University of Oviedo

MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES

Cofinanciado por
la Unión Europea

Getting the most out of your data: simulation-based inference techniques for Higgs physics

From histogram-based analyses to unbinned inference with neural networks

Sergio Sánchez Cruz
University of Oviedo / ICTEA

September 29, 2026

Roadmap

1. A standard Higgs analysis at the LHC and its (potential) limitations
2. Building optimal observables with the likelihood ratio trick
3. Simulation-based inference (SBI) methods with augmented simulations
4. Higgs-specific use cases and methods
5. Systematic uncertainties: towards an unbinned analysis.

Roadmap

1. A standard Higgs analysis at the LHC and its (potential) limitations
2. Building optimal observables with the likelihood ratio trick
3. Simulation-based inference (SBI) methods with augmented simulations
4. Higgs-specific use cases and methods
5. Systematic uncertainties: towards an unbinned analysis.

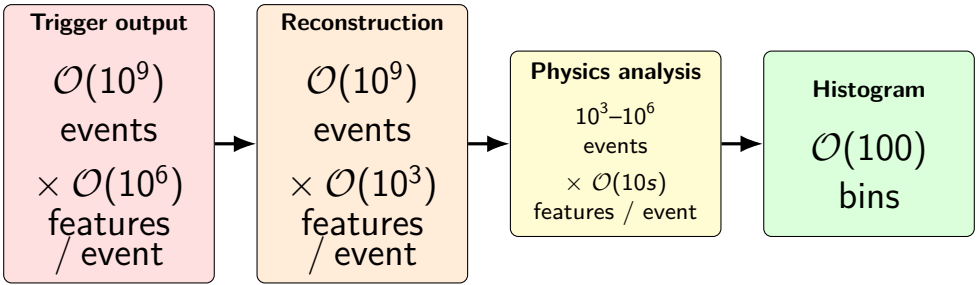
Caveat

- Much of the content of this talk are not yet "consolidated" analysis techniques
- So far, none of the LHC experiments have released a full SBI analysis
- Further developments are expected, this talk is a snapshot of the current status
- Some interesting bibliography in the back-up

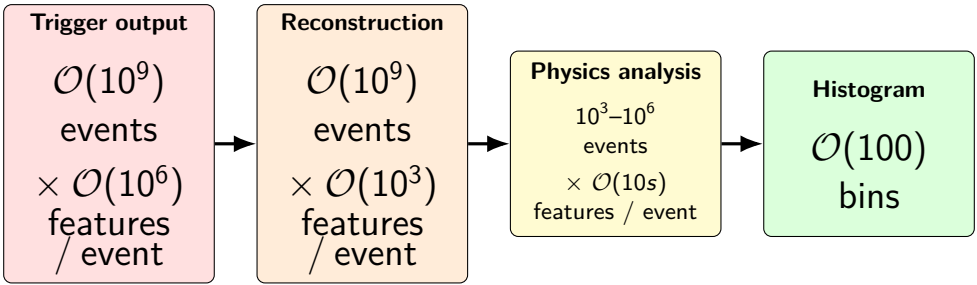
Section 1

Introduction: a standard Higgs analysis at the LHC

A standard Higgs analysis at the LHC from a information point of view



A standard Higgs analysis at the LHC from a information point of view



Each step compresses the information available in the previous step, but **there's some inherent information loss**

When Does Information Loss Matter?

For a fixed, simple hypothesis, the **Neyman–Pearson lemma tells us how to construct an optimal summary**. A conventional analysis can be close to optimal when separating one signal from one background.

The challenge appears when interesting physics is not so simple

When Does Information Loss Matter?

For a fixed, simple hypothesis, the **Neyman–Pearson lemma tells us how to construct an optimal summary**. A conventional analysis can be close to optimal when separating one signal from one background.

The challenge appears when interesting physics is not so simple

Systematics

The best classifier at nominal conditions need not be the best summary once nuisance effects matter.

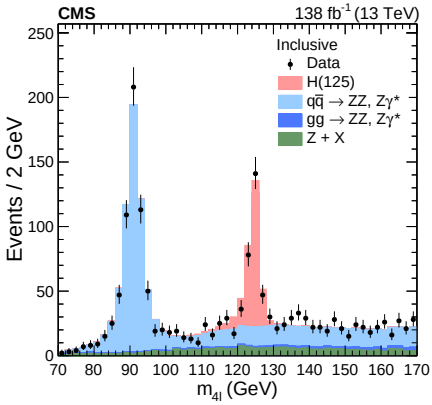
Many parameters

EFT and κ fits have no single summary that is optimal for every parameter direction.

Nonlinear response

When signals vary nonlinearly with the POIs, the optimal summary changes across parameter space.

Into the binned statistical model



CMS HIG-21-019

Let y be the analysis observable and divide its range into bins $i = 1, \ldots, B$.

$$n_i \sim \text{Poisson}(\nu_i(\boldsymbol{\theta}, \boldsymbol{\eta})), \qquad \nu_i = \mathcal{L}\,\sigma_i(\boldsymbol{\theta}, \boldsymbol{\eta}) + b_i(\boldsymbol{\eta}).$$

- $\mathcal{L}$: integrated luminosity.
- σ_i: signal expectation in bin i.
- b_i: backgrounds and their modeling uncertainties.
- $\boldsymbol{\theta}$: parameters of interest; $\boldsymbol{\eta}$: nuisance parameters.

The binned likelihood

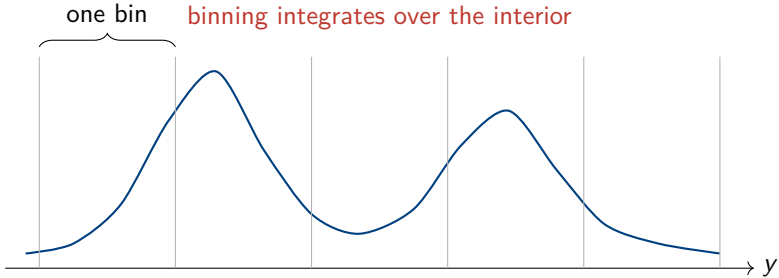
$$\mathcal{L}_{\text{bin}}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \prod_{i=1}^{B} \frac{\nu_i(\boldsymbol{\theta}, \boldsymbol{\eta})^{n_i} e^{-\nu_i(\boldsymbol{\theta}, \boldsymbol{\eta})}}{n_i!} \, \pi(\boldsymbol{\eta}).$$

- $\nu_i(\boldsymbol{\theta}, \boldsymbol{\eta})$: prediction for the mean number of events in bin i.
- n_i: observed count in bin i; $i = 1, \ldots, B$ labels the B histogram bins.
- $\boldsymbol{\theta}$: parameters of interest; $\boldsymbol{\eta}$: nuisance parameters describing systematic effects.
- $\pi(\boldsymbol{\eta})$ encodes auxiliary measurements or constraint terms.
- The profile likelihood removes nuisance parameters; hats denote the parameter values that maximize $\mathcal{L}$:

$$\lambda(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta}, \widehat{\boldsymbol{\eta}}_{\boldsymbol{\theta}})}{\mathcal{L}(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\eta}})}.$$

This model is a convenient choice, but may not be optimal, event if the variable choice is been designed with care

