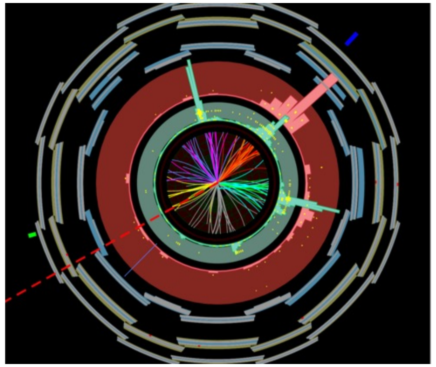
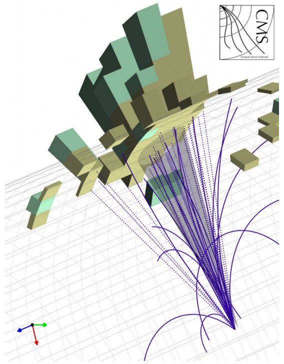
Uncertainties in Machine Learning: When are they needed in HEP?

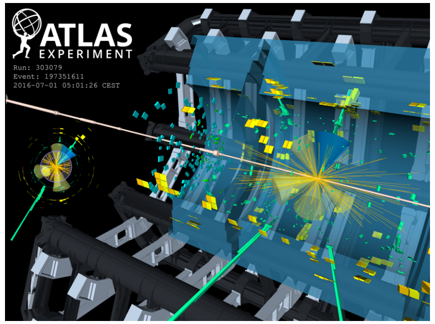
Michael Kagan
SLAC

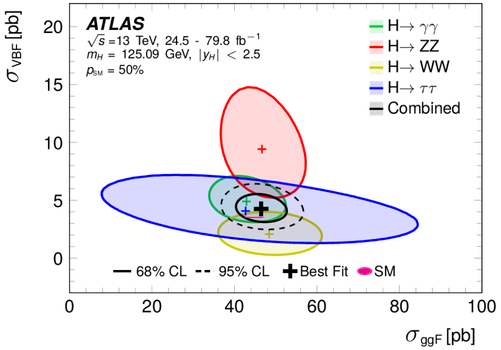
Systematic Effects and Nuisance Parameters in Particle Physics Data Analyses, Banff
April 27, 2023

Machine Learning Is Used Across HEP Data Analysis







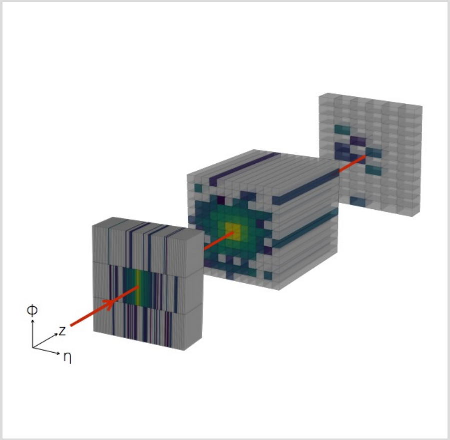


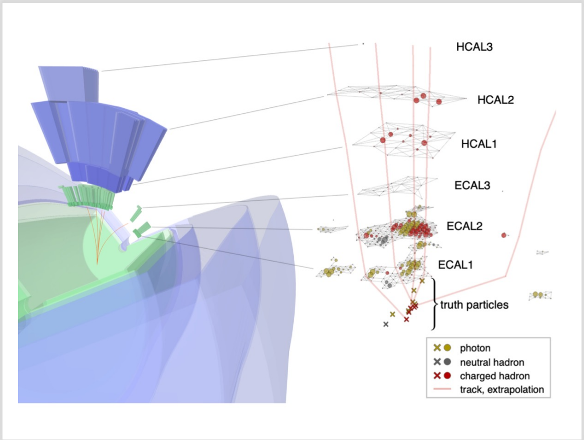
Data Acquisition, Simulation

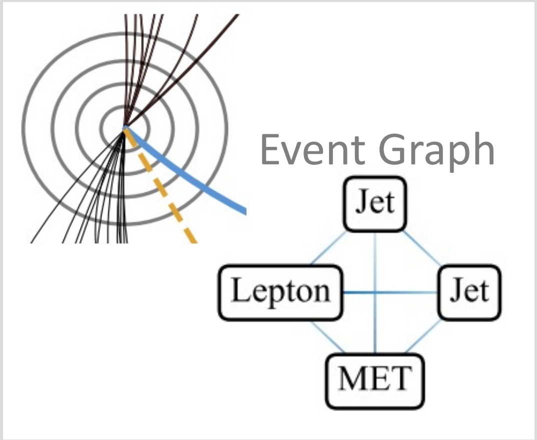
Particle Reconstruction

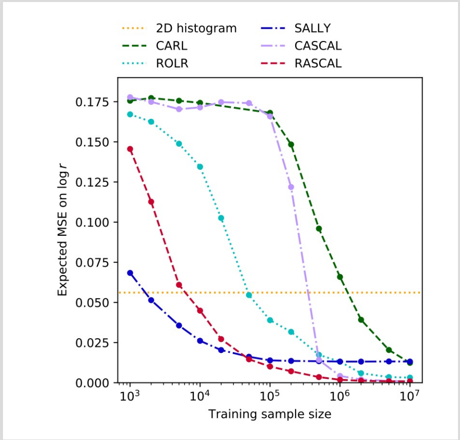
Event Selection & Reconstruction, Background estimation

Parameter Estimation









Questions of Concern

Is there uncertainty from using the ML Model?

What kinds of uncertainty?

What if the ML model did not “perfectly” fit the data?

When does it matter?

This Talk

How to deal with **HEP *Systematic Uncertainties*** when using ML Models?

Tommaso’s Talk

Also nice discussion in: [PDG review of ML in HEP]

Supervised Learning Setup

Training Data:

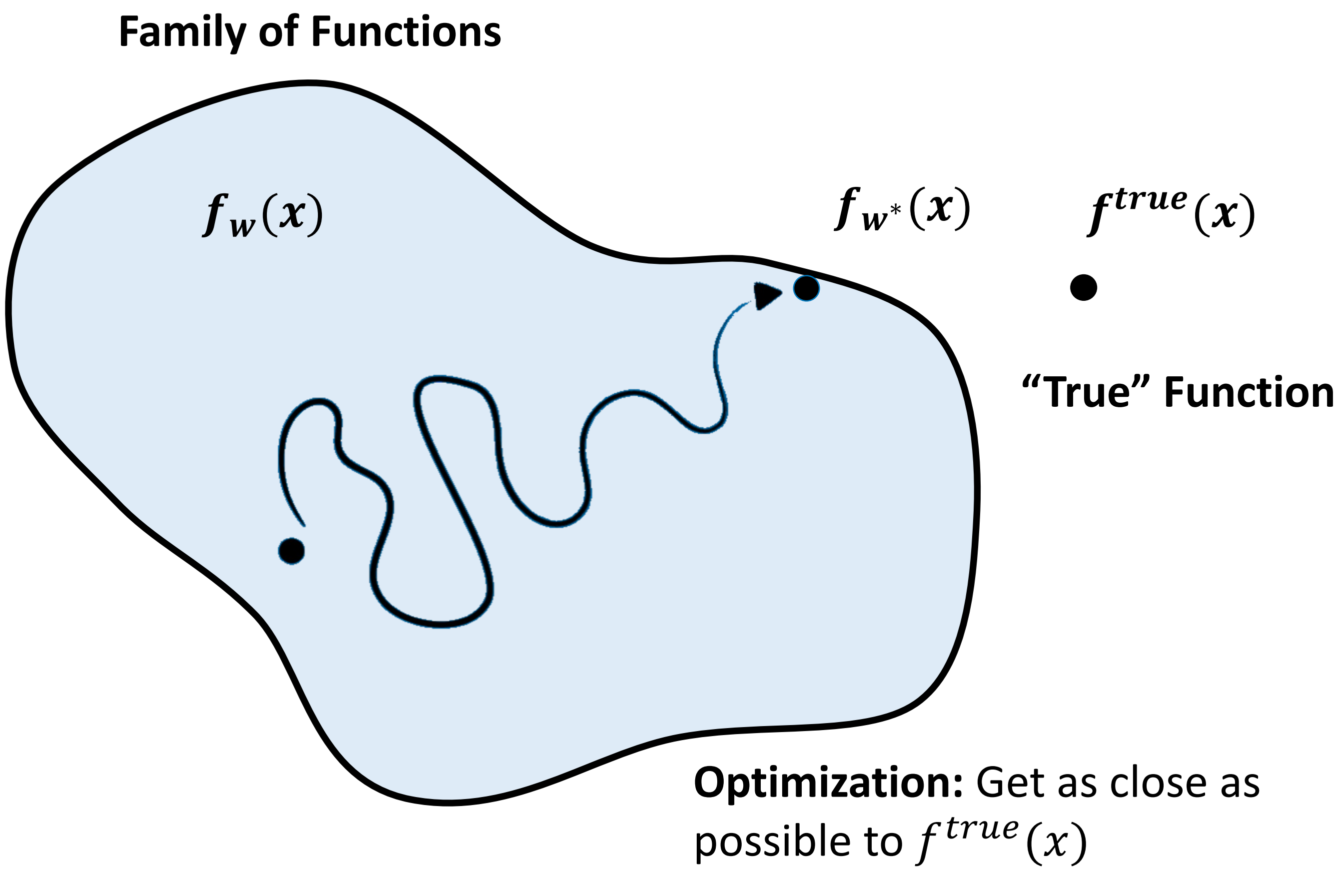
- $\mathcal{D} = \{x_i, y_i\}$ = features and target
- $x, y \sim p(x, y)$

Goal:

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

Learning:

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



Types of Uncertainties in ML

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

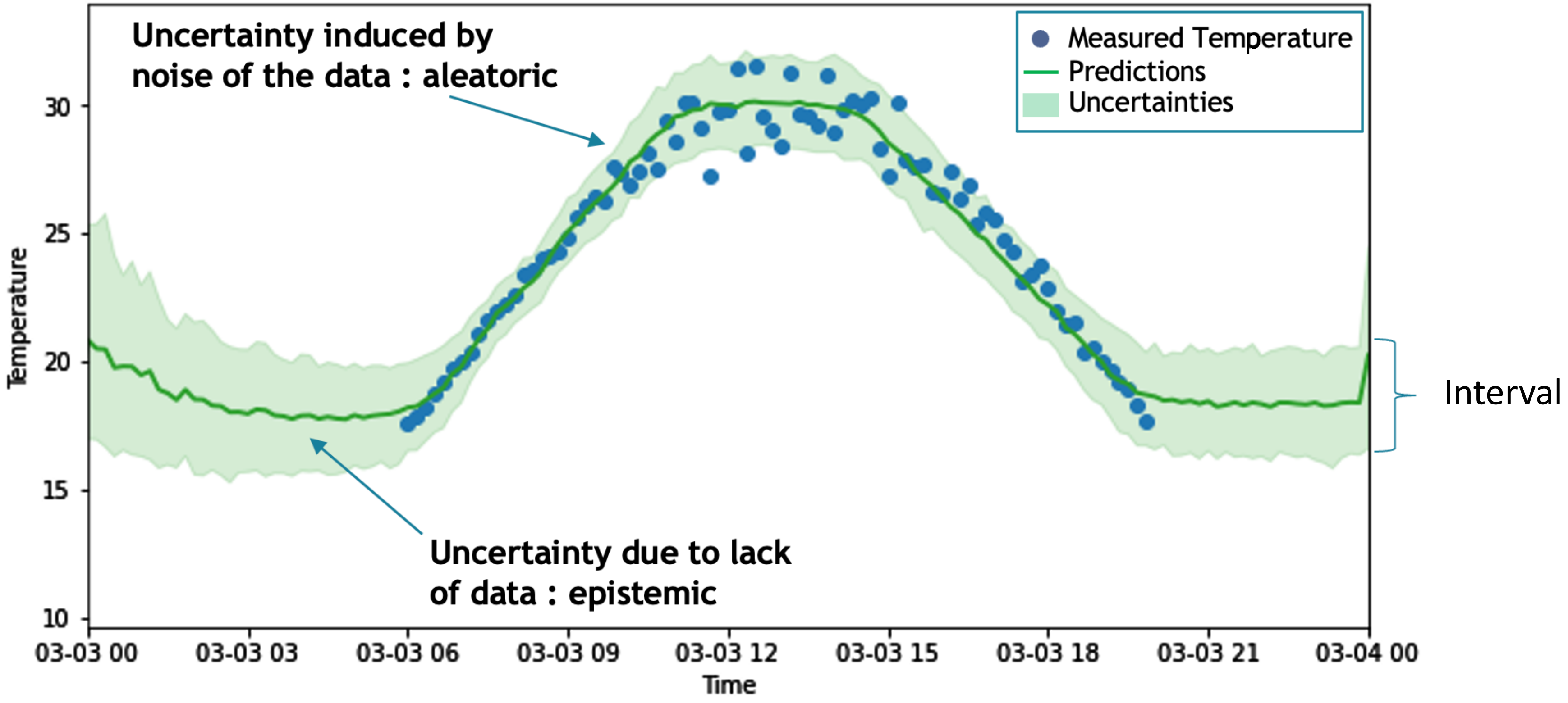
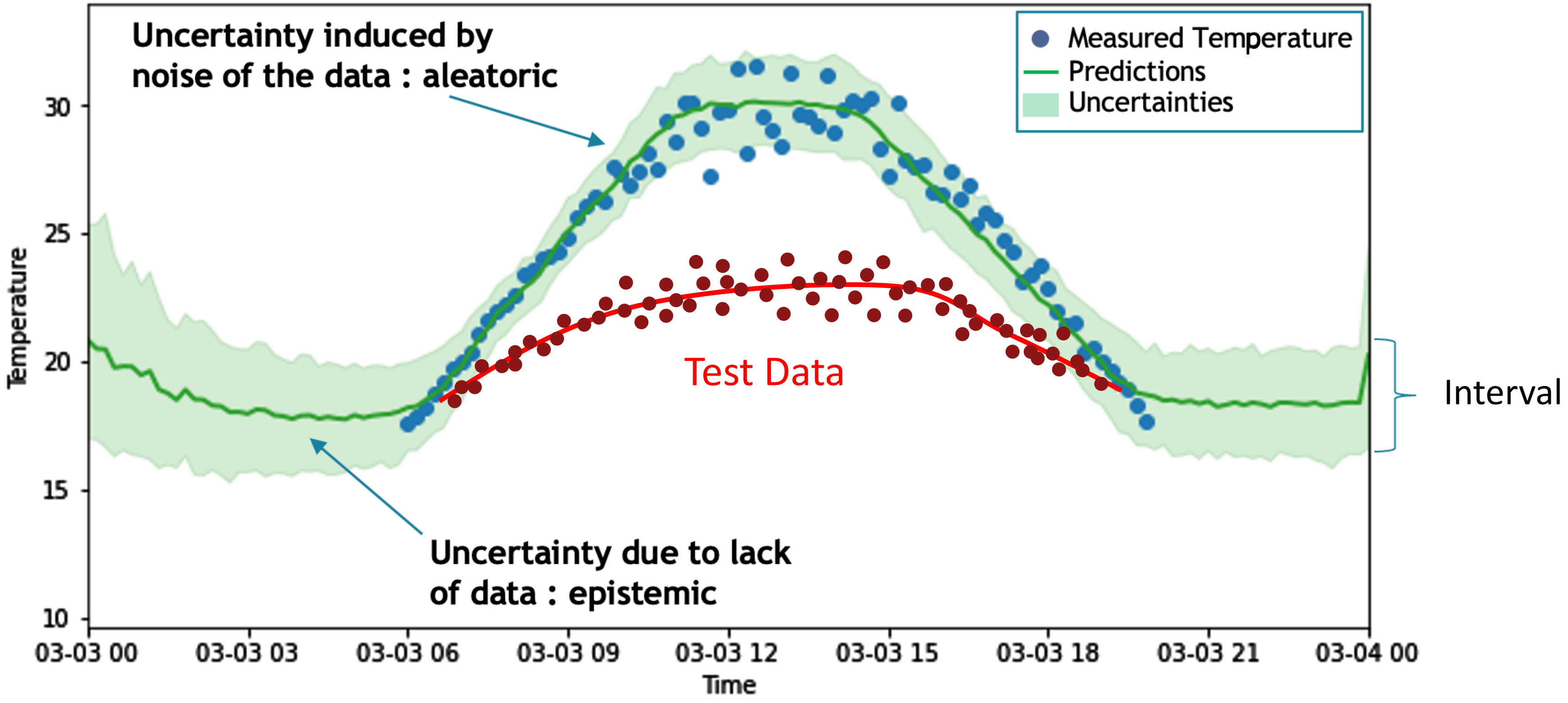


Image Credit: N. Brunel

Types of Uncertainties in ML

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

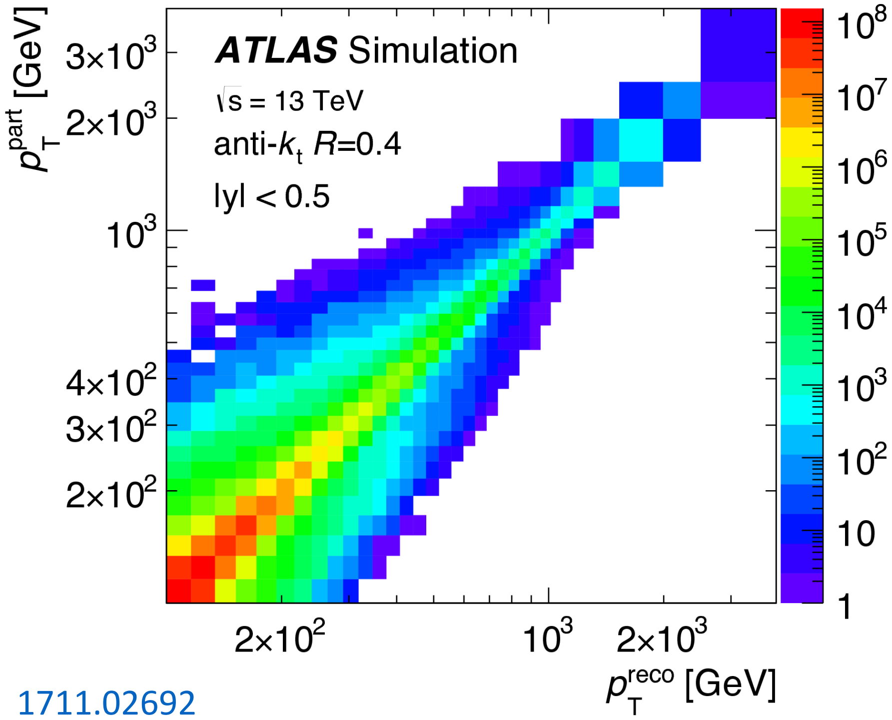
Image Credit: N. Brunel

Aleatoric Uncertainty

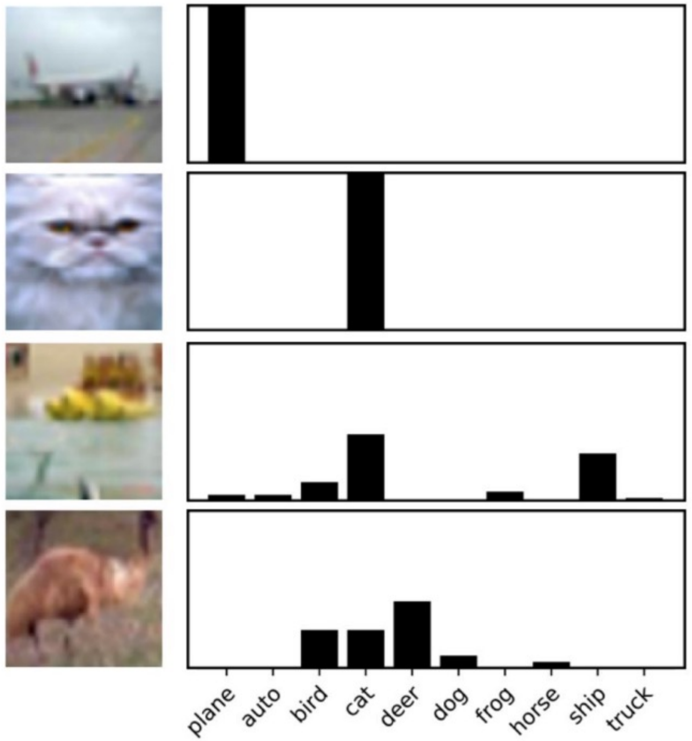
Often called "Statistical Uncertainty" and considered "Irreducible"

Variability in outcome of experiment due to inherently random effects

- Example: $y = f(x) + \epsilon$ where $\epsilon \sim N(0, \sigma) \rightarrow y \sim N(f(x), \sigma)$



1711.02692



Battleday et al. 2019

Epistemic Uncertainty

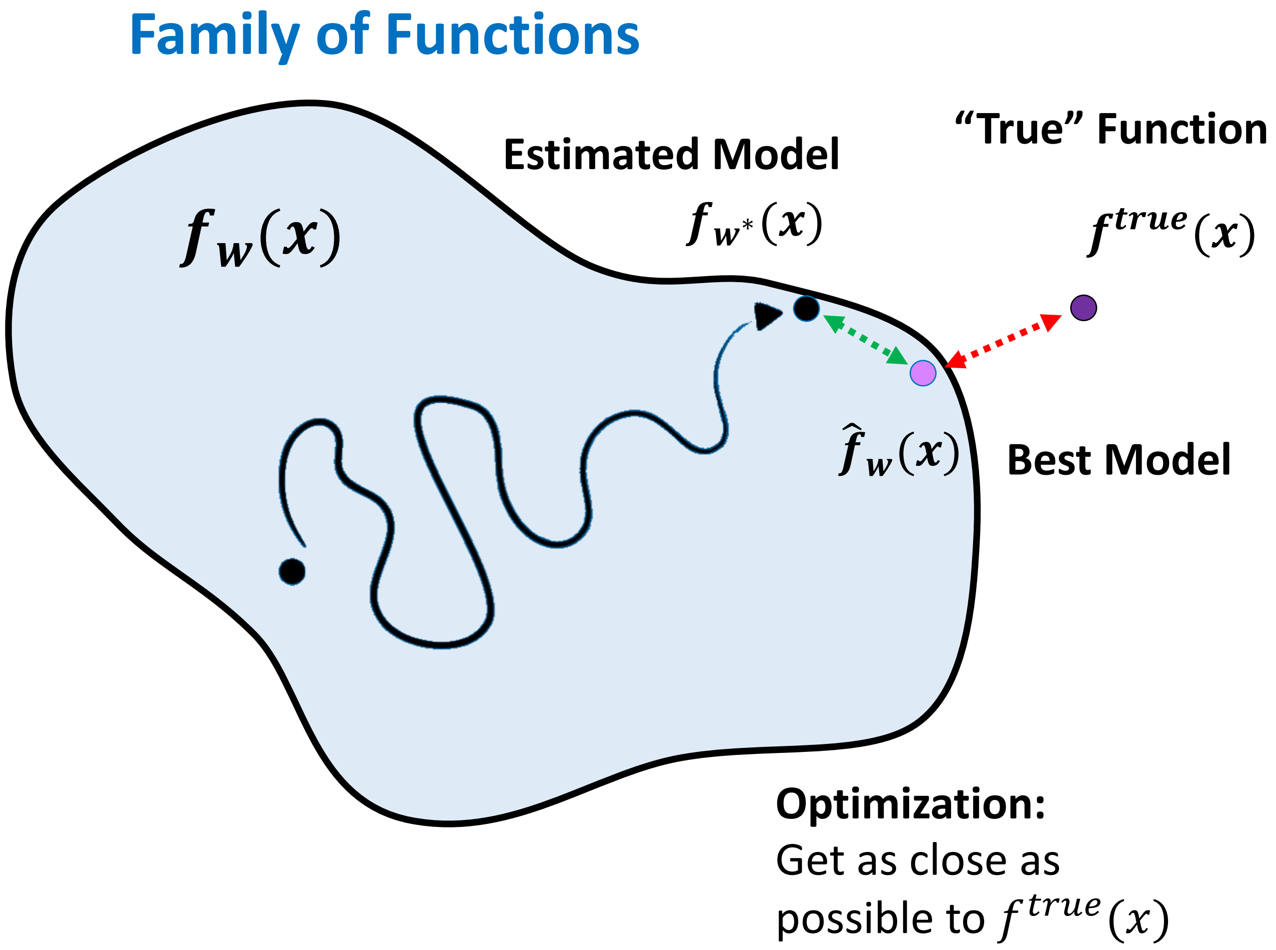
Lack of knowledge about best model

Main origins in ML

- *Estimation error:*
  Training data just a sample of possible observations

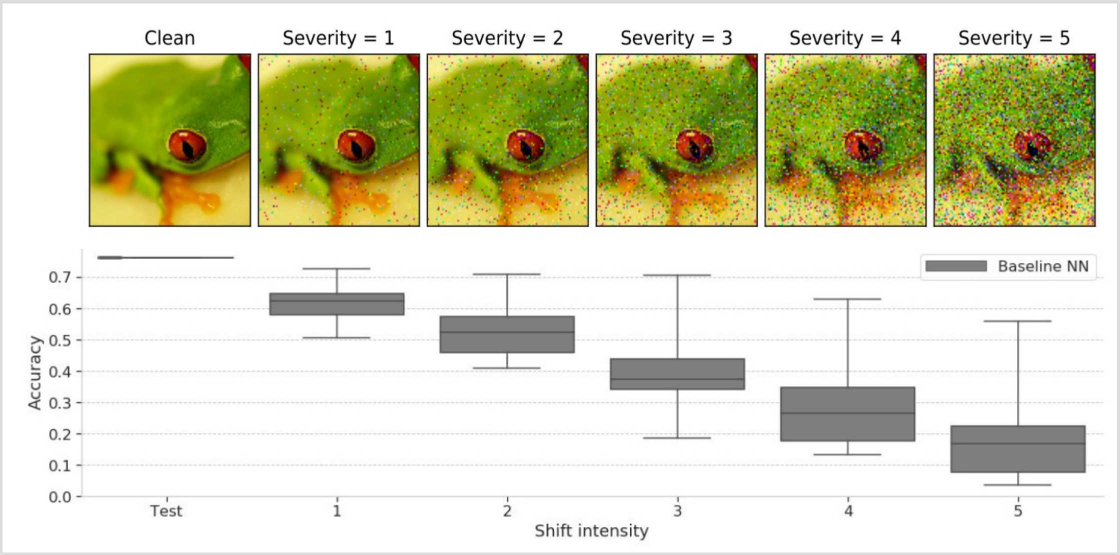
- *Approximation error:*
  no model (in model class) can capture unknown true model

