MadMiner: Likelihood-free machine learning inference

Davide Valsecchi

PhD School PREFIT 2020 seminar
06/05/2020

Effective field theory

$$\mathcal{L}_{\mathrm{EFT}} = \mathcal{L}_{\mathrm{SM}} + \sum_i \frac{f_i}{\Lambda^2}\,\mathcal{O}_i + \ldots$$

Operators

- all possible interactions between SM particles mediated by new physics
- fixed by SM particles + SM symmetries + expansion in $1/\Lambda$, independent of high-energy physics
- affect rates + kinematics

$$\mathcal{O}_{\phi,1} = (D_\mu\phi)^\dagger \phi\, \phi^\dagger D^\mu \phi \qquad \mathcal{O}_{GG} = (\phi^\dagger\phi)\, G^a_{\mu\nu} G^{\mu\nu\, a}$$

$$\mathcal{O}_{\phi,2} = \frac{1}{2}\partial_\mu(\phi^\dagger\phi)\,\partial^\mu(\phi^\dagger\phi) \qquad \mathcal{O}_{BB} = -\frac{g'^2}{4}(\phi^\dagger\phi)\, B_{\mu\nu} B^{\mu\nu}$$

$$\mathcal{O}_{\phi,3} = \frac{1}{3}(\phi^\dagger\phi)^3 \qquad \mathcal{O}_{WW} = -\frac{g^2}{4}(\phi^\dagger\phi)\, W^a_{\mu\nu} W^{\mu\nu\, a}$$

$$\mathcal{O}_{\phi,4} = (\phi^\dagger\phi)\,(D_\mu\phi)^\dagger D^\mu\phi \qquad \mathcal{O}_{BW} = -\frac{g\,g'}{4}(\phi^\dagger\sigma^a\phi)\, B_{\mu\nu} W^{\mu\nu\, a}$$

$$\mathcal{O}_{B} = \frac{ig'}{2}(D^\mu\phi)^\dagger D^\nu\phi\, B_{\mu\nu}$$

$$\mathcal{O}_{W} = \frac{ig}{2}(D^\mu\phi)^\dagger \sigma^a D^\nu\phi\, W^a_{\mu\nu}$$

Wilson coefficients

- precise measurement of these parameters is one of the most important goals of the LHC
- can be translated to high-energy physics parameters

Higher-order terms

- suppressed by additional factors of $E^2 \,/\, \Lambda^2$

Inference is the key

Likelihood ratio

The Neyman-Pearson lemma states that the **likelihood ratio** test between two simple hypothesis H_0 (null hypothesis) and H_1 (alternate hypothesis) is the most powerful test for a given significance level α.

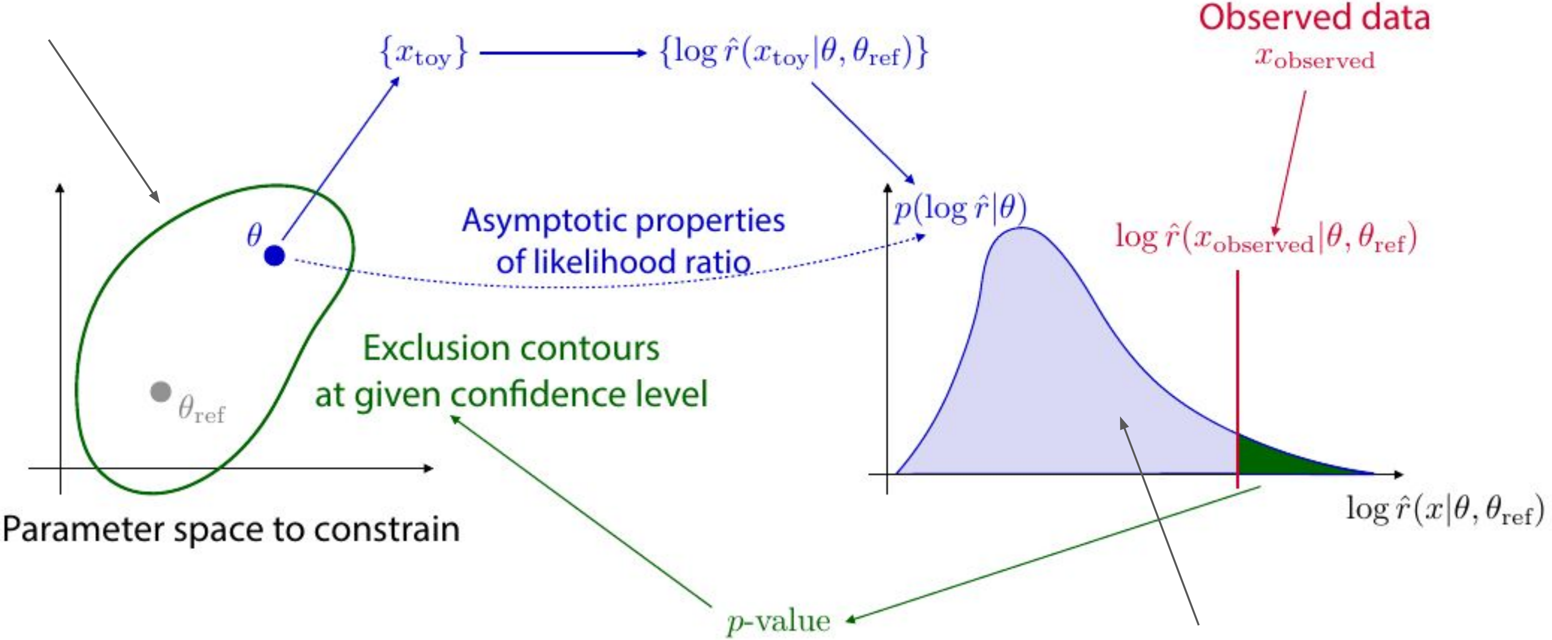
The likelihood ratio is the main **test statistics** used in LHC experiments for:

- Observation of new signals
- Limits extraction for signal models
- Signal strength measurements

Limit setting (frequentist, LHC experiments)

EFT parameters space

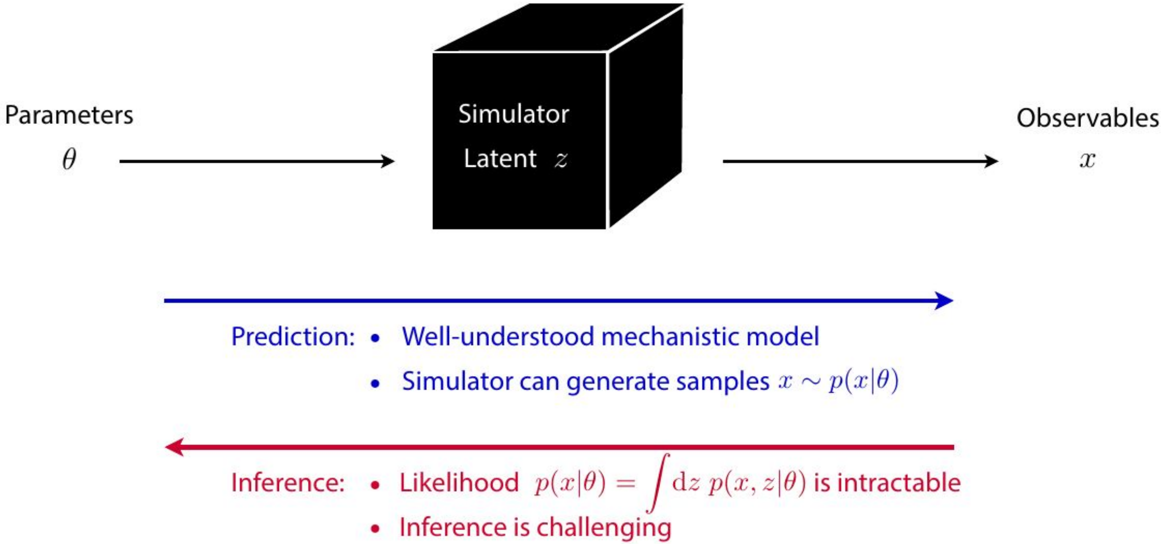
likelihood ratio



likelihood ratio distribution by toys

Simulation-based "likelihood-free" inference

- To extract a **measurement** (of EFT parameters) from **data**, a **statistical model** is needed.
- In HEP powerful MonteCarlo tools are used to build predictions, stacking several processes on top of each other:
  - hard parton interaction, UE, parton shower, hadronization, detector showering, reconstruction...



