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 Hitchhiker's 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/
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

- Statistical Uncertainty
- Approximation Error
- Epistemic Uncertainty
- Systematic Uncertainty
- Data Uncertainty
- Aleatoric Uncertainty
- Theory Uncertainty
- Estimation Error
- Model Uncertainty
- Distribution Shift
Machine Learning in the Data Analysis Pipeline

$\theta$ – 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?
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
  $$w^* = \arg \min_w L = \arg \min_w \frac{1}{N} \sum_i \mathcal{L}(y, f_w(x))$$
Reconstruction, data selection, event classification enable us to define powerful summary statistics

\[ T_{w^*,\phi}(x) : \mathbb{R}^{10^8} \rightarrow \mathbb{R} \]

Estimate likelihood for frequentist parameter inference:

\[ p(T_{w^*,\phi}(x) \mid \lambda(\theta)) \]

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

† Ignoring Systematics for the moment
Optimal vs. Correct

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

\[ T_{w^*,\phi}(x) : \mathbb{R}^{10^8} \rightarrow \mathbb{R} \]

Estimate likelihood for frequentist parameter inference:

\[ p(T_{w^*,\phi}(x)|\lambda(\theta)) \]

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

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

† Ignoring Systematics for the moment

Related / similar discussions:
Cranmer talk, Nachman, 1909.03081
Optimal vs. Correct

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

\[ T_{w^*,\phi}(x) : \mathbb{R}^{10^8} \to \mathbb{R} \]

Estimate likelihood for frequentist parameter inference:

\[ p \left( T_{w^*,\phi}(x) \mid \lambda(\theta) \right) \]

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

† Ignoring Systematics for the moment

Related / similar discussions:
Cranmer talk, Nachman, 1909.03081
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
Domain / Distribution / Dataset Shift

\[ p_{\text{TEST}}(x, y) \neq p_{\text{TRAIN}}(x, y) \]

Examples:

- **Covariate Shift**: \( p(y|x) \) fixed but \( p_{\text{TEST}}(x) \neq p_{\text{TRAIN}}(x) \)
- **Label Shift**: \( p(x|y) \) fixed but \( p_{\text{TEST}}(y) \neq p_{\text{TRAIN}}(y) \)
- **Concept Shift**: \( p(y) \) fixed but \( p_{\text{TEST}}(x|y) \neq p_{\text{TRAIN}}(x|y) \)
## Imperfect Correspondence: My View*

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

*Even within the ML community, these terms can be ambiguous*
Uncertainty Estimation Approaches in Deep Learning

Aleatoric Uncertainty

Randomness of data → Typically described by probability distributions

**Density Networks**

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

Train neural network to predict per-example parameters

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

*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:

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

**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:

Noise $z \sim p(z)$ → Generator → GAN → Discriminator → \{Real, Fake\}

“Real” data

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

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

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

Aleatoric Uncertainty in HEP: Neural Unfolding

\[ p_{\text{reco}}(y) = \int p(y|x)p_{\text{true}}(x) \quad y = Rx \]

Continuous Form  
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?

→ 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 \( hh \rightarrow 4b \) example: Uncertainties from Deep ensembles & data bootstrap

ATLAS Preliminary

\( \sqrt{s} = 13 \text{ TeV}, 126 \text{ fb}^{-1} \)

\( ggF \) CR1

\( 4b \) Data

Normalized 2b Data

Stat. Error

Events / 25 GeV

\( m_{HH} \text{ [GeV]} \)

\( 4b / 2b \)
Bayesian Methods

\[
p(y|x, D) = \int p(y|x, w)p(w|D)dw \approx \frac{1}{N} \sum_{i=1...N} p(y|x, w_i)
\]

Aleatoric Uncertainty: Density Model

Model Uncertainty: Posterior on weights
Bayesian Methods

\[ p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw \approx \frac{1}{N} \sum_{i=1\ldots N}^{\infty} p(y|x, w_i) \]

