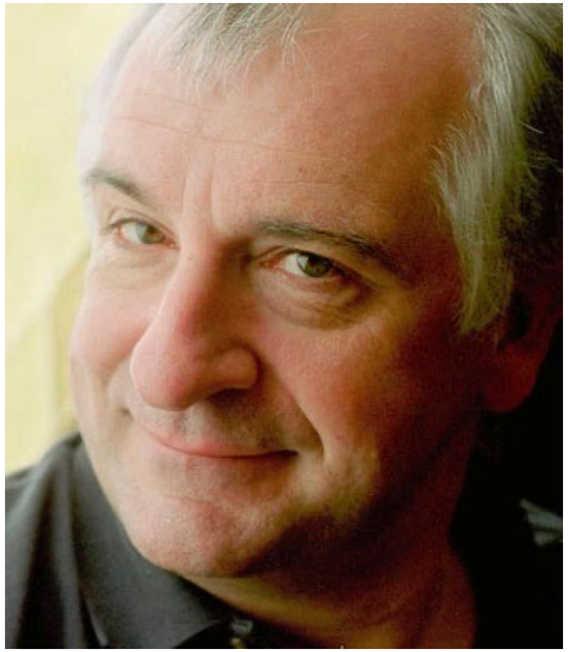
On Relating Uncertainties in Machine Learning and High Energy Physics

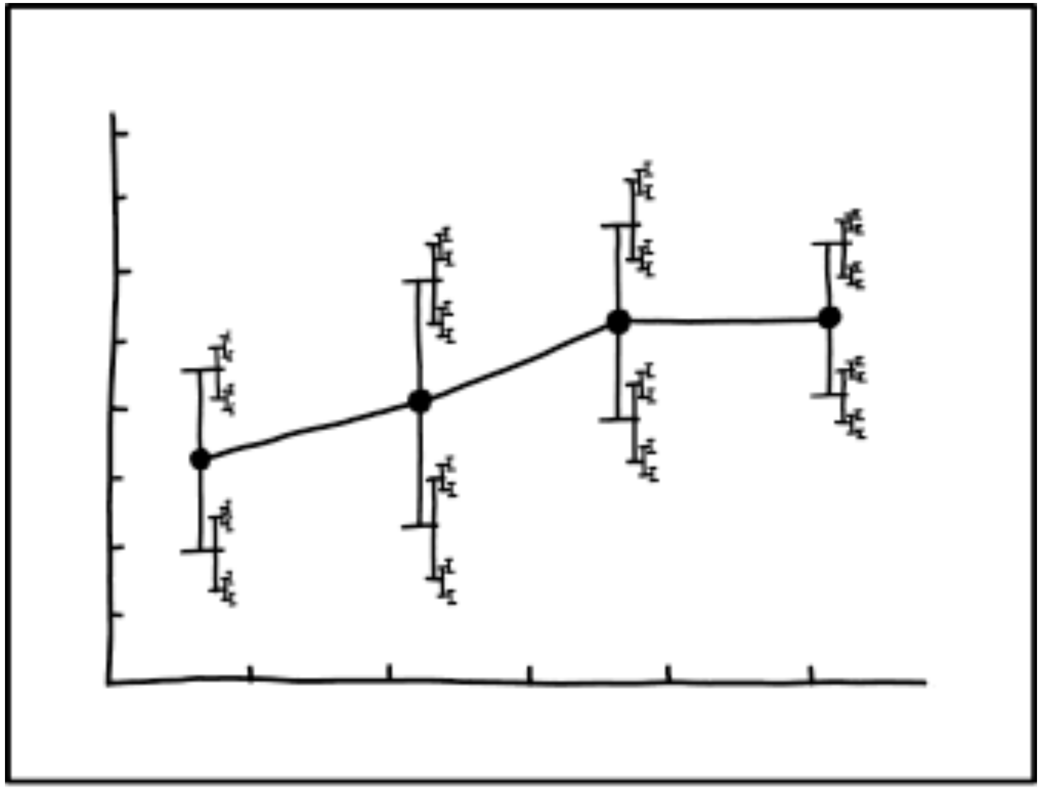
Michael Kagan
SLAC

PHYSTAT / CERN Data Science Seminar
November 16, 2022

We demand rigidly defined areas of doubt and uncertainty!



- Douglas Adams,
The Hitchhikers Guide to the Galaxy



I DON'T KNOW HOW TO PROPAGATE ERROR CORRECTLY, SO I JUST PUT ERROR BARS ON ALL MY ERROR BARS.

https://xkcd.wtf/2110/

Snowmass Machine Learning Report

CompF3: Machine Learning

Phiala Shanahan, Kazuhiro Terao, Daniel Whiteson (Editors)

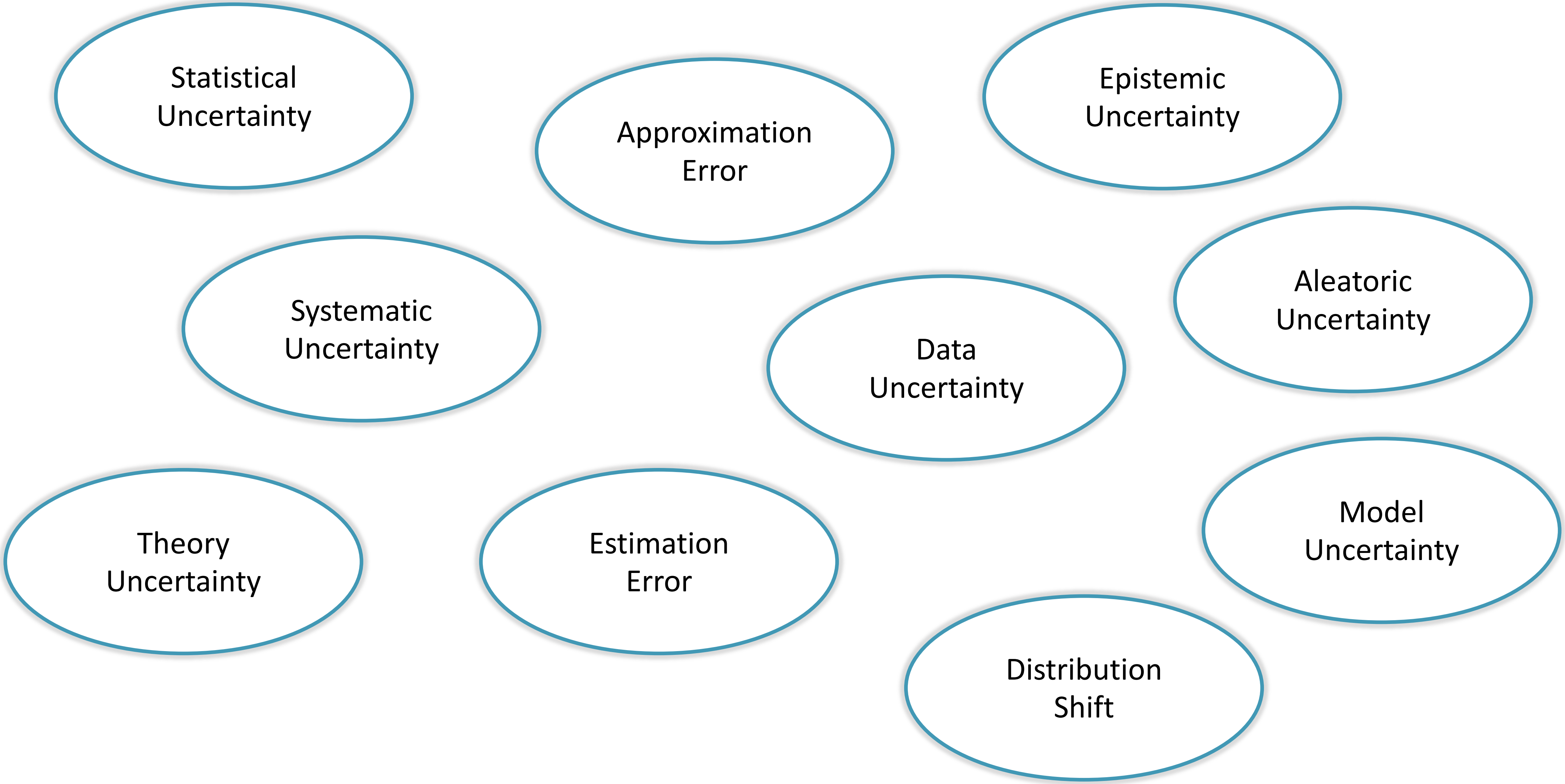
Including contributions from White Paper authors:

Gert Aarts[1,2], Andreas Adelmann[3], N. Akchurin[4], Andrei Alexandru[5,6], Oz Amram[7], Anders Andreassen[8], Artur Apresyan[9], Camille Avestruz[10], Rainer Bartoldus[11], Keith Bechtol[12], Kees Benkendorfer[13,14], Gabriele Benelli[59], Catrin Bernius[11], Alexander Bogatskiy[15], Blaz Bortolato[16], Denis Boyda[17,18], Gustaaf Brooijmans[19], Paolo Calafiura[13], Salvatore Calì[20,18], Florencia Canelli[21], Grigorios Chachamis[22], S.V. Chekanov[17], Deming Chen[23], Thomas Y. Chen[40], Aleksandra Ćiprijanović[9], Jack H. Collins[11], Andrew J. Connolly[24], Michael Coughlin[25], Biwei Dai[26], J. Damgov[4], Gage DeZoort[27], Daniel Diaz[28], Barry M. Dillon[16,29], Ioan-Mihail Dinu[7], Zhongtian Dong[30], Julien Donini[31], Javier Duarte[28], S. Dugad[32], Cora Dvorkin[33], D. A. Faroughy[21], Matthew Feickert[28], Yongbin Feng[9], Michael Fenton[58], Sam Foreman[17], Felipe F. De Freitas[34], Lena Funcke[20,18,35], P. G C[4], Abhijith Gandrakota[9], Sanmay Ganguly[36], Lehman H. Garrison[15], Spencer Gessner[11], Aishik Ghosh[58], Julia Gonsk[19], Matthew Graham[48], Lindsey Gray[9], S. Grönroos[37], Daniel C. Hackett[20,18], Philip Harris[20], Scott Hauck[24], Christian Herwig[9], Burt Holzman[9], Walter Hopkins[17], Shih-Chieh Hsu[24], Jin Huang[38], Yi Huang[38], Xiao-Yong Jin[17], Michael Kagan[11], Alan Kah[19], Jernej F. Kamenik[16,39], Raghav Kansal[28], Georgia Karagiorgi[40], Gregor Kasieczka[41], Erik Katsavounidis[20], Elham E Khoda[24], Charanjit K. Khosa[42,43], Thomas Kipf[44], Patrick Komiske[20], Matthias Komm[37], Risi Kondor[45], Evangelos Kourlitis[17], Claudius Krause[46], K. Lamichhane[4], Luc Le Pottier[13,10], Meifeng Lin[38], Yin Lin[20,18], Mia Liu[47], Nan Lu[48], Biagio Lucini[49,1], J. Martinez[4], Pablo Martín-Ramiro[13,50], Andrej Matevc[16,39], William Patrick McCormack[20], Eric Metodiev[20], Vinicius Mikuni[21], David W. Miller[45], Siddharth Mishra-Sharma[33,18,6], Samadrita Mukherjee[32], Daniel Murnane[13], Benjamin Nachman[13,51], Gautham Narayan[23], Mark Neubauer[23], Jennifer Ngadiuba[9], Scarlet Norberg[60], Brian Nord[9,4], Inês Ochoa[52], Jan T. Offermann[45], Sang Eon Park[20], Kevin Pedro[9], Cristían Peña[9], Alexx Perloff[61], Mariel Pettee[13], Maurizio Pierini[37], T. Quast[37], Dylan Rankin[20], Yihui Ren[38], Marcel Rieger[37], Jean-Roch Vlimant[48], Avik Roy[23], Veronica Sanz[42,53], Nilai Sarda[20], Claire Savard[61], Alexander Scheinker[54], Uroš Seljak[13,51,26], Brian Sheldon[28], David Shih[46], Chase Shimmin[55], Aleks Smolkovic[16], George Stein[13,26], Cristina Mantilla Suarez[9], Manuel Szewc[56], Savannah Thais[27], Jesse Thaler[20], Dmitrii Torbunov[38], Nhan Tran[9], Steven Tsan[28], Silviu-Marian Udrescu[20], S. Undleeb[4], Louis Vaslin[31], Francisco Villaescusa-Navarro[15,27], V. Ashley Villar[57], Brett Viren[38], Jean-Roch Vlimant[48], A. Whitbeck[4], Daniel Williams[19], Daniel Winklehner[20], Si Xie[48], Tingjun Yang[9], Haiwang Yu[38], and Mikaeel Yunus[20]

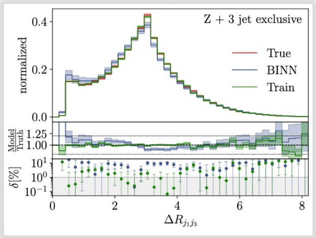
Uncertainty Quantification, Validation and Interpretability It is vital that physicists can validate the decisions of ML models and quantify their uncertainty, a goal made easier if the inner workings of the models are conceptually accessible to physicists. HEP is not alone in this concern, and can benefit from work in the wider community. The HEP community should support continued research into interpretable AI and uncertainty quantification (UQ), including making public benchmark data sets for rigorous testing and comparison of approaches to physically interpretable AI-UQ for physics, and supporting challenges and competitions to create and compare methods of uncertainty quantification, including bias mitigation.

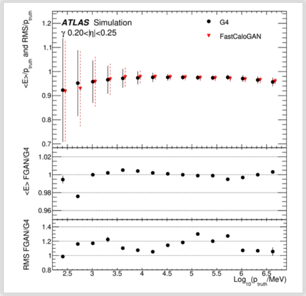
2209.07559

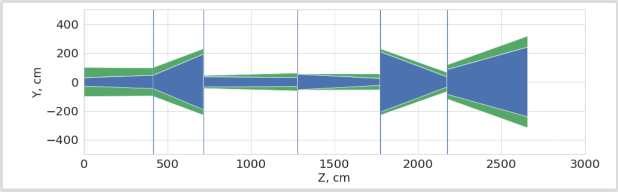
Many Terminologies Around Uncertainty



Machine Learning in the Data Analysis Pipeline







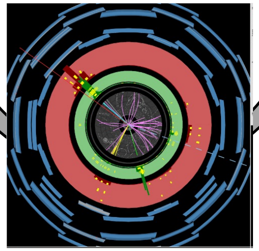
θ – Physics Parameters

Matrix Elements, PDF’s

Fragmentation / Hadronization

Detector Interactions

Data Generation



$\hat{\theta}$ – Parameter Estimates

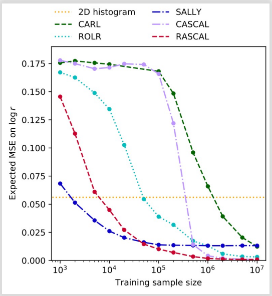
Hypothesis Testing

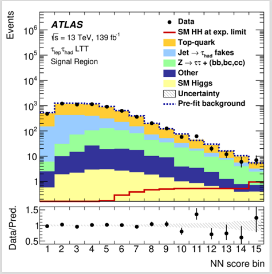
Event Selection / Reconstruction

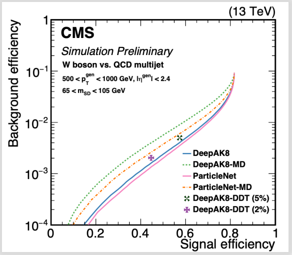
Particle Reconstruction

Trigger

Data Analysis







Primary Questions of Concern

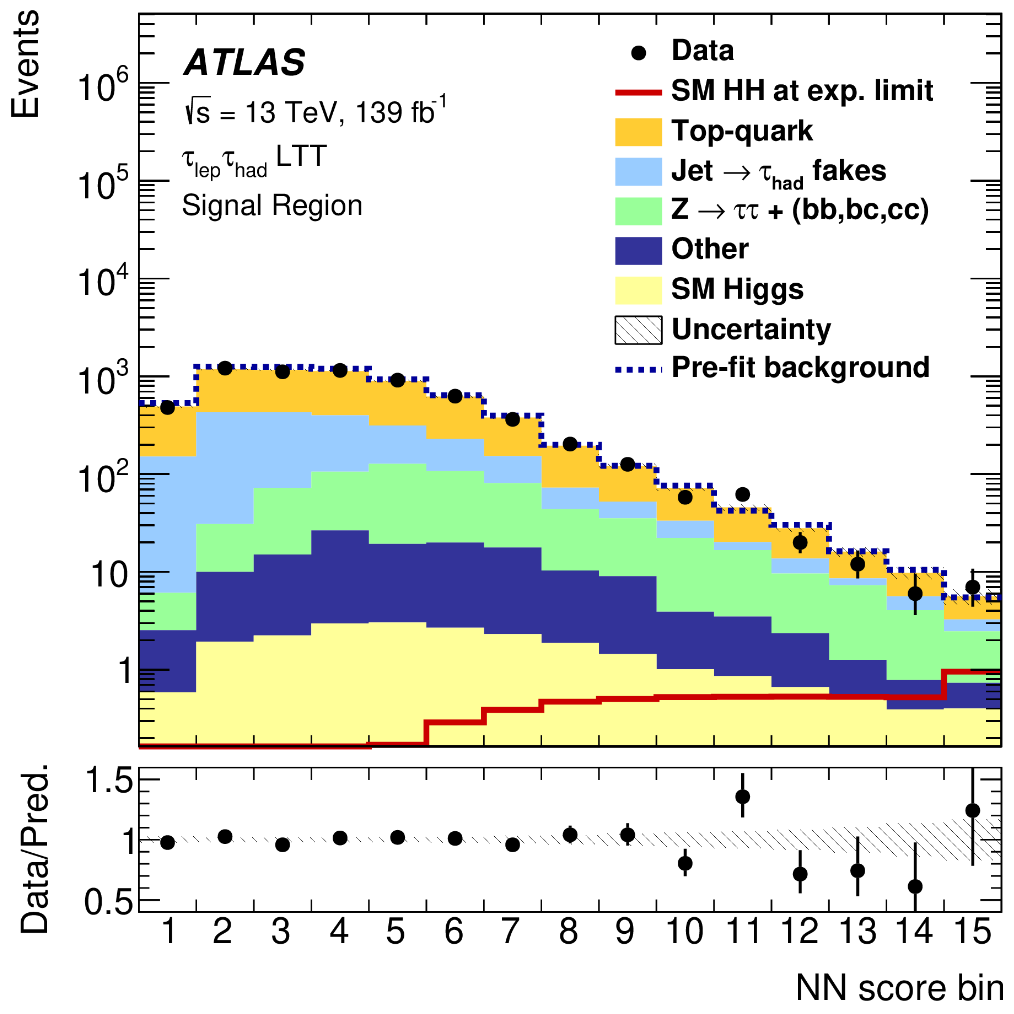
How to deal with ***Systematic Uncertainties*** when using Machine Learning Models?

Is there uncertainty from using the ML Model?

ML "Model Uncertainty":

What if the ML model did not "perfectly" fit the data?

When does it matter?



2209.10910

Supervised Learning Setup

Training Data:

- $\mathcal{D} = \{x_i, y_i\}$ = features and target
- $x, y \sim p(x, y)$

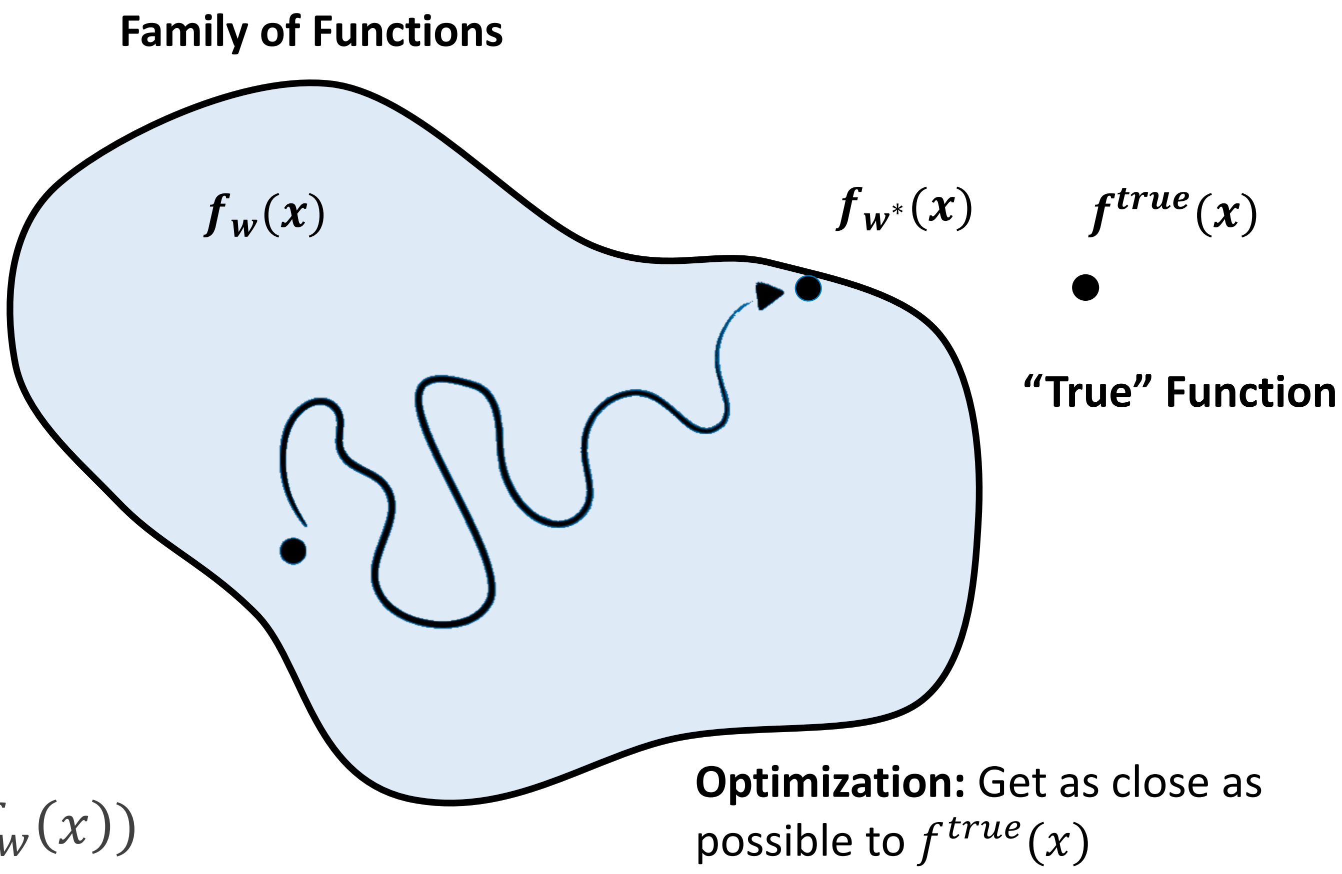
Goal:

- Learn $f_w(x) = \hat{y}$
- w = model weights

Learning:

- Optimize Loss

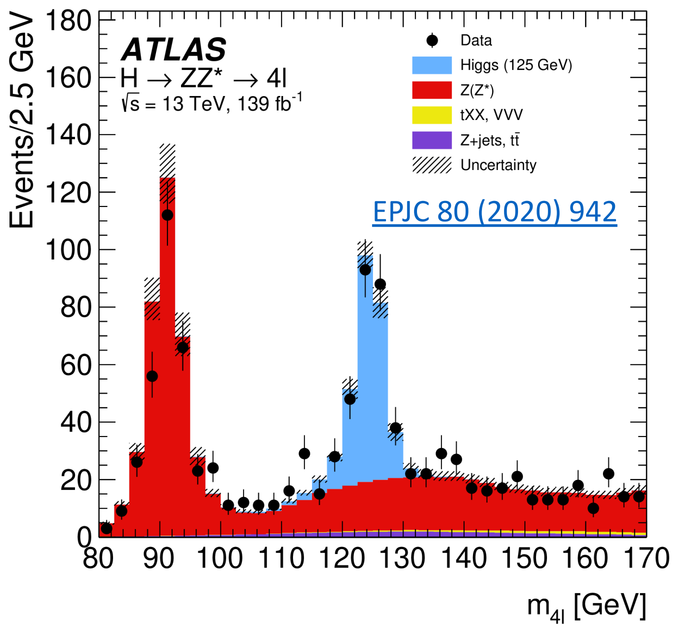
$$\mathrm{w}^* = \arg\min_w L = \arg\min_w \frac{1}{N} \sum_i \mathcal{L}(y, f_w(x))$$



Optimal vs. Correct

Reconstruction, data selection, event classification enable us to define powerful summary statistics

$$T_{w^*,\phi}(x)\colon\ \mathbb{R}^{10^8} \to \mathbb{R}$$



Estimate likelihood for frequentist parameter inference:

$$p\left(T_{w^*,\phi}(x) \middle|\ \lambda(\theta)\ \right)$$

- θ = physics parameters of interest
- w^* =Learned params, ϕ = Reco / analysis params
- $\lambda(\cdot)$ = parameters of probability density
  e.g. mean of Poisson / Gaussian density

Related / similar discussions:
Cranmer talk, Nachman, 1909.03081

† Ignoring Systematics for the moment

Optimal vs. Correct

Reconstruction, data selection, event classification enable us to define powerful summary statistics

