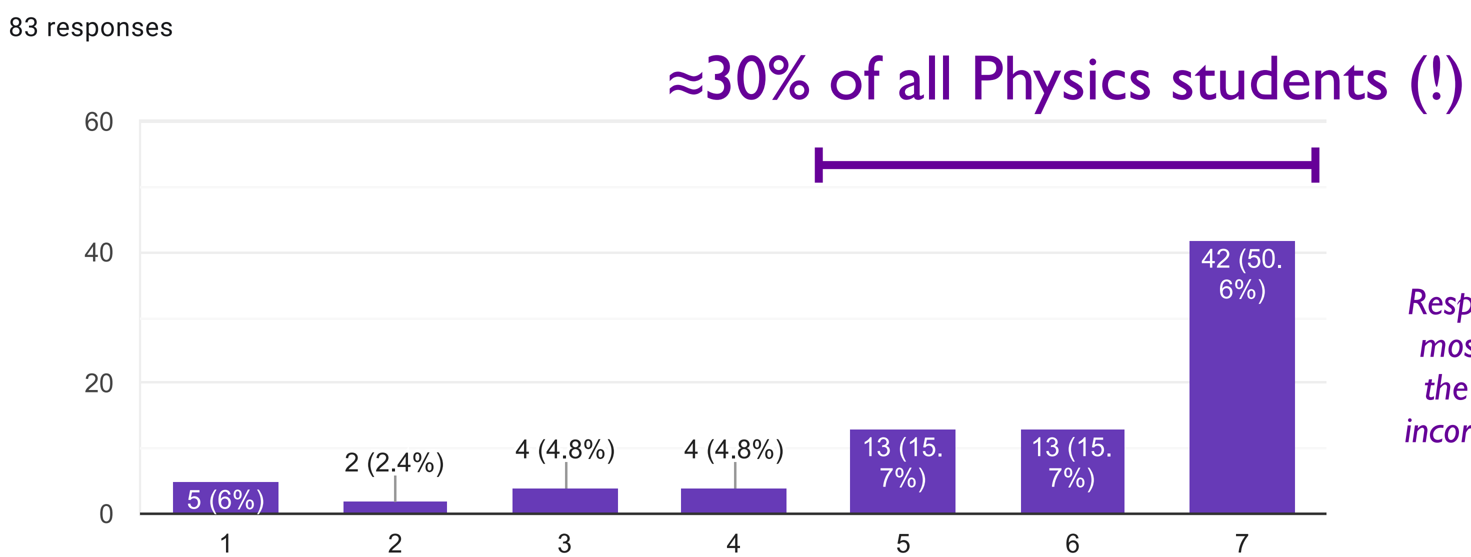
New! PhD in Physics, Statistics & Data Science

≈ Physics PhD + 4 courses (probability, statistics, computation, data analysis)

MIT | Physics

MIT STATISTICS + DATA SCIENCE CENTER

How interested would you be in submitting and defending a PhD thesis that uses statistical methods in a substantial way?



Respondent #11: "I think ML is the most important thing happening in the world right now and should be incorporated into any STEM degree."

Congratulations,
Dr. Constantin Weisser!
(March 30, 2021)

MIT PhysSDS PhD Co-Chairs: JDT & Mike Williams
[https://physics.mit.edu/academic-programs/graduate-students/psds-phd/]

Stats 101: Two Typical Point Estimates

I measure x_{obs} and want to infer/estimate the parameter θ

Bayesian:
Posterior Mean

$$\theta_{\text{MSE}} = \int d\theta\, \theta\, p(\theta \,|\, x_{\text{obs}})$$

↑ Hmm, why this subscript…

$$\hookrightarrow \; = p(x_{\text{obs}} \,|\, \theta) \frac{p(\theta)}{\int d\theta\, p(x_{\text{obs}} \,|\, \theta)\, p(\theta)}$$

likelihood (prior-independent) — prior dependence

Frequentist:
Maximum Likelihood

$$\theta_{\text{MLE}} = \text{argmax}_{\theta}\, p(x_{\text{obs}} \,|\, \theta)$$

Which one of these is more "natural" from the machine learning perspective?

Naive Machine Learning Inference

I have a sample of $\{x, \theta\}$ pairs…

I'll do a similar calculation later in the talk

Training Loss: $$\mathscr{L}_{\mathrm{MSE}} = \left\langle \big(\theta - f(x)\big)^2 \right\rangle$$

Asymptotically: $$\langle\,\rangle \Rightarrow \int dx\, d\theta\, p(x,\theta)$$

Minimum: $$\frac{\delta \mathscr{L}_{\mathrm{MSE}}}{\delta f} = 0 \quad \Rightarrow \quad f(x) = \int d\theta\, \theta\, p(\theta\,|\,x)$$

Euler-Lagrange

Machine Learned: $$\theta_{\mathrm{MSE}} = f(x_{\mathrm{obs}})$$

Same as Bayesian Posterior Mean!

Because machine learning involves training on data, you naively have prior dependence built in

Later this talk: How to nevertheless derive frequentist quantities using clever tricks!

What do we mean by "Uncertainty"?

This word is heavily overloaded, which makes it challenging to discuss "uncertainty quantification"

Uncertainty ≈ Lack of Information

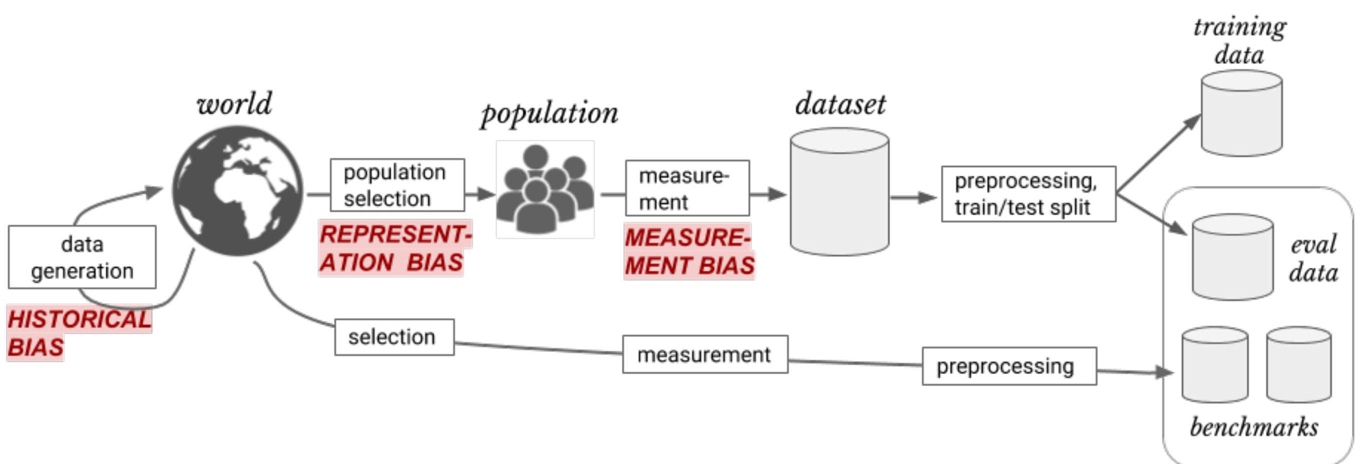
Lack of information about what?

ML Tutorials: Aleatoric (intrinsic randomness)
vs. Epistemic (modeling inadequacies)

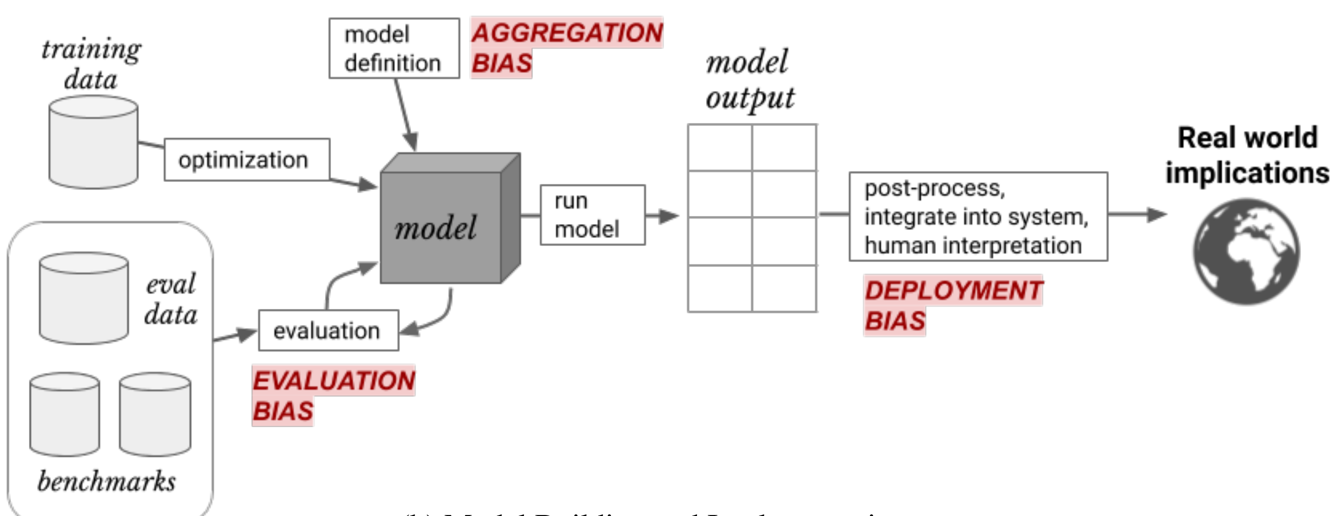
Wikipedia: Parameter, parametric variability, structural, algorithmic, experimental, interpolation, …

Zooming Out: AI Ethics

“A Framework for Understanding Unintended Consequences of Machine Learning”



(a) Data Generation



(b) Model Building and Implementation

1. **Historical bias** arises when there is a misalignment between world as it is and the values or objectives to be encoded and propagated in a model. It is a normative concern with the state of the world, and exists even given perfect sampling and feature selection.
2. **Representation bias** arises while defining and sampling a development population. It occurs when the development population under-represents, and subsequently fails to generalize well, for some part of the use population.
3. **Measurement Bias** arises when choosing and measuring features and labels to use; these are often proxies for the desired quantities. The chosen set of features and labels may leave out important factors or introduce group- or input-dependent noise that leads to differential performance.
4. **Aggregation bias** arises during model construction, when distinct populations are inappropriately combined. In many applications, the population of interest is heterogeneous and a single model is unlikely to suit all subgroups.
5. **Evaluation bias** occurs during model iteration and evaluation. It can arise when the testing or external benchmark populations do not equally represent the various parts of the use population. Evaluation bias can also arise from the use of performance metrics that are not appropriate for the way in which the model will be used.
6. **Deployment Bias** occurs after model deployment, when a system is used or interpreted in inapppropriate ways.

