Neural Simulation-Based Inference at the LHC

ML4Jets 2026, Vienna

Jay Sandesara

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

https://indico.global/event/15240/

iris hep
Institute for Research & Innovation
in Software for High Energy Physics

What is Simulation-Based Inference ?

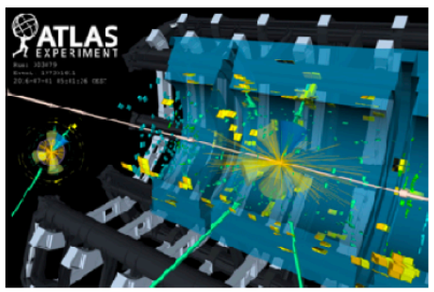
What is Neural Simulation-Based Inference ?

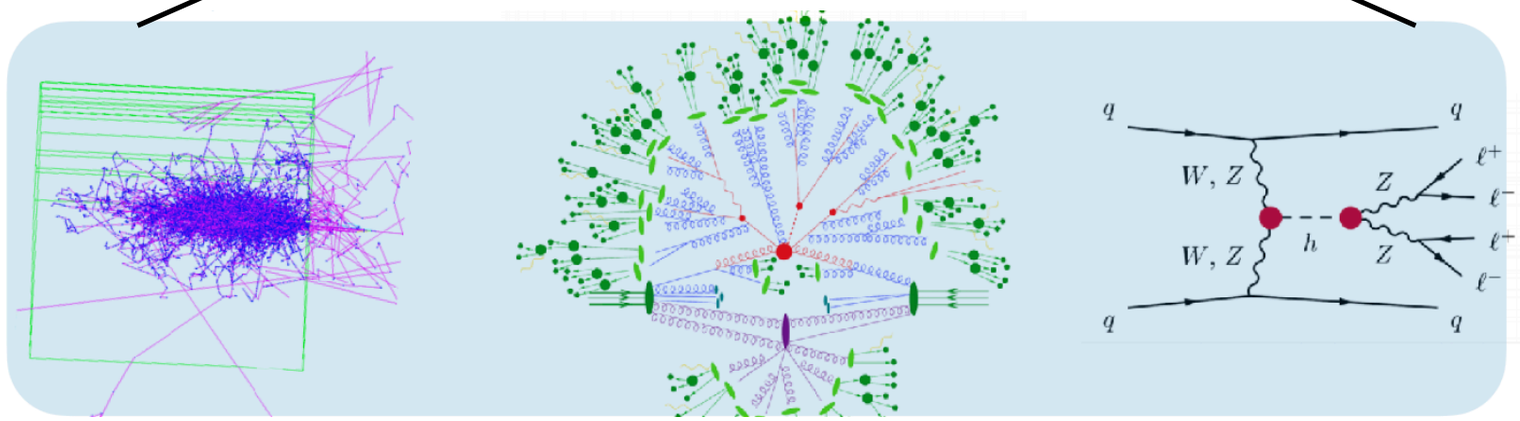
How do we apply this to the LHC ?

Introduction

"Latent" or unobserved space

What we observe





What we measure

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + i\bar{\psi}\not{D}\psi + h.c. + \bar{\psi}_i y_{ij} \psi_j \phi + h.c. + |D_\mu \phi|^2 - V(\phi)$$

Theory model parameterized by some μ

Data x

$O(10^8)$-dimensional x-space

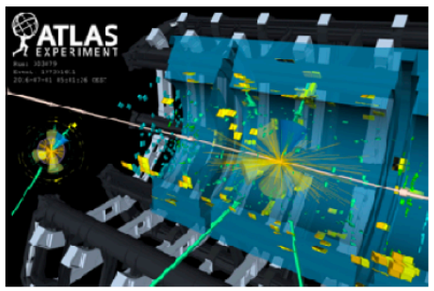
"High-dimensional"

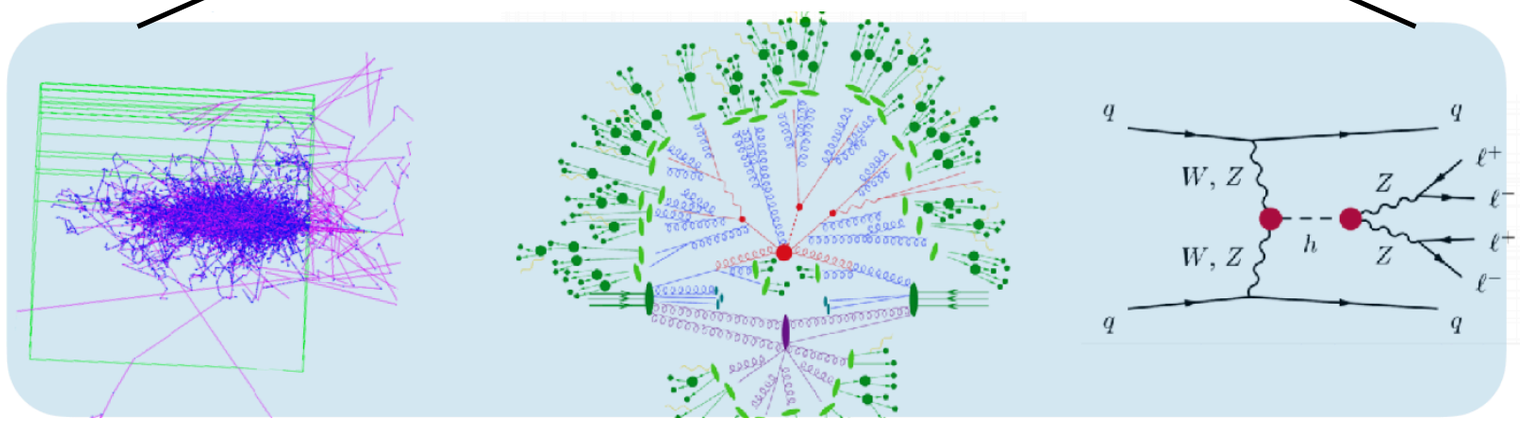
Visuals adapted from Lukas' slides

Introduction

"Latent" or unobserved space

What we observe





What we measure

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + i\bar{\psi}\not{D}\psi + h.c. + \bar{\psi}_i y_{ij} \psi_j \phi + h.c. + |D_\mu \phi|^2 - V(\phi)$$

Theory model parameterized by some μ

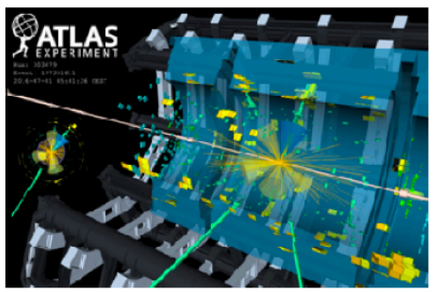
Data x

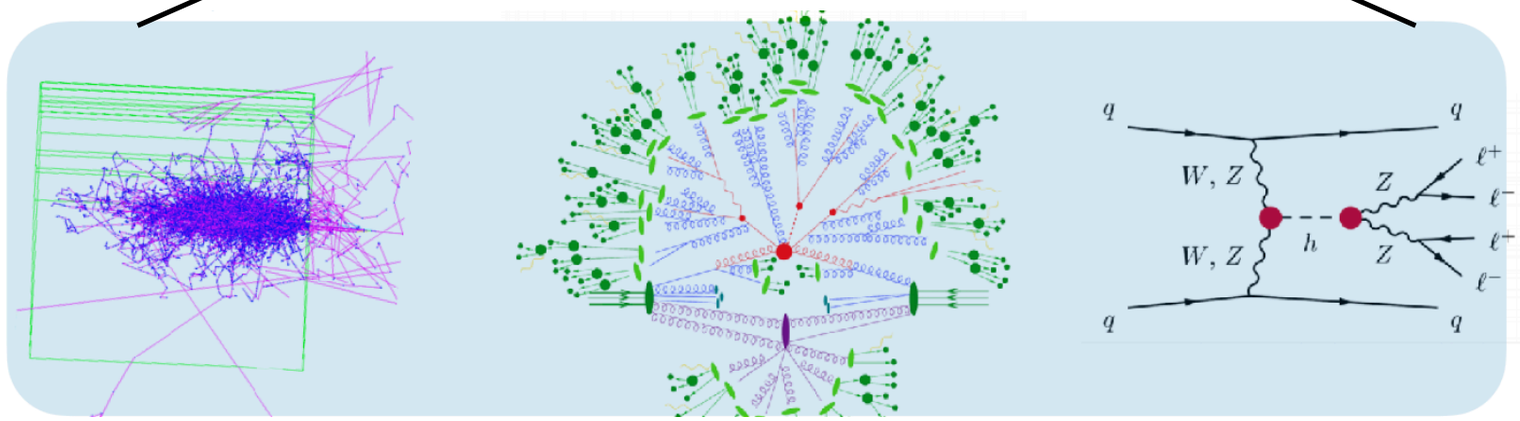
Inference goal: Likelihood $L(\mu; x) = p(x \mid \mu)$

Introduction

"Latent" or unobserved space

What we observe





What we measure

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + i\bar{\psi}\not{D}\psi + h.c. + \bar{\psi}_i y_{ij} \psi_j \phi + h.c. + |D_\mu \phi|^2 - V(\phi)$$

Theory model parameterized by some μ

Data x

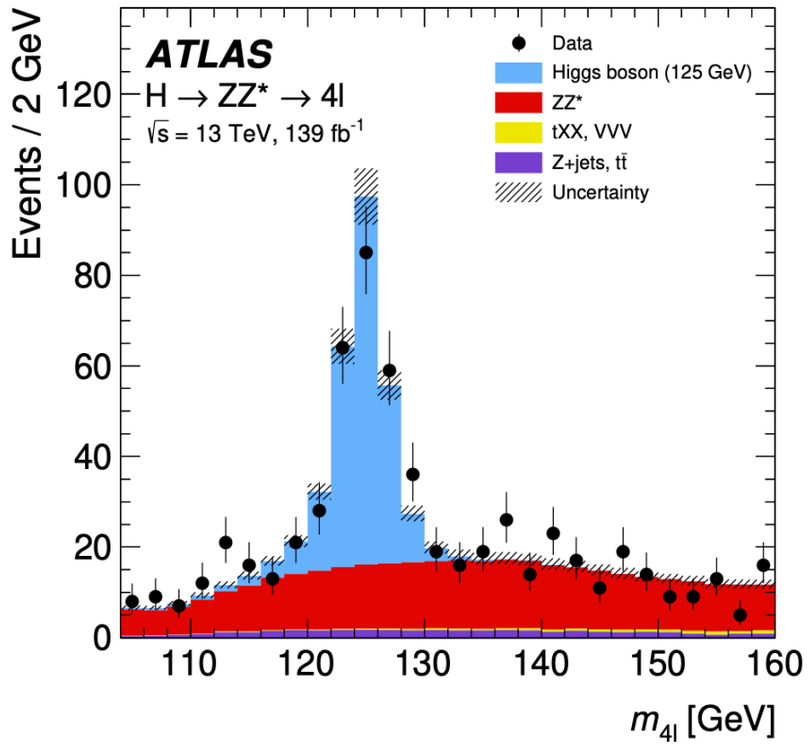
Inference goal: Likelihood $L(\mu; x) = p(x \mid \mu)$

"Likelihood-Free" Regime at the LHC

Simulate events $x \sim p(x \mid \mu)$ ✓

Compute likelihood $x, \mu \to p(x \mid \mu)$ ✗

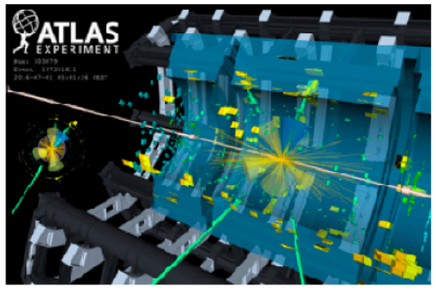
"Likelihood-Free" Regime → Simulation-Based Inference (SBI)

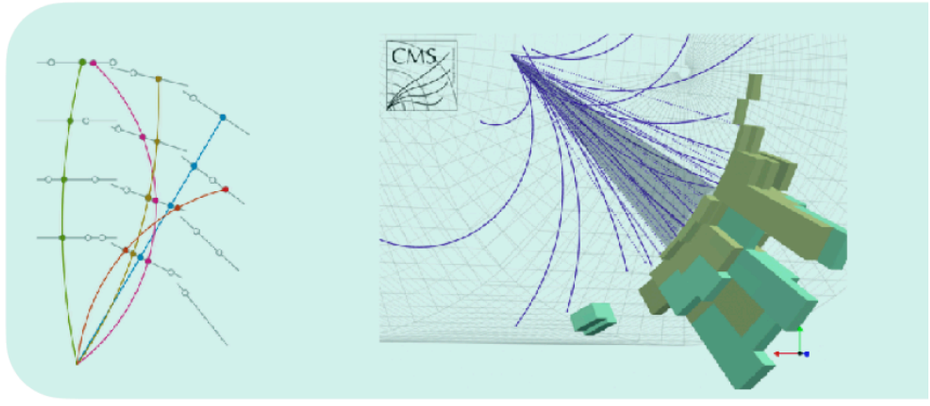


We have always been using SBI

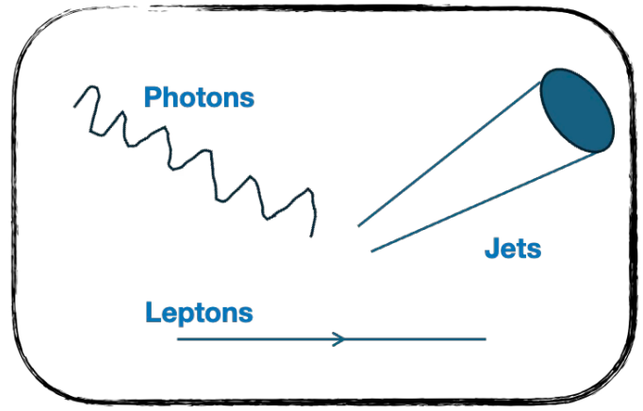
Traditional Inference

How do we fix the implicit model issue in traditional inference?





Physics objects & their momentum



$O(10^8)$-dimensional
x-space

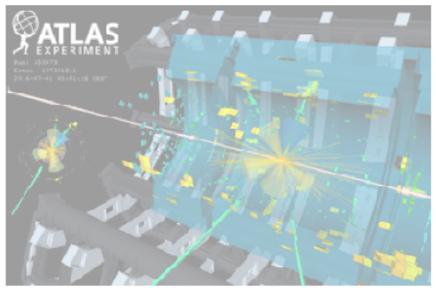
$O(10^2)$-dimensional
x-space

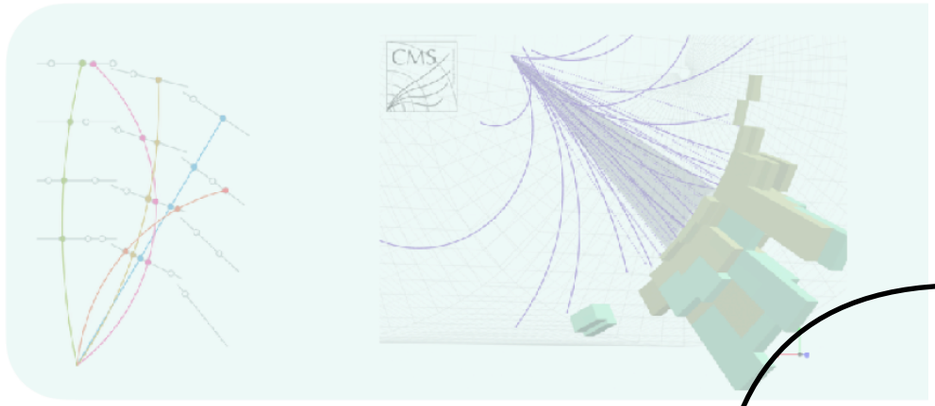
dimensionality reduction via reconstruction algorithms

Figure credit: Lukas Heinrich

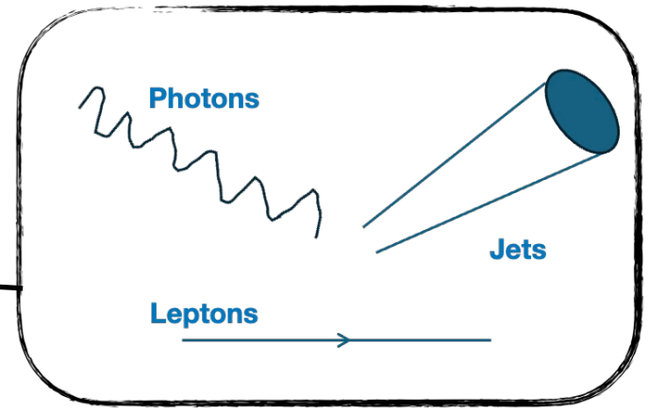
Traditional Inference

How do we fix the implicit model issue in traditional inference?





Physics objects & their momentum $x \in \mathbb{R}^{O(10^2)}$



$$L(\mu,\nu) = \left[\underbrace{\text{Pois}(n \mid \lambda(\mu,\nu))}_{\text{Extended Poisson term}} \prod_{e \in \text{events}} \underbrace{p\left(x_e \mid \mu,\nu\right)}_{\substack{\text{Probability Density} \\ \text{(intractable)}}} \right] \cdot \prod_{m \in \text{systs}} \underbrace{p_{\text{constr.}}(a_m \mid \nu_m)}_{\text{Constraint likelihood term}}$$

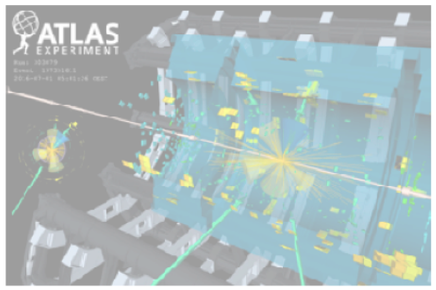
$n \rightarrow$ **observed # of events**
$\lambda \rightarrow$ **expected # of events**

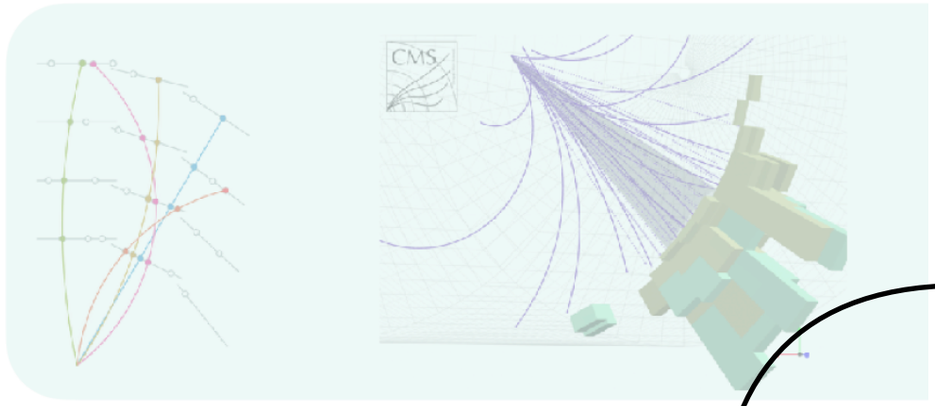
$\mu \rightarrow$ **parameters of interest**
$\nu \rightarrow$ **nuisance parameters**

$a_m \rightarrow$ **global observable associated with** m^{th} **nuisance parameter**

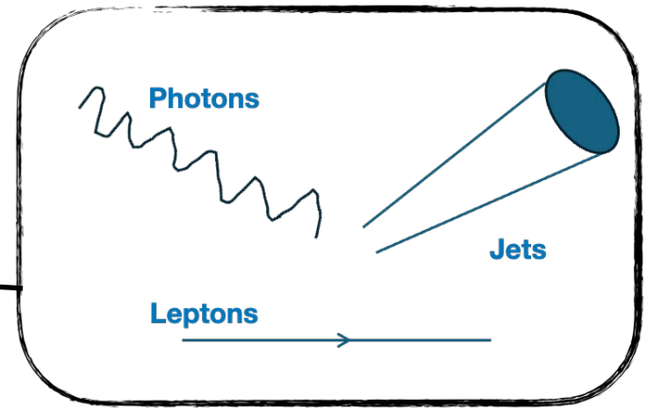
Traditional Inference

How do we fix the implicit model issue in traditional inference?