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

Prior on weights

Intractable Integral
Approximating the Posterior

\[ p(w|D) \] 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

Randomly drop connections between neurons, using Bernoulli distribution

Can be viewed as a Variational Approximation

\[ f(x) \rightarrow \left\{ \begin{array}{ll}
\text{Mean}[f^1 \ldots f^N] \\
\text{Var}[f^1 \ldots f^N]
\end{array} \right. \]

Different random Dropouts

Gal, Ghahramani, 1506.02142
Comparisons

Method
- Dropout
- Ensemble
- GP
- Linear Regression
- LL Dropout
- LL SVI
- SVI
Comparisons with Data Corruptions

Kompa et. al, 2010.03039
Systematic Uncertainties / Domain Shift in HEP
Systematic Uncertainties

Source of uncertainty

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>( \mu_{\text{VH}(\rightarrow c\bar{c})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>21.5</td>
</tr>
<tr>
<td>Statistical</td>
<td>16.2</td>
</tr>
<tr>
<td>Systematics</td>
<td>14.0</td>
</tr>
</tbody>
</table>

Statistical uncertainties

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>( \mu_{\text{VH}(\rightarrow c\bar{c})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data statistics only</td>
<td>13.0</td>
</tr>
<tr>
<td>Floating normalisations</td>
<td>7.2</td>
</tr>
</tbody>
</table>

Theoretical and modelling uncertainties

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>( \mu_{\text{VH}(\rightarrow c\bar{c})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( VHF(\rightarrow c\bar{c}) )</td>
<td>2.1</td>
</tr>
<tr>
<td>( Z+\text{jets} )</td>
<td>7.7</td>
</tr>
<tr>
<td>Top-quark</td>
<td>5.6</td>
</tr>
<tr>
<td>W+\text{jets}</td>
<td>3.4</td>
</tr>
<tr>
<td>Diboson</td>
<td>0.8</td>
</tr>
<tr>
<td>( VHF(\rightarrow b\bar{b}) )</td>
<td>0.8</td>
</tr>
<tr>
<td>Multi-Jet</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Simulation statistics

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>( \mu_{\text{VH}(\rightarrow c\bar{c})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jets</td>
<td>3.7</td>
</tr>
<tr>
<td>Leptons</td>
<td>0.4</td>
</tr>
<tr>
<td>( E_{\text{T}} ) jets</td>
<td>0.5</td>
</tr>
<tr>
<td>Pile-up and luminosity</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Flavour tagging

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>( \mu_{\text{VH}(\rightarrow c\bar{c})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c ) -jets</td>
<td>2.3</td>
</tr>
<tr>
<td>( b ) -jets</td>
<td>1.2</td>
</tr>
<tr>
<td>Light-jets</td>
<td>0.7</td>
</tr>
<tr>
<td>( r ) -jets</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Truth-flavour tagging

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>( \mu_{\text{VH}(\rightarrow c\bar{c})} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta R ) correction</td>
<td>3.0</td>
</tr>
<tr>
<td>Residual non-closure</td>
<td>1.4</td>
</tr>
</tbody>
</table>

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$^1$

---

Work In Progress
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 \( \nu \).

- **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 \( \nu \).

From [PDG review of ML in HEP](https://pdg.lbl.gov/2022/reviews/rpp2022reno-sr.pdf)
Error Propagation – Standard Approach

Train on \( \{x_i^0, y_i^0\} \) w/ nominal nuisance \( \nu_0 \) → learn fixed model \( f(x) \)

Evaluate on \( \nu \) variations → Observe effects

\[ \nu = \nu_0 \quad \nu = \nu_1 \]
Error Propagation – Standard Approach

Train on \( \{ x_i^0, y_i^0 \} \) w/ nominal nuisance \( \nu_0 \) \( \rightarrow \) learn fixed model \( f(x) \)

Evaluate on \( \nu \) variations \( \rightarrow \) Observe effects

\[
p(x|\nu) \rightarrow \int_{\{x:f(x)>c\}} p(x|\nu) dx
\]