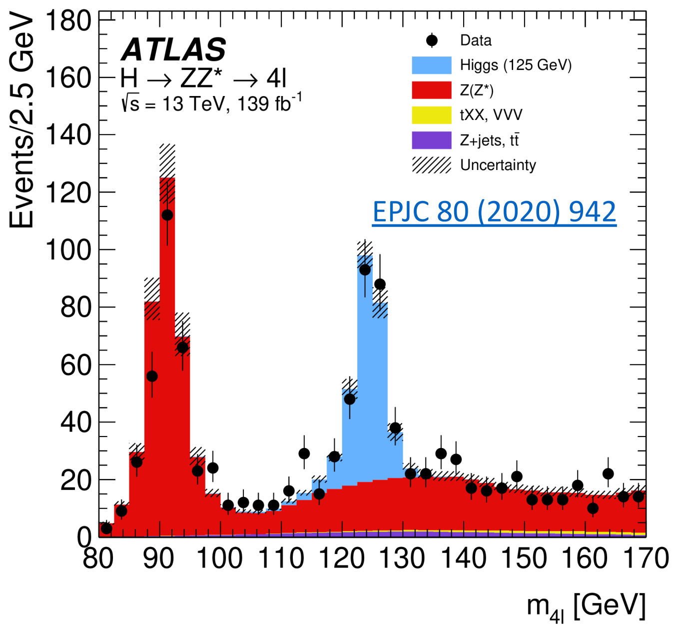
$$T_{w^*,\phi}(x)\colon\ \mathbb{R}^{10^8} \to \mathbb{R}$$

Estimate likelihood for frequentist parameter inference:

$$p\left(T_{w^*,\phi}(x) \middle|\, \lambda(\theta)\,\right)$$

Changing summary statistic $T(x)$ affects optimality of result, but not correctness

- Reconstruction, event classification, …
- ***Not a question of ML model uncertainty***



EPJC 80 (2020) 942

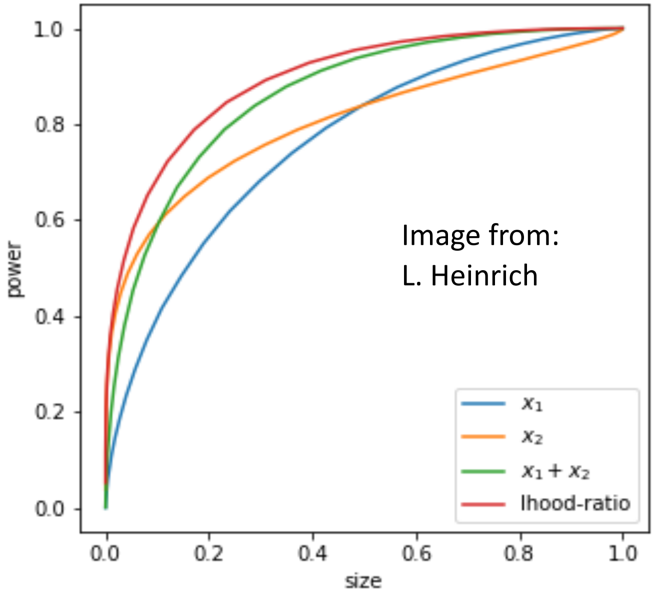


Image from:
L. Heinrich

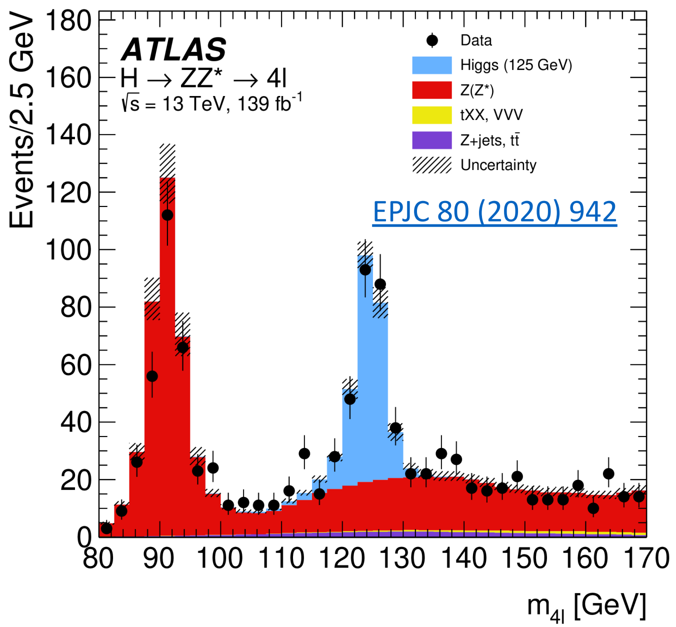
Related / similar discussions:
Cranmer talk. Nachman, 1909.03081

† Ignoring Systematics for the moment

Optimal vs. Correct

Reconstruction, data selection, event classification enable us to define powerful summary statistics

$$T_{w^*,\phi}(x)\colon\ \mathbb{R}^{10^8} \to \mathbb{R}$$



EPJC 80 (2020) 942

Estimate likelihood for frequentist parameter inference:

$$p\left(T_{w^*,\phi}(x)\middle|\ \lambda(\theta)\ \right)$$

ML models that affect $\lambda(\cdot)$

- Background estimation, simulations, …
- Affects compatibility of statistical model with data
- ***Quality of ML model could lead to uncertainty,***
  Or requires additional systematic uncertainties

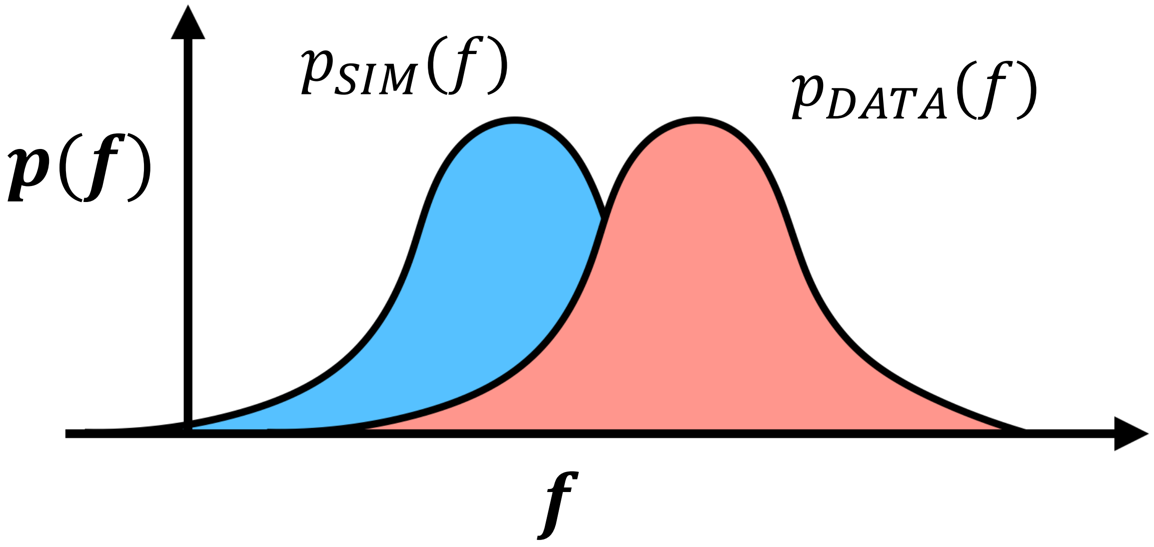
Related / similar discussions:
Cranmer talk, Nachman, 1909.03081

† Ignoring Systematics for the moment

The Effect of Systematic Uncertainties

Systematic Uncertainties

- Simulation used for training $f_w(x)$
- Simulation not a perfect model of data
- $p_{SIM}(x,y) \neq p_{DATA}(x,y)$



Problem:

- Evaluating $f_w(x)$ will result in different distributions in simulation and data

Must consider how to handle systematic uncertainties for all ML models

Uncertainty, ML, and HEP – Menu for Today

How do Machine Learners think about uncertainty?

What kinds of uncertainty is relevant?

How do we estimate these uncertainties, when we need to?

How can we incorporate systematic uncertainties in HEP ML models?

This talk: An incomplete look at an ongoing research area

- Uncertainties workshop at Learning to Discover → this talk started there
- Great new ML review in PDG: [Cranmer, Seljak, Terao, 2021]
- Snowmass paper on uncertainty for ML in HEP: [2208:03284]
- Book Chapter: [Dorigo, de Castro Manzano]

Uncertainties in Machine Learning

Types of Uncertainties

Aleatoric Uncertainty:

Inherent variations in data, e.g. due to randomness of the process

Epistemic Uncertainty:

Due to lack of knowledge, lack of data, incomplete information

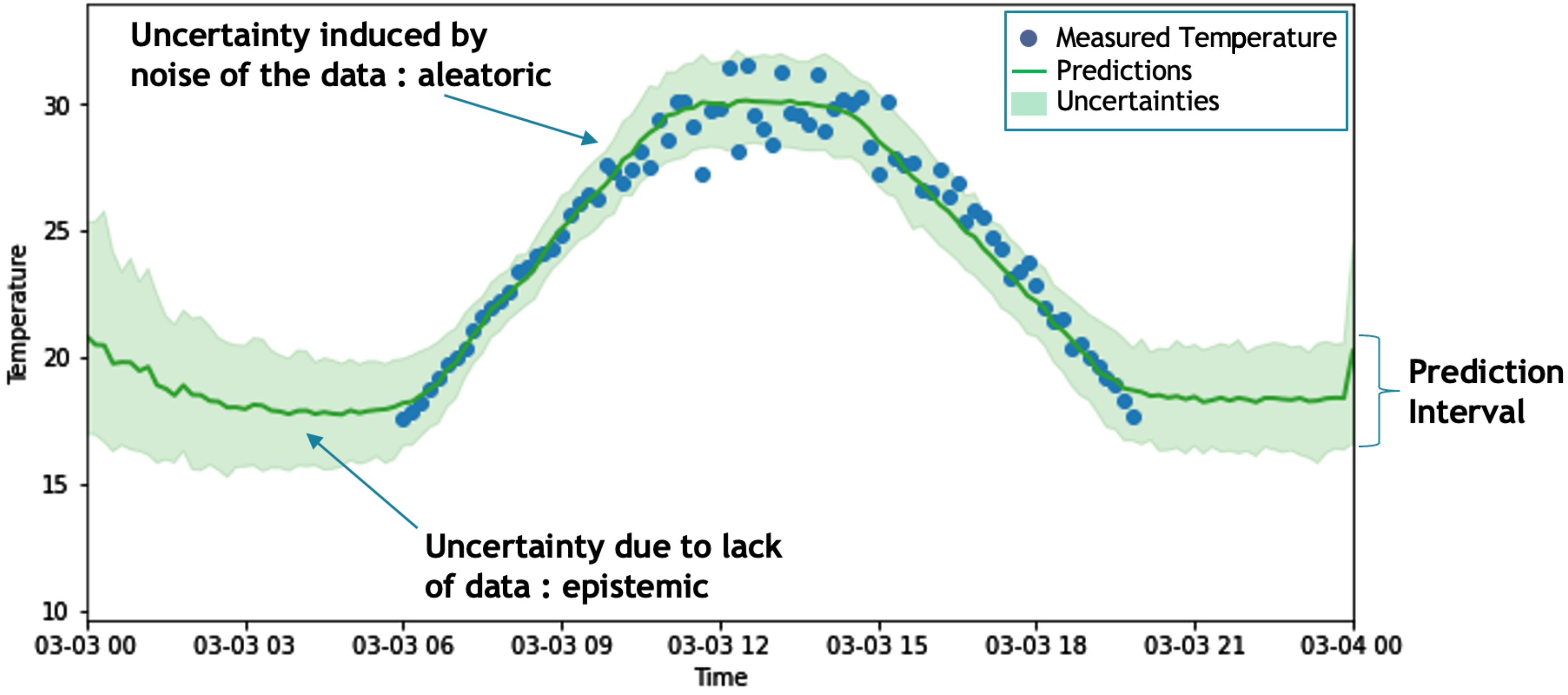
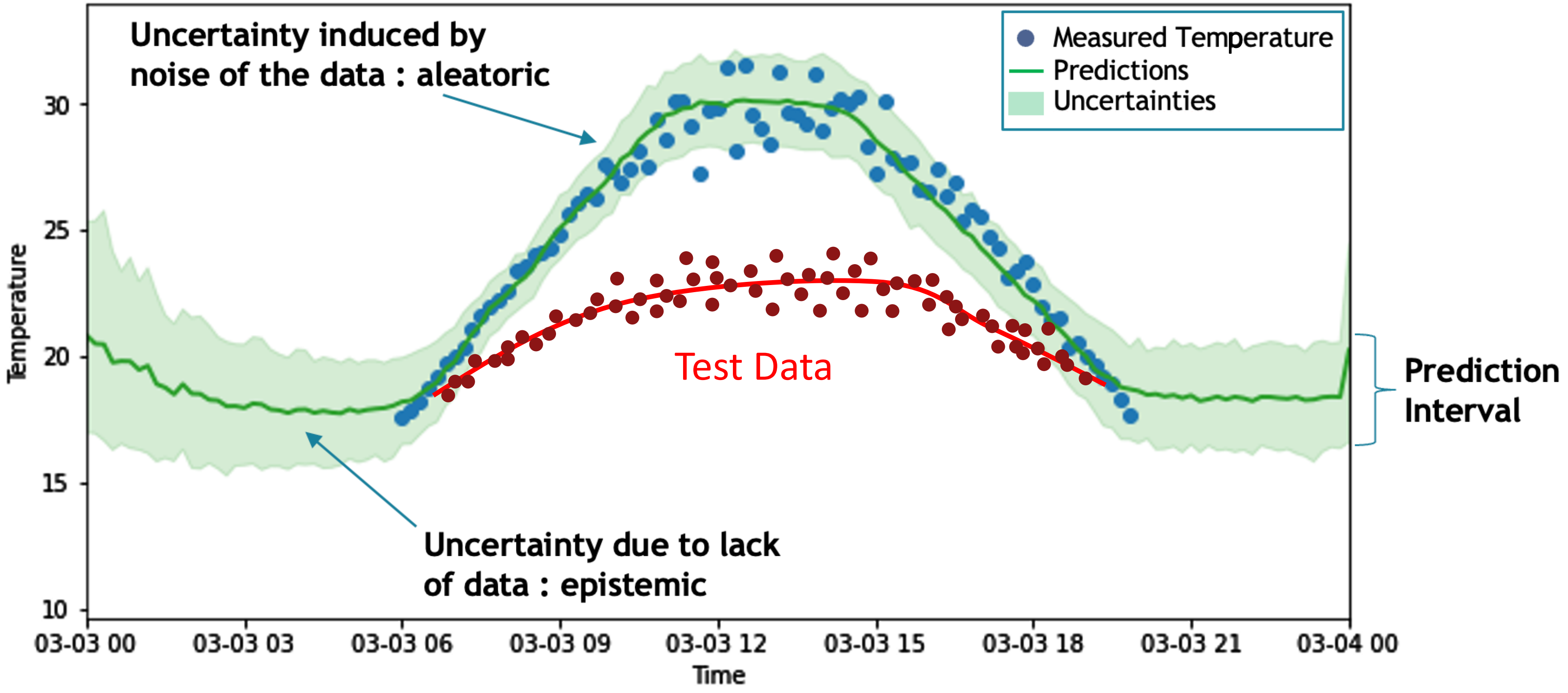


Image Credit: N. Brunel

Types of Uncertainties

Aleatoric Uncertainty:
Inherent variations in data, e.g. due to randomness of the process

Epistemic Uncertainty:
Due to lack of knowledge, lack of data, incomplete information



Domain Shift:
Test data is different from training data

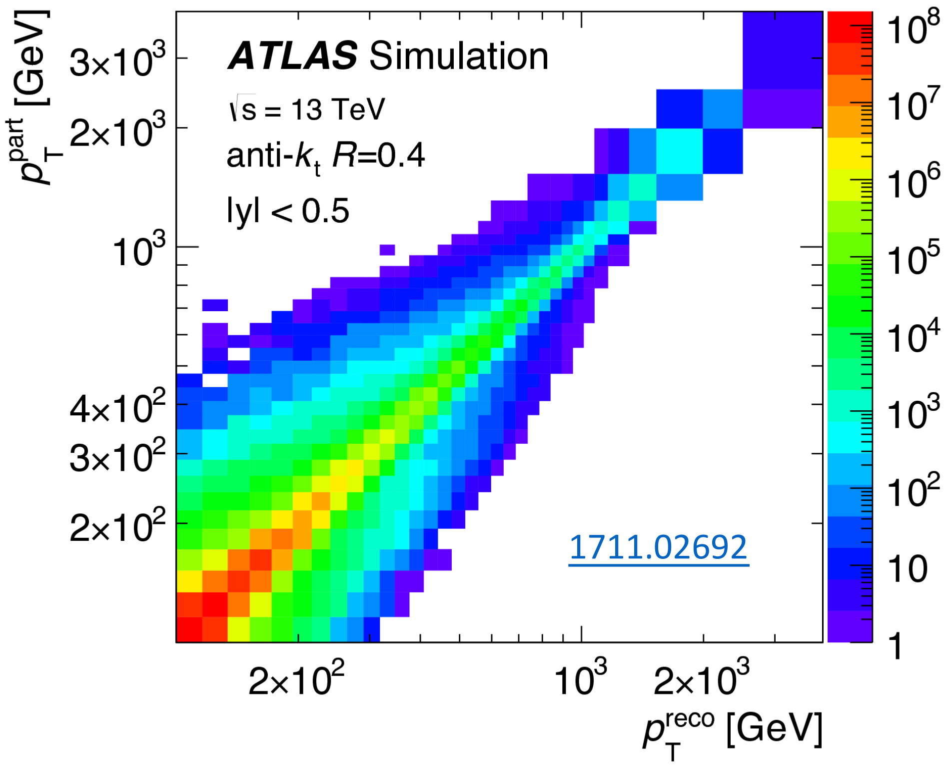
Image Credit: N. Brunel

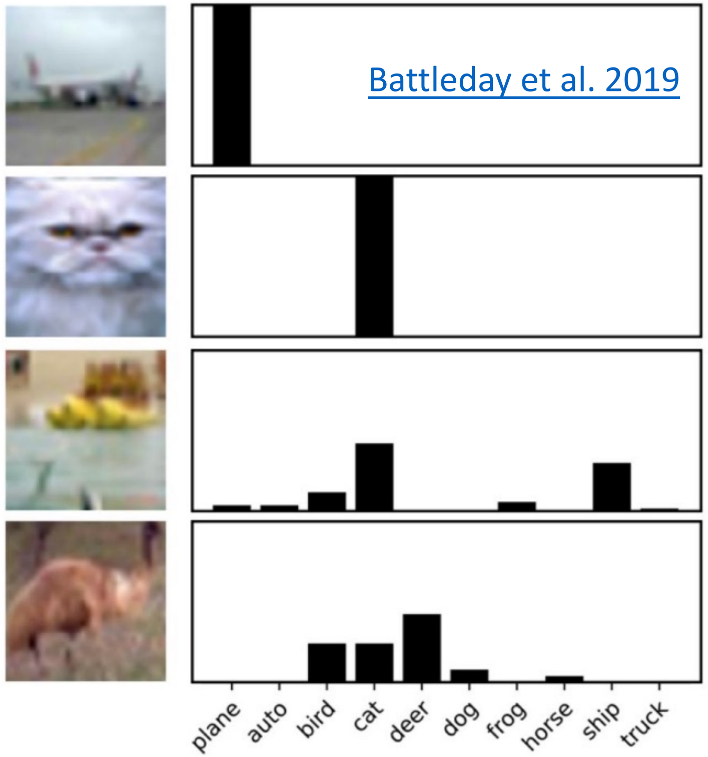
Aleatoric Uncertainty

Often called “Statistical Uncertainty”

Variability in outcome of experiment due to inherently random effects

Often considered “irreducible"





Epistemic Uncertainty

Lack of knowledge about the best model

Main origins in ML

- Estimation error: Training data just a sample of possible observations
- Approximation error: no model (in model class) can capture unknown true model

Often considered "reducible" with more data or more complex model

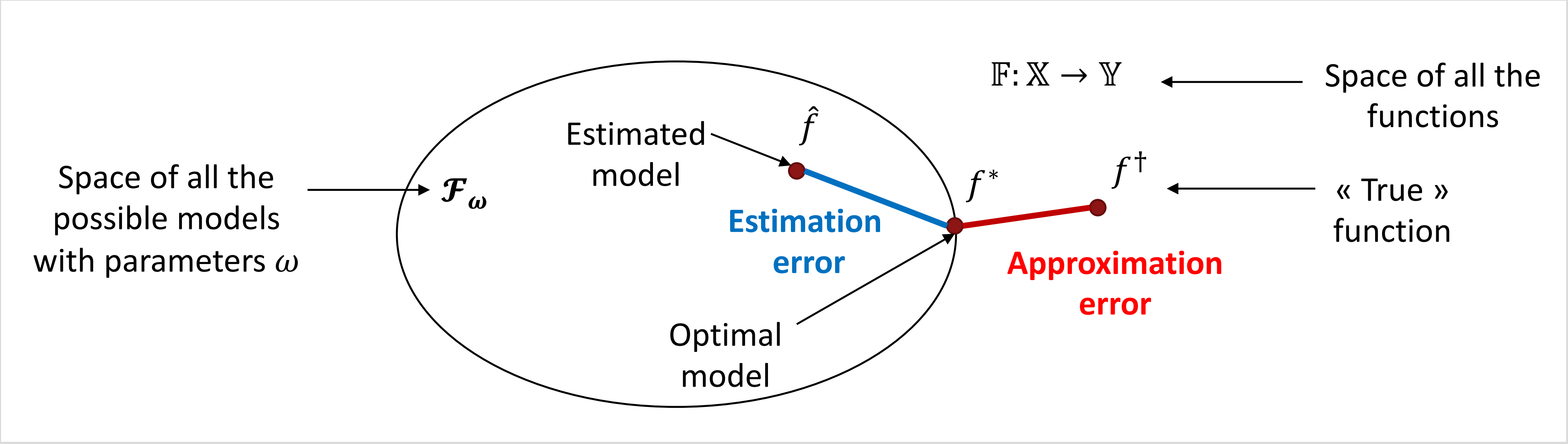


Image credit: G. Daniel

Domain / Distribution / Dataset Shift

$$p_{TEST}(x,y) \neq p_{TRAIN}(x,y)$$

Examples:

- Covariate Shift: $p(y|x)$ fixed but $p_{TEST}(x) \neq p_{TRAIN}(x)$
- Label Shift: $p(x|y)$ fixed but $p_{TEST}(y) \neq p_{TRAIN}(y)$
- Concept Shift: $p(y)$ fixed but $p_{TEST}(x|y) \neq p_{TRAIN}(x|y)$

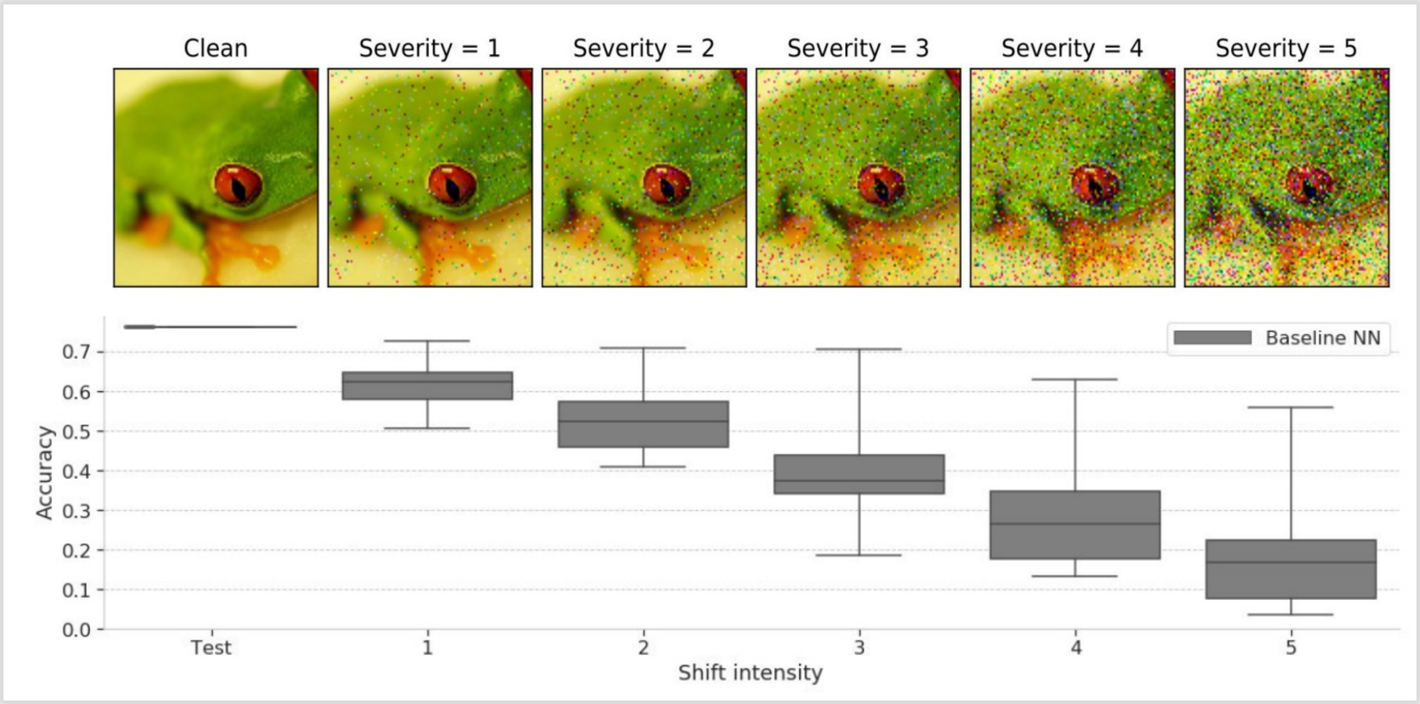


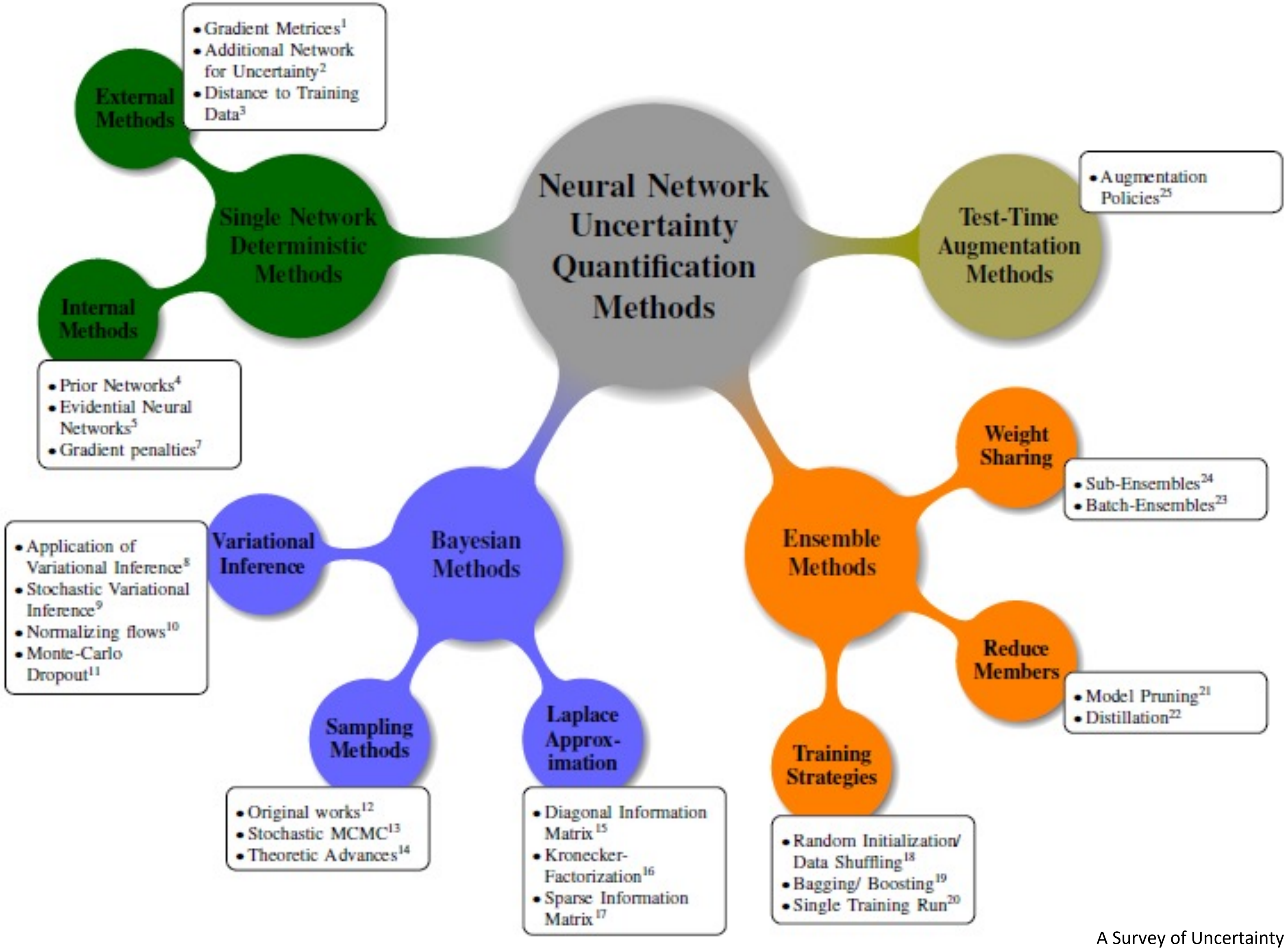
Image: [1903.12261](1903.12261)