The likelihood ratio is intractable

x are the observables, after shower, detector, and reconstruction, θ are the parameters of interest, *z* are the parton-level momenta.

matrix element calculation

$$p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta) = \int \mathrm{d}z \; p(x|z) \, p(z|\theta)$$

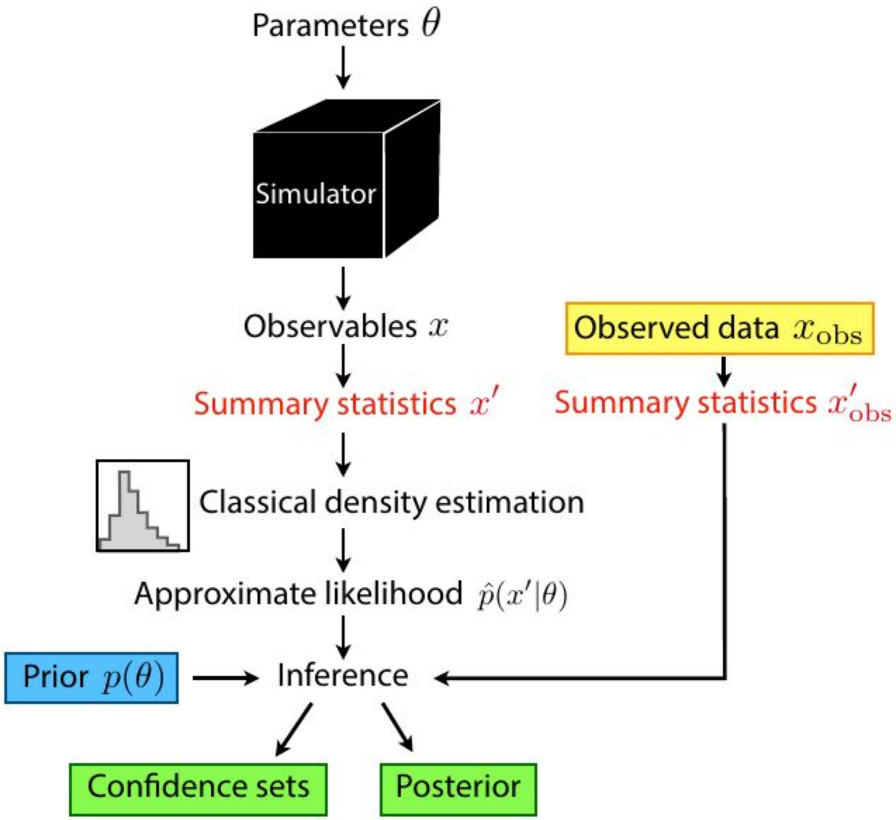
$$\boxed{p(z|\theta) = \frac{1}{\sigma(\theta)} \frac{\mathrm{d}\sigma(\theta)}{\mathrm{d}z} \,,}$$

$$p(x|z) = \int \mathrm{d}z_{\mathrm{detector}} \int \mathrm{d}z_{\mathrm{shower}} \; p(x|z_{\mathrm{detector}}) \; p(z_{\mathrm{detector}}|z_{\mathrm{shower}}) \; p(z_{\mathrm{shower}}|z)$$

The explicit likelihood function is intractable because it involves integrals over all the possible paths of the simulation that can involve millions of random number.

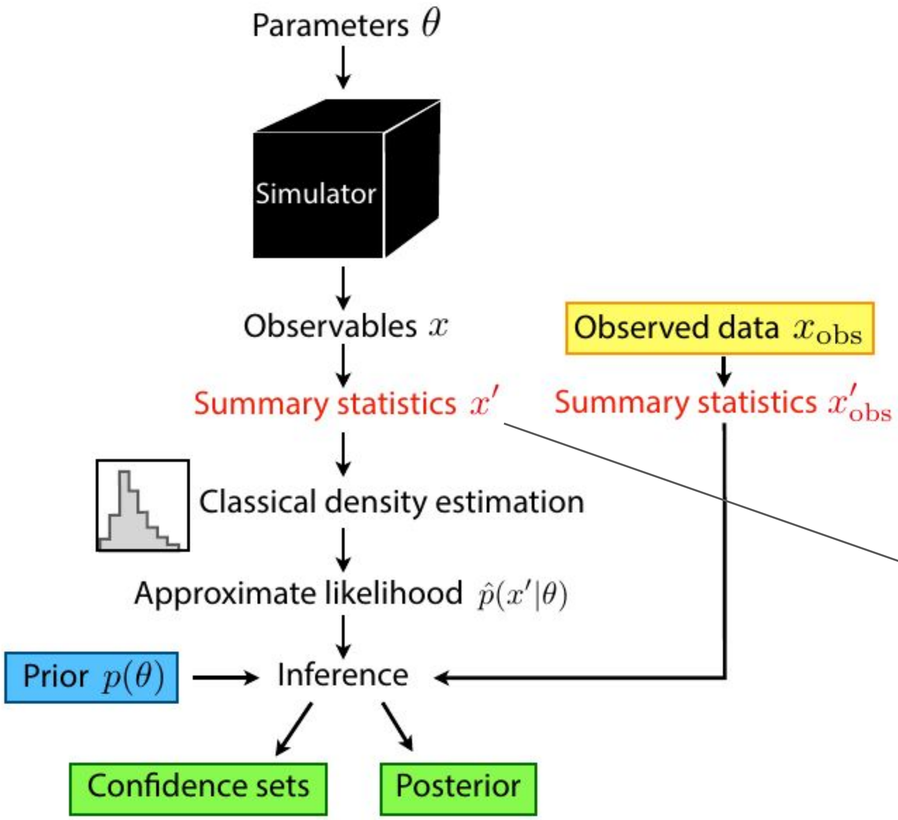
The matrix element calculation istead is tractable thanks to tools like Madgraph, Madmax and morphing (more details laters).