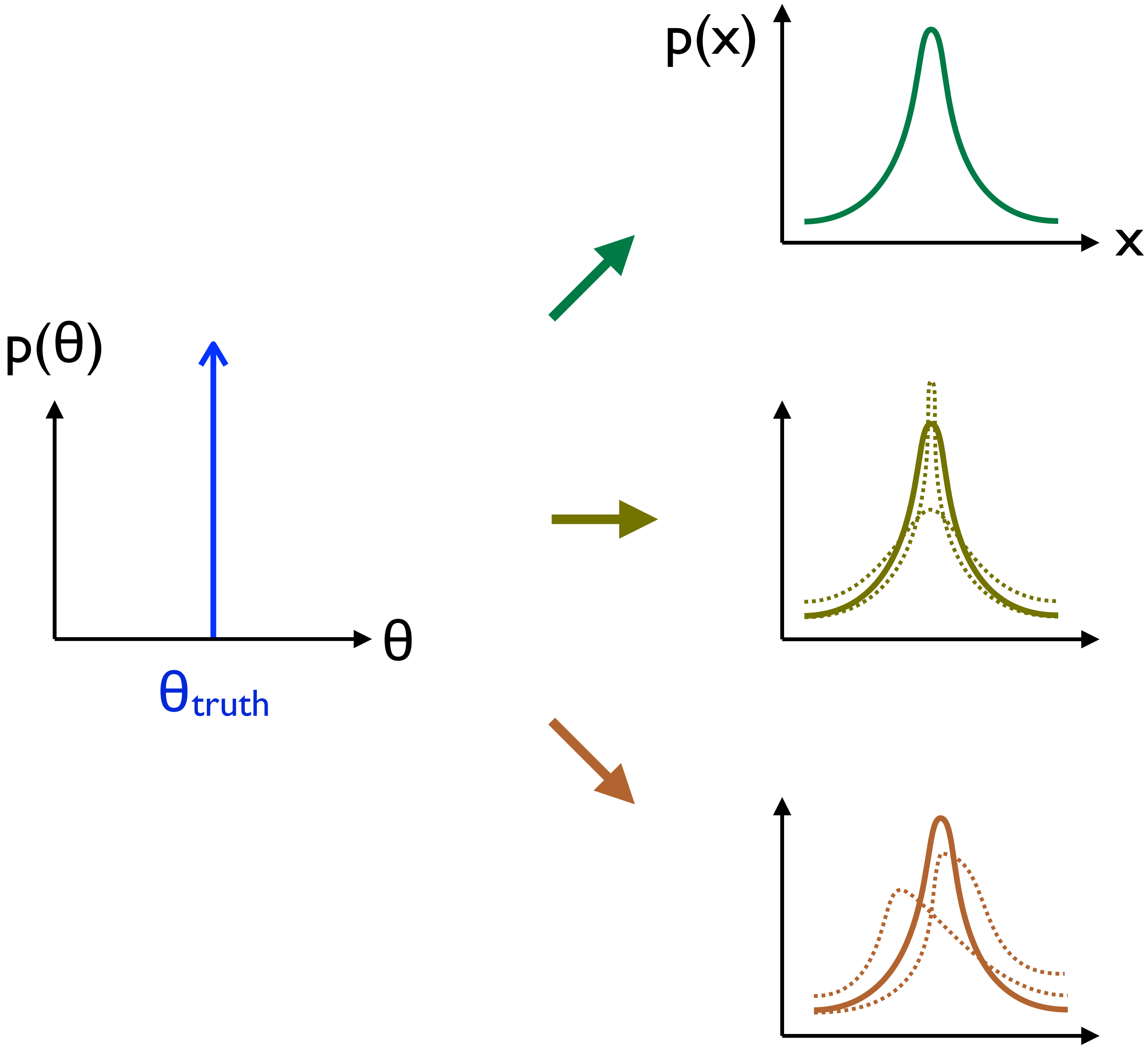
In physics, “bias” ≈ “systematic uncertainty”

[h/t David Kaiser, MIT SERC; Suresh, Guttag, EAAMO 2021]

Three Levels of Uncertainty

Not exhaustive!

All affect scientific decisions



"Resolution"

E.g. gaussian smearing with known σ
Aleatoric, easy to estimate with ML

"Parameter"

E.g. gaussian smearing with unknown σ
Usual HEP meaning of "uncertainty"

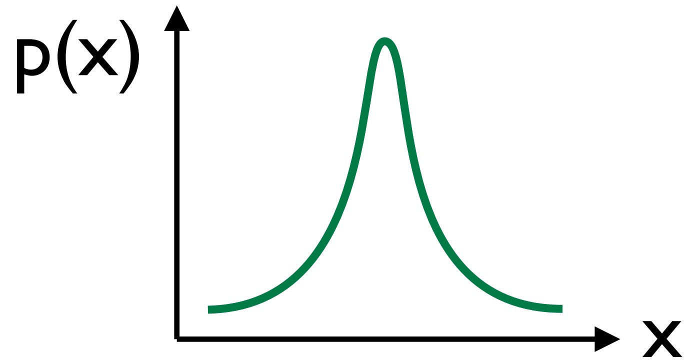
"Model Selection"

E.g. unknown non-gaussian smearing
Epistemic, hard to estimate with ML

Three Levels of Uncertainty

All affect scientific decisions

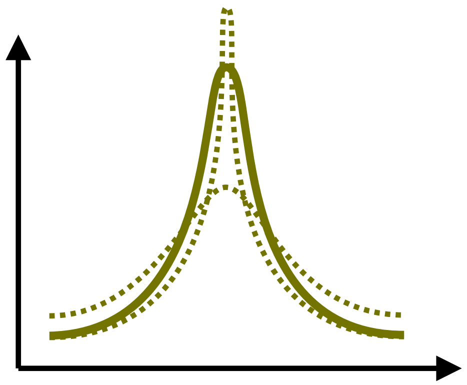
New approach using
Gaussian Ansatz



"Resolution"

E.g. gaussian smearing with known σ
Aleatoric, easy to estimate with ML

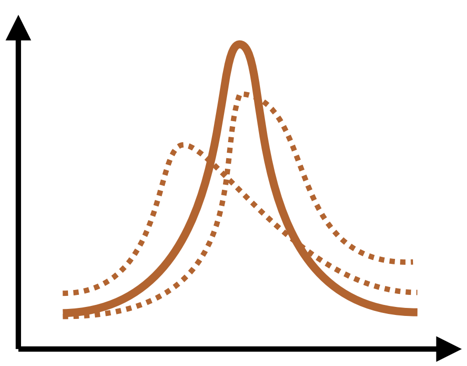
Solvable with enough
coffee/compute



"Parameter"

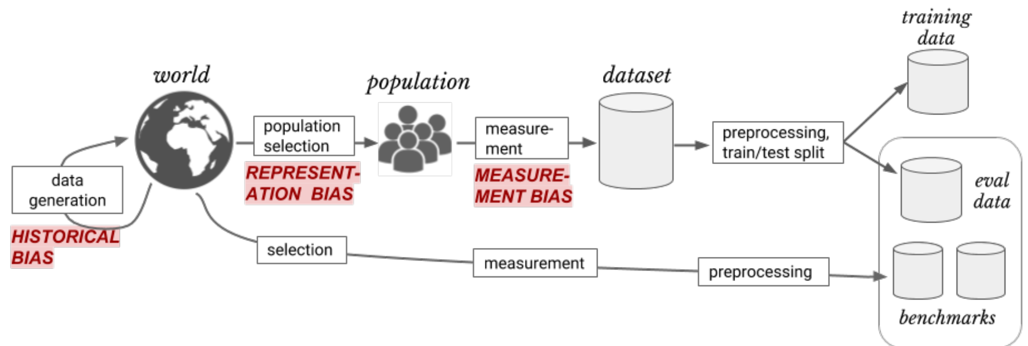
E.g. gaussian smearing with unknown σ
Usual HEP meaning of "uncertainty"