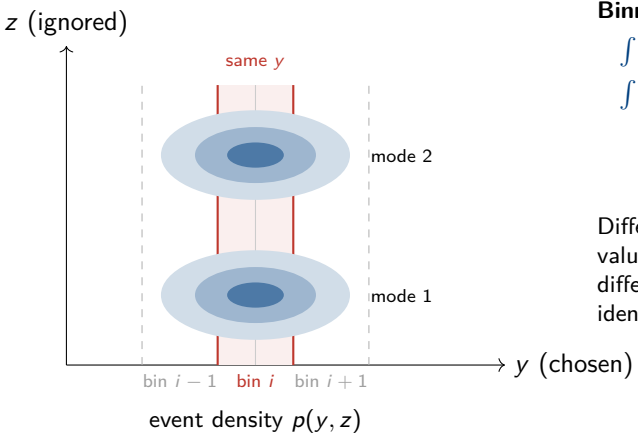
Where the information goes



continuous event information

The binned expectation is $\nu_i = \int_{\text{bin } i} \nu(y)\,\mathrm{d}y$. Different shapes inside a bin can produce the same ν_i.

The cost of choosing one observable



Binning marginalizes over:

- ∫ The interior of a single bin in y
- ∫ **All other event degrees of freedom** (z, w, . . .)

$$\nu_i = \int_{\text{bin } i} \int_{\text{all other vars}} p(\boldsymbol{x})\,\mathrm{d}y\,\mathrm{d}z\,\mathrm{d}(\text{rest})$$

Different events with the same $y \in$ bin i but different values of z or other observables can have very different physics implications, yet they all contribute identically to ν_i.

Can we get away with binning better?

More bins preserve more shape information

- But simulation samples become sparse.
- Templates require interpolation across parameters.
- Nuisance variations multiply the number of templates.
- Empty or low-count bins make the fit fragile.

It's not only about binning

- A single variable cannot be optimal for complex hypotheses (multiple POIs, nonlinear effects, nuisance parameters)
- A d-dimensional histogram has $\mathcal{O}(B^d)$ bins
- Correlations carry information that 1D projections hide
- No single hand-designed observable is globally optimal over parameter space

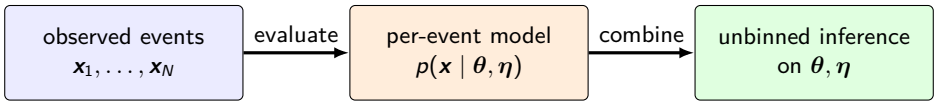
At the end of the day, binned approaches have inherent limitations due to the multidimensionality of data and the physical model

Section 2

The likelihood ratio trick

The goal of simulation-based inference

The goal is obtaining the probability density $p(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\eta})$ for each given event

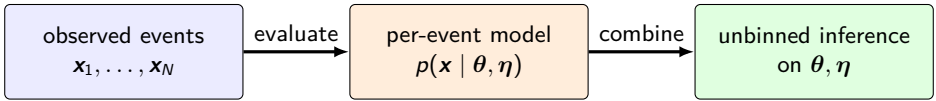


$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \text{Poisson}(N|\nu(\boldsymbol{\theta},\boldsymbol{\eta})) \prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})\ \pi(\boldsymbol{\eta}) \tag{1}$$

$\boldsymbol{x}_e$ are the high-dimensional event representation (objects kinematics variables, high level variables, etc)

The goal of simulation-based inference

The goal is obtaining the probability density $p(\boldsymbol{x}_i|\boldsymbol{\theta},\boldsymbol{\eta})$ for each given event



$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \text{Poisson}(N|\nu(\boldsymbol{\theta},\boldsymbol{\eta})) \prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})\ \pi(\boldsymbol{\eta}) \tag{1}$$

$\boldsymbol{x}_e$ are the high-dimensional event representation (objects kinematics variables, high level variables, etc)

The challenge: $p(\boldsymbol{x} \mid \boldsymbol{\theta},\boldsymbol{\eta})$ cannot be computed analytically

Simulation-based inference

Monte Carlo generators:

- do not provide an analytical expression for the likelihood,
- but they can generate samples from the distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta})$

Simulation-based inference

Monte Carlo generators:

- do not provide an analytical expression for the likelihood,
- but they can generate samples from the distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta})$

Simulation-based inference use data generated with Monte Carlo to build a surrogate model for the likelihood (ratio) $\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta})$ (or $\hat{r}(\boldsymbol{x} \mid \theta_0, \theta_1) \equiv \widehat{\frac{p(\boldsymbol{x}|\theta_0,\boldsymbol{\eta})}{p(\boldsymbol{x}|\theta_1,\boldsymbol{\eta}))}}$)

If accurate, inference can be performed using the surrogate model as if it were the true likelihood

Likelihood ratios are enough for a test

For two hypotheses, the Neyman–Pearson lemma says the most powerful test is a threshold on

$$r(\boldsymbol{x}; \theta_1, \theta_2) = \frac{p(\boldsymbol{x} \mid \theta_1)}{p(\boldsymbol{x} \mid \theta_2)}.$$

In practice, we can use one fixed reference point θ_{ref}:

$$r(\boldsymbol{x}; \theta_1, \theta_2) = \frac{r(\boldsymbol{x}; \theta_1, \theta_{\mathrm{ref}})}{r(\boldsymbol{x}; \theta_2, \theta_{\mathrm{ref}})}.$$

For independent events, the dataset-level ratio is

$$R(\mathcal{D}; \theta_1, \theta_2) = \prod_{e=1}^{N} r(\boldsymbol{x}_e; \theta_1, \theta_2), \qquad \log R = \sum_{e=1}^{N} \log r(\boldsymbol{x}_e; \theta_1, \theta_2).$$

The likelihood ratio trick

A simplified starting point

For now, consider two simple hypotheses: fixed parameter points with no nuisance parameters. The same likelihood-ratio idea will be generalized to parameter estimation and systematics later

The likelihood ratio trick

A simplified starting point

For now, consider two simple hypotheses: fixed parameter points with no nuisance parameters. The same likelihood-ratio idea will be generalized to parameter estimation and systematics later

We train a classifier on samples from $p_1(\boldsymbol{x})$ and $p_0(\boldsymbol{x})$ with binary cross-entropy loss:

$$\mathcal{L}(d) = -\mathbb{E}_{p_1} \log d(\boldsymbol{x}) - \mathbb{E}_{p_0} \log(1 - d(\boldsymbol{x})).$$

At each $\boldsymbol{x}$, the optimal classifier satisfies

$$d^*(\boldsymbol{x}) = \frac{p_1(\boldsymbol{x})}{p_1(\boldsymbol{x}) + p_0(\boldsymbol{x})} \Rightarrow \frac{d^*(\boldsymbol{x})}{1 - d^*(\boldsymbol{x})} = \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} = r(\boldsymbol{x}).$$

