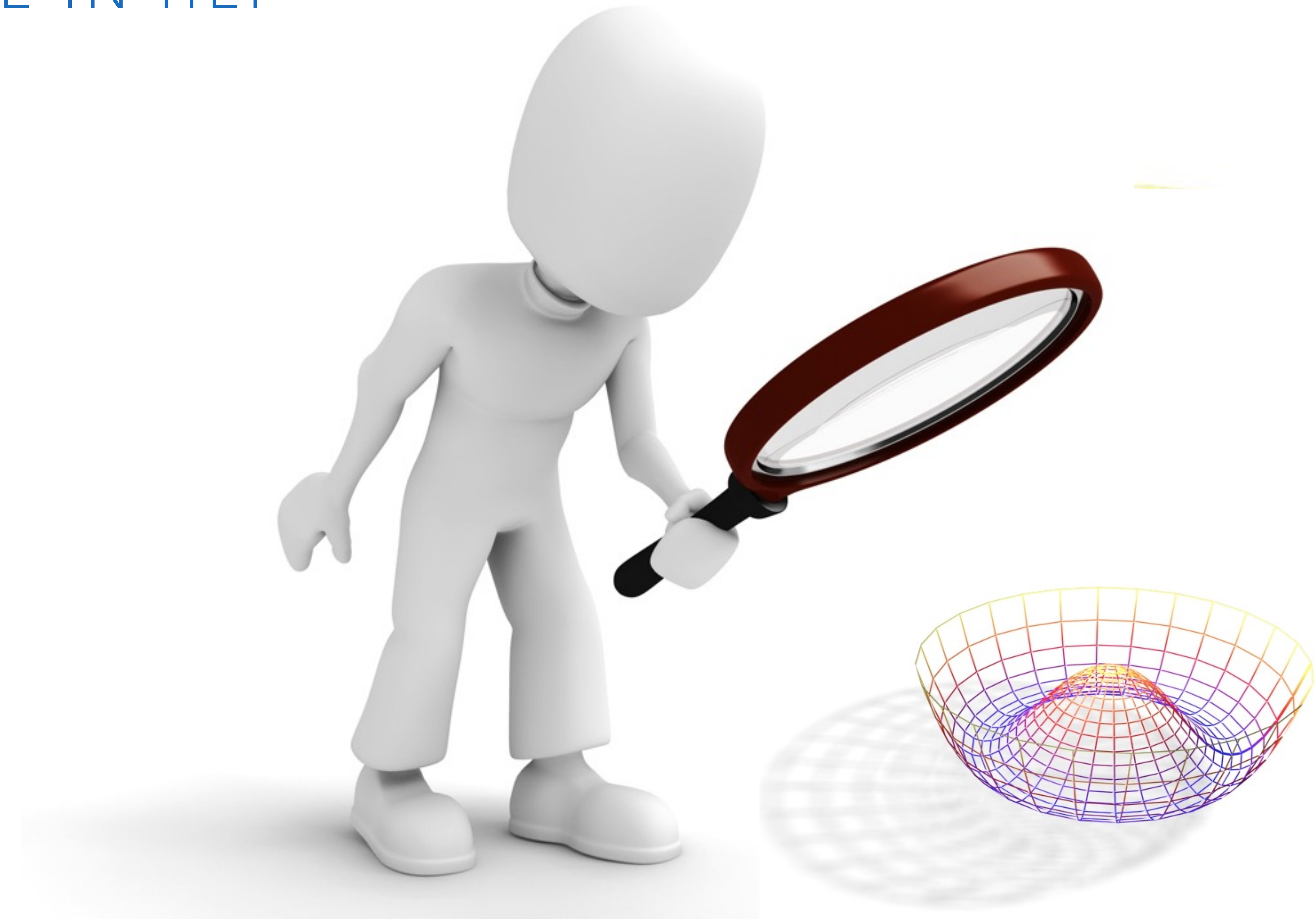
NYU Center for
Data Science

Center for
Cosmology and
Particle Physics

MADMINER: A PYTHON BASED TOOL FOR SIMULATION-BASED INFERENCE IN HEP



@KyleCranmer
New York University
Department of Physics
Center for Data Science
CILVR Lab

Acknowledgements


Johann Brehmer


Gilles Louppe


Juan Pavez


Markus Stoye

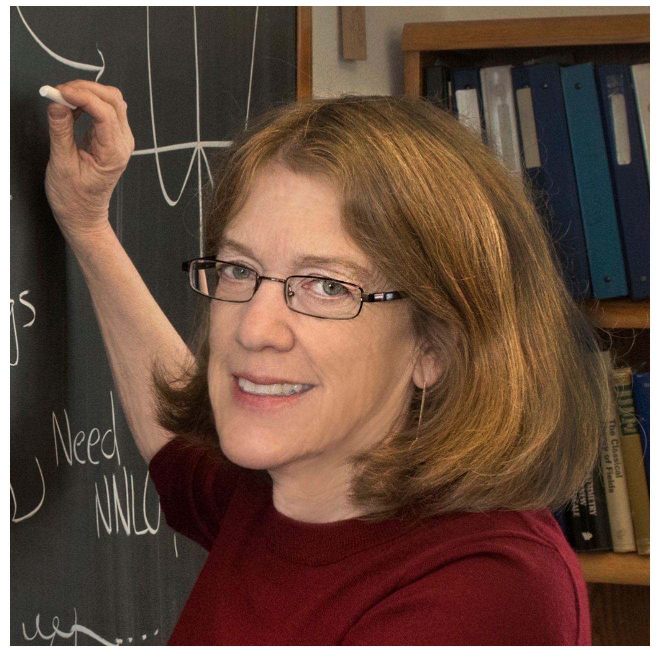

Felix Kling


Irina Espejo


Sinclert Perez

Special thanks to Johann for slides I borrowed


Tilman Plehn


Sally Dawson


Sam Homiller


Matthew Feickert

NYU

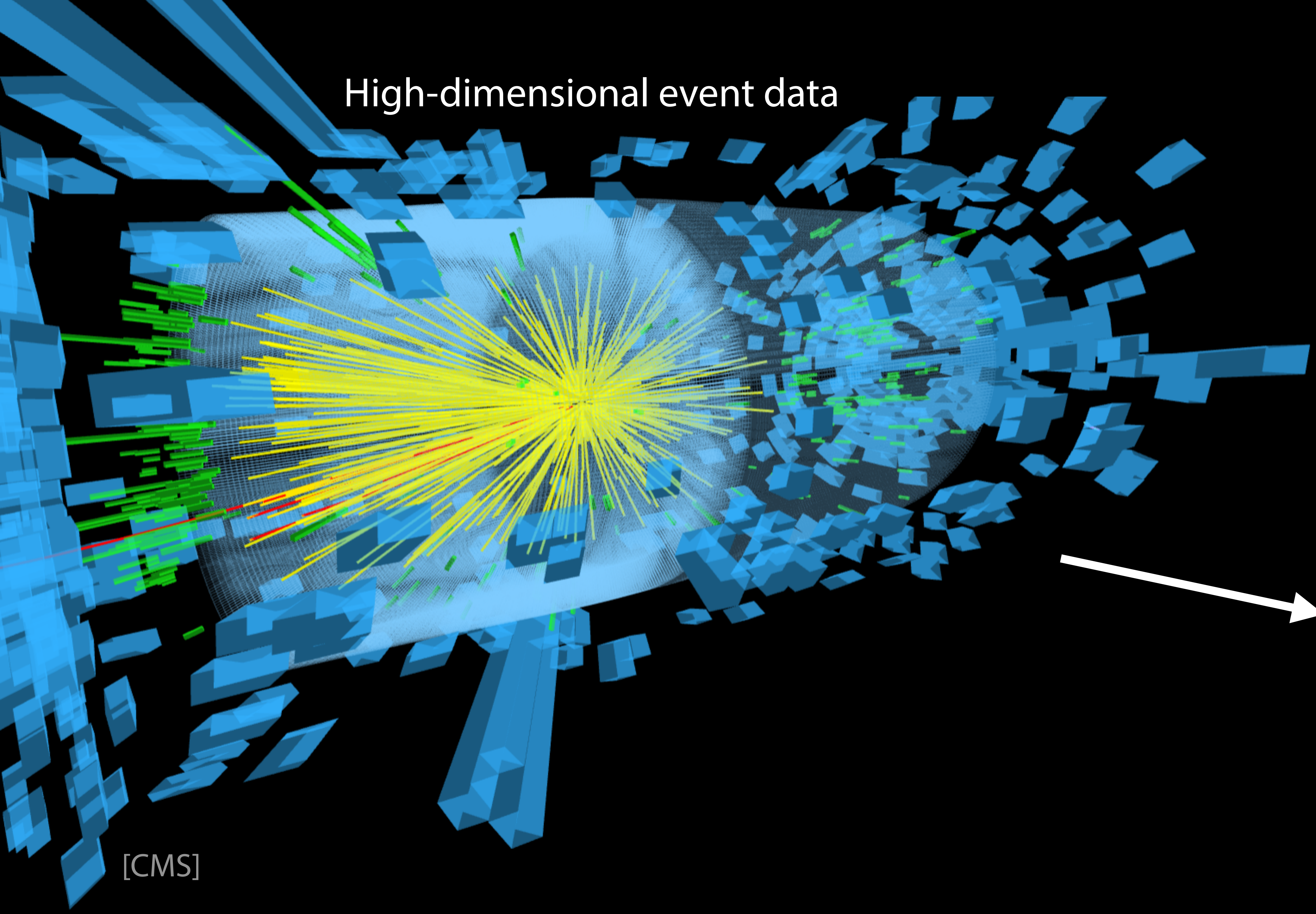
MSDSE
Moore - Sloan
Data Science Environments

dianahep

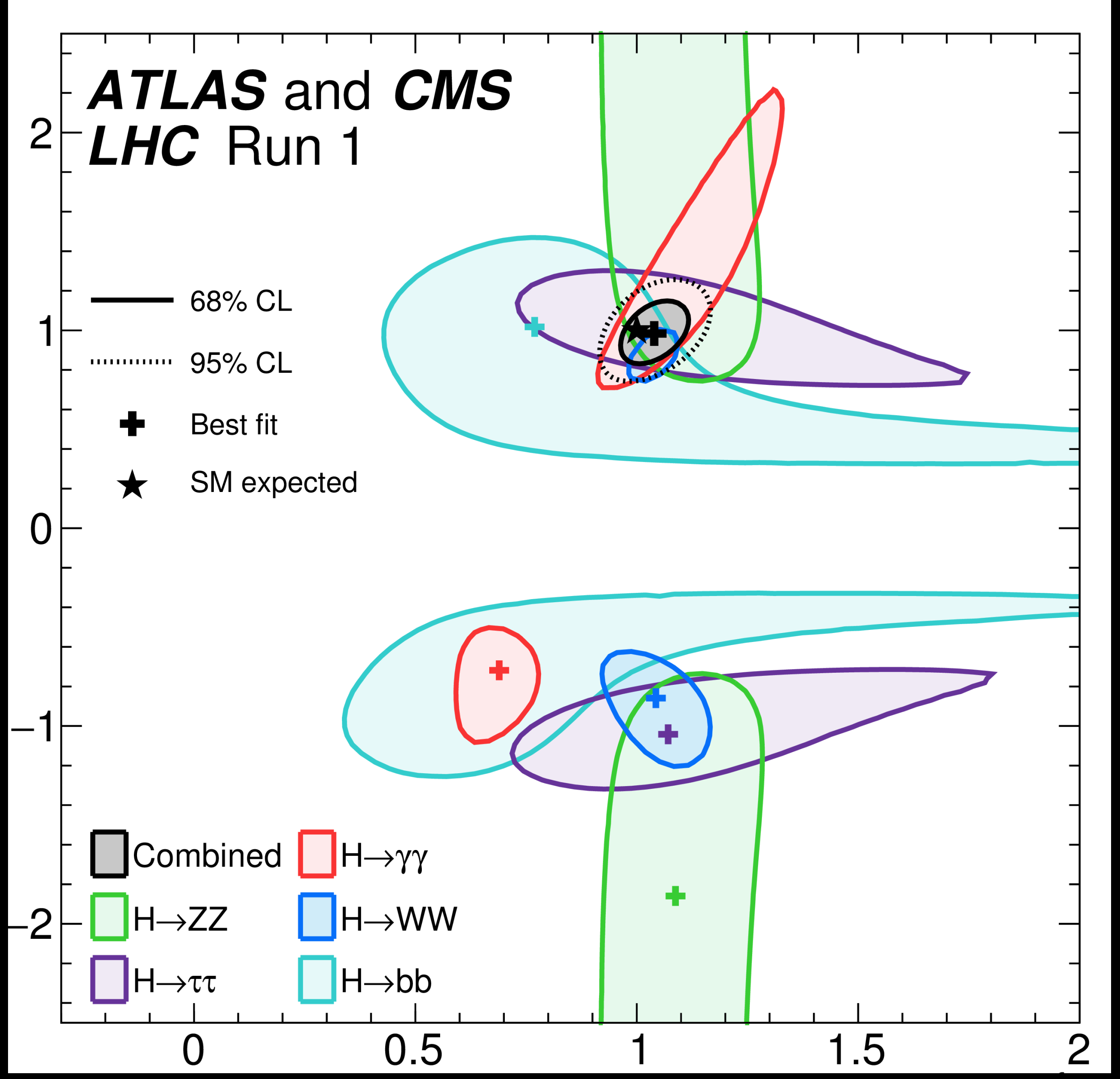
iris hep
Institute for Research and Innovation in Software for High Energy Physics

The SCAILFIN Project
scailfin.github.io

High-dimensional event data



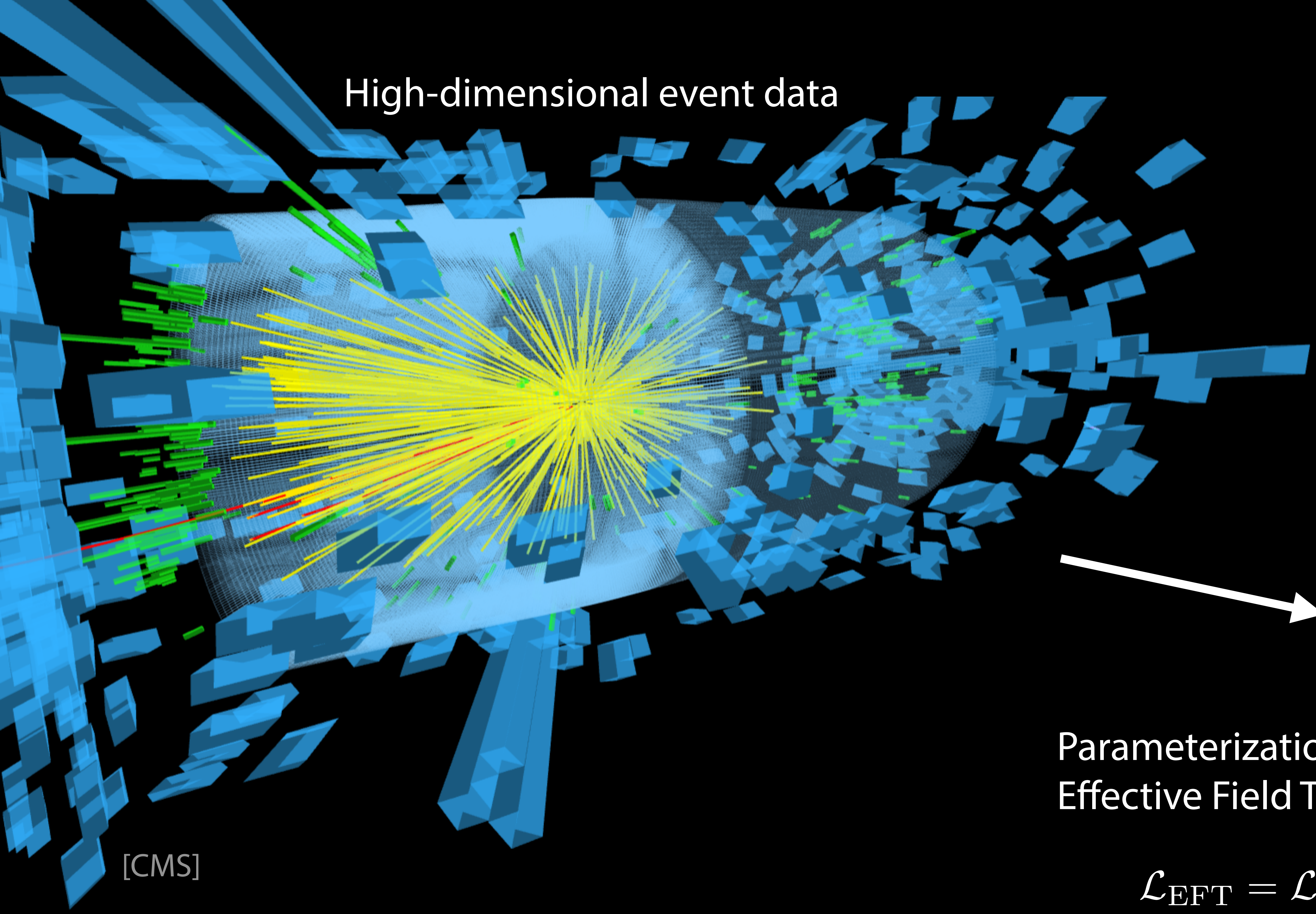
[CMS]