The Frontier for UQ
in ML/HEP

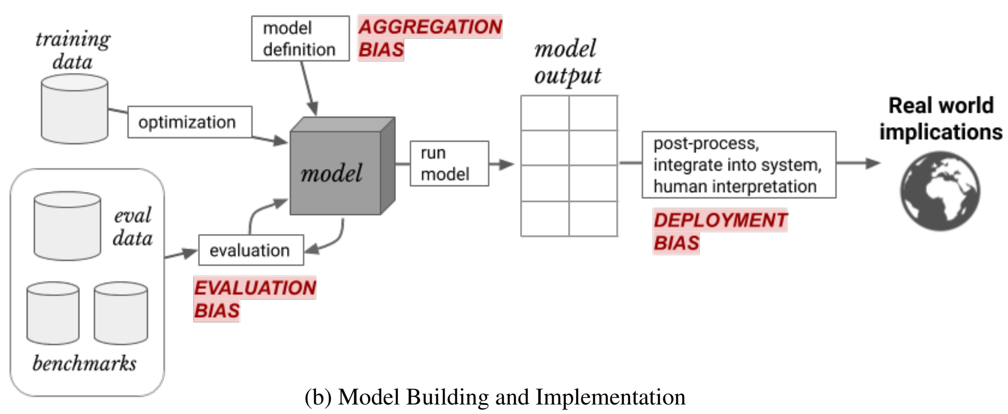


"Model Selection"

E.g. unknown non-gaussian smearing
Epistemic, hard to estimate with ML



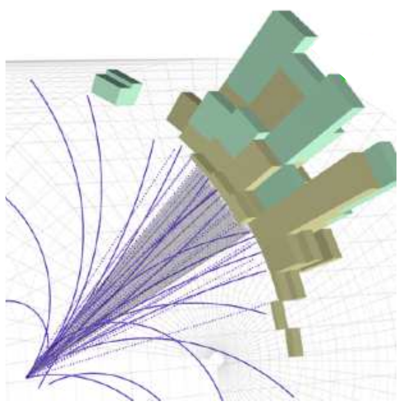
(a) Data Generation



(b) Model Building and Implementation

Uncertainty quantification for machine learning is as multi-faceted as UQ for traditional statistics

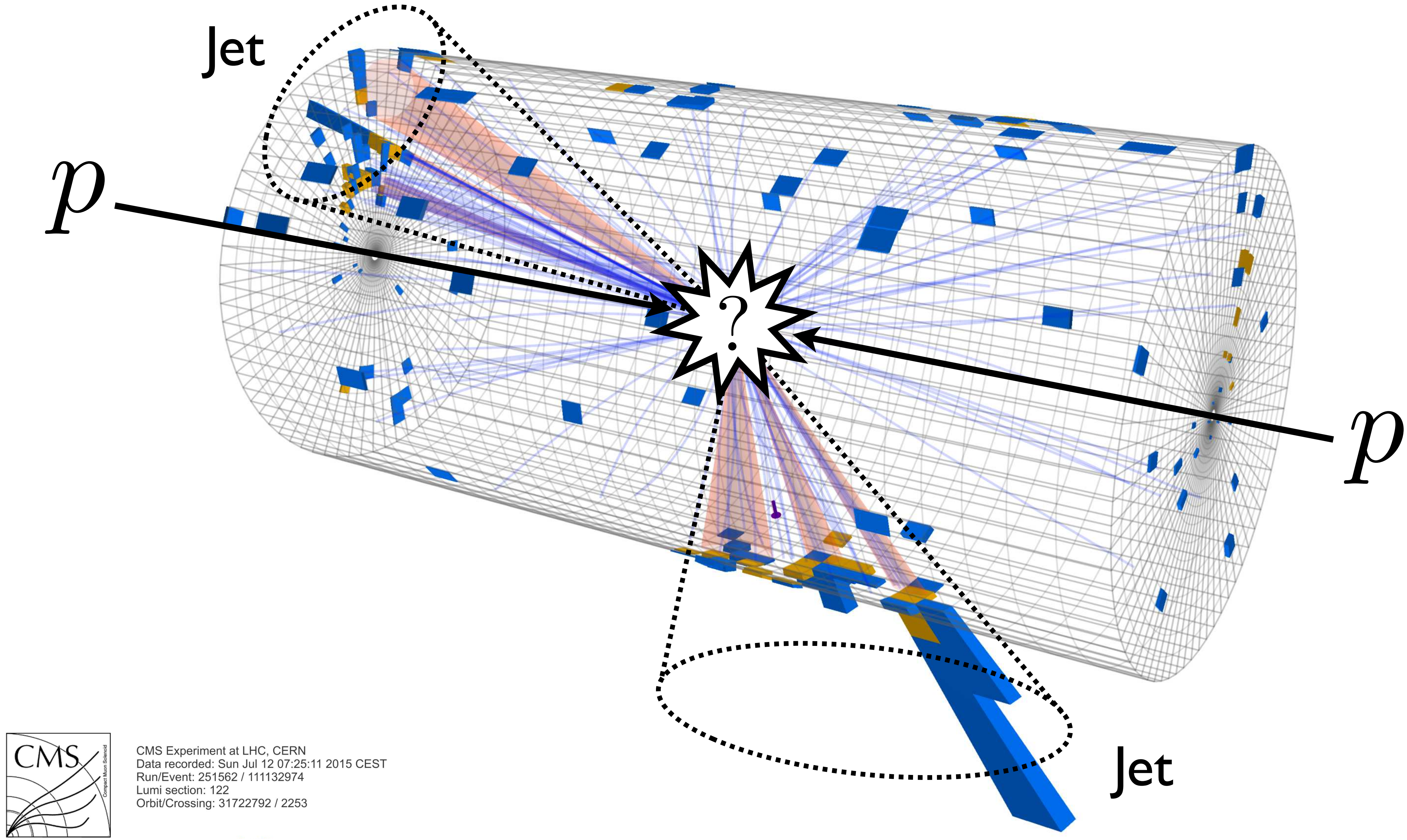
Learning and Uncertainties



Correlation for Calibration

The Next Frontier for UQ in HEP/ML

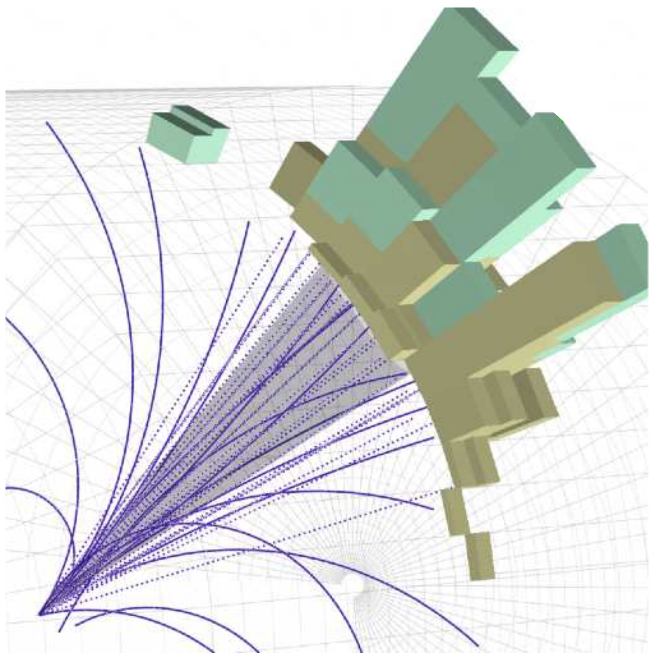
My Research Focus: Jets at the LHC



Simulation-based Calibration

As a theorist, I'm as surprised as you are that I care about this problem

Point estimate for single observation



⇒

Jet p_T

Measured Quantity: x

Inferred Quantity: z

Assumption: $p(x|z)$ is perfectly known through detector simulation

Separate "data-based calibration" is needed if detector is not perfectly modeled

Simulation-based Calibration

Point estimate for single observation

As a theorist, I'm as surprised as you are that I care about this problem

Even if inferred quantities are low dimensional, measured quantities can be high dimensional

By simultaneously measuring more quantities, we can improve the resolution

Assumption: $p(x|z)$ is perfectly known through detector simulation

Separate "data-based calibration" is needed if detector is not perfectly modeled

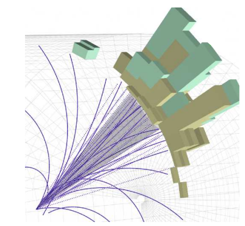
(Aside: Why calibrate when you can just unfold?)

Calibration: Correcting individual observation
Unfolding: Correcting distribution of observations

Folk Theorem: Calibration yields a more diagonal response matrix for better unfolding

Analytic Calibration

If you had perfect knowledge of p(x|z)



x ⇒ Jet p_T

z

independent of prior on p(z)

Log Likelihood: $$T(x,z) = \log \frac{p(x\,|\,z)}{p(x)}$$ + any function of x alone

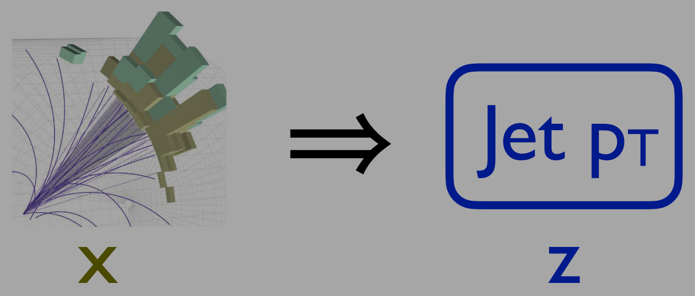
Calibration: $$\hat{z}(x) = \text{argmax}_z \, T(x,z)$$

Resolution: $$[\hat{\sigma}_z^2(x)]_{ij} = -\left[\frac{\partial^2 T(x,z)}{\partial z_i \, \partial z_j}\right]^{-1}\Bigg|_{z=\hat{z}}$$

This is textbook frequentist maximum likelihood calibration

Analytic Calibration

If you had perfect knowledge of p(x|z)



Question: How can we ensure learned T(x,z) is nicely differentiable?