Physics objects & their momentum $x \in \mathbb{R}^{O(10^2)}$



$$L(\mu,\nu) = \left[\underbrace{\text{Pois}(n \mid \lambda(\mu,\nu))}_{\text{Extended Poisson term}} \underbrace{\prod_{e \in \text{events}} p\left(x_e \mid \mu,\nu\right)}_{\text{Probability Density (intractable)}} \right] \cdot \underbrace{\prod_{m \in \text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)}_{\text{Constraint likelihood term}}$$

✓ ✗ ✓

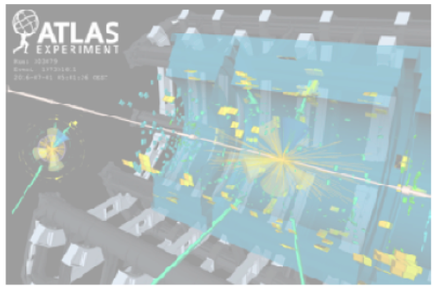
$n \rightarrow$ **observed # of events**
$\lambda \rightarrow$ **expected # of events**

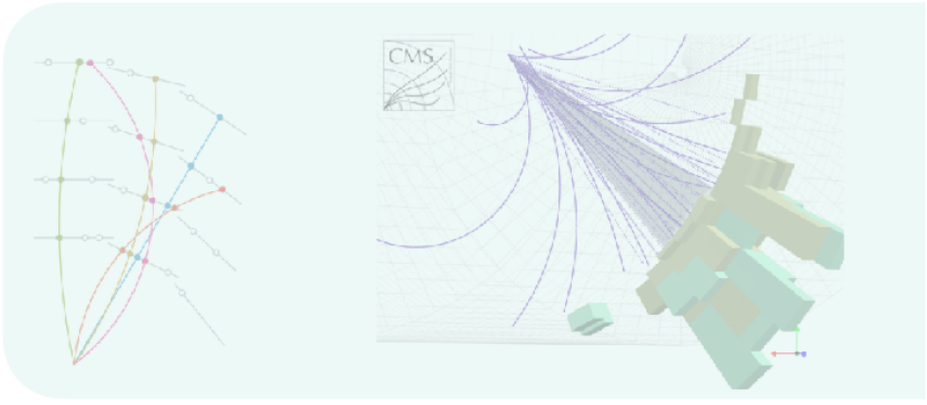
$\mu \rightarrow$ **parameters of interest**
$\nu \rightarrow$ **nuisance parameters**

$a_m \rightarrow$ **global observable associated with** m^{th} **nuisance parameter**

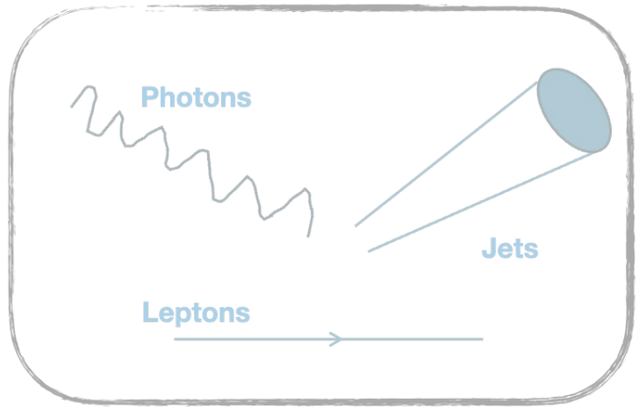
Traditional Inference

How do we fix the implicit model issue in traditional inference?





Physics objects & their momentum $x \in \mathbb{R}^{O(10^2)}$



$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e \in \text{events}} p\left(x_e \mid \mu,\nu\right)\right] \cdot \prod_{m \in \text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e \in \text{events}} p\left(s(x_e) \mid \mu,\nu\right)\right] \cdot \prod_{m \in \text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

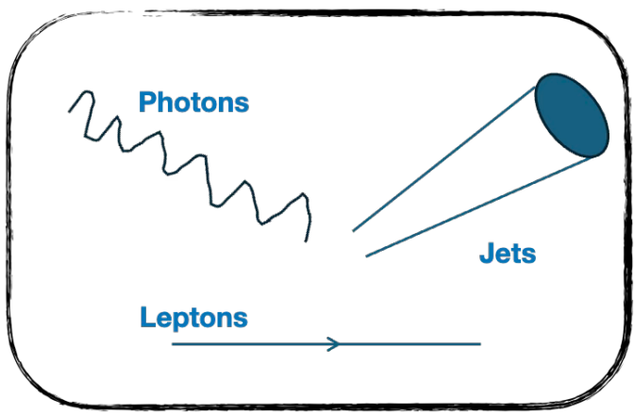
Rewrite with a dimensionality reduction mapping function $x \to s(x)$ e.g. $m_{4\ell}(x)$

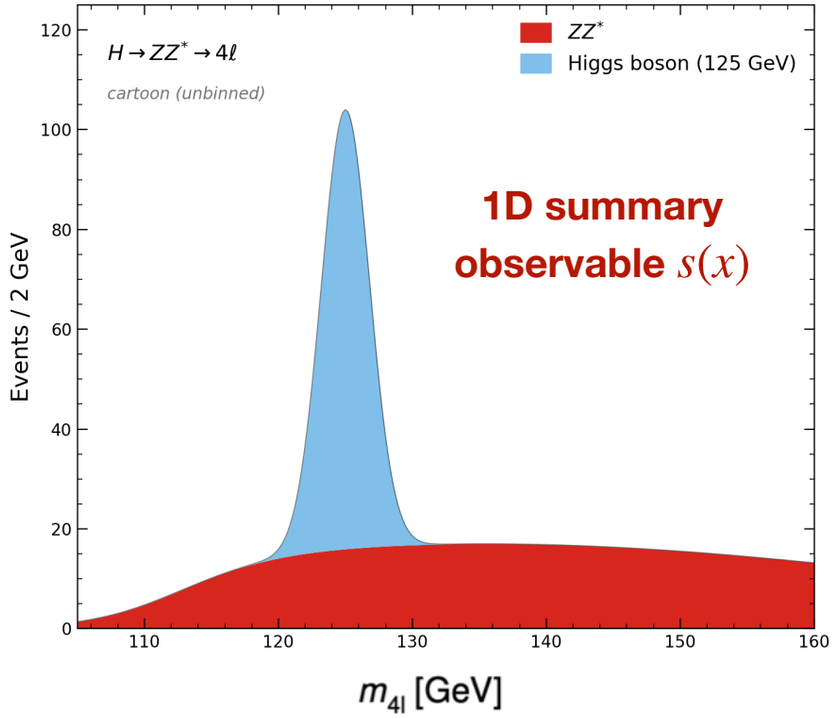
Traditional Inference

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e\in\text{events}} p\left(x_e \mid \mu,\nu\right)\right] \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e\in\text{events}} p\left(s(x_e) \mid \mu,\nu\right)\right] \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

Physics objects & their momentum $x \in \mathbb{R}^{O(10^2)}$





$$p\left(x_e \mid \mu,\nu\right) = \frac{1}{\sigma(\mu,\nu)}\frac{d\sigma}{dx}(x;\mu,\nu)$$

$$p\left(s(x_e) \mid \mu,\nu\right) = \frac{1}{\sigma(\mu,\nu)}\frac{d\sigma}{ds}(s;\mu,\nu)$$

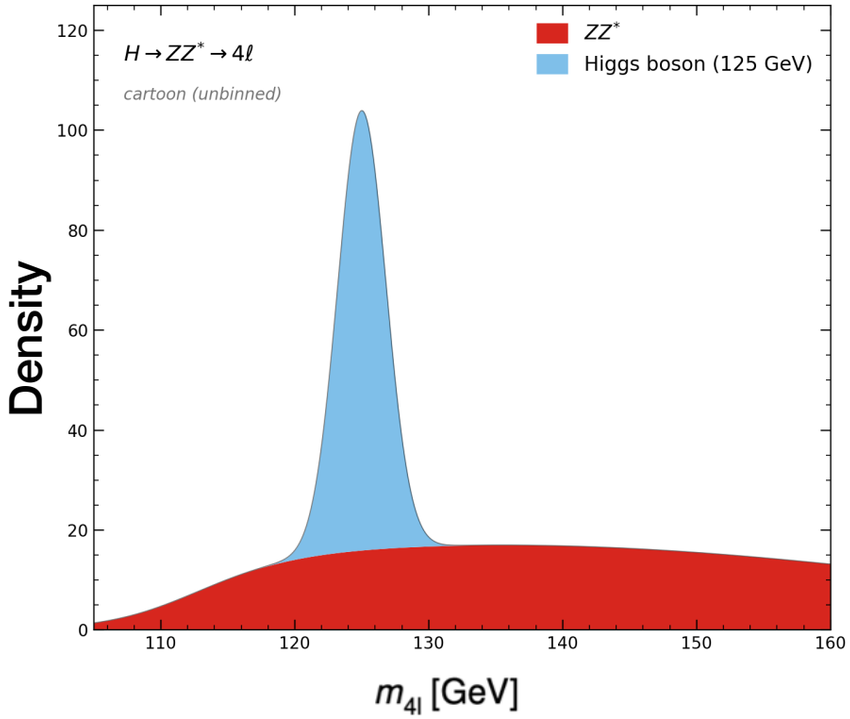
Traditional Inference

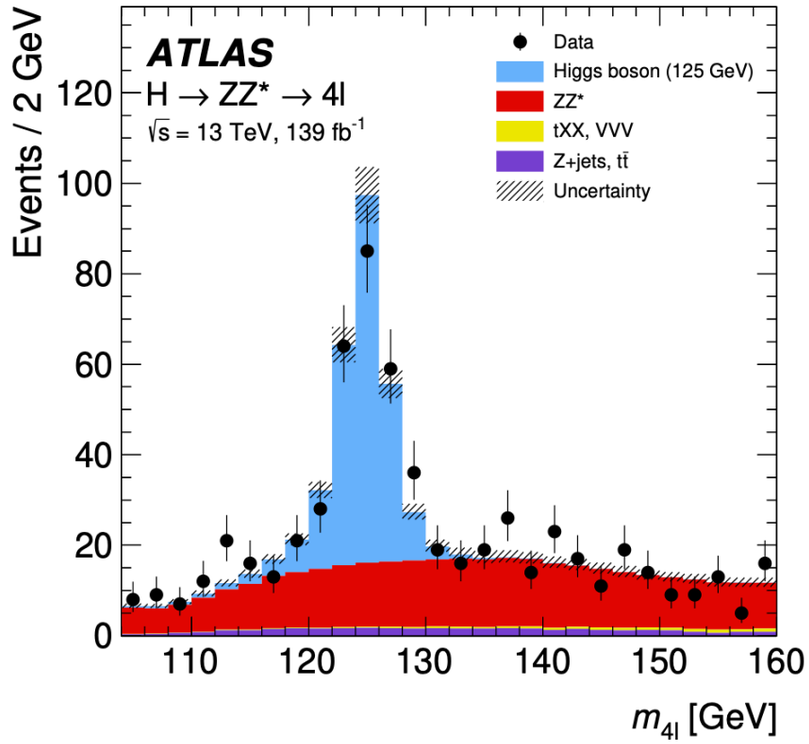
Continuous truth

$$p\left(s(x_e)\,|\,\mu,\nu\right) = \frac{1}{\sigma(\mu,\nu)}\frac{d\sigma}{ds}(s;\mu,\nu)$$

Binned model for density estimation

$$q_{\text{hist}}\left(s(x_e)\,|\,\mu,\nu\right) = \frac{1}{\lambda(\mu,\nu)}\frac{\lambda_b(\mu,\nu)}{\Delta_b}$$





The bin counts are modeled using simulation histograms

$\lambda_b \to$ expected yield in bin b

"Simulation-based" Inference

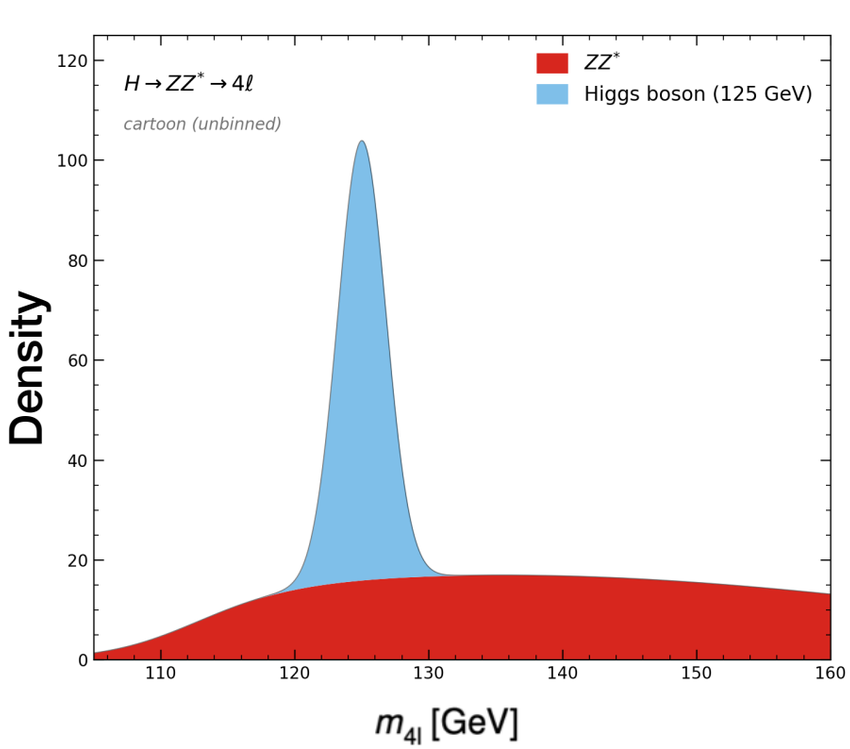
Traditional Inference

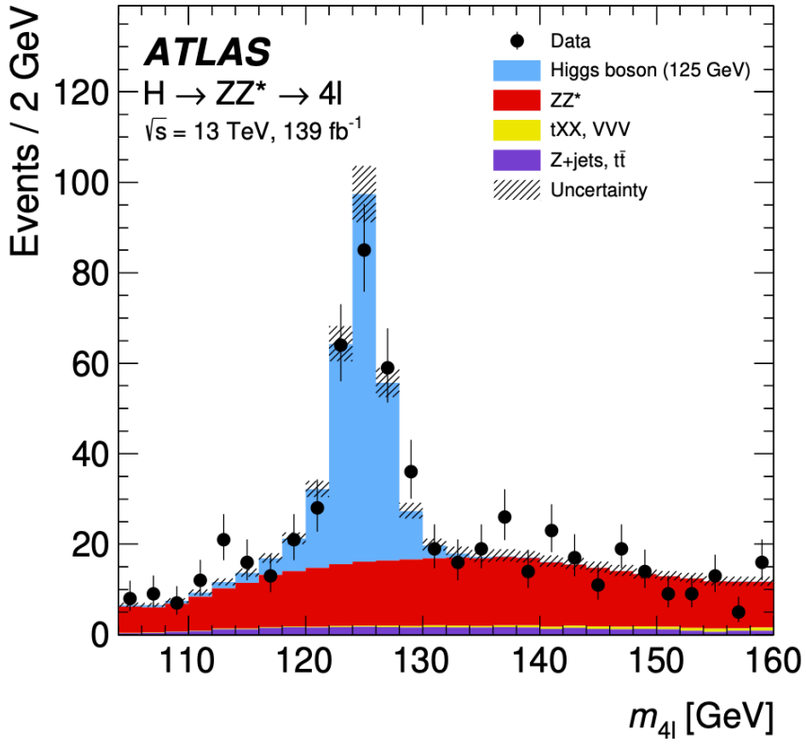
Continuous truth

$$p\left(s(x_e)\,|\,\mu,\nu\right) = \frac{1}{\sigma(\mu,\nu)}\frac{d\sigma}{ds}(s;\mu,\nu)$$

Binned model for density estimation

$$q_{\text{hist}}\left(s(x_e)\,|\,\mu,\nu\right) = \frac{1}{\lambda(\mu,\nu)}\frac{\lambda_b(\mu,\nu)}{\Delta_b}$$





The "reduced" model then uses <u>histograms as density estimators:</u>

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e\in\text{events}} q_{\text{hist}}\left(s(x_e)\,|\,\mu,\nu\right)\right] \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m\,|\,\nu_m)$$

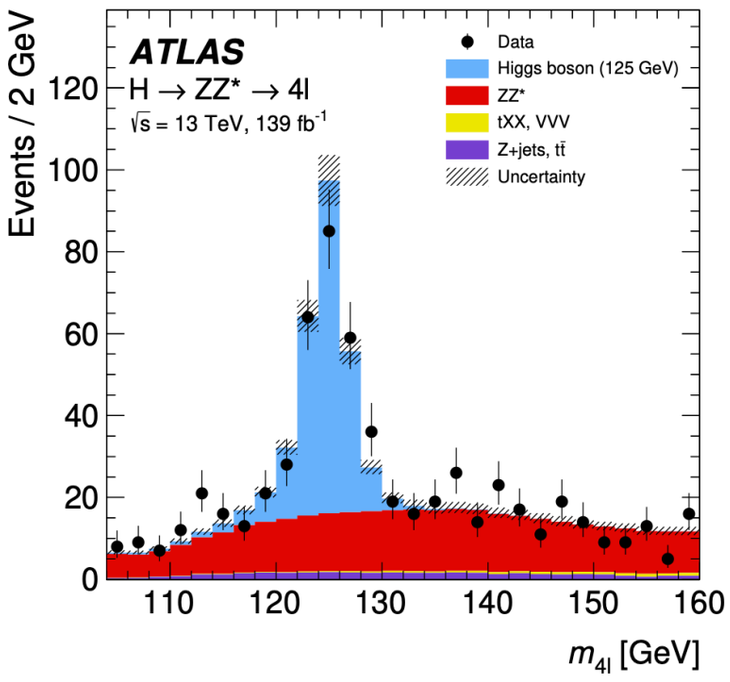
Traditional Inference

The "reduced" model then uses histograms as density estimators:

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e\in\text{events}} q_{\text{hist}}\left(s(x_e)\,|\,\mu,\nu\right)\right] \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m\,|\,\nu_m)$$

Equivalent to the product of binned Poisson likelihoods

$$L(\mu,\nu) = \text{const} \cdot \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$