Standard procedure

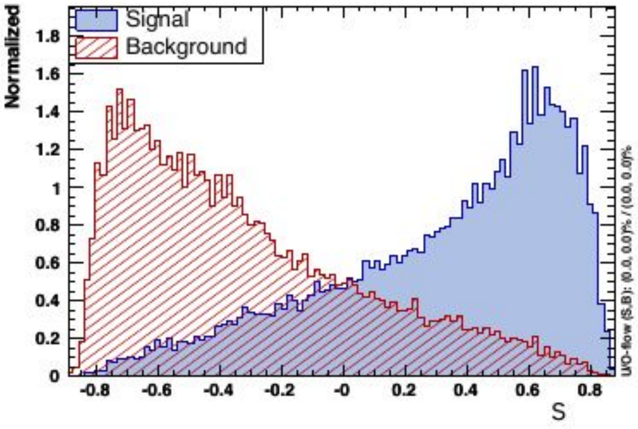


- The standard procedure is to choose one (or two) kinematical observables as **summary statics.**
- From MC, an estimator for the likelihood ratio is built from **histograms** (basic density estimation method)
- Loss of information in compression to summary statistics

“MVA approach”



- MultiVariate Analysis techniques can be used to build summary statistics using lots of kinematical features:
  - Boosted Decision Trees (BDT)
  - Support Vector Machines (SVM)
  - Neural Networks (NN)
- Usually they are used to build histograms and then estimate likelihood ratio for signal/background identification



Likelihood ratio trick

- **Binary classifiers** (such as Deep Neural Networks) are trained minimizing the cross-entropy loss using on training dataset $\{x_e\}$

$$L[\hat{s}] = -\frac{1}{N}\sum_e \left(y_e \log \hat{s}(x_e) + (1-y_e)\log(1-\hat{s}(x_e))\right)$$

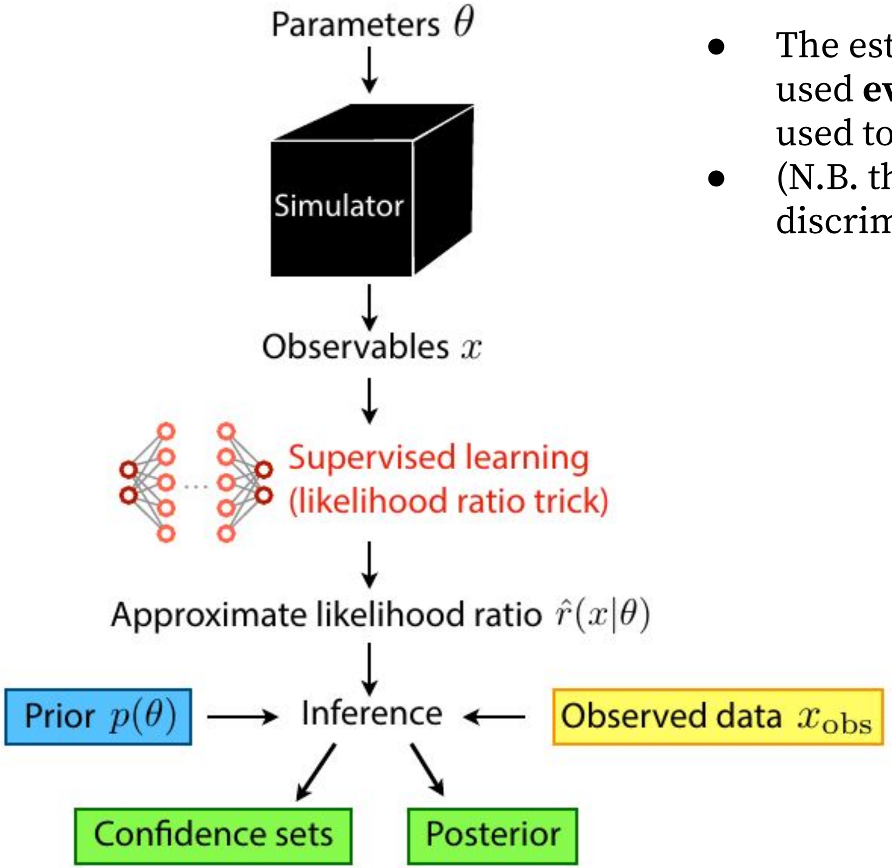
- The optimal decision function that is regressed is: $$s(x|\theta_0,\theta_1) = \frac{p(x|\theta_1)}{p(x|\theta_0)+p(x|\theta_1)}$$

- An **estimator** for the likelihood ratio is therefore:

$$\boxed{\hat{r}(x|\theta_0,\theta_1) = \frac{1-\hat{s}(x|\theta_0,\theta_1)}{\hat{s}(x|\theta_0,\theta_1)}}$$

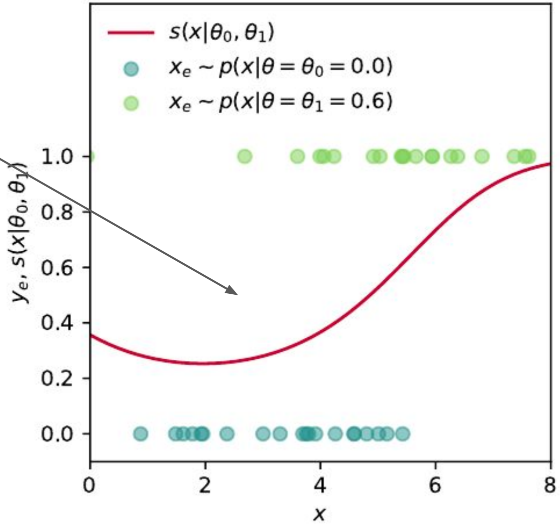
Likelihood ratio “trick”

Inference by likelihood ratio trick



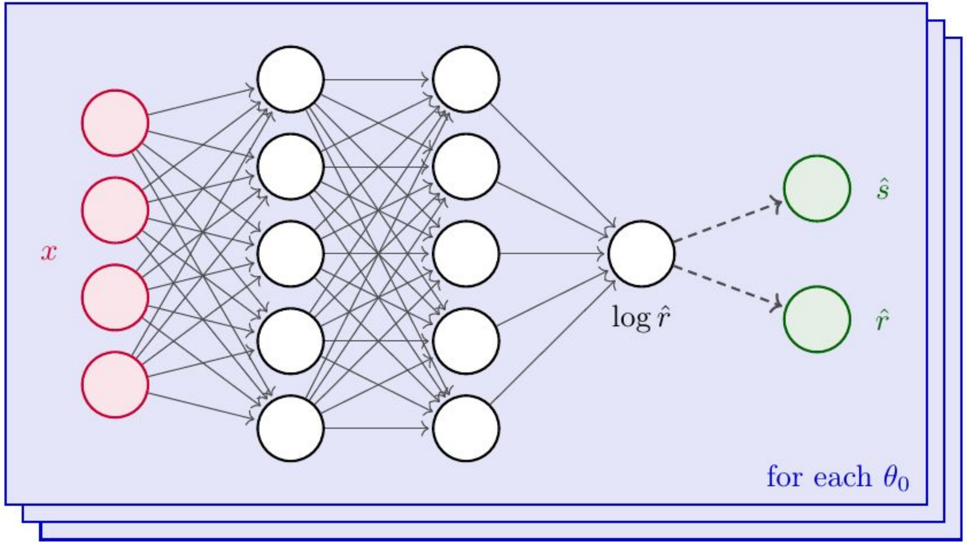
- The estimator for the likelihood ratio is then used **event by event** for inference directly, **not** used to build a summary statistics
- (N.B. the goal is not to do signal vs background discrimination but inference on θ parameters)

optimal decision function



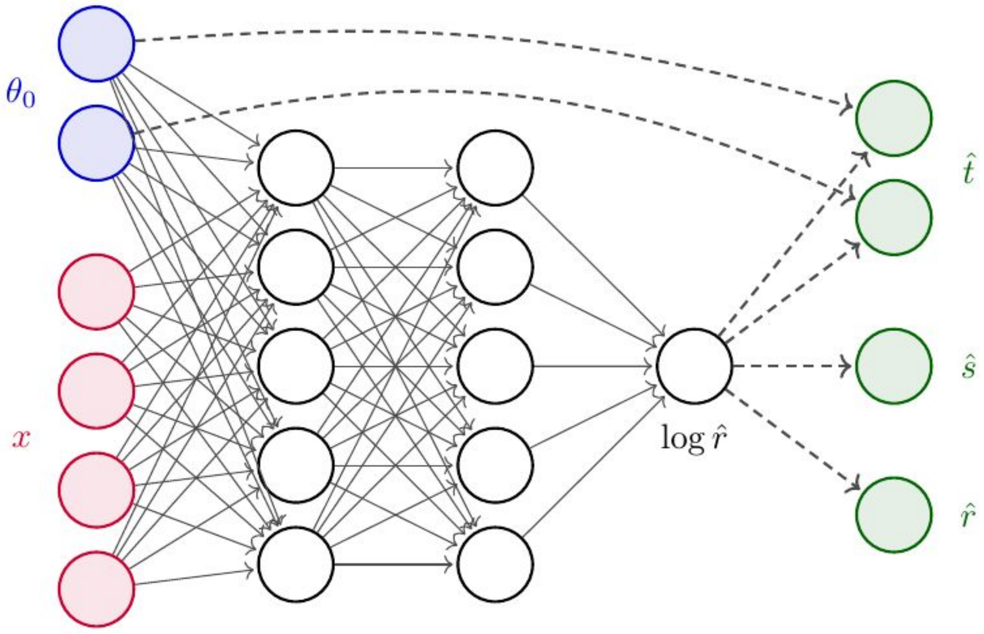
How to handle the parameters space (1)