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



Domain / Distribution / Dataset Shift

Image: [1903.12261](1903.12261)



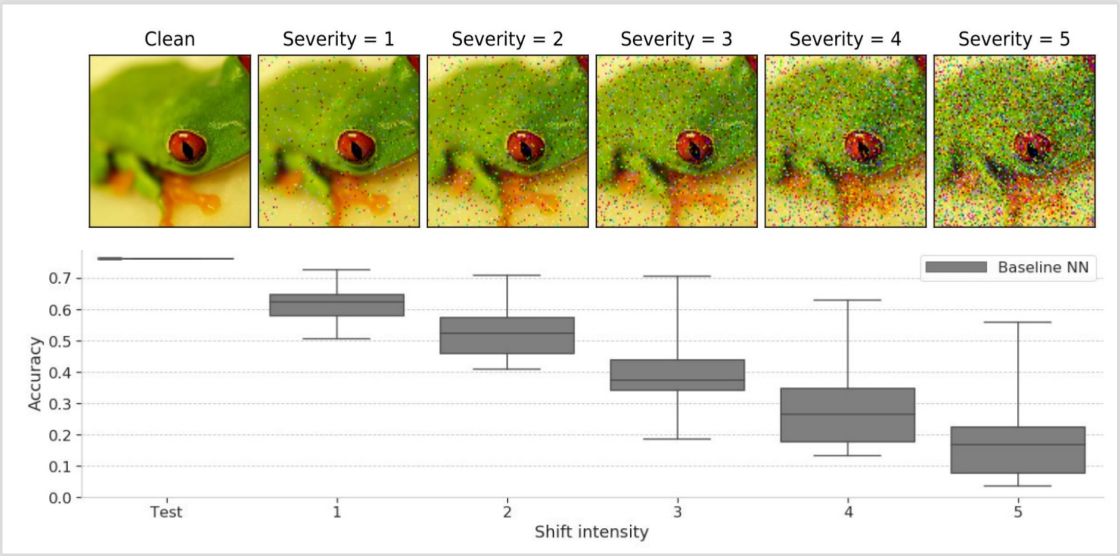
$$p_{TEST}(x, y) \neq p_{TRAIN}(x, y)$$

Training distribution different from distribution model is applied to

Domain / Distribution / Dataset Shift

Image: 1903.12261

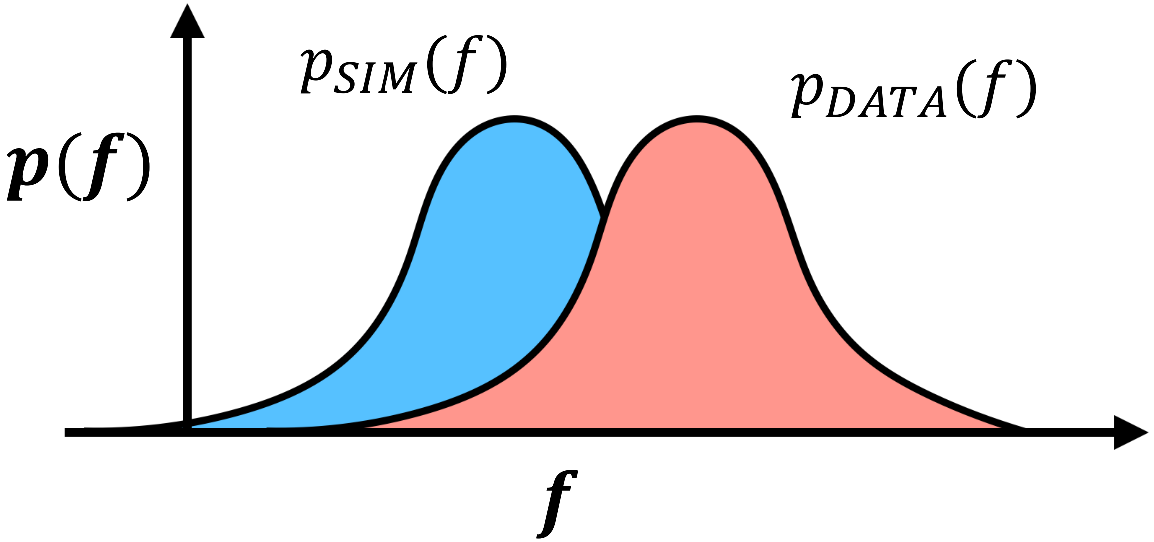
$$p_{TEST}(x,y) \neq p_{TRAIN}(x,y)$$



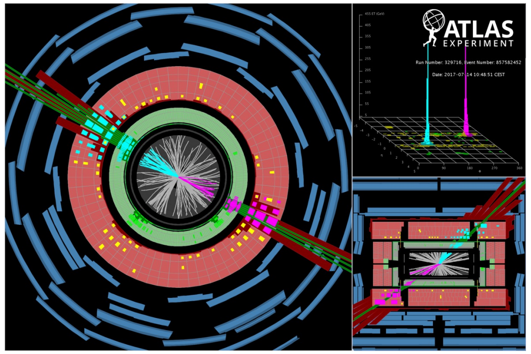
Training distribution different from distribution model is applied to

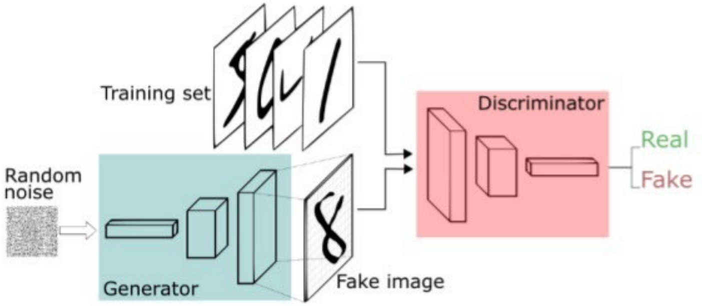
Similar to Systematic Uncertainties

- Simulation used to train model
- But Simulation not perfect model of data
- ML model trained on simulation may act differently in data



What Kind of Model Are We Talking About?





Physics Model in Simulator

Model data generation process using Physics Knowledge

Epistemic Uncertainty:

- Lack of knowledge of data generation process
- Leads to data/simulation mismatch
- Systematic Uncertainties

Machine Learning Model

Fit to data, relatively little inductive bias in model design and optimization

Epistemic Uncertainty:

- Often assumes training data = test data
- Lack of knowledge about which are the best parameters of model after training

Also see PDG review of ML in HEP

When are ML Model Uncertainties Needed?

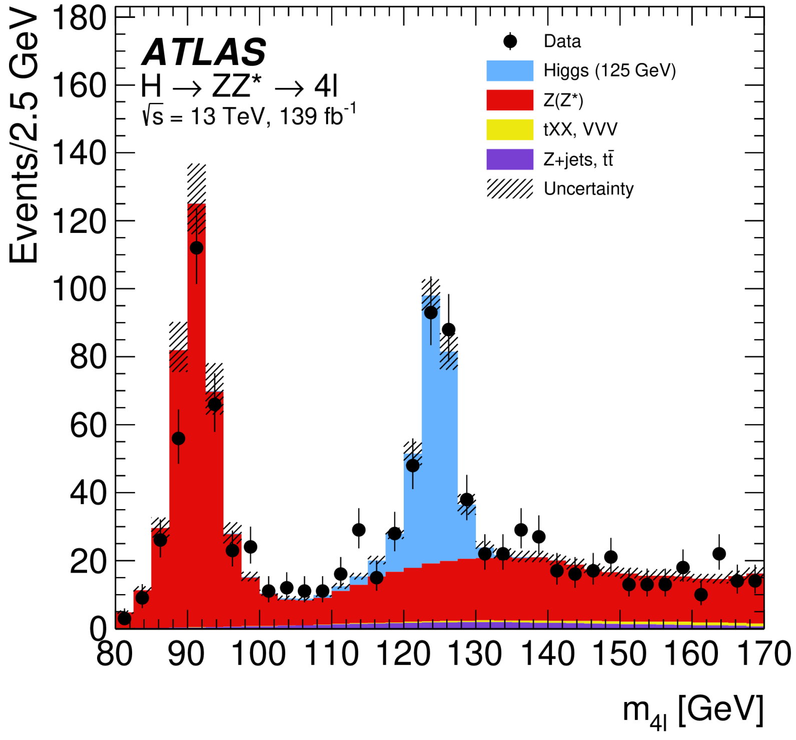
Reconstruction, data selection, event classification enable us to define powerful **summary statistics**

$$T(x):\ \mathbb{R}^{10^8} \to \mathbb{R}$$

Estimate likelihood for frequentist parameter inference:

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

- θ = physics parameters of interest
- $\lambda(\cdot)$ = parameters of probability density
  e.g. mean of Poisson / Gaussian density



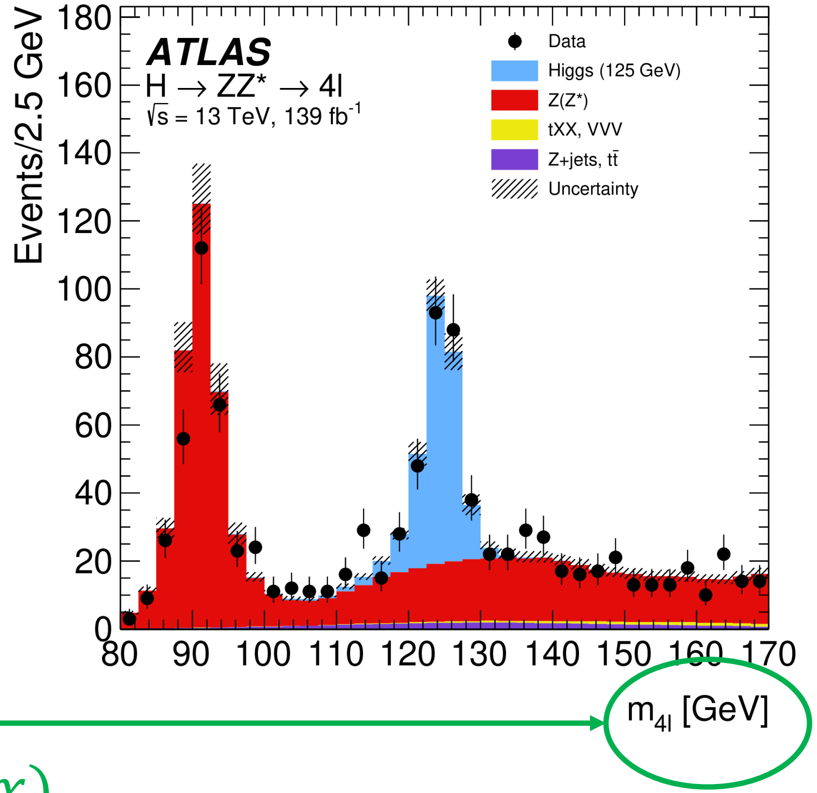
EPJC 80 (2020) 942

† Ignoring Systematics for the moment

When are ML Model Uncertainties Needed?