Per-bin observed event count

Per-bin expected event count

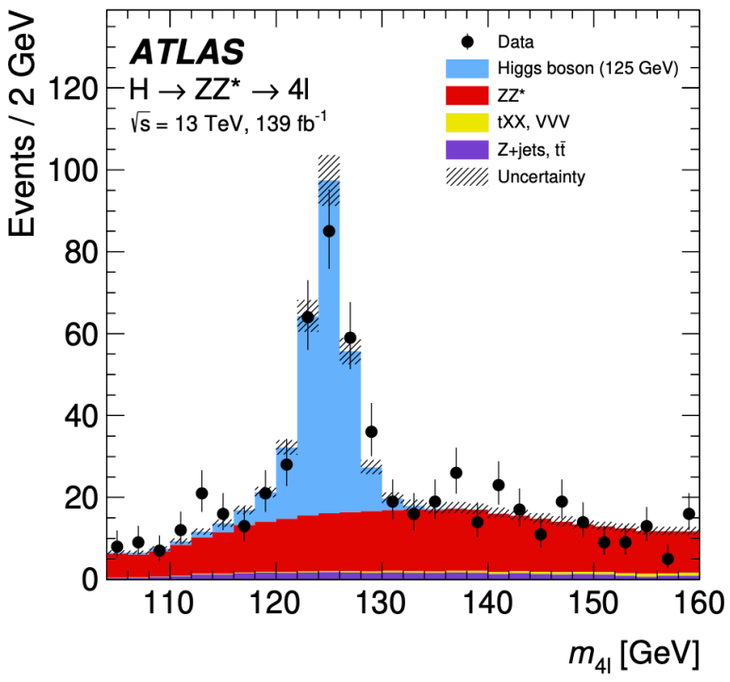
Traditional Inference

The "reduced" model then uses histograms as density estimators:

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e\in\text{events}} q_{\text{hist}}\left(s(x_e)\,|\,\mu,\nu\right)\right] \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m\,|\,\nu_m)$$

Equivalent to the product of binned Poisson likelihoods

$$L(\mu,\nu) = \text{const} \cdot \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$



Frequentist test statistic

$$t_\mu = -2\ln\frac{L(\mu,\hat{\hat{\nu}})}{L(\hat{\mu},\hat{\nu})}$$

Allows us to perform parameter inference

Traditional Inference

The expected rate needs to be modeled continuously as a function of parameters

$$L(\mu,\nu) = \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

Need to build a continuous model as a function of
high-dimensional parameter space

What we have:

expected event counts λ_b at discrete values of high-dimensional parameter space

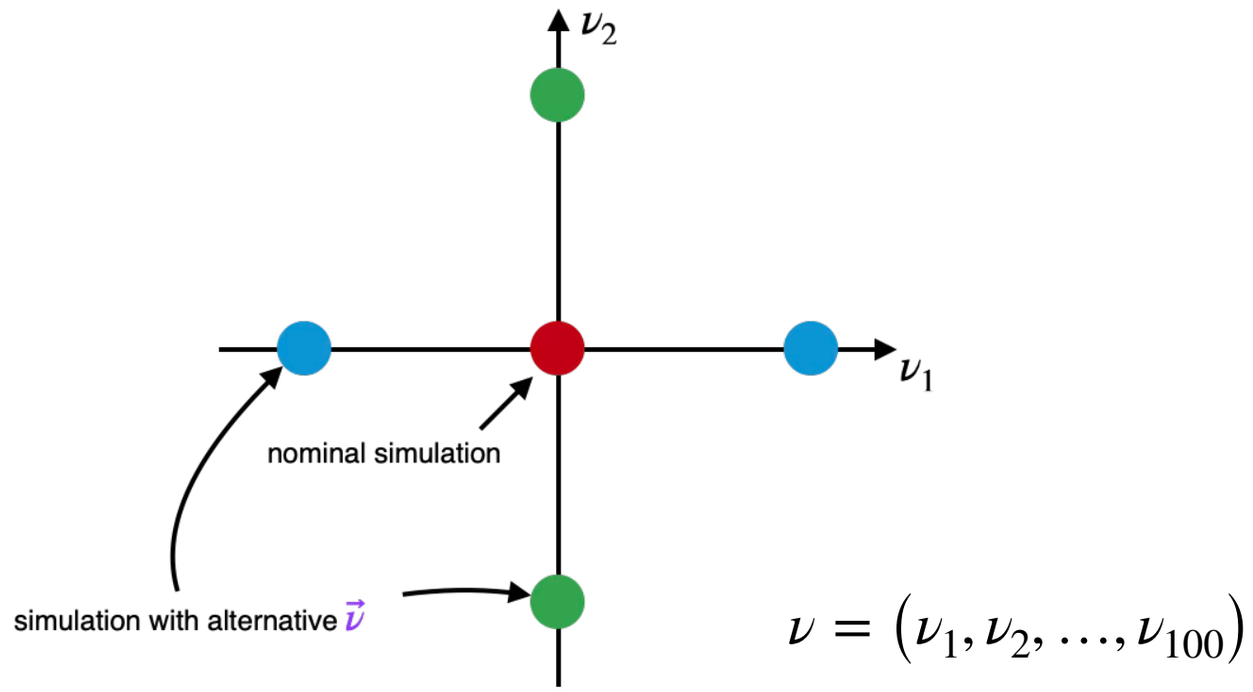


Figure credit: Alex Held

Traditional Inference

The expected rate needs to be modeled continuously as a function of parameters

$$L(\mu,\nu) = \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

Need to build a continuous model as a function of
high-dimensional parameter space

What we have:

expected event counts λ_b at discrete values of high-dimensional parameter space

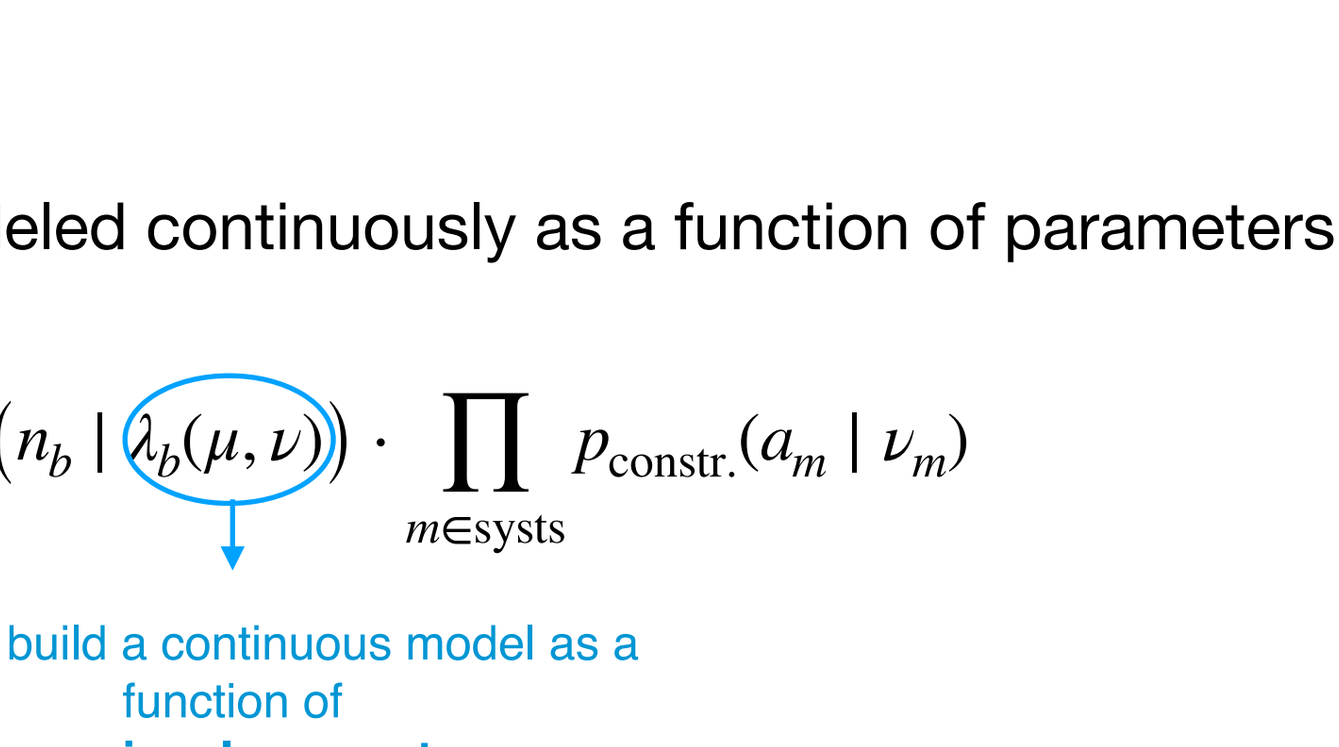


Figure credit: Alex Held

Traditional Inference

The expected rate needs to be modeled continuously as a function of parameters

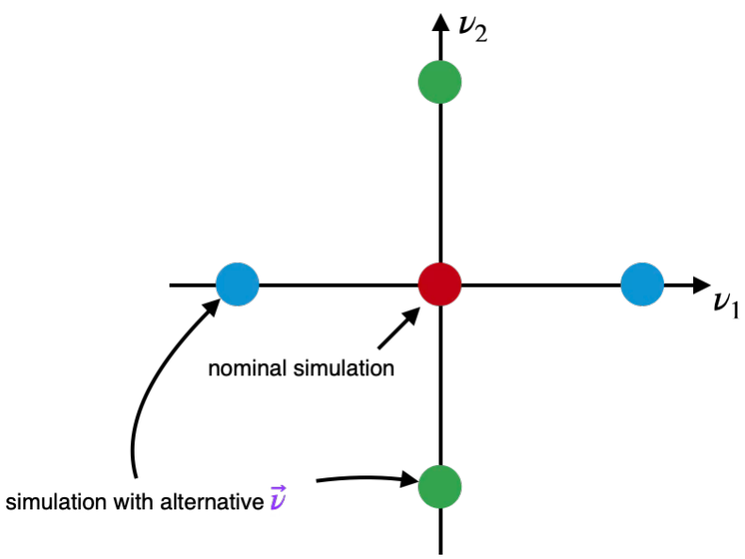
$$L(\mu,\nu) = \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

$$\sum_{s\in\text{samples}} \kappa_s(\mu) \cdot \prod_{m} \lambda_{s,b}(\nu_m)$$

"HistFactory" parametric model for continuous bin morphing:

factorised response and parametric interpolation

$$\nu = (\nu_1, \nu_2, \ldots, \nu_{100})$$



Assume impacts from each ν_m are independent

$\kappa_s(\mu) \rightarrow$ known factorisable, analytic dependence

$$\lambda_{s,b}(\nu) = \prod_m \lambda_{s,b}(\nu_m)$$

Figure credit: Alex Held

Traditional Inference

The expected rate needs to be modeled continuously as a function of parameters

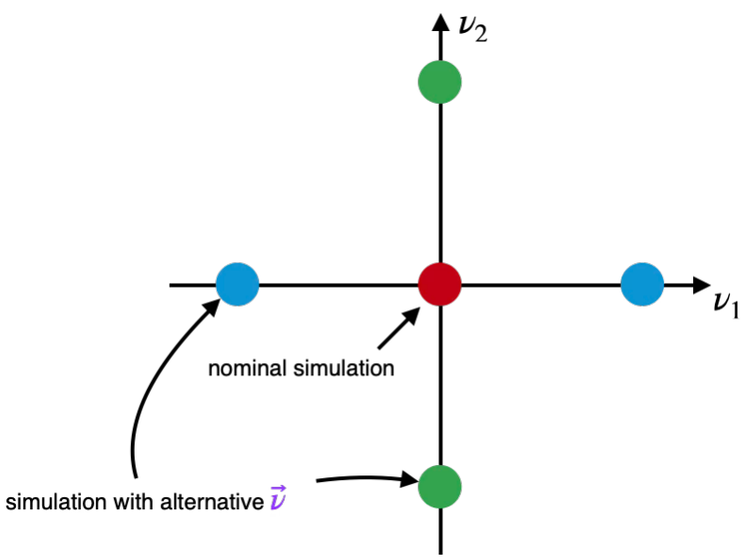
$$L(\mu,\nu) = \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

$$\sum_{s\in\text{samples}} \kappa_s(\mu) \cdot \prod_{m} \lambda_{s,b}(\nu_m)$$

"HistFactory" parametric model for continuous bin morphing:

factorised response and parametric interpolation

$$\nu = (\nu_1, \nu_2, \ldots, \nu_{100})$$



Assume impacts from each ν_m are independent

$\kappa_s(\mu) \rightarrow$ known factorisable, analytic dependence

$$\lambda_{s,b}(\nu) = \prod_{m} \lambda_{s,b}(\nu_m)$$

Remaining goal: estimate $\lambda_{s,b}(\nu_m)$ from sparse simulations

$$\nu_m = -1, 0, 1$$

Figure credit: Alex Held

Traditional Inference

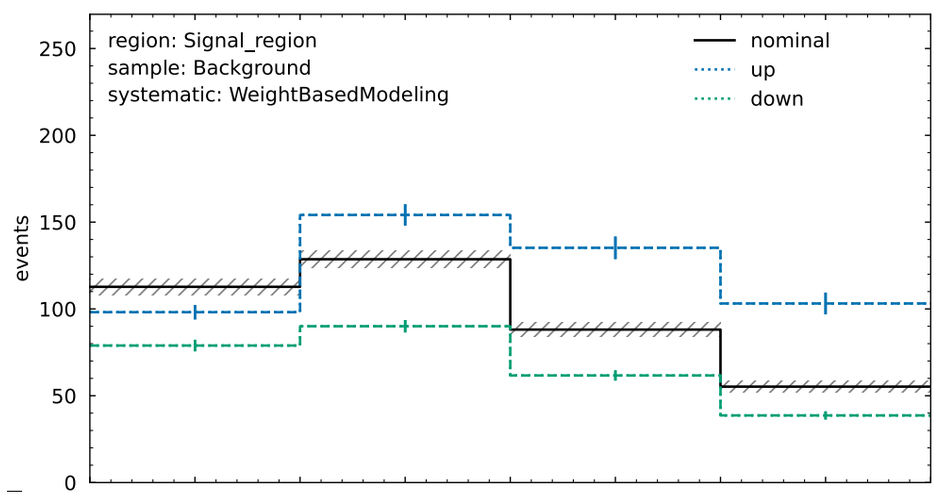
The expected rate needs to be modeled continuously as a function of parameters

$$L(\mu,\nu) = \prod_{b\in\text{bins}} \text{Pois}\left(n_b \mid \lambda_b(\mu,\nu)\right) \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

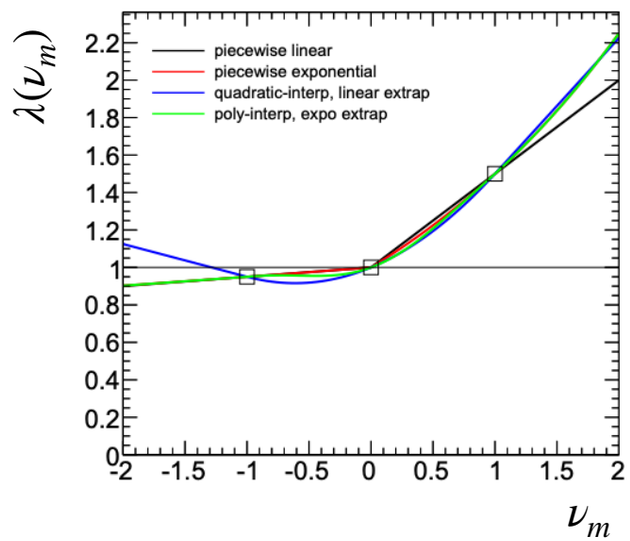
$$\sum_{s\in\text{samples}} \kappa_s(\mu) \cdot \prod_m \lambda_{s,b}(\nu_m)$$

"HistFactory" parametric model for continuous bin morphing:

factorised response and parametric interpolation

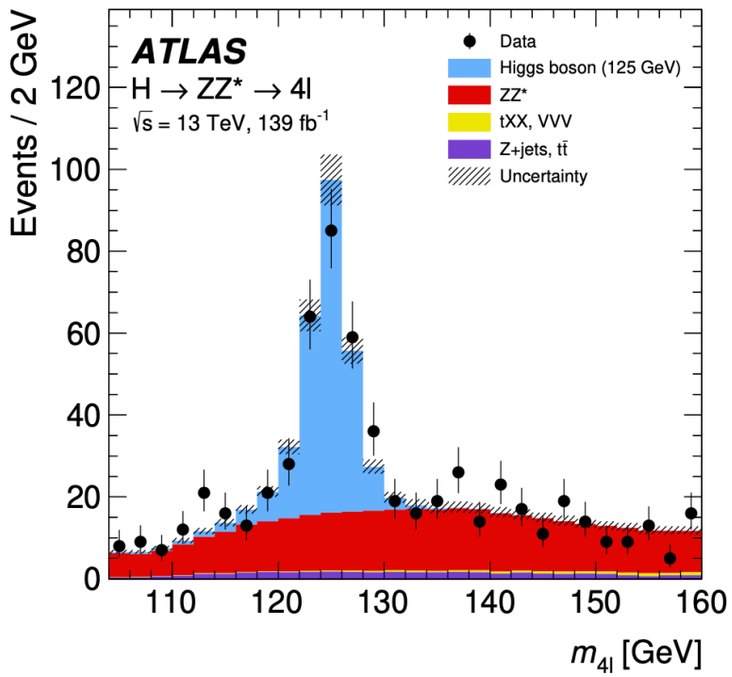


toy example: templates for each $\nu_m = -1, 0, +1$



parametric fit between templates per-bin and per ν_m

Figure credit: Alex Held



Summary-based Histogram SBI

Pros: easy, cheap and asymptotically unbiased density estimation

Cons: lose sensitive information from high-dimensional data

Why use neural SBI?

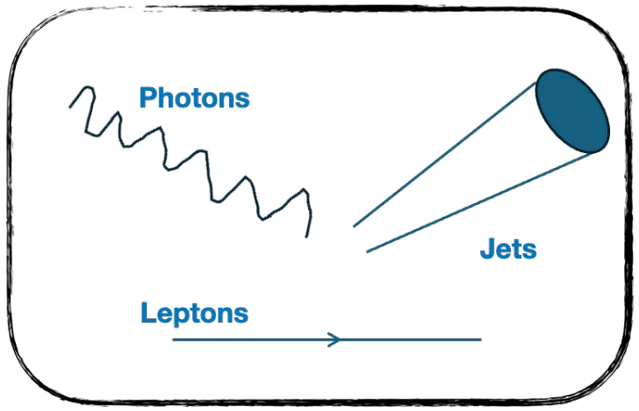
What if we can exploit the fully differential high-dimensional data to perform statistical inference?