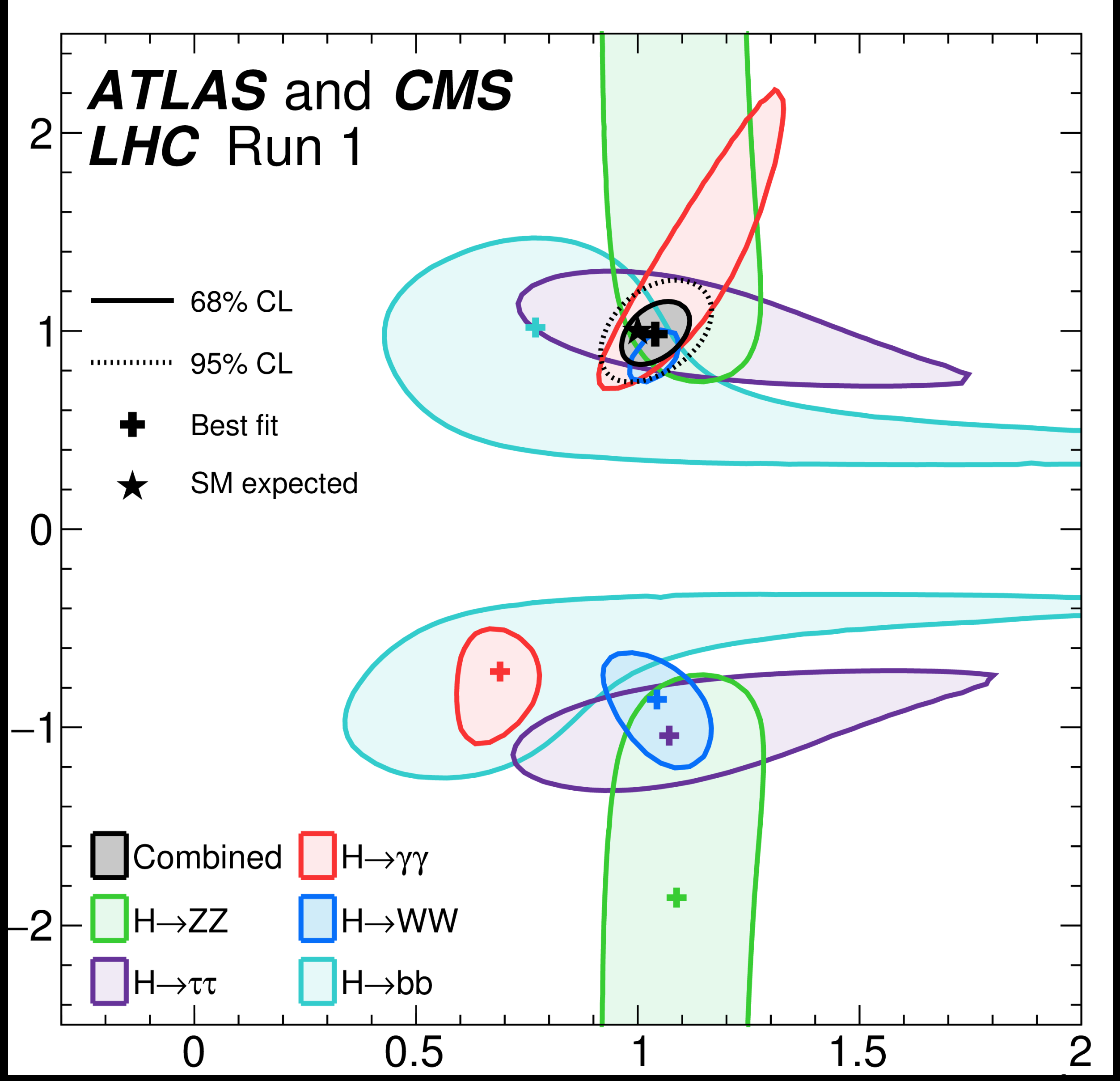
[ATLAS, CMS 1606.02266]

Precision constraints on new physics

High-dimensional event data



[CMS]



[ATLAS, CMS 1606.02266]

Precision constraints on new physics

Parameterization e.g. in Effective Field Theory:

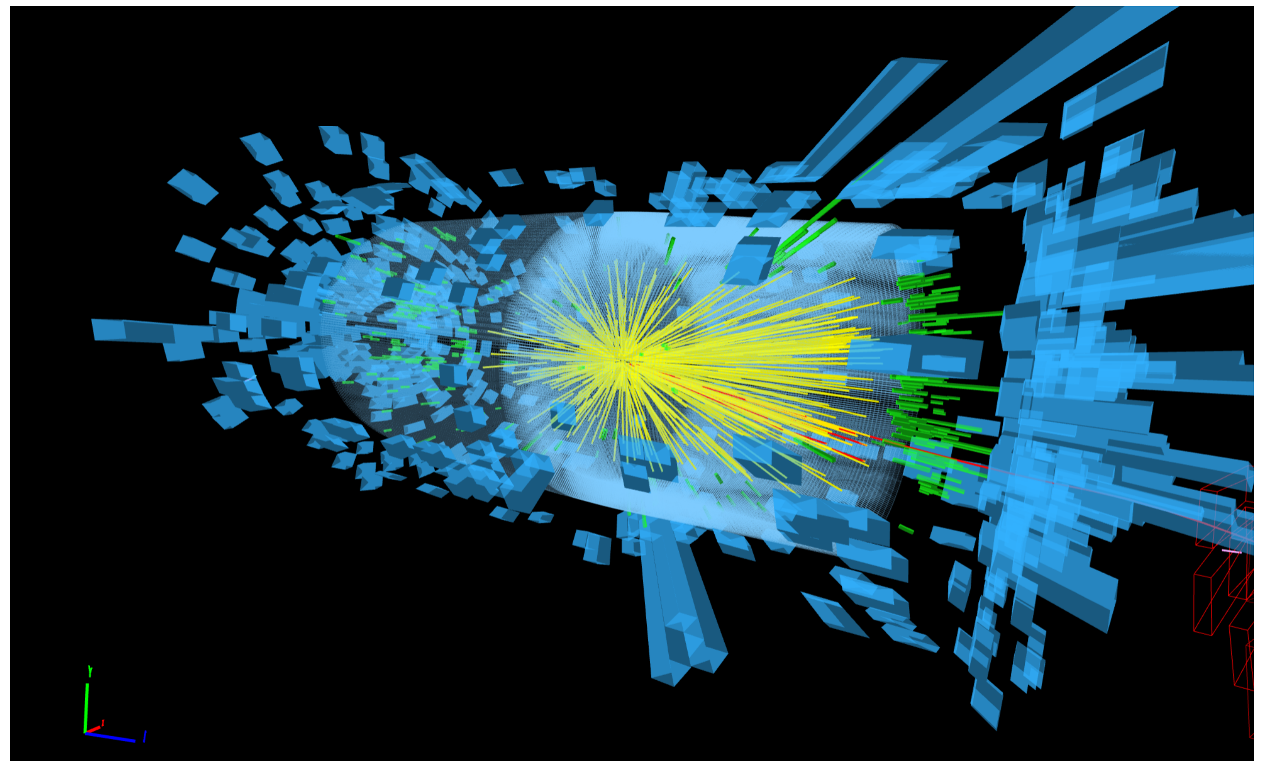
systematic expansion of new physics around Standard Model

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

10s to 100s "universal" parameters to measure

The likelihood is a key object

Let θ denote the coefficients of higher dimensional operators in the Lagrangian, x be high-dimensional data associated to an event, and $p(x \mid \theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma}{dx}$ be the distribution for the data

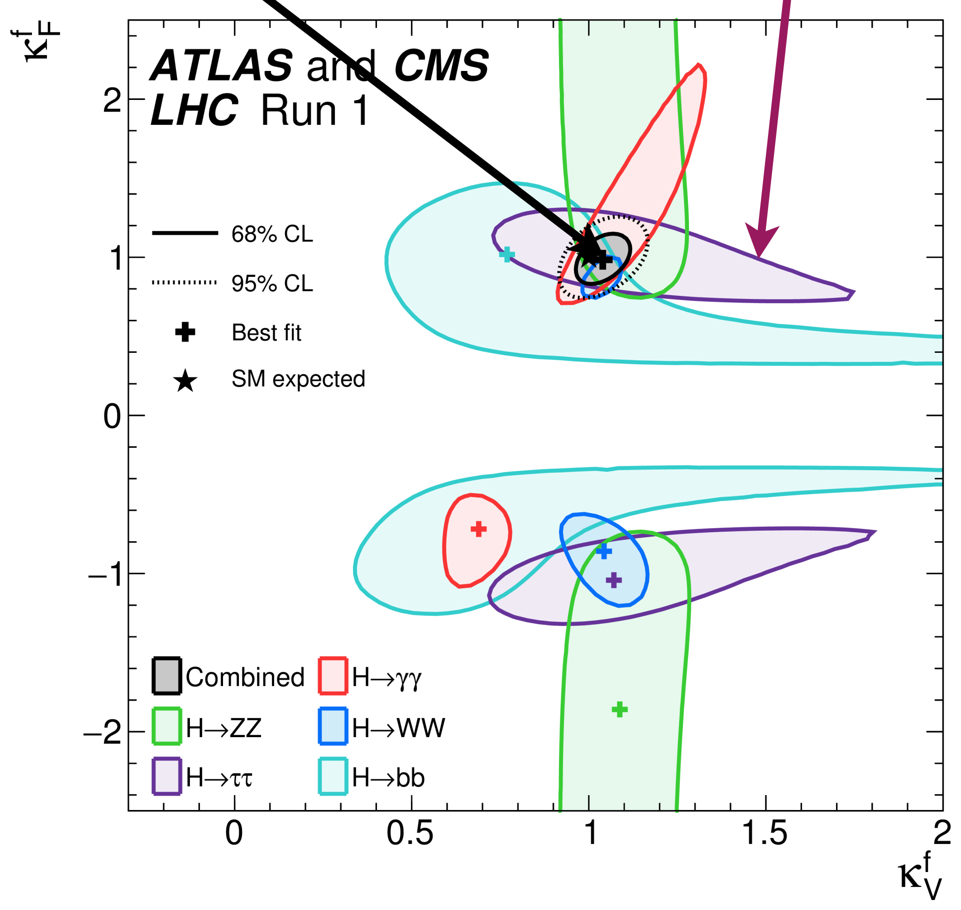


High-dimensional event data x

→ Likelihood function $p(x|\theta)$ →

Maximum-likelihood estimator

Confidence limits based on likelihood ratio tests



Constraints on parameters θ

Now for some bad news….

Particle physics processes do not have a tractable likelihood function.

Modeling particle physics processes

Theory parameters

θ

Evolution

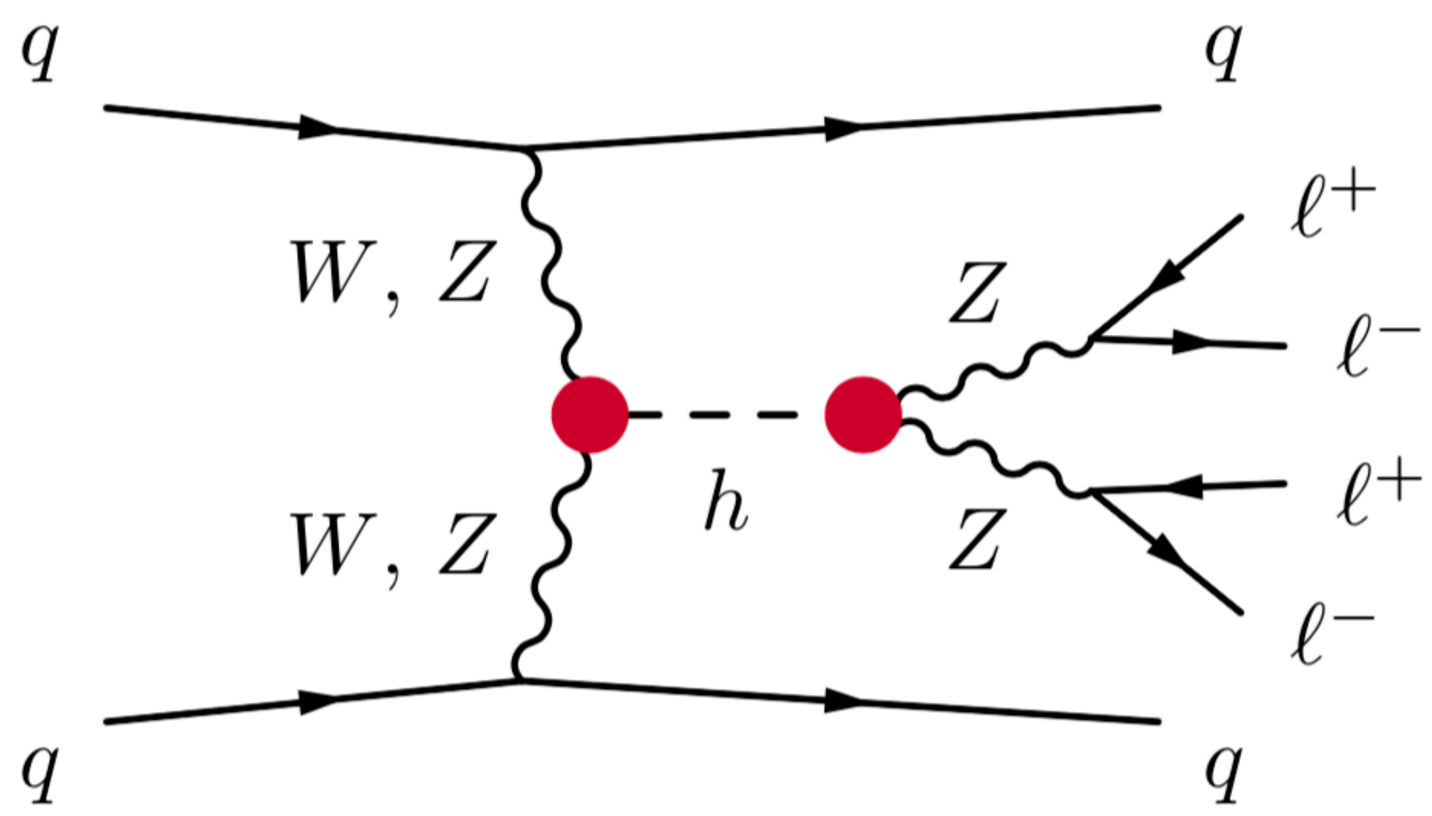
Modeling particle physics processes

Latent variables

Parton-level momenta

Theory parameters

$z_p \longleftarrow \theta$

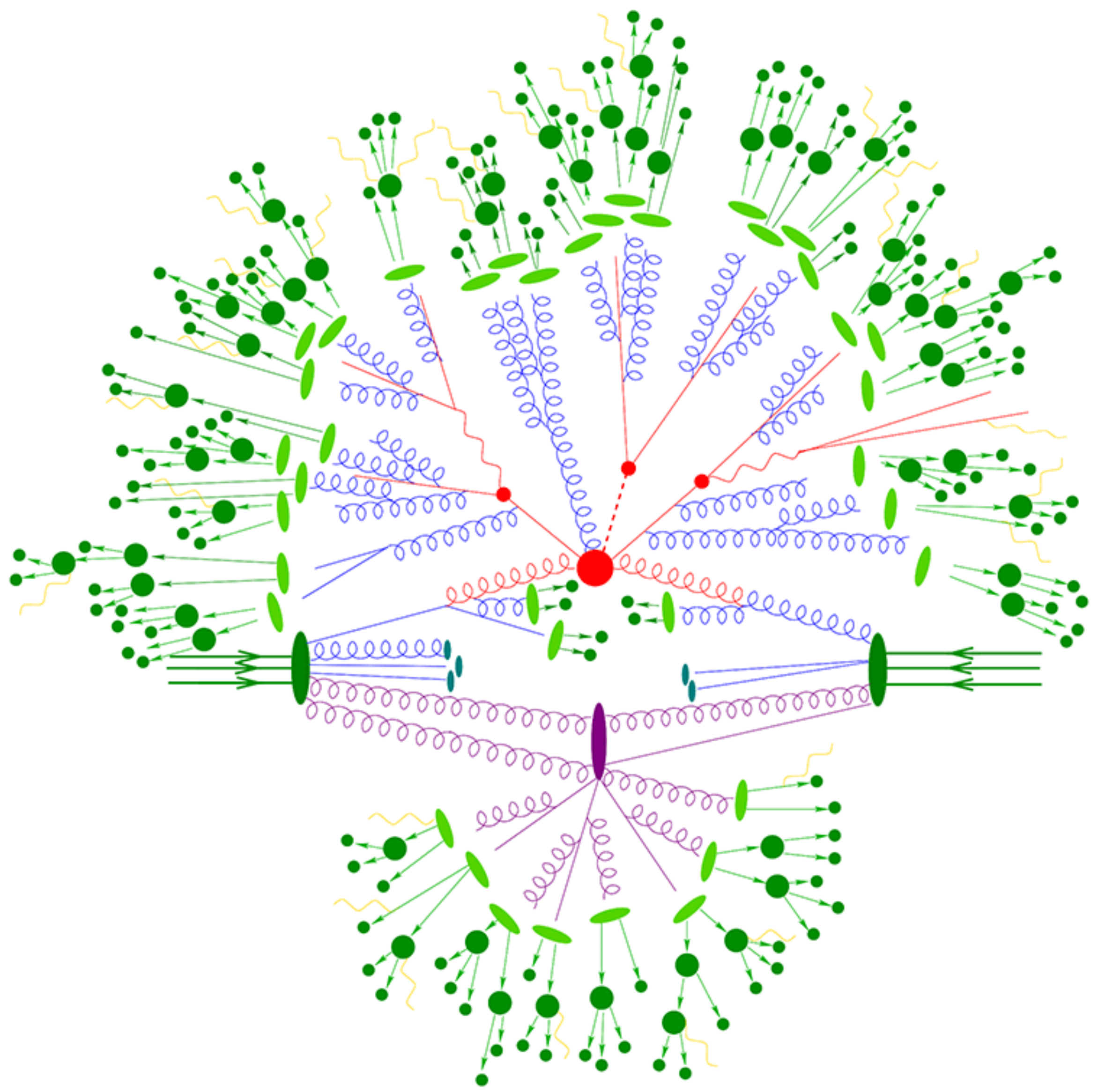


Evolution

Modeling particle physics processes

Latent variables

Shower splittings		Parton-level momenta		Theory parameters
z_s	$\longleftarrow$	z_p	$\longleftarrow$	θ