Question of optimality:

- Did ML get best reconstruction or event selection?
- Effects definition of discriminating variables, but doesn't affect compatibility with data



Things that affect $T(x)$

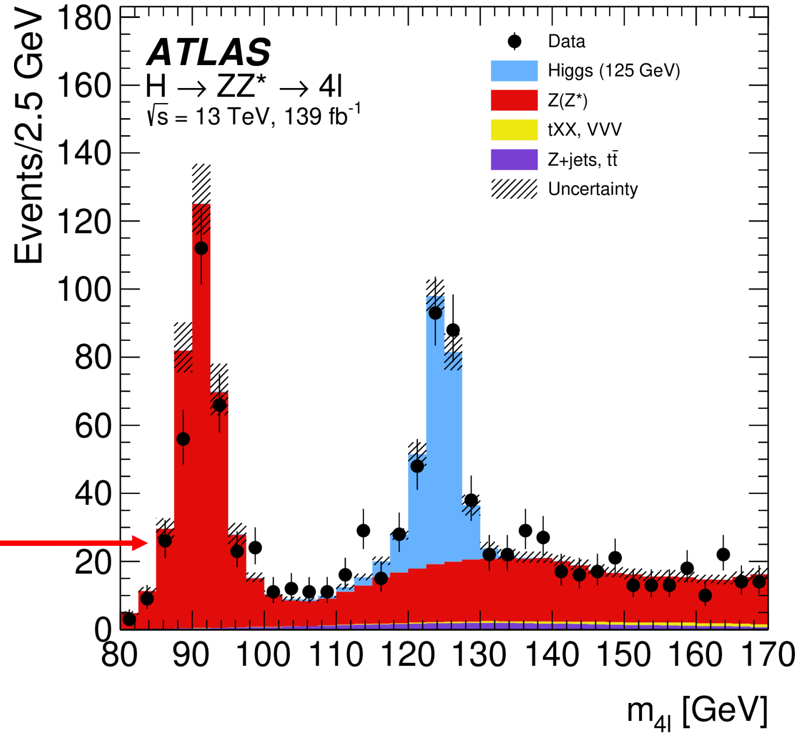
EPJC 80 (2020) 942

† Ignoring Systematics for the moment

When are ML Model Uncertainties Needed?

Question of optimality:

- Did ML get best reconstruction or event selection?
- Effects definition of discriminating variables, but doesn't affect compatibility with data



Things that affect $p(\cdot|\,\lambda(\theta))$

EPJC 80 (2020) 942

Questions of correctness:

- Did ML learn an accurate fast simulation?
- Did ML learn a good background estimate?
- Effects statistical model & compatibility with data!

† Ignoring Systematics for the moment

ML Model Uncertainty can look like Systematic Uncertainties

Example:

Simulation not a perfect model of data

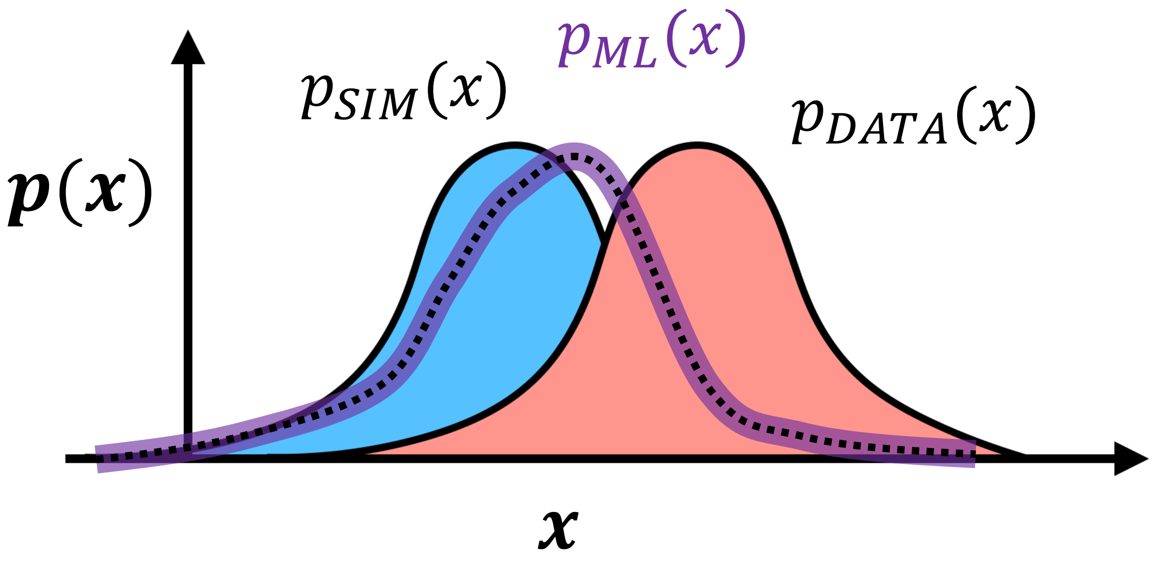
- Calibration procedure using control data

Fit ML model as simulator surrogate

ML Surrogate not a perfect model of simulator or data

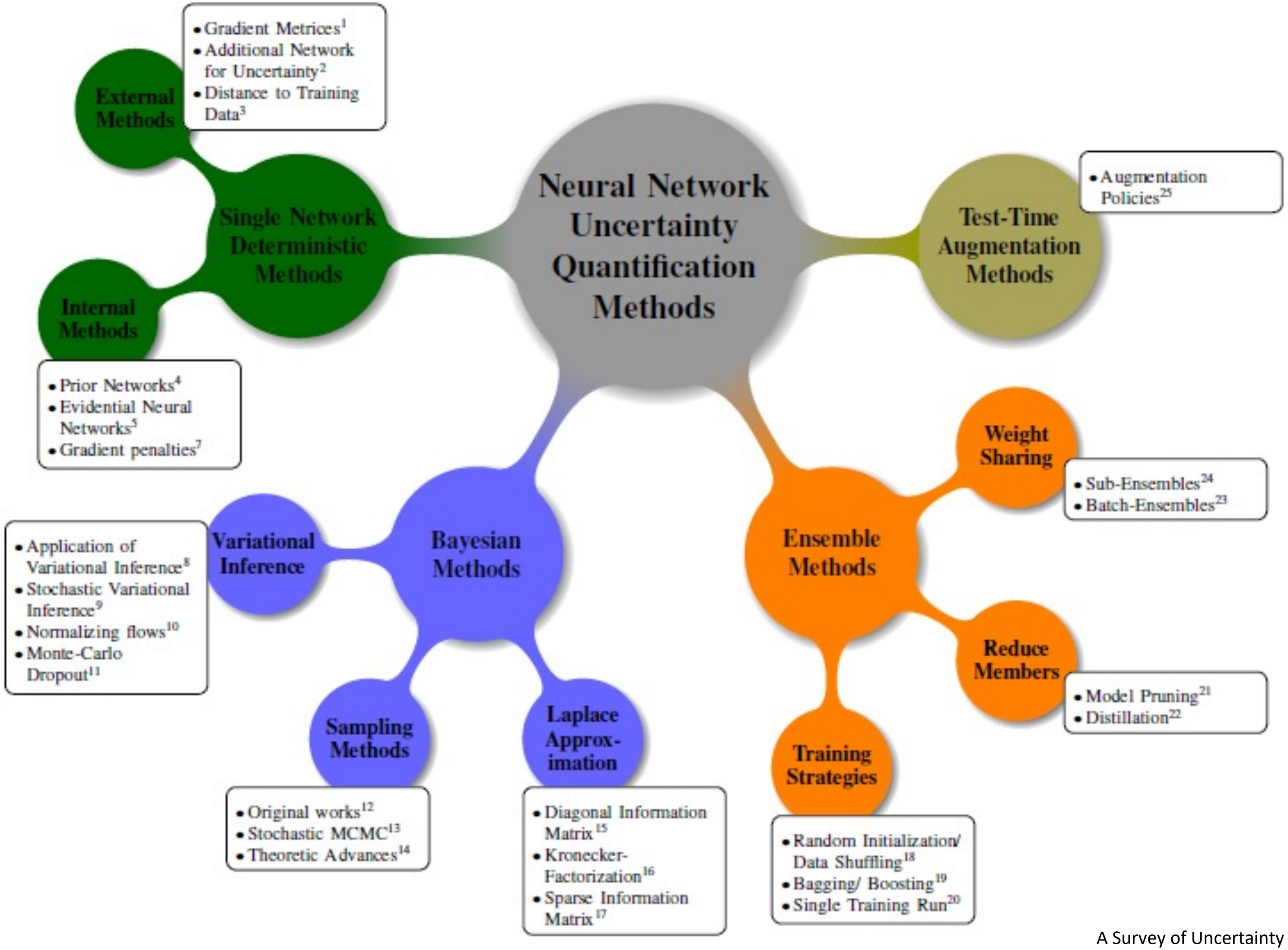
What to do:

- Estimate epistemic / model uncertainty?
- Calibrate surrogate with control data & estimate systematic uncertainty?



Uncertainty Estimation Approaches in Deep Learning

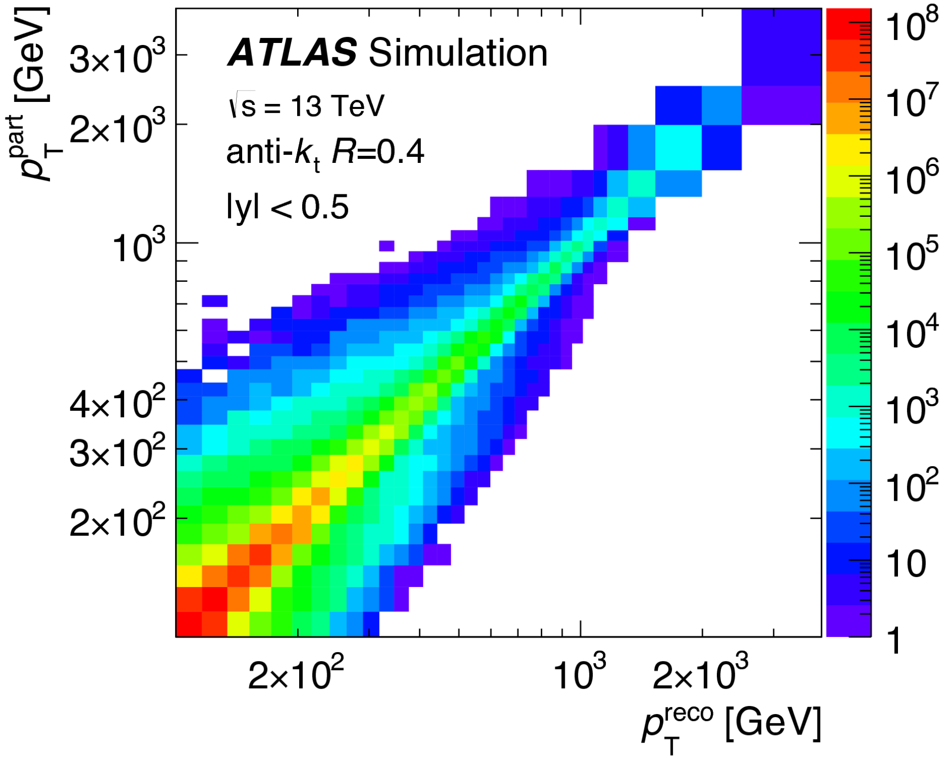
Uncertainty Estimation Approaches in Deep Learning



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

Modeling Aleatoric Uncertainty

Intrinsic randomness in data → Typically described by probability distributions



Modeling Aleatoric Uncertainty

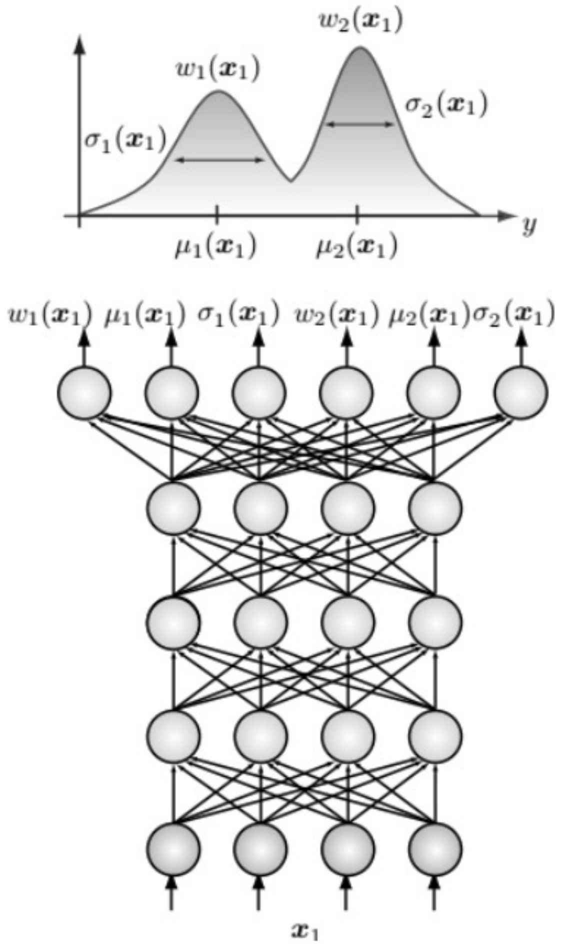
Intrinsic randomness in data → Typically described by probability distributions