The likelihood ratio trick

A simplified starting point

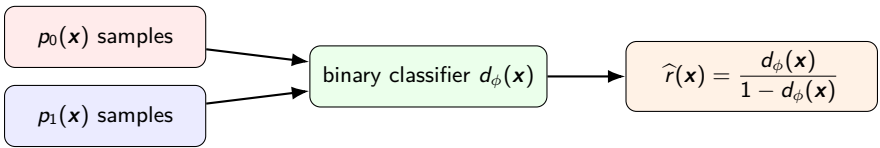
For now, consider two simple hypotheses: fixed parameter points with no nuisance parameters. The same likelihood-ratio idea will be generalized to parameter estimation and systematics later

We train a classifier on samples from $p_1(\boldsymbol{x})$ and $p_0(\boldsymbol{x})$ with binary cross-entropy loss:

$$\mathcal{L}(d) = -\mathbb{E}_{p_1} \log d(\boldsymbol{x}) - \mathbb{E}_{p_0} \log(1 - d(\boldsymbol{x})).$$

At each $\boldsymbol{x}$, the optimal classifier satisfies

$$d^*(\boldsymbol{x}) = \frac{p_1(\boldsymbol{x})}{p_1(\boldsymbol{x}) + p_0(\boldsymbol{x})} \Rightarrow \frac{d^*(\boldsymbol{x})}{1 - d^*(\boldsymbol{x})} = \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} = r(\boldsymbol{x}).$$



Section 3

Extensions of the likelihood ratio trick

More on generators...

Generators in particle physics incorporate the full chain of physics and detector effects, including:

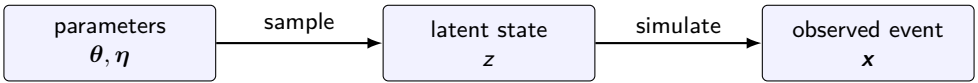
- Generation of the hard scattering process
- Parton shower and hadronization
- Detector simulation
- Signal processing and reconstruction

More on generators...

Generators in particle physics incorporate the full chain of physics and detector effects, including:

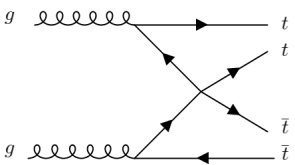
- Generation of the hard scattering process
- Parton shower and hadronization
- Detector simulation
- Signal processing and reconstruction

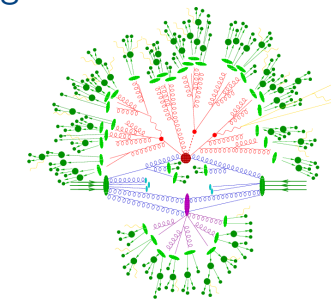
Luckily, these are factorized into a sequence of conditional distributions, which are sampled sequentially

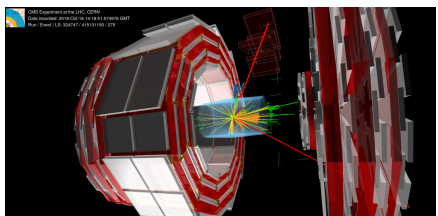


Generators do not only sample from $p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta})$, but from the joint distribution of the latent state z and the joint density $p(\boldsymbol{x}, z \mid \boldsymbol{\theta}, \boldsymbol{\eta})$

A few more words on generators







Parton-level

(Normally) tractable

- $p(z \mid \boldsymbol{\theta}, \boldsymbol{\eta})$

Particle-level

Intractable

- $p(\boldsymbol{x}_{\text{particle}}, z \mid \boldsymbol{\theta}, \boldsymbol{\eta})$

Detector-level

Intractable

- $p(\boldsymbol{x}_{\text{reco}}, \boldsymbol{x}_{\text{particle}}, z \mid \boldsymbol{\theta}, \boldsymbol{\eta})$

Example: Why matrix-element reweighting works

Let z denote the parton-level state and let $K(\boldsymbol{x}_{\mathrm{reco}} \mid z)$ collect shower, hadronization, detector, and reconstruction effects.

$$p(\boldsymbol{x}_{\mathrm{reco}}, z \mid \theta) = \underbrace{p_{\mathrm{hard}}(z \mid \theta)}_{\text{matrix element}} \quad \underbrace{K(\boldsymbol{x}_{\mathrm{reco}} \mid z)}_{\text{shower + detector response}} \quad .$$

Example: Why matrix-element reweighting works

Let z denote the parton-level state and let $K(\boldsymbol{x}_{\mathrm{reco}} \mid z)$ collect shower, hadronization, detector, and reconstruction effects.

$$p(\boldsymbol{x}_{\mathrm{reco}}, z \mid \theta) = \underbrace{p_{\mathrm{hard}}(z \mid \theta)}_{\text{matrix element}} \quad \underbrace{K(\boldsymbol{x}_{\mathrm{reco}} \mid z)}_{\text{shower + detector response}} \quad .$$

If the response is independent of the parameter point, we can define a weight as:

$$w(z; \theta_1, \theta_0) = \frac{p_{\mathrm{hard}}(z \mid \theta_1)}{p_{\mathrm{hard}}(z \mid \theta_0)}.$$

Note that this is similar to the likelihood ratio!

Improving the likelihood ratio trick

Let's assume an scenario where we can do sample reweighting:

- Usually when the two hypotheses are defined at parton level (e. g. variations couplings)
- The augmented dataset has access to the joint density ratio $r(\boldsymbol{x}, z; \theta_1, \theta_0) = \frac{p_{\text{hard}}(z|\theta_1)}{p_{\text{hard}}(z|\theta_0)} = w(z; \theta_1, \theta_0)$
- Note that $r(\boldsymbol{x}, z; \theta_1, \theta_0)$ is known, while $r(\boldsymbol{x}; \theta_1, \theta_0)$ is not

Improving the likelihood ratio trick

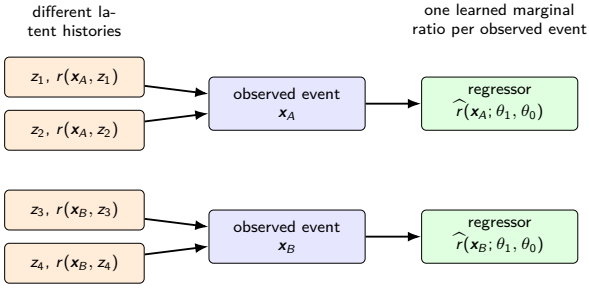
Let's assume an scenario where we can do sample reweighting:

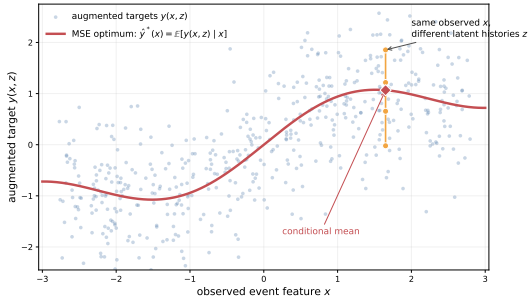
- Usually when the two hypotheses are defined at parton level (e. g. variations couplings)
- The augmented dataset has access to the joint density ratio $r(\boldsymbol{x}, z; \theta_1, \theta_0) = \frac{p_{\mathrm{hard}}(z|\theta_1)}{p_{\mathrm{hard}}(z|\theta_0)} = w(z; \theta_1, \theta_0)$
- Note that $r(\boldsymbol{x}, z; \theta_1, \theta_0)$ is known, while $r(\boldsymbol{x}; \theta_1, \theta_0)$ is not