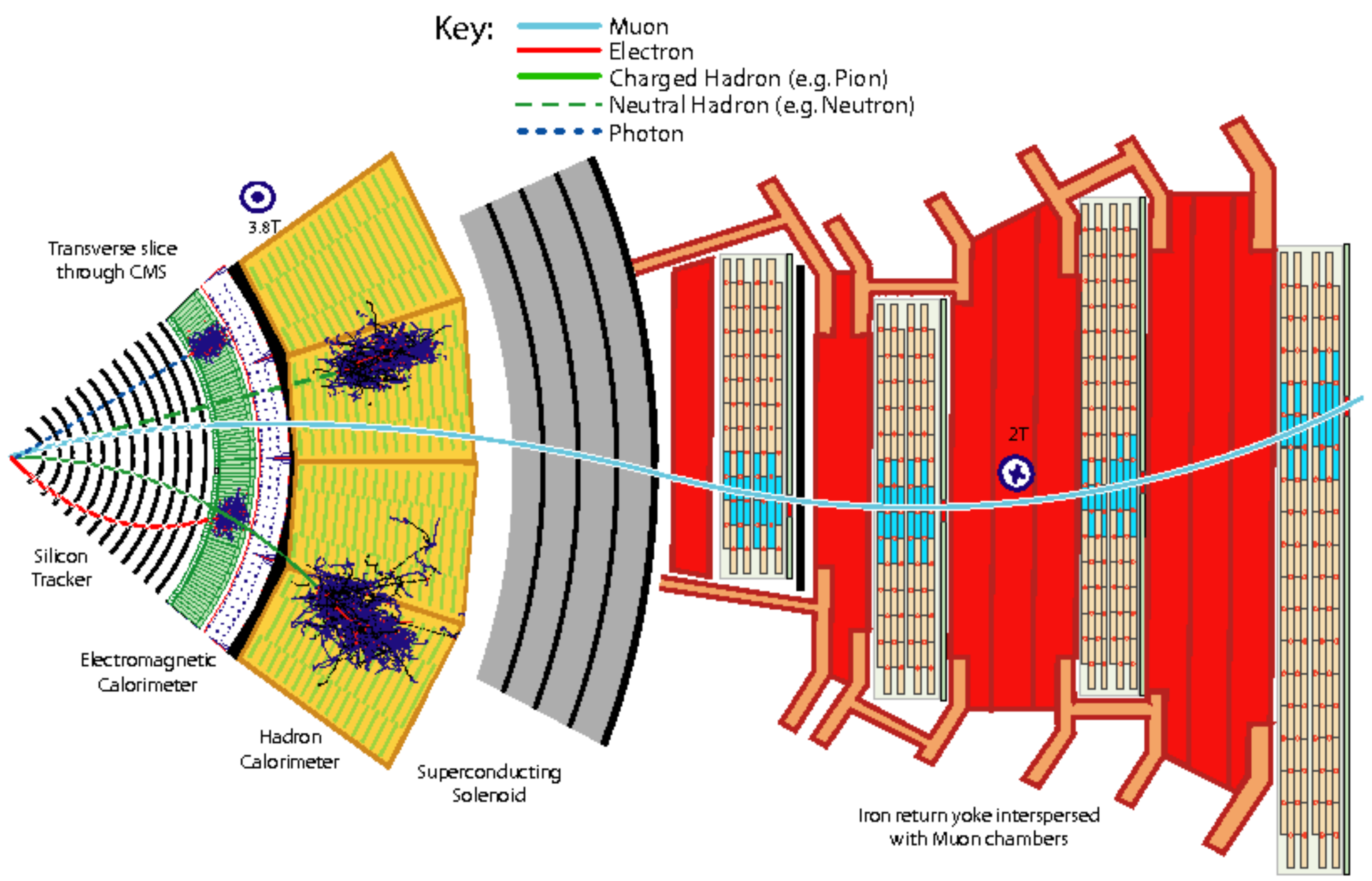
[F. Krauss]

$\longleftarrow$ Evolution

Modeling particle physics processes

Latent variables

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



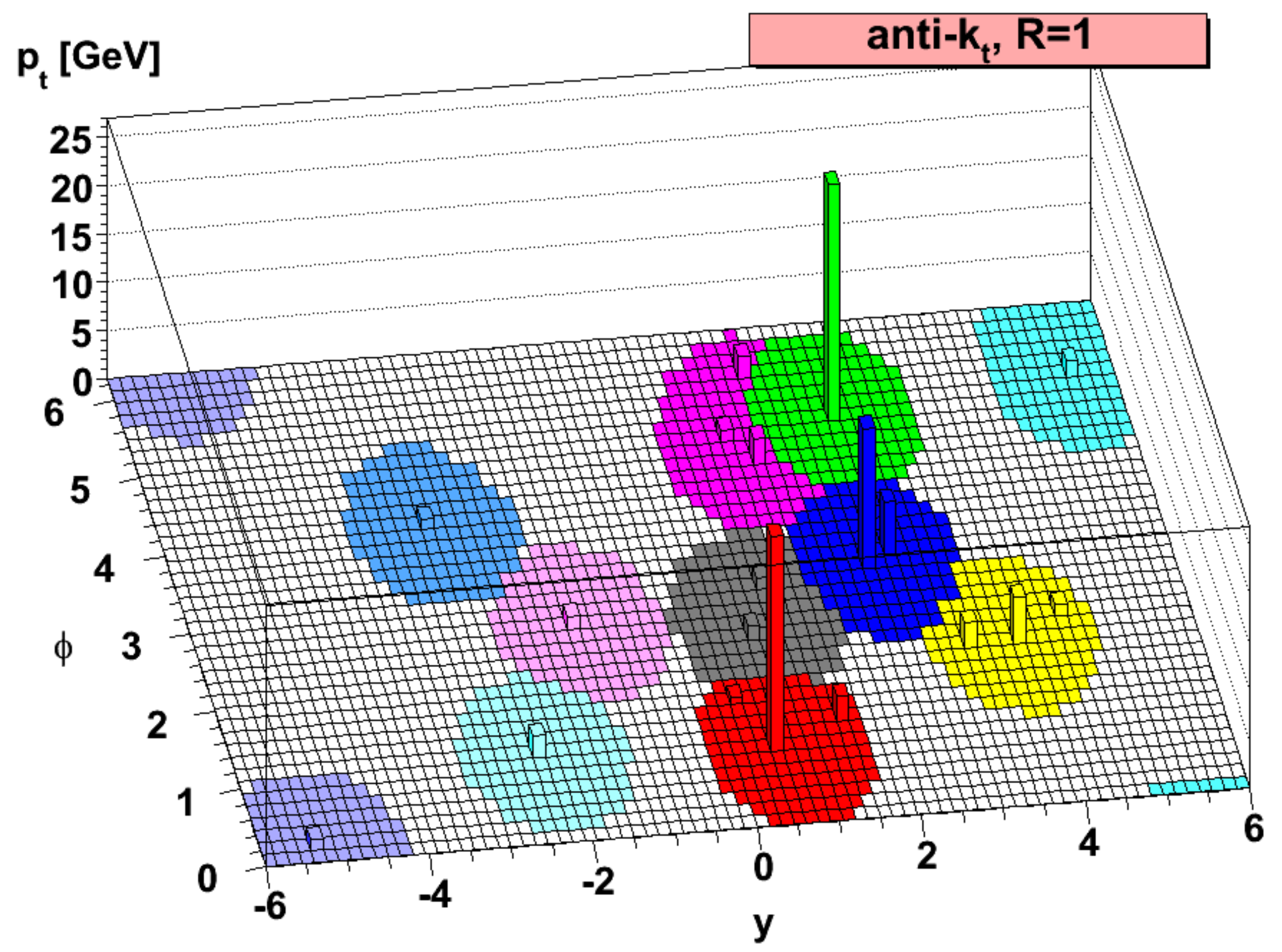
[CMS]

Evolution

Modeling particle physics processes

Latent variables

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



[M. Cacciari, G. Salam, G. Soyez 0802.1189]

← Evolution

Modeling particle physics processes

Latent variables

Observables		Detector interactions		Shower splittings		Parton-level momenta		Theory parameters
x	$\longleftarrow$	z_d	$\longleftarrow$	z_s	$\longleftarrow$	z_p	$\longleftarrow$	θ
Sample from	$p(x\|z_d)$		$p(z_d\|z_s)$		$p(z_s\|z_p)$		$p(z_p\|\theta)$	

GEANT4 A SIMULATION TOOLKIT

MADGRAPH5_aMC@NLO

Prediction (simulation)

Modeling particle physics processes

Latent variables

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

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

Inference →

Modeling particle physics processes

Latent variables

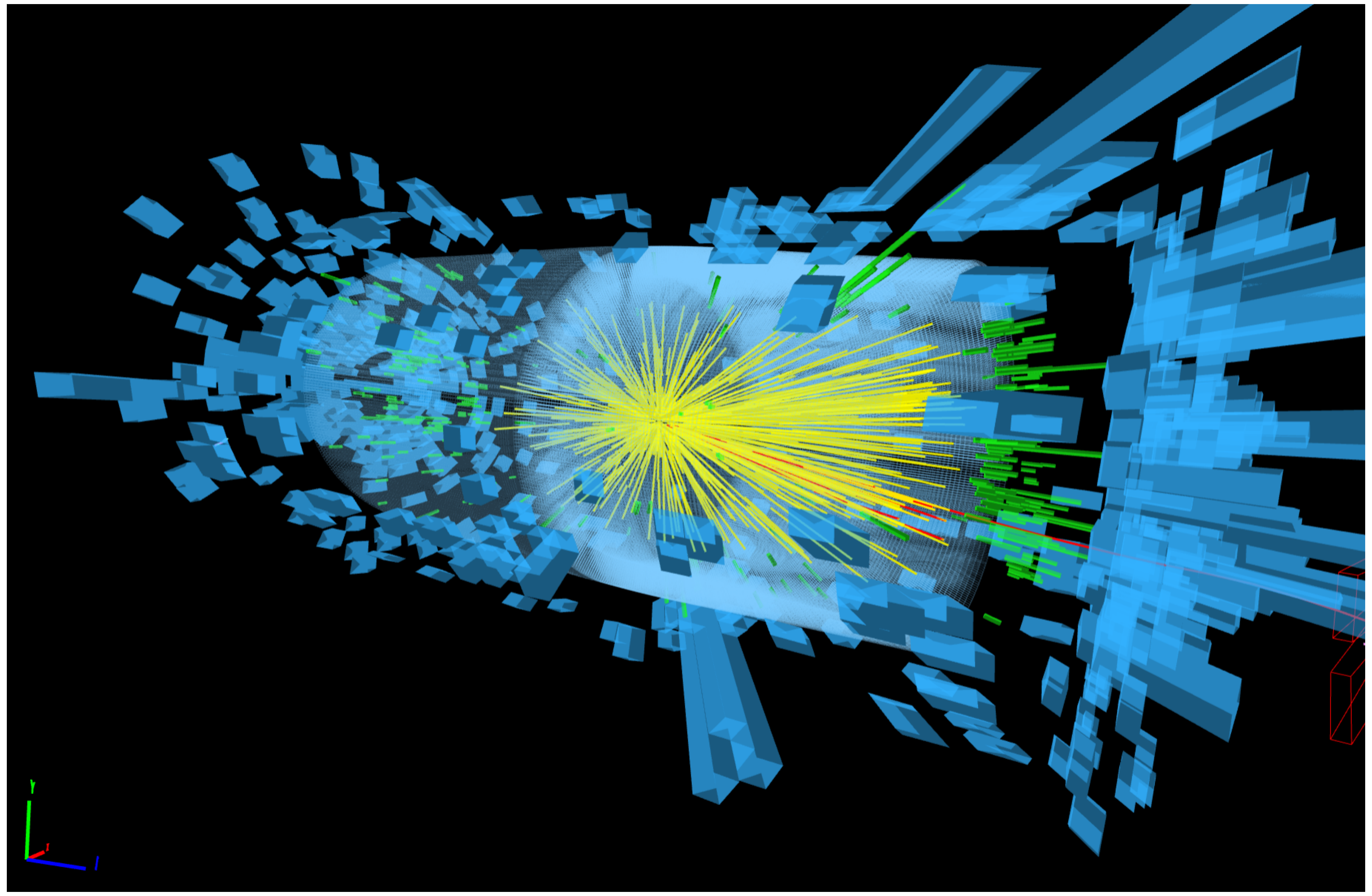
Observables		Detector interactions		Shower splittings		Parton-level momenta		Theory parameters
x	←	z_d	←	z_s	←	z_p	←	θ

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

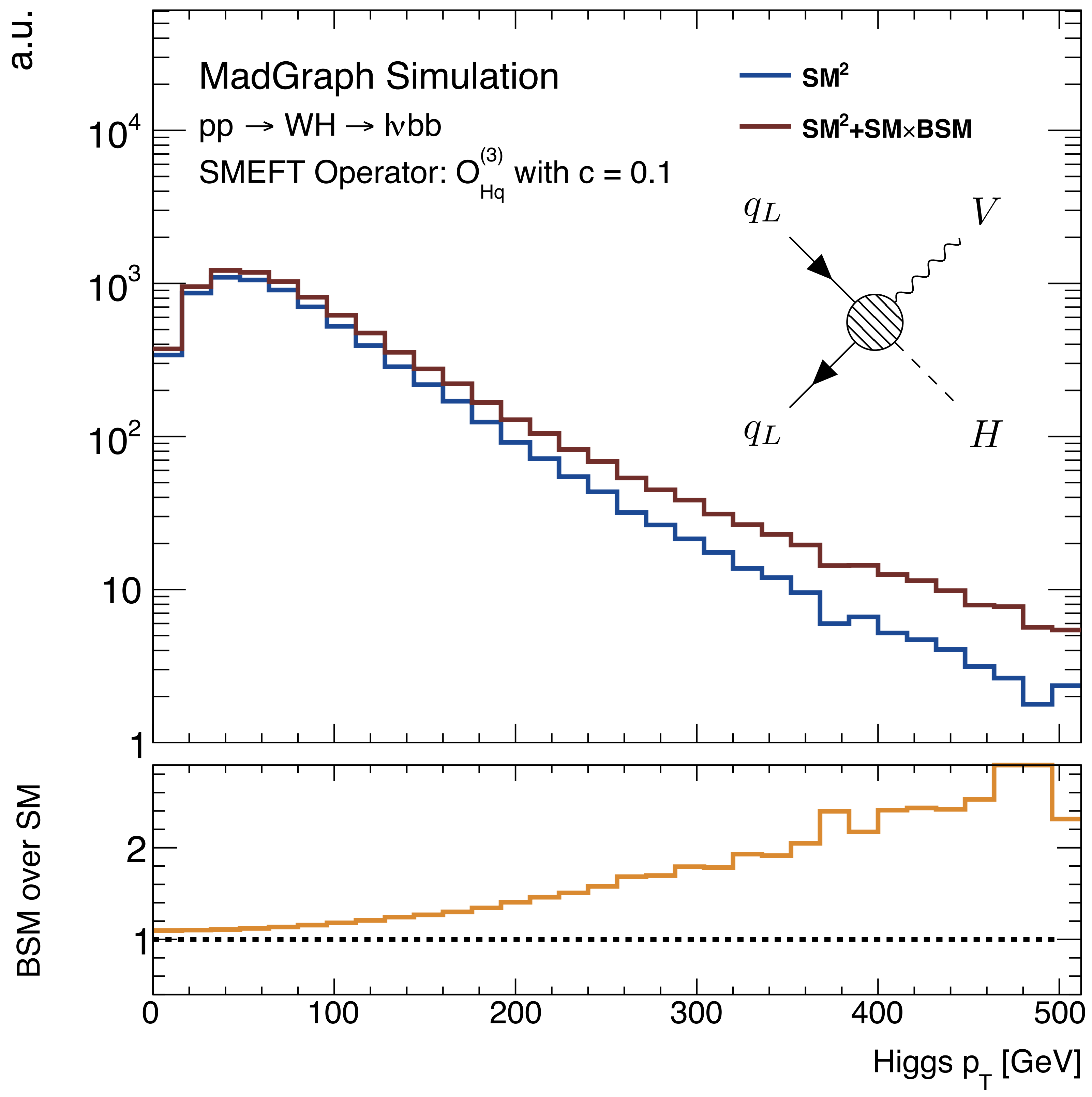
It's infeasible to calculate the integral over this enormous space!