Density Networks

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

Train neural network to predict per-example parameters

$$f(x) \rightarrow \phi(x)$$



Mixture density network

Image Credit: Bishop, *Pattern Recognition and Machine Learning*

Modeling Aleatoric Uncertainty

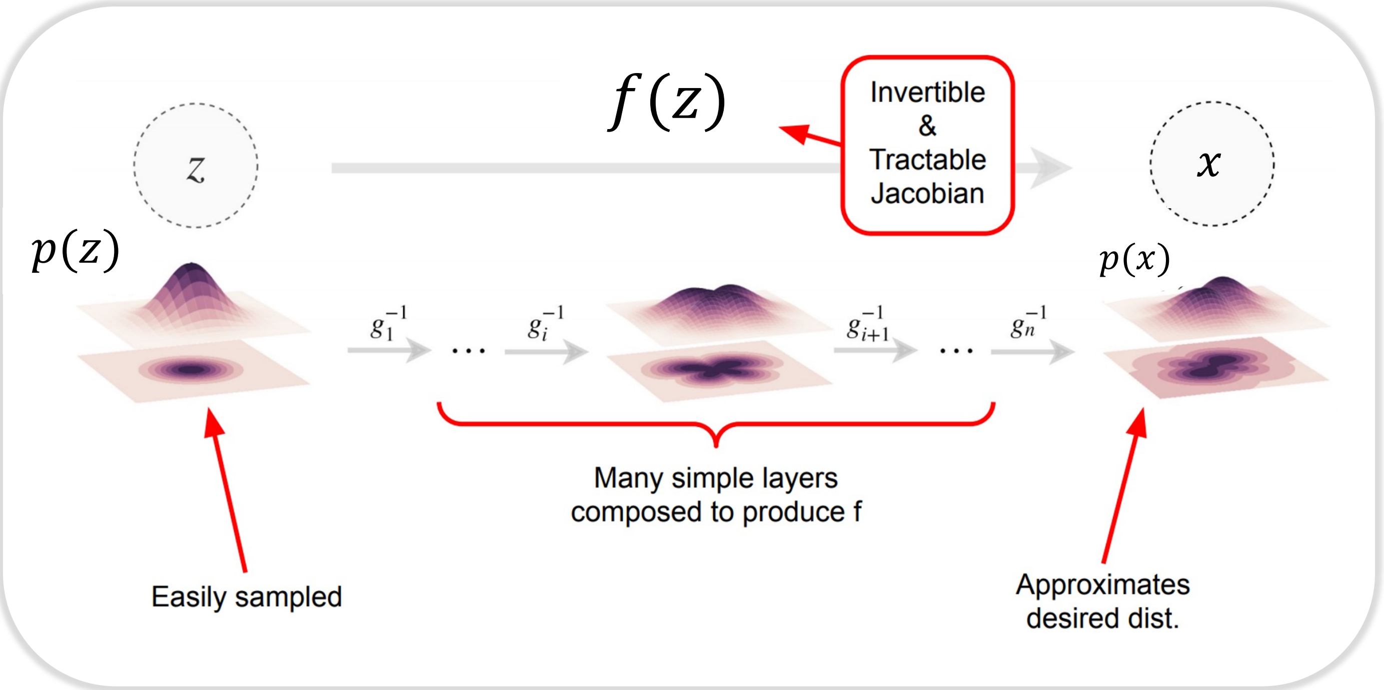
Intrinsic randomness in data → Typically described by probability distributions

Normalizing Flow
Approximate density $p(x)$ by learning to transform noise z into data

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

Normalizing flows use invertible functions with tractable Jacobian s.t.

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



Modeling Aleatoric Uncertainty

Intrinsic randomness in data → Typically described by probability distributions

Normalizing Flow

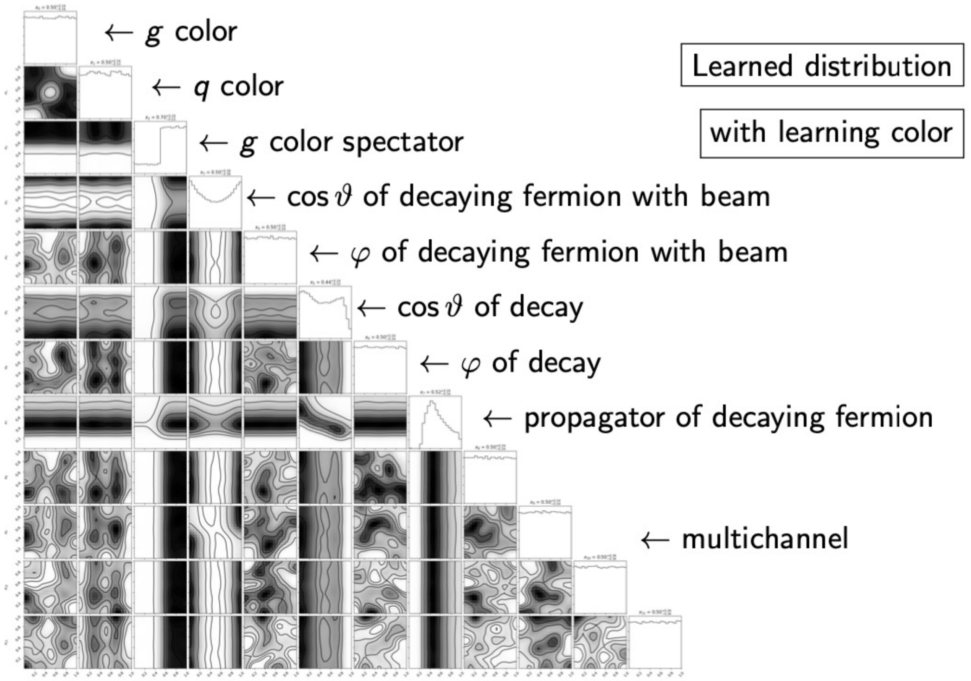
Approximate density $p(x)$ by learning to transform noise z into data

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

Normalizing flows use invertible functions with tractable Jacobian s.t.

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

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



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

Epistemic Uncertainty

Uncertainty from lack of knowledge about best model

- E.g. from only have finite training stats.

Often framed as :

"What networks could I have fit to my data?"

Or

"What are the uncertainties on the network weights that I fit to data?"

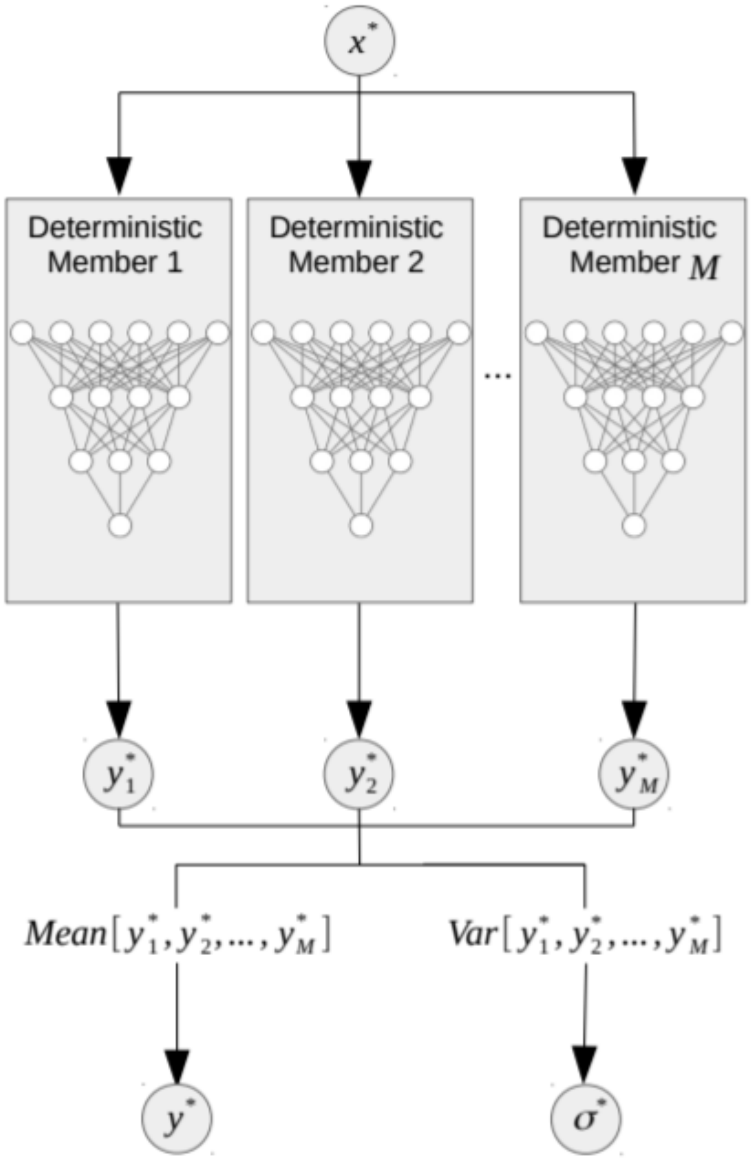
Epistemic Uncertainty with Deep Ensembles

Ensembling:

- Retrain network from multiple initializations

Can be coupled with Bootstrapping

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

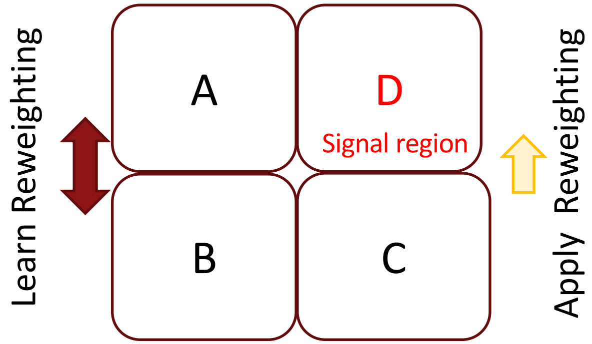


Lakshminarayanan, Pritzel, Blundell, [1612.01474](), [Nixon, Lakshminarayanan, Tran, 2020]()

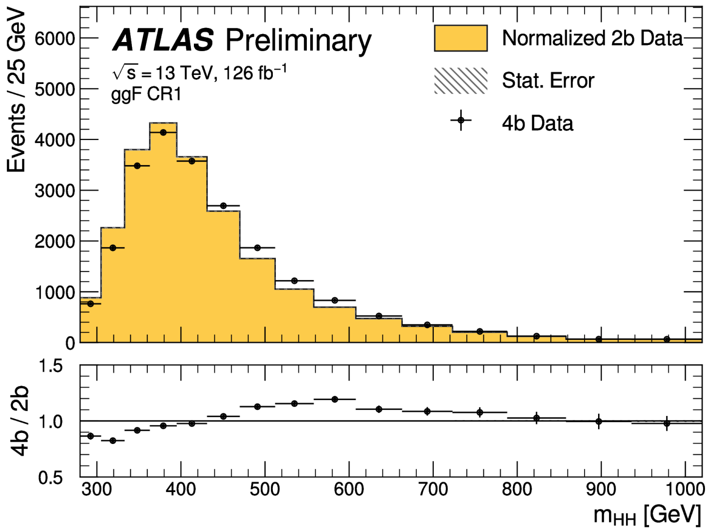
Model Uncertainty in ML-based Background Estimation