High-dimensional Inference

The full likelihood model in an ATLAS or CMS analysis looks like this:

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e \in \text{events}} p(x_e \mid \mu,\nu) \right] \cdot \prod_{m \in \text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

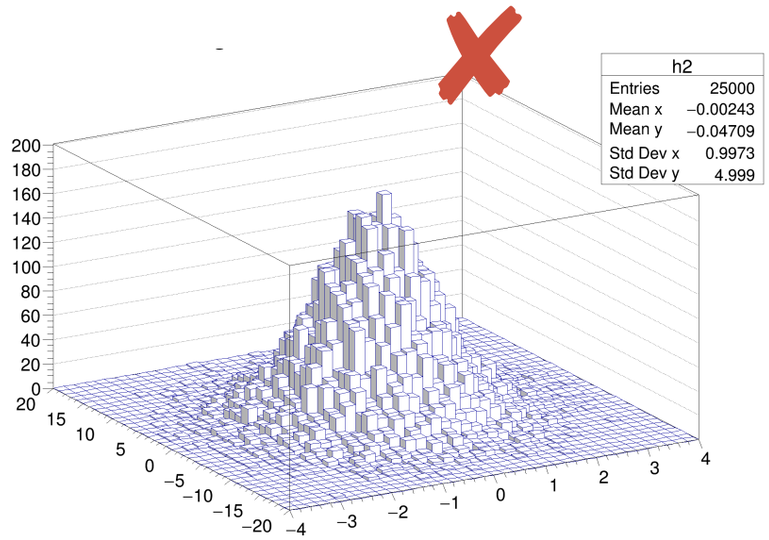
Estimation goal:
per i.i.d. event density function



Physics objects & their momentum
~10-100 dimensional vector

$$x \sim p\left(x \mid \mu,\nu\right)$$

Simulated high-dimensional data



$$q_{\text{hist}}\left(x \mid \mu,\nu\right) \approx p(x \mid \mu,\nu)$$

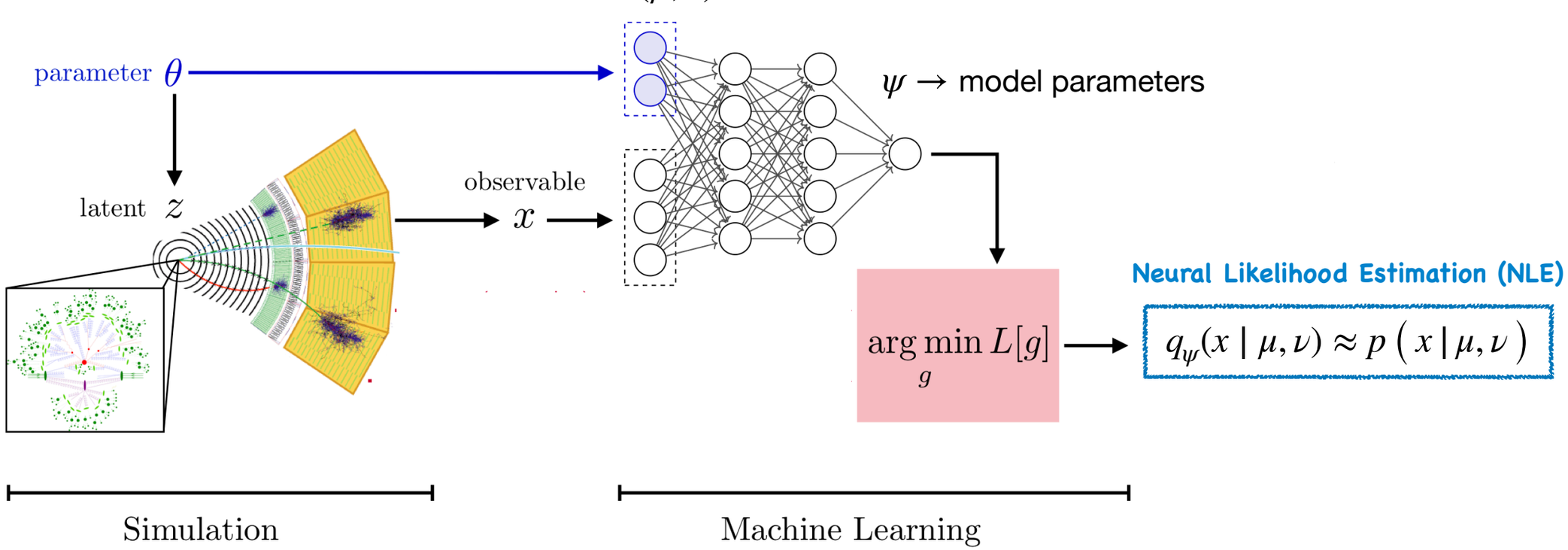
Curse of dimensionality - need more simulations to model x than feasible

Neural Simulation-Based Inference

The full likelihood model in an ATLAS or CMS analysis looks like this:

$$L(\mu,\nu) = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e \in \text{events}} p(x_e \mid \mu,\nu) \right] \cdot \prod_{m \in \text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

MadMiner
[1805.00020, 1805.00013]



neural SBI in a nutshell: use deep learning surrogates for the modeling of probability density

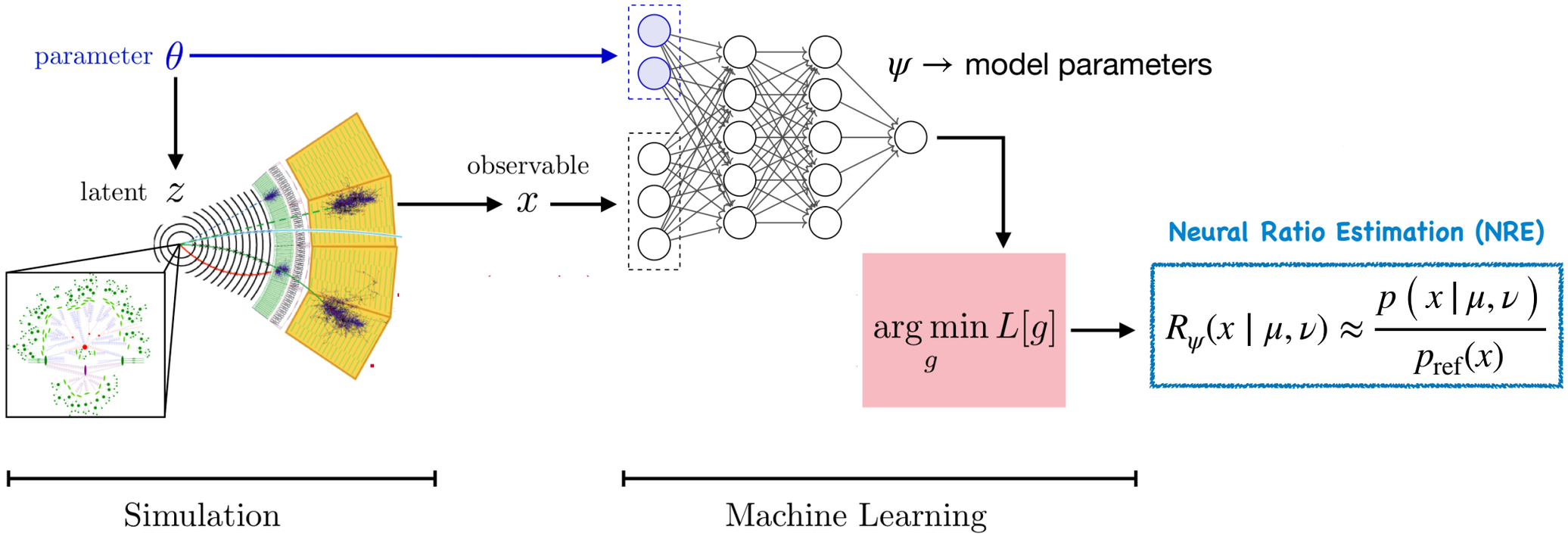
Neural Simulation-Based Inference

The full likelihood model in an ATLAS or CMS analysis looks like this:

$$\frac{L(\mu,\nu)}{\prod_e p_{\text{ref}}(x)} = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e \in \text{events}} \frac{p(x_e \mid \mu,\nu)}{p_{\text{ref}}(x)} \right] \cdot \prod_{m \in \text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

$$\theta = (\mu, \nu)$$

MadMiner
[1805.00020,
1805.00013]



neural SBI in a nutshell: use deep learning surrogates for the modeling of probability density (ratios)

Neural Simulation-Based Inference

For a frequentist test, the parameter-independent p_{ref} can be arbitrarily chosen

$$\frac{L(\mu,\nu)}{\prod_e p_{\text{ref}}(x_e)} = \left[\text{Pois}(n \mid \lambda(\mu,\nu)) \prod_{e\in\text{events}} \frac{p(x_e \mid \mu,\nu)}{p_{\text{ref}}(x_e)}\right] \cdot \prod_{m\in\text{systs}} p_{\text{constr.}}(a_m \mid \nu_m)$$

Density ratios can be estimated using well-trained classification models.

Easier than density modelling

Test statistic remains unaffected

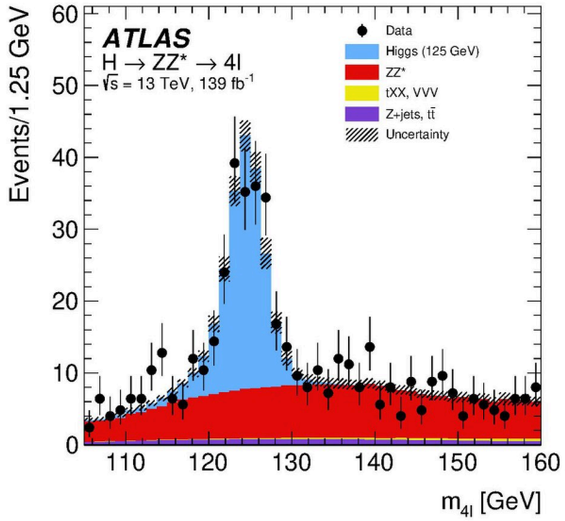
$$t_\mu = -\ln \frac{L(\mu,\hat{\hat{\nu}})\,/\,\cancel{p_{\text{ref}}(x_e)}}{L(\hat{\mu},\hat{\nu})\,/\,\cancel{p_{\text{ref}}(x_e)}}$$

p_{ref} can be chosen arbitrarily
eventually cancels out

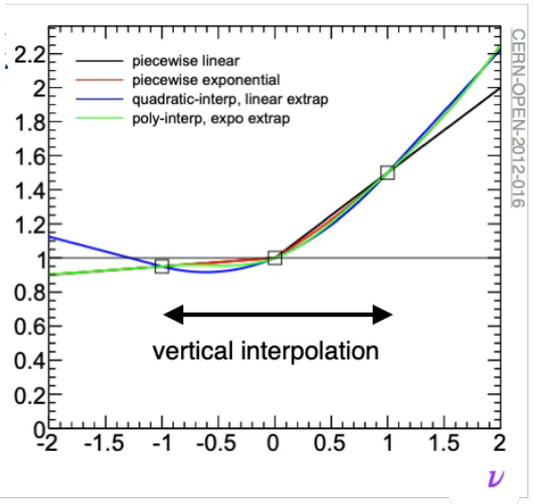
The test retains all the desirable asymptotic properties that come with profile negative likelihood ratio test [1007.1727]

SBI Landscape

Traditional SBI

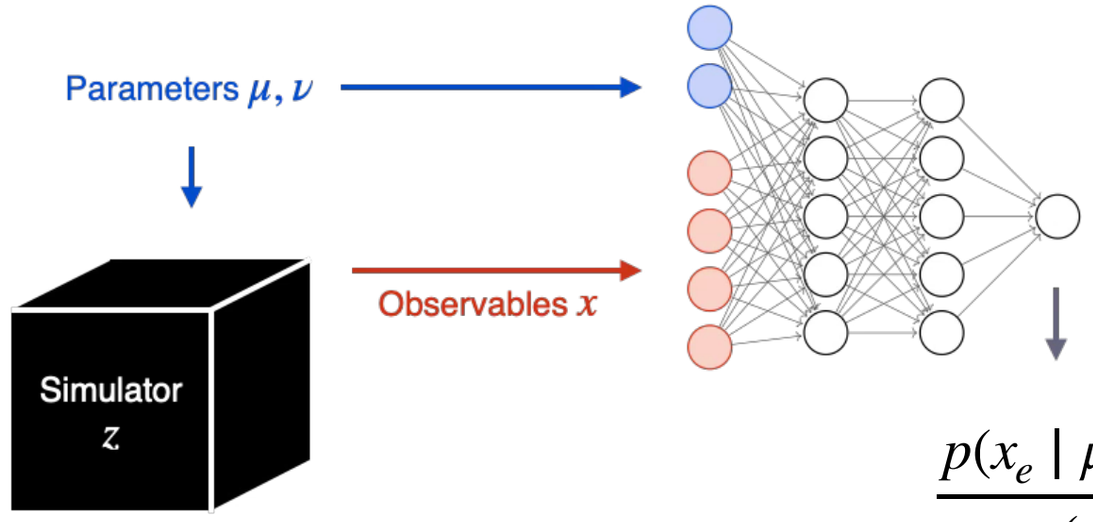


interpolation in one bin



parametric = simple analytical dependency

NSBI



$$\frac{p(x_e \mid \mu, \nu)}{p_{\text{ref}}(x_e)}$$

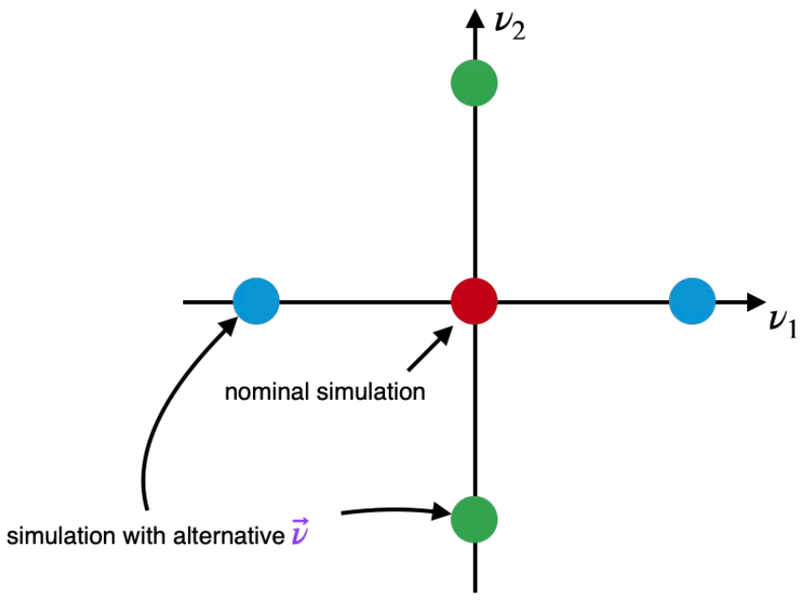
non-parametric = NNs, Gaussian Processes, etc.

	x	μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric

Can we apply NSBI to LHC analysis?

Sparse simulations

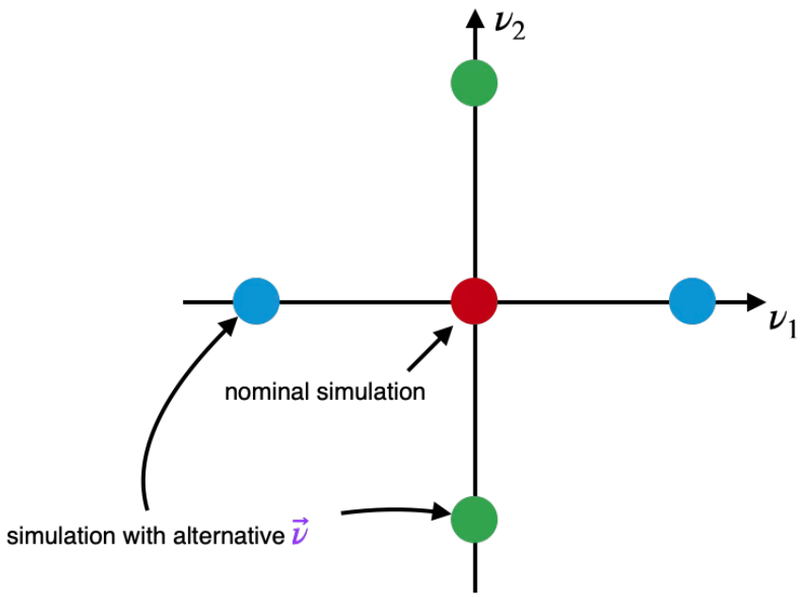
$$\nu = (\nu_1, \nu_2, \ldots, \nu_{100})$$



Challenge: Only sparsely populated parameter space in 100-1000 dimensions for event samples in ATLAS and CMS.

Sparse simulations

$$\nu = (\nu_1, \nu_2, \ldots, \nu_{100})$$



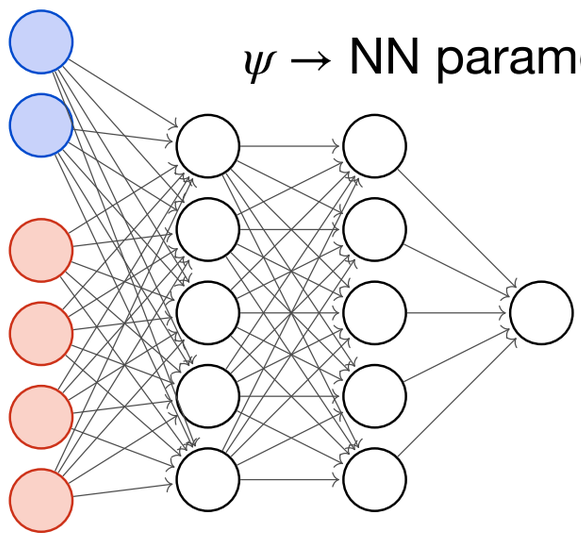
Challenge: Only sparsely populated parameter space in 100-1000 dimensions for event samples in ATLAS and CMS.