The marginal ratio is the ratio of integrals, equivalently a conditional average under θ_0:

$$\begin{aligned} r(\boldsymbol{x}; \theta_1, \theta_0) &= \frac{\int \mathrm{d}z\, p(\boldsymbol{x}, z \mid \theta_1)}{\int \mathrm{d}z\, p(\boldsymbol{x}, z \mid \theta_0)} \\ &= \int \mathrm{d}z\, p(z \mid \boldsymbol{x}, \theta_0) \frac{p(\boldsymbol{x}, z \mid \theta_1)}{p(\boldsymbol{x}, z \mid \theta_0)} \\ &= \mathbb{E}_{p(z|\boldsymbol{x}, \theta_0)}[r(\boldsymbol{x}, z; \theta_1, \theta_0)]\,. \end{aligned}$$

Regression learns the marginal likelihood ratio





With (θ_1, θ_0) fixed, minimize the expected regression loss:

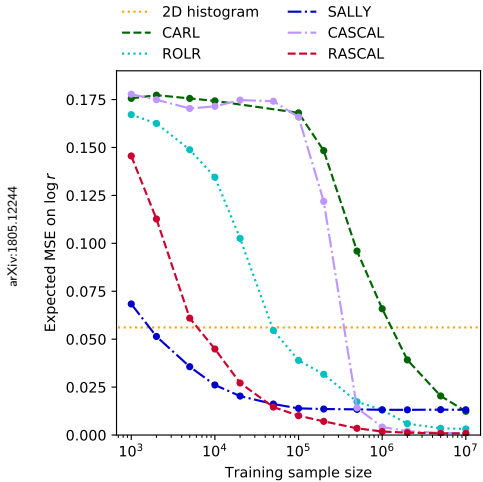
$$\widehat{r}^*(\boldsymbol{x}; \theta_1, \theta_0) = \arg\min_{\widehat{r}} \mathbb{E}\left[\left(r(\boldsymbol{x}, z; \theta_1, \theta_0) - \widehat{r}(\boldsymbol{x}; \theta_1, \theta_0)\right)^2\right].$$

The ROLR loss in the training

- A network can be trained that minimizes that loss
- Taking simulated augmented events $\{(\boldsymbol{x}_j, z_j)\}_{j=1}^{N_{\text{train}}}$

$$\widehat{L}_{\text{ROLR}}(\phi) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left[r(\boldsymbol{x}_j, z_j; \theta_1, \theta_0) - \widehat{r}_\phi(\boldsymbol{x}_j; \theta_1, \theta_0) \right]^2 .$$

- This is **ROLR**: Regression On Likelihood Ratio.



arXiv:1805.12244

- ROLR tends to achieve a better performance than the likelihood ratio trick

In reality, there are no simple hypotheses in Higgs

- Any problem of interest is more complex than two simple hypotheses
- We need to do proper inference on a continuous θ, ν space
- The method only allows to test $((\theta_0, \nu_0)$ vs $(\theta_1, \nu_1))$

The name of the game is to learn a **parameterized likelihood ratio** $r(\boldsymbol{x}; \theta_0, \theta_{ref})$ that can be evaluated for any parameter point θ_0.

Training a classifier for a continuum assamble of hypotheses is not feasible, one needs to inject additional information

SALLY: score compression for local inference

At a reference point θ_{ref}, the score is

$$t(\boldsymbol{x} \mid \theta_{\mathrm{ref}}) = \nabla_\theta \log p(\boldsymbol{x} \mid \theta)|_{\theta_{\mathrm{ref}}} .$$

SALLY: score compression for local inference

At a reference point θ_{ref}, the score is

$$t(\boldsymbol{x} \mid \theta_{\mathrm{ref}}) = \nabla_\theta \log p(\boldsymbol{x} \mid \theta)|_{\theta_{\mathrm{ref}}} .$$

One can regress the joint score $t(\boldsymbol{x}, z \mid \theta_{\mathrm{ref}})$ on $\boldsymbol{x}$ to obtain $\widehat{t}(\boldsymbol{x})$

SALLY: score compression for local inference

At a reference point θ_{ref}, the score is

$$t(\boldsymbol{x} \mid \theta_{\mathrm{ref}}) = \nabla_\theta \log p(\boldsymbol{x} \mid \theta)|_{\theta_{\mathrm{ref}}} .$$

One can regress the joint score $t(\boldsymbol{x}, z \mid \theta_{\mathrm{ref}})$ on $\boldsymbol{x}$ to obtain $\widehat{t}(\boldsymbol{x})$

For augmented samples $\{(\boldsymbol{x}_j, z_j)\}_{j=1}^{N_{\mathrm{train}}}$, use the vector MSE loss

$$\widehat{L}_{\mathrm{SALLY}}(\phi) = \frac{1}{N_{\mathrm{train}}} \sum_{j=1}^{N_{\mathrm{train}}} \left\| \widehat{t}_\phi(\boldsymbol{x}_j) - t(\boldsymbol{x}_j, z_j \mid \theta_{\mathrm{ref}}) \right\|^2 .$$

For nearby parameter points, the score is locally sufficient to first order in $\theta - \theta_{\mathrm{ref}}$.

RASCAL: ratio regression augmented with the score

RASCAL augments ROLR with the joint score. It learns a parameterized likelihood ratio $\widehat{r}_\phi(\boldsymbol{x}; \theta_1, \theta_0)$ directly from augmented simulator targets.

$$\mathcal{L}_{\mathrm{RASCAL}} = \left[\widehat{r}_\phi(\boldsymbol{x}; \theta_1, \theta_0) - r(\boldsymbol{x}, z; \theta_1, \theta_0)\right]^2 + \alpha \left\|\nabla_{\theta_1} \log \widehat{r}_\phi(\boldsymbol{x}; \theta_1, \theta_0) - t(\boldsymbol{x}, z \mid \theta_1)\right\|^2 .$$

- The first term is the ROLR loss: it regresses the joint likelihood ratio on the observed event.
- The second term matches the derivative of the learned ratio to the joint score.
- There is no classifier term: RASCAL is a regression-based method.

- RASCAL is a generalization of ROLR to parameterized likelihood ratios
- It is also a generalization of SALLY to nonlocal inference
- Other methods, such as SALLINO, SALLYS, and CASCAL, are variations on this framework (check the bib in the back-up)

Recap

- The likelihood ratio trick is a “good” technique to obtain surrogates of density ratios in general cases
- For some physical problems, “augmenting” the Monte Carlo datasets gives additional information
- Dedicated loss functions profit from the augmented datasets to enhance convergence and robustness

Section 4

Physical cases

Physical cases: EFTs and κ-framework

- The κ-framework and the SMEFT vary the couplings that enter in the Feynman diagrams

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \sum_i \frac{c_i}{\Lambda^2}\mathcal{O}_i$$

$$\sigma \sim \sigma_{SM} + \sum_i c_i\sigma_i + \sum_{i,j} c_i c_j \sigma_{ij}$$

Physical cases: EFTs and κ-framework

- The κ-framework and the SMEFT vary the couplings that enter in the Feynman diagrams

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \sum_i \frac{c_i}{\Lambda^2}\mathcal{O}_i$$

$$\sigma \sim \sigma_{SM} + \sum_i c_i\sigma_i + \sum_{i,j} c_i c_j \sigma_{ij}$$

The probability density follows the same functional dependence

From coupling variations to a likelihood-ratio form

The same amplitude expansion changes the event-level differential rate. Relative to the SM point, the likelihood ratio can be parameterized as

$$r(\boldsymbol{x};\theta,\mathrm{SM}) = 1 + \sum_i c_i\, l_i(\boldsymbol{x}) + \sum_{i,j} c_i c_j\, q_{ij}(\boldsymbol{x}).$$

- $l_i(\boldsymbol{x})$ describes the event-dependent interference response.
- $q_{ij}(\boldsymbol{x})$ describes the event-dependent EFT-squared response.
- The polynomial form is known from the physics; SBI learns its dependence on the observed event $\boldsymbol{x}$.