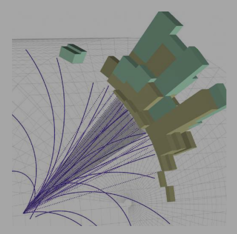
Answer: Use Gaussian Ansatz to set form of T(x,z)!

Resolution: $$[\hat{\sigma}_z^2(x)]_{ij} = -\left[\frac{\partial^2 T(x,z)}{\partial z_i\, \partial z_j}\right]^{-1}\Bigg|_{z=\hat{z}}$$

This is textbook frequentist maximum likelihood calibration

Analytic Calibration

If you had perfect knowledge of p(x|z)



x $\Rightarrow$ Jet p_T

z

independent of prior on p(z)

Log Likelihood: $$T(x,z) = \log \frac{p(x\,|\,z)}{p(x)}$$ + any function of x alone

Question: How can machine learning be used to estimate T(x,z)?

Answer: Estimate mutual information between x and z!

This is textbook frequentist maximum likelihood calibration

Introducing the Gaussian Ansatz

Named because of its resemblance to log of Gaussian likelihood density

Modern machine learning uses differentiable programming, but some activation functions have poorly-behaved derivatives

Second-order Taylor expansion around z = B(x):

$$T(x,z) = A(x) + \big(z - B(x)\big) \cdot D(x) + \frac{1}{2}\big(z - B(x)\big)^T \cdot C(x,z) \cdot \big(z - B(x)\big)$$

Note full z dependence here

Functions A, B, C, and D are parametrized as neural networks

Dots indicate index contractions

[Gambhir, Nachman, JDT, PRL 2022]

Introducing the Gaussian Ansatz

Named because of its resemblance to log of Gaussian likelihood density

Modern machine learning uses differentiable programming, but some activation functions have poorly-behaved derivatives

Second-order Taylor expansion around z = B(x):

$$T(x,z) = A(x) + \cancel{\big(z - B(x)\big) \cdot D(x)} + \frac{1}{2}\big(z - B(x)\big)^T \cdot C(x,z) \cdot \big(z - B(x)\big)$$

Note full z dependence here

Functions A, B, C, and D are parametrized as neural networks

Dots indicate index contractions

No loss of expressivity with this form even if D(x) → 0

[Gambhir, Nachman, JDT, PRL 2022]

Introducing the Gaussian Ansatz

Named because of its resemblance to log of Gaussian likelihood density

Modern machine learning uses differentiable programming, but some activation functions have poorly-behaved derivatives

Second-order Taylor expansion around z = B(x):

$$T(x,z) = A(x) + \cancel{\big(z - B(x)\big) \cdot D(x)} + \frac{1}{2}\big(z - B(x)\big)^T \cdot C(x,z) \cdot \big(z - B(x)\big)$$

Note full z dependence here

Easy to read off calibration!

$$\hat{z}(x) = B(x) \qquad \hat{\sigma}_z^2(x) = -\Big[C(x, B(x))\Big]^{-1}$$

[Gambhir, Nachman, JDT, PRL 2022]

Re-introducing Mutual Information

Using Donsker-Varadhan representation of Kullback-Leibler divergence

Mutual Information:

a.k.a. KL divergence between joint distribution and product of marginals

$$I(X;Z) = \int dx\, dz\, p(x,z)\, \log \frac{p(x,z)}{p(x)\, p(z)}$$

DV Representation:

$$\mathscr{L}_{\mathrm{DVR}}[T] = -\left(\mathbb{E}_{P_{XZ}}[T] - \log \mathbb{E}_{P_X \otimes P_Z}[e^T]\right)$$

Bound:

$$I(X;Z) \geq -\min_T \mathscr{L}_{\mathrm{DVR}}[T]$$

Saturated When:

This is what we need for calibration!

$$T(x,z) = \log \frac{p(x,z)}{p(x)\, p(z)}$$

+ any constant

[Donsker, Varadhan, CPAM 1975; used in Belghazi, Baratin, Rajeswar, Ozair, Bengio, Courville, Hjelm, ICML 2018]

Bottom Line:

To do frequentist calibration, all* you have to do is input the Gaussian Ansatz for $T(x,z) = \log p(x|z)/p(x)$ and use machine learning to minimize the DVR loss

See ACORE-LFI for alternative frequentist approach

[Dalmasso, Masserano, Zhao, Izbicki, Lee, arXiv 2022]

Bottom Line:

To do frequentist calibration, all* you have to do is input the Gaussian Ansatz for $T(x,z) = \log p(x|z)/p(x)$ and use machine learning to minimize the DVR loss

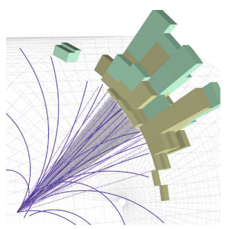
See ACORE-LFI for alternative frequentist approach

[Dalmasso, Masserano, Zhao, Izbicki, Lee, arXiv 2022]

If you are anything like me, then the above derivation is not very satisfying…

Results using Public Collider Data

Simulated jets with $p_T \in [695, 705]$ GeV from CMS



x ⇒ Jet p_T

z

$$\hat{z}(x) = \text{argmax}_z \, T(x, z)$$

$$[\hat{\sigma}_z^2(x)]_{ij} = - \left[\frac{\partial^2 T(x,z)}{\partial z_i \, \partial z_j} \right]^{-1} \Bigg|_{z=\hat{z}}$$

Model	Mean $\hat{p}_T$ [GeV]	Mean $\hat{\sigma}_{p_T}$ [GeV]	$I(X;Z)$
DNN	698 ± 37.7	35.7 ± 2.1	1.23
EFN	695 ± 37.3	32.6 ± 2.3	1.26
PFN	697 ± 36.9	32.5 ± 2.5	1.27
PFN-PID	695 ± 35.1	$\mathbf{30.8 \pm 3.6}$	**1.32**
CMS 2011	695 ± 38.4	36.9 ± 1.7	–

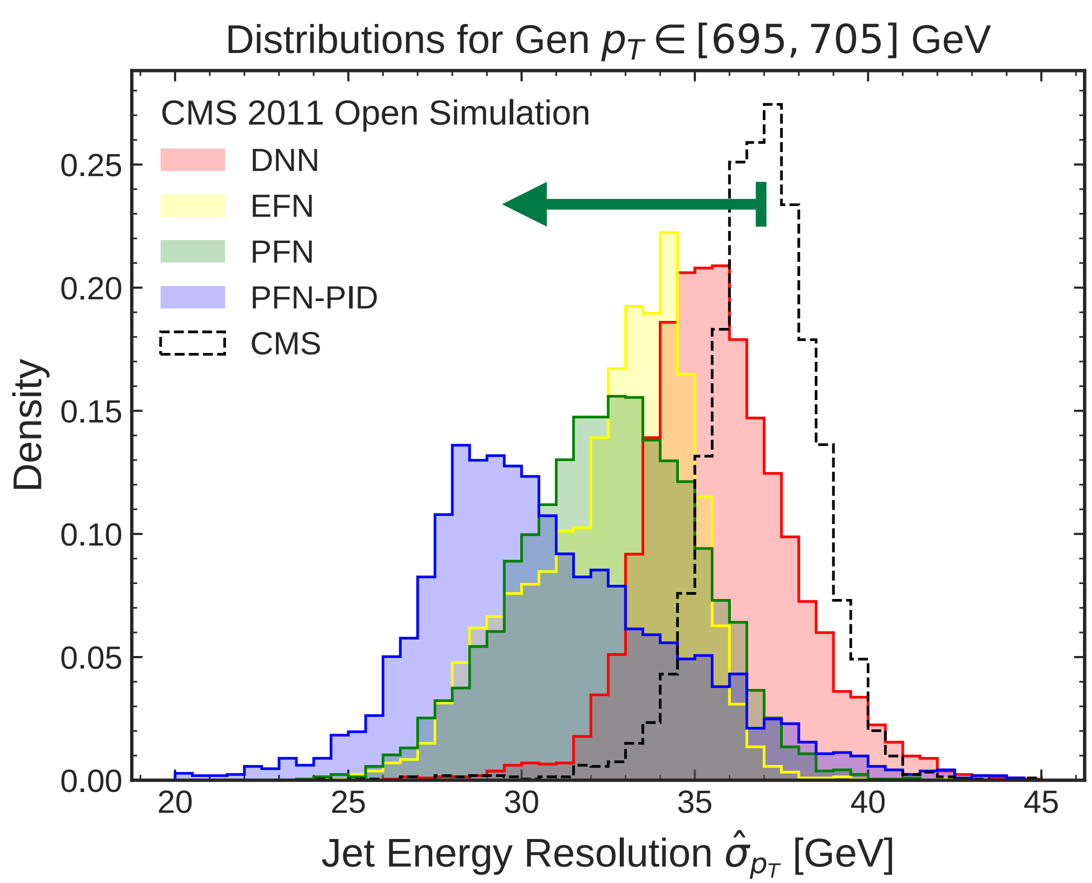
More expressive model ⇒ increasing MI ⇒ improved resolution