Neural Networks would not know how to model parameter space where there are no simulations

$\mu, \nu \sim \pi(\mu, \nu)$

$x \sim p(x \mid \mu, \nu)$

$x \sim p_{\text{ref}}(x)$



$$R_\psi(x; \mu, \nu) \approx \frac{p(x_e \mid \mu, \nu)}{p_{\text{ref}}(x_e)}$$

Need to sample continuously and densely from some $\pi(\mu, \nu)$ to model the ratios

For technical details, see e.g. MadMiner [1805.00020, 1805.00013]

Factorised Approach

The HistFactory model for per-event density ratio:

Differential cross-section term (high-dimensional inference)

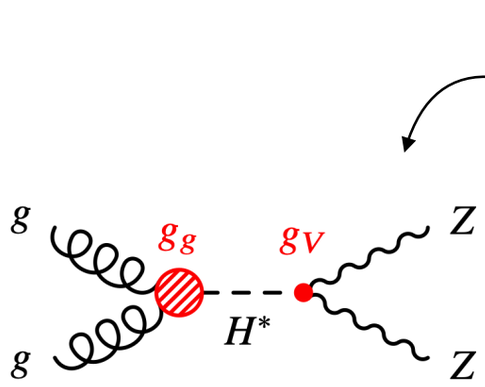
$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \frac{p_s\left(x_e \mid \mu, \nu\right)}{p_{ref}(x_e)}$$

Total yields (computed via HistFactory)

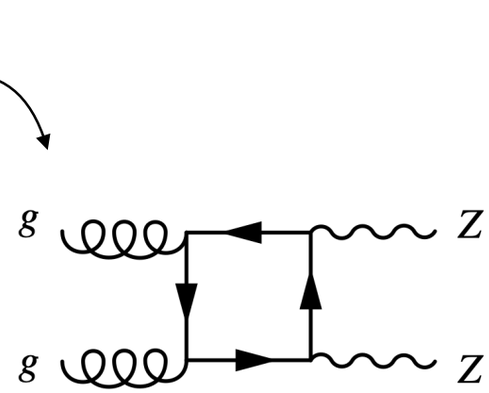
Factorised Approach

The HistFactory model for per-event density ratio:

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s\in\text{samples}} \lambda_s(\mu,\nu)} \boxed{\sum_{s\in\text{samples}}} \lambda_s(\mu,\nu) \cdot \frac{p_s\left(x_e \mid \mu, \nu\right)}{p_{ref}(x_e)}$$



Signal to measure $p_s = p_{\text{Sig}}$



Background with same final state $p_s = p_{\text{Bkg}}$

E.g.: "signal strength"parameter

$$p(x \mid \mu) = \frac{1}{\mu \cdot \lambda_{\text{Sig}} + \lambda_{\text{Bkg}}} \left[\mu \cdot \lambda_{\text{Sig}} \cdot p_{\text{Sig}}(x) + \lambda_{\text{Bkg}} \cdot p_{\text{Bkg}}(x)\right]$$

Factorised Approach

The HistFactory model for per-event density ratio:

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \frac{p_s\left(x_e \mid \mu, \nu\right)}{p_{ref}(x_e)}$$

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

Known dependencies (μ-dependence)

Systematic uncertainty reweight factor (x, ν-dependence)

Nominal Density ratio (x-dependence)

Goal: "factorised" targets to tackle independently

Factorised Approach

The HistFactory model for per-event density ratio:

Systematic uncertainty reweight factor

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

Known dependencies

Nominal density ratio (parameter-independent)

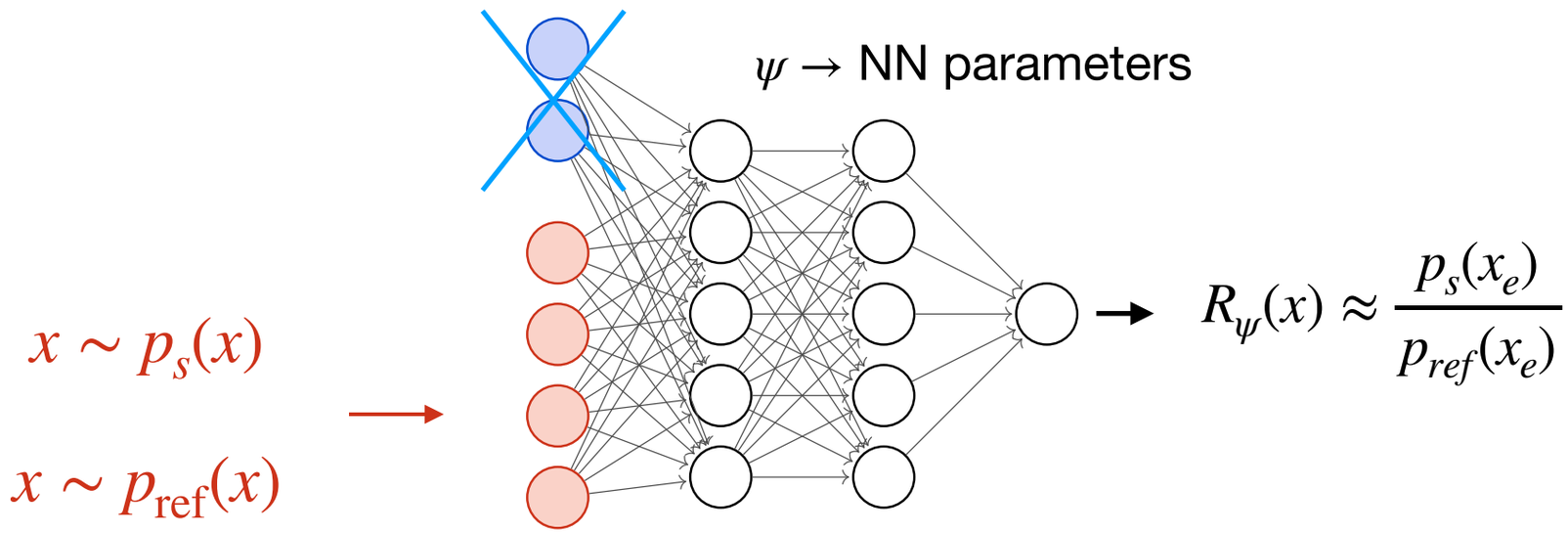
Factorised Approach

The HistFactory model for per-event density ratio:

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

Nominal density ratio (parameter-independent)

Parameter-independent model
no dense parameter sampling needed!



Semi-parametric SBI

The HistFactory model for per-event density ratio:

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

$$\approx \prod_{m}^{N_{syst}} \left[\frac{p_s(x_e \mid \nu_m)}{p_s(x_e)}\right]$$

HistFactory-style model

$$\nu = \left(\nu_1, \nu_2, \ldots, \nu_{N_{\text{syst}}}\right)$$

Assumption 1:

impact from independent sources of systematic uncertainties factorise

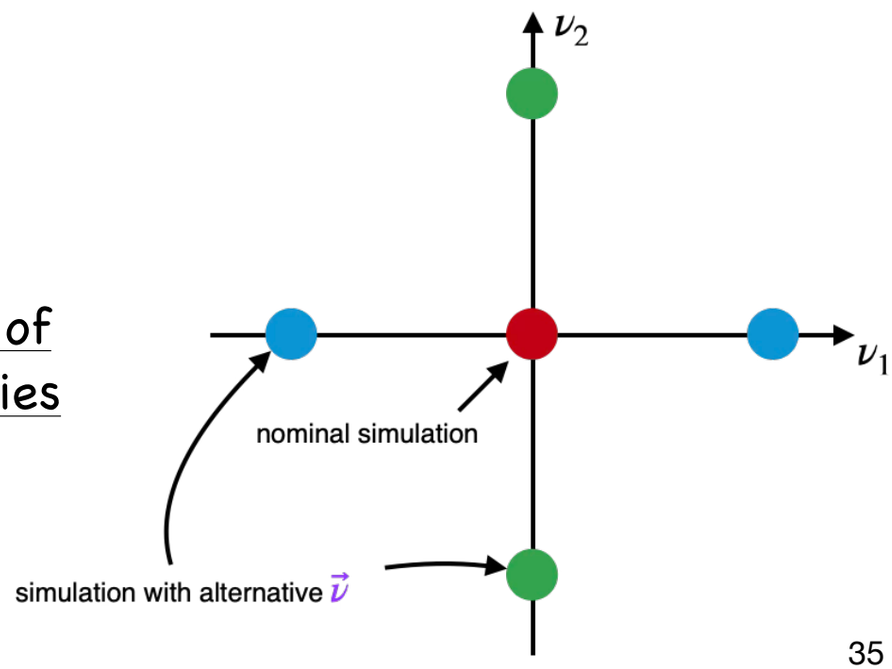


Figure credit: Alex Held

Semi-parametric SBI

The HistFactory model for per-event density ratio:

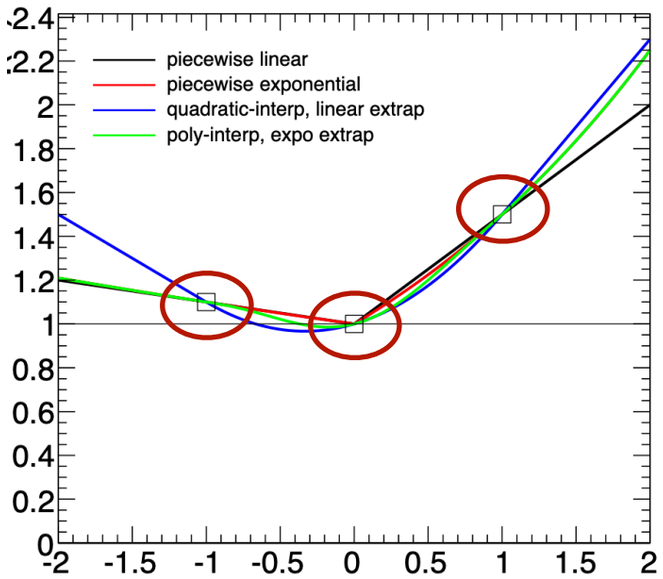
$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu,\nu)} \sum_{s \in \text{samples}} \lambda_s(\mu,\nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

$$\approx \prod_{m}^{N_{syst}} \text{Interp}\left[\frac{p_s(x_e \mid \nu_m = +1)}{p_s(x_e)}, 1, \frac{p_s(x_e \mid \nu_m = -1)}{p_s(x_e)}\right]$$

HistFactory-style model

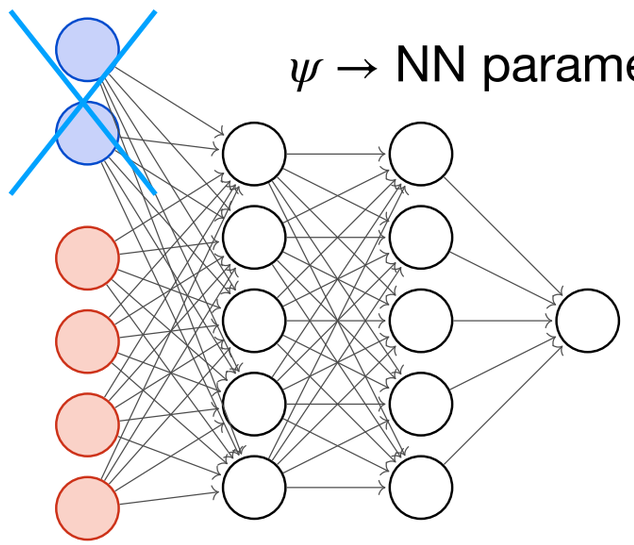
$$\frac{p_s(x_{ce} \mid \nu_m)}{p_s(x_{ce})}$$

Interp [.]



Assumption 2:

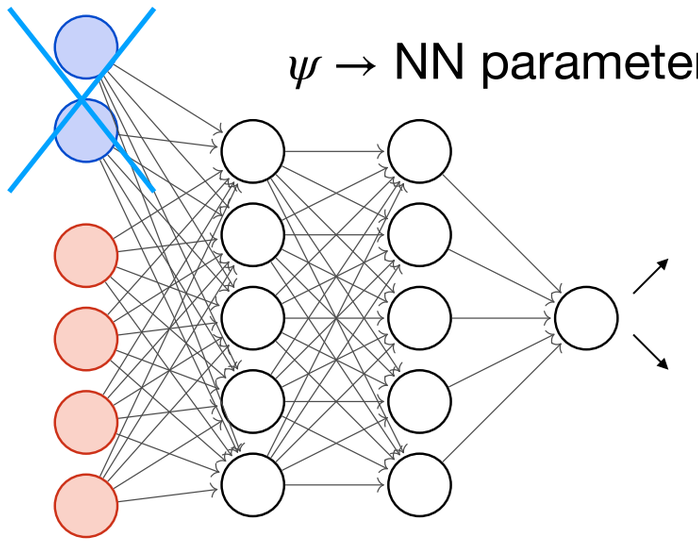
semi-parametric model for parameterised density ratios



$$R_\psi \approx \frac{p_s(x_e \mid \nu_m = \pm 1)}{p_s(x_e)}$$

Train two NNs per nuisance-parameter

SBI Landscape

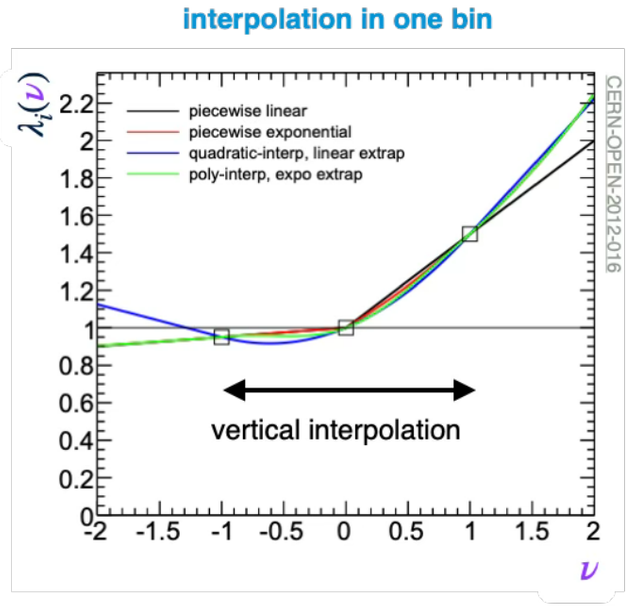


$\psi \rightarrow$ NN parameters

$$\frac{p_s(x_e \,|\, \nu_m = \pm 1)}{p_s(x_e)}$$

$$\frac{p_s(x_e)}{p_{\text{ref}}(x_e)}$$

+



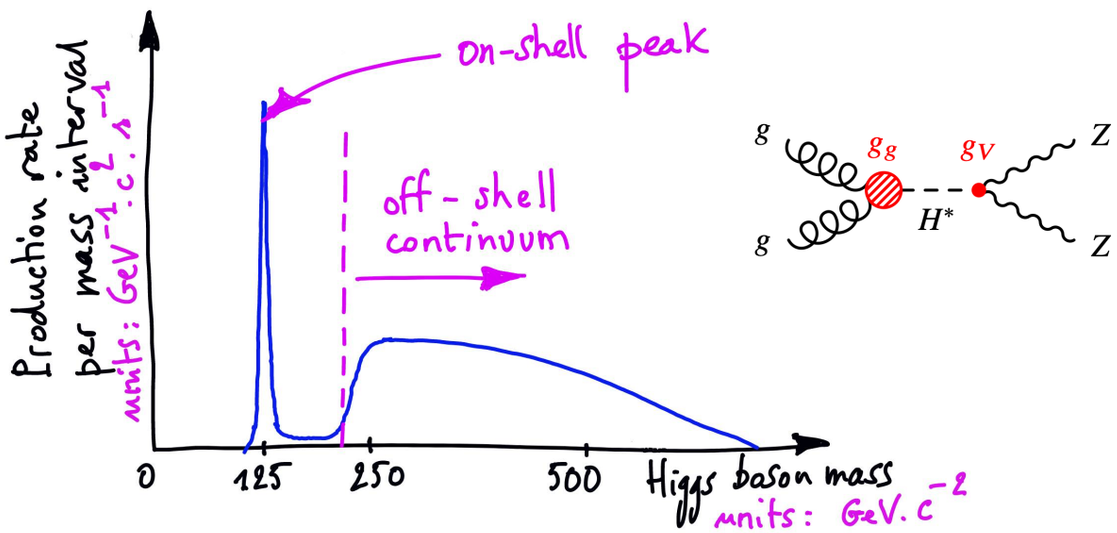
non-parametric models for modeling x

parametric ansatz for μ, ν-dependence

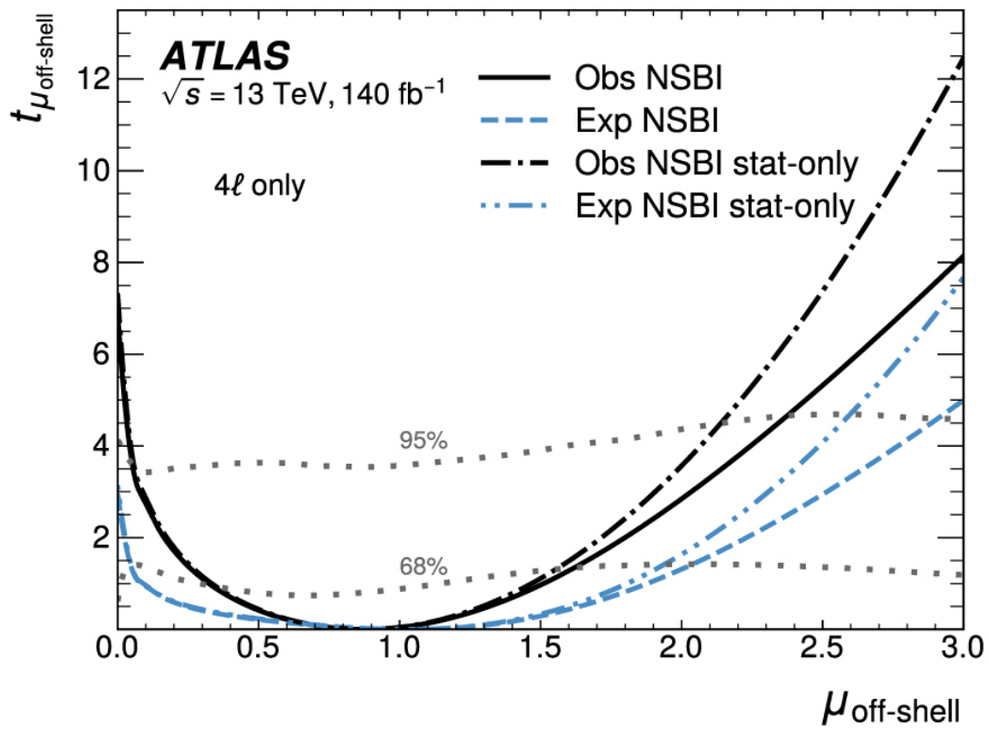
Ongoing effort mapping out the SBI landscape more generally (Cranmer & Sandesara)

	x	μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric
Semi-parametric SBI (POI&Nuisance)	non-parametric	parametric	parametric

Semi-parametric SBI



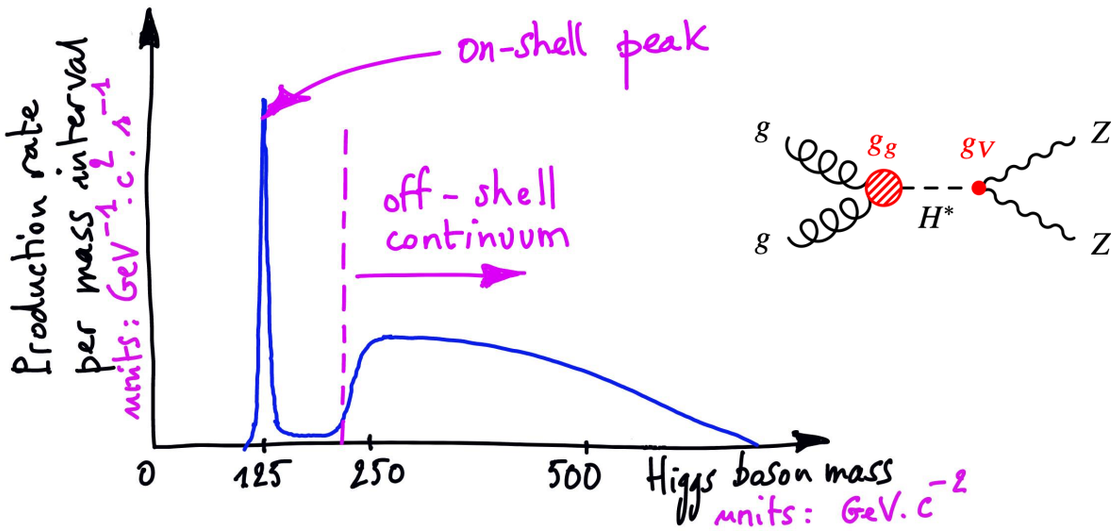
Rep. Prog. Phys. 88 057803

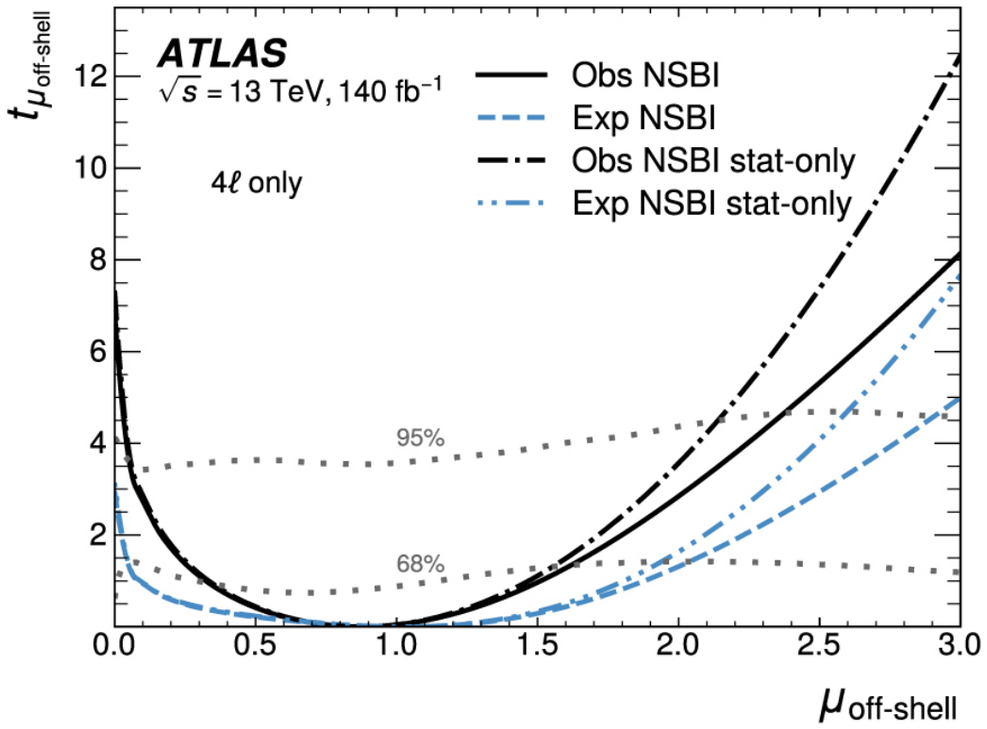


First ever measurement with real data using fully-differential high-dimensional information

	x	μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric
Semi-parametric SBI (POI&Nuisance)	non-parametric	parametric	parametric