- **Point by Point:** Scan the parameter space randomly or with a grid. Usually fix θ_1 as a reference (SM) and scan θ_2. For each pair (θ_1, θ_2) an estimator for the likelihood ratio $\hat{r}(x|\theta_0, \theta_1)$ is regressed from MC samples extracted with these parameters. The final results are interpolated between different parameter space points

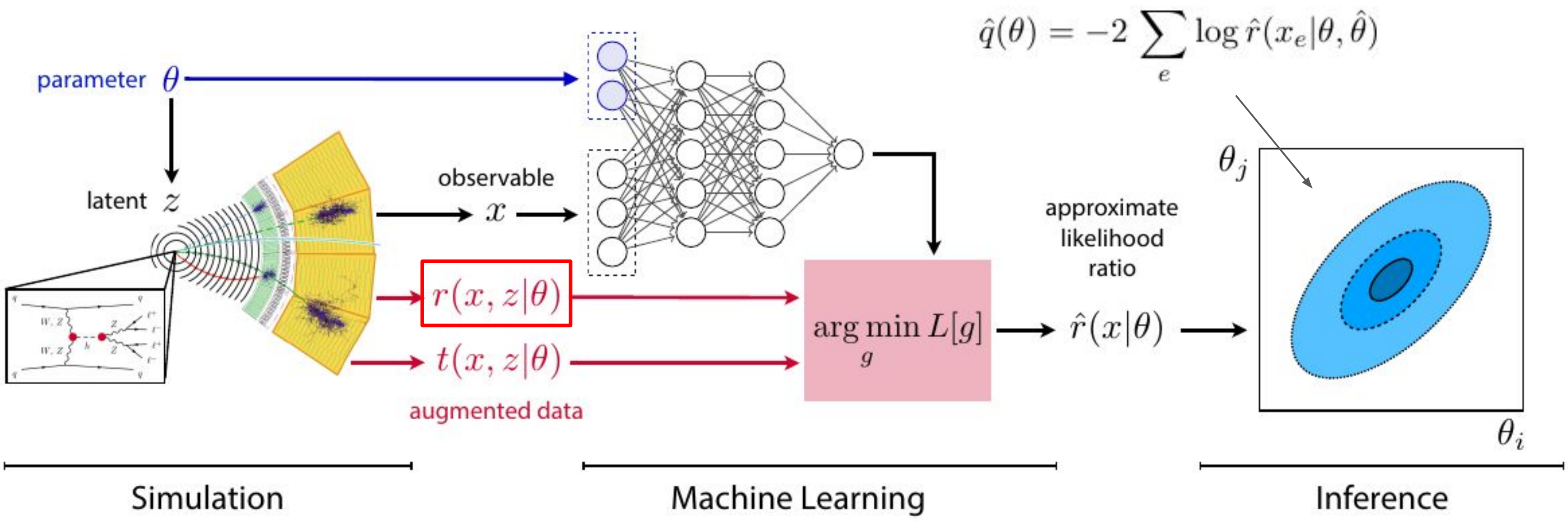


How to handle the parameters space (2)

- **Agnostic parameterized estimators:** The likelihood ratio is estimated as the full model $\hat{r}(x|\theta_0, \theta_1)$ a function of both *x* and (θ_1, θ_2). The estimator can learn the smooth dependence of the likelihood ratio on the physics parameters and does not require any interpolation in the end.



Learning with augmented data



"Mining gold": Extract additional information from simulator

Use this information to train estimator for likelihood ratio

Limit setting with standard hypothesis tests

Information can be extracted from the simulator

Latent variables

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
x	z_d	z_s	z_p	θ

$$x \longleftarrow z_d \longleftarrow z_s \longleftarrow z_p \longleftarrow \theta$$

$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad \boxed{p(z_p|\theta)}$$

Parton-level likelihood
is given by matrix element
and can be evaluated!

A **joint likelihood ratio** is defined on the each specific event and depends only on parton momenta and θ parameters

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)} = \frac{p(x|z_d)}{p(x|z_d)} \frac{p(z_d|z_s)}{p(z_d|z_s)} \frac{p(z_s|z_p)}{p(z_s|z_p)} \boxed{\frac{p(z_p|\theta_0)}{p(z_p|\theta_1)} \sim \frac{|\mathcal{M}(z_p|\theta_0)|^2}{|\mathcal{M}(z_p|\theta_1)|^2}}$$

Unfortunately integral of ratio != ratio of integral but we can use this information.

$$L[\hat{g}(x)] = \int \mathrm{d}x\, \mathrm{d}z\; p(x, z|\theta)\; |g(x, z) - \hat{g}(x)|^2$$
$$= \int \mathrm{d}x\; \underbrace{\left[\hat{g}^2(x) \int \mathrm{d}z\; p(x, z|\theta) - 2\hat{g}(x) \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z) + \int \mathrm{d}z\; p(x, z|\theta)\, g^2(x, z)\right]}_{F(x)} .$$

Functional for function *g(x)* that tries to approximate function *g(x,z)*

$$0 = \left.\frac{\delta F}{\delta \hat{g}}\right|_{g^*} = 2\hat{g} \underbrace{\int \mathrm{d}z\; p(x, z|\theta)}_{=p(x|\theta)} - 2 \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z) \longrightarrow g^*(x) = \frac{1}{p(x|\theta)} \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z) .$$

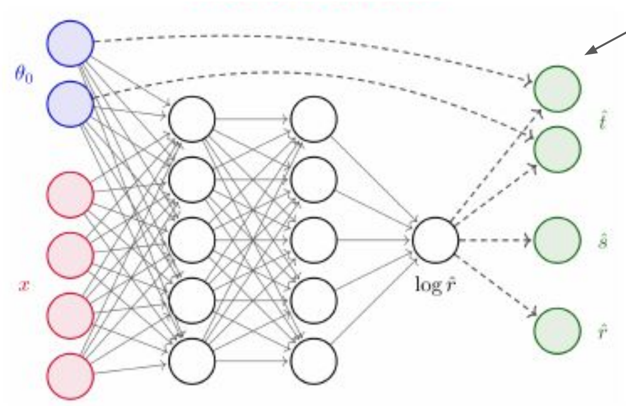
Identifying *g(x,z)* with the join likelihood ratio that we can calculate we find the that the function that minimizes the functional L is the **true likelihood ratio**

$$g^*(x) = \frac{1}{p(x|\theta_1)} \int \mathrm{d}z\; p(x, z|\theta_1)\, \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} = r(x|\theta_0, \theta_1)$$