Gains primarily from using jet substructure to assist jet calibration

[Gambhir, Nachman, JDT, PRL 2022;
using CMS Open Data processed by Komiske, Mastandrea, Metodiev, Naik, JDT, PRD 2020]

Results using Public Collider Data

Simulated jets with $p_T \in [695, 705]$ GeV from CMS



Gaussian Ansatz:

Using high-dimensional inference to improve experimental resolution

More expressive model ⇒ increasing MI ⇒ improved resolution

Gains primarily from using jet substructure to assist jet calibration

[Gambhir, Nachman, JDT, PRL 2022;
using CMS Open Data processed by Komiske, Mastandrea, Metodiev, Naik, JDT, PRD 2020]

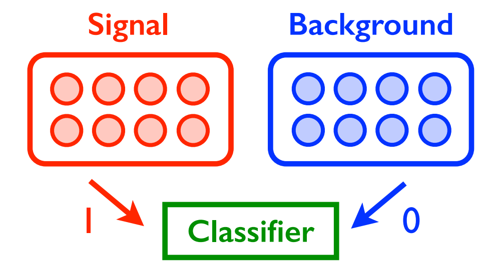
Extracting some broader lessons

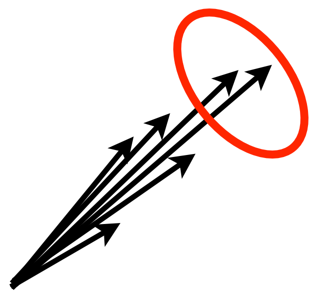
Gaussian Ansatz $\Rightarrow$ Model Engineering

DV Representation $\Rightarrow$ Loss Engineering

Quark/Gluon Classification

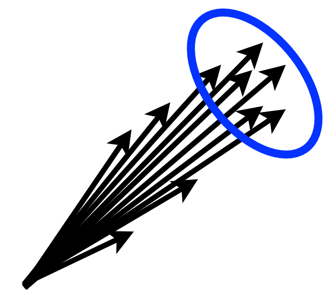
"Hello, World!" of Jet Physics



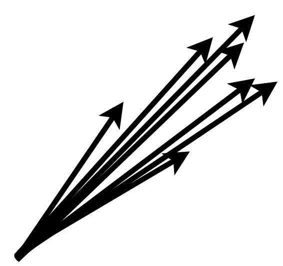


Quark
$C_q = 4/3$

vs.



Gluon
$C_g = 3 = 9/3$

Find $h\Big($  $\Big)$ such that

$$h(\textbf{Quark}) = 1$$

$$h(\textbf{Gluon}) = 0$$

Best you can do: $$h(\mathcal{J}) = \left(1 + \frac{p(\mathcal{J}|\mathrm{G})}{p(\mathcal{J}|\mathrm{Q})}\right)^{-1}$$

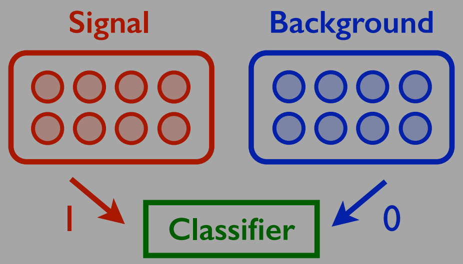
(Neyman-Pearson lemma)

Likelihood ratio yields optimal binary classifier (and vice versa)

[see e.g. Gras, Höche, Kar, Larkoski, Lönnblad, Plätzer, Siódmok, Skands, Soyez, JDT, JHEP 2017;
Komiske, Metodiev, Schwartz, JHEP 2017; Komiske, Metodiev, JDT, JHEP 2018]

Quark/Gluon Classification

"Hello, World!" of Jet Physics



Model Engineering: Find function h that captures known structure of problem

Loss Engineering: Find functional L[h] whose minimum yields desired properties

Best you can do: $h(\mathcal{J}) = \left(1 + \frac{p(\mathcal{J}|\mathrm{G})}{p(\mathcal{J}|\mathrm{Q})}\right)^{-1}$

(Neyman-Pearson lemma)

Likelihood ratio yields optimal binary classifier (and vice versa)

[see e.g. Gras, Höche, Kar, Larkoski, Lönnblad, Plätzer, Siódmok, Skands, Soyez, JDT, JHEP 2017; Komiske, Metodiev, Schwartz, JHEP 2017; Komiske, Metodiev, JDT, JHEP 2018]

Re-introducing the Likelihood Ratio Trick

Key example of simulation-based inference

Goal: Estimate p(x) / q(x)

Training Data: Finite samples P and Q

Learnable Function: f(x) parametrized by, e.g., neural networks

Loss Function(al): $$L = -\langle \log f(x) \rangle_P + \langle f(x) - 1 \rangle_Q$$

Asymptotically: $$\arg\min_{f(x)} L = \frac{p(x)}{q(x)}$$ *Likelihood ratio*

$$-\min_{f(x)} L = \int dx\, p(x) \log \frac{p(x)}{q(x)}$$ *Kullback–Leibler divergence*

[see e.g. D'Agnolo, Wulzer, PRD 2019; simulation-based inference in Cranmer, Brehmer, Louppe, PNAS 2020; relation to f-divergences in Nguyen, Wainwright, Jordan, AoS 2009; Nachman, JDT, PRD 2021]

Re-introducing the Likelihood Ratio Trick

Key example of simulation-based inference

Asymptotically, same structure as Lagrangian mechanics!

Action: $$L = \int \mathrm{d}x \, \mathcal{L}(x)$$

Lagrangian: $$\mathcal{L}(x) = -p(x) \log f(x) + q(x) \big(f(x) - 1 \big)$$

Euler-Lagrange: $$\frac{\partial \mathcal{L}}{\partial f} = 0$$

Solution: $$f(x) = \frac{p(x)}{q(x)}$$

Requires shift in focus from solving problems to specifying problems

[see e.g. D'Agnolo, Wulzer, PRD 2019; simulation-based inference in Cranmer, Brehmer, Louppe, PNAS 2020; relation to f-divergences in Nguyen, Wainwright, Jordan, AoS 2009; Nachman, JDT, PRD 2021]

The Landscape of Losses

$$L[f] = -\int dx \left(p(x|\theta_A)\, A(f(x)) + p(x|\theta_B)\, B(f(x)) \right)$$

Loss Name	$A(f)$	$B(f)$	$\text{argmin}_f\, L[f]$	Integrand of $-\min_f L[f]$	Related Divergence/Distance
Binary Cross Entropy	$\log f$	$\log(1-f)$	$\frac{p_A}{p_A+p_B}$	$p_A \log \frac{p_A}{p_A+p_B} + (A \leftrightarrow B)$	$2\big(\text{Jensen-Shannon} - \log 2\big)$
Mean Squared Error	$-(1-f)^2$	$-f^2$	$\frac{p_A}{p_A+p_B}$	$-\frac{p_A p_B}{p_A+p_B}$	$\frac{1}{2}\big(\text{Triangular} - 1\big)$
Square Root	$\frac{-1}{\sqrt{f}}$	$-\sqrt{f}$	$\frac{p_A}{p_B}$	$-2\sqrt{p_A p_B}$	$2\big(\text{Hellinger}^2 - 1\big)$
Maximum Likelihood Cl.	$\log f$	$1-f$	$\frac{p_A}{p_B}$	$p_A \log \frac{p_A}{p_B}$	Kullback–Leibler

We have considerable flexibility in choosing the loss

[table from Nachman, JDT, PRD 2021]

The Trick Behind the DVR Loss

"Local": $$\mathscr{L}_{\text{MLC}}[T] = -\int dx\, p(x)\, T(x) \quad + \quad \int dx\, q(x)\, \left(e^{T(x)} - 1\right)$$

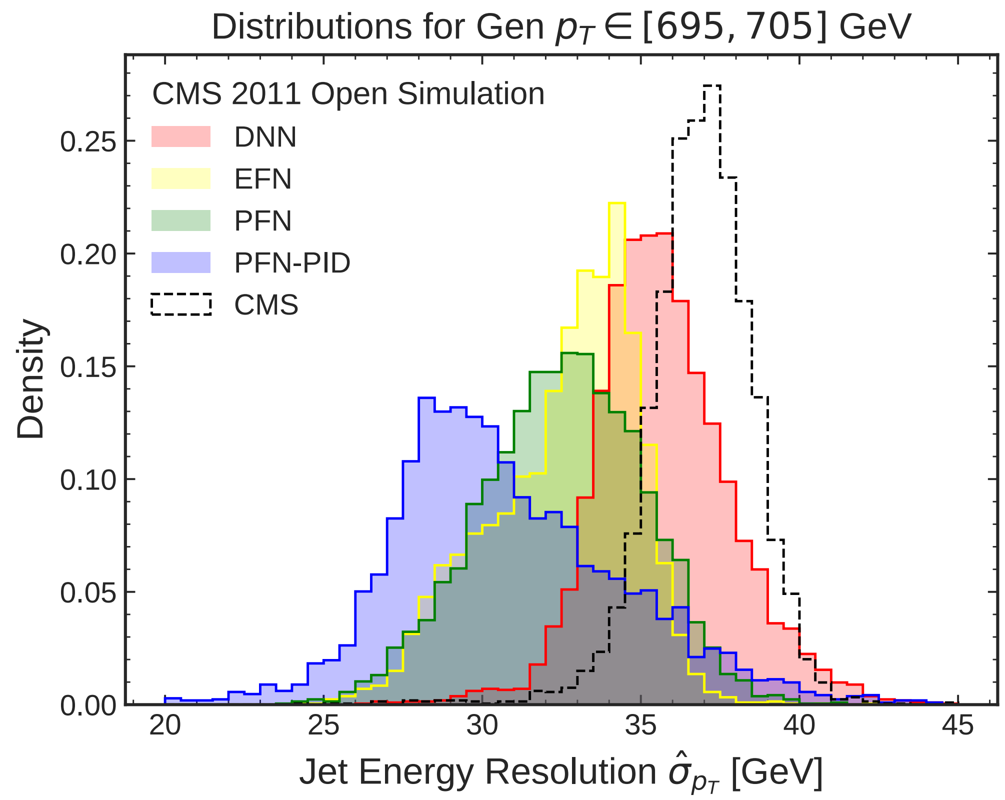
"Non-local": $$\mathscr{L}_{\text{DVR}}[T] = -\int dx\, p(x)\, T(x) + \log \int dx\, q(x)\, \left(e^{T(x)}\right)$$