Imperfect Correspondence: My View*

Machine Learning	HEP
Aleatoric uncertainty • “Statistical” / “Data” Uncertainty • Uncertainty Inherent to data • Not reduced w/ more data	Detector Noise Resolutions
	Stat. errors in HEP ?
Epistemic uncertainty • “Model” Uncertainty • Uncertainty from Imperfect knowledge • Reduces with more data	Systematic errors induced by ML model training on finite stats.
Domain Shift • Imperfect model of data generation process	Systematic Uncertainties from data / simulation differences

*Even within the ML community, these terms can be ambiguous

Uncertainty Estimation Approaches in Deep Learning



A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Aleatoric Uncertainty

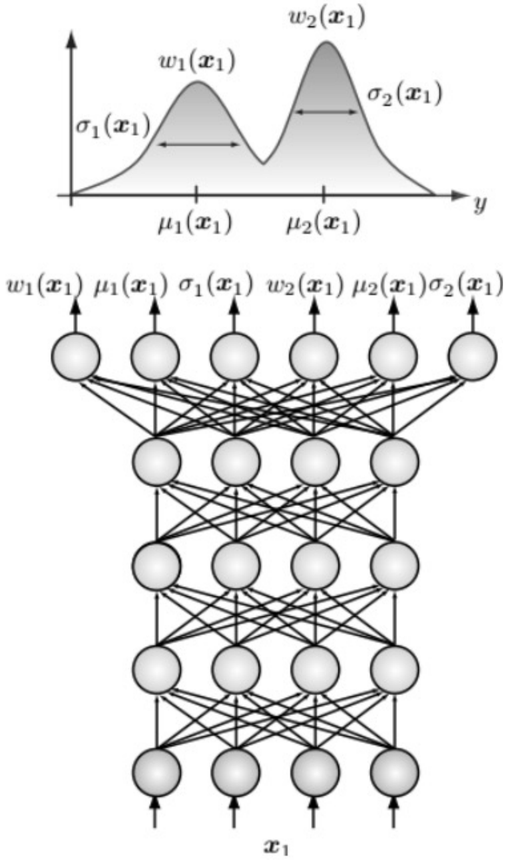
Randomness of data → Typically described by probability distributions

Density Networks

Define density $p_\phi(y|x)$ with parameters ϕ

Train neural network to predict per-example parameters

$$f(x) \to \phi(x)$$



Mixture density network

Aleatoric Uncertainty

Randomness of data → Typically described by probability distributions

Generative Models:

Aim to approximate a density, $p(x)$

Train NN to transform noise $z \sim p(z)$ into data:

$$\hat{x} = f_w(z), \qquad p(\hat{x}) \approx p_{data}(x)$$

Implicit models:
can only generate sample synthetic data, e.g. GANS

Explicit models:
can also evaluate density, e.g. Normalizing Flows

StyleGAN v2

(Karras et al, 2019)

Aleatoric Uncertainty in HEP with Generative Models

Simulators slow / hard to sample from → approximate with Generative Model

Generative Adversarial Networks:

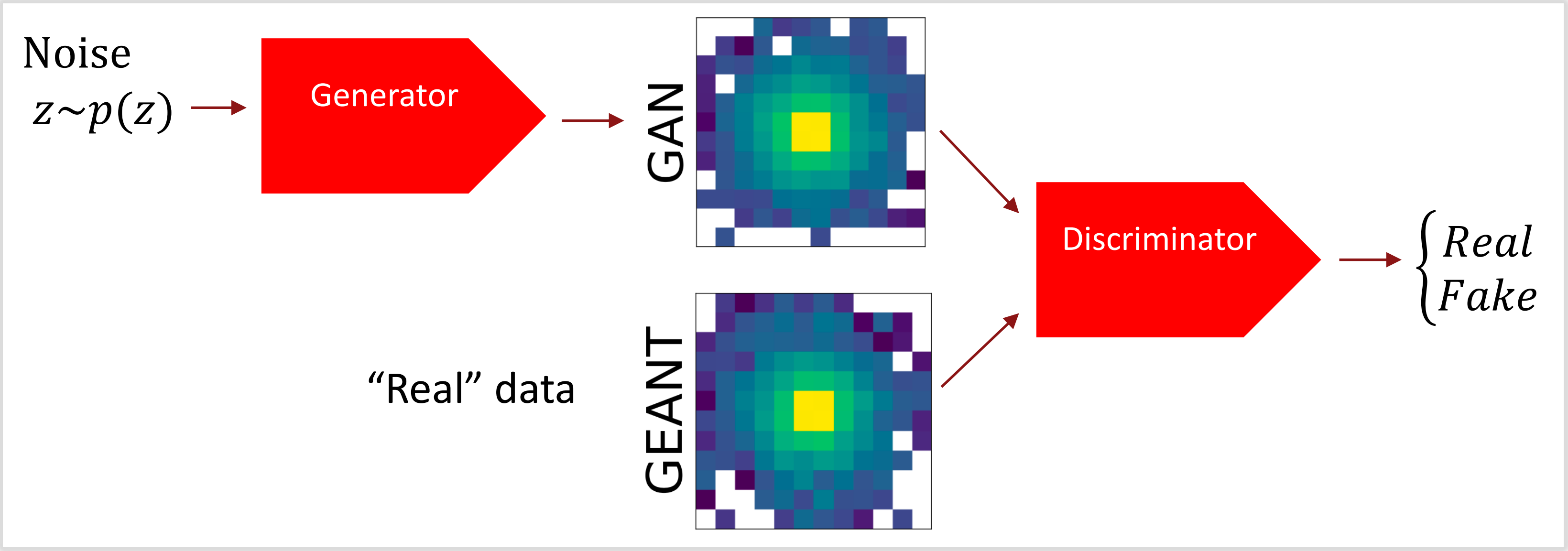
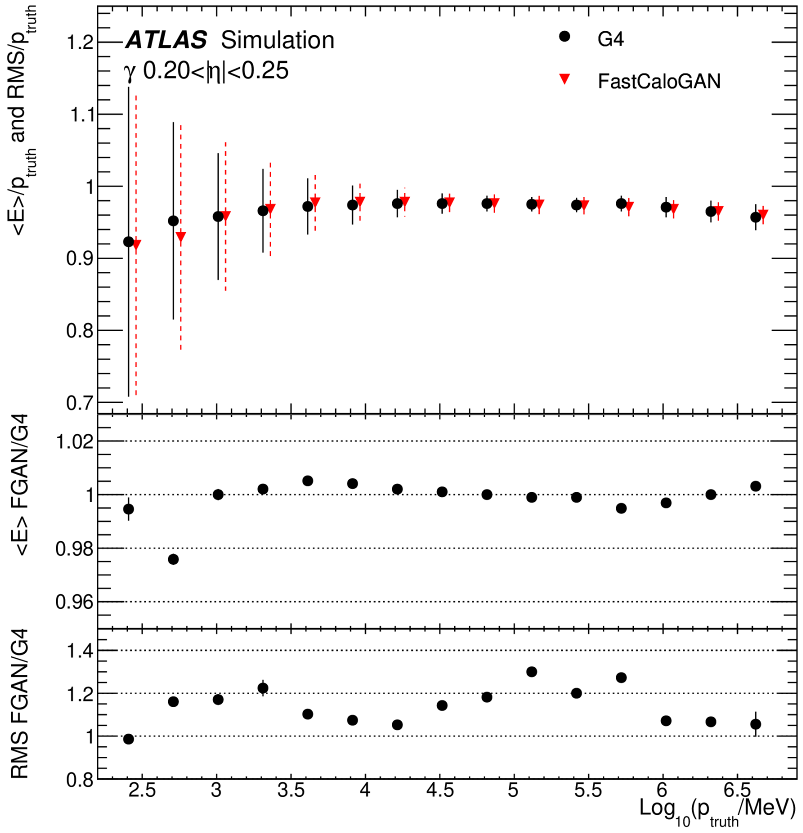


Image Credit: 1712.10321

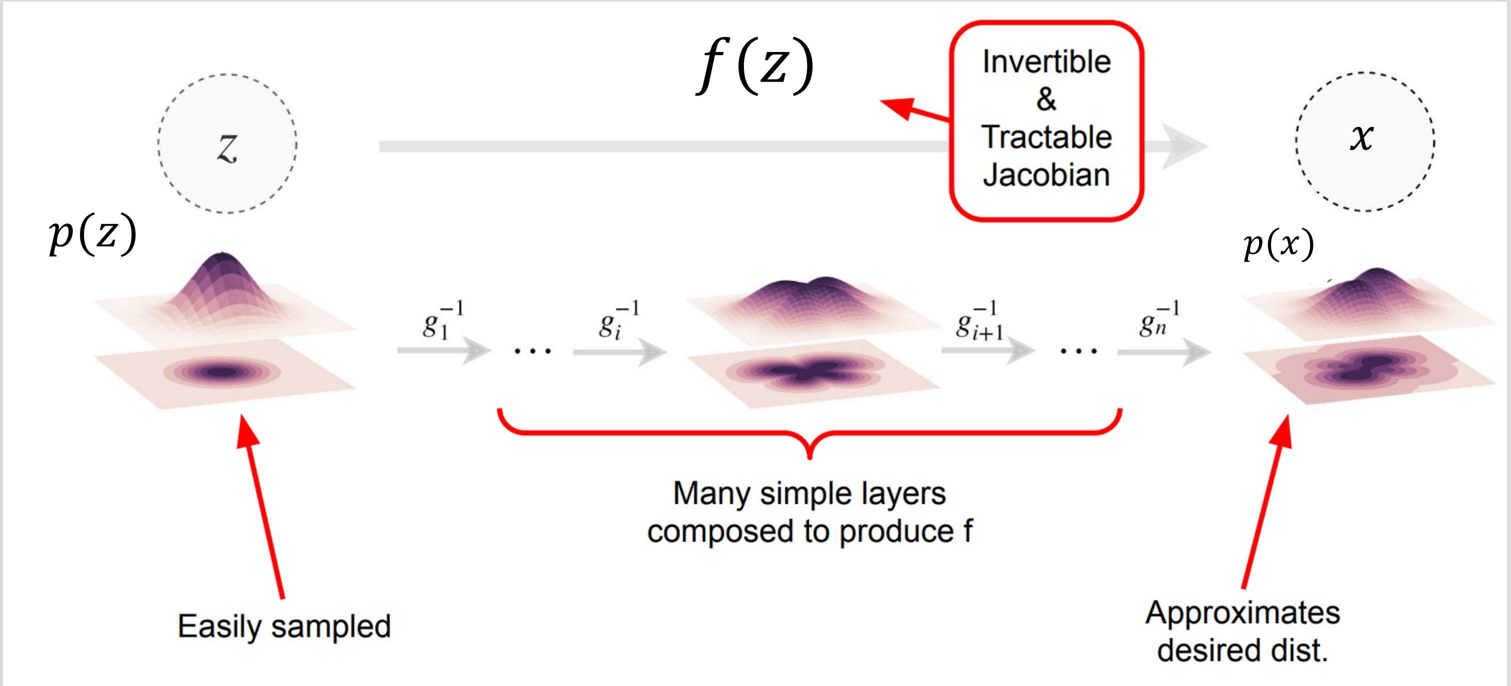


arXiv:2109.02551

Aleatoric Uncertainty in HEP with Generative Models

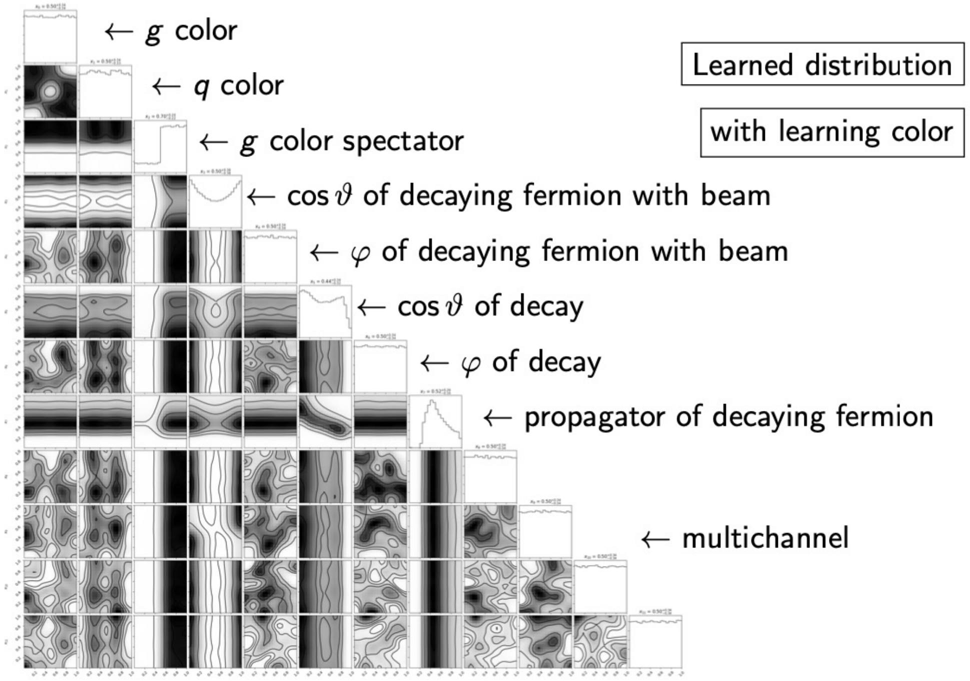
Simulators slow / hard to sample from → approximate with Generative Model

Normalizing Flows



$$p_x(\boldsymbol{x}) = p_z(\boldsymbol{z}) \left| \det \left(\frac{\partial f(\boldsymbol{z})}{d\boldsymbol{z}} \right)^{-1} \right|$$

Example: Learning $e^+e^- \to 3j$ Matrix Elements



Gao, et. al, 2001.05486, Gao, et. al, 2001.10028

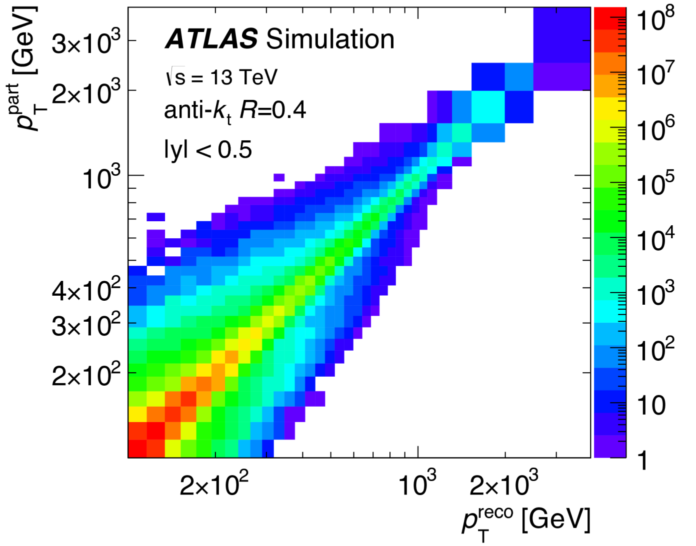
Aleatoric Uncertainty in HEP: Neural Unfolding

$$p_{reco}(y) = \int p(y|x)p_{true}(x)$$

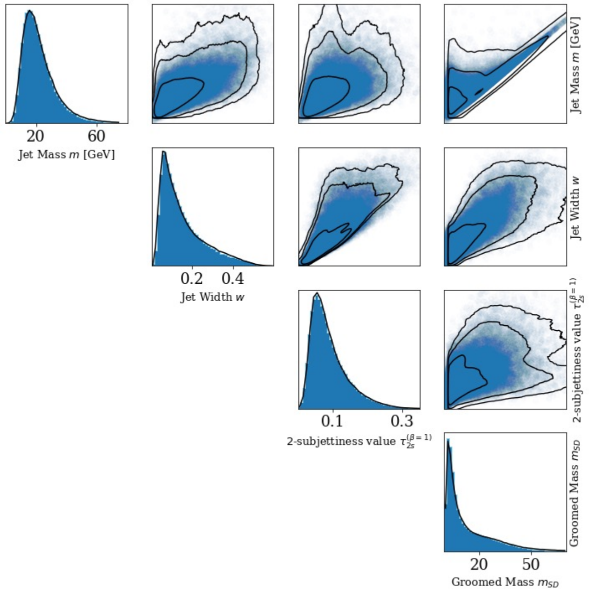
Continuous Form

$$\boldsymbol{y} = \boldsymbol{Rx}$$

Discrete Form



Response Matrix in unfolding → Aleatoric Uncertainty



Several recent methods using ML to model the response and enable high-dimensional continuous unfolding

- E.g. [2011.05836](), [2006.06685](), [1911.09107]()

What if ML learns the wrong generative model or response?

$\rightarrow$ Understanding ML Model / Epistemic Uncertainties

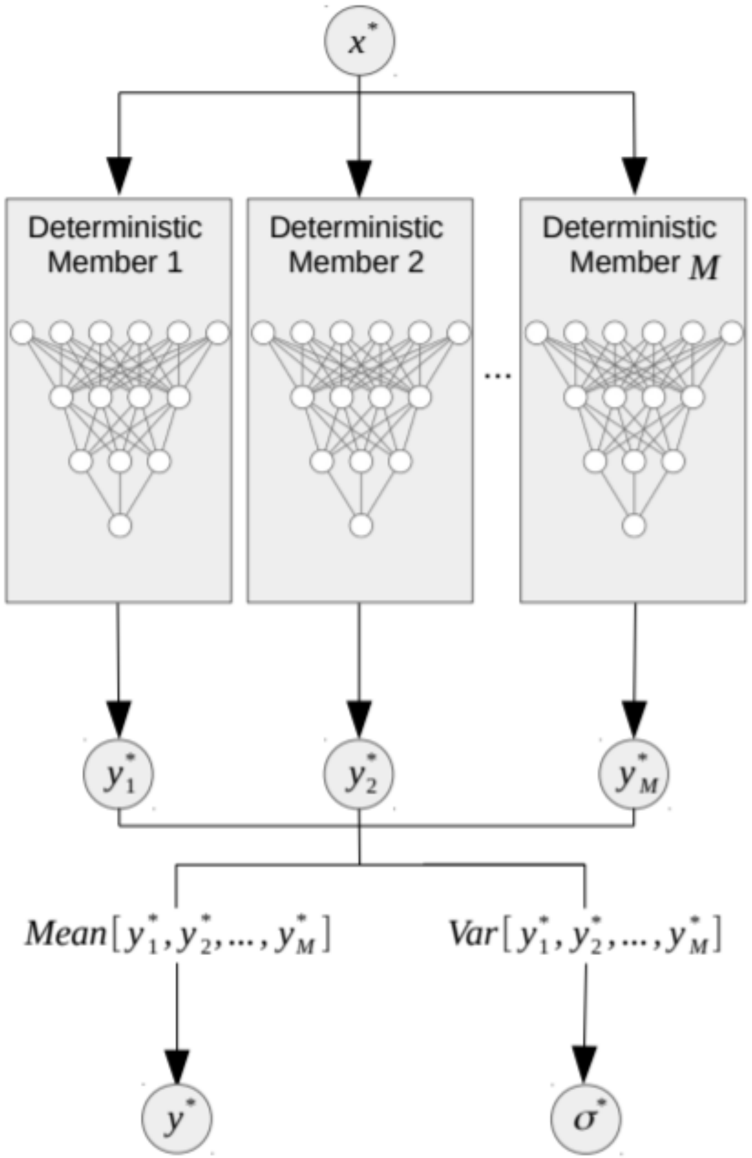
Epistemic Uncertainty with Deep Ensembles

Ensembling:

- Retrain network from multiple initializations

Can be coupled with Bootstrapping

- Randomly sample data, with replacement, to define each model's training set

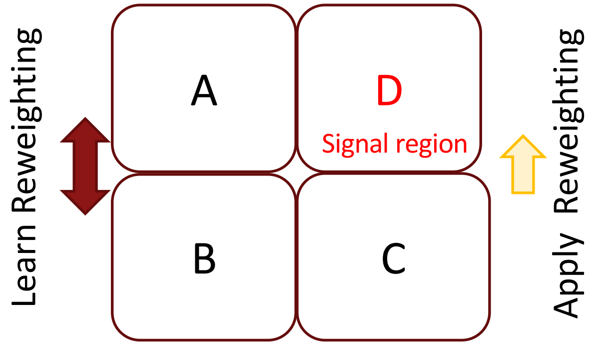


Lakshminarayanan, Pritzel, Blundell, [1612.01474](), [Nixon, Lakshminarayanan, Tran, 2020]()

Model Uncertainty in ML-based Background Estimation

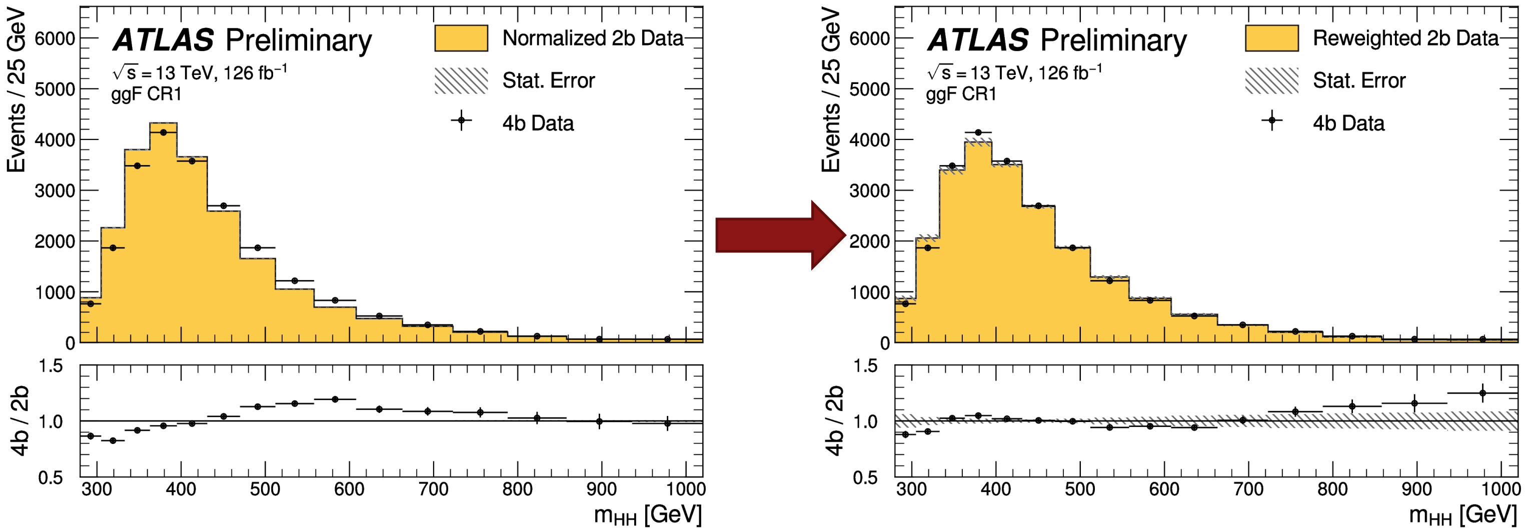
High-Dimensional "ABCD" method with NN's

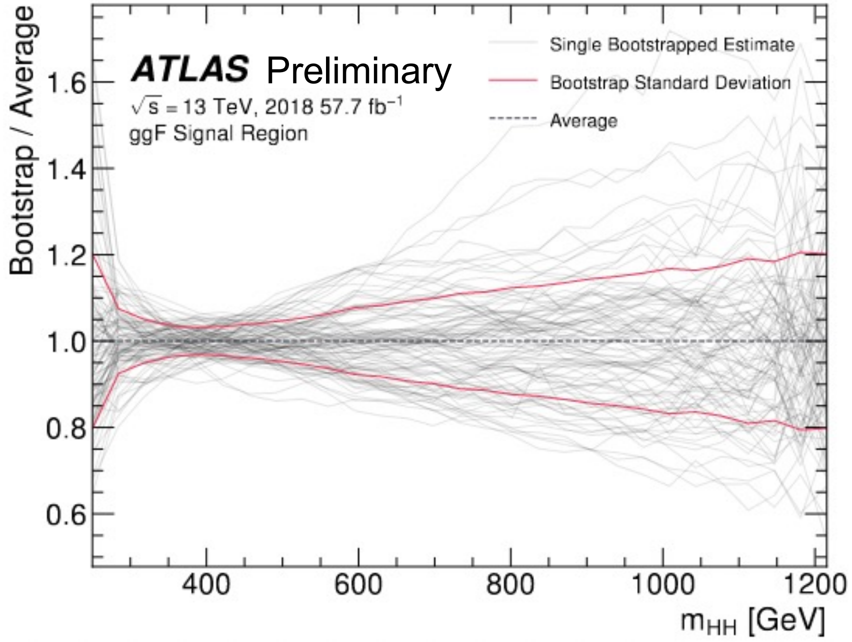
- Learn reweighting using classifiers: $w(x) \approx \frac{p_A(x)}{p_B(x)}$
- Estimate background: $\hat{p}_D(x) = w(x)p_C(x)$



What if we didn't learn accurate weights?