Inference →

What we usually do: number counting or singly differential



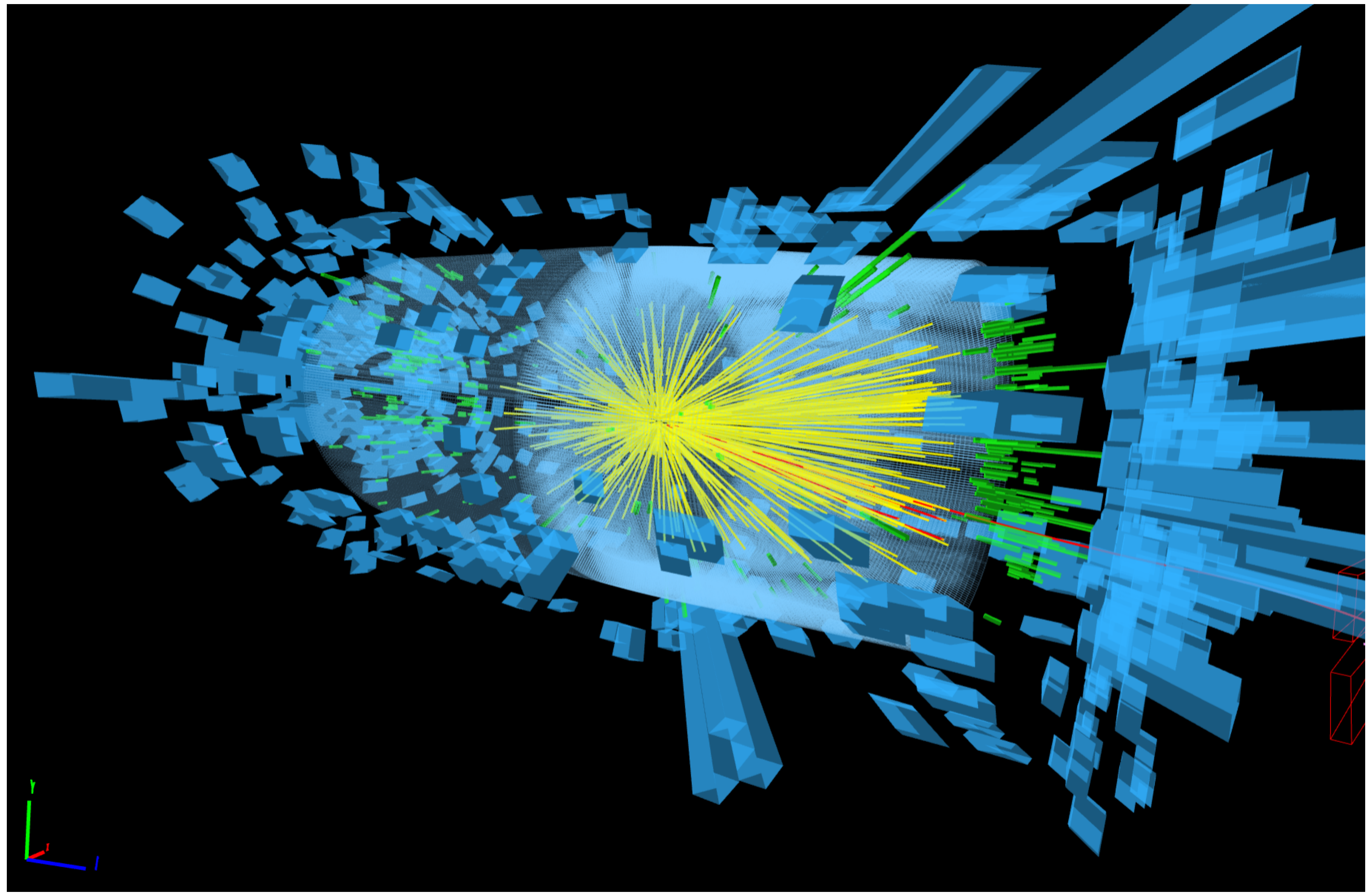
SMEFT: $O_{Hq}^{(3)} = 0.1$

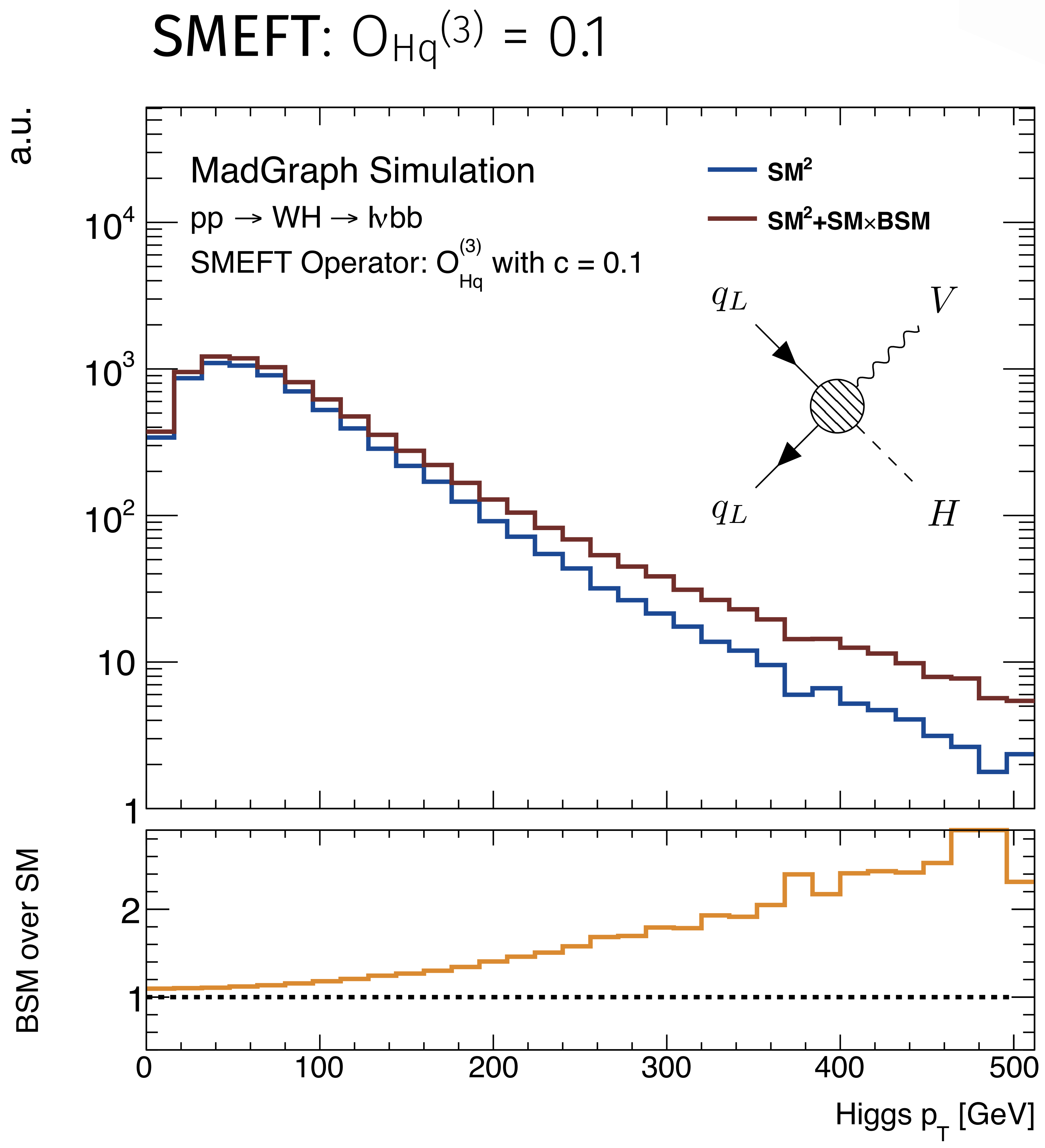


High-dimensional event data x

$p(x|\theta)$ cannot be calculated

What we usually do: number counting or singly differential





High-dimensional event data x

One or two summary statistics x'

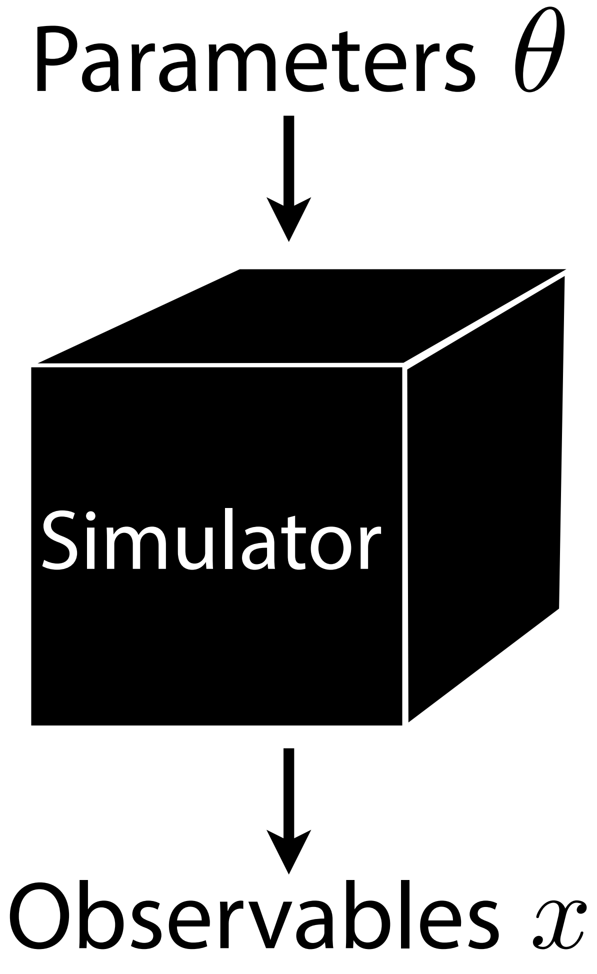
$p(x|\theta)$ cannot be calculated

$p(x'|\theta)$ can be estimated with histograms

n.b. "summary statistic" = a sensitive observable

Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]

Parameters θ

↓

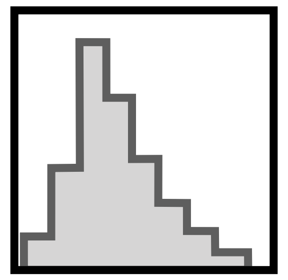
↓

Observables x

↓

Summary statistics x'

↓



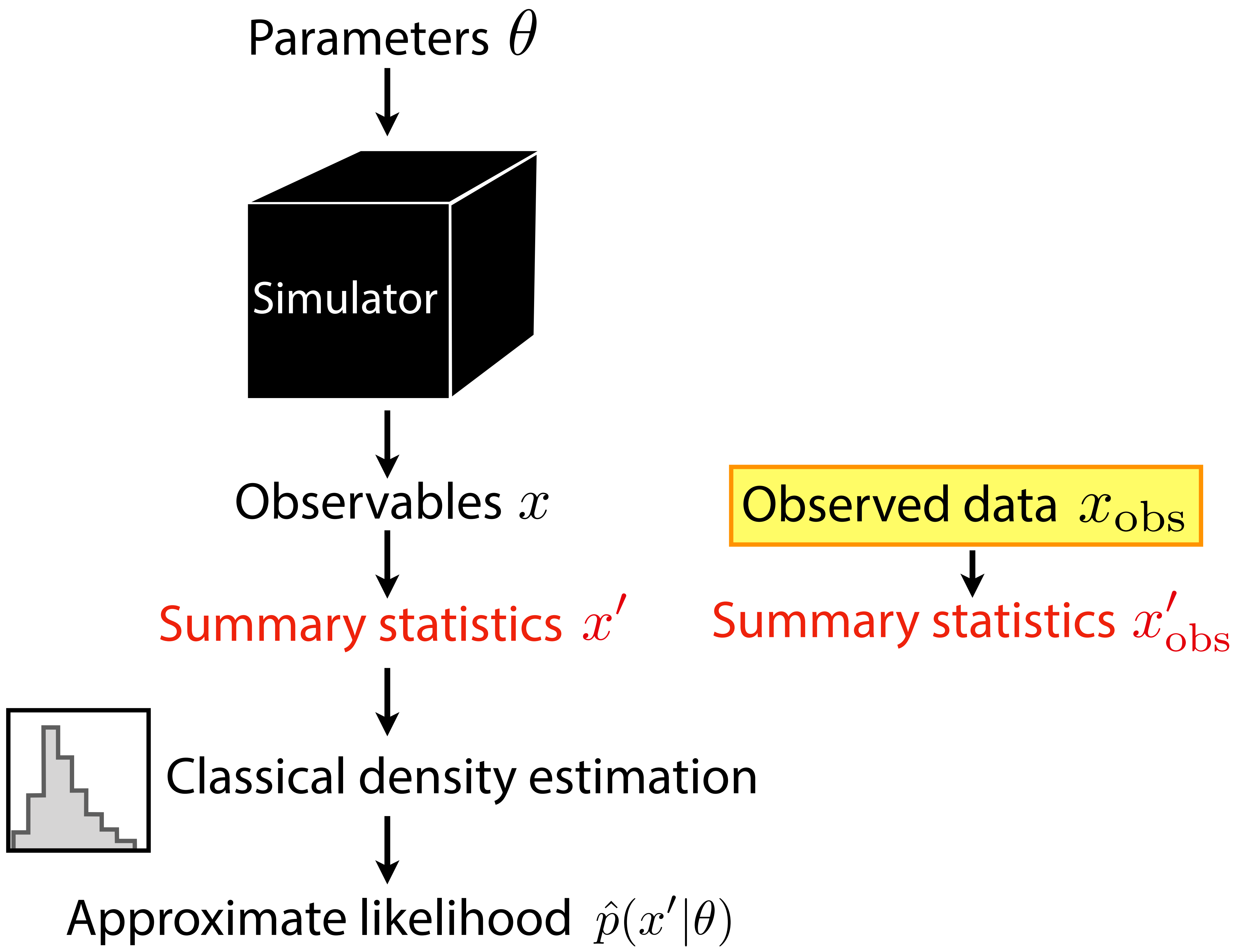
Classical density estimation

↓

Approximate likelihood $\hat{p}(x'|\theta)$

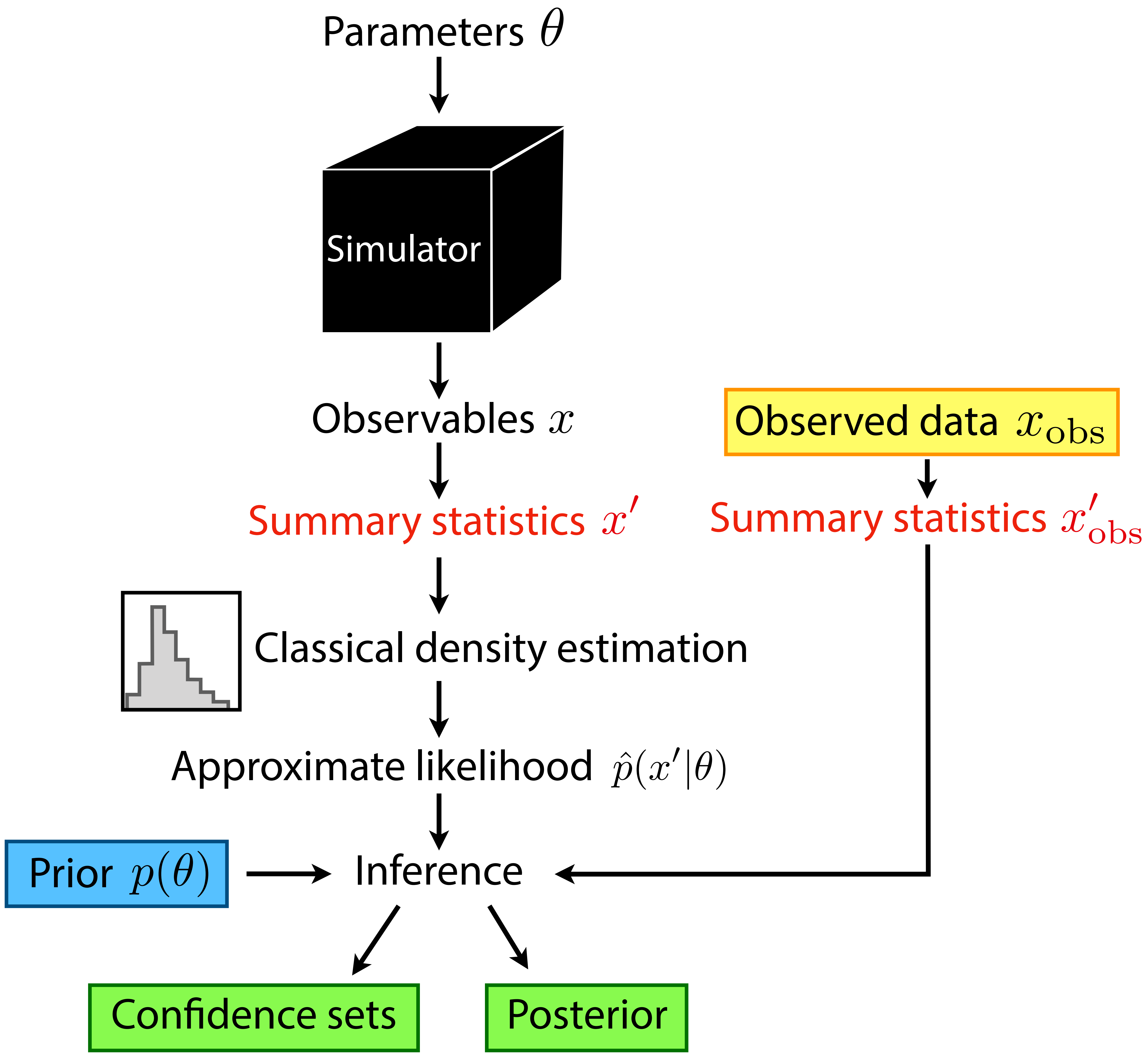
Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



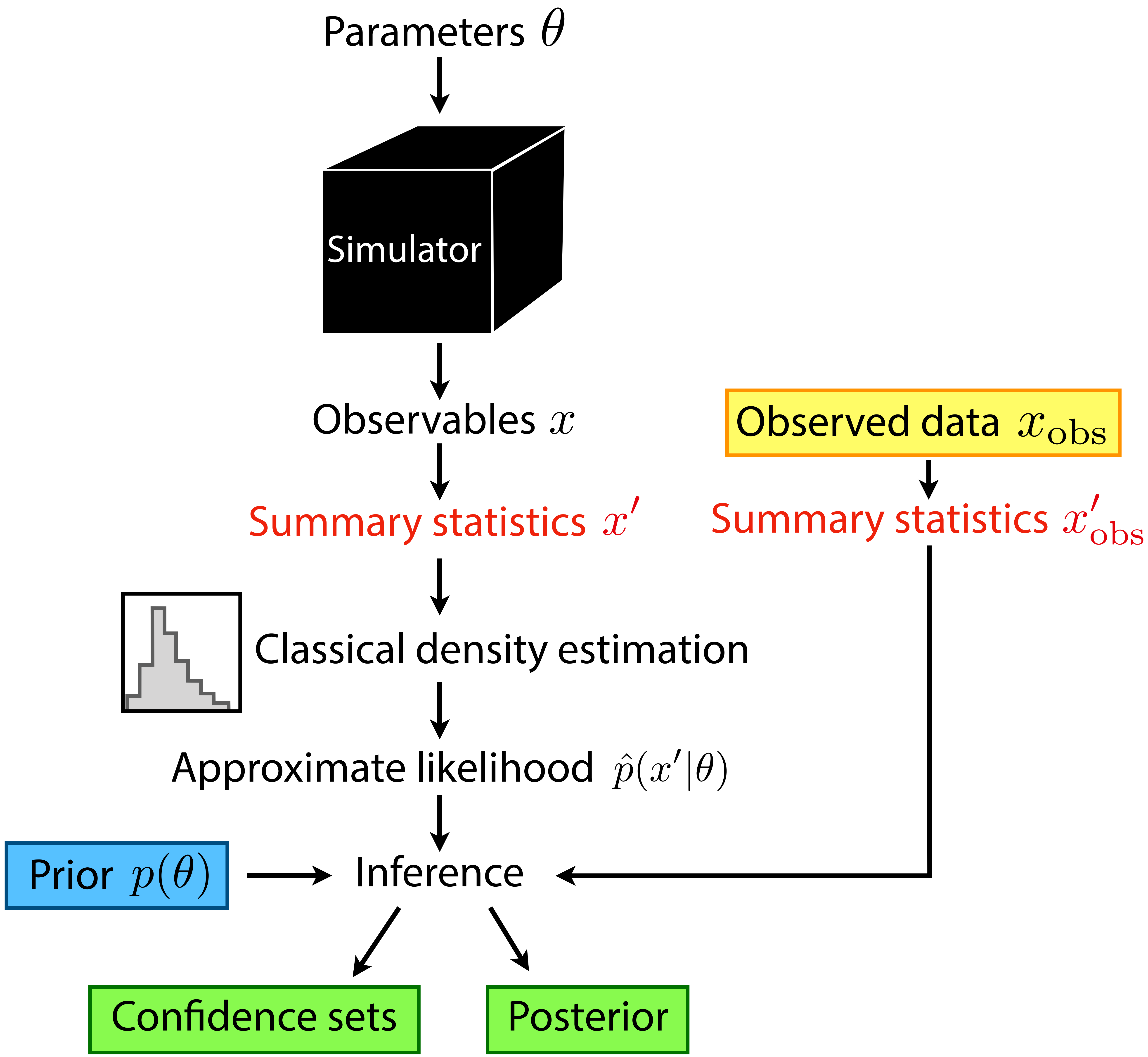
Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



Inference by estimating the likelihood

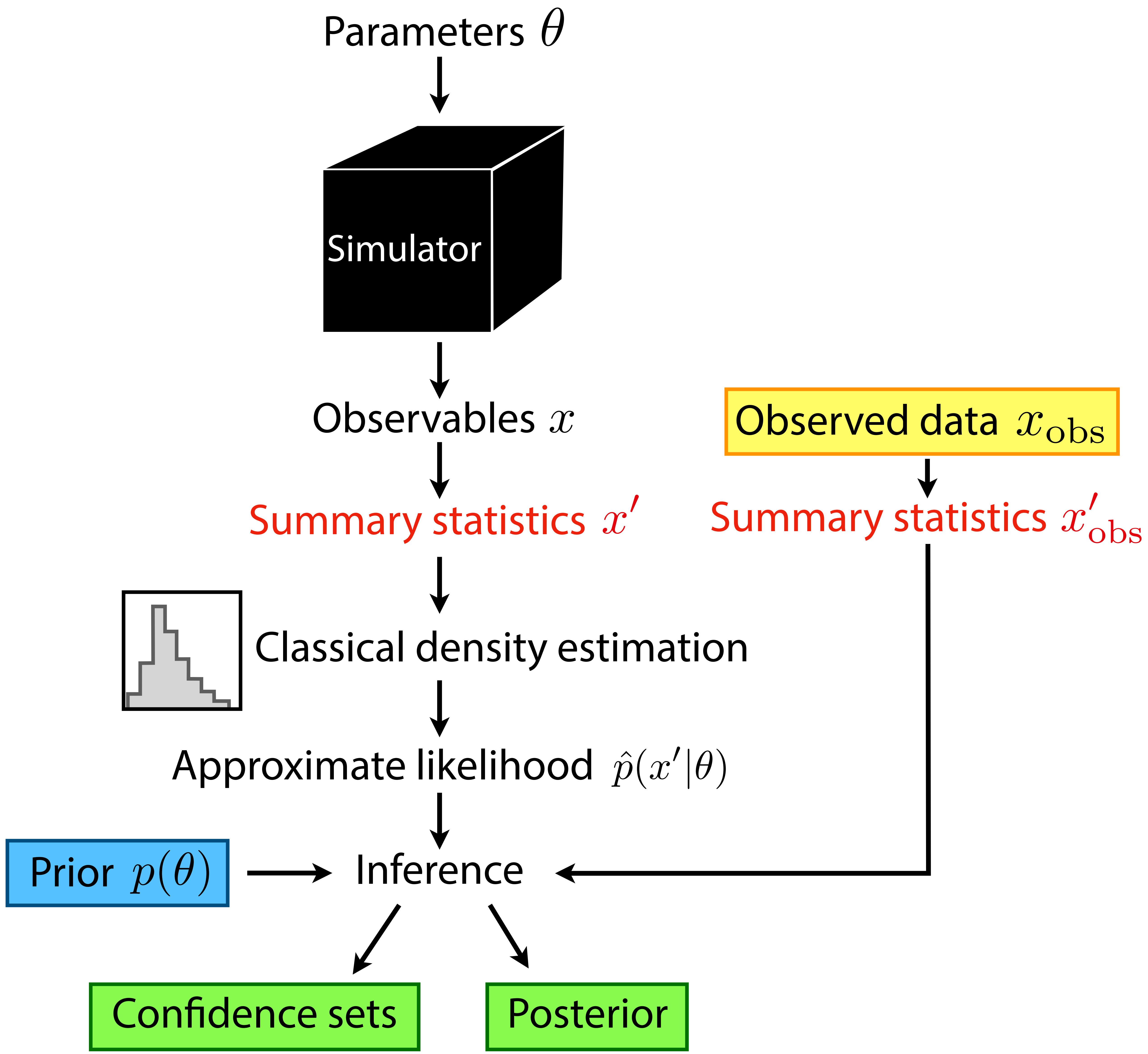
[e.g. P. Diggle, R. Gratton 1984]



- Compression to summary statistics loses information & reduces quality of inference
- Curse of dimensionality: does not scale to more than a few summary statistics

Inference by estimating the likelihood

[e.g. P. Diggle, R. Gratton 1984]



- Compression to summary statistics loses information & reduces quality of inference
- Curse of dimensionality: does not scale to more than a few summary statistics

pyhf ∂ifferentiable ℒikelihoods zfit cabinetry

What if we could estimate the likelihood...