$$\frac{\delta \mathscr{L}_{\text{MLC}}}{\delta T} = -\,p(x) \quad + \quad q(x)\, e^{T(x)} \qquad \Rightarrow \qquad T(x) = \log \frac{p(x)}{q(x)}$$

$$\frac{\delta \mathscr{L}_{\text{DVR}}}{\delta T} = -\,p(x) + \frac{q(x)\, e^{T(x)}}{\int dy\, q(y)\, e^{T(y)}} \qquad \Rightarrow \qquad T(x) = \log \frac{p(x)}{q(x)} + c$$

This can be set to any constant!

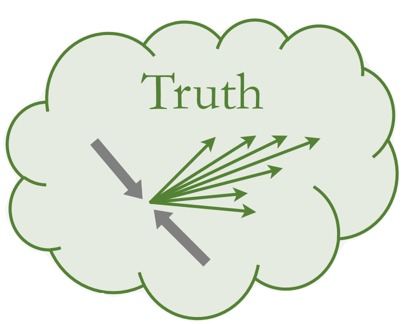
DVR provides a stricter bound on KL divergence than MLC, which is why DVR is preferred for our calibration purposes



To learn (resolution-style) uncertainties the frequentist way, first use simulation-based inference to extract likelihoods

Learning and Uncertainties

Correlation for Calibration

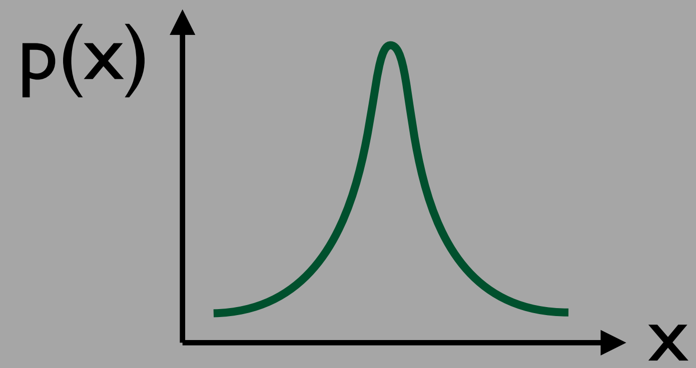


The Next Frontier for UQ in HEP/ML

Three Levels of Uncertainty

All affect scientific decisions

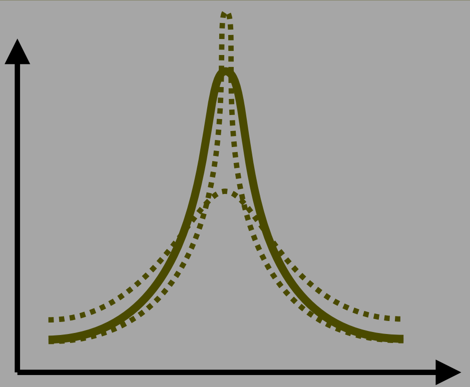
New approach using Gaussian Ansatz



"Resolution"

E.g. gaussian smearing with known σ
Aleatoric, easy to estimate with ML

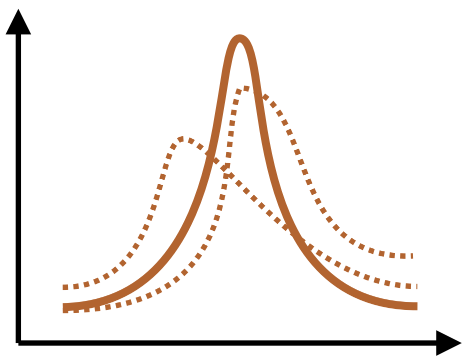
Solvable with enough coffee/compute



"Parameter"

E.g. gaussian smearing with unknown σ
Usual HEP meaning of "uncertainty"

The Frontier for UQ in ML/HEP



"Model Selection"

E.g. unknown non-gaussian smearing
Epistemic, hard to estimate with ML

From Models to Parameters

With enough nuisance parameters,
model selection is "solved" via parameter estimation

Modern machine learning involves setting a huge range of hyperparameters, including those related to initialization and optimization

O(N) parameters means $O(N^2)$ covariance entries

Does model selection even make sense at large N?

Machine learning is worming its way into all aspects of the HEP workflow, increasing the importance of robust UQ