Regressing the linear and quadratic responses

For augmented events $(\boldsymbol{x}_j, z_j)$ generated at the SM reference point, the simulator provides the coefficient targets $l_i(\boldsymbol{x}_j, z_j)$ and $q_{ij}(\boldsymbol{x}_j, z_j)$. Train networks that only receive $\boldsymbol{x}_j$:

$$\widehat{L}_{l_i}(\phi) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left[l_i(\boldsymbol{x}_j, z_j) - \widehat{l}_{\phi,i}(\boldsymbol{x}_j) \right]^2 ,$$

$$\widehat{L}_{q_{i,j}}(\phi) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left[q_{ij}(\boldsymbol{x}_j, z_j) - \widehat{q}_{\phi,ij}(\boldsymbol{x}_j) \right]^2 .$$

Regressing the linear and quadratic responses

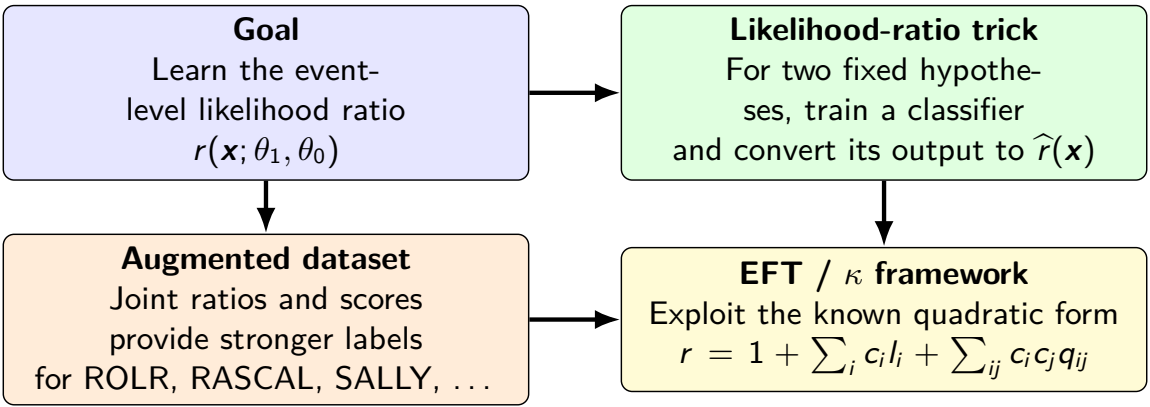
For augmented events $(\boldsymbol{x}_j, z_j)$ generated at the SM reference point, the simulator provides the coefficient targets $l_i(\boldsymbol{x}_j, z_j)$ and $q_{ij}(\boldsymbol{x}_j, z_j)$. Train networks that only receive $\boldsymbol{x}_j$:

$$\widehat{L}_{l_i}(\phi) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left[l_i(\boldsymbol{x}_j, z_j) - \widehat{l}_{\phi,i}(\boldsymbol{x}_j) \right]^2 ,$$

$$\widehat{L}_{q_{i,j}}(\phi) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left[q_{ij}(\boldsymbol{x}_j, z_j) - \widehat{q}_{\phi,ij}(\boldsymbol{x}_j) \right]^2 .$$

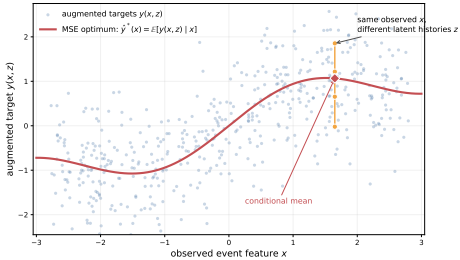
- The training target can be a single network with l_i and q_{ij} output heads: $\widehat{L} = \widehat{L}_l + \widehat{L}_q$.
- In the large-sample limit, each prediction is the conditional mean under the SM reference, e.g. $\widehat{l}_i(\boldsymbol{x}) \to \mathbb{E}[l_i(\boldsymbol{x}, z) \mid \boldsymbol{x}, \mathrm{SM}]$.
- Insert $\widehat{l}_i(\boldsymbol{x})$ and $\widehat{q}_{ij}(\boldsymbol{x})$ into the polynomial ratio on the previous slide.

Intermediate wrap-up: learning the likelihood ratio



How does this work in practice?

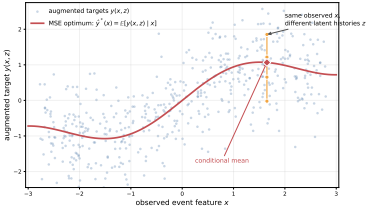
- It depends on the problem and the implementation
- When running a regression of the type $L = \sum_i (y_i(\boldsymbol{x}, z) - \hat{y}_i(z))^2$ one is learning the conditional expectation of the target $y_i(\boldsymbol{x}, z)$ given the observed event $\boldsymbol{x}$
- ... in other words, one is averaging over the latent space z

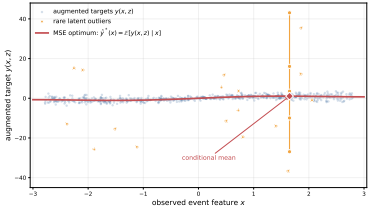


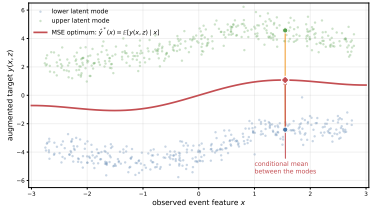
How does this work in practice?

Different factors can affect the convergence:

- Broad conditional distribution in $z \Rightarrow$ e.g. parton kinematics not well resolved from reco
- Outliers $\Rightarrow$ poor MC statistics, parameter space not well sampled
- Multimodal distribution $\Rightarrow$ unbalanced background samples
- Complexity also grows with the dimensionality of the latent space z!
- Omitting useful information to regress the parton space is very damaging to convergence!
- *Dimensionality is key!*

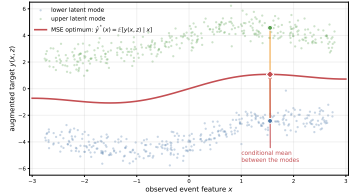






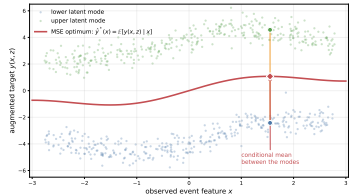
Factorization will help you!