- for high-dimensional observables, including correlations?

  like MEM: no need to pick summary statistics

- including state-of-the-art shower and detector models?

  allowing for extra radiation, no need for transfer functions

- in microseconds?

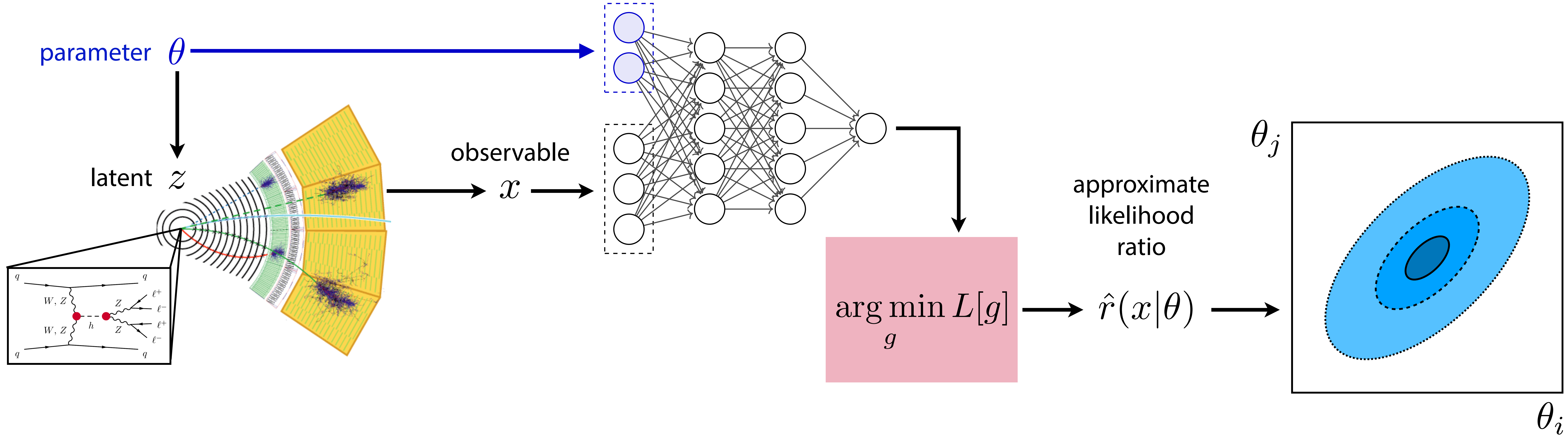
  amortized inference: train once, then always evaluate fast

- requiring less training examples than established machine learning methods?

  using matrix element information: "ML version of MEM"

Learning with Simulated Data

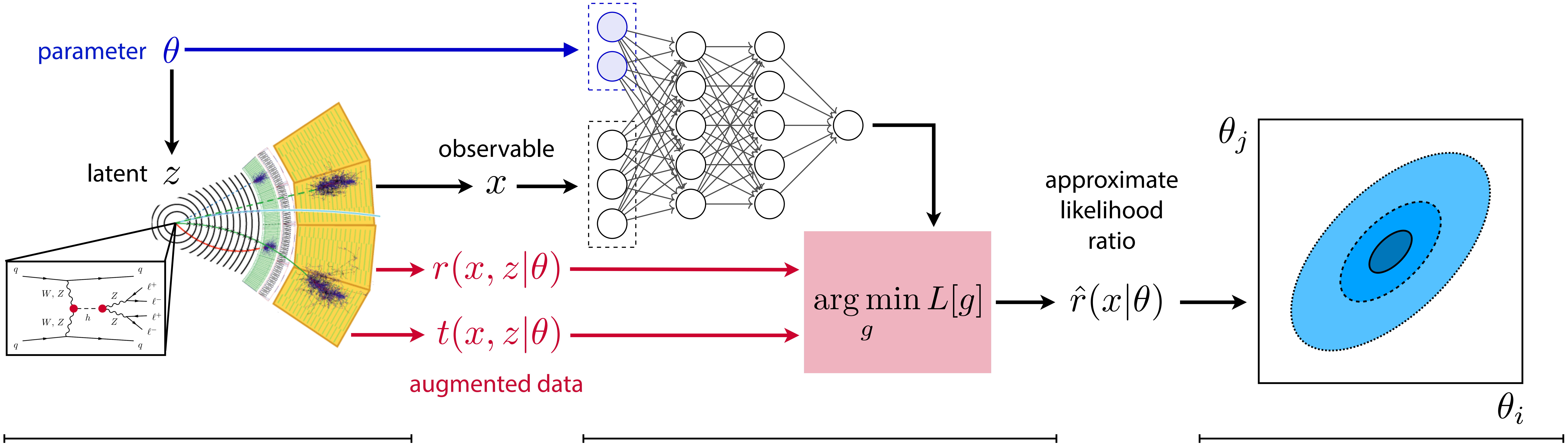
arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90



Simulation	Machine Learning	Inference
“Mining gold”: Extract additional information from simulator	Use this information to train estimator for likelihood ratio	Limit setting with standard hypothesis tests

Learning with Augmented Data

arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90



Simulation

“Mining gold”: Extract additional information from simulator

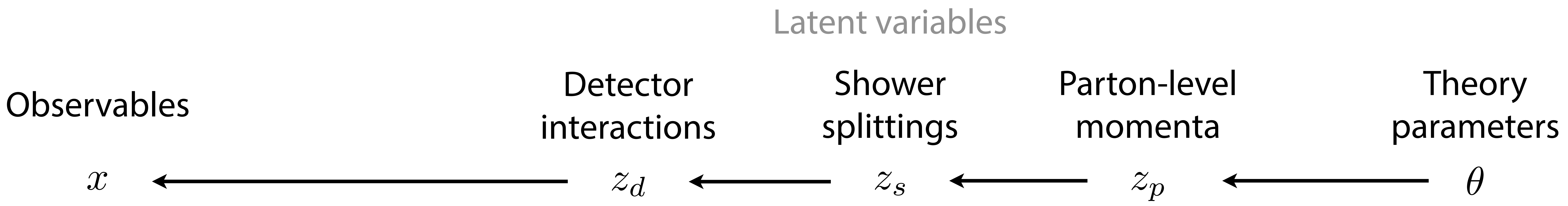
Machine Learning

Use this information to train estimator for likelihood ratio

Inference

Limit setting with standard hypothesis tests

Mining gold from the simulator



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

Intractable integrals

Mining gold from the simulator

Latent variables

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

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

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

⇒ For each simulated event, we can calculate the joint likelihood ratio which depends on the specific evolution of the simulation:

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

The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

("How much more likely is this simulated event, including all intermediate states, for θ_0 compared to θ_1?")

We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

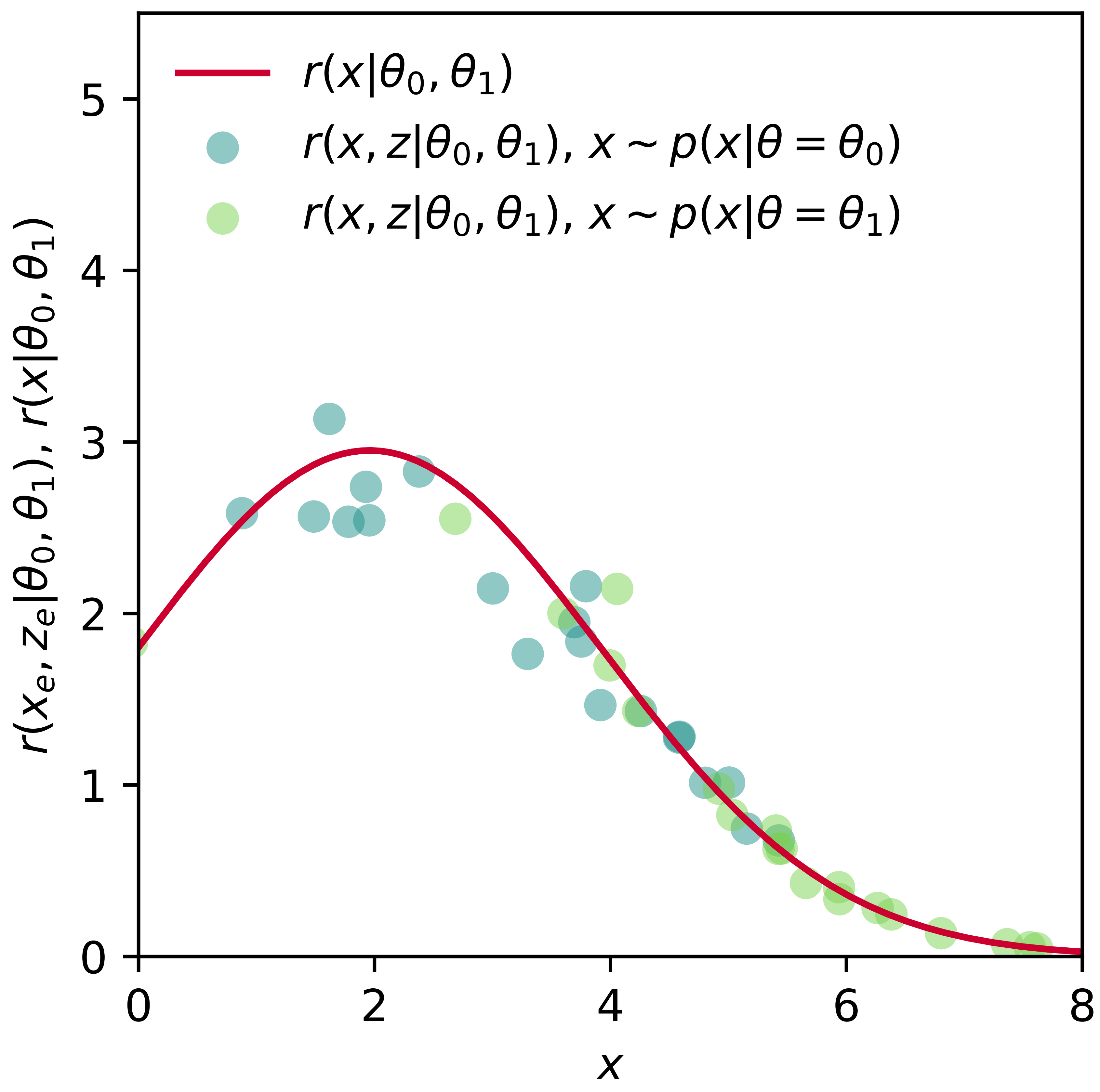
("How much more likely is the observation x for θ_0 compared to θ_1?")

The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

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



We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

↓

We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

With $r(x, z|\theta_0, \theta_1)$, we define a functional like

$$L_r[\hat{r}(x|\theta_0, \theta_1)] = \int \mathrm{d}x \int \mathrm{d}z \; p(x, z|\theta_1) \left[(\hat{r}(x|\theta_0, \theta_1) - r(x, z|\theta_0, \theta_1))^2 \right]$$

It is minimized by

$$\mathbb{E}_{z \sim p(z|x, \theta_1)} [r(x, z|\theta_0, \theta_1)] = \underset{\hat{r}(x|\theta_0, \theta_1)}{\arg\min} \; L_r[\hat{r}(x|\theta_0, \theta_1)]!$$

(And we can sample from $p(x, z|\theta)$ by running the simulator.)

The value of gold

We can calculate the joint likelihood ratio

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$

↓

We want the likelihood ratio function

$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

With $r(x, z|\theta_0, \theta_1)$, we define a functional like

$$L_r[\hat{r}(x|\theta_0, \theta_1)] = \int \mathrm{d}x \int \mathrm{d}z \; p(x, z|\theta_1) \left[(\hat{r}(x|\theta_0, \theta_1) - r(x, z|\theta_0, \theta_1))^2 \right]$$

It is minimized by

$$\mathbb{E}_{z \sim p(z|x,\theta_1)} [r(x, z|\theta_0, \theta_1)] = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min} \; L_r[\hat{r}(x|\theta_0, \theta_1)]!$$

(And we can sample from $p(x, z|\theta)$ by running the simulator.)

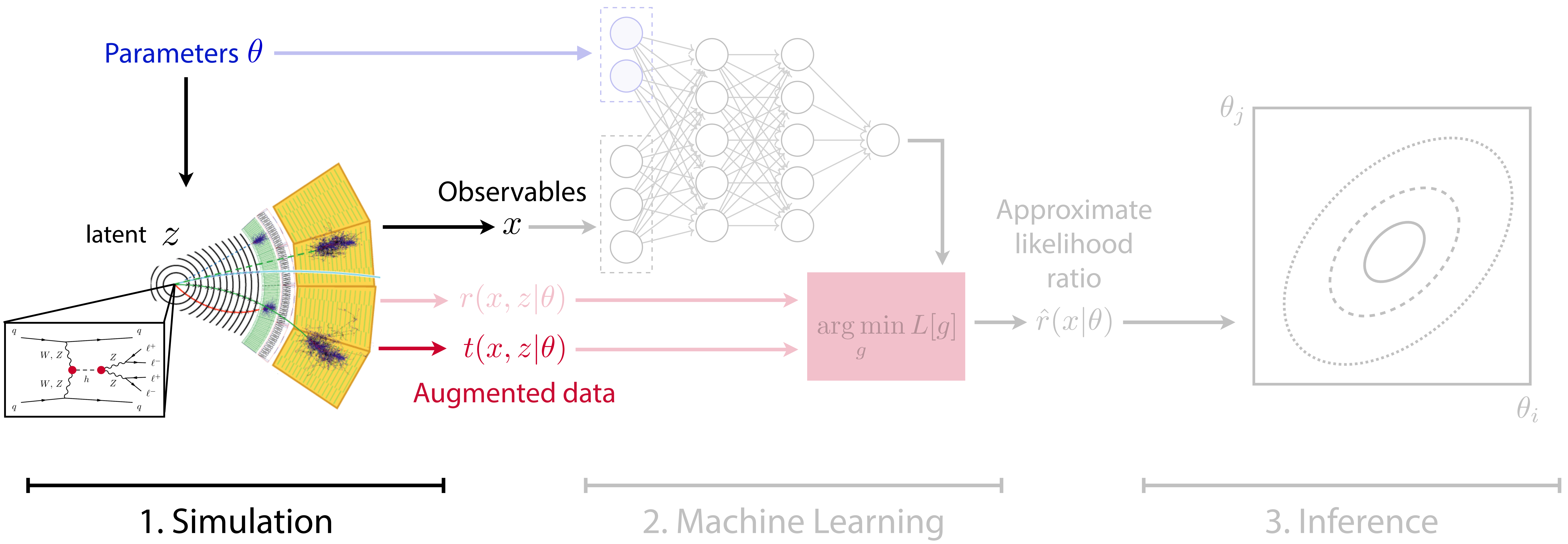
…. and then magic …

$$\mathbb{E}_{z \sim p(z|x,\theta_1)} [r(x, z|\theta_0, \theta_1)] = \int \mathrm{d}z \; p(z|x, \theta_1) \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)}$$

$$= \int \mathrm{d}z \; \frac{p(x, z|\theta_1)}{p(x|\theta_1)} \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)}$$

$$= r(x|\theta_0, \theta_1) \; !$$

Learning with Augmented Data



Learning the score (related to optimal observables)

Similar to the joint likelihood ratio, from the simulator we can extract the joint score

$$t(x, z|\theta_0) \equiv \nabla_\theta \log p(x, z_d, z_s, z_p|\theta)\Big|_{\theta_0}$$

↓

We want the score

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

Learning the score (related to optimal observables)

Similar to the joint likelihood ratio, from the simulator we can extract the joint score

$$t(x, z|\theta_0) \equiv \nabla_\theta \log p(x, z_d, z_s, z_p|\theta)\Big|_{\theta_0}$$

↓

We want the score

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

Given $t(x, z|\theta_0)$,
we define the functional

$$L_t[\hat{t}(x|\theta_0)] = \int \mathrm{d}x \int \mathrm{d}z \;\; p(x, z|\theta_0) \; \left[\left(\hat{t}(x|\theta_0) - t(x, z|\theta_0) \right)^2 \right].$$

One can show it is minimized by

$$t(x|\theta_0) = \underset{\hat{t}(x|\theta_0)}{\arg\min} \, L_t[\hat{t}(x|\theta_0)] \, .$$

Again, we implement this minimization through machine learning.