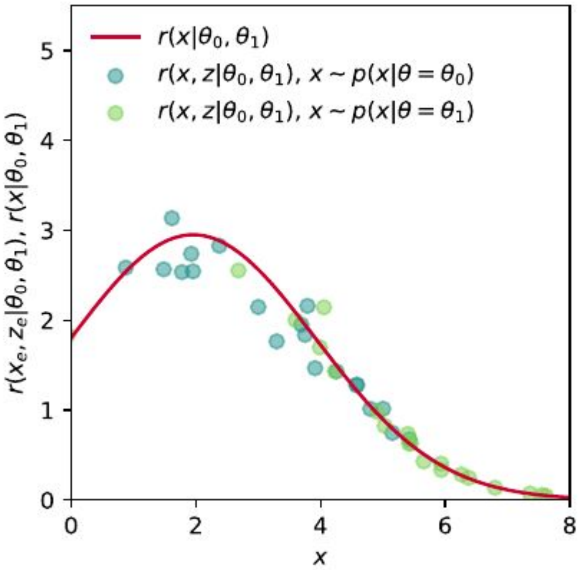
Likelihood ratio augmented estimator

In **practice,** the estimator for the likelihood rati $\hat{r}(x|\theta_0,\theta_1)$ is built with a sufficiently expressive function (a neural network) minimizing the loss over all the events (x_e, z_e) sampled according the denominator hypothesis $\theta_{1.}$

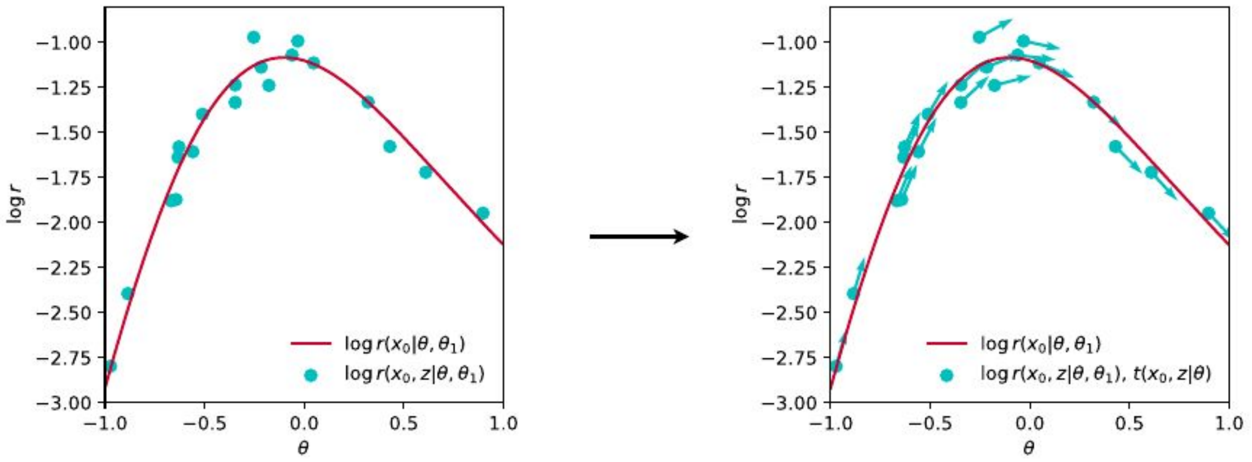
$$L[\hat{r}(x|\theta_0,\theta_1)] = \frac{1}{N} \sum_{(x_e,z_e)\sim p(x,z|\theta_1)} |r(x_e, z_{\mathrm{all},e}|\theta_0,\theta_1) - \hat{r}(x_e|\theta_0,\theta_1)|^2$$



$r(x,z|\theta_0,\theta_1)$ are scattered around $r(x|\theta_0,\theta_1)$



More augmented data: the score



"score"

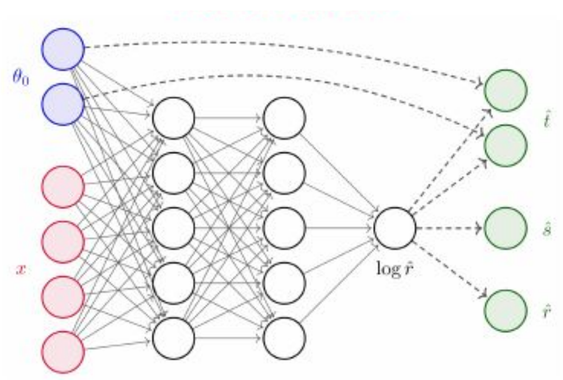
$$t(x|\theta_0) = \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$$

The score quantifies the relative change of the likelihood under infinitesimal changes in the parameter space. The true score is intractable, but not the joint score.

$$\begin{aligned} t(x_e, z_{\text{all}\,e}|\theta_0) &\equiv \nabla_\theta \log p(x_e, z_{\text{detector}\,e}, z_{\text{shower}\,e}, z_e|\theta_0) \\ &= \frac{p(x_e|z_{\text{detector}\,e})}{p(x_e|z_{\text{detector}\,e})} \frac{p(z_{\text{detector}\,e}|z_{\text{shower}\,e})}{p(z_{\text{detector}\,e}|z_{\text{shower}\,e})} \frac{p(z_{\text{shower}\,e}|z_e)}{p(z_{\text{shower}\,e}|z_e)} \frac{\nabla_\theta p(z_e|\theta)}{p(z_e|\theta)}\Bigg|_{\theta_0} \\ &= \frac{\nabla_\theta p(z_e|\theta)}{p(z_e|\theta)}\Bigg|_{\theta_0} \end{aligned}$$

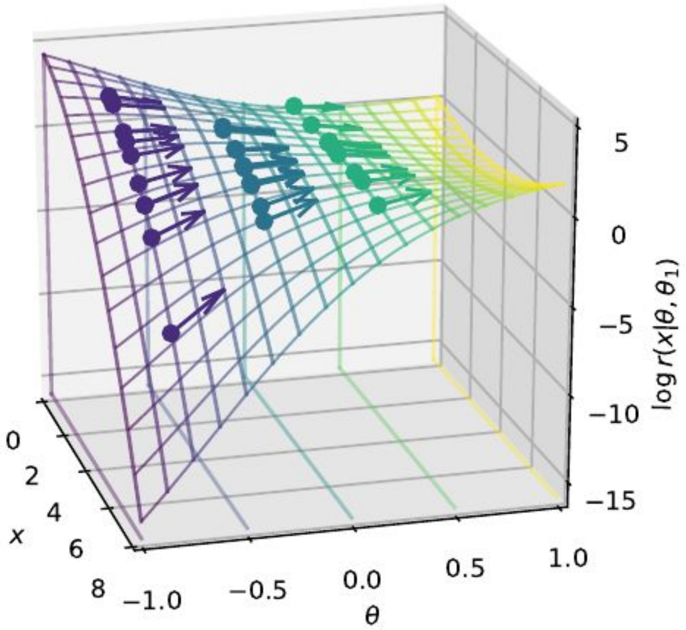
The reasoning in slide 15 applies also for the joint score, so we can build an estimator for the true score using the joint score extracted for each event in the simulation

$$L[\hat{t}(x|\theta)] = \frac{1}{N} \sum_{(x_e, z_e) \sim p(x,z|\theta)} |t(x_e, z_{\text{all},e}|\theta) - \hat{t}(x_e|\theta)|^2$$



In order to be able to be able to take advantage of the score the estimator for the likelihood ratio needs to be **differentiable** with respect of the parameters θ.