Semi-parametric SBI





First ever measurement with real data using fully-differential high-dimensional information

Rep. Prog. Phys. 88 067801

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Rep. Prog. Phys. 88 (2025) 067801
DOI: 10.1088/1361-6633/add370

CERN-EP-2024-305
June 16, 2025

An implementation of neural simulation-based inference for parameter estimation in ATLAS

The ATLAS Collaboration

Rep. Prog. Phys. 88 057803

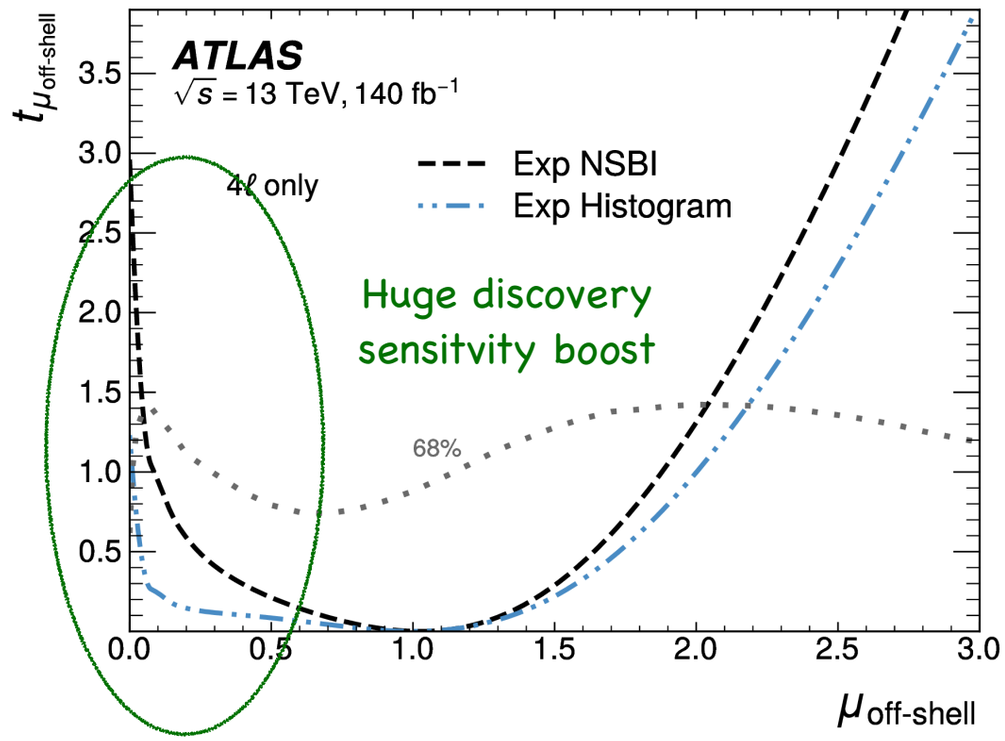
EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Rep. Prog. Phys. 88 (2025) 057803
DOI: DOI:10.1088/1361-6633/adcd9a

CERN-EP-2024-298
May 23, 2025

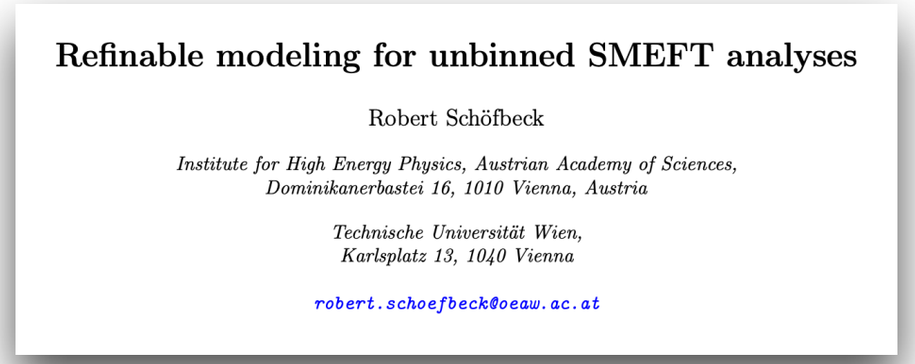
Measurement of off-shell Higgs boson production in the $H^* \to ZZ \to 4\ell$ decay channel using a neural simulation-based inference technique in 13 TeV pp collisions with the ATLAS detector

The ATLAS Collaboration



Both curves use same data, Histogram uses NN observable for $x \to s(x)$

Semi-parametric SBI



Refinable modeling for unbinned SMEFT analyses

Robert Schöfbeck

Institute for High Energy Physics, Austrian Academy of Sciences, Dominikanerbastei 16, 1010 Vienna, Austria

Technische Universität Wien, Karlsplatz 13, 1040 Vienna

robert.schoefbeck@oeaw.ac.at

Similar semi-parametric idea

Different implementation strategy

Nuisance parameter dependent ratio:

$$\exp\left(\underbrace{\nu_A \hat{\Delta}_A(\boldsymbol{x})}\right) \simeq \frac{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{\mathrm{d}\sigma(\boldsymbol{x}|\boldsymbol{0})}$$

Parametric ansatz

$\hat{\Delta}(x) \rightarrow$ NN outputs

Each $x-$dependent term is then fitted by minimizing the loss:

$$L[\hat{\Delta}_A(\boldsymbol{x})] = \sum_{\nu\in\mathcal{V}} L_{\mathrm{CE}}[\nu_A \hat{\Delta}_A(\boldsymbol{x})]$$

$$= \sum_{\nu\in\mathcal{V}} \left(\left\langle \mathrm{Soft}^+(\nu_A \hat{\Delta}_A(\boldsymbol{x})) \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{0}} + \left\langle \mathrm{Soft}^+(-\nu_A \hat{\Delta}_A(\boldsymbol{x})) \right\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}} \right)$$

	x	μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric
Semi-parametric SBI (POI&Nuisance)	non-parametric	parametric	parametric

We are now optimally constraining ALL parameters, with high-dimensional information from simulators.

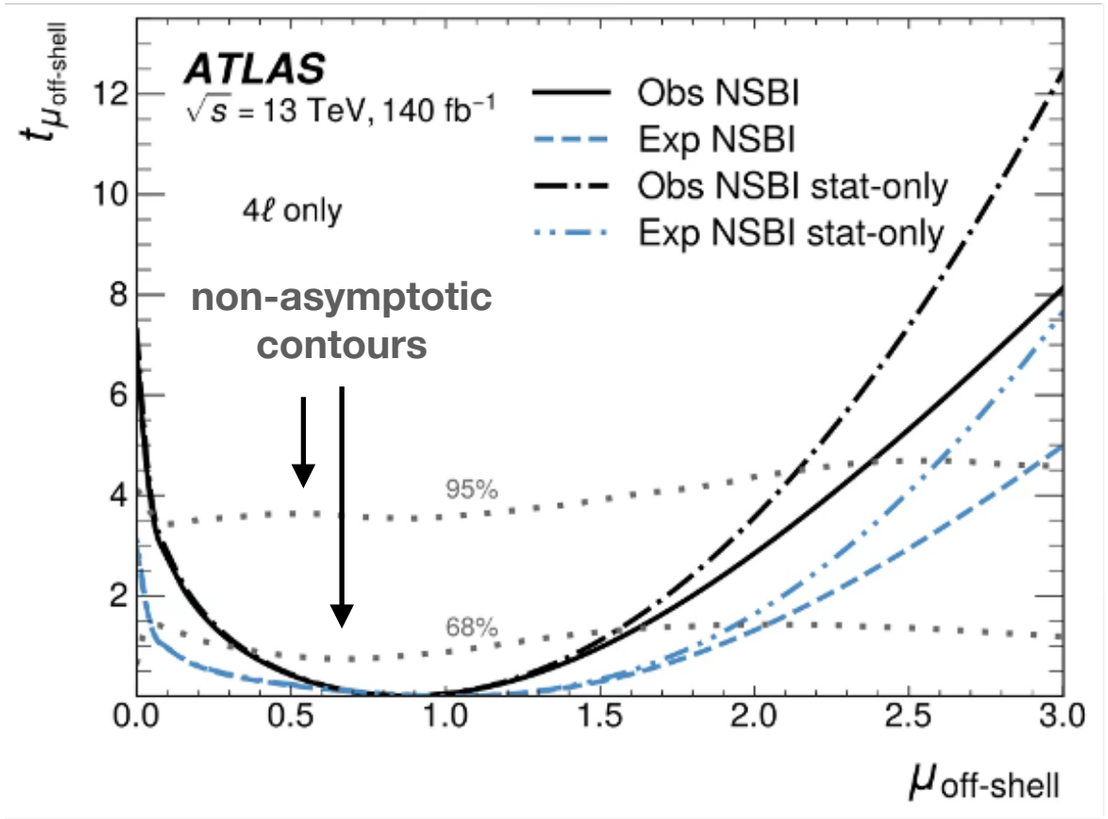
Neyman Construction

What if the neural networks are mis-modelled?

- The resulting test statistic we build does not retain the asymptotic properties typical of profile negative log-likelihood ratio test.

- Neyman Construction with simulator-sampled events more essential than before: build calibrated confidence intervals.

Rep. Prog. Phys. 88 057803



Robust Model Building

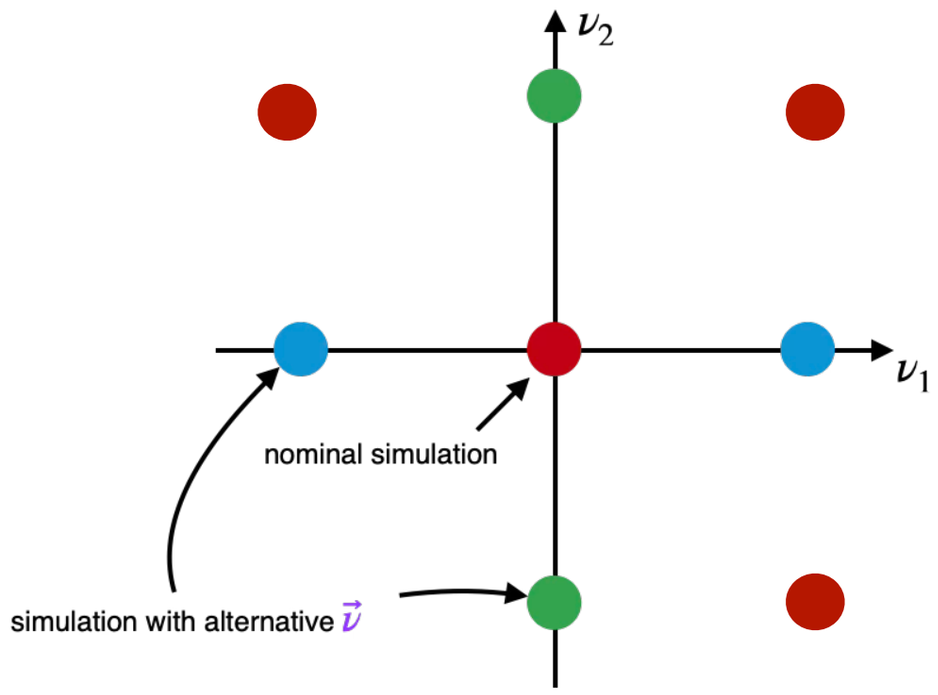
The HistFactory model for per-event density ratio:

$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \boxed{\frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)}} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

Can we do better than the HistFactory assumptions?

Goal:

modeling parameter space dependence more robustly



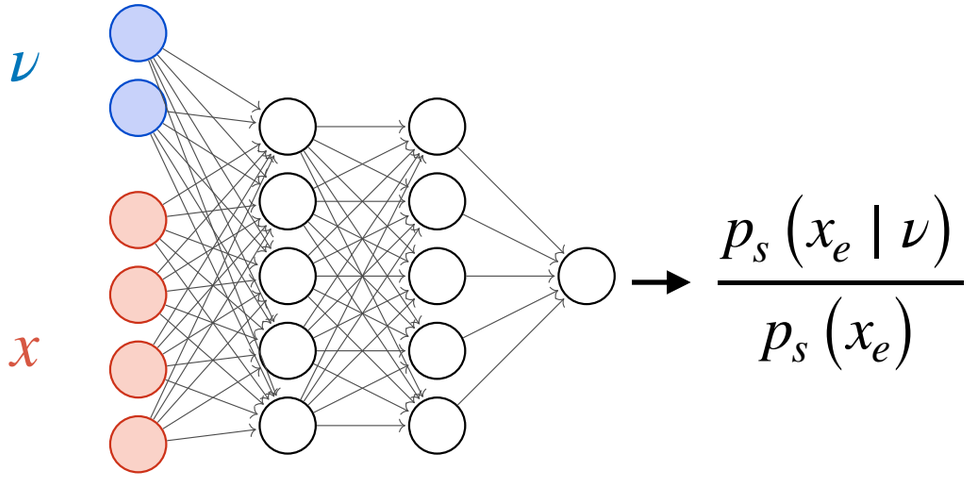
What if we generate simulations at more parameter points?

Robust Model Building

The HistFactory model for per-event density ratio:

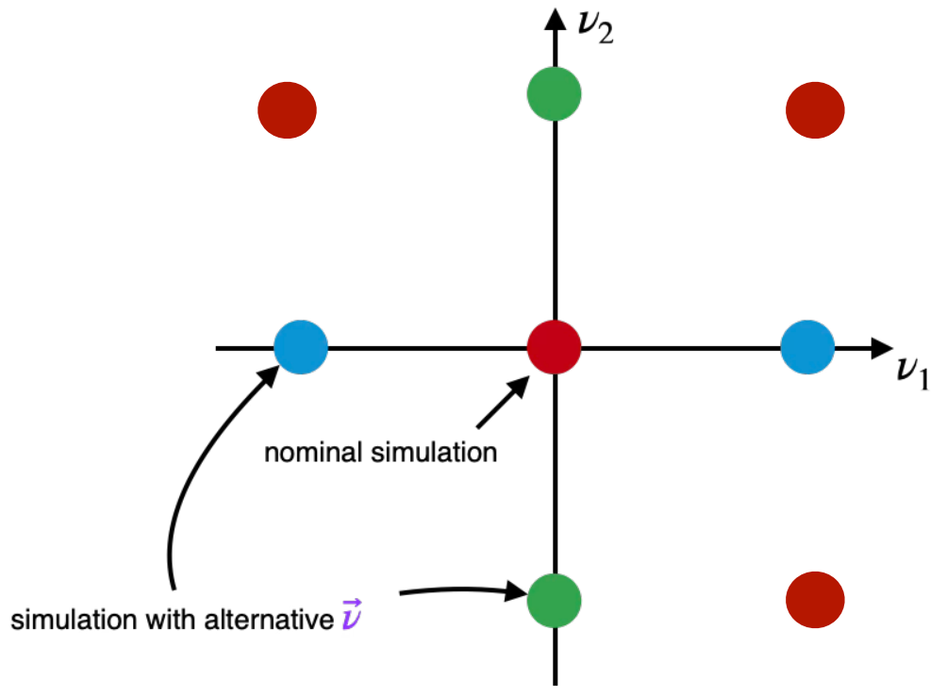
$$\frac{p\left(x_e \mid \mu, \nu\right)}{p_{\text{ref}}(x_e)} = \frac{1}{\sum_{s \in \text{samples}} \lambda_s(\mu, \nu)} \sum_{s \in \text{samples}} \lambda_s(\mu, \nu) \cdot \left[\prod_{\mu} \kappa_s(\mu)\right] \cdot \boxed{\frac{p_s\left(x_e \mid \nu\right)}{p_s\left(x_e\right)}} \cdot \frac{p_s(x_e)}{p_{ref}(x_e)}$$

Can we do better than the HistFactory assumptions?



Option: train parameterized neural network classifiers?

Challenge: hard to regularise



What if we generate simulations at more parameter points?

Robust Model Building

Semi-parametric option:

more robust parametric approach with relaxed factorisation assumptions.