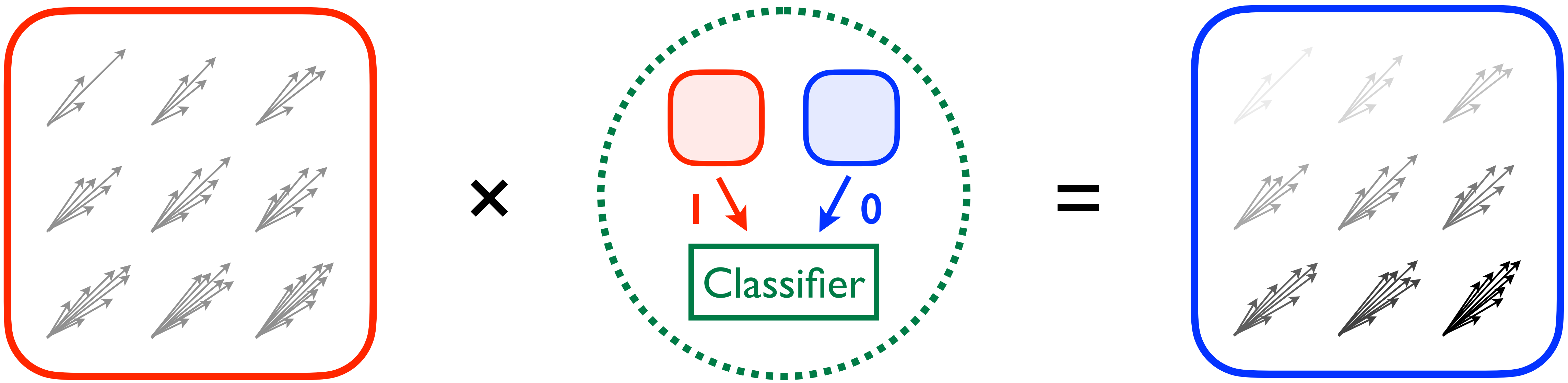
LRT: From Tautology to Essential Tool

$$q(x) \times \frac{p(x)}{q(x)} = p(x)$$

Generate samples according to Q

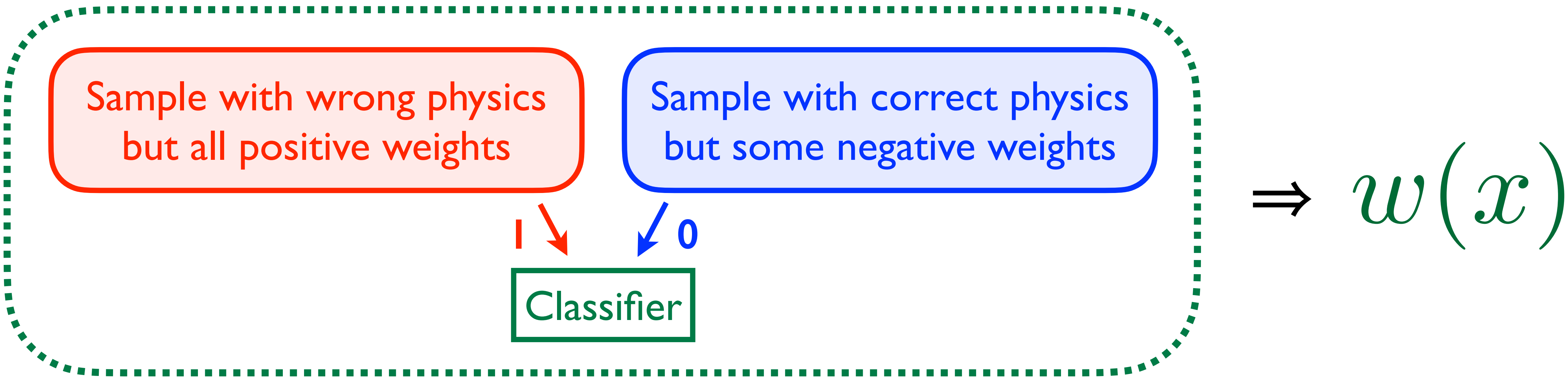
Weight each sample by likelihood ratio

Obtain weighted samples distributed according to P

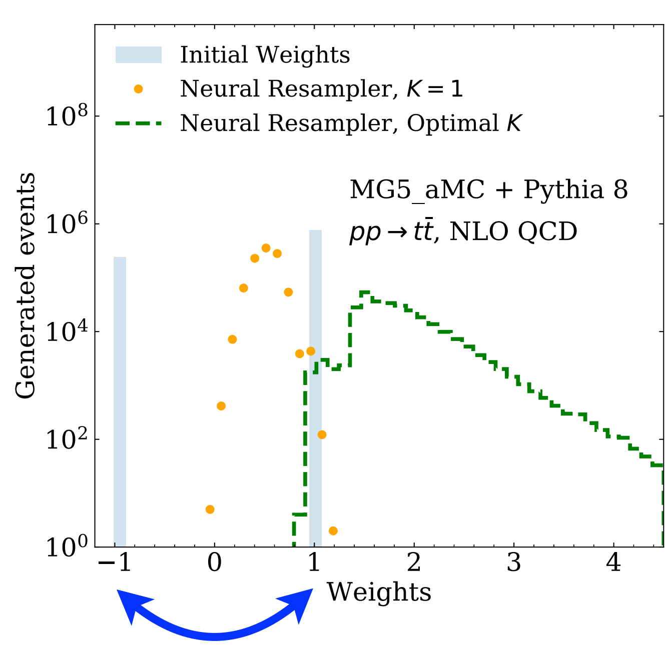


With large enough data samples,
binary classification yields weighted simulation

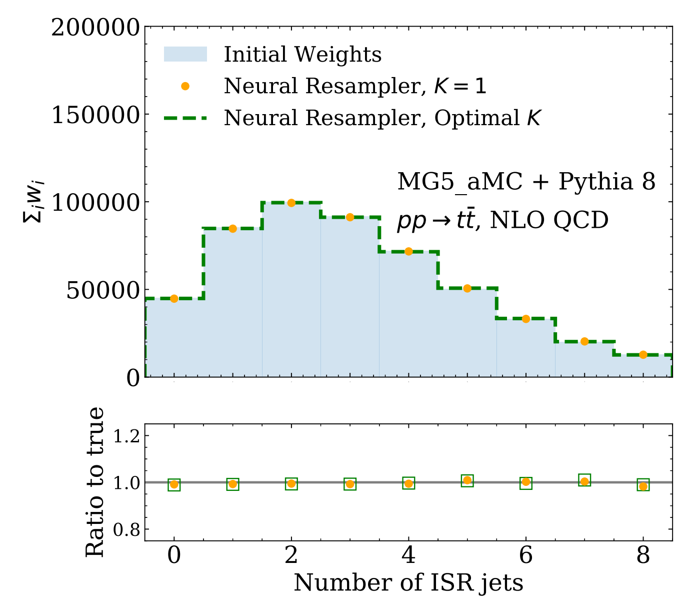
E.g. Neural Resampling for MC Beyond LO



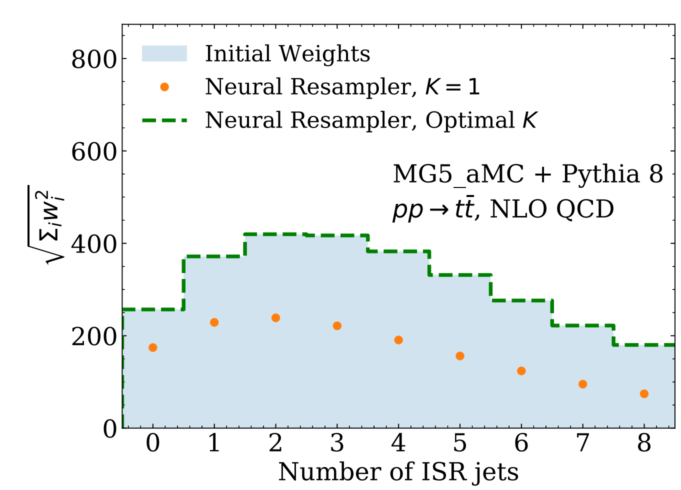
MC@NLO: large weight cancellations



Reweighting recovers desired distribution



Resampling recovers desired uncertainties



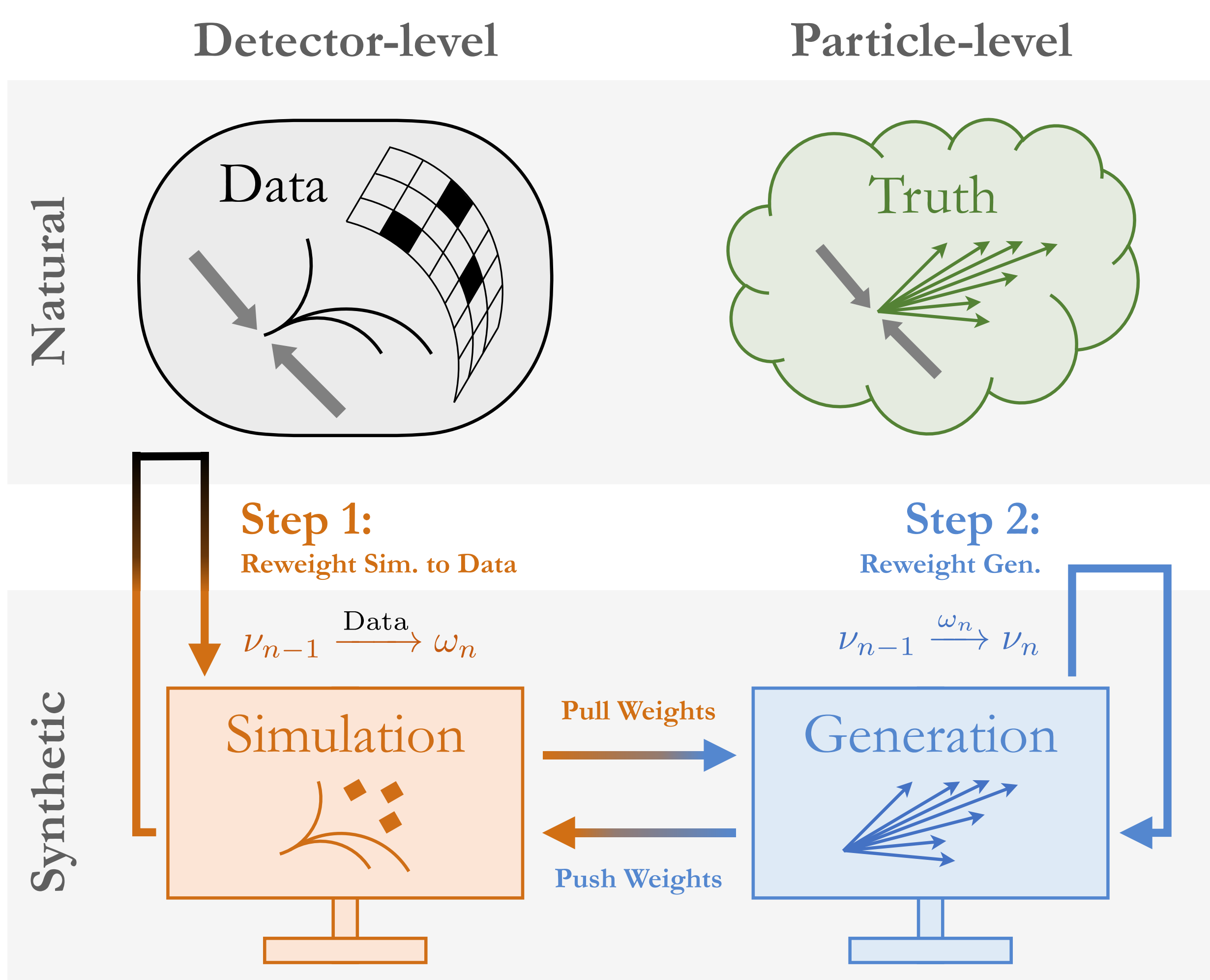
Using custom ML strategy

[Nachman, JDT, PRD 2020; inspired by Andersen, Gutschow, Maier, Prestel, EPJC 2020]

E.g.: Detector Unfolding

OmniFold

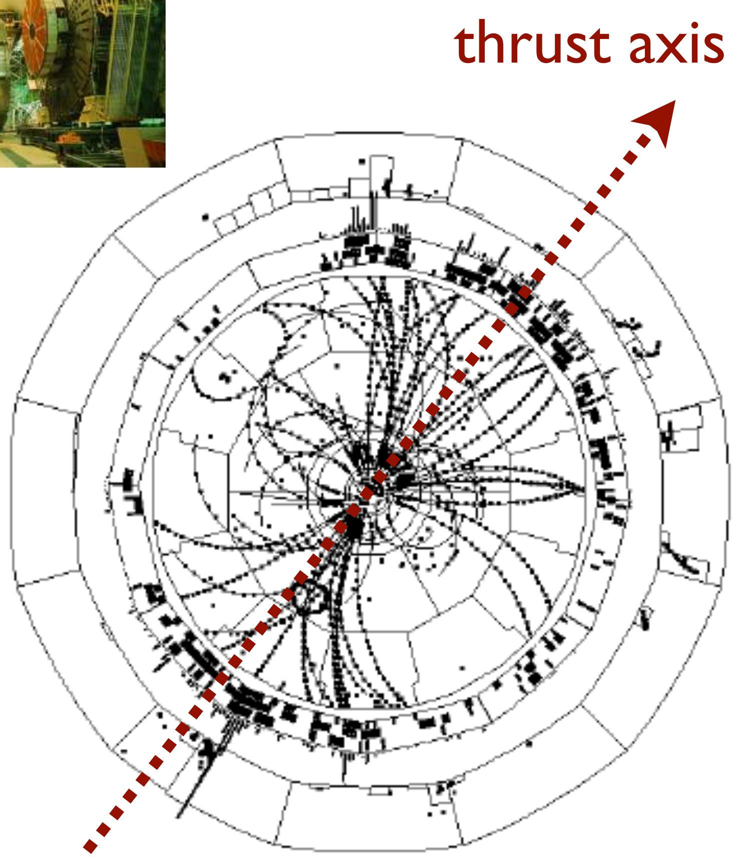
Multi-dimensional unbinned ***detector corrections*** ***via iterated application of*** *machine-learned reweighting*