Neural networks using θ as inputs are a natural solution for this problem



Loss functions

Calibrated classifiers (CARL): Use standard cross-entropy loss for binary classification to regress a decision function $\hat{s}(x|\theta_0, \theta_1)$ with samples generated according θ_1 and θ_0 . Then extract an estimator for the likelihood ratio

$$\hat{r}(x|\theta_0, \theta_1) = \frac{1 - \hat{s}(x|\theta_0, \theta_1)}{\hat{s}(x|\theta_0, \theta_1)}$$

Ratio regression (ROLR): direct regression for likelihood ratio $\hat{r}(x|\theta_0, \theta_1)$ sampling from θ_1 for $y_e = 1$ and from θ_0 for $y_e = 0$

$$L[\hat{r}(x|\theta_0, \theta_1)] = \frac{1}{N} \sum_{(x_e, z_e, y_e)} \left(y_e \left| r(x_e, z_e|\theta_0, \theta_1) - \hat{r}(x|\theta_0, \theta_1) \right|^2 + (1 - y_e) \left| \frac{1}{r(x_e, z_e|\theta_0, \theta_1)} - \frac{1}{\hat{r}(x|\theta_0, \theta_1)} \right|^2 \right)$$

Ratio + score regression (RASCAL): implement a regressor with a differentiable architecture in order to extract the score. Minimize combined loss:

$$L[\hat{r}(x|\theta_0, \theta_1)] = \frac{1}{N} \sum_{(x_e, z_e, y_e)} \left[y_e \left| r(x_e, z_e|\theta_0, \theta_1) - \hat{r}(x_e|\theta_0, \theta_1) \right|^2 + (1 - y_e) \left| \frac{1}{r(x_e, z_e|\theta_0, \theta_1)} - \frac{1}{\hat{r}(x_e|\theta_0, \theta_1)} \right|^2 + \alpha \, (1 - y_e) \left| t(x_e, z_e|\theta_0) - \hat{t}(x_e|\theta_0) \right|^2 \right]$$

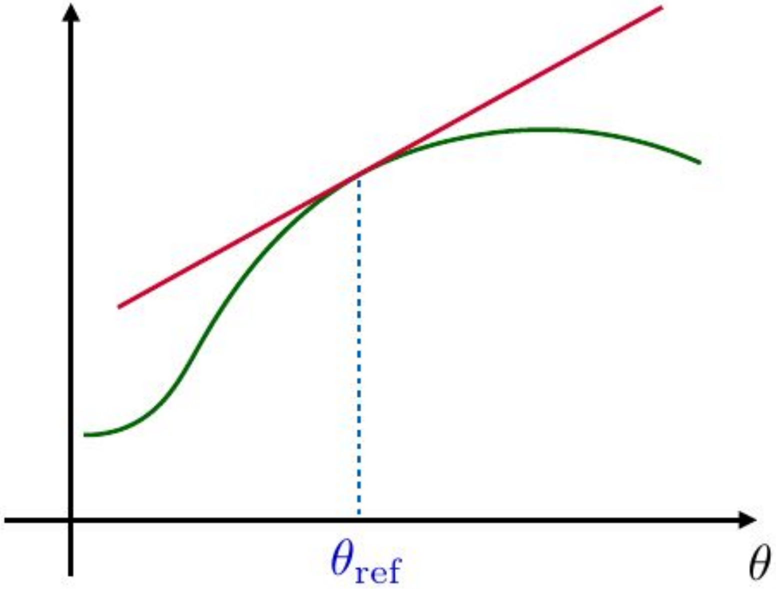
Local model and optimal observables

Taylor expansion of the score log-likelihood around a reference parameter point (θ_{SM})

$$\log p(x|\theta) = \log p(x|\theta_{\rm ref}) + \underbrace{\nabla_\theta \log p(x|\theta)\Big|_{\theta_{\rm ref}}}_{\equiv t(x|\theta_{\rm ref})} \cdot (\theta - \theta_{\rm ref}) + \mathcal{O}\left((\theta - \theta_{\rm ref})^2\right)$$

In the neighborhood of θ_{REF}, near the SM:

- The score vector components are sufficient statistics
- knowing the full likelihood ratio is as powerful as knowing the score
- The score is the **most powerful** observable



But the true score is intractable

Local score regression and density estimation (SALLY)

A precisely estimated score vector is the ideal summary statistics, in the neighborhood of the SM.

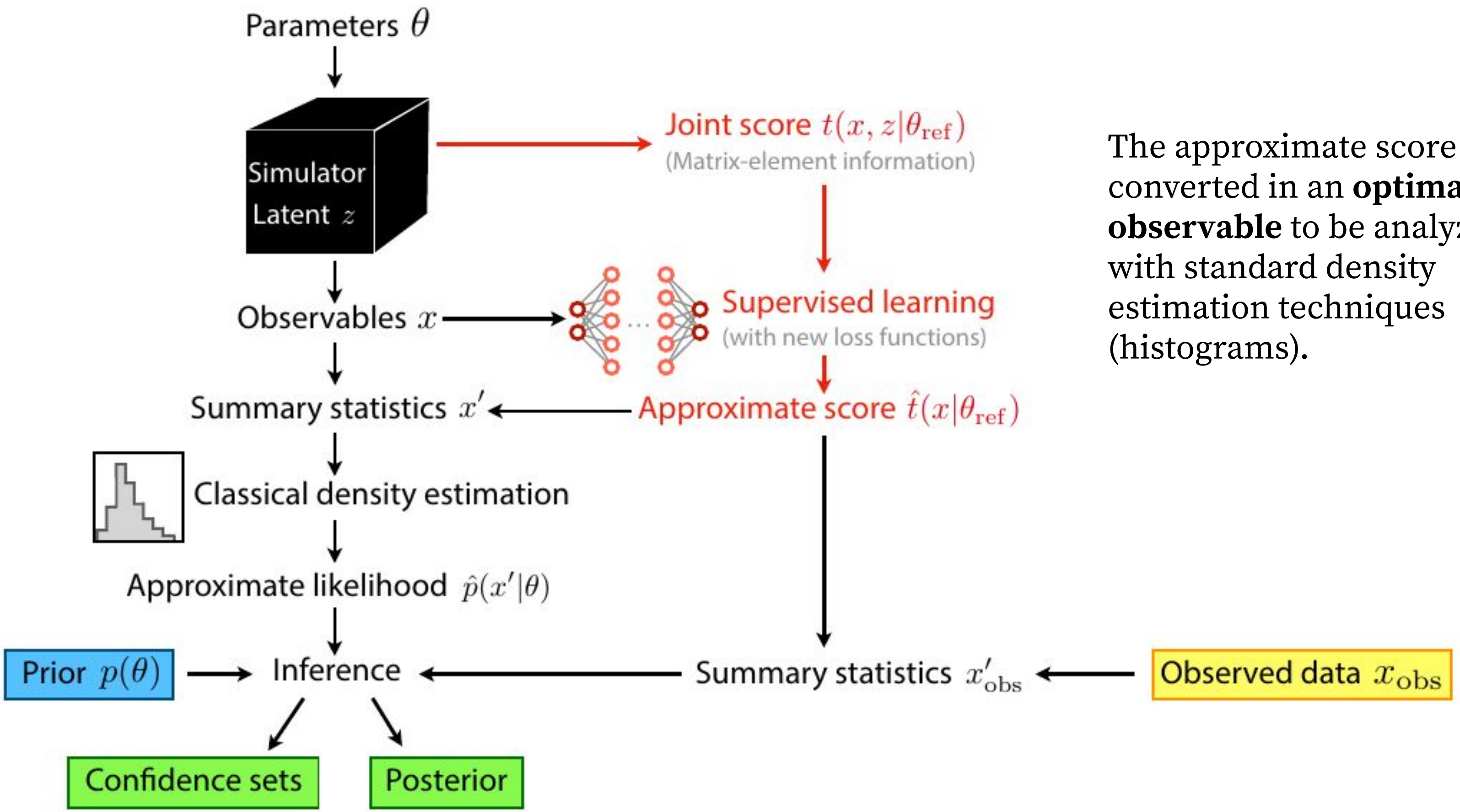
- The score is estimated using the joint score with events sampled from $(x_e, z_e) \sim p(x, z|\theta_{\text{score}})$
- In a second step the likelihood density is estimated with histograms or other density estimation techniques, obtaining the estimator for likelihood ratio.

$$\hat{r}(x|\theta_0, \theta_1) = \frac{\hat{p}\left(\hat{t}(x|\theta_{\text{score}}) \,\middle|\, \theta_0\right)}{\hat{p}\left(\hat{t}(x|\theta_{\text{score}}) \,\middle|\, \theta_1\right)}$$

SALLINO: Measurements of density in high-dimensional parameter spaces can be computationally expensive. We can use the local model and the score to build a single scale encapsulating all information on the likelihood ratio in the local approximation.

$$\hat{h}(x|\theta_0, \theta_1) \equiv \hat{t}(x|\theta_{SM}) \cdot (\theta_0 - \theta_1)$$

Neural optimal observables (SALLY)



The approximate score can be converted in an **optimal observable** to be analyzed with standard density estimation techniques (histograms).