Unbinned inclusive cross-section measurements with machine-learned systematic uncertainties

Lisa Benato, Cristina Giordano, Claudius Krause, Ang Li,
Robert Schöfbeck, Dennis Schwarz, Maryam Shooshtari, and Daohan Wang
Institute for High Energy Physics, Austrian Academy of Sciences Dominikanerbastei 16, 1010 Vienna, Austria
(Dated: August 26, 2025)

Example:

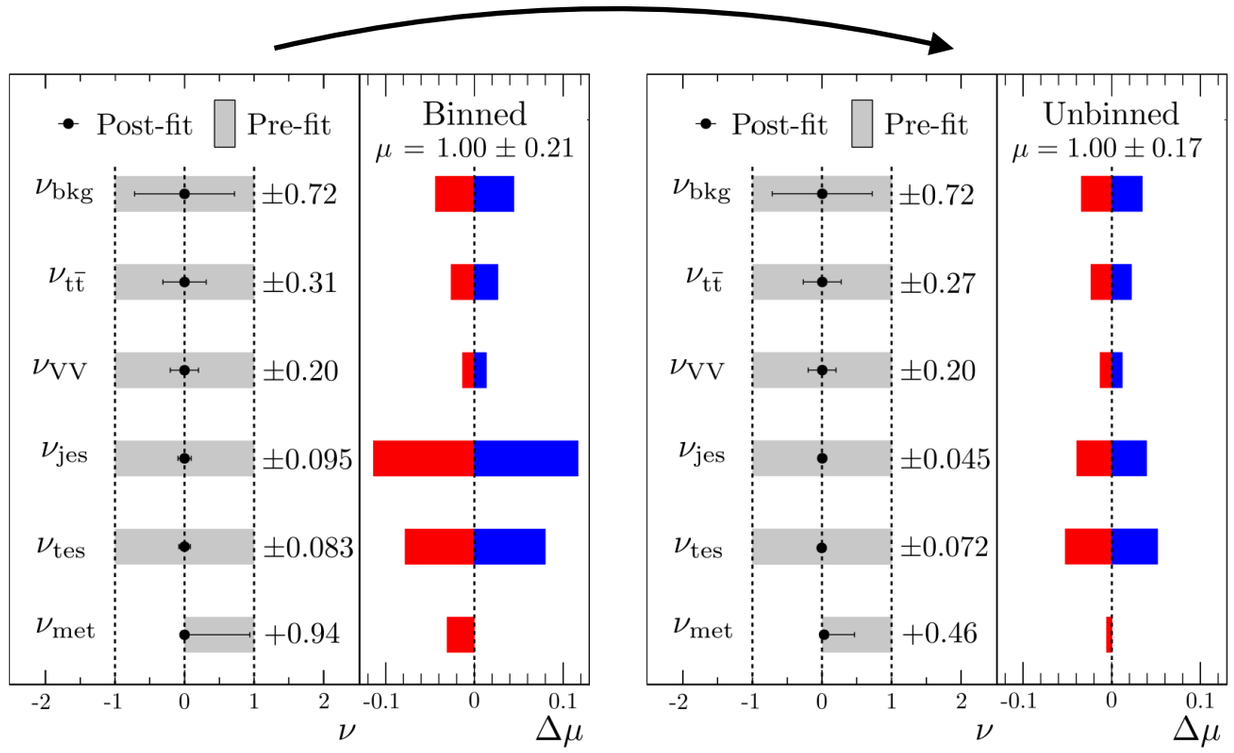
$$\nu_A \hat{\Delta}_A(\boldsymbol{x}) = \nu_{\text{tes}}\,\hat{\Delta}_{\text{tes}}(\boldsymbol{x}) + \nu_{\text{jes}}\,\hat{\Delta}_{\text{jes}}(\boldsymbol{x}) + \nu_{\text{met}}\,\hat{\Delta}_{\text{met}}(\boldsymbol{x})$$
$$+ \nu_{\text{tes}}^2\,\hat{\Delta}_{\text{tes,tes}}(\boldsymbol{x}) + \nu_{\text{jes}}^2\,\hat{\Delta}_{\text{jes,jes}}(\boldsymbol{x}) + \nu_{\text{met}}^2\,\hat{\Delta}_{\text{met,met}}(\boldsymbol{x})$$
$$+ \boxed{\nu_{\text{tes}}\nu_{\text{jes}}\,\hat{\Delta}_{\text{tes,jes}}(\boldsymbol{x})} + \boxed{\nu_{\text{jes}}\nu_{\text{met}}\,\hat{\Delta}_{\text{jes,met}}(\boldsymbol{x})}$$
$$+ \boxed{\nu_{\text{tes}}\nu_{\text{met}}\,\hat{\Delta}_{\text{tes,met}}(\boldsymbol{x})}.$$

Example of simultaneous variations

$\hat{\Delta}(x) \rightarrow$ NN outputs (unparameterized)

The polynomial ansatz can be made arbitrarily higher-order for expressivity

Reduced impact on sensitivity from NPs when fitting with SBI



low-dimensional summary

NSBI likelihood fit

Robust Model Building

For more details see the blueprint workshop talk

	x	μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric
Semi-parametric SBI (POI&Nuisance)	non-parametric	parametric	parametric
Semi-parametric SBI (POI& GP Nuisance)	non-parametric with NN	parametric	GP not NN

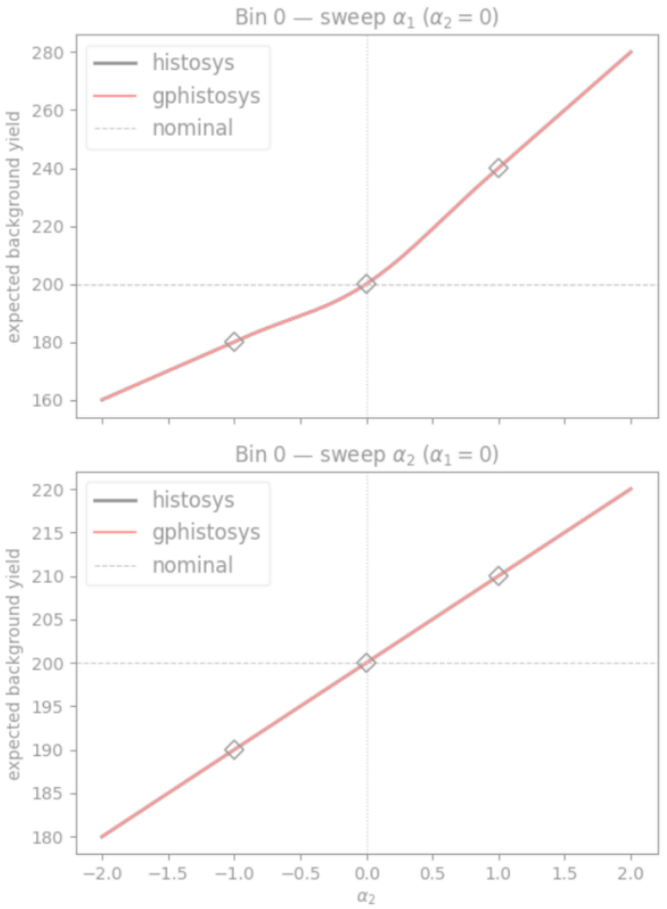
Idea: flexibility of non-parametric model with control of parametric one

Gaussian Processes are an excellent non-parametric replacement.

Regularisable with parametric priors, more expressive, and come with robust uncertainty quantification.

Ongoing work on GP modeling (Cranmer, Heinrich & Sandesara)

Easy to regularise with choice of prior



Gaussian Process = HistFactory

when densely sampled simulations are **not** available

(HistFactory used as prior)

Semi-regularised Approach

For more details see the blueprint workshop talk

	x	θ_μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric
Semi-parametric SBI (POI&Nuisance)	non-parametric	parametric	parametric
Semi-parametric SBI (POI& GP Nuisance)	non-parametric with NN	parametric	GP not NN

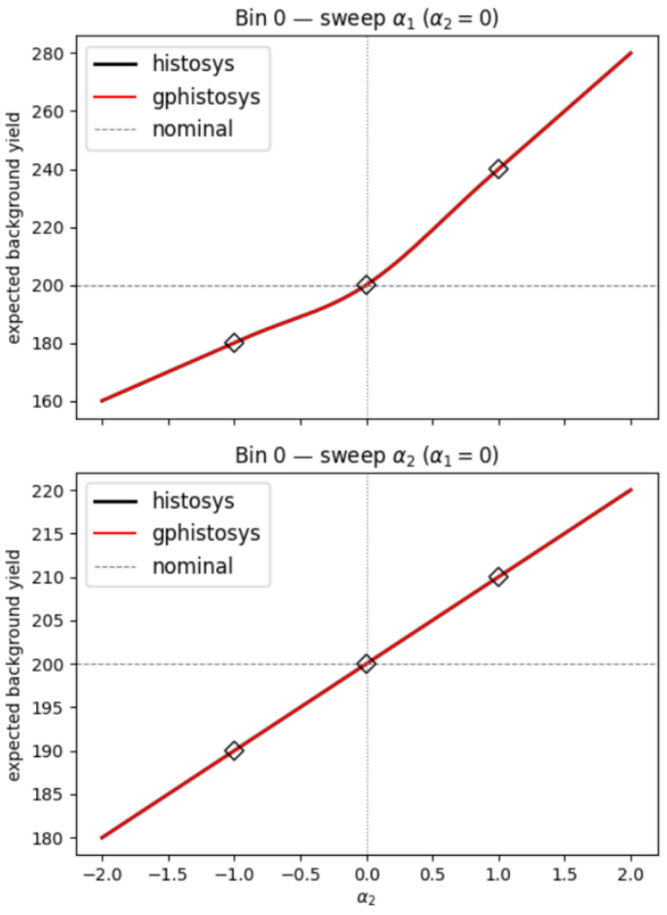
Idea: flexibility of non-parametric model with control of parametric one

Gaussian Processes are an excellent non-parametric replacement.

Regularisable with parametric priors, more expressive, and come with robust uncertainty quantification.

Ongoing work on GP modeling (Cranmer, Heinrich & Sandesara)

Easy to regularise with choice of prior



Gaussian Process = HistFactory

when densely sampled simulations are **not** available

(HistFactory used as prior)

What is Simulation-Based Inference ?

SBI

NSBI

	x	μ	ν
Traditional Statistical Model	parametric	parametric	parametric
Full Neural SBI	non-parametric	non-parametric	non-parametric
Semi-parametric SBI (POI&Nuisance)	non-parametric	parametric	parametric
Semi-parametric SBI (POI& GP Nuisance)	non-parametric with NN	parametric	GP not NN

NSBI for LHC

What is Neural Simulation-Based Inference ?

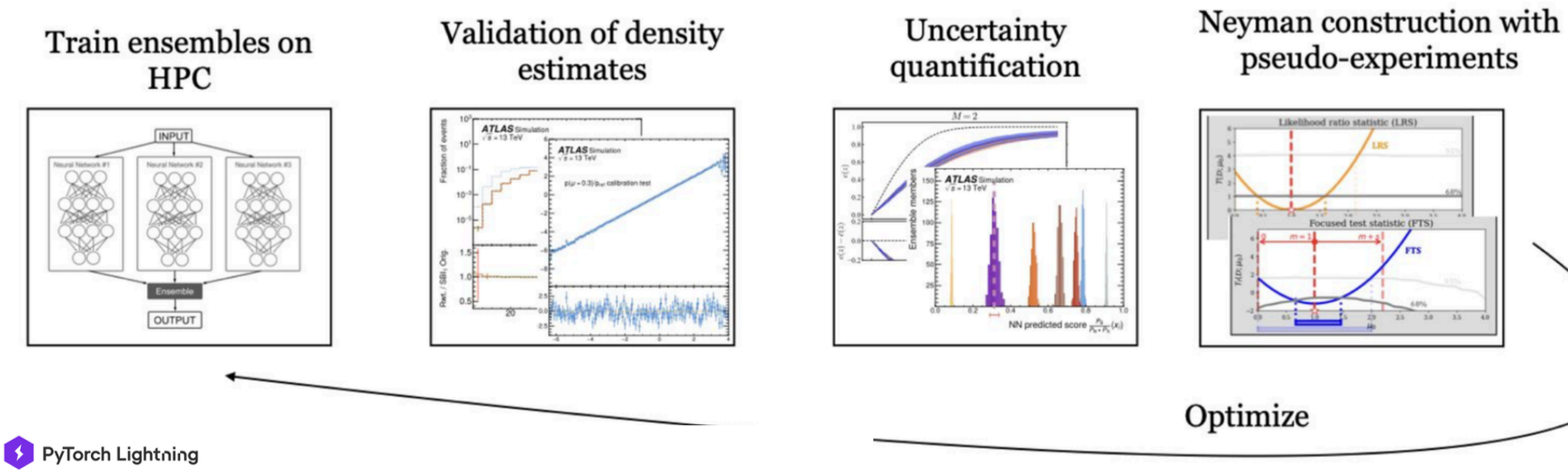
What is the future of NSBI?

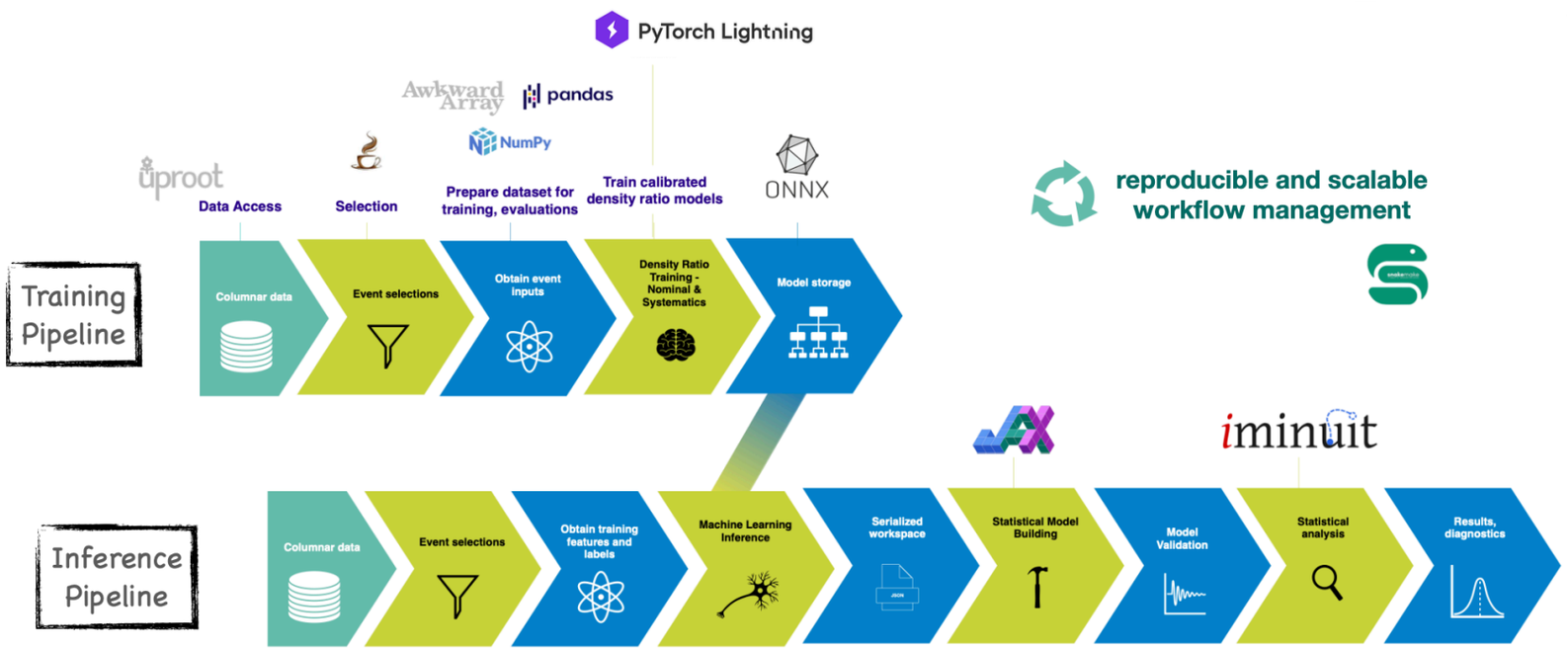
Building a Roadmap for wider adoption

- Organised a "blueprint" workshop earlier this year, aiming to build a roadmap towards developing an ecosystem of tools that enables broad adoption of neural SBI at the LHC.

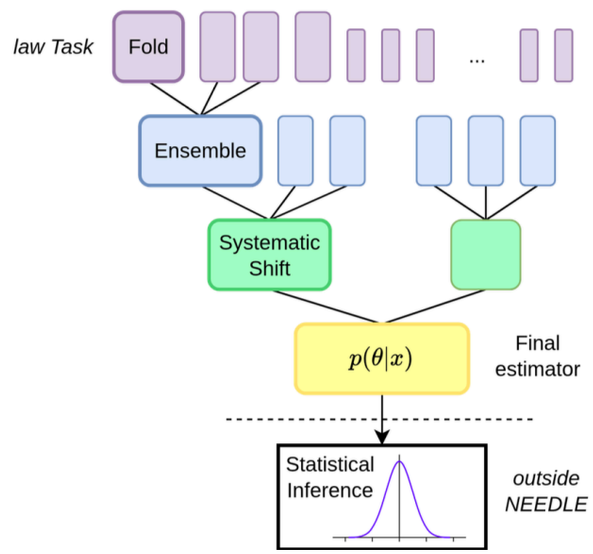
- Community whitepaper being written now outlining the plans and discussions.

Ghosh, et al. [talk]





Sandesara, et al. [talk]

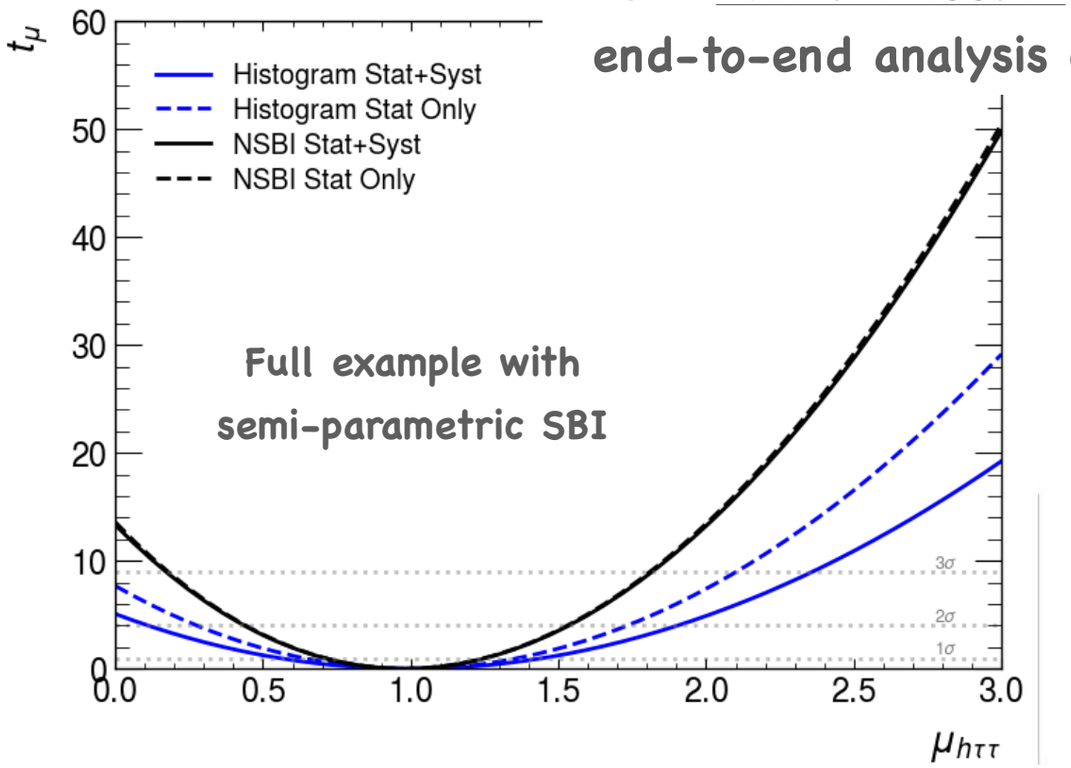


Schmidt, et al. [talk]

Building a Roadmap for wider adoption

Source:

See nsbi-lhc-toolkit for an end-to-end analysis example



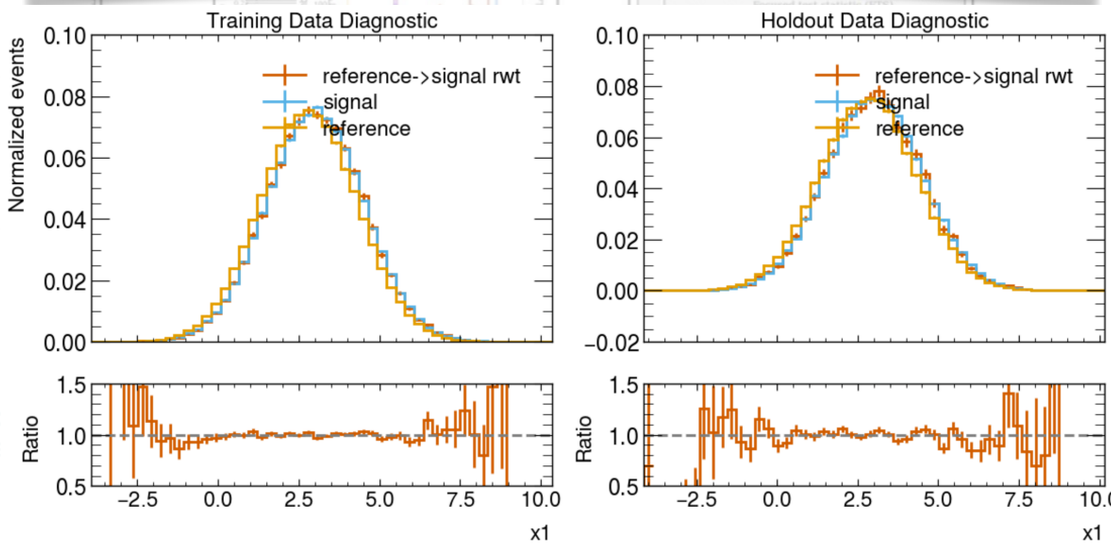
Systematic uncertainties

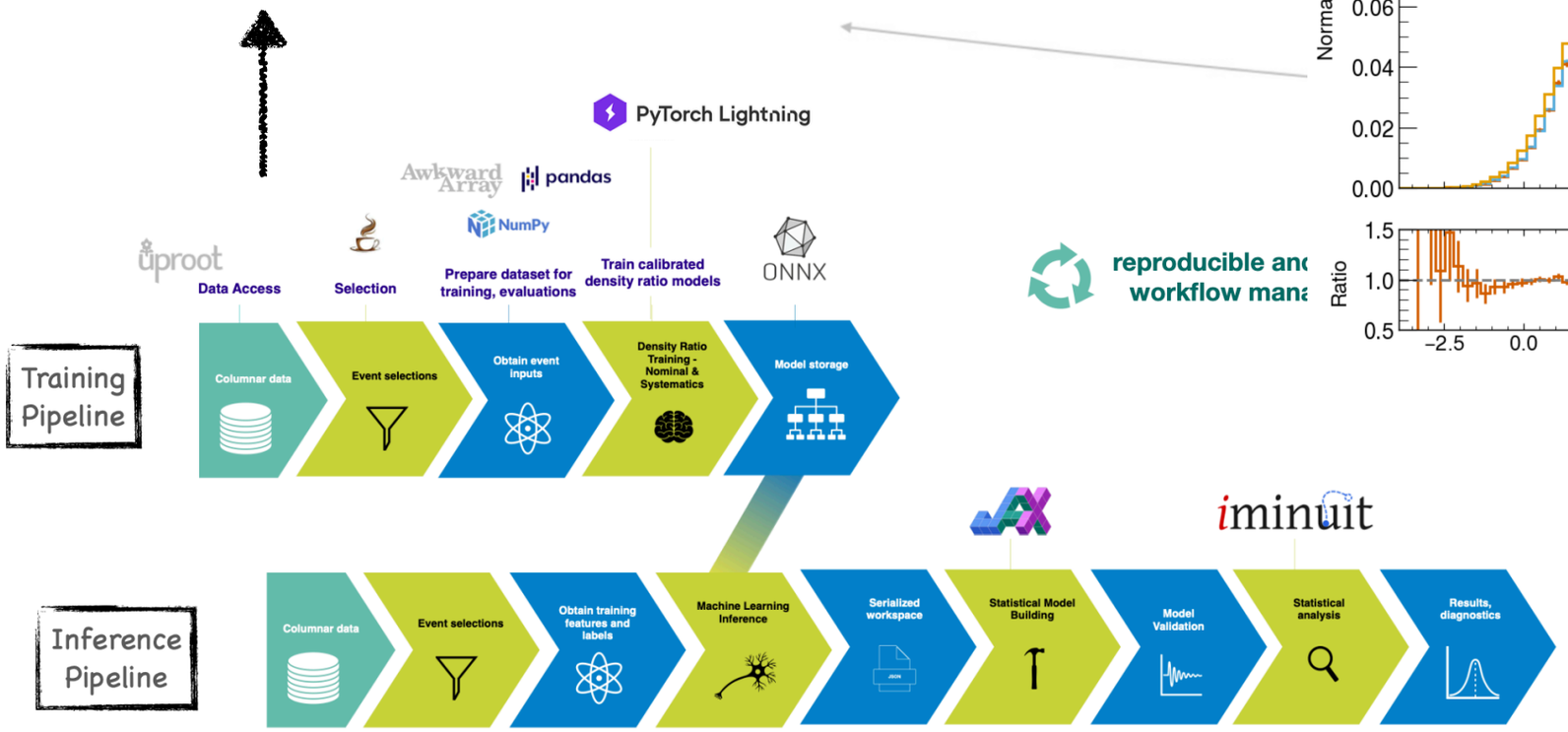
Shape + normalization uncertainties point to the up/down varied ROOT files:

```
Systematics:
  - Name: JES
    Type: NormPlusShape
    Nominal: 0
    Samples: [htautau, ztautau, ttbar]
    Up:
      - SampleName: htautau
        Path: ./saved_datasets/dataset_JES_up.root
        Tree: tree_htautau
        Weight: weights
    Dn:
      - SampleName: htautau
        Path: ./saved_datasets/dataset_JES_dn.root
        Tree: tree_htautau
        Weight: weights
```

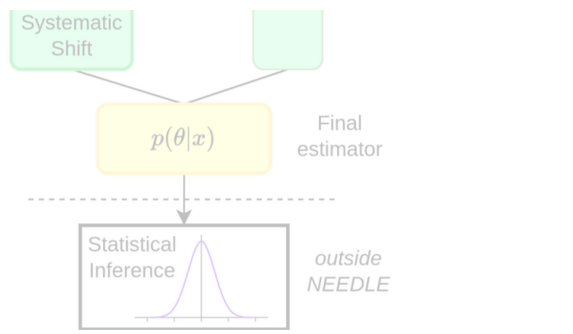
- **Nominal: 0** — the nuisance parameter starts at zero (the Gaussian constraint is centred here).
- The workspace builder computes variation ratios (varied / nominal) automatically from the histograms.

Validation of den… estimates





reproducible and… workflow mana…

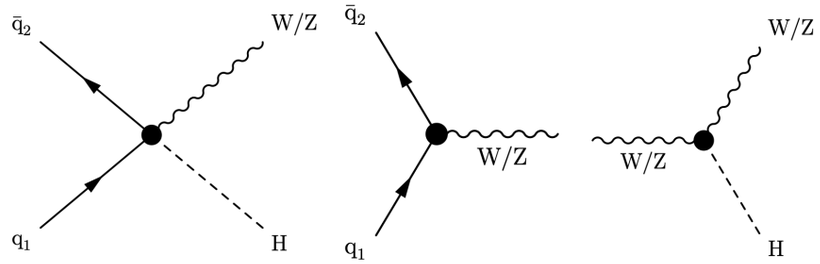


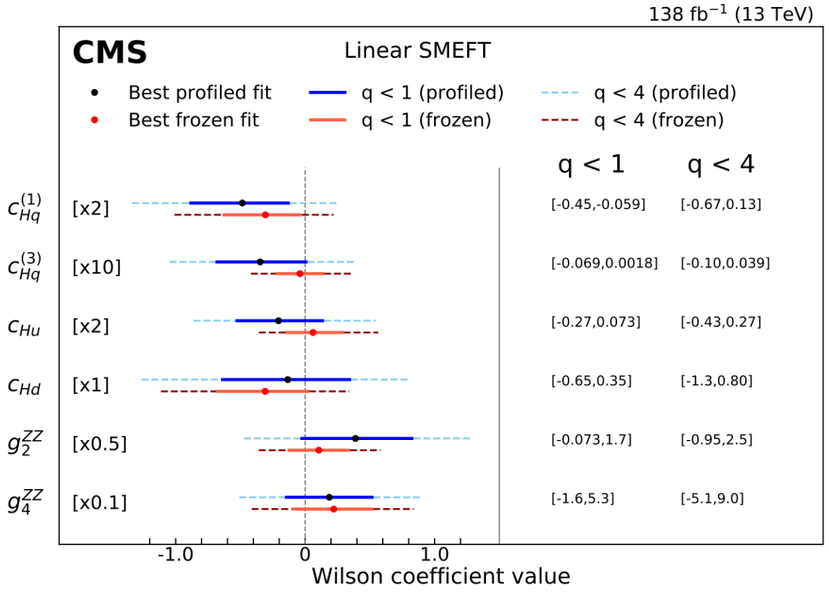
Sandesara, et al. [talk]

Schmidt, et al. [talk]

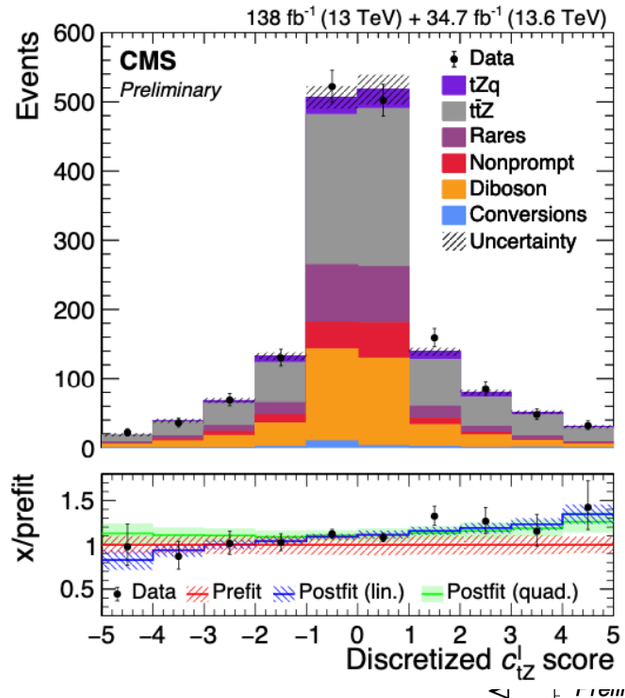
Summary and Outlook

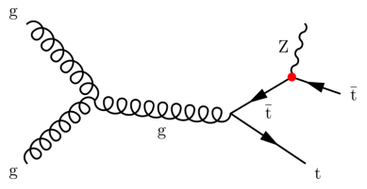
- Sufficient summary-based SBI not covered in this talk - can achieve dimensionality reduction $x \to s(x)$ without losing sensitivity to parameters.

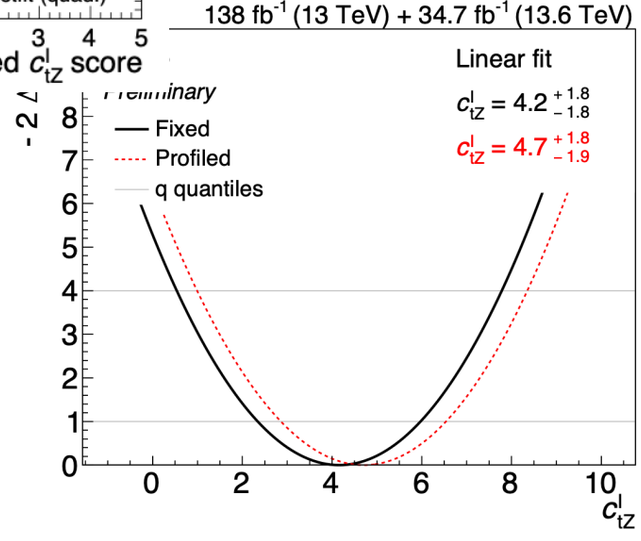




[2411.16907]





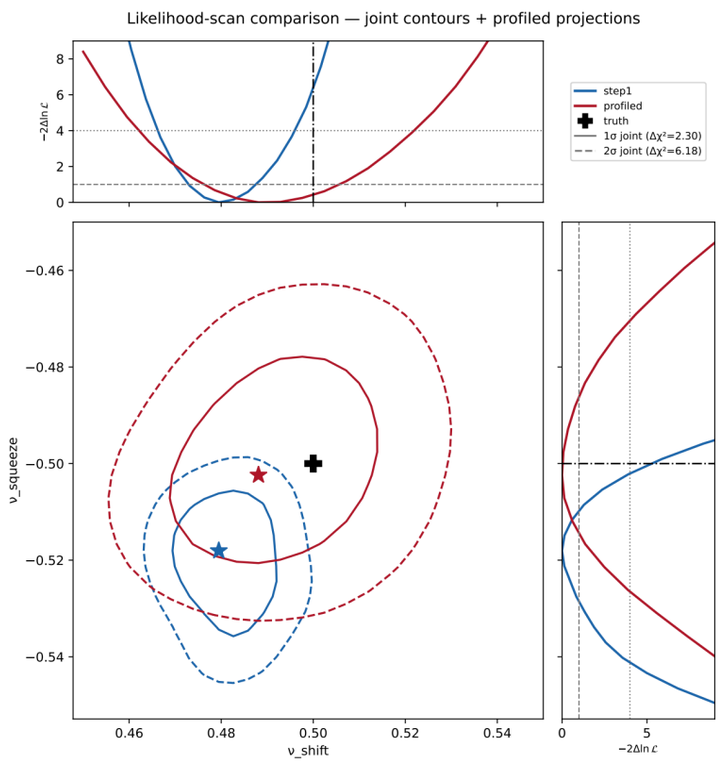


[2411.16907, 2405.13524]

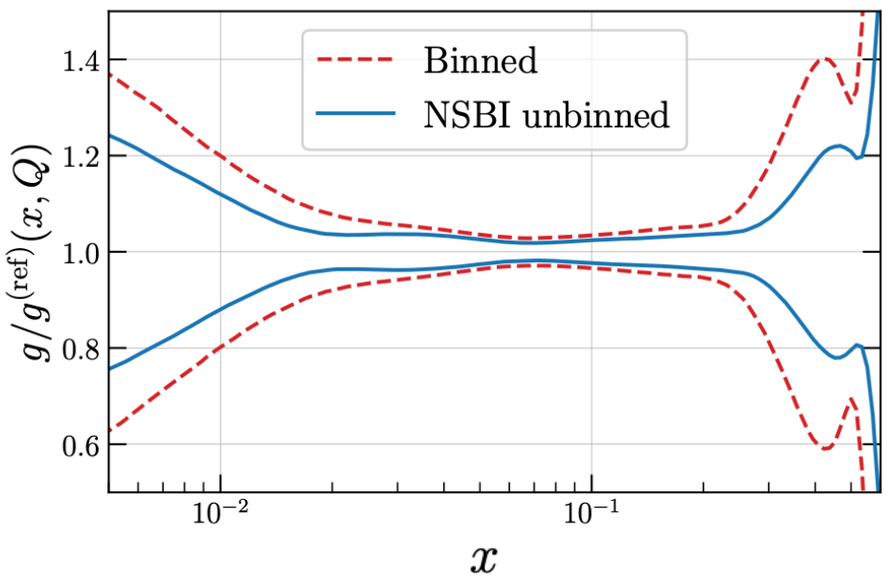
Summary and Outlook

- Sufficient summary-based SBI not covered in this talk - can achieve dimensionality reduction $x \to s(x)$ without losing sensitivity to parameters.

- A lot of studies and work are being done to make NSBI robust and scalable in real experiments.



Systematics aware semi-parametric training but with distributions directly, via Normalizing Flows [2602.13184]



Reduce the reliance on coarse approximations of uncertainties and their correlations, which are propagated as nuisance parameters. [2604.13157]

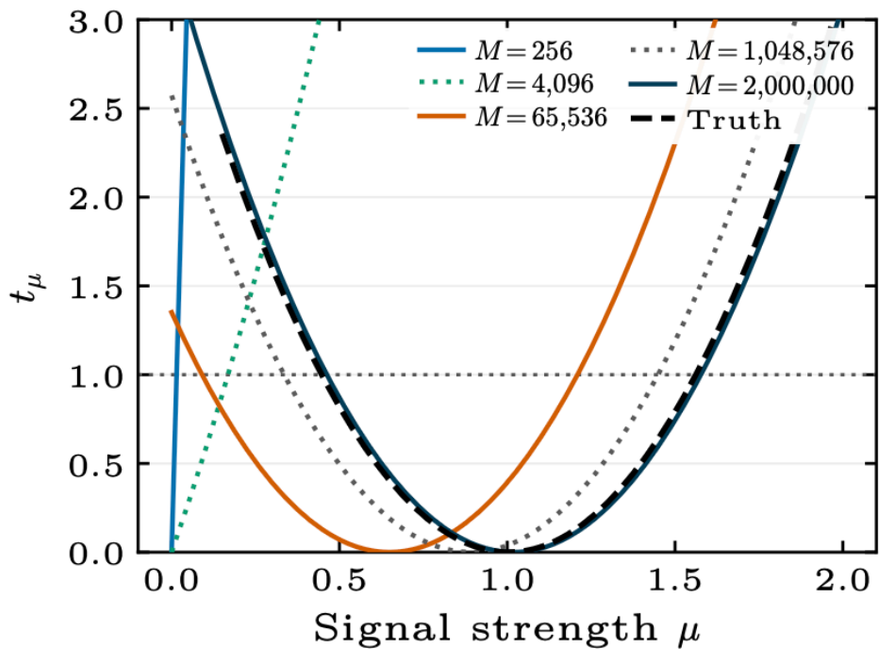
Summary and Outlook

- Sufficient summary-based SBI not covered in this talk - can achieve dimensionality reduction $x \to s(x)$ without losing sensitivity to parameters.

- A lot of studies and work are being done to make NSBI robust and scalable in real experiments.

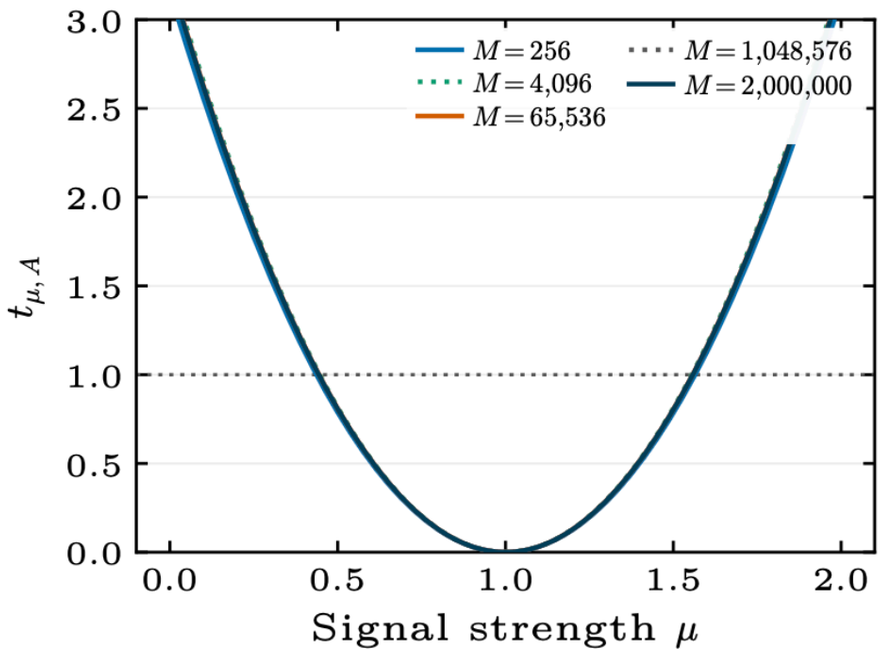
Extend the Asimov idea now to the unbinned approach [2609.14136]

Simulator sample fit



2M events to reproduce unbiased truth

Asimov sample fit



256 events reproduce the unbiased truth

Summary and Outlook

- Sufficient summary-based SBI not covered in this talk - can achieve dimensionality reduction $x \to s(x)$ without losing sensitivity to parameters.

- A lot of studies and work are being done to make NSBI robust and scalable in real experiments.

- We are entering the era of precision: NSBI is an integral technology for accelerating the discovery potential of our very expensive experiment.

Continue to beat projections!

example figure from here