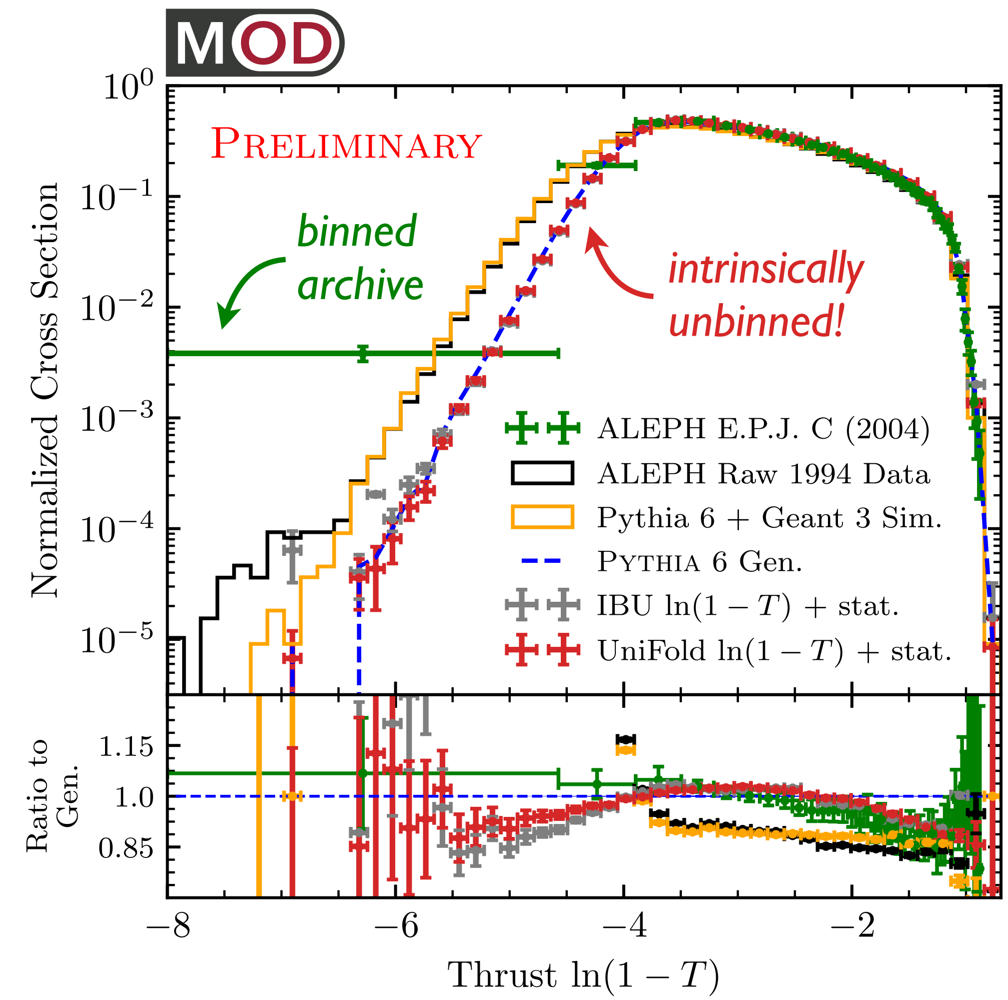


[Andreassen, Komiske, Metodiev, Nachman, JDT, PRL 2020; + Suresh, ICLR SimDL 2021;
Komiske, McCormack, Nachman, PRD 2021; see unfolding comparison in Petr Baron, APPB 2021]
[see alternative in Bellagente, Butter, Kasieczka, Plehn, Rousselot, Winterhalder, Ardizzone, Köthe, SciPost 2020]

E.g.: Detector Unfolding

Back to the Future with ALEPH Archival Data





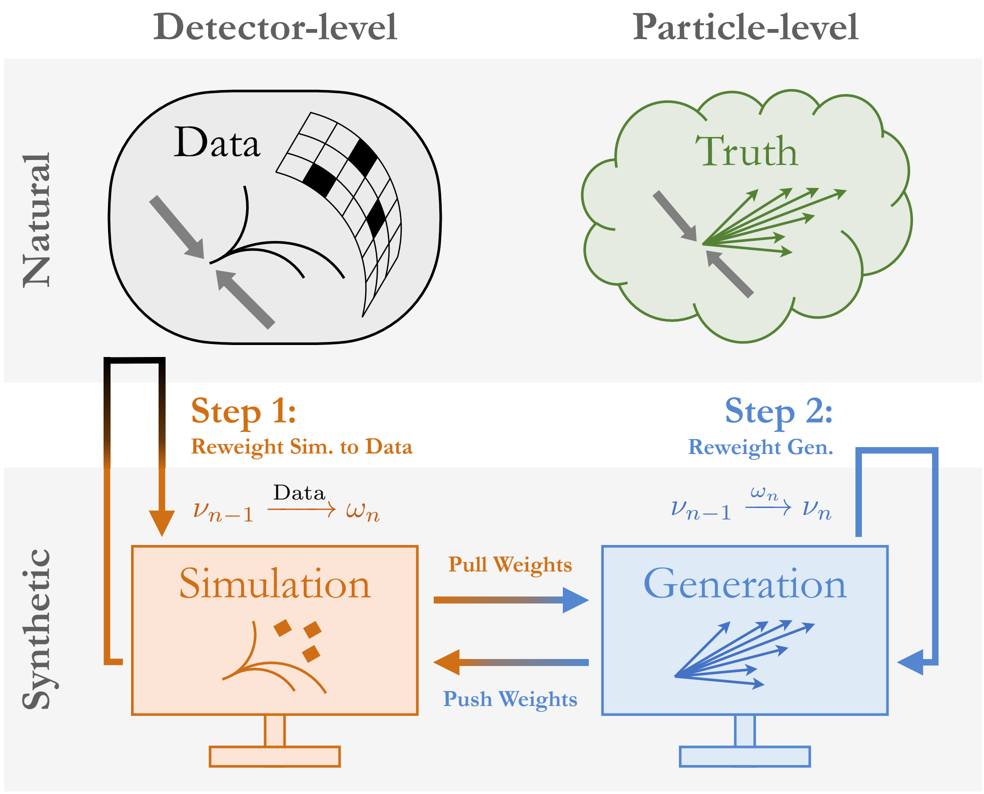
[talk by Badea, ICHEP 2020; cf. ALEPH, EPJC 2004]
[see also Badea, Baty, Chang, Innocenti, Maggi, McGinn, Peters, Sheng, JDT, Lee, PRL 2019; H1, DIS2021]

[Andreassen, Komiske, Metodiev, Nachman, JDT, PRL 2020; + Suresh, ICLR SimDL 2021; Komiske, McCormack, Nachman, PRD 2021; see unfolding comparison in Petr Baron, APPB 2021]
[see alternative in Bellagente, Butter, Kasieczka, Plehn, Rousselot, Winterhalder, Ardizzone, Köthe, SciPost 2020]

How do you estimate uncertainties on the learned likelihood itself?

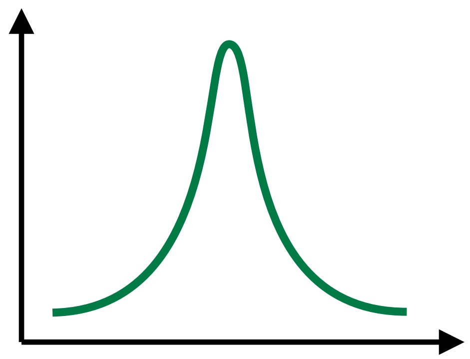
How do you even know if you remembered to train your model?!

This is a type of "algorithmic" uncertainty



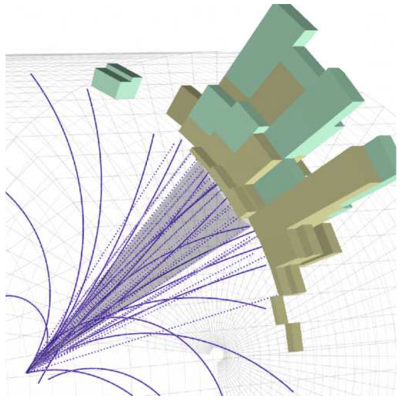
Progress in uncertainty quantification must keep pace with the development of exciting new ML/HEP strategies

Summary



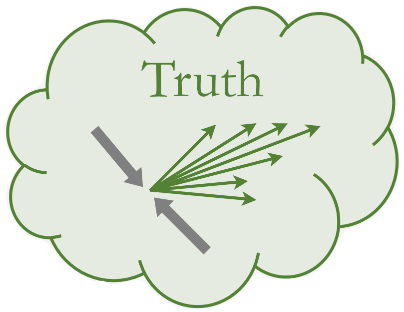
Learning and Uncertainties

Different types of uncertainty require different strategies for uncertainty quantification



Correlation for Calibration

With help from the Gaussian Ansatz and DVR loss, we can do frequentist calibration with improved resolution



The Next Frontier for UQ in HEP/ML

Machine learning will force us to confront the challenge of model selection with very large numbers of (hyper)parameters