- Multimodal behaviours and other instabilities may come in the presence of different processes
- One can easily factorize the likelihood:



Factorization will help you!

- Multimodal behaviours and other instabilities may come in the presence of different processes
- One can easily factorize the likelihood:

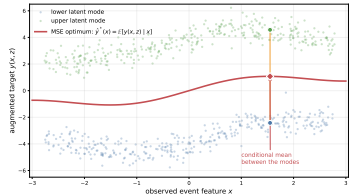


One has a likelihood ratio

$$r(x, \boldsymbol{\theta}, \mathrm{SM}) = \frac{n_1(s_1 + l_1\boldsymbol{\theta} + q_1\boldsymbol{\theta}^2) + n_2(s_2 + l_2\boldsymbol{\theta} + q_2\boldsymbol{\theta}^2)}{n_1 s_1 + n_2 s_2}$$

Factorization will help you!

- Multimodal behaviours and other instabilities may come in the presence of different processes
- One can easily factorize the likelihood:



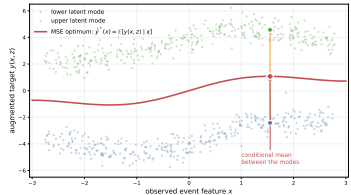
One has a likelihood ratio

$$r(x, \boldsymbol{\theta}, \mathrm{SM}) = \frac{n_1(s_1 + l_1\boldsymbol{\theta} + q_1\boldsymbol{\theta}^2) + n_2(s_2 + l_2\boldsymbol{\theta} + q_2\boldsymbol{\theta}^2)}{n_1 s_1 + n_2 s_2}$$

$$r(x, \boldsymbol{\theta}, \mathrm{SM}) = n_1 \frac{1 + \frac{l_1}{s_1}\boldsymbol{\theta} + \frac{q_1}{s_1}\boldsymbol{\theta}^2}{n_1 + n_2\frac{s_2}{s_1}} + n_2 \frac{1 + \frac{l_2}{s_2}\boldsymbol{\theta} + \frac{q_2}{s_2}\boldsymbol{\theta}^2}{n_2 + n_1\frac{s_1}{s_2}}$$

Factorization will help you!

- Multimodal behaviours and other instabilities may come in the presence of different processes
- One can easily factorize the likelihood:



One has a likelihood ratio

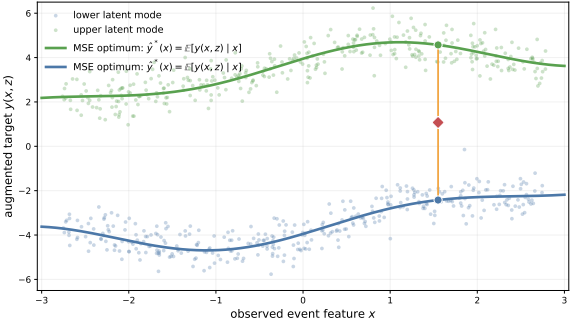
$$r(x, \boldsymbol{\theta}, \mathrm{SM}) = \frac{n_1(s_1 + l_1\boldsymbol{\theta} + q_1\boldsymbol{\theta}^2) + n_2(s_2 + l_2\boldsymbol{\theta} + q_2\boldsymbol{\theta}^2)}{n_1 s_1 + n_2 s_2}$$

$$r(x, \boldsymbol{\theta}, \mathrm{SM}) = n_1 \frac{1 + \frac{l_1}{s_1}\boldsymbol{\theta} + \frac{q_1}{s_1}\boldsymbol{\theta}^2}{n_1 + n_2\frac{s_2}{s_1}} + n_2 \frac{1 + \frac{l_2}{s_2}\boldsymbol{\theta} + \frac{q_2}{s_2}\boldsymbol{\theta}^2}{n_2 + n_1\frac{s_1}{s_2}}$$

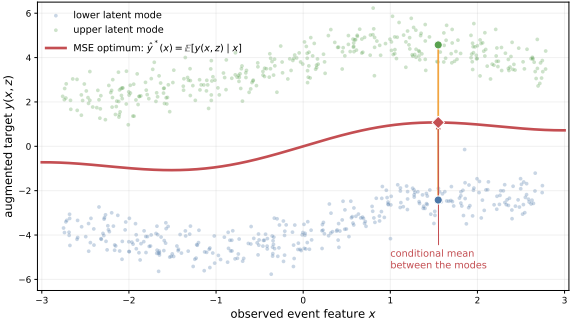
- l_i/s_i, q_i/s_i are what you need to obtain for the two processes *separately*
- s_1/s_2 can be obtained with a classifier between the two processes

Factorization will help you

Factorized training



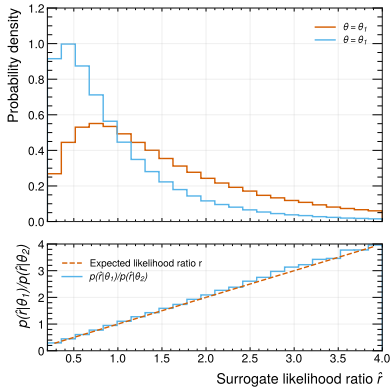
Unified training



- Mathematically, the two procedures are the same, but factorization is less prone to instabilities

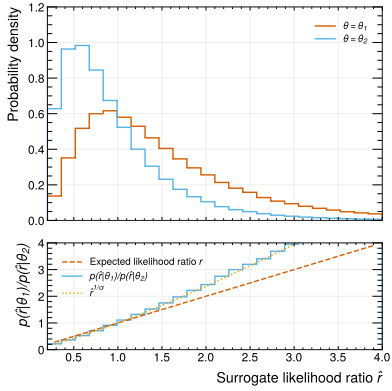
Calibration

- One way to make sure the training has converged is **checking that the surrogate likelihood ratio is distributed as a likelihood ratio**



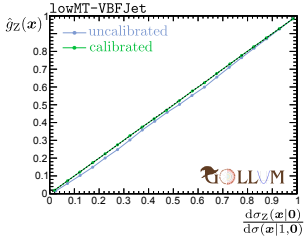
Calibration

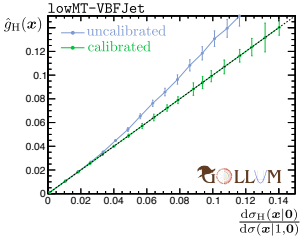
- One way to make sure the training has converged is **checking that the surrogate likelihood ratio is distributed as a likelihood ratio**
- Modern discriminators may not converge perfectly to the likelihood ratio
- One can correct for this effect post-training checking the distribution



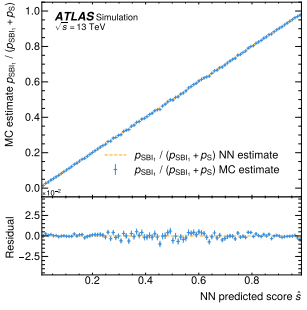
A few examples of real life calibrations

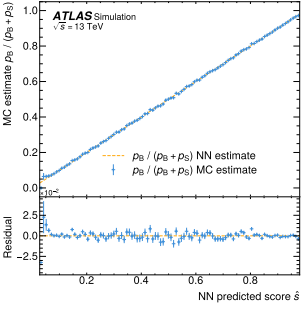
arXiv:2505.05544





arXiv:2412.01600





How does it work in practice

- First attempts are often hard, but don't give up!
- Building toy examples were things are under control is extremely useful both to understand the method and test the implementation
- Remember that, in cases, the latent space is large and the integration won't be trivial... This is not a textbook $y = f(x)$ regression

What do we have so far?