MadMiner automates all of these methods.

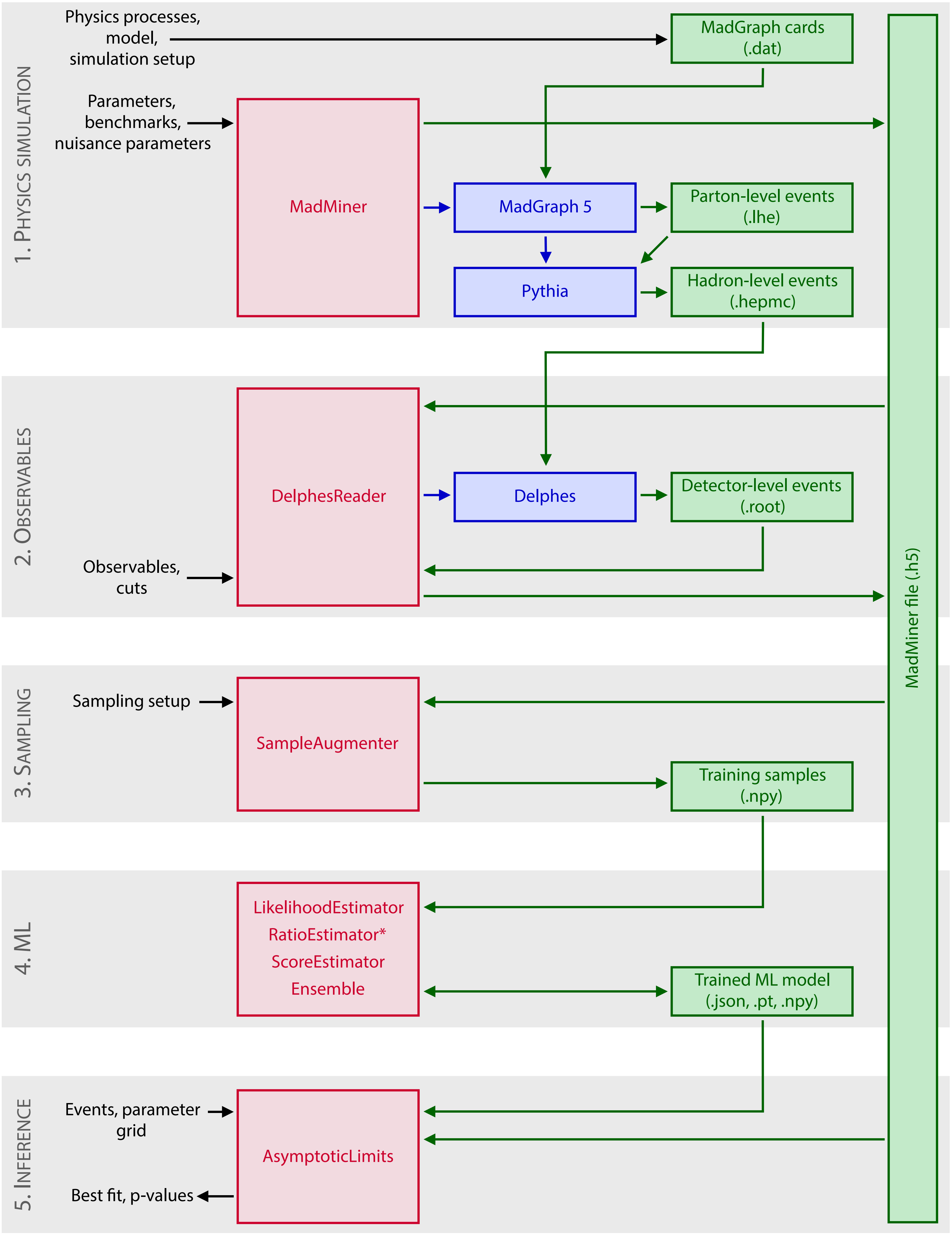
[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

MadMiner

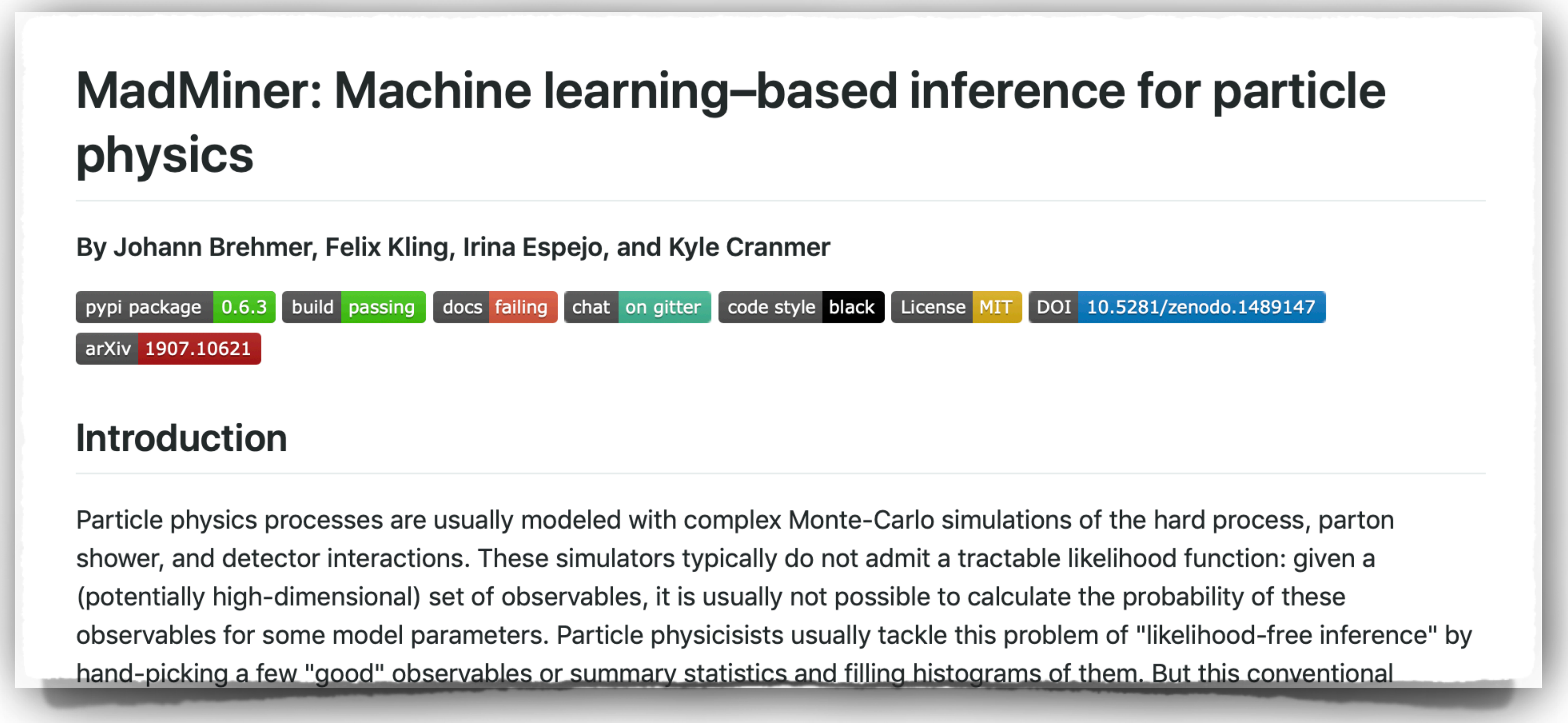
[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

New Python package MadMiner makes it straightforward to apply the new techniques to LHC problems

- Out of the box: Pheno-level analyses
  - MadGraph, Pythia, Delphes, (could be GEANT4)
  - Systematic uncertainties from PDF / scale variation
- Scalable to state-of-the-art experimental tools
  - Mostly requires bookkeeping of fully differential cross sections
- Modular interface
  - Extensive documentation
  - Embedded into Python / ML ecosystem



MadMiner resources


MadMiner: Machine learning–based inference for particle physics

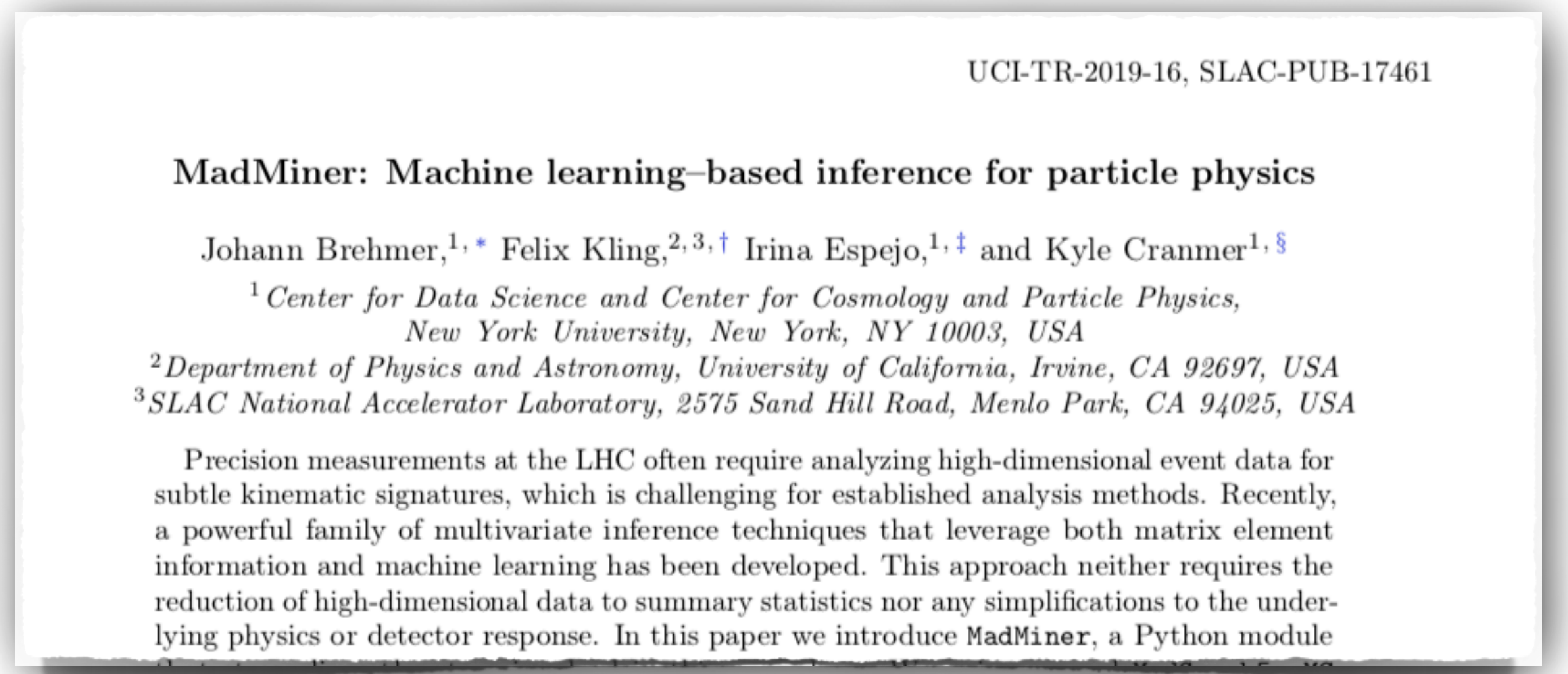
By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package 0.6.3 | build passing | docs failing | chat on gitter | code style black | License MIT | DOI 10.5281/zenodo.1489147 | arXiv 1907.10621

Introduction

Particle physics processes are usually modeled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional

Repository and tutorials:
github.com/johannbrehmer/madminer


UCI-TR-2019-16, SLAC-PUB-17461

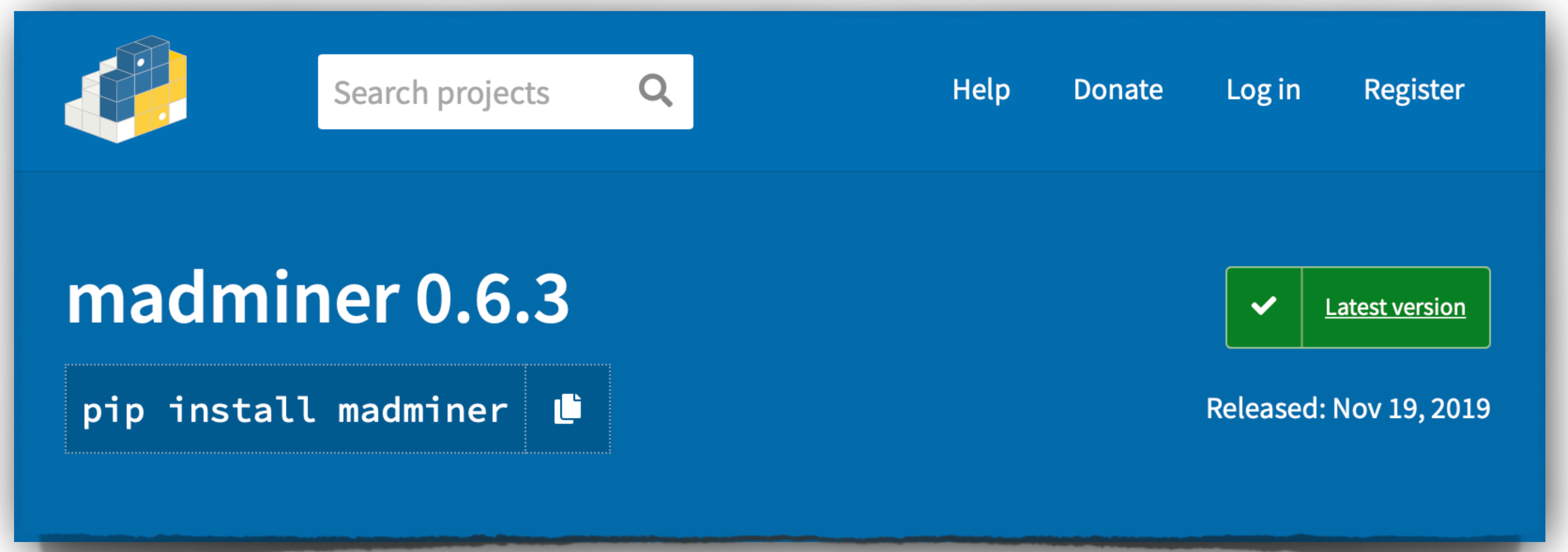
MadMiner: Machine learning–based inference for particle physics

Johann Brehmer,[1,*] Felix Kling,[2,3,†] Irina Espejo,[1,‡] and Kyle Cranmer[1,§]

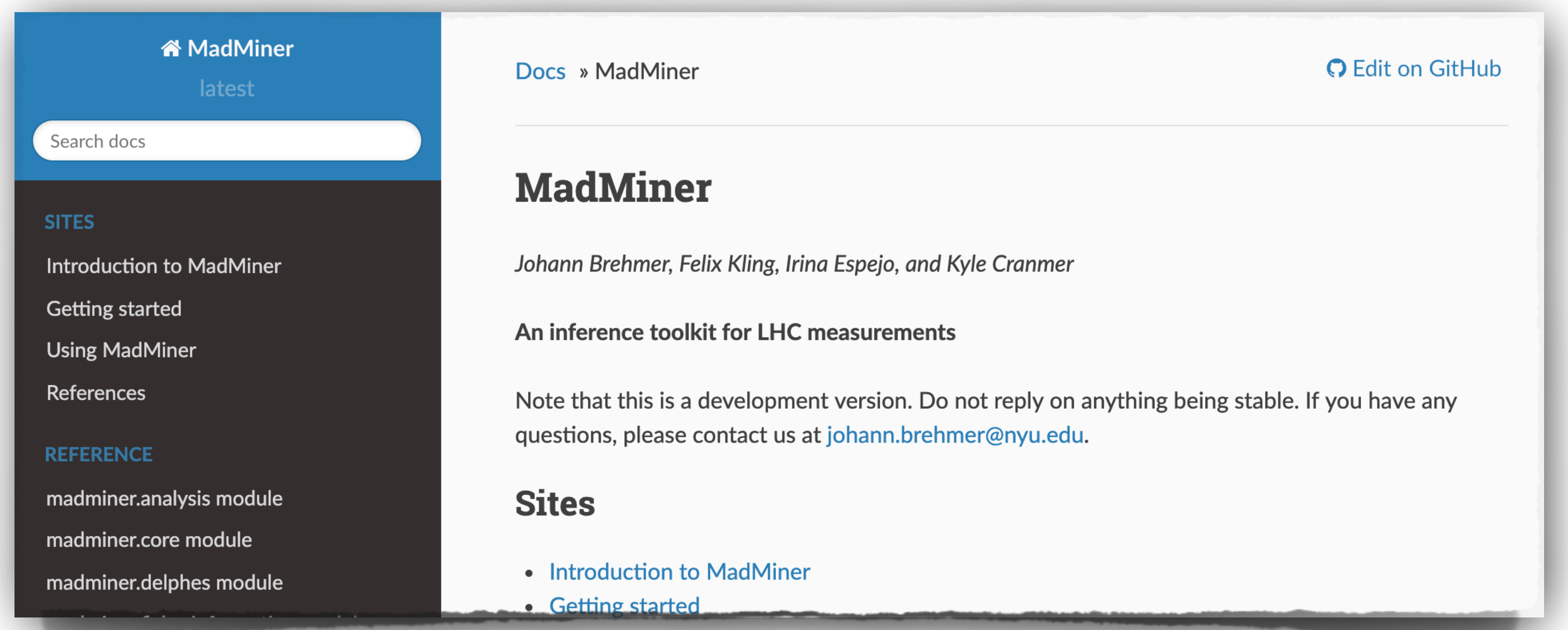
[1] *Center for Data Science and Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[2] *Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[3] *SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA*

Precision measurements at the LHC often require analyzing high-dimensional event data for subtle kinematic signatures, which is challenging for established analysis methods. Recently, a powerful family of multivariate inference techniques that leverage both matrix element information and machine learning has been developed. This approach neither requires the reduction of high-dimensional data to summary statistics nor any simplifications to the underlying physics or detector response. In this paper we introduce MadMiner, a Python module