Samples generation

- To train the described estimator a large number of simulated events in the θ parameter space is needed.
- Realistic SMEFT measurements should include contributibution from many operators: an efficient way to generate samples in different points of the parameters space is needed

- The structure of EFT amplitudes can be exploited to write the parton level likelihood as

$$|g_1 M_{SM} + g_2 M_{BSM}|^2 = g_1^2 |M_{SM}|^2 + 2 g_1 g_2 Re\left[M_{SM}^* M_{BSM}\right] + g_2^2 |M_{BSM}|^2 \longrightarrow \quad p(z|\theta) = \sum_{c'} \tilde{w}_{c'}(\theta)\, f_{c'}(z)$$

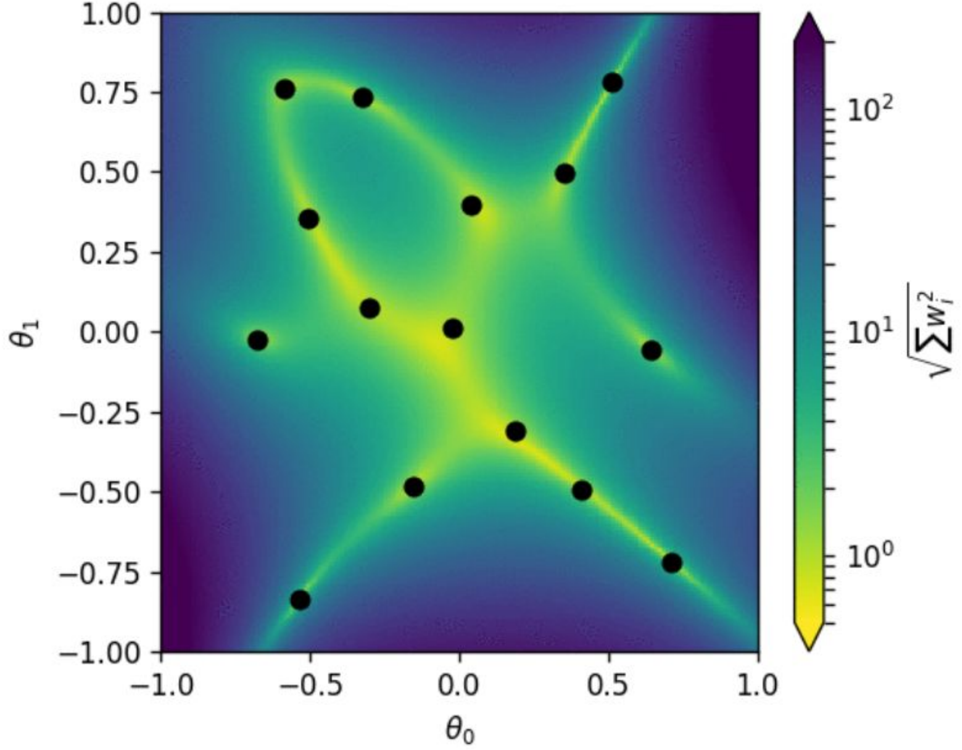
- Taking c' basis parameter point θ_c a mixture model can be written as

$$\boxed{p(z|\theta) = \sum_{c} w_c(\theta)\, p(z|\theta_c)}$$

This **morphing** procedure allow us to extract the full likelihood at parton level **from a finite set of evaluations of basis densities.** This implies that also the intractable full likelihood can be decomposed and this method can be used to train the models described:

$$p(x|\theta) = \sum_{c} w_c(\theta)\, p_c(x)$$

Morphing weights



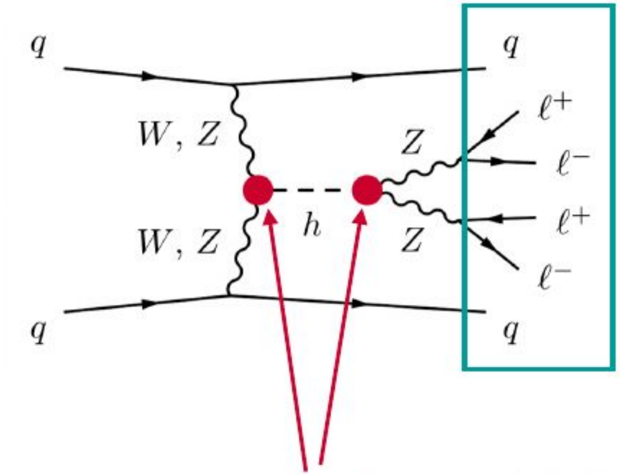
- The morphing weights can become large in portions of the phase space

→ challenge on the numerical stability of the method.

- For 2 BSM operators affecting VBF Higgs production and decay, a 15-D vector space is needed.

→ 15 different θ_C generations

at least 16-dimensional observable space



Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \boxed{\frac{f_W}{\Lambda^2}} \underbrace{\frac{ig}{2}\,(D^\mu\phi)^\dagger\,\sigma^a\,D^\nu\phi\,W^a_{\mu\nu}}_{\mathcal{O}_W} - \boxed{\frac{f_{WW}}{\Lambda^2}} \underbrace{\frac{g^2}{4}\,(\phi^\dagger\phi)\,W^a_{\mu\nu}\,W^{\mu\nu\,a}}_{\mathcal{O}_{WW}}$$