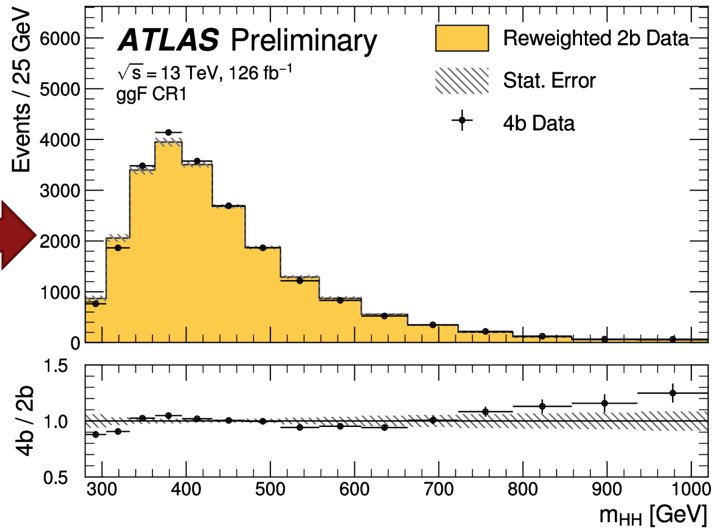
High-Dimensional "ABCD" method with NN's

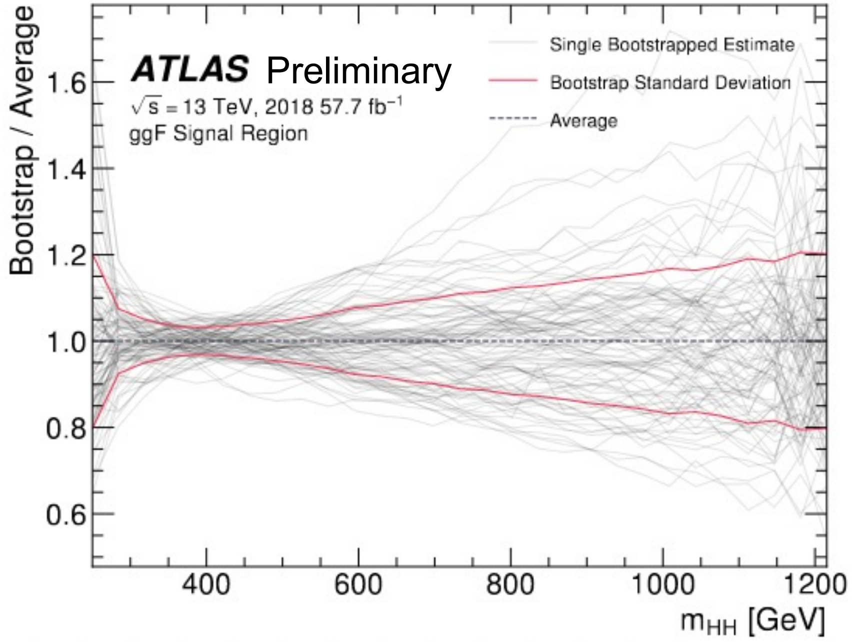
- Learn weighting using classifiers: $w(x) \approx \frac{p_A(x)}{p_B(x)}$
- Estimate background: $\hat{p}_D(x) = w(x)p_C(x)$



ATLAS $hh \to 4b$: Uncertainty from Deep ensembles & data bootstrap

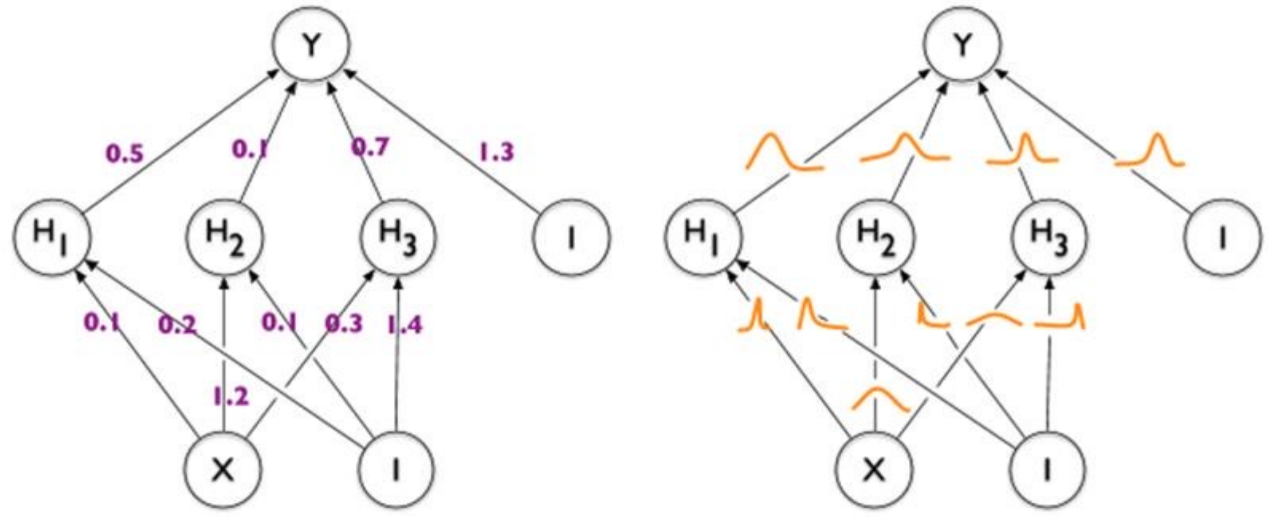






ATLAS-CONF-2022-035

Bayesian Methods

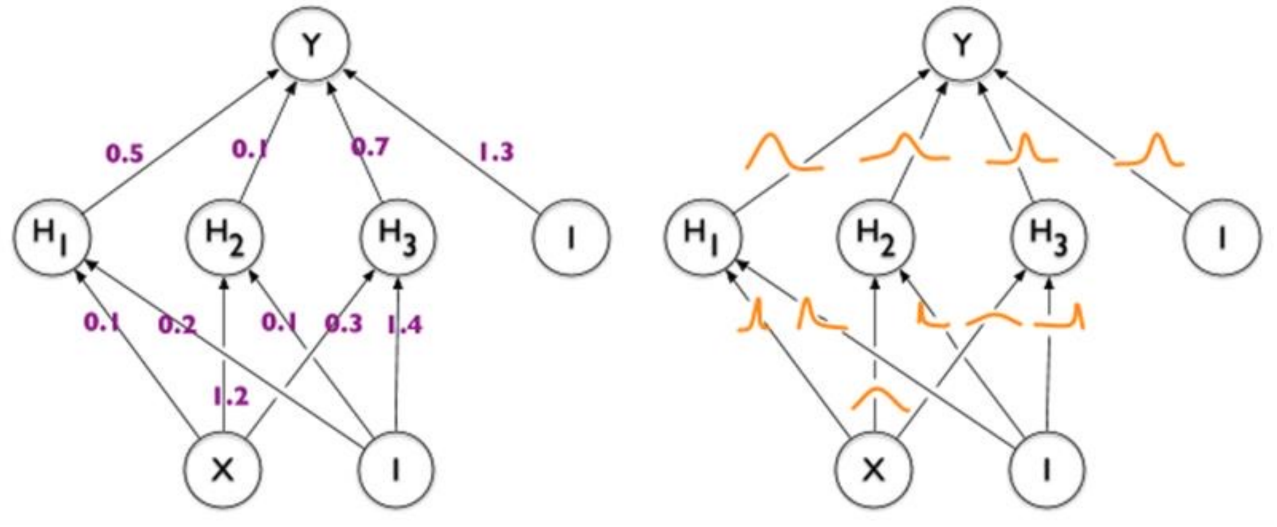


$$p(y|x,\mathcal{D}) = \int p(y|x,w)p(w|\mathcal{D})dw \approx \frac{1}{N} \sum_{\substack{i=1\ldots N \\ w_i \sim p(w|\mathcal{D})}} p(y|x,w_i)$$

Aleatoric Uncertainty:
Density Model

Model Uncertainty:
Posterior on weights

Bayesian Methods



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

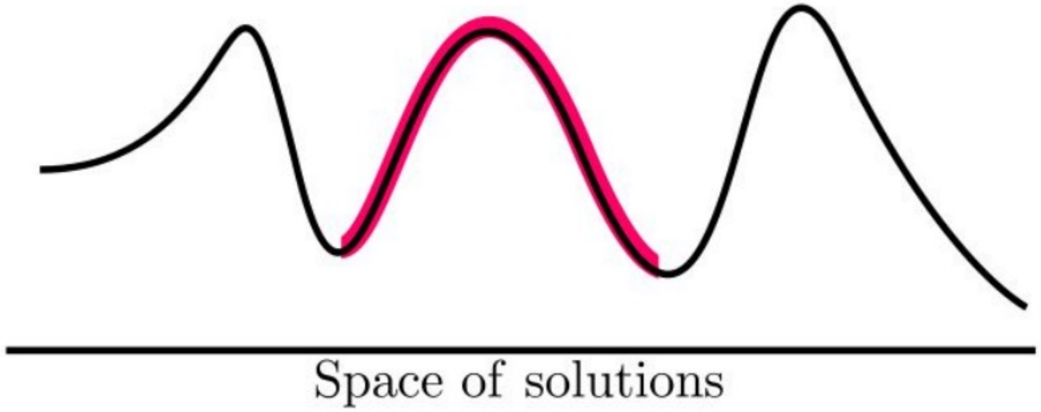
$$p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{\int p(\mathcal{D}|w)p(w)dw}$$