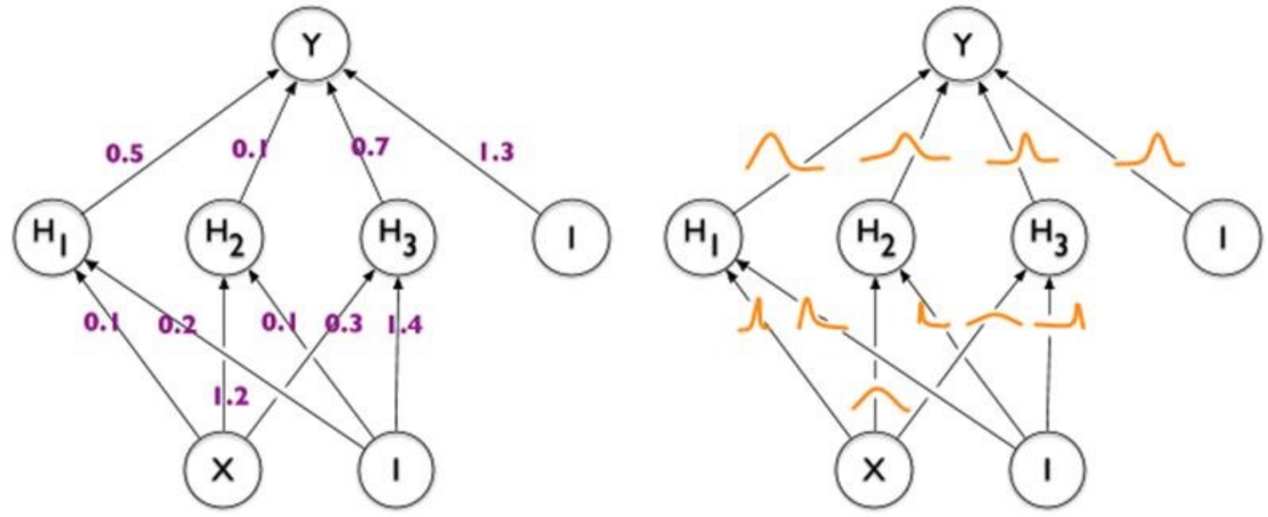
- ATLAS $hh \to 4b$ example: Uncertainties from Deep ensembles & data bootstrap





ATLAS-CONF-2022-035

Bayesian Methods

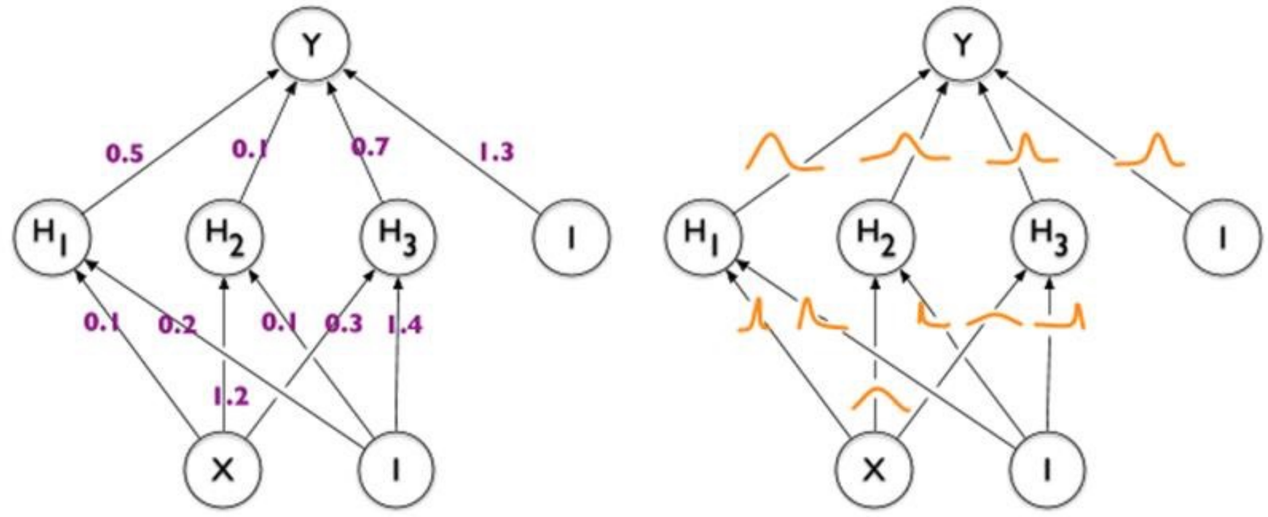


$$p(y|x,\mathcal{D}) = \int p(y|x,w)p(w|\mathcal{D})dw \approx \frac{1}{N} \sum_{\substack{i=1\ldots N \\ w_i \sim p(w|\mathcal{D})}} p(y|x,w_i)$$

Aleatoric Uncertainty:
Density Model

Model Uncertainty:
Posterior on weights

Bayesian Methods



$$p(y|x,\mathcal{D}) = \int p(y|x,w)p(w|\mathcal{D})dw \approx \frac{1}{N} \sum_{\substack{i=1\ldots N \\ w_i \sim p(w|\mathcal{D})}} p(y|x,w_i)$$

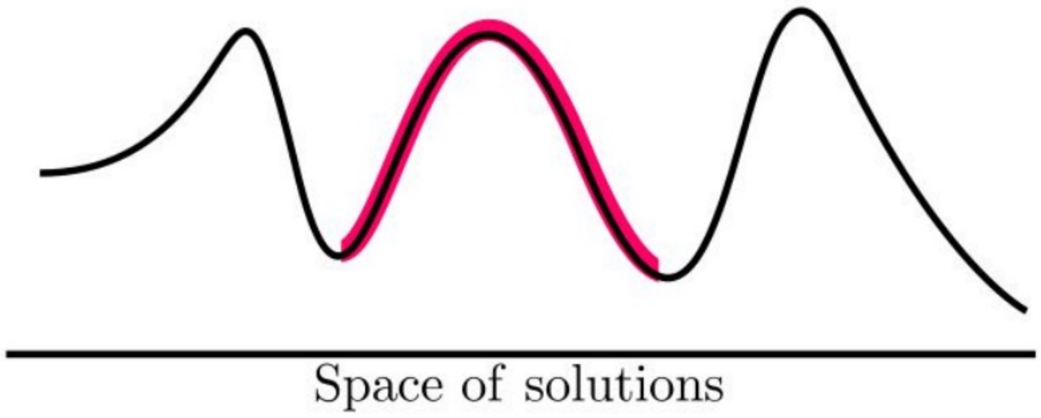
$$p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{\int p(\mathcal{D}|w)p(w)dw}$$

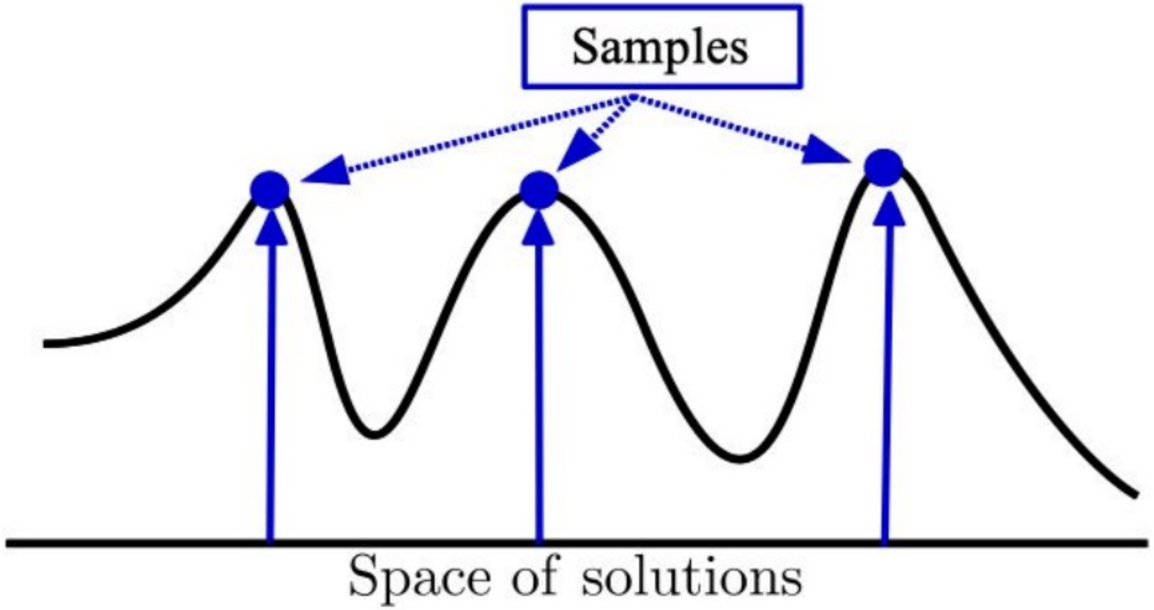
Prior on weights

Intractable Integral

Approximating the Posterior

$p(w|\mathcal{D})$ is multi-modal and complex in NN → approximation methods





Local approximations

- Locally, covering one mode well
  e.g. with a simpler distribution $q(w;\lambda)$
  - Variational inference
  - Laplace approximation

Sampling

- Summarize using samples
  - MCMC
  - Hamiltonian Monte Carlo
  - Stochastic Gradient Langevin Dynamics

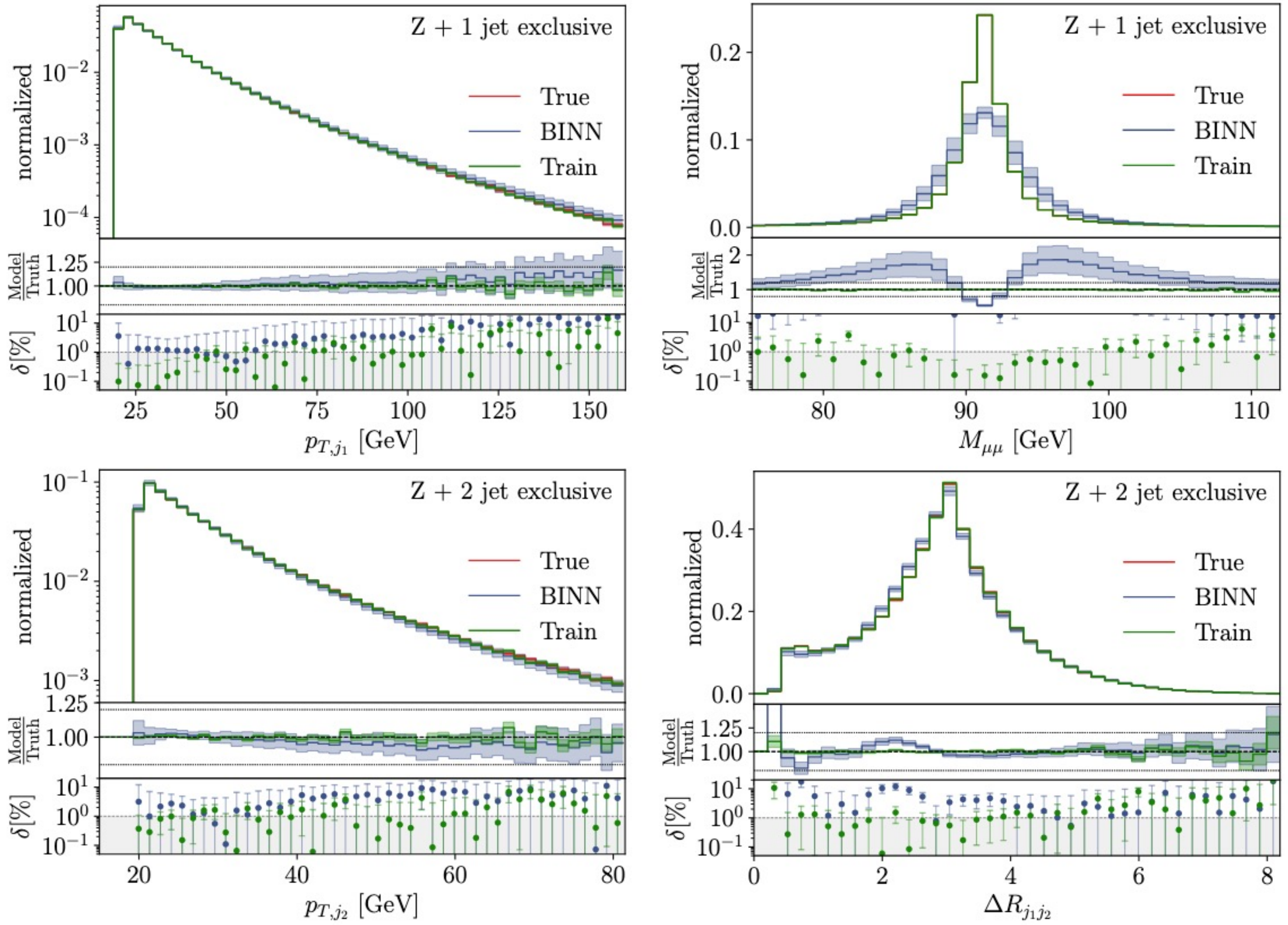
Slide credit: B. Lakshminarayanan

Bayesian Methods for HEP Generative Models

Model Uncertainty on ML models for Event Generators

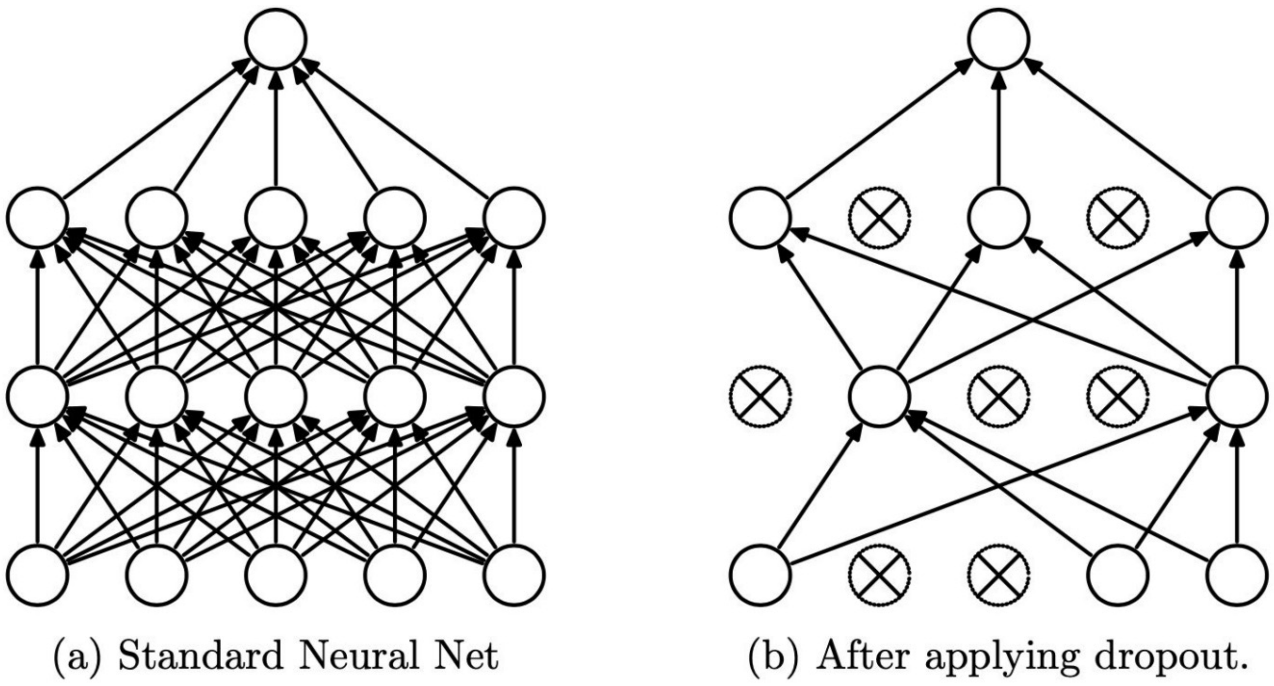
"Bayesian Normalizing Flow"

- Density Model: Normalizing Flow
- Model Uncertainty:
  Variational Posterior over weights



Bellagente et. al, 2104.04543
Butter et. al, 2110.13632

Monte Carlo Dropout



$$f(x) \rightarrow \begin{cases} Mean[f^1 \dots f^N] \\ Var[f^1 \dots f^N] \end{cases}$$

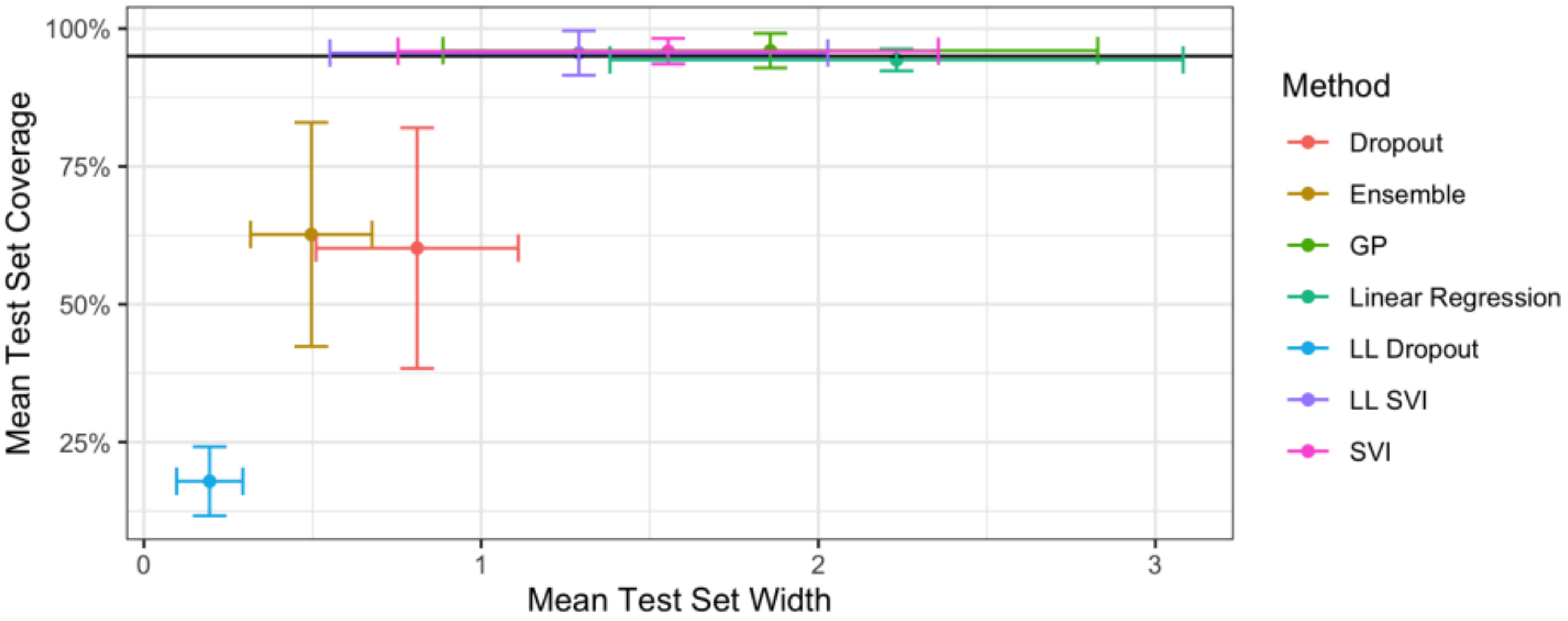
Different random Dropouts

Randomly drop connections between neurons, using Bernoulli distribution

Can be viewed as a Variational Approximation

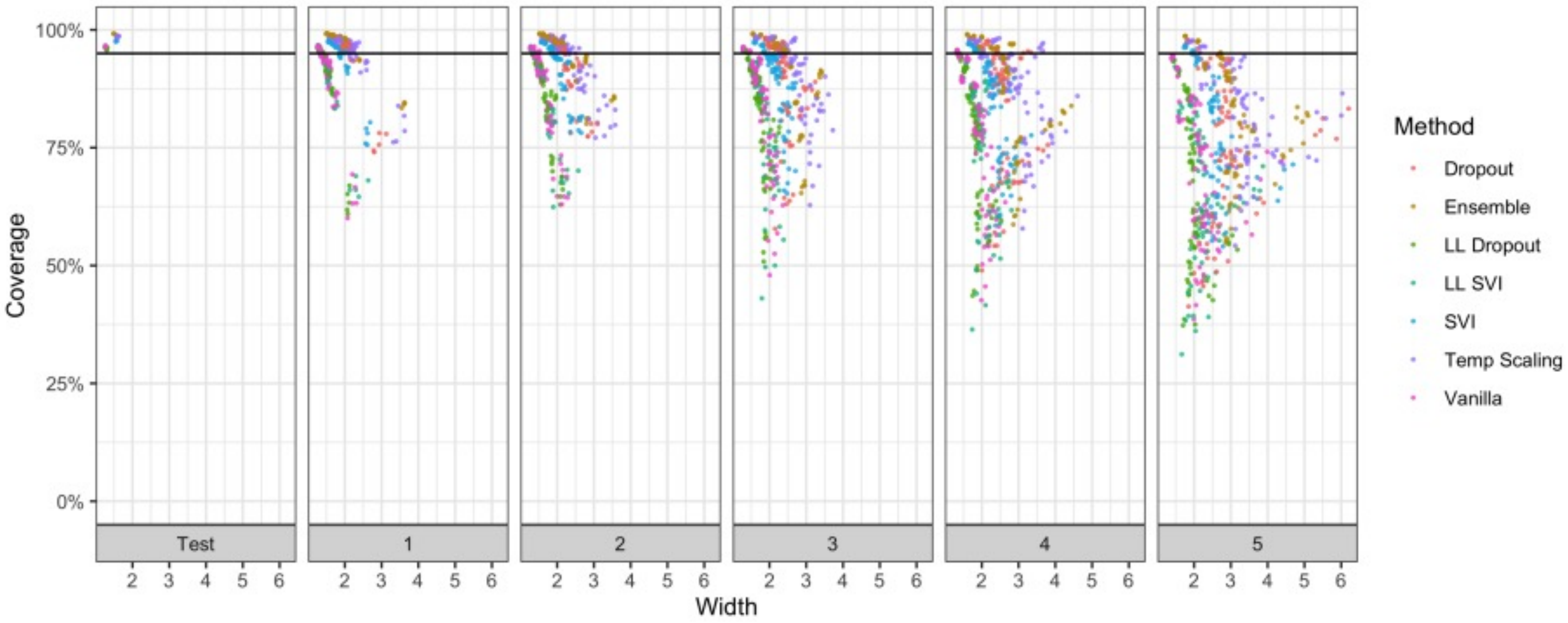
Gal, Ghahramani, [1506.02142](1506.02142)