Paper with detailed explanations:
1907.10621



Installation:
`pip install madminer`

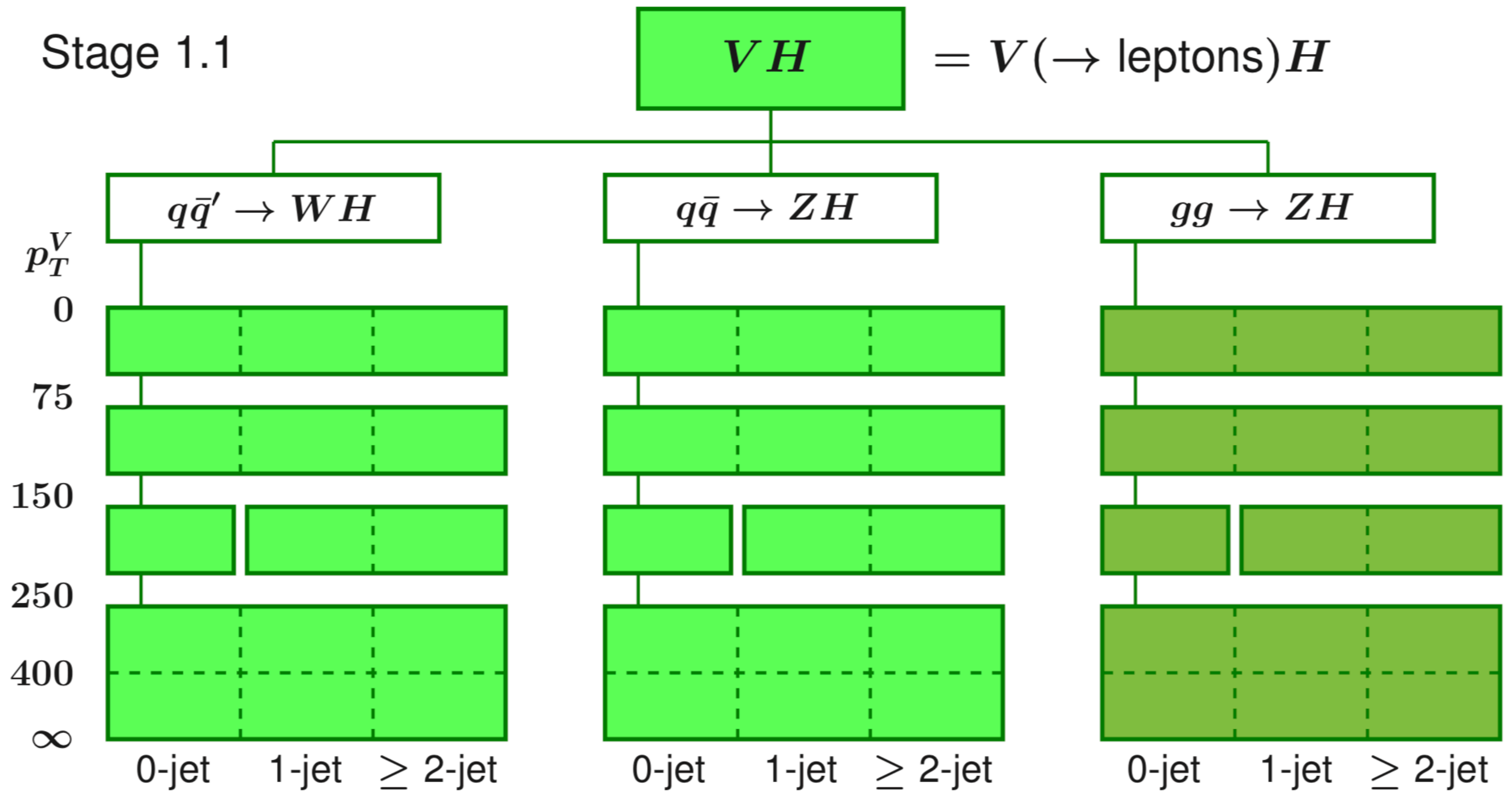


API documentation:
madminer.readthedocs.io

An example: STXS vs. MadMiner in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]



- Let's check! How much information on

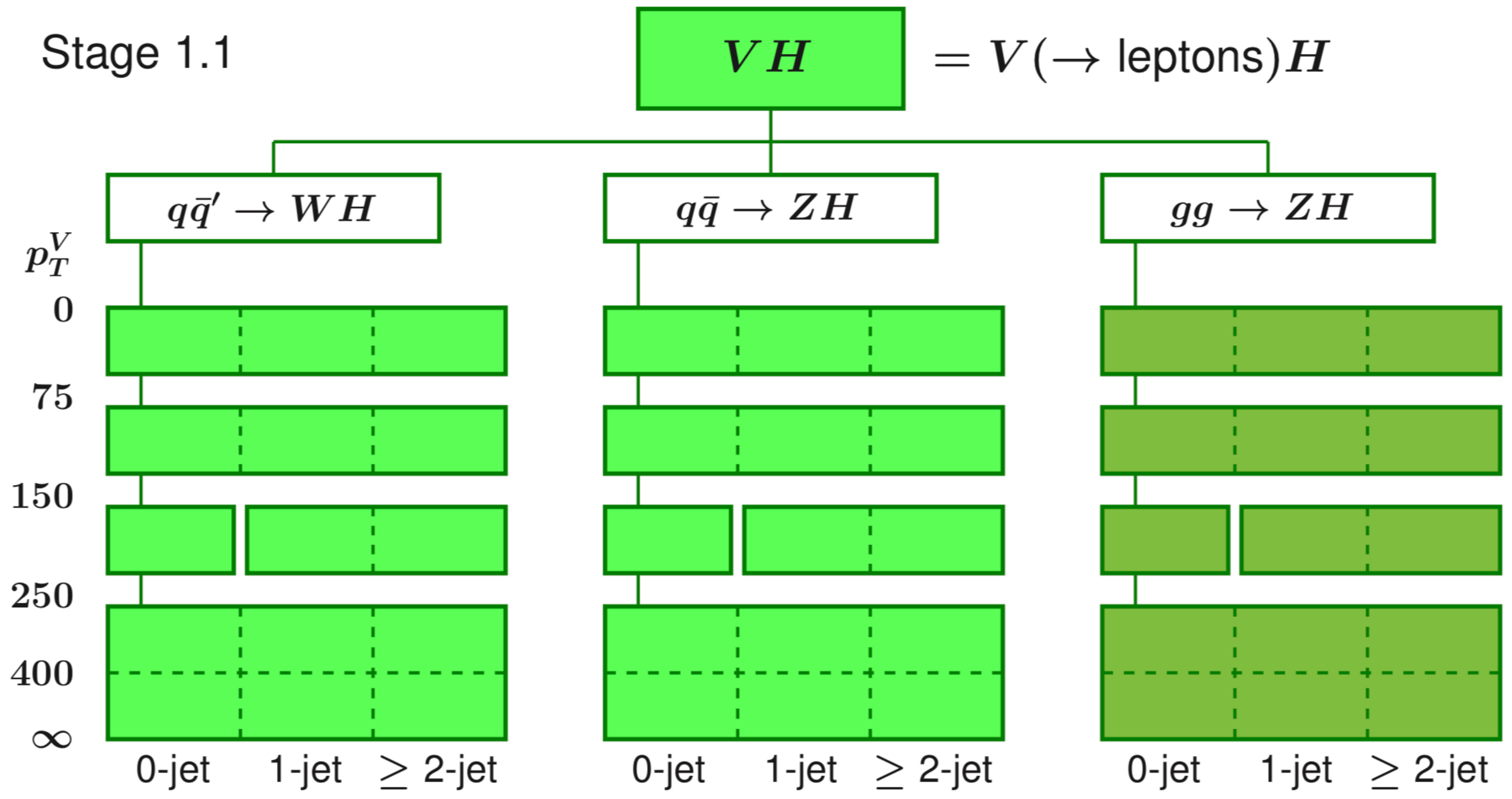
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L) ,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$?

An example: STXS vs. MadMiner in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]

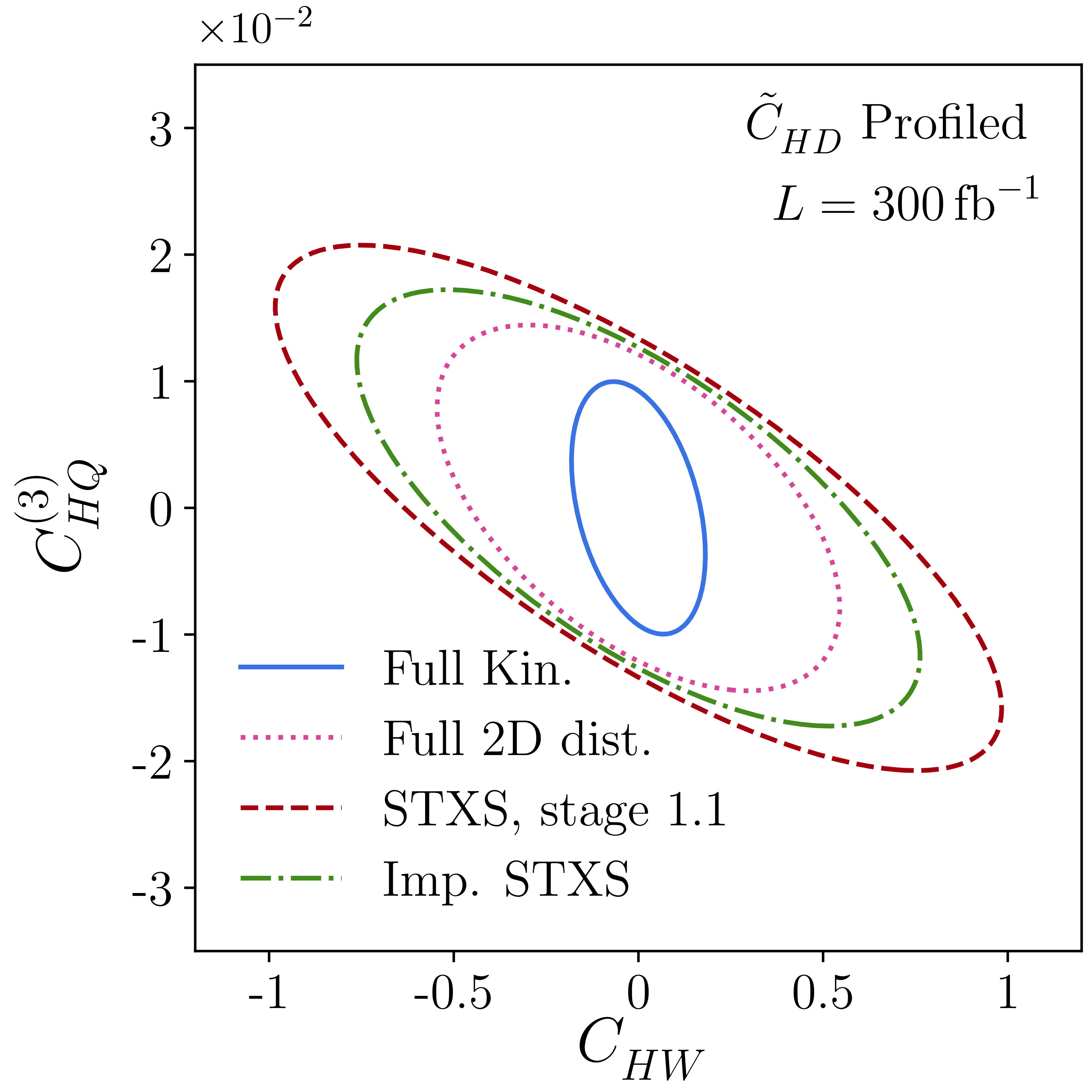


- Let's check! How much information on

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i \overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\,,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$?

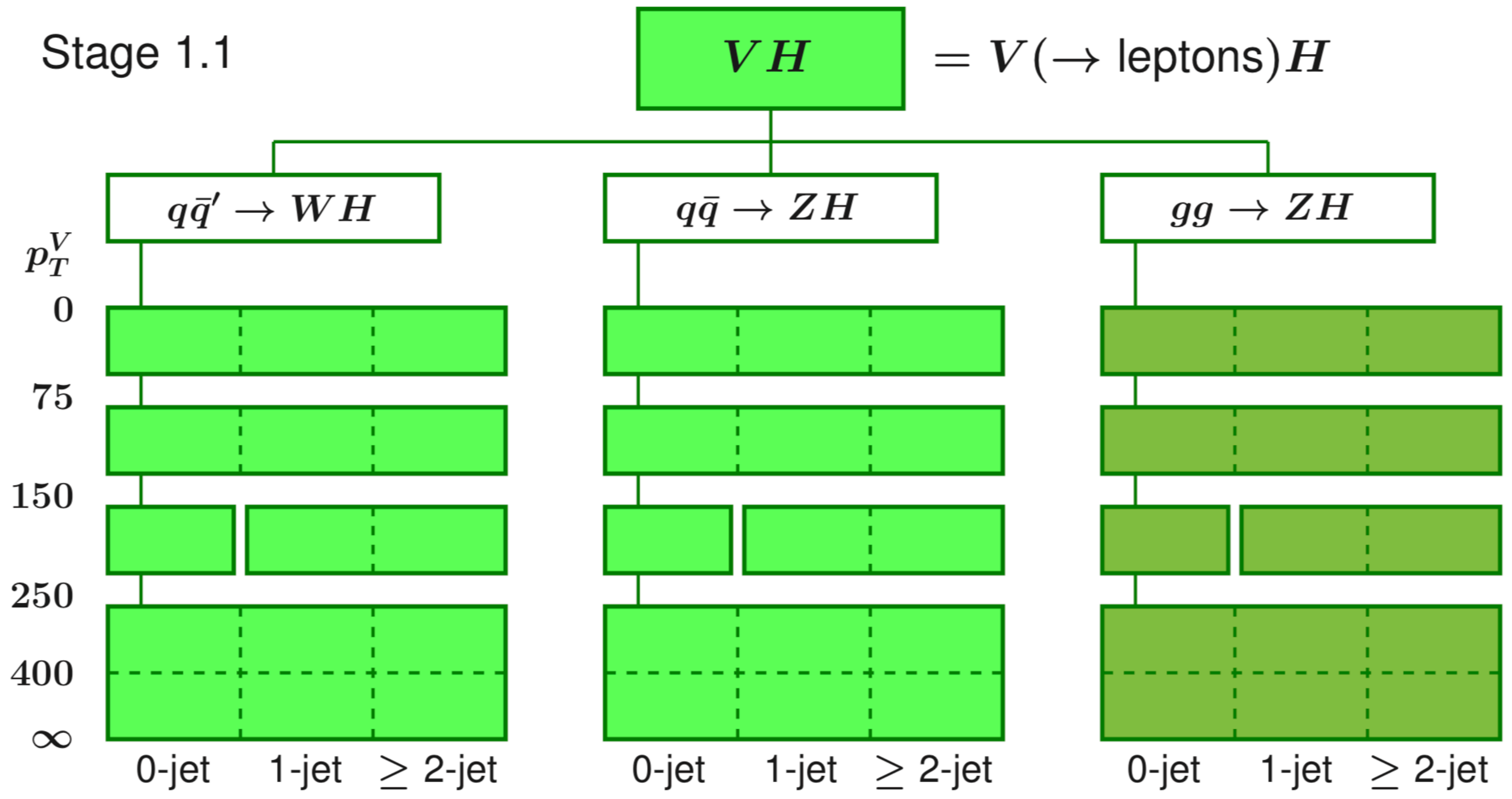
- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



An example: STXS vs. MadMiner in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]

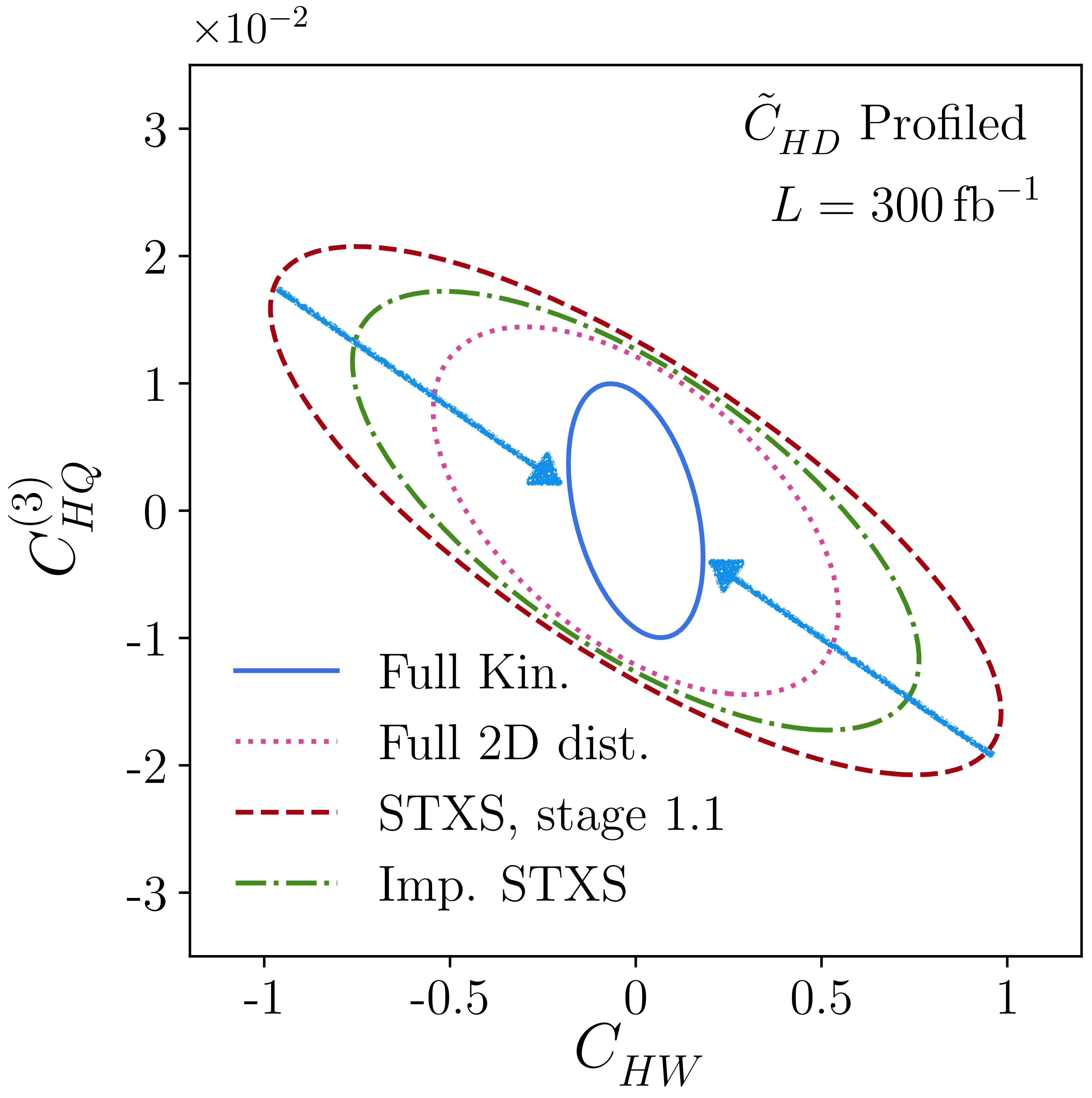


- Let's check! How much information on

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i \overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$?