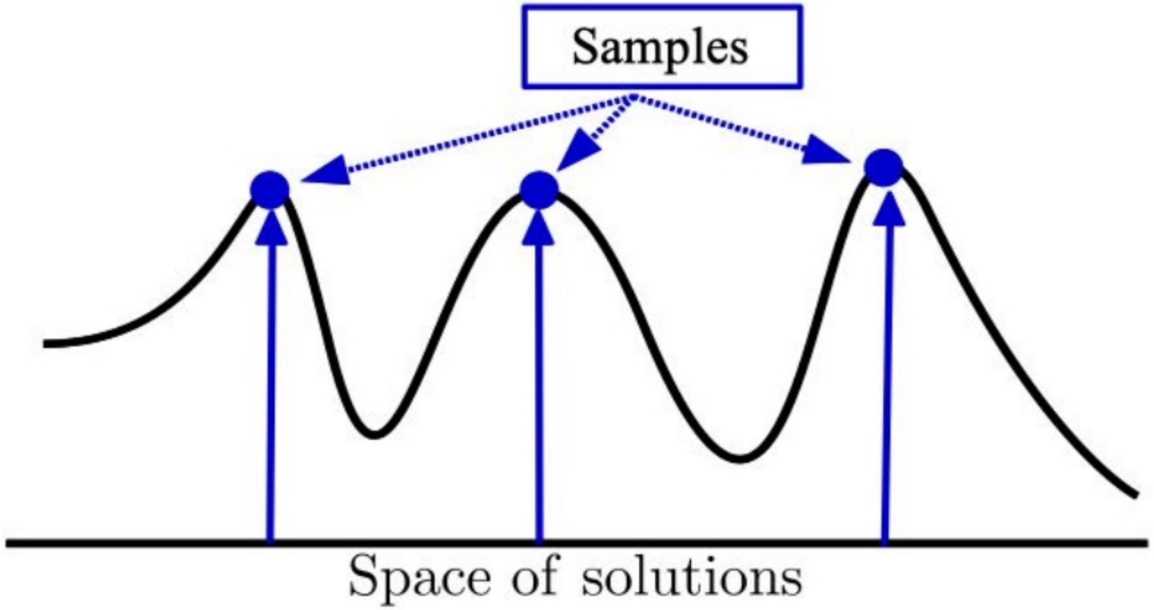
Prior on weights

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

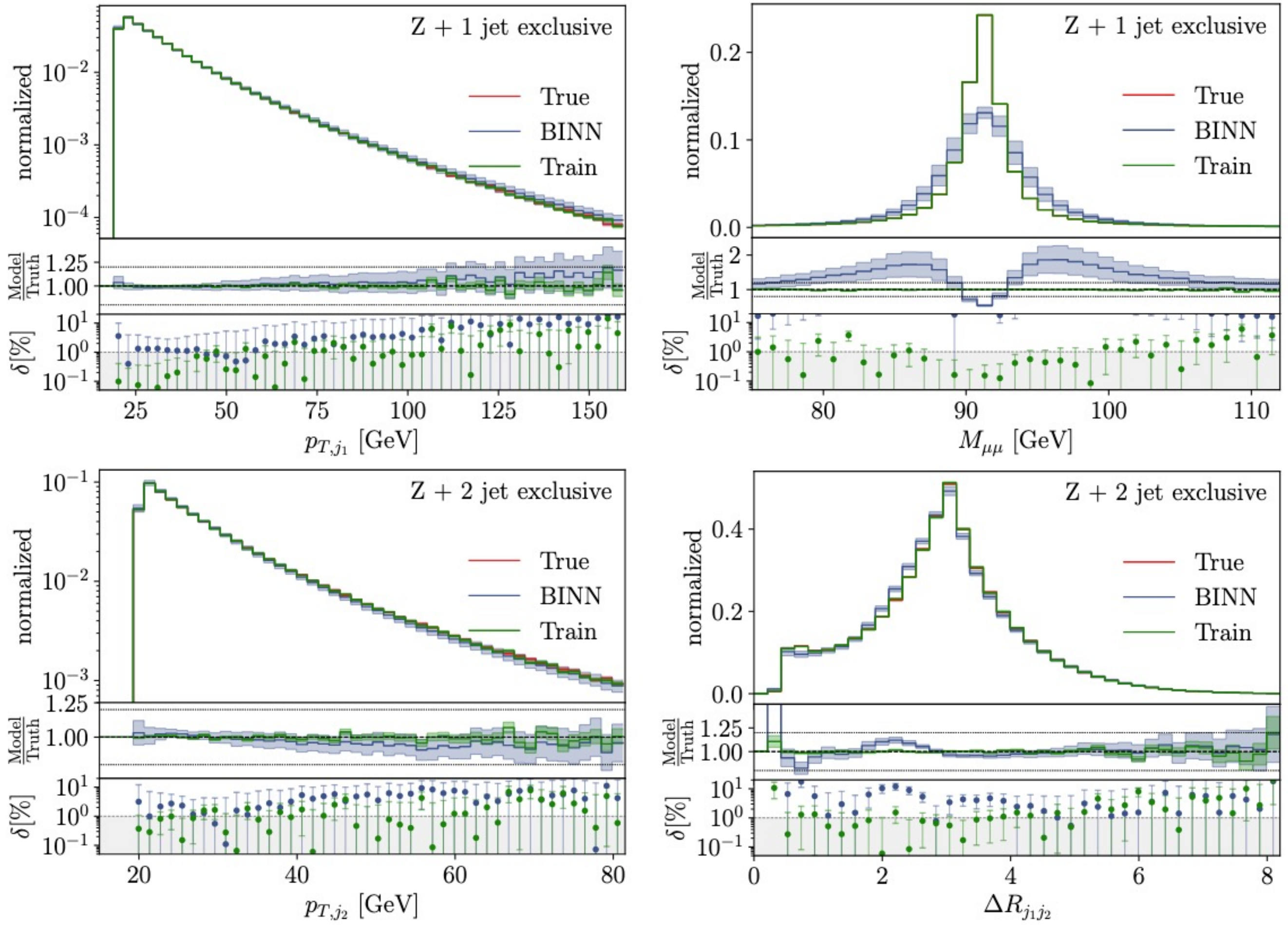
Slide credit: B. Lakshminarayanan

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

Wrapping Up

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 – Uncertainty on approximate LR or calibration procedure?
- Anomaly Detection
- …

Are current ML Uncertainty Quantification methods sufficient for our needs?

Backup

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

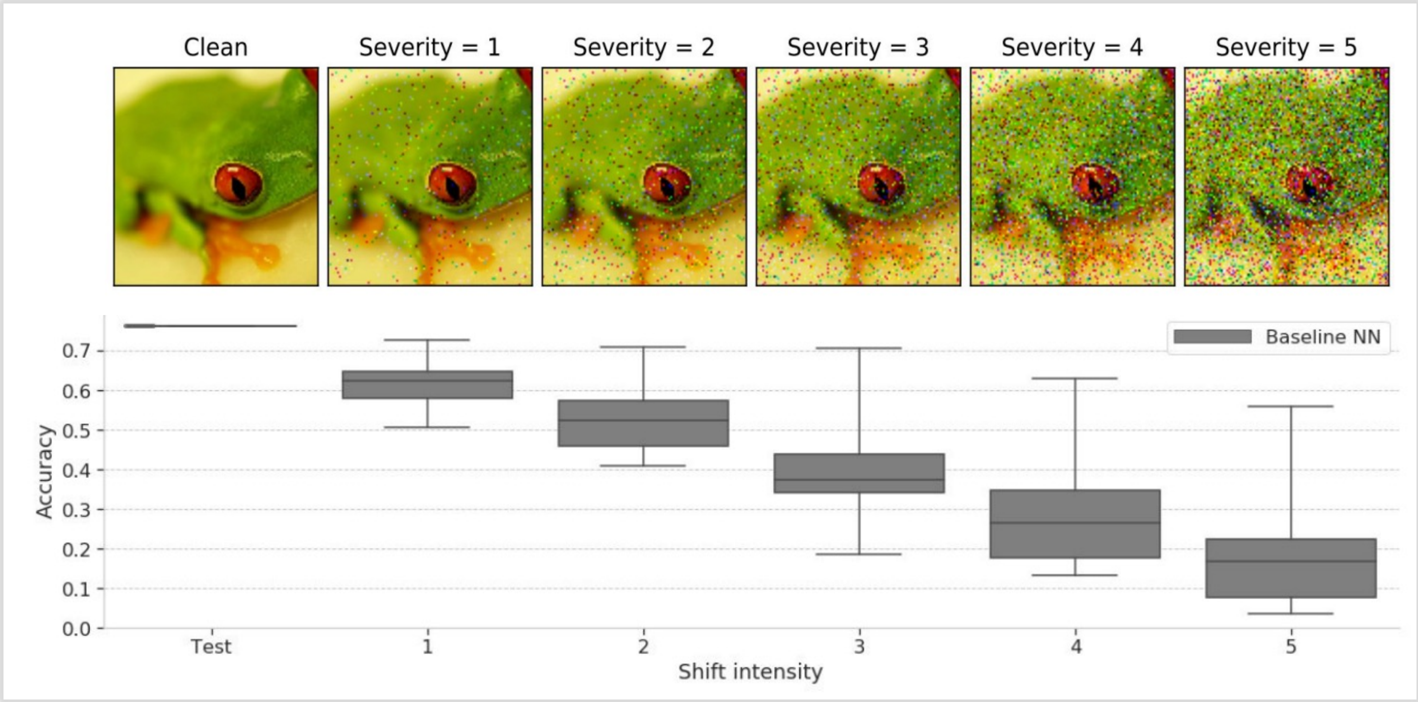


Image: [1903.12261](1903.12261)

What Kind of Model Are We Talking About?

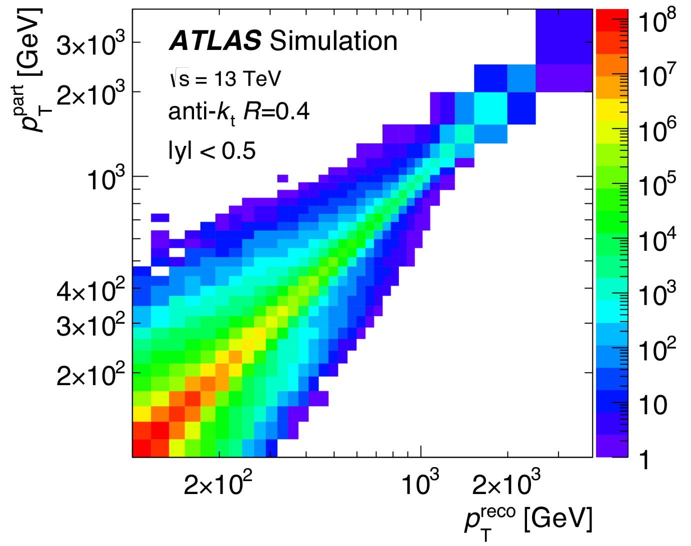
Perhaps a more important distinction between the perspective of physicists and machine learning researchers has to do with the use of the term "model" and what exactly is uncertain. In physics, the systematic and epistemic uncertainty is typically associated to our understanding of the underlying physics and "the model" usually refers to the physics model, detector model encapsulated in a simulation. In contrast, for machine learning research, "the model" usually refers to the trained model $f \in F$ used as described in Section 41.2.1 (or the class of functions F itself). This makes sense if we recall that in the bulk of machine learning research, one has little insight into the process that generated the data (e.g. images of cats and dogs, natural language, etc.). In that sense, the epistemic uncertainty in machine learning is usually associated to uncertainty in the model parameters φ after training, which would be reduced if one could collect more training data (see Ref. [328] for this point of view).

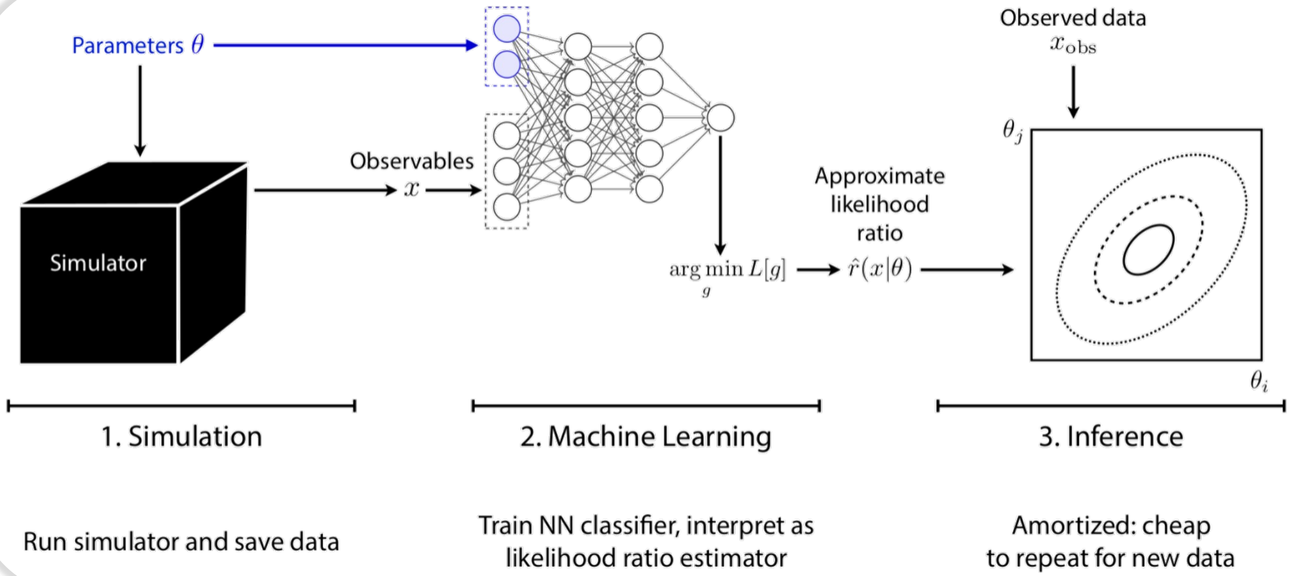
From talk by K. Cranmer
Also in PDG review of ML in HEP

When are ML Model Uncertainties Needed?

Neural Unfolding Methods

- Many methods, e.g. reweighting, neural posterior estimation, neural empirical Bayes
- Poorly fit ML model of response matrix
  → bad unfolded spectra





Simulation-Base Inference

- Example: neural network approximates likelihood ratio
- Poorly fit ML model
  → bad model of LR
  → poor parameter inference

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

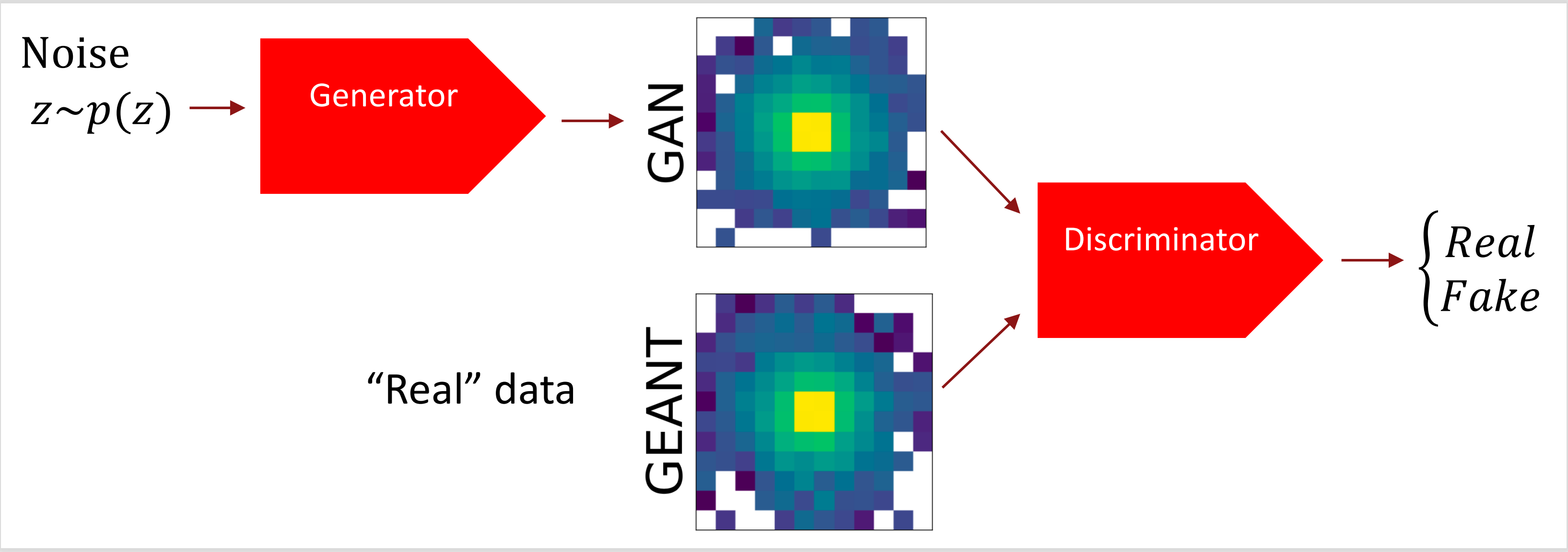
StyleGAN v2

(Karras et al, 2019)