Comparisons



Kompa et. al, 2010.03039

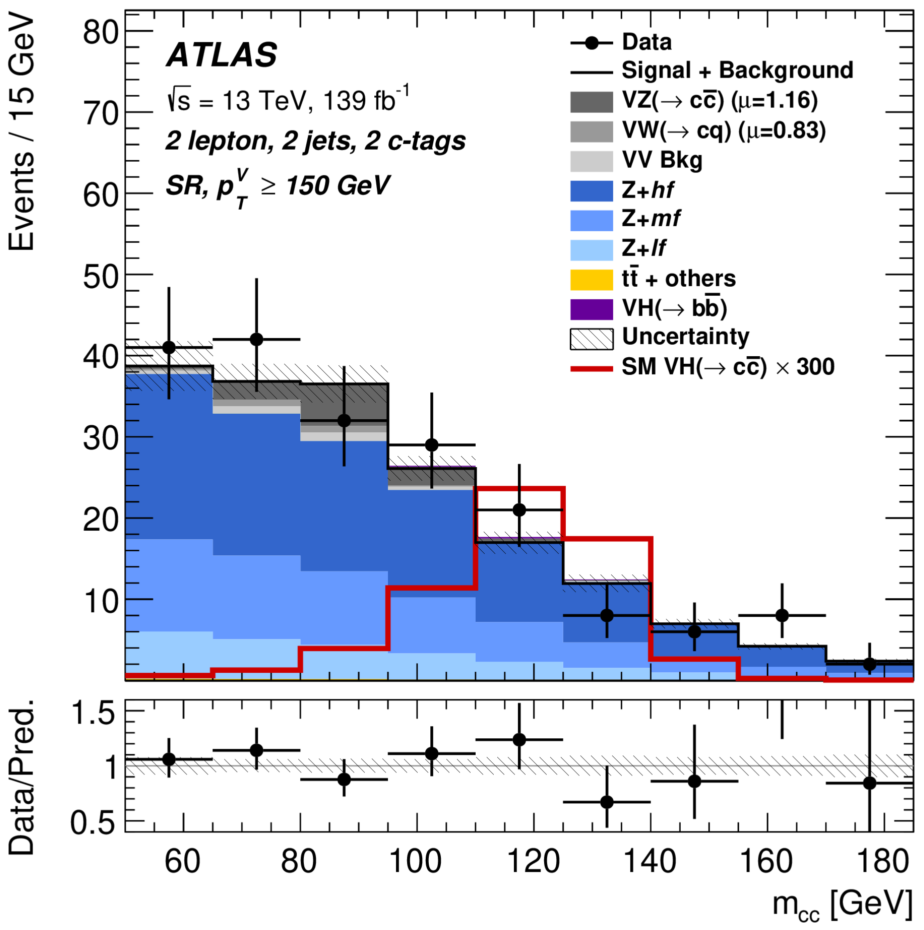
Comparisons with Data Corruptions



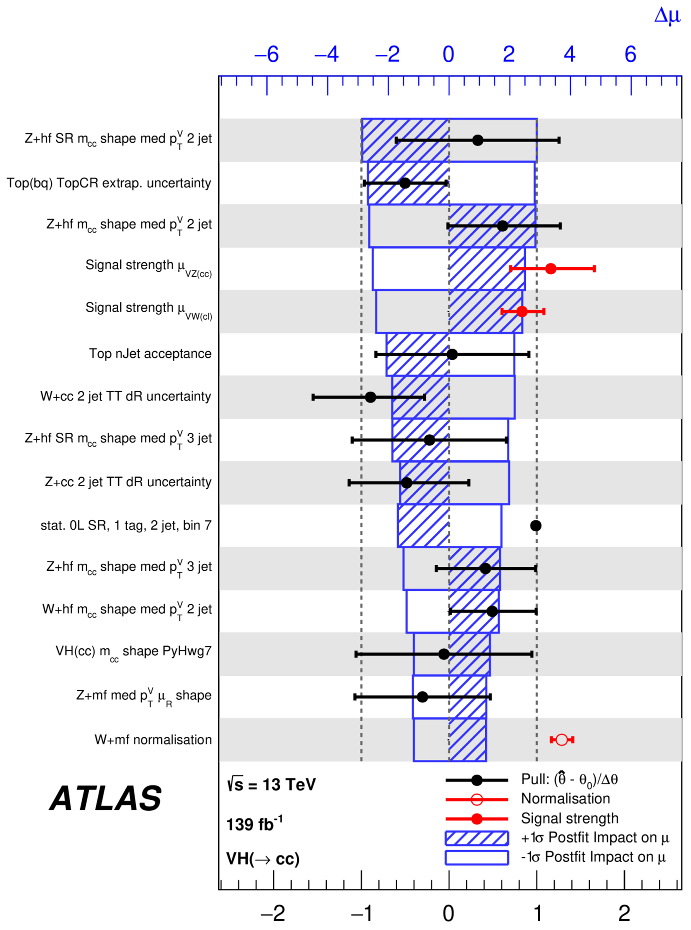
Kompa et. al, 2010.03039

Systematic Uncertainties / Domain Shift in HEP

Systematic Uncertainties



Source of uncertainty		$\mu_{VH(c\bar{c})}$
Total		21.5
Statistical		16.2
Systematics		14.0
Statistical uncertainties		
Data statistics only		13.0
Floating normalisations		7.2
Theoretical and modelling uncertainties		
$VH(\rightarrow c\bar{c})$		2.1
Z+jets		7.7
Top-quark		5.6
W+jets		3.4
Diboson		0.8
$VH(\rightarrow b\bar{b})$		0.8
Multi-Jet		1.0
Simulation statistics		5.1
Experimental uncertainties		
Jets		3.7
Leptons		0.4
$E_{\mathrm{T}}^{\mathrm{miss}}$		0.5
Pile-up and luminosity		0.4
Flavour tagging	c-jets	2.3
	b-jets	1.2
	light-jets	0.7
	τ-jets	0.4
Truth-flavour tagging	ΔR correction	3.0
	Residual non-closure	1.4

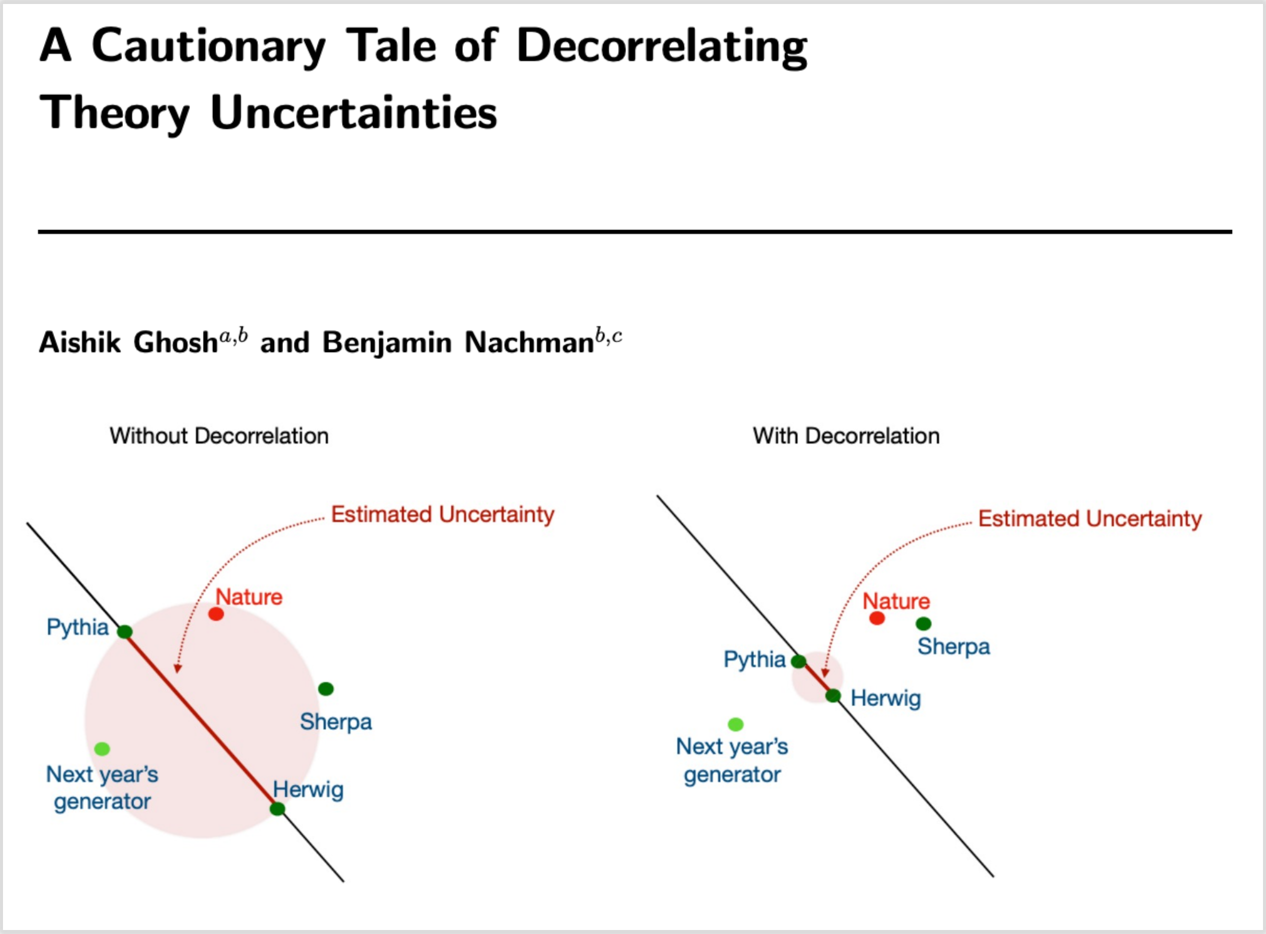


arXiv:2201.11428

Theory uncertainties? … Not going to discuss here

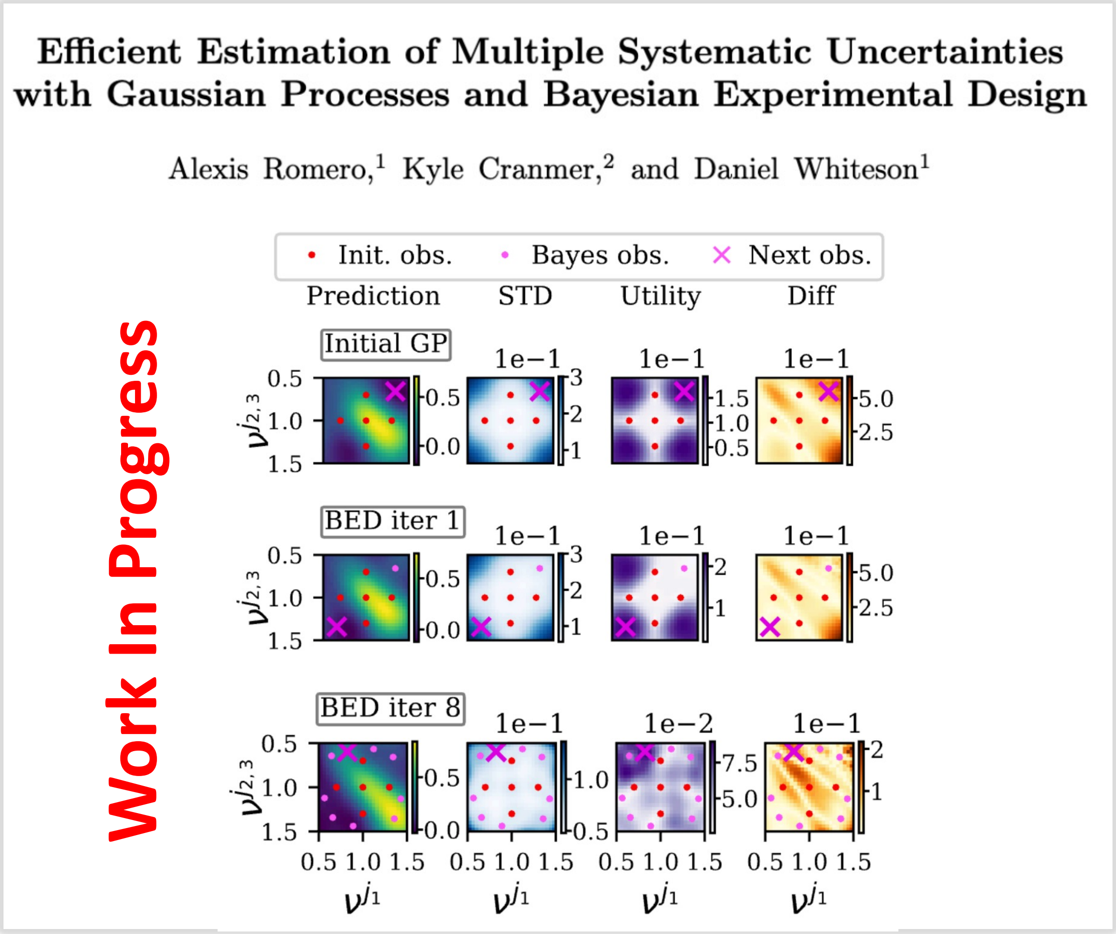
See nice recent PHYSTAT talk from D. Whiteson

See nice recent paper: Ghosh, Nachman, 2109.08159



A Cautionary Tale of Decorrelating Theory Uncertainties

Aishik Ghosh[a,b] and Benjamin Nachman[b,c]



Efficient Estimation of Multiple Systematic Uncertainties with Gaussian Processes and Bayesian Experimental Design

Alexis Romero,[1] Kyle Cranmer,[2] and Daniel Whiteson[1]

Systematic Uncertainty = Domain / Distribution Shift

Unlike ML, we measure / parameterize possible variations over domains

$$x \sim p(x|\theta, \nu)$$

Parameterized family of likelihood models

Nuisance Parameter: Parameterizing variations

Often can constrain from auxiliary measurements: $p(x_{aux}|\nu)$
(i.e. from calibrations for reconstructed objects)

How to deal w/ systematic uncertainties in HEP-ML models?

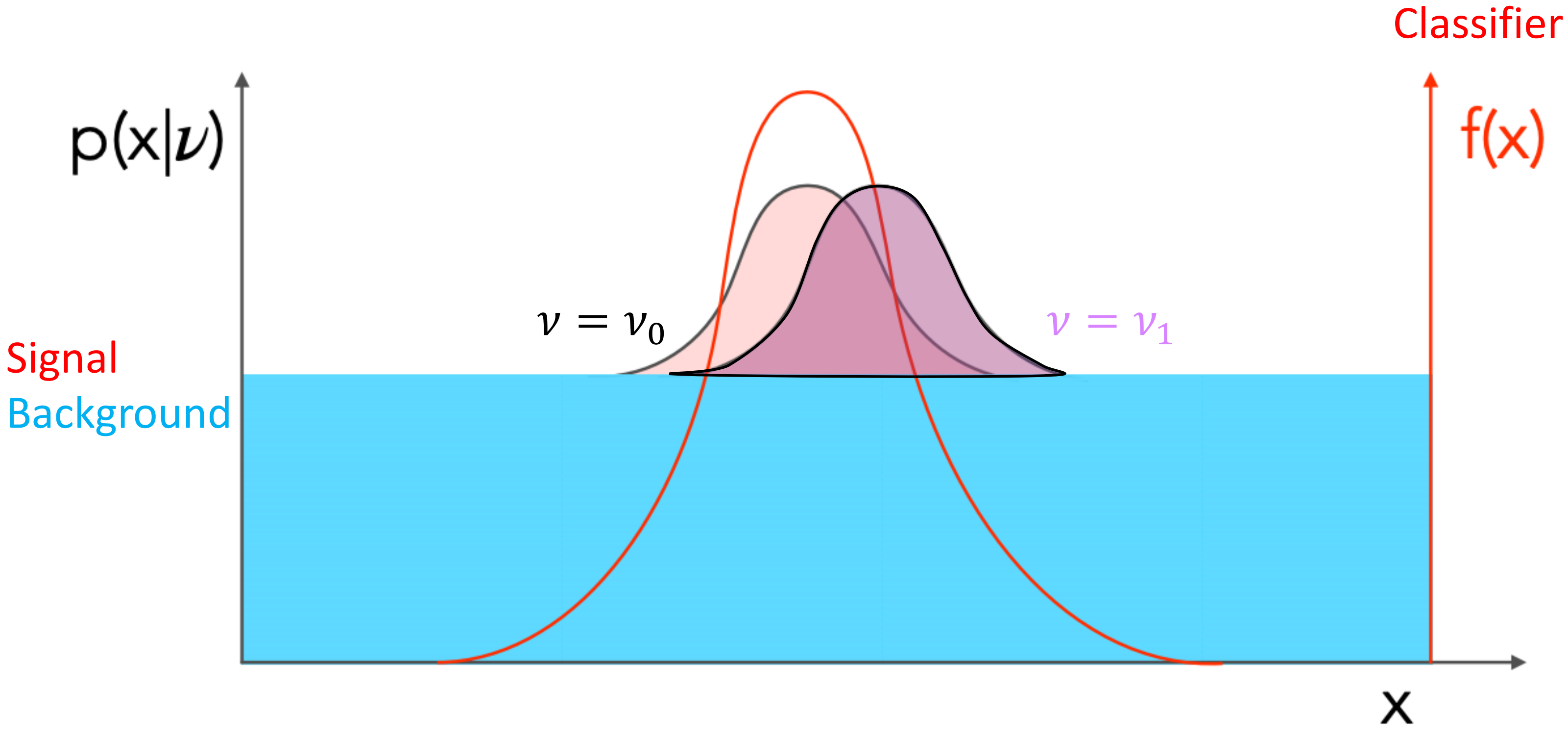
- **propagation of errors:** one works with a model $f(x)$ and simply characterizes how uncertainty in the data distribution propagate through the function to the down-stream task irrespective of how it was trained.
- **domain adaptation:** one incorporates knowledge of the distribution for domains (or the parameterized family of distributions $p(x|y,\nu)$) into the training procedure so that the performance of $f(x)$ for the down-stream task is robust or insensitive to the uncertainty in ν.
- **parameterized models:** instead of learning a single function of the data $f(x)$, one learns a family of functions $f(x;\nu)$ that is explicitly parameterized in terms of nuisance parameters and then accounts for the dependence on the nuisance parameters in the down-stream task.
- **data augmentation:** one trains a model $f(x)$ in the usual way using training dataset from multiple domains by sampling from some distribution over ν.

From PDG review of ML in HEP

Error Propagation – Standard Approach

Train on $\{x_i^0, y_i^0\}$ w/ nominal nuisance ν_0 → learn fixed model $f(x)$

Evaluate on ν variations → Observe effects



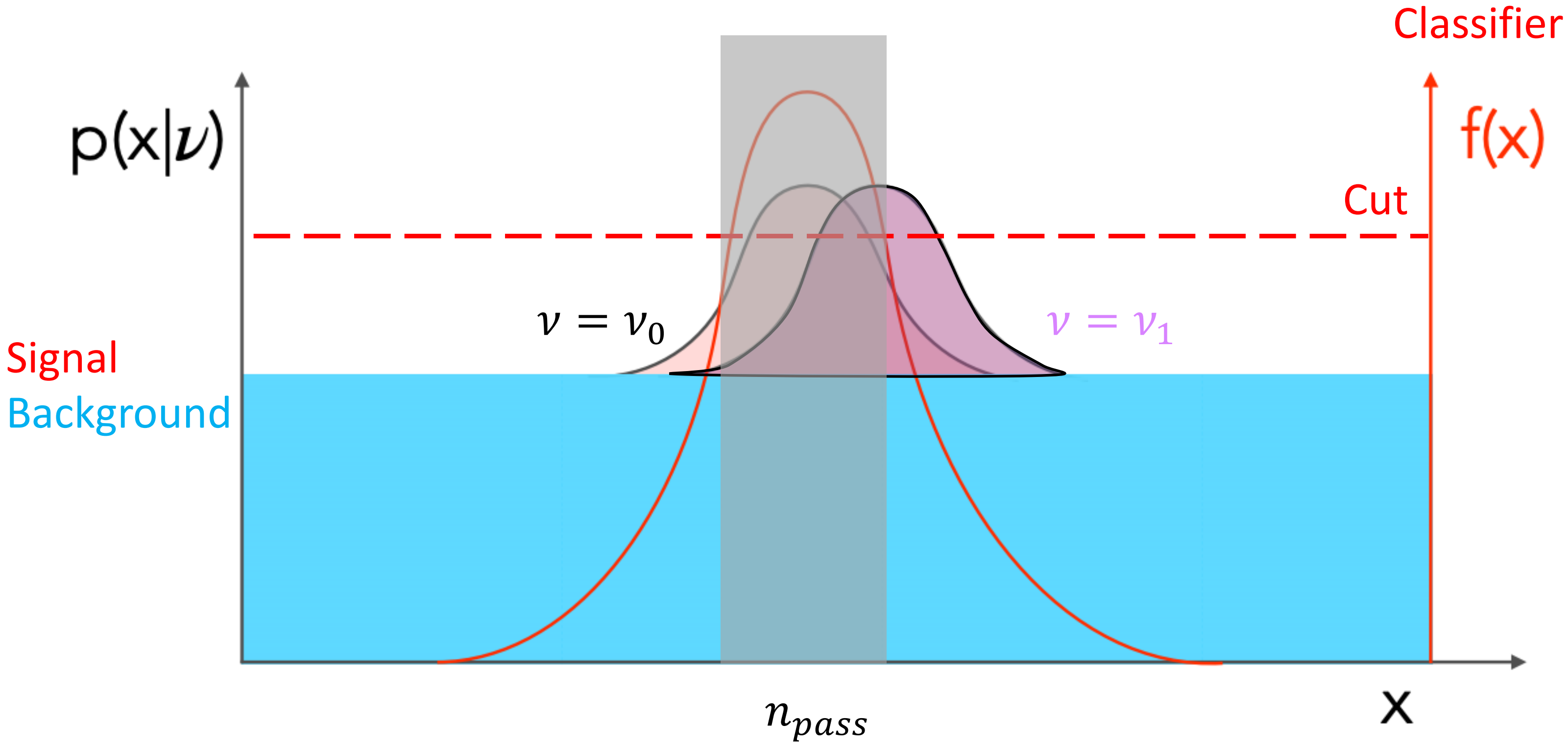
Slide Credit: K. Cranmer

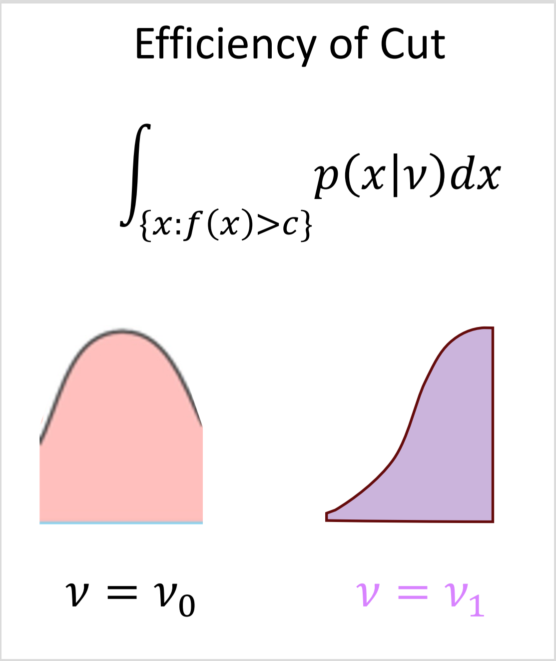
Error Propagation – Standard Approach

Train on $\{x_i^0, y_i^0\}$ w/ nominal nuisance ν_0 → learn fixed model $f(x)$

Evaluate on ν variations → Observe effects

NOT Optimal
Under Variations



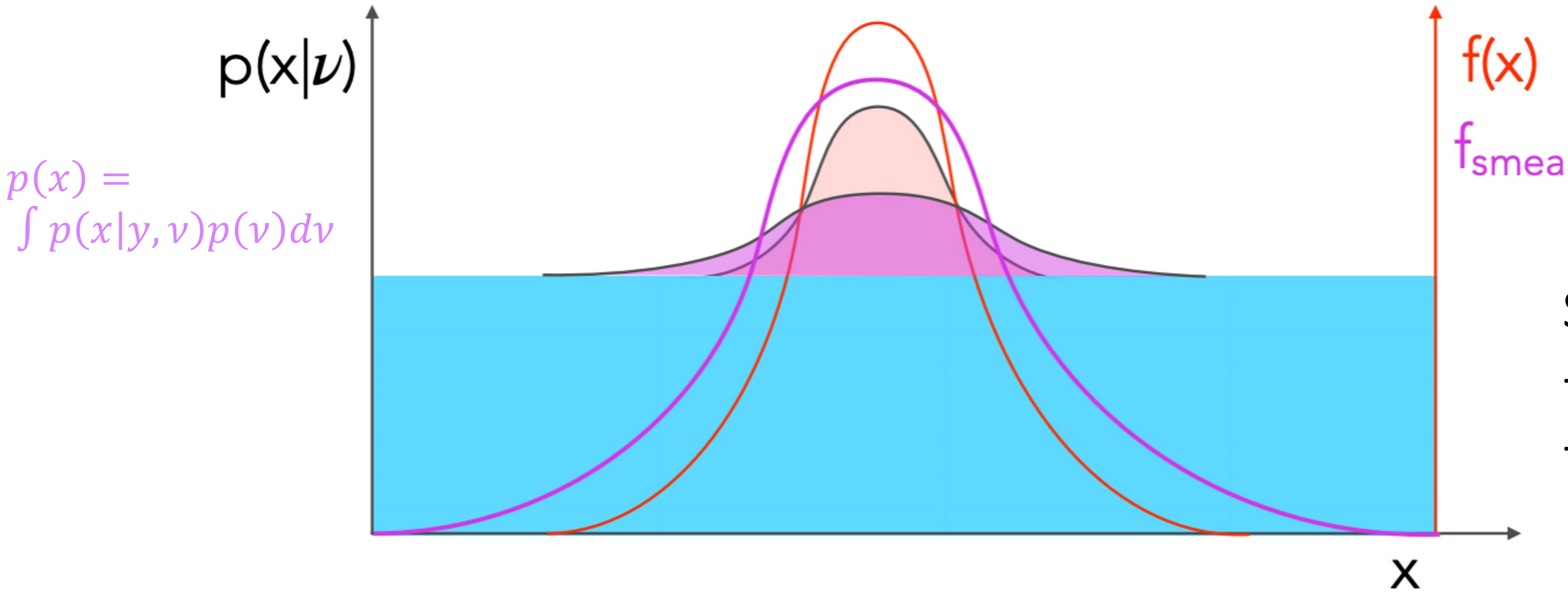


Slide Credit: K. Cranmer

Data Augmentation / Marginalization

Training sample includes ν variations: $x \sim \int p(x|y,\nu)p(\nu)d\nu$

Smeared samples → “smeared” fixed model $f_{smeared}(x)$



Smeared Classifier:

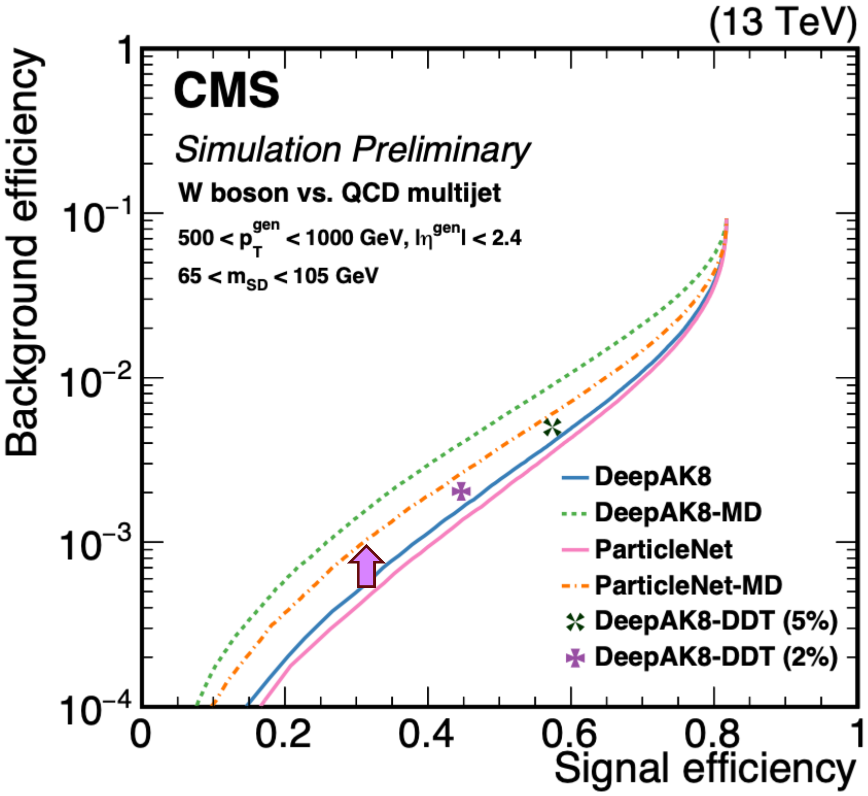
- Less sensitive to variations
- Not optimal for any ν

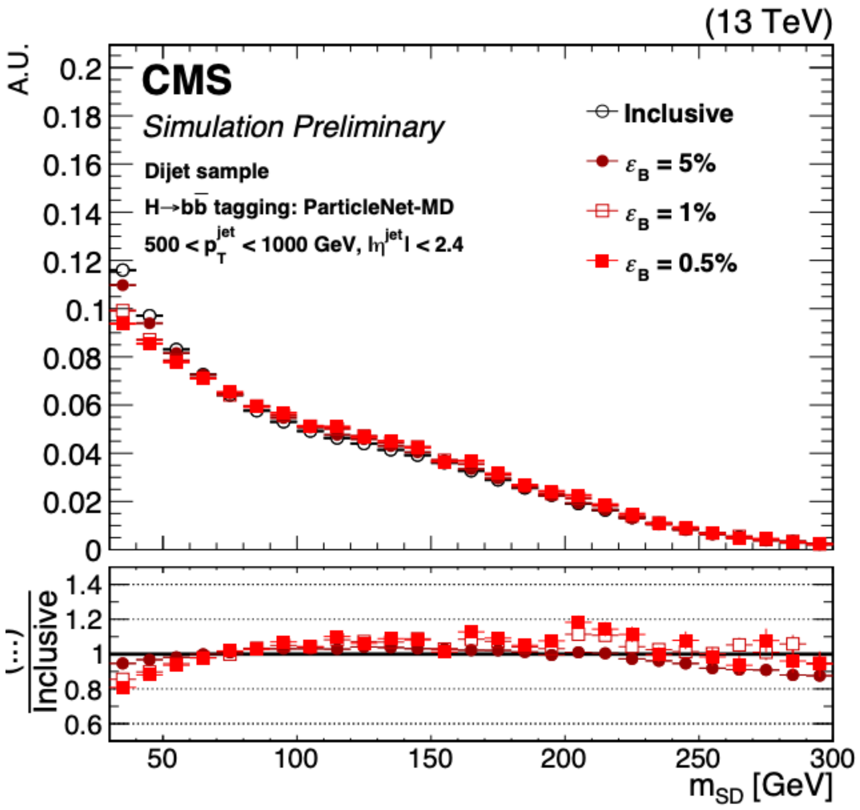
Slide Credit: K. Cranmer

Data Augmentation / Marginalization

Training sample includes ν variations: $x \sim \int p(x|y,\nu)p(\nu)d\nu$

Smeared samples → "smeared" fixed model $f_{smeared}(x)$





Related Example:
CMS Boosted Jet Tagging w/ ParticleNet Graph NN

Training on flat mass distribution

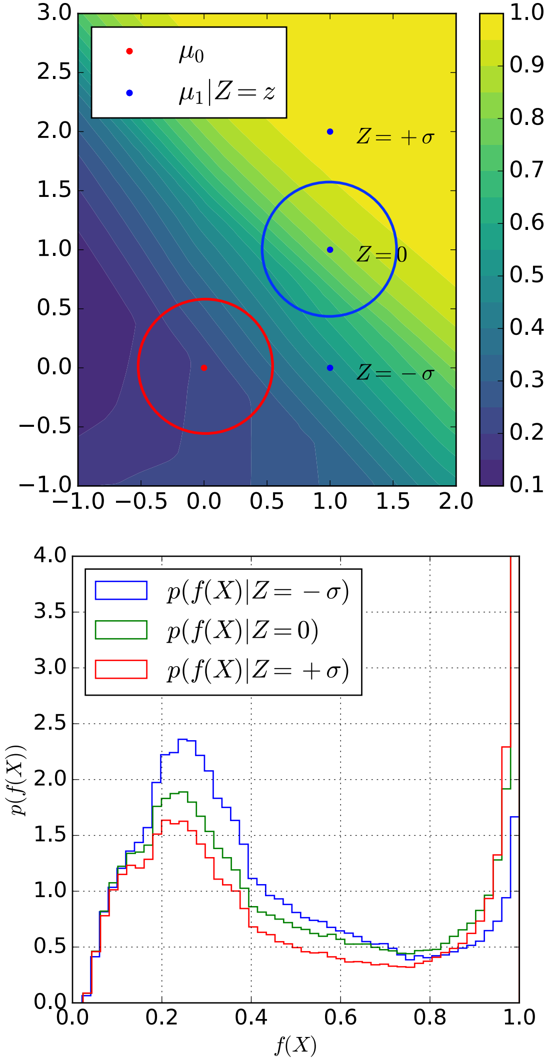
CMS-DP-2020-002

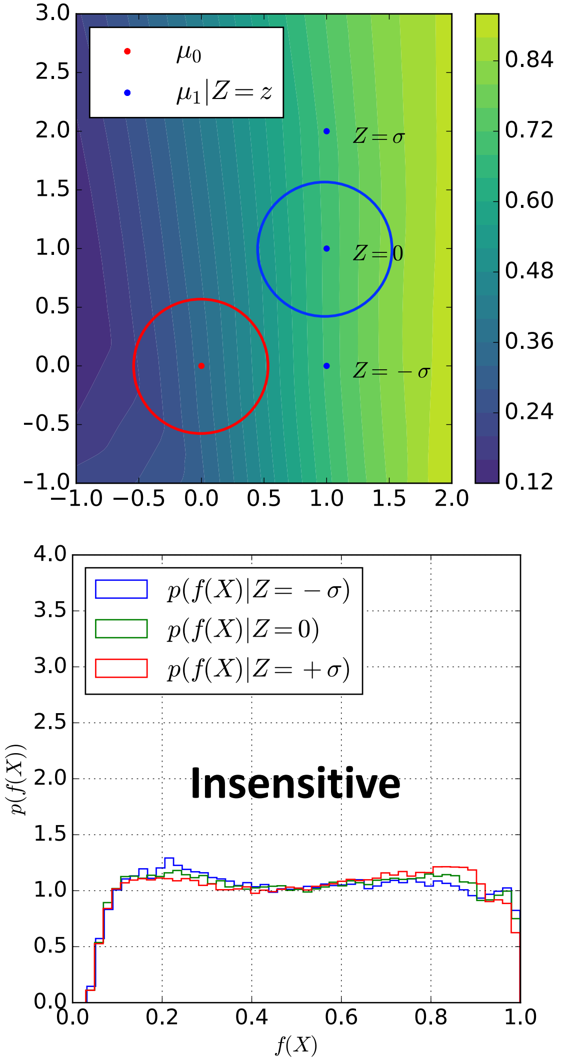
Pivoting / Enforcing Domain Invariance

Want to train model $f(x)$ such that: $p(f|\nu) = p(f)$

f is a *pivotal quantity*

Normal Training





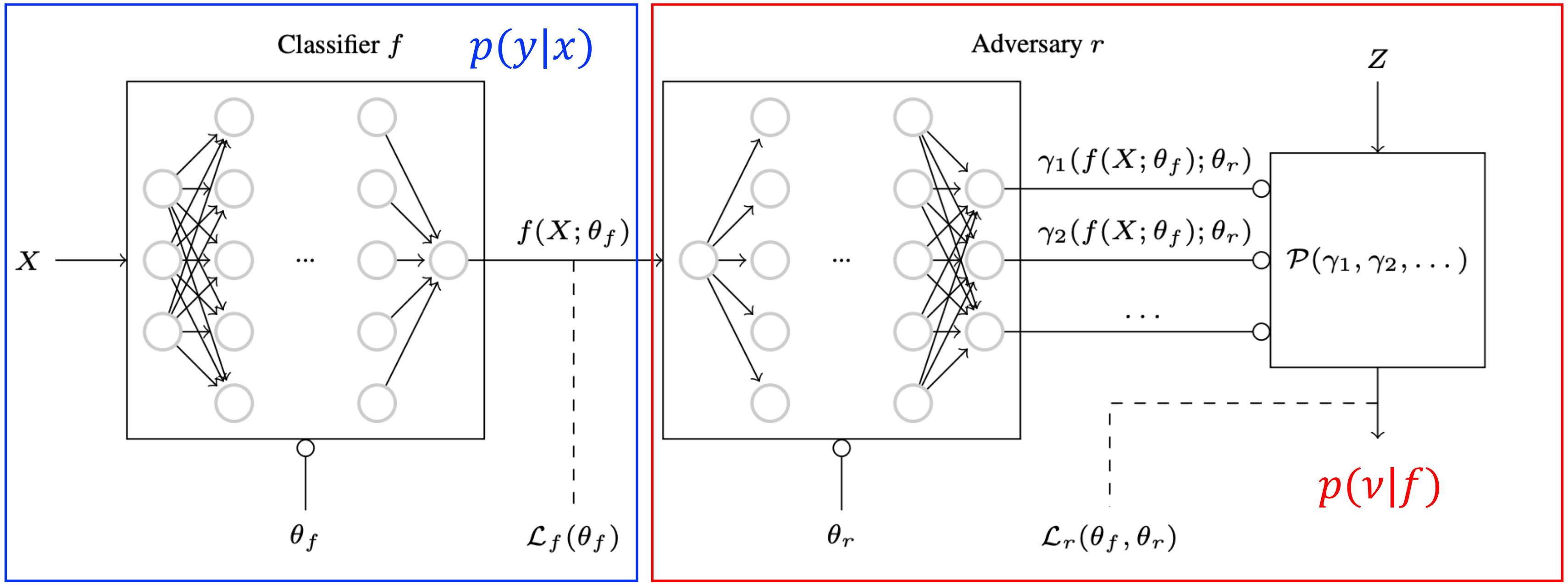
Pivot

Louppe, MK, Cranmer, [1611.01046](1611.01046)

Pivoting / Enforcing Domain Invariance

Adversarial Approach:

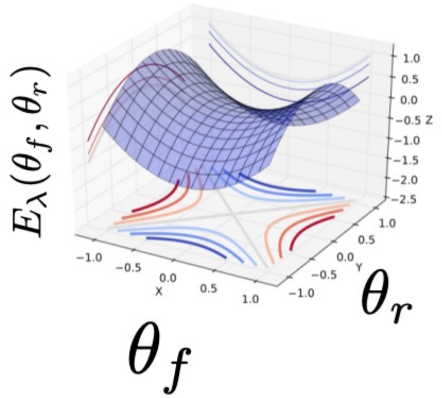
Adversary to predict ν from model output f



Min-Max Game: Penalize Classifier when Adversary succeeds

$$\hat{\theta}_f,\hat{\theta}_r = \arg\min_{\theta_f}\max_{\theta_r} E(\theta_f,\theta_r).$$

$$E_\lambda(\theta_f,\theta_r) = \mathcal{L}_f(\theta_f) - \lambda\mathcal{L}_r(\theta_f,\theta_r)$$

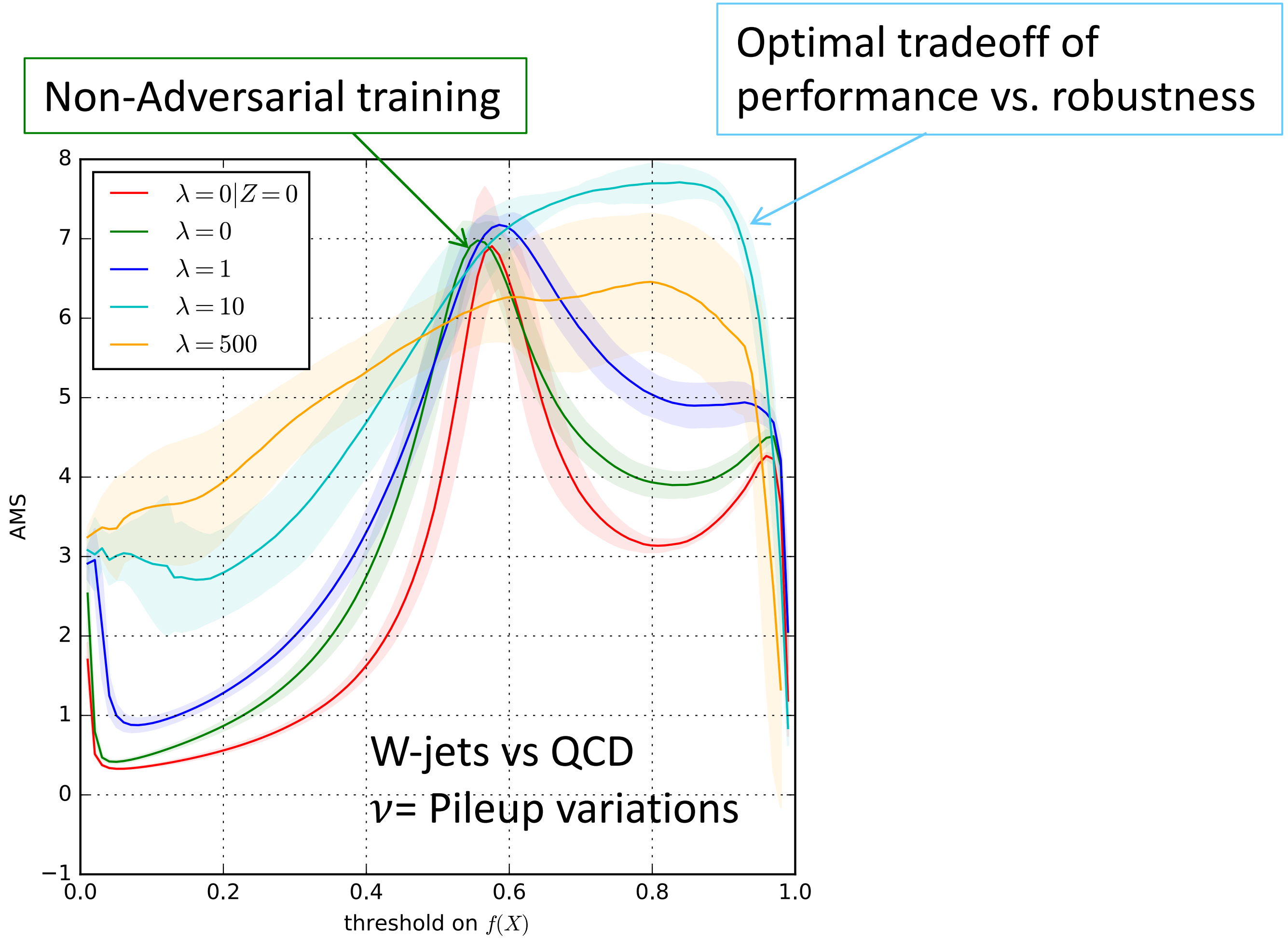


"Regularize" training with Adversary

Louppe, MK, Cranmer, [1611.01046](1611.01046)

Pivoting / Enforcing Domain Invariance

Adversarial Approach:



Louppe, MK, Cranmer, [1611.01046](1611.01046)

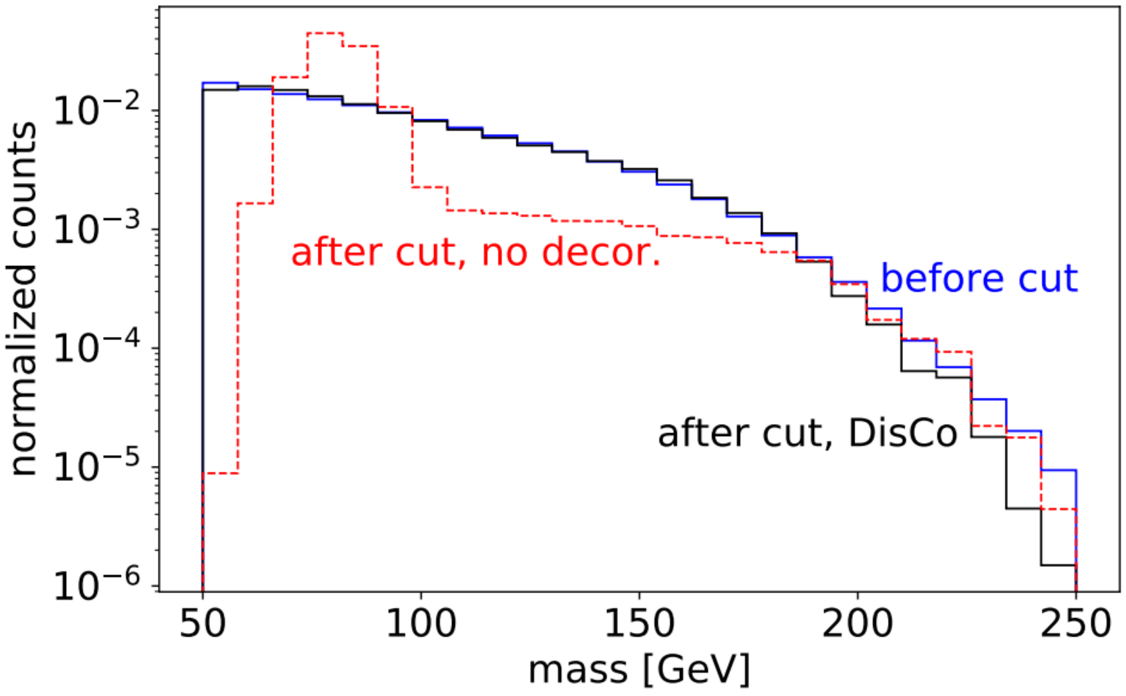
Pivoting / Enforcing Domain Invariance

Regularizing Correlations: Non-adversarial approach

Example: *Disco Fever: Robust Networks Through Distance Correlation*

$$L = L_{classifier}(\vec{y}, \vec{y}_{true}) + \lambda\, \mathrm{dCorr}^2_{y_{true}=0}(\vec{m}, \vec{y})$$

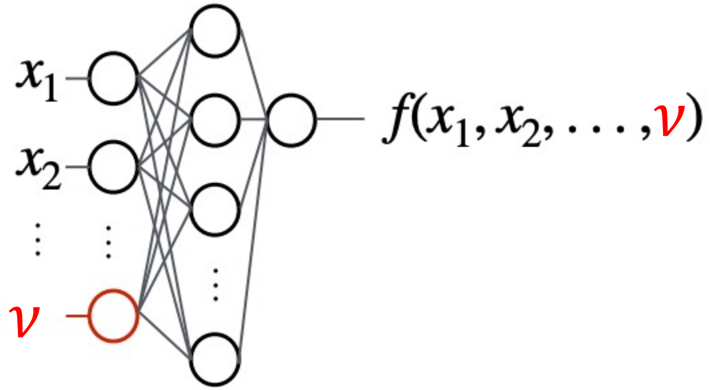
$$\begin{aligned}\mathrm{dCov}^2(X,Y) = &\langle |X - X'||Y - Y'|\rangle \\ &+ \langle |X - X'|\rangle\langle |Y - Y'|\rangle \\ &- 2\langle |X - X'||Y - Y''|\rangle\end{aligned}$$

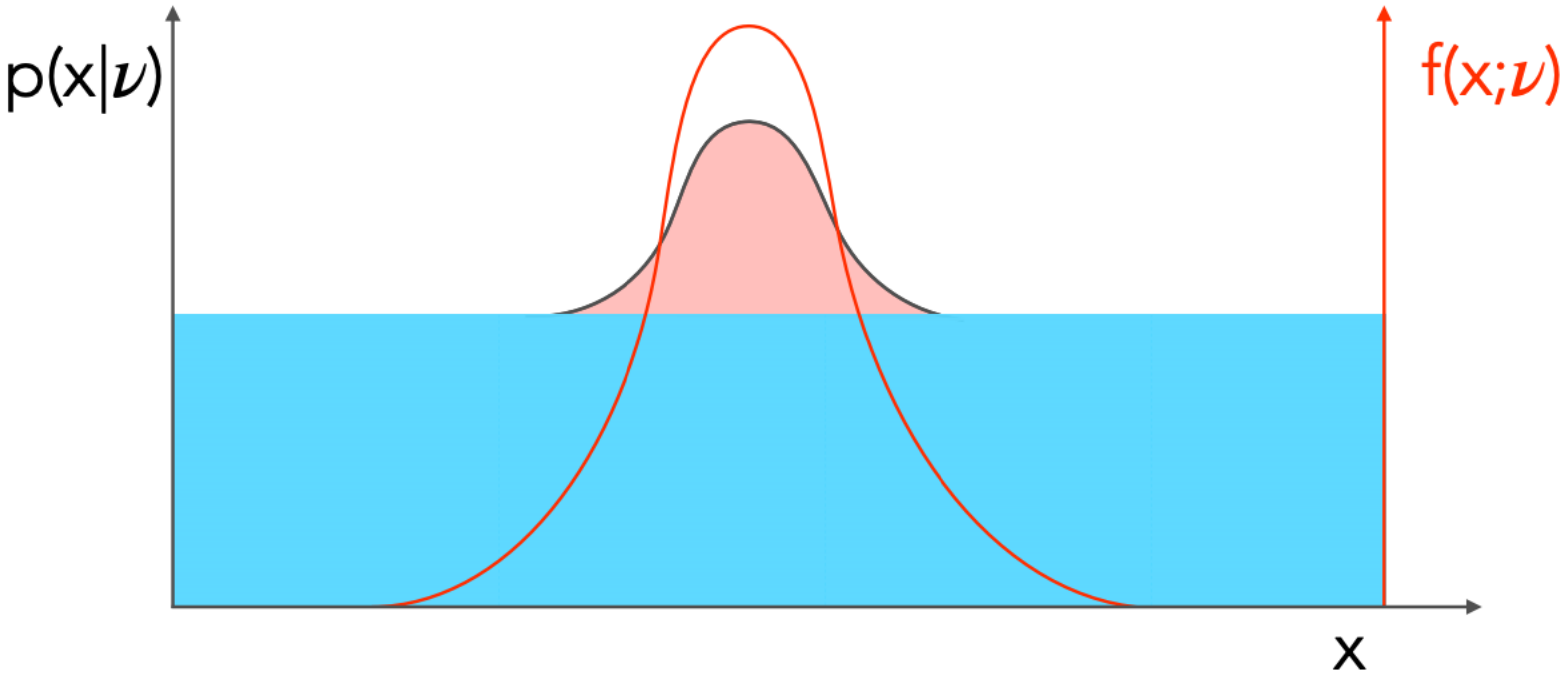


Kasieczka, Shih, [2001.05310](2001.05310)

Parameterizing Models

Train with nuisance parameters as input $\{x_i, y_i, \nu_i\} \rightarrow$ learn model $f(x; \nu)$