NOT Optimal Under Variations
Data Augmentation / Marginalization

Training sample includes $\nu$ variations:

$$x \sim \int p(x | y, \nu) p(\nu) d\nu$$

Smeared samples $\rightarrow$ “smeared” fixed model $f_{\text{smeared}}(x)$

Smeared Classifier:
- Less sensitive to variations
- Not optimal for any $\nu$

Slide Credit: K. Cranmer
Data Augmentation / Marginalization

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

Smeared samples $\rightarrow$ “smeared” fixed model $f_{\text{smeared}}(x)$

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

Training on flat mass distribution
Pivoting / Enforcing Domain Invariance

Want to train model $f(x)$ such that:

$$p(f | \nu) = p(f)$$

$f$ is a **pivotal quantity**

Louppe, MK, Cranmer, [1611.01046](https://arxiv.org/abs/1611.01046)
Pivoting / Enforcing Domain Invariance

Adversarial Approach:

Adversary to predict $\nu$ from model output $f$

Min-Max Game: Penalize Classifier when Adversary succeeds

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

$$E(\theta_f, \theta_r) = L_f(\theta_f) - \lambda L_r(\theta_f, \theta_r).$$

“Regularize” training with Adversary

Louppe, MK, Cranmer, 1611.01046
Adversarial Approach:

- Non-Adversarial training
- Optimal tradeoff of performance vs. robustness

- W-jets vs QCD
- ν = Pileup variations

Louppe, MK, Cranmer, 1611.01046
Pivoting / Enforcing Domain Invariance

Regularizing Correlations: Non-adversarial approach

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

\[
L = L_{\text{classifier}}(\bar{y}, \bar{y}_{\text{true}}) + \lambda \text{dCov}^2_{\bar{y}_{\text{true}} = 0}(\bar{m}, \bar{y})
\]

\[
d\text{Cov}(X, Y) = \langle |X - X'||Y - Y'| \rangle \\
+ \langle |X - X'| \rangle \langle |Y - Y'| \rangle \\
- 2 \langle |X - X'||Y - Y'| \rangle
\]
Parameterizing Models

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

Slide Credit: K. Cranmer

Cranmer, Louppe, Pavez, 1506.02169
Parameterizing Models

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

Fixed Model

Parameterized Model

No Systematic Variation

With Systematic Variation

Ghosh, Nachman, Whiteson, 2105.08742
Comparing Approaches

Example:
- Classifier: $h \rightarrow \tau\tau$ vs Bkg
- Uncertainty: $\tau$ energy scale

Parameterized Classifier:
$$f(x; \nu)$$

How to choose the $\nu$?
→ Profile in Likelihood

Data with Nominal Nuisance value
Data with Varied Nuisance value

Ghosh, Nachman, Whiteson, 2105.08742
Cranmer, Louppe, Pavez, 1506.02169
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

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

\[ T(x) : \mathbb{R}^{10^8} \rightarrow \mathbb{R} \]

Histogram for density estimation, with bin counts:

\[ \{t_i\}_{i=1}^{n_{bins}} \]

Binned Likelihood: \( p(t_i | \theta, \nu) = \text{Poiss}(t_i | \mu(\theta, \nu)) \)

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

\[ p(T(x) | \theta) \]
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!
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
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
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

\[ \mathcal{L} = \sum_{n=1}^{N} \langle \log p_X(x_n | \theta) \rangle_{\theta \sim q_\phi(\theta)} - \text{KL} \left( q_\phi(\theta), p(\theta) \right) \]

- 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 $x_{\text{obs}} \sim p(x_{\text{obs}} | \theta_{\text{true}})$.

Then, estimate the posterior

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

Or a likelihood ratio

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

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:

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

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

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \approx \hat{r}(x|\theta)p(\theta)$$
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 $\theta$ is contained within the interval is larger than the nominal coverage probability $1 - \alpha$, then the approximate posterior $\hat{p}(\theta|x)$ has coverage.
Can Inference Goal Drive Training?

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

Examples: **NEOS** and **INFERNO**