So far, we have obtained a surrogate model for $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta}_0)$ for "physical" cases, like EFT or the κ framework

$\boldsymbol{\eta}_0$ is the nominal value of the nuisances

Remember, our original goal was to get $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta})$, so we can do inference using

$$\text{Poisson}(N|\nu(\boldsymbol{\theta}, \boldsymbol{\eta})) \prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta}, \boldsymbol{\eta}) \, \pi(\boldsymbol{\eta})$$

What do we have so far?

So far, we have obtained a surrogate model for $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta}_0)$ for "physical" cases, like EFT or the κ framework

$\boldsymbol{\eta}_0$ is the nominal value of the nuisances

Remember, our original goal was to get $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta})$, so we can do inference using

$$\text{Poisson}(N|\nu(\boldsymbol{\theta}, \boldsymbol{\eta})) \prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta}, \boldsymbol{\eta}) \, \pi(\boldsymbol{\eta})$$

We are not done yet but $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta}_0)$ is already useful!

Examples of SBI without systematics

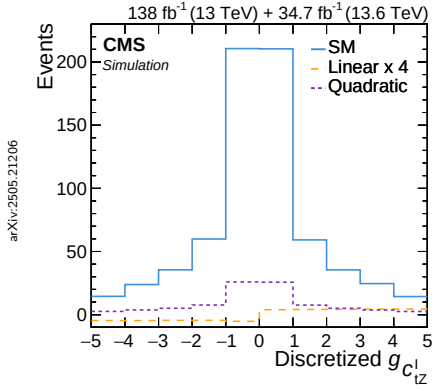
SBI targeting interference

Remember: Interference terms are the leading order in the EFT expansion

$$\mathcal{M} = \mathcal{M}_{\mathrm{SM}} + \sum_i \frac{c_i}{\Lambda^2} \mathcal{M}_i$$

$$|\mathcal{M}|^2 = \underbrace{|\mathcal{M}_{\mathrm{SM}}|^2}_{\sigma_{\mathrm{SM}}} + \underbrace{\sum_i c_i \, 2\,\mathrm{Re}(\mathcal{M}_{\mathrm{SM}} \mathcal{M}_i^*)}_{\sum_i c_i \sigma_i} + \underbrace{\sum_{i,j} c_i c_j \, \mathcal{M}_i \mathcal{M}_j^*}_{\sum_{i,j} c_i c_j \sigma_{ij}} .$$

- SALLY is able to regress the likelihood score
- The score is a sufficient statistics of the vicinity of a reference value
- When picking $c_i = 0$, SALLY is the best discriminator for the *interference*



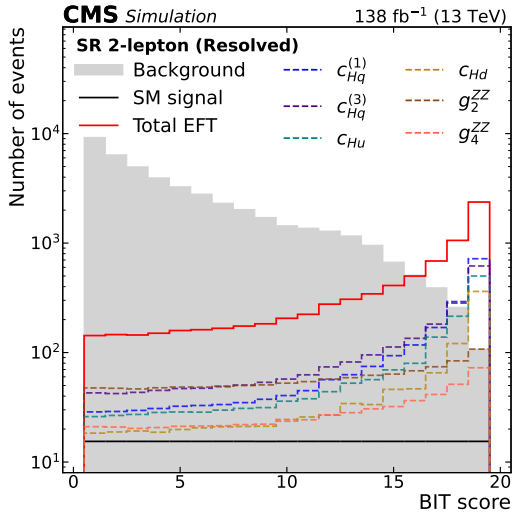
- Approach followed in e.g. arXiv:2505.21206 by CMS

Examples of SBI without systematics

SBI to build parametric optimal observables

- $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta}_0)$ is the **optimal classifier** for each value of $\boldsymbol{\theta}$
- In practice, one needs settle on a max few 1-dimensional classifiers
- However, having the explicit dependence of $r(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\eta}_0)$ is useful
- arXiv:2411.16907 by CMS uses this explicit dependence to settle on a "good" discriminator that works well for everything

arXiv:2411.16907



Section 5

Towards an unbinned analysis

Towards an unbinned analysis

Our final objective is

$$\text{Poisson}(N|\nu(\boldsymbol{\theta},\boldsymbol{\eta}))\prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})\ \pi(\boldsymbol{\eta})$$

Towards an unbinned analysis

Our final objective is

$$\text{Poisson}(N|\nu(\boldsymbol{\theta},\boldsymbol{\eta}))\prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})\ \pi(\boldsymbol{\eta})$$

So far, we have:

- The Poisson term $\Rightarrow$ "simple", the same as in your favourite analysis
- For some cases, we have $p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}_0)$

Towards an unbinned analysis

Our final objective is

$$\text{Poisson}(N|\nu(\boldsymbol{\theta},\boldsymbol{\eta}))\prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})\ \pi(\boldsymbol{\eta})$$

So far, we have:

- The Poisson term $\Rightarrow$ "simple", the same as in your favourite analysis
- For some cases, we have $p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}_0)$

But we are missing the dependence on the systematic uncertainties

Towards an unbinned analysis

Our final objective is

$$\text{Poisson}(N|\nu(\boldsymbol{\theta},\boldsymbol{\eta}))\prod_{e=1}^{N} p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})\ \pi(\boldsymbol{\eta})$$

So far, we have:

- The Poisson term $\Rightarrow$ "simple", the same as in your favourite analysis
- For some cases, we have $p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}_0)$

But we are missing the dependence on the systematic uncertainties

- This dependence may be completely arbitrary
- One could train a parametric network to learn that
- but it's unfeasible/tricky with $\mathcal{O}(100)$ systematic uncertainties
- We cannot play the same trick as in the EFT case

Likelihoods with systematic uncertainties

Once again, factorization can help us

$$r(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}) = \frac{p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta}_0)} = \frac{p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta})}\frac{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta}_0)}$$

Likelihoods with systematic uncertainties

Once again, factorization can help us

$$r(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}) = \frac{p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta}_0)} = \frac{p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta})}\frac{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta}_0)}$$

$$\simeq \underbrace{\frac{p(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}_0)}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta}_0)}}_{r(\boldsymbol{x}_e|\boldsymbol{\theta},\boldsymbol{\eta}_0)} \underbrace{\frac{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta})}{p(\boldsymbol{x}_e|\boldsymbol{\theta}_{SM},\boldsymbol{\eta}_0)}}_{s(\boldsymbol{x}_e|\boldsymbol{\eta})}$$

Often, we can assume that uncertainties (e.g. detector response) are independent of the POIs.