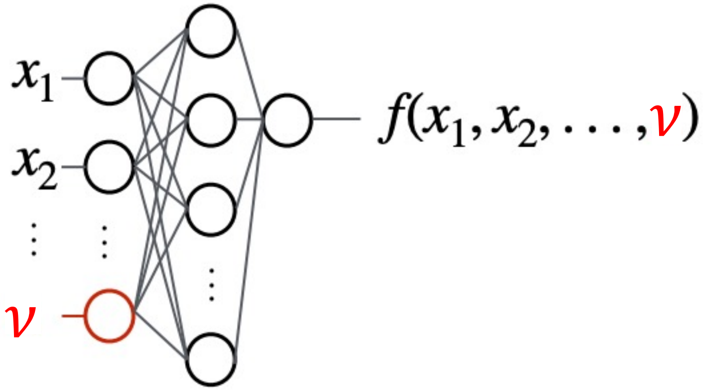


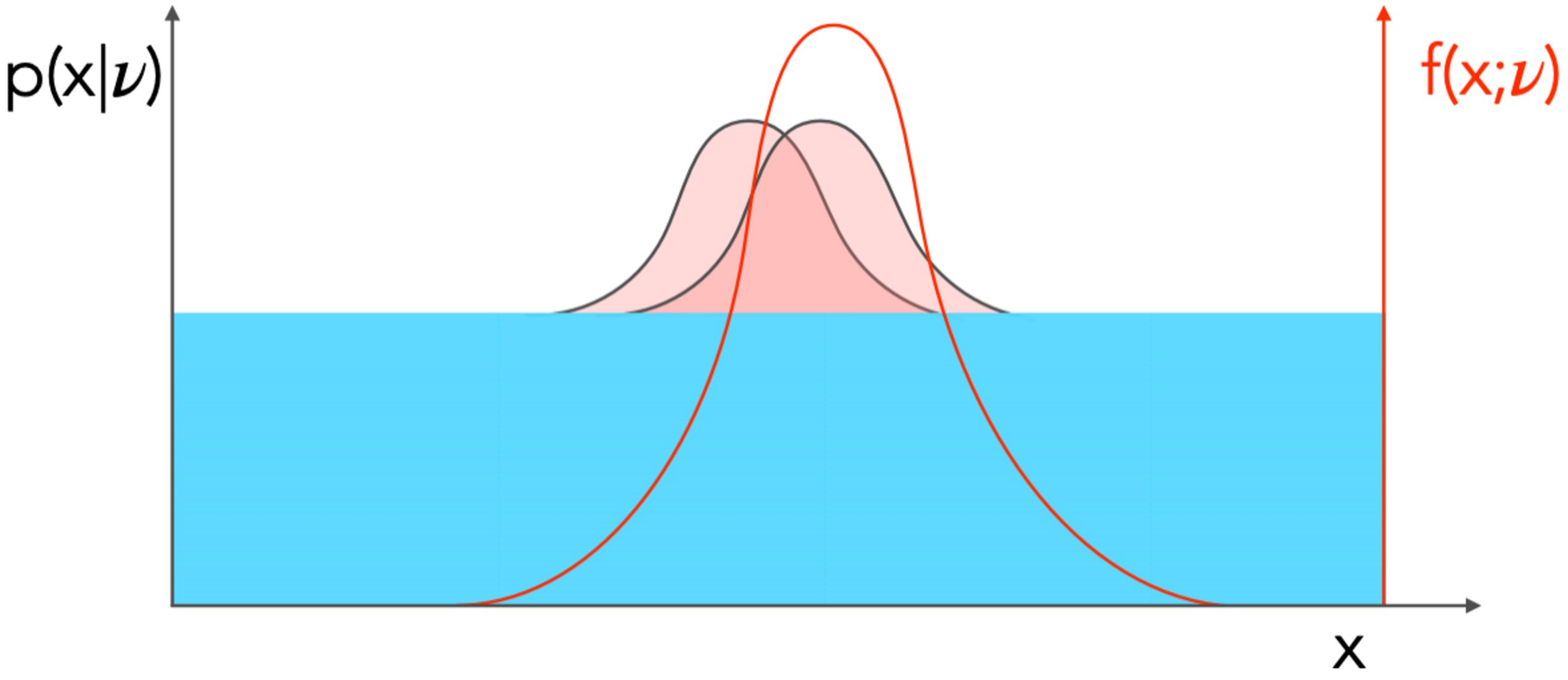
Cranmer, Louppe, Pavez, 1506.02169

Slide Credit: K. Cranmer

Parameterizing Models

Train with nuisance parameters as input $\{x_i, y_i, \nu_i\} \rightarrow$ learn model $f(x; \nu)$





Cranmer, Louppe, Pavez, 1506.02169

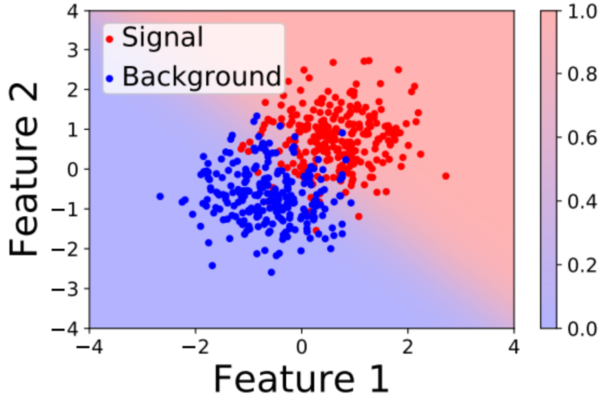
Slide Credit: K. Cranmer

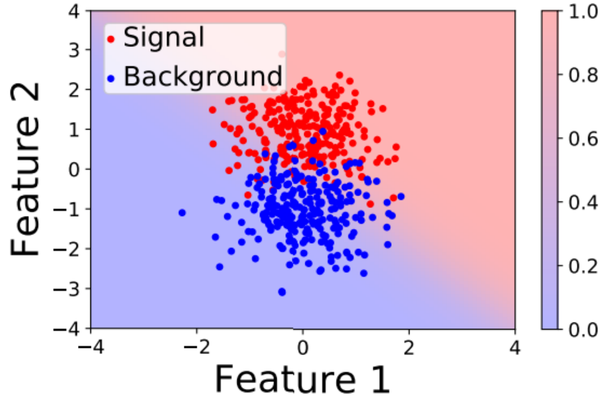
Parameterizing Models

No Systematic Variation

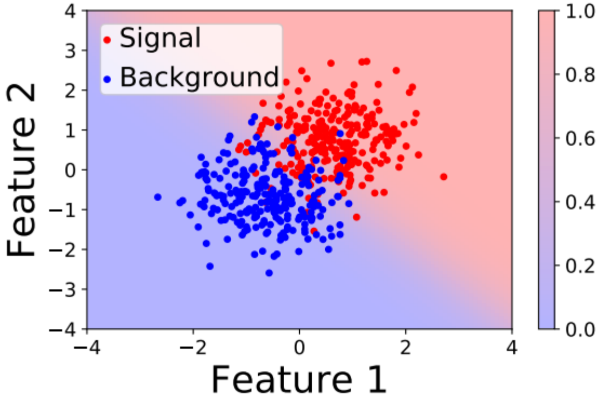
With Systematic Variation

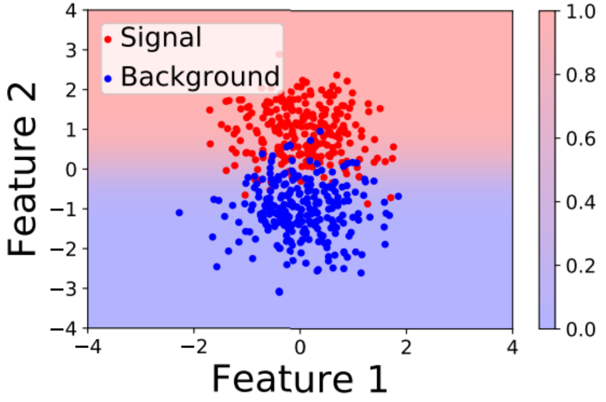
Fixed Model





Parameterized Model





Ghosh, Nachman, Whiteson, [2105.08742](2105.08742)

Comparing Approaches

Example:

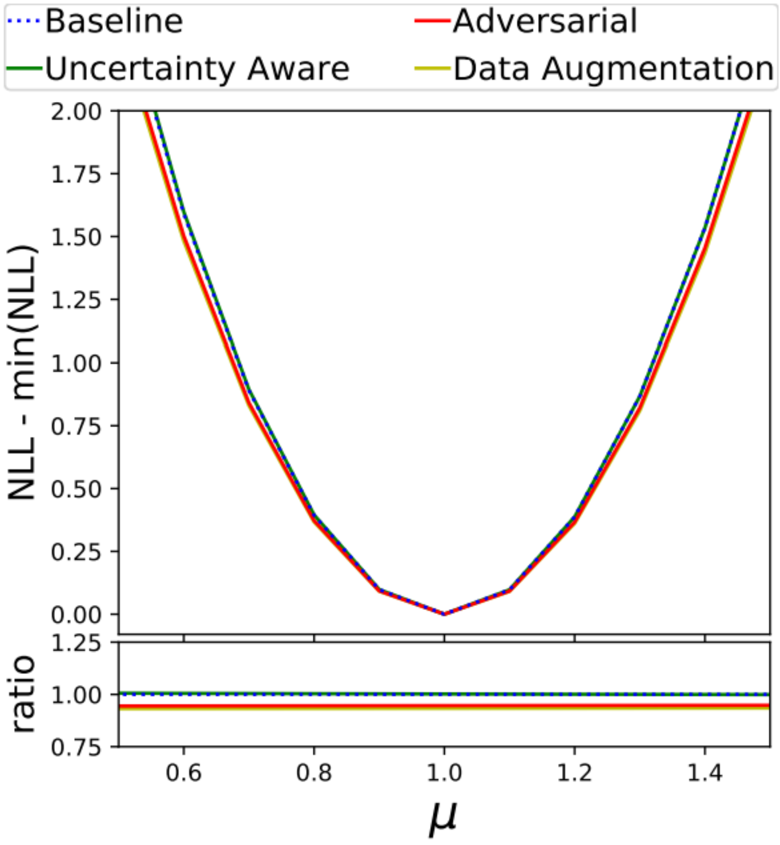
- Classifier: $h \to \tau\tau$ vs Bkg
- Uncertainty: τ energy scale

Parameterized Classifier:

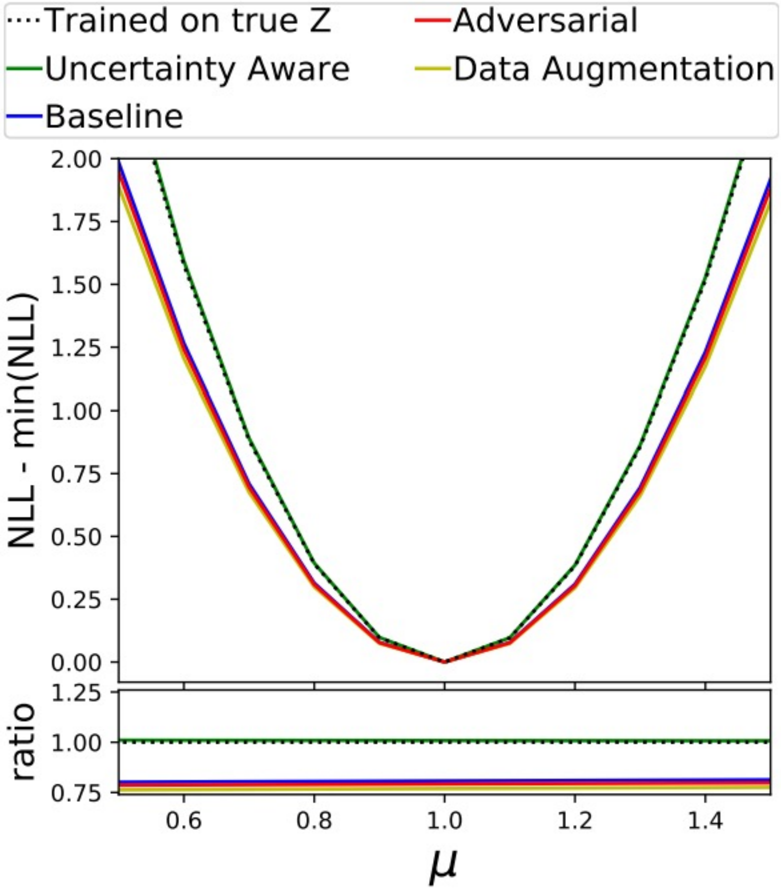
$$f(x;\nu)$$

How to choose the ν?
$\rightarrow$Profile in Likelihood

Cranmer, Louppe, Pavez, 1506.02169



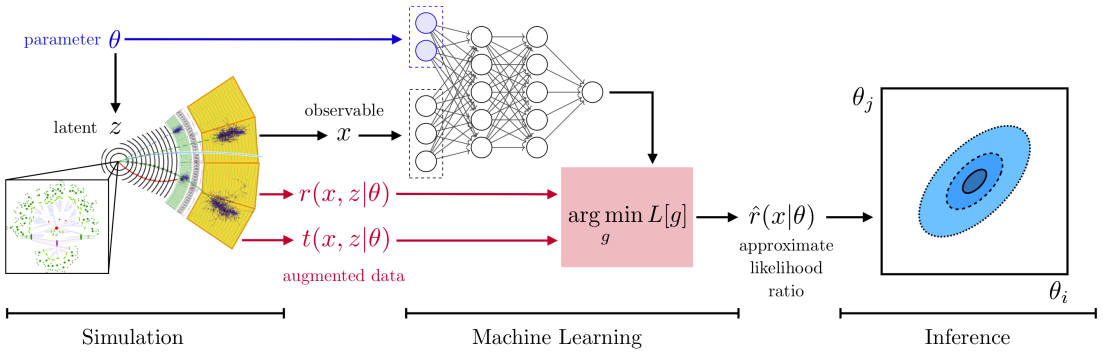
Data with Nominal Nuisance value

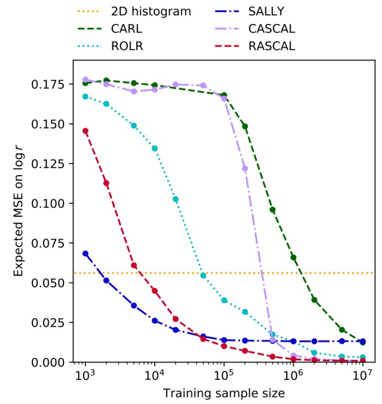


Data with Varied Nuisance value

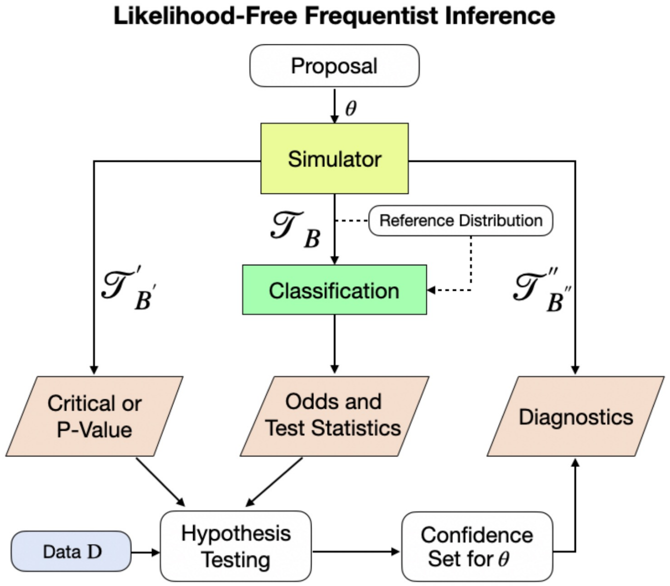
Ghosh, Nachman, Whiteson, 2105.08742

Simulation-Based Inference:
Estimating Likelihood Ratios with parameterized Models



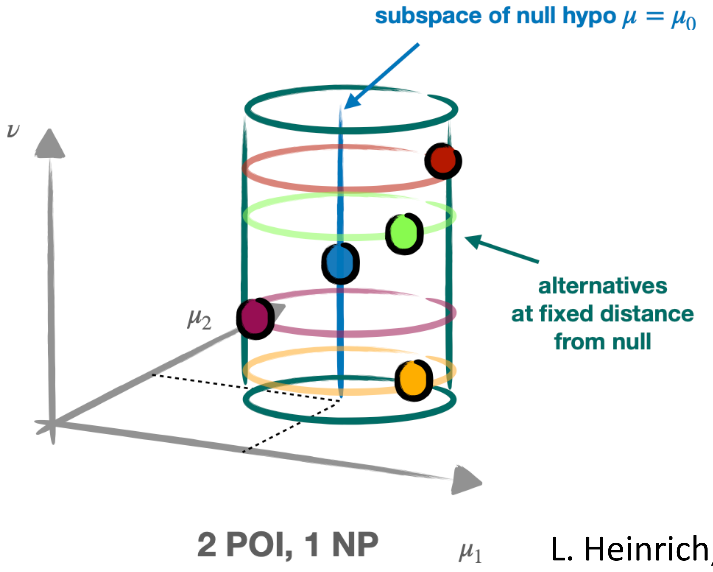


Brehmer, Louppe, Pavez, Kling, Espejo, Cranmer [1, 2, 3]



Dalmasso, Masserano, Zhao, Izbicki, Lee, 2107.03920
Dalmasso, Izbicki, Lee, 2002.10399

Learning Profile Likelihood Ratios



L. Heinrich, 2203.13079

Summary

Uncertainty when using ML in HEP → How and Where?

- Lots of ML research on estimating Data uncertainty & Model Uncertainty
- Must examine each application & how well calibrated the methods are?

Many areas where Model Uncertainty may be important (not all discussed today)

- ML-based Simulation and Background estimation
- Fast ML in the Trigger – Uncertainty in real-time decision making
- Simulation-based inference – estimating likelihood ratio directly with ML
- Anomaly Detection
- …

Systematics will always remain a challenge, and understanding how to deal with them in ML models has made progress on several fronts

Backup

Standard HEP Inference

Reconstruction, data selection, event classification enable us to define powerful summary statistics

$$T(x)\colon\ \mathbb{R}^{10^8} \rightarrow \mathbb{R}$$

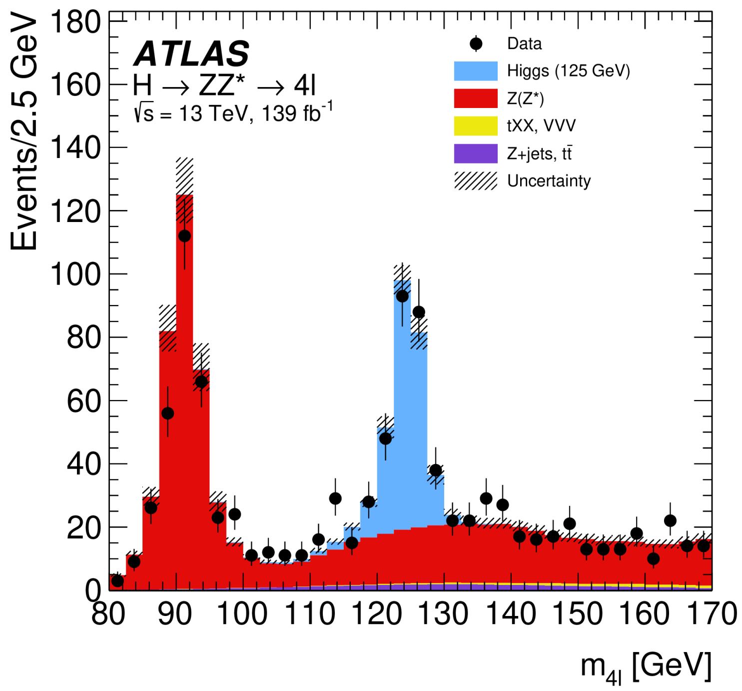
Histogram for density estimation, with bin counts:

$$\{t_i\}_{i=1\ldots n_{bins}}$$

Binned Likelihood: $p(t_i|\theta,\nu) = Poiss(t_i|\mu(\theta,\nu))$

Test Statistic: $\lambda(\theta) = \log\frac{\prod_i p(t_i|\theta,\hat{\hat{\nu}})}{\prod_i p(t_i|\hat{\theta},\hat{\nu})}$

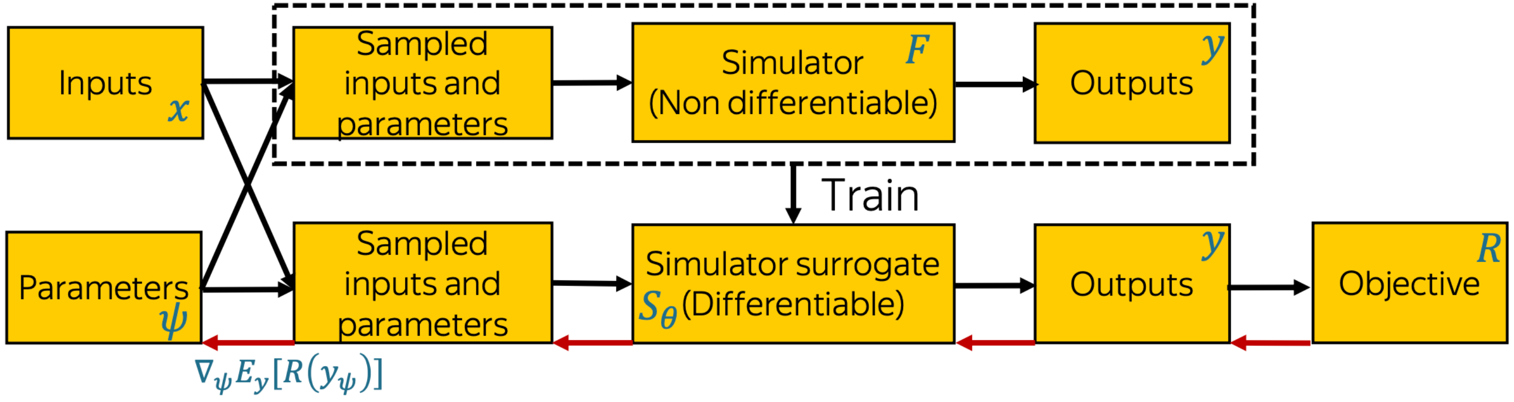
EPJC 80 (2020) 942

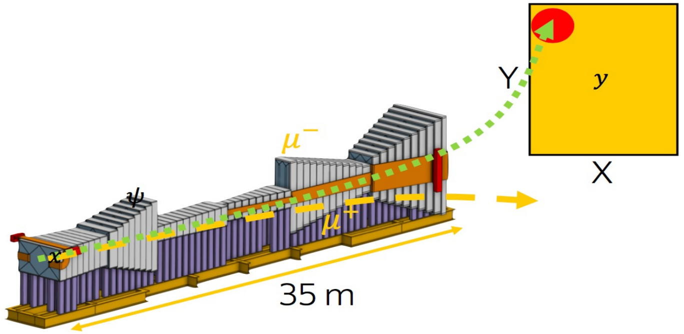


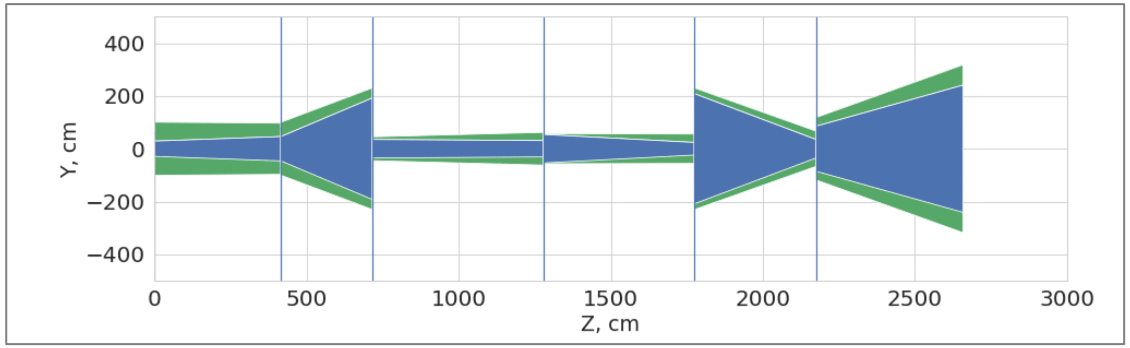
$$p(T(x)|\,\theta)$$

Aleatoric Uncertainty in HEP with Generative Models

Optimizing detector design with Generative Model base Surrogate Simulator





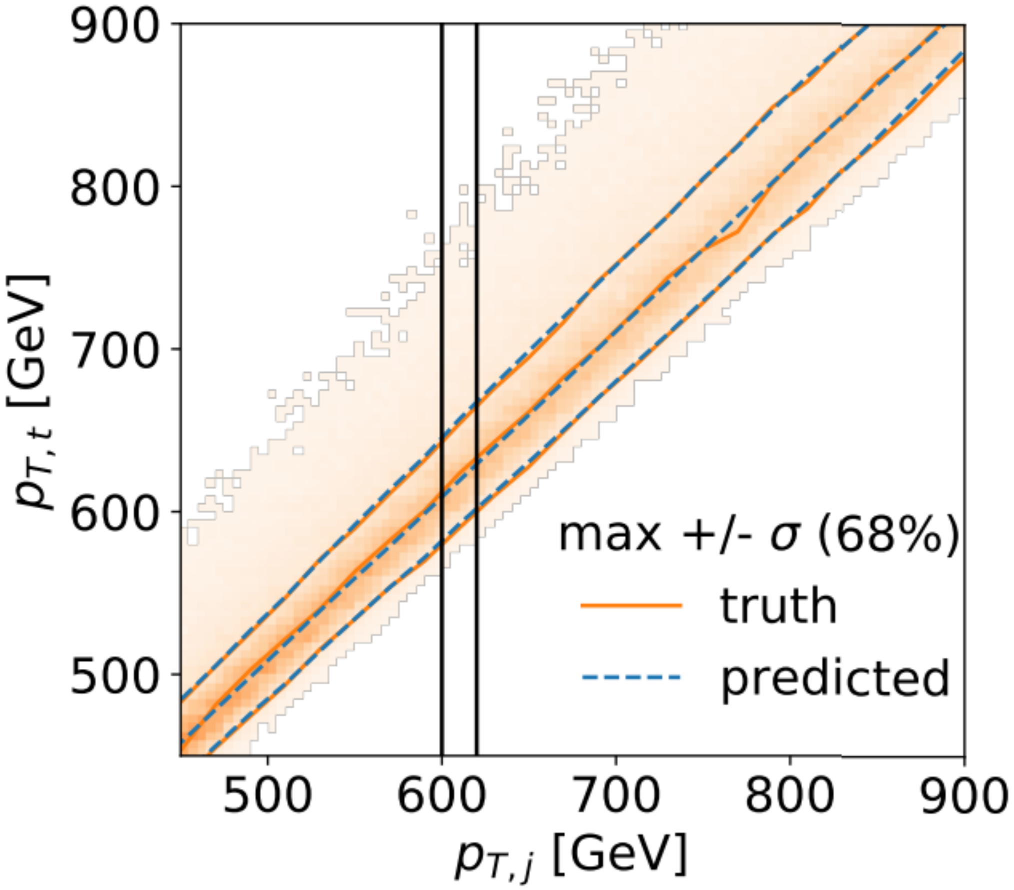


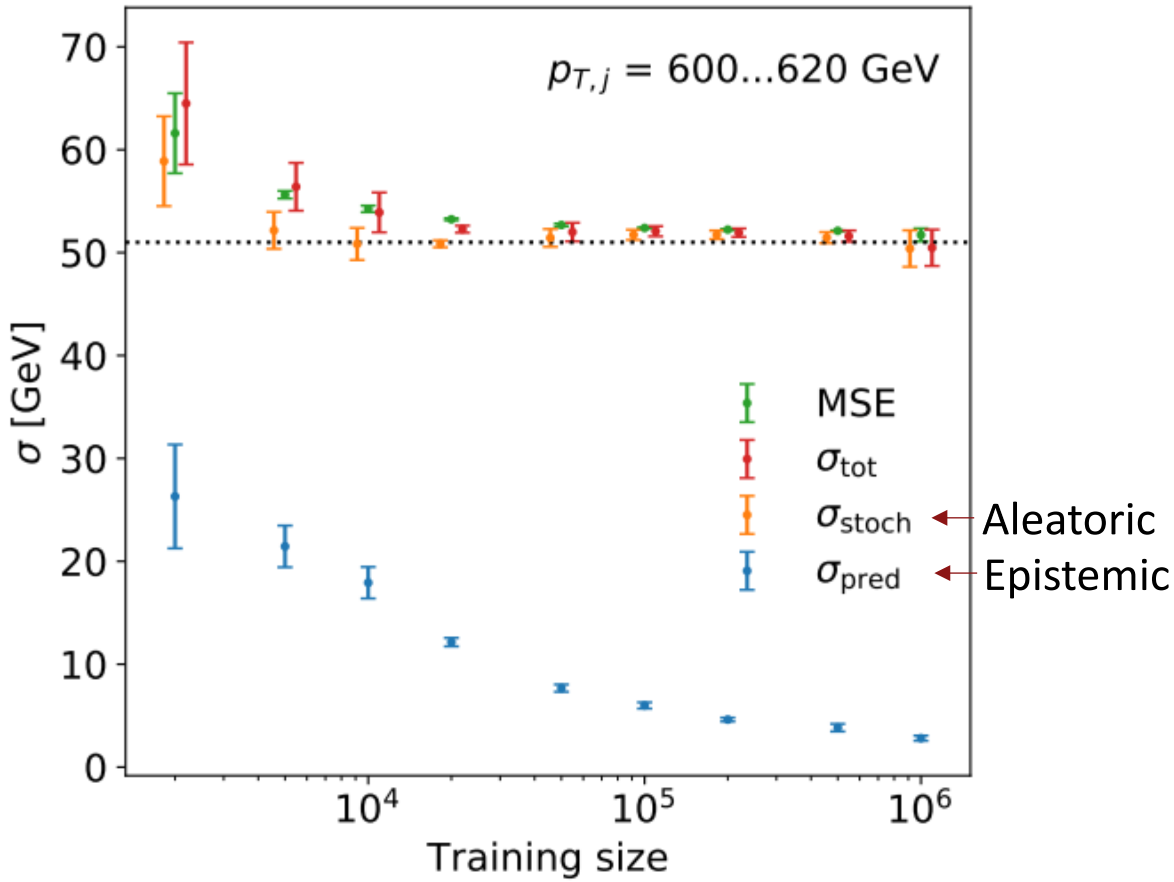
Example: SHiP Magnet Optimization

Reduced length and weight over previous design!