Aleatoric Uncertainty in HEP with Generative Models

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

Generative Adversarial Networks:



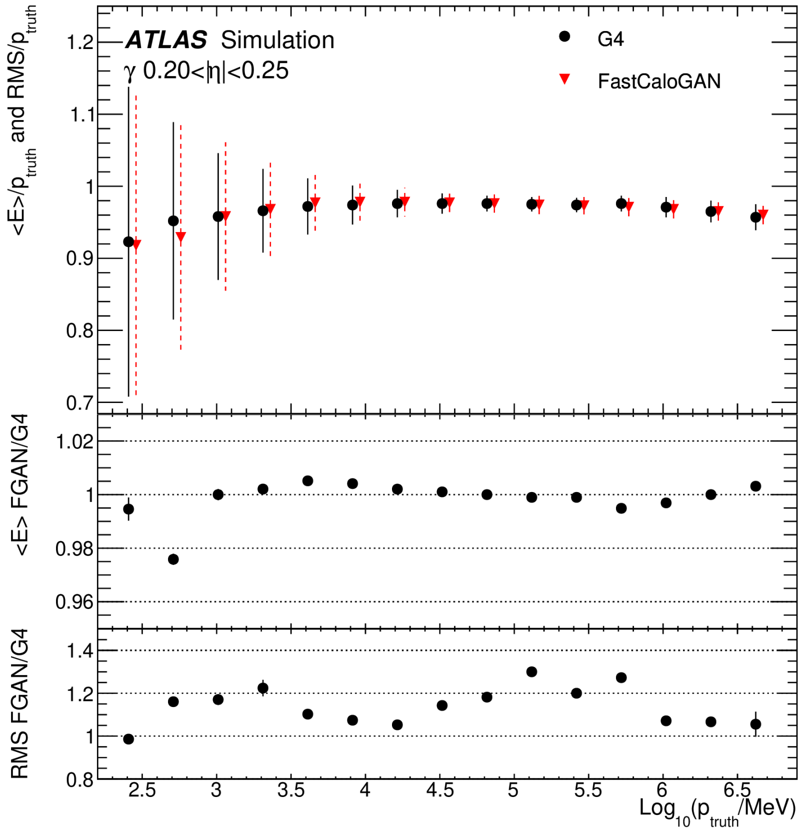
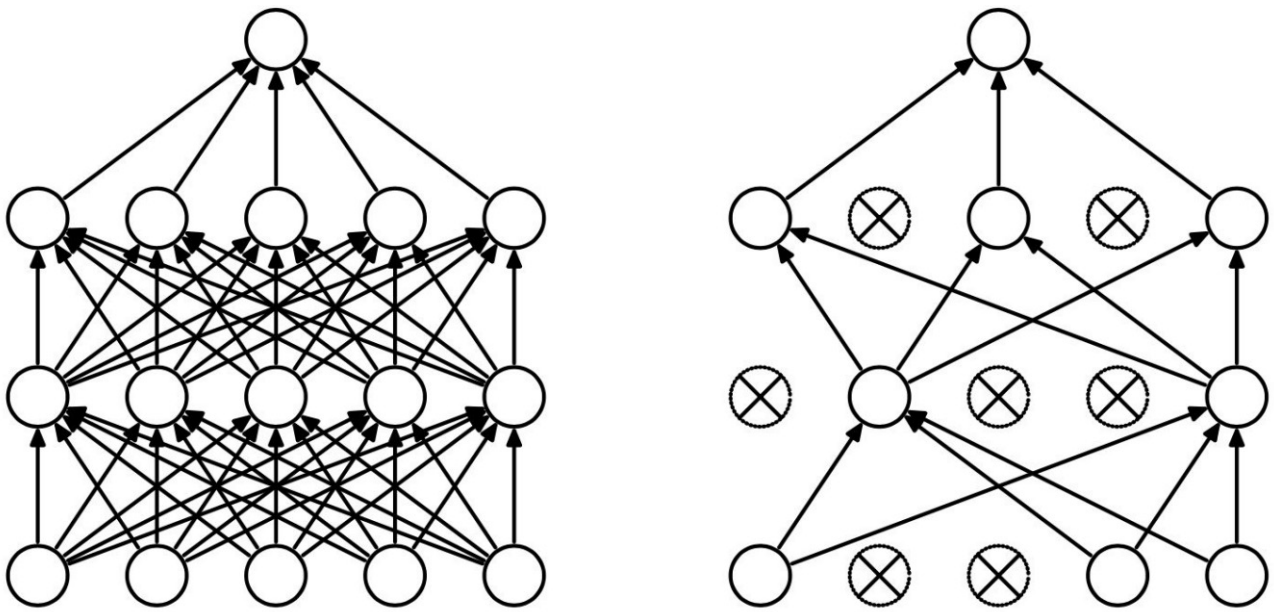


Image Credit: 1712.10321

arXiv:2109.02551

Monte Carlo Dropout for Epistemic Uncertainty



(a) Standard Neural Net

(b) After applying dropout.

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

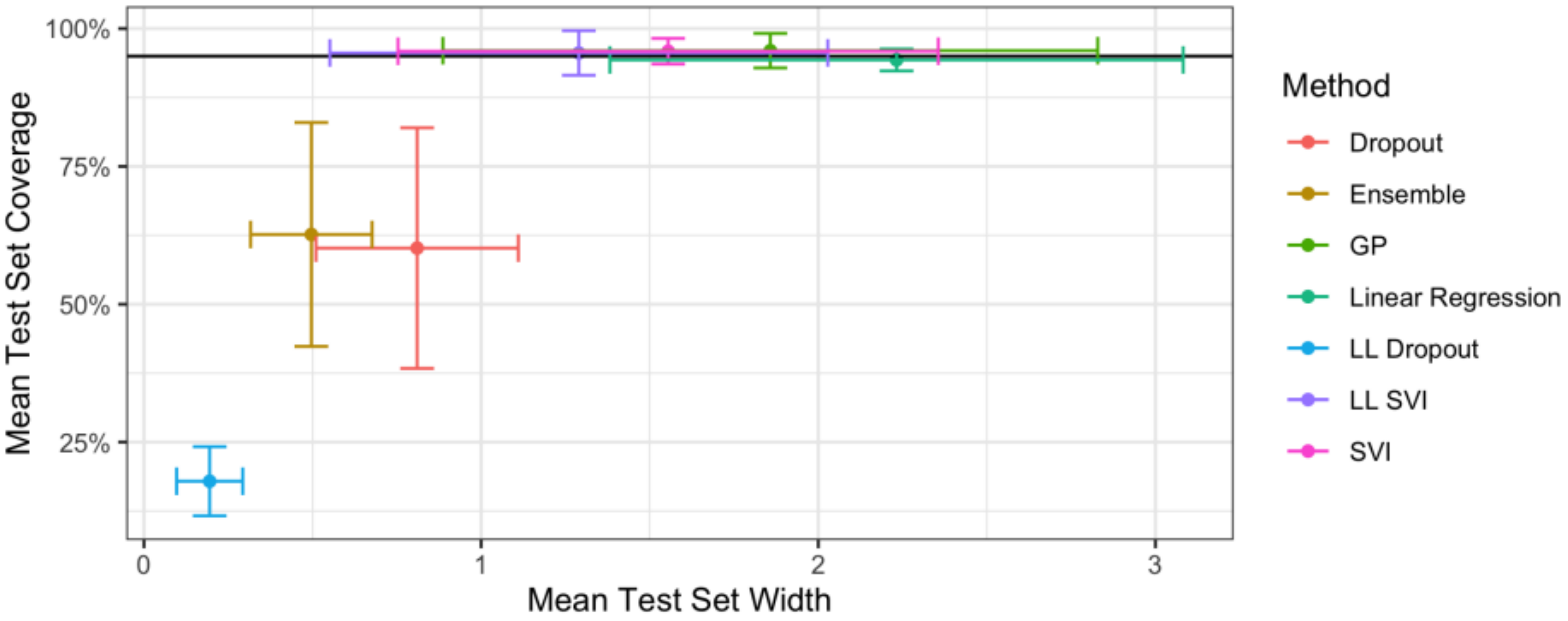
Different random Dropouts

Randomly drop connections between neurons, using Bernoulli distribution

Can be viewed as a Variational Approximation

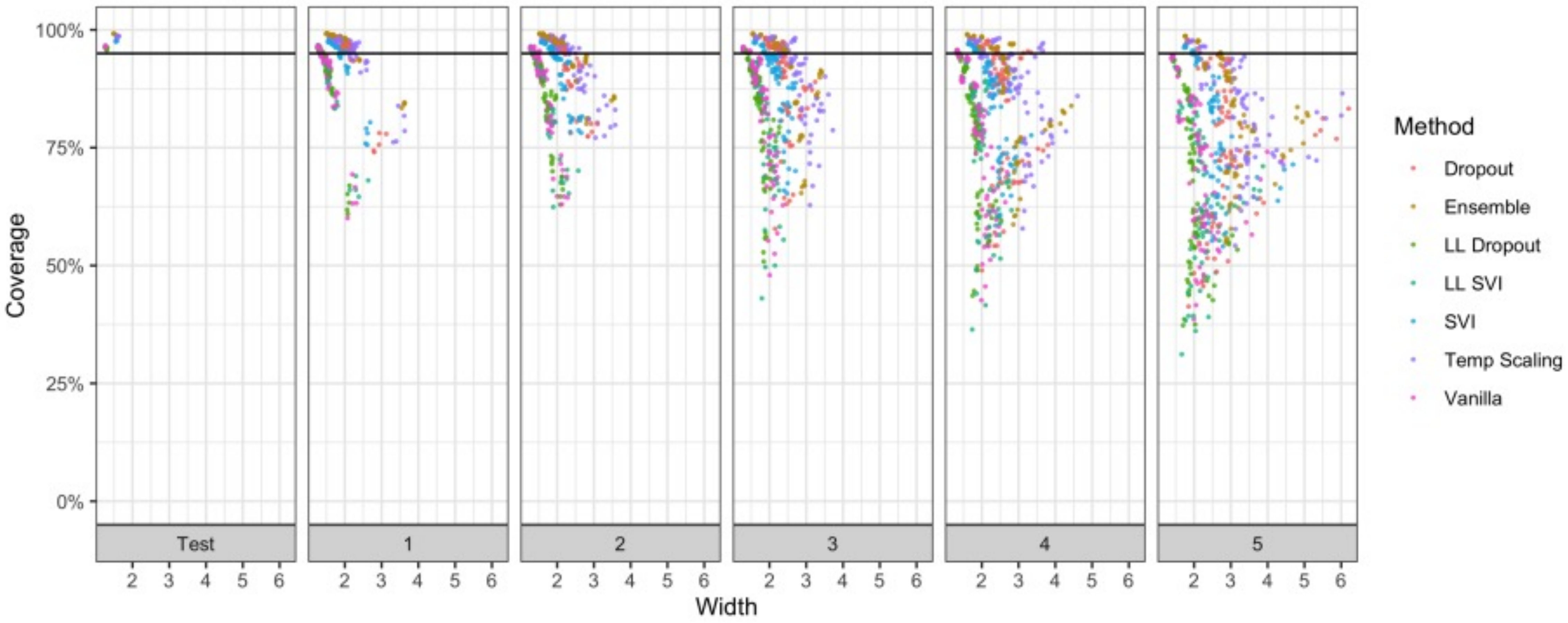
Gal, Ghahramani, [1506.02142](1506.02142)

Comparisons



[Kompa et. al, 2010.03039]()

Comparisons with Data Corruptions



Kompa et. al, 2010.03039