Likelihoods with systematic uncertainties

The quantity $r(\boldsymbol{x}_e|\boldsymbol{\theta}, \boldsymbol{\eta}_0)$ we already have it, but we need $s(\boldsymbol{x}_e|\boldsymbol{\eta})$

Likelihoods with systematic uncertainties

The quantity $r(\boldsymbol{x}_e|\boldsymbol{\theta}, \boldsymbol{\eta}_0)$ we already have it, but we need $s(\boldsymbol{x}_e|\boldsymbol{\eta})$

We can further simplify the task by factorizing into uncorrelated systematic uncertainties

$$s(\boldsymbol{x}_e|\boldsymbol{\eta}) = \prod_k s_k(\boldsymbol{x}_e|\boldsymbol{\eta}_k)$$

where $\boldsymbol{\eta}_k$ are (manageable) groups of systematic uncertainties

Likelihoods with systematic uncertainties

To obtain $s_k(\boldsymbol{x}|\boldsymbol{\eta})$ one can use an Ansatz, e.g.

$$s_k(\boldsymbol{x}|\boldsymbol{\eta}) = \exp\left(\sum_{\text{poly}} \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})\right)$$

- $\sum_{\text{poly}} \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})$ is a multinomial on $\boldsymbol{\eta}_k$

Likelihoods with systematic uncertainties

To obtain $s_k(\boldsymbol{x}|\boldsymbol{\eta})$ one can use an Ansatz, e.g.

$$s_k(\boldsymbol{x}|\boldsymbol{\eta}) = \exp\left(\sum_{\text{poly}} \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})\right)$$

- $\sum_{\text{poly}} \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})$ is a multinomial on $\boldsymbol{\eta}_k$
- This expression allows to have an arbitrarily defined set of variations around the nominal
- The exponential is useful to impose that the likelihood ratio is positive

Likelihoods with systematic uncertainties

To obtain $s_k(\boldsymbol{x}|\boldsymbol{\eta})$ one can use an Ansatz, e.g.

$$s_k(\boldsymbol{x}|\boldsymbol{\eta}) = \exp\left(\sum_{\text{poly}} \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})\right)$$

- $\sum_{\text{poly}} \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})$ is a multinomial on $\boldsymbol{\eta}_k$
- This expression allows to have an arbitrarily defined set of variations around the nominal
- The exponential is useful to impose that the likelihood ratio is positive
- Δ_k can be obtained training discriminators between nominal and smeared simulations

Wrapping up

We are now ready to write the likelihood ratio:

$$r = \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta}_0)}{p(\boldsymbol{x} \mid \theta_{SM}, \boldsymbol{\eta}_0)} \prod_k \exp\left(\sum \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})\right)$$

Wrapping up

We are now ready to write the likelihood ratio:

$$r = \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta}_0)}{p(\boldsymbol{x} \mid \theta_{SM}, \boldsymbol{\eta}_0)} \prod_k \exp\left(\sum \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})\right)$$

- The term in blue describes the dependence of the probability density on the POIs
  - $N_{\text{proc}} - 1$ discriminants, where N_{proc} is the number of processes
  - At least one discriminant per POI
  - For κ-frameworks and EFT, the number of discriminators grows quadratically

Wrapping up

We are now ready to write the likelihood ratio:

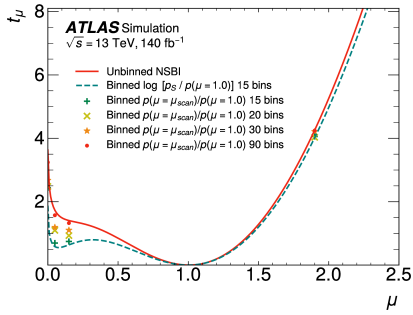
$$r = \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\eta}_0)}{p(\boldsymbol{x} \mid \theta_{SM}, \boldsymbol{\eta}_0)} \prod_k \exp\left(\sum \boldsymbol{\eta}_k \Delta_k(\boldsymbol{x})\right)$$

- The term in blue describes the dependence of the probability density on the POIs
  - $N_{\text{proc}} - 1$ discriminants, where N_{proc} is the number of processes
  - At least one discriminant per POI
  - For κ-frameworks and EFT, the number of discriminators grows quadratically
- The term in red describes the dependence on the nuisance parameters
  - At least one discriminant per nuisance parameter
  - Two networks for the up/down variations
  - Additional networks may be needed for correlated systematics
  - Process-dependent systematics can also be included

Sensitive in realistic examples

Off-shell Higgs production (arXiv:2412.01600)

- Single discriminant has a degenerated structure, from the interference
- Surrogate likelihood provides the best sensitivity
- Binned analyses reach unbinned sensitivity (only) with a fine binned



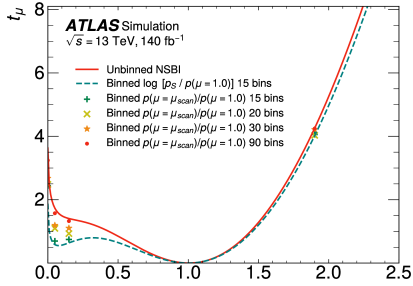
Sensitive in realistic examples

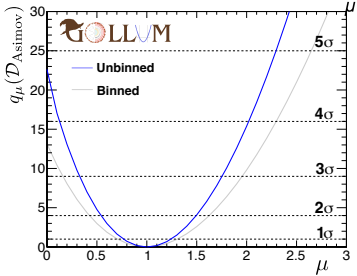
Off-shell Higgs production (arXiv:2412.01600)

- Single discriminant has a degenerated structure, from the interference
- Surrogate likelihood provides the best sensitivity
- Binned analyses reach unbinned sensitivity (only) with a fine binned

$\mathrm{H} \to \tau\tau$ **inclusive cross section (arXiv:2505.05544)**

- Improvement of sensitivity of 20% when going from binned to unbinned
- Unbinned allows to decouple signal from the systematic uncertainties





Summary

- Presented new tools for analysis techniques, which more efficiently allow to use the information contained in data
- Surrogate likelihood models can be used to construct optimal observables in "complex" physics models
- Surrogate models *including systematic uncertainties* provide the optimal summary statistic
- Need a careful work of training, validation, curation, etc

Summary

- Presented new tools for analysis techniques, which more efficiently allow to use the information contained in data
- Surrogate likelihood models can be used to construct optimal observables in "complex" physics models
- Surrogate models *including systematic uncertainties* provide the optimal summary statistic
- Need a careful work of training, validation, curation, etc

Thanks for your attention!

For reading

Quite a lof of developments in the last years, here's a (non-exhaustive) list of starting points

- ATLAS Collaboration (2025). "An implementation of neural simulation-based inference for parameter estimation in ATLAS". In: *Reports on Progress in Physics*. DOI: 10.1088/1361-6633/add370. arXiv: 2412.01600 [physics.data-an].
- Benato, Lisa et al. (2025). "Unbinned inclusive cross-section measurements with machine-learned systematic uncertainties". In: *Physical Review D*. DOI: 10.1103/zwzt-1rrw. arXiv: 2505.05544 [hep-ph].
- Brehmer, Johann et al. (2020). "Mining gold from implicit models to improve likelihood-free inference". In: *Proceedings of the National Academy of Sciences*. DOI: 10.1073/pnas.1915980117. arXiv: 1805.12244 [stat.ML].
- Schöfbeck, Robert (2025). "Refinable modeling for unbinned SMEFT analyses". In: *Machine Learning: Science and Technology*. DOI: 10.1088/2632-2153/ad9fd1. arXiv: 2406.19076 [hep-ph].