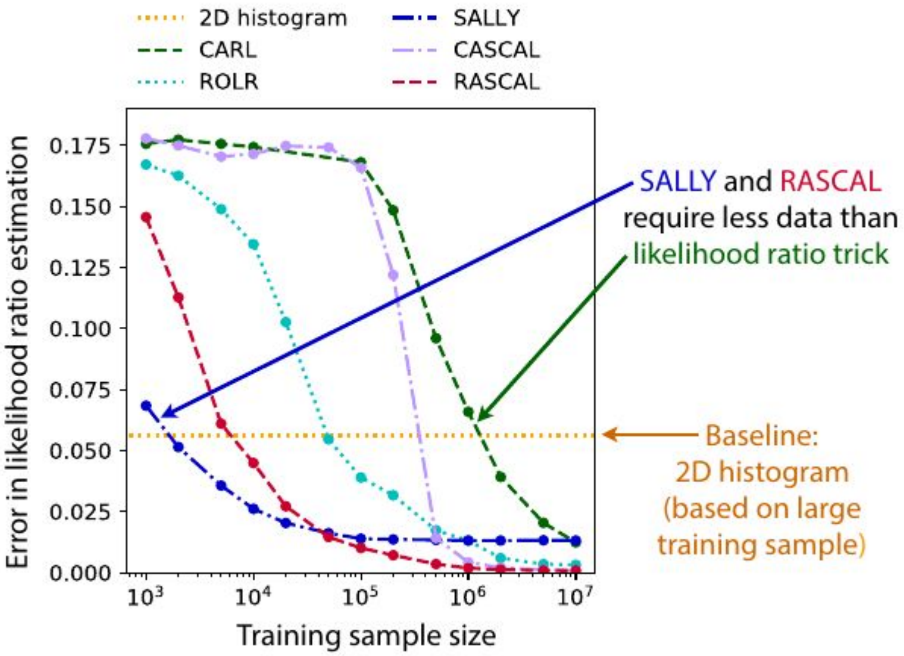
Goal: constrain the two EFT parameters

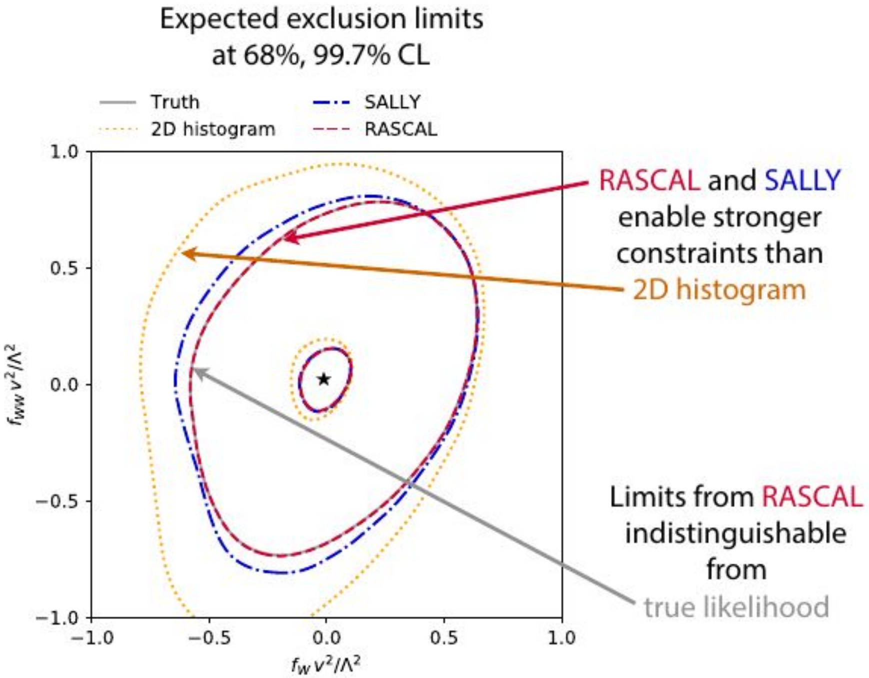
- new inference methods
- baseline: 2d histogram analysis of jet momenta & angular correlations

Two scenarios:

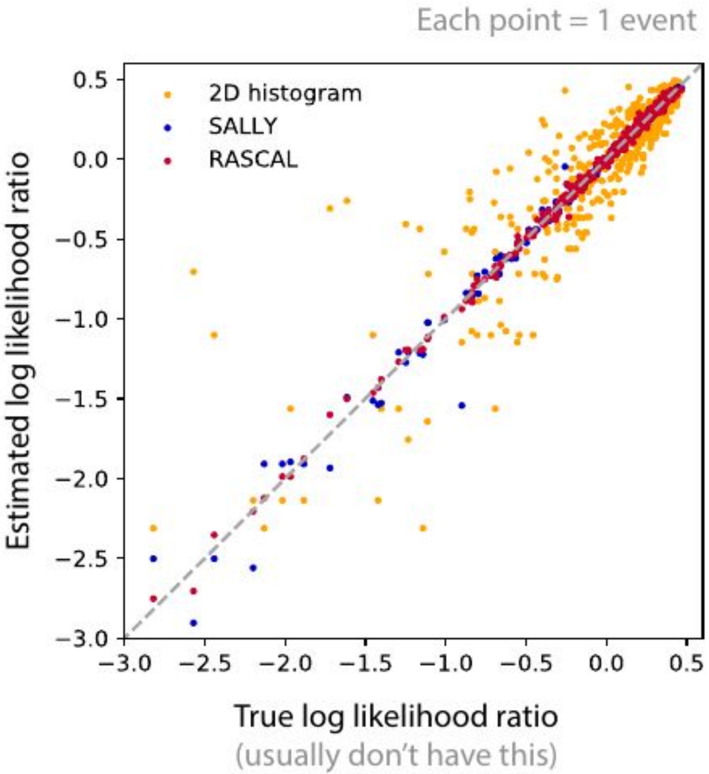
- Simplified setup in which we can compare to true likelihood
- “Realistic” simulation with approximate detector effects

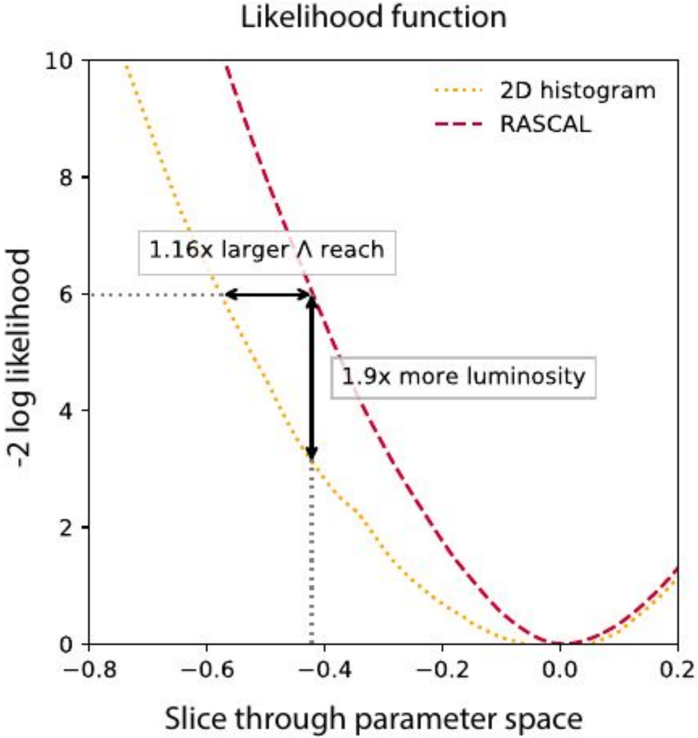
Stronger constraints with less training data





16% greater reach (~90% more data)





Conclusion

- Modern machine learning techniques combined with sophisticated simulation tools (as Madgraph and Madmax) can efficiently improve the power of inference on EFT parameters with the respect of traditional methods.

- Exploiting the structure of particle physics processes, these new analysis techniques extract additional information from the event generators, and use this information to train precise estimators for likelihood ratios

- This scales well to high-dimensional parameter spaces such as that of effective field theories. The new methods do not require any approximations on the hard process, parton shower, or detector effects, and the likelihood ratio for any event and hypothesis pair can be evaluated fast on modern hardware

- The resulting models for the likelihood ratio demonstrate an improved power in limit extractions.

References

Papers:

- [arxiv:1805.00013] Constraining Effective Field Theories with Machine Learning
- [arxiv:1805.00020] A Guide to Constraining Effective Field Theories with Machine Learning
- [arxiv:1805.12244] Mining gold from implicit models to improve likelihood-free inference

Graphs and schema from Kyle Cranmer lessons at the PREFIT 2020 school:

- https://indico.cern.ch/event/817757/contributions/3712508/attachments/1998432/3334682/PREFIT-lecture1.pdf
- https://indico.cern.ch/event/817757/contributions/3712517/attachments/1998425/3334673/PREFIT20-lecture2.pdf