Shirobokov, Belavin, MK, Ustyuzhanin, Baydin, [2002.04632](2002.04632)

Bayesian Neural Networks for Jet Energy Estimation



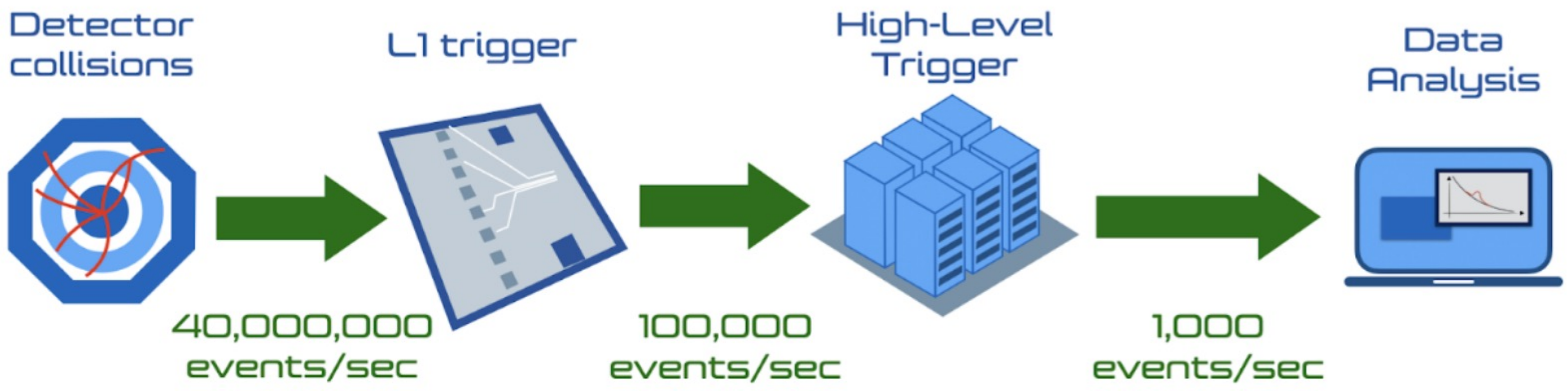


Gaussian Variational Posterior over weights

Gaussian Density Network for p_T predictions

Kasieczka, Luchman, Otterpohl, Plehn, 2003.11099

Uncertainties for ML in Trigger Systems



Decision Theory / Risk Management Problems

- Decisions are irrevocable and constrained by total rate

How certain we are about an ML prediction could change our decision!

Consideration for ML model uncertainties is important here

Image Credit: [2107.02157]

What if the generative model doesn't perfectly fit data?

Potentially bad description of data! → Case for Epistemic / Model Uncertainty

"Bayesian Normalizing Flow" with Variational Inference

$$\mathcal{L} = \sum_{n=1}^{N} \langle \log p_X(x_n|\theta) \rangle_{\theta \sim q_\phi(\theta)} - \mathrm{KL}\big(q_\phi(\theta), p(\theta)\big)$$

Likelihood: Normalizing Flow

Weights sampled from posterior

Variational Posterior: Gaussian

Simulation Based Inference (SBI)

Start with

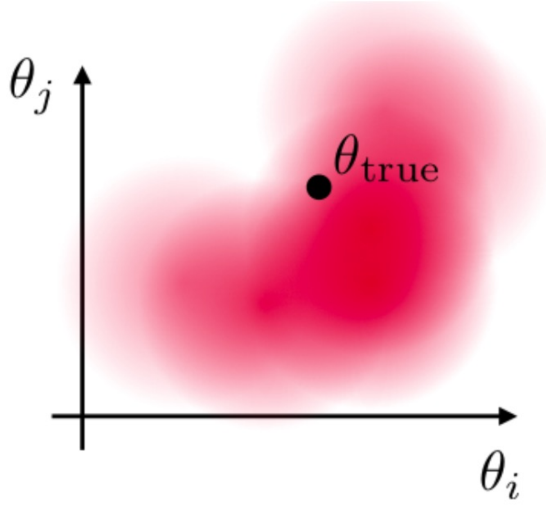
- a simulator that can generate N samples $x_i \sim p(x_i|\theta_i)$,
- a prior model $p(\theta)$,
- observed data $x_{\mathrm{obs}} \sim p(x_{\mathrm{obs}}|\theta_{\mathrm{true}})$.

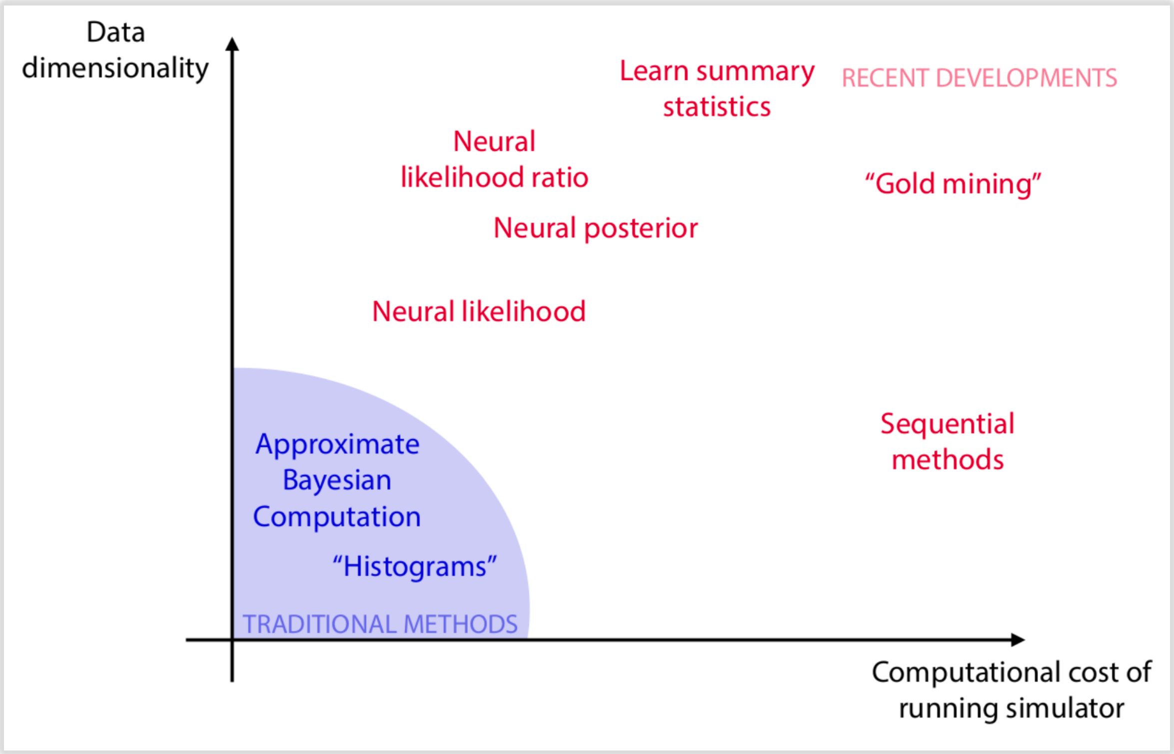
Then, estimate the posterior

$$p(\theta|x_{\mathrm{obs}}) = \frac{p(x_{\mathrm{obs}}|\theta)p(\theta)}{p(x_{\mathrm{obs}})}$$

Or a likelihood ratio

$$r(\theta) = \frac{p(x_{obs}|\theta)}{p(x_{obs}|\theta_0)}$$





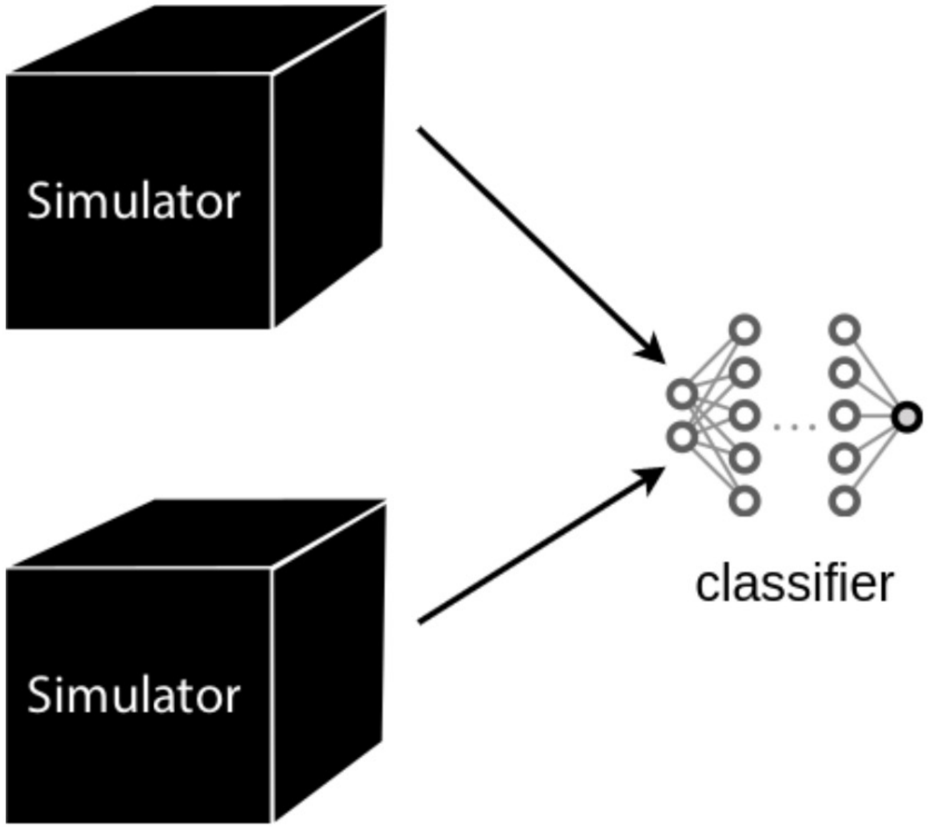
Slide Credit: G. Louppe

Neural Ratio Estimation

The likelihood-to-evidence $r(x|\theta) = \frac{p(x|\theta)}{p(x)} = \frac{p(x,\theta)}{p(x)p(\theta)}$ ratio can be learned, even if neither the likelihood nor the evidence can be evaluated:

$x, \theta \sim p(x, \theta)$

$x, \theta \sim p(x)p(\theta)$



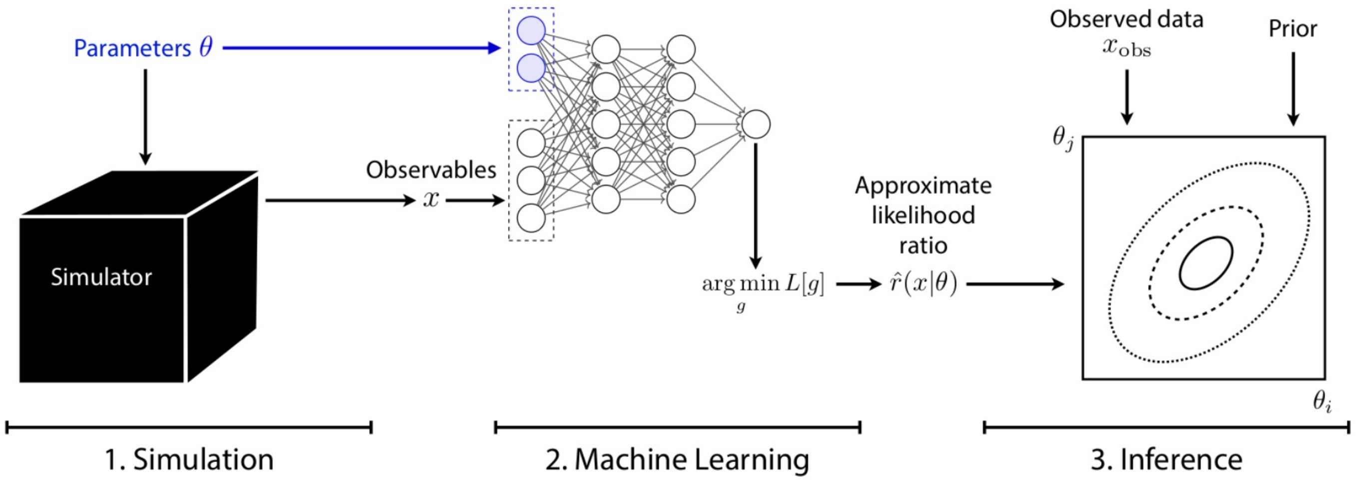
$\hat{r}(x|\theta)$

Slide Credit: G. Louppe

Neural Ratio Estimation

The likelihood-to-evidence $r(x|\theta) = \frac{p(x|\theta)}{p(x)} = \frac{p(x,\theta)}{p(x)p(\theta)}$ ratio can be learned, even if neither the likelihood nor the evidence can be evaluated:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \approx \hat{r}(x|\theta)p(\theta)$$



Run simulator and save data

Train NN classifier, interpret as likelihood ratio estimator

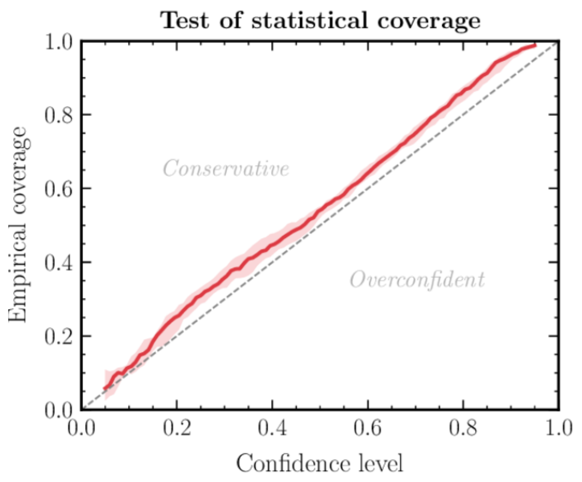
Amortized: cheap to repeat for new data

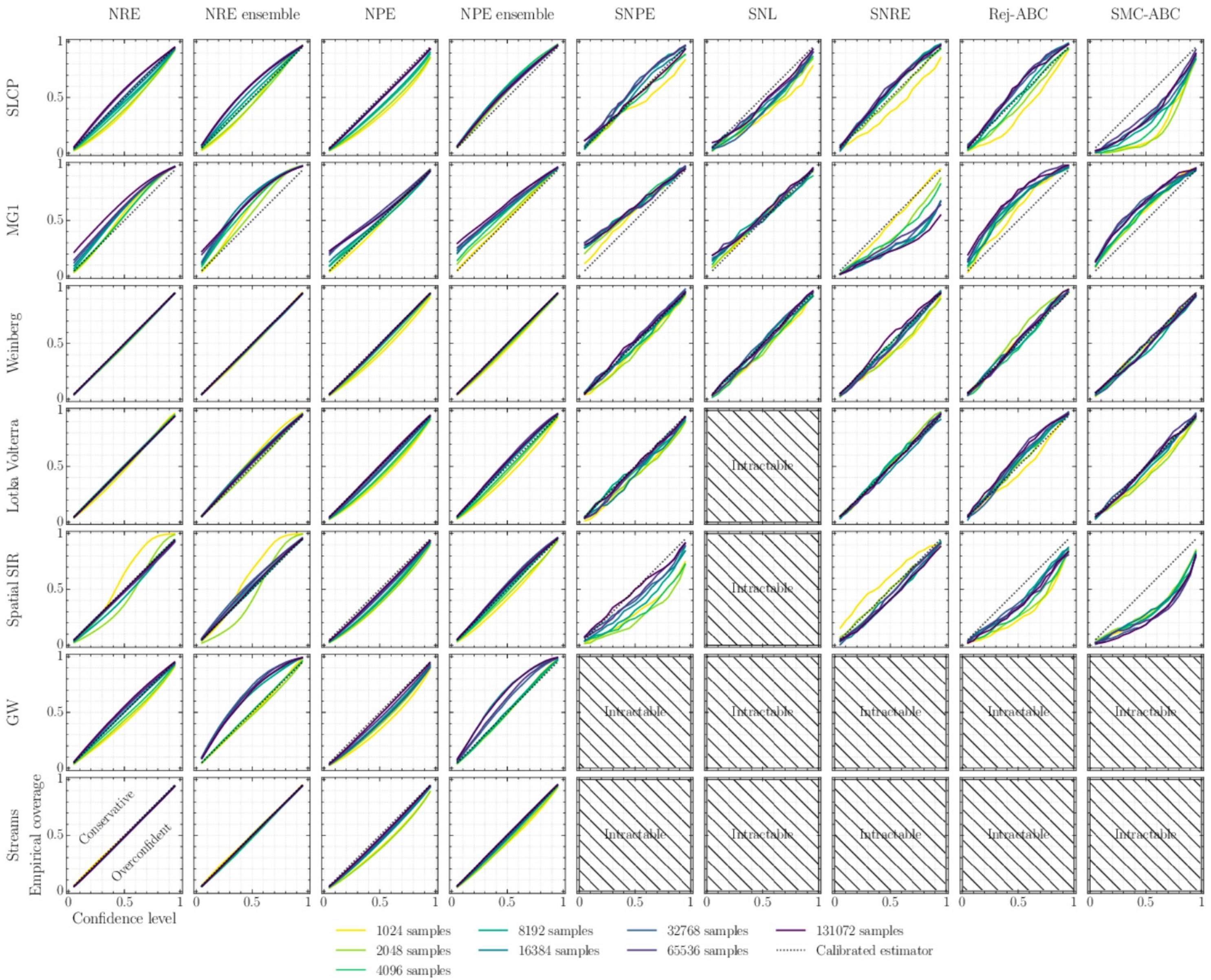
Slide Credit: G. Louppe

But proceed with caution! … model checking, evaluation, and criticism

Coverage diagnostic:

- For $x, \theta \sim p(x, \theta)$, compute the $1-\alpha$ credible interval based on $\hat{p}(\theta|x)$.
- If the fraction of samples for which θ is contained within the interval is larger than the nominal coverage probability $1-\alpha$, then the approximate posterior $\hat{p}(\theta|x)$ has coverage.



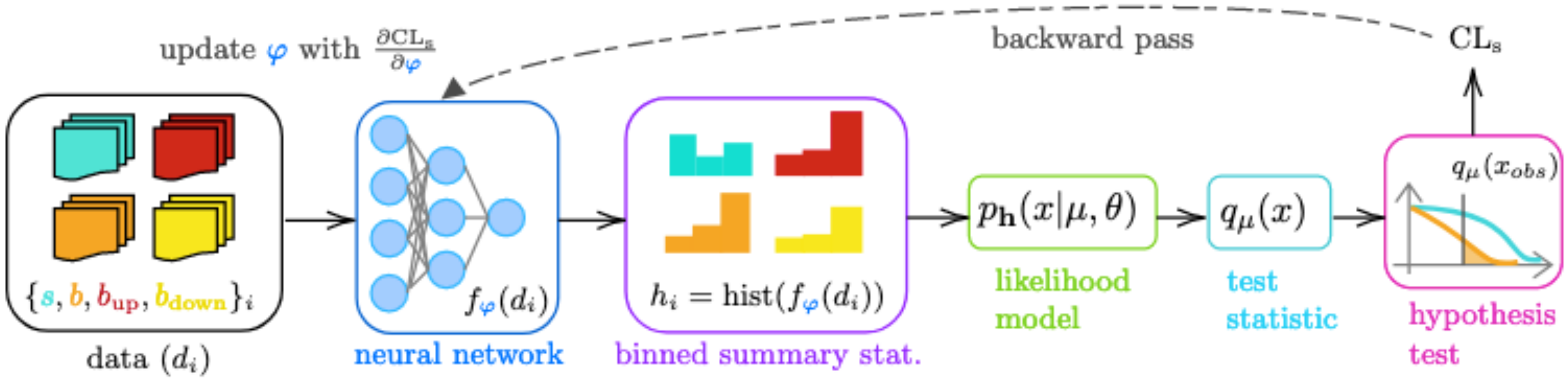


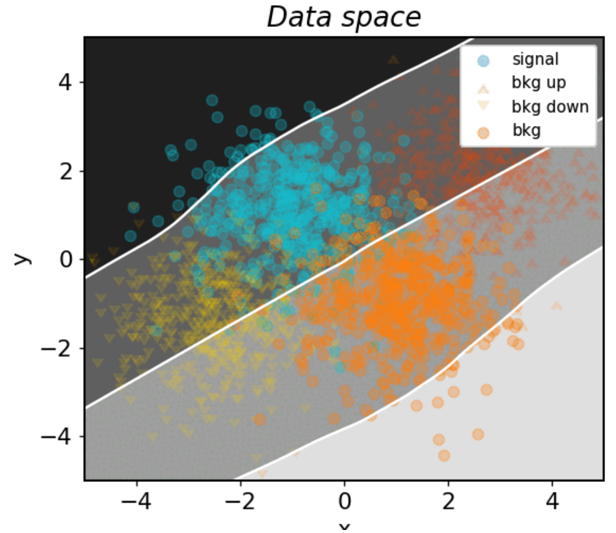
Slide Credit: G. Louppe

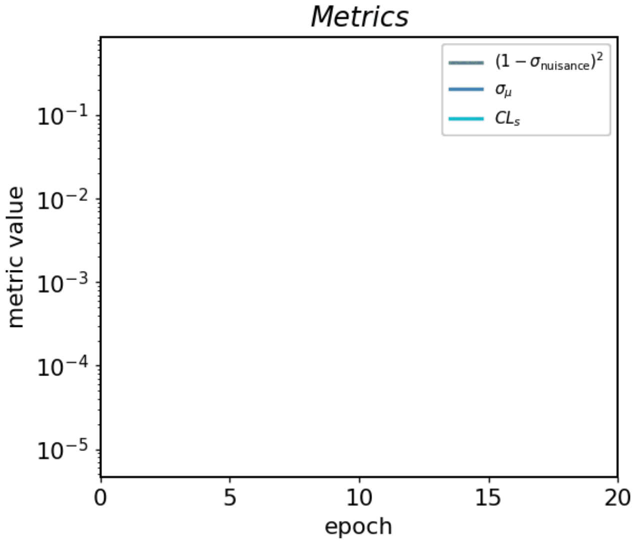
Can Inference Goal Drive Training?

Train summary statistic $T_w(x)$ to optimize inference goal

Examples: NEOS and INFERNO







Simpson, Heinrich, 2203.05570