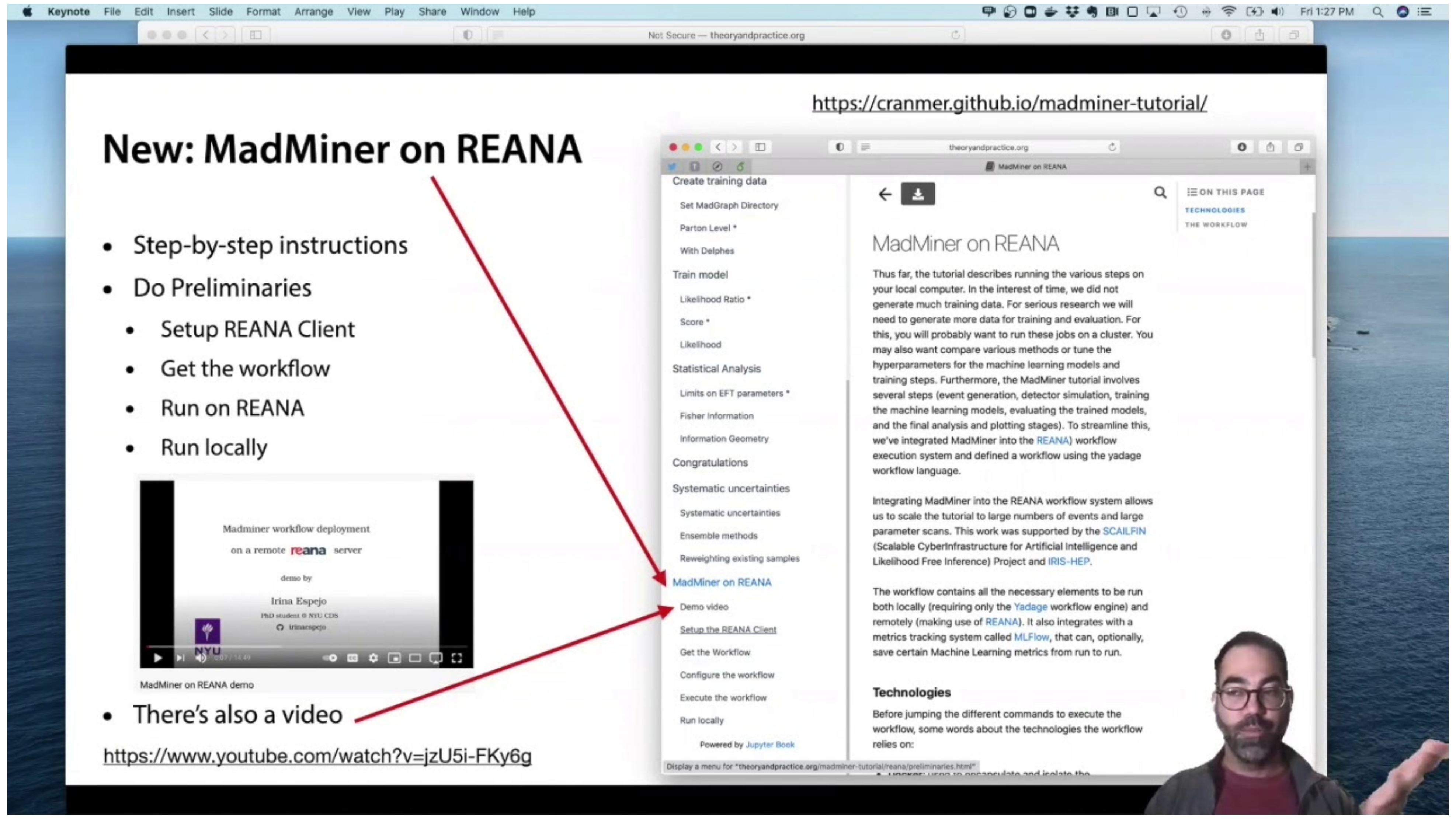
- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



Video tutorials

1h15m video tutorial through

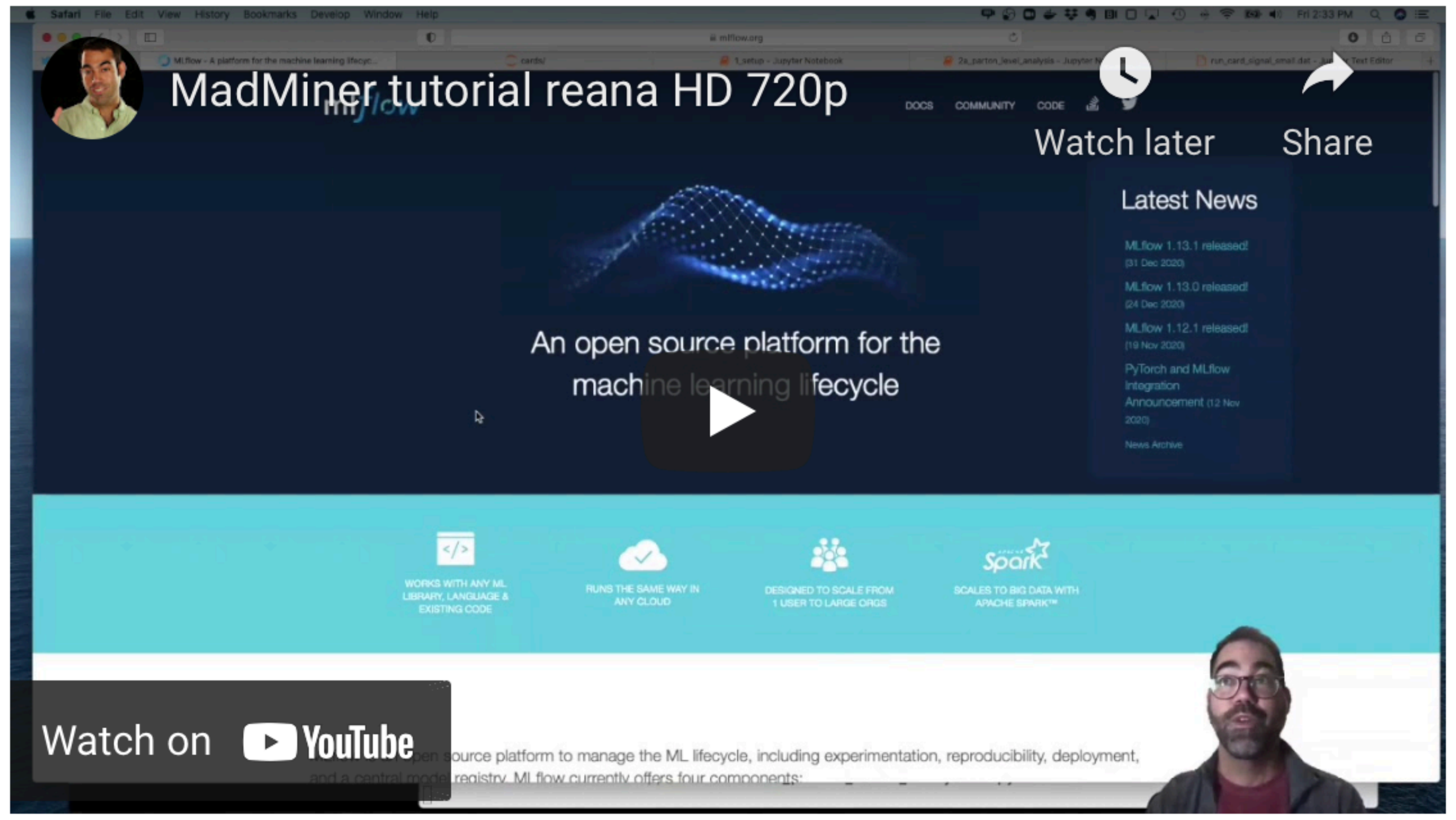
https://www.youtube.com/watch?v=3kERPu5Wmgk



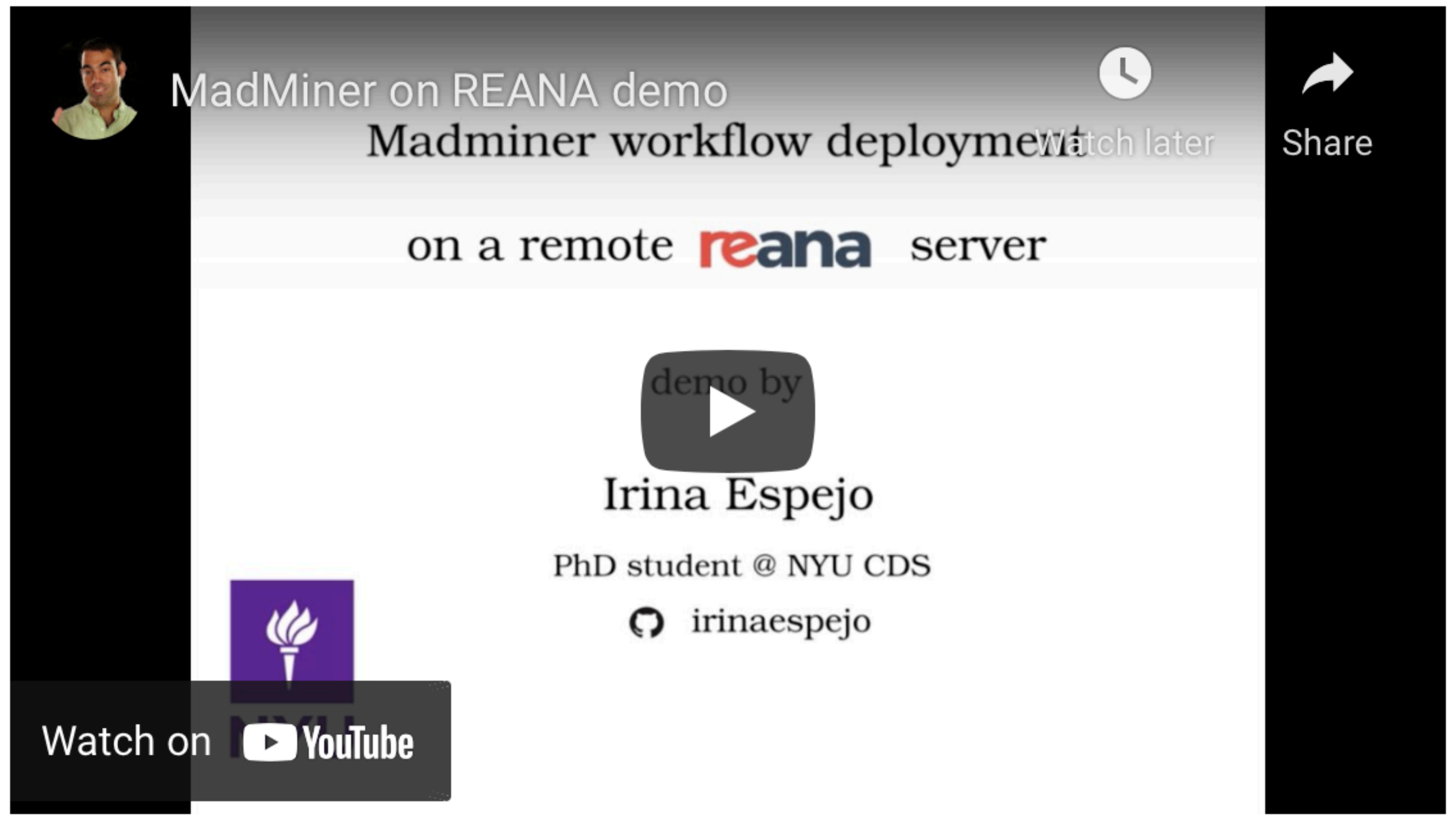
30m video running on REANA

https://www.youtube.com/watch?v=IccGgEdUQJ8

Below Kyle Cranmer gives an overview of the MadMiner workflows and how to s REANA and yadage.



Below Irina Espejo gives a demo of running MadMiner on the REANA instance a Brookhaven National Lab.



reana

Reproducible research data analysis platform

Flexible	Scalable	Reusable	Free
Run many computational workflow engines.	Support for remote compute clouds.	Containerise once, reuse elsewhere. Cloud-native.	Free Software. MIT licence. Made with ♥ at CERN.

MadMiner Tutorial JupyterBook

http://theoryandpractice.org/madminer-tutorial/intro

MadMiner Tutorial

Introduction

Demo video

MadMiner Tutorial

Preliminaries

Overview

Define process to study *

Morphing

Interactive Morphing Demo

Create training data

Set MadGraph Directory

Parton Level *

With Delphes

Train model

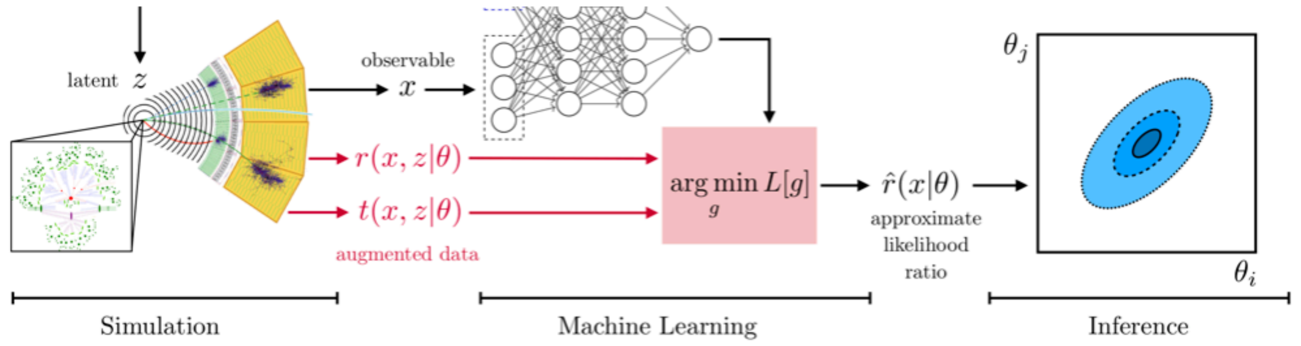
Likelihood Ratio *

Score *

Likelihood

Statistical Analysis

Limits on EFT parameters *



Introduction to MadMiner

Particle physics processes are usually modelled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicisists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional approach discards the information in all other observables and often does not scale well to high-dimensional problems.

In the three publications "Constraining Effective Field Theories With Machine Learning", "A Guide to Constraining Effective Field Theories With Machine Learning", and "Mining gold from implicit models to improve likelihood-free inference", a new approach has been developed. In a nut shell, additional information is extracted from the simulations that is closely related to the matrix elements that determine the hard process. This "augmented data" can be used to train neural networks to efficiently approximate arbitrary likelihood ratios. We playfully call this process "mining gold" from the simulator, since this information may be hard to get, but turns out to be very valuable for inference.

But the gold does not have to be hard to mine. This package automates these inference strategies. It wraps around the simulators MadGraph and Pythia, with different options for the detector simulation. All steps in the analysis chain from the simulation to the extraction of the augmented data, their processing, and the training and evaluation of the neural estimators are implemented.

Special thanks to help with docker, binder, etc. points 👇

Irina Espejo

Sinclert Perez

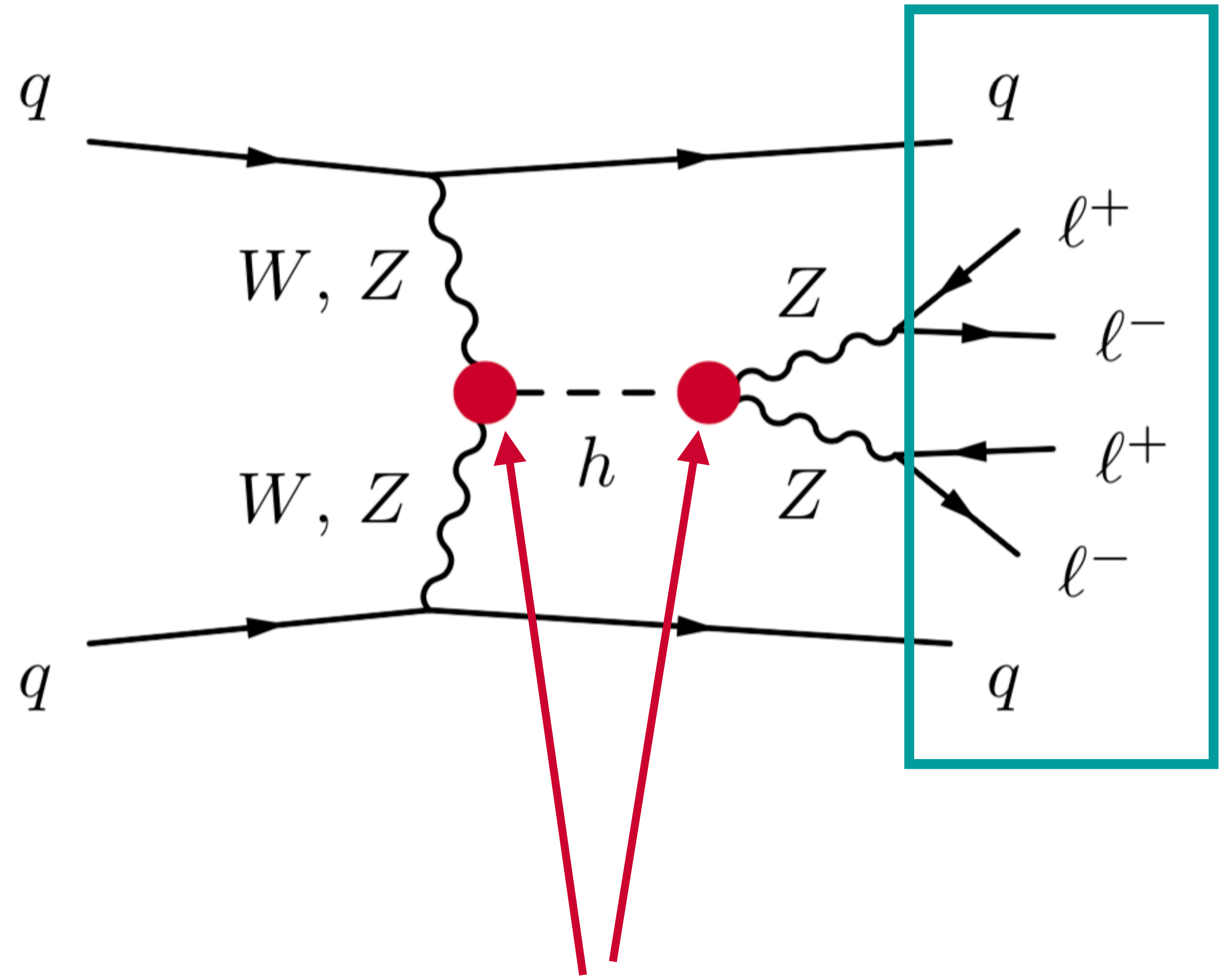
Matthew Feickert

More examples & Backup

Proof of concept: Higgs production in weak boson fusion

[JB, K. Cranmer, G. Louppe, J. Pavez
1805.00013, 1805.00020]

at least 16-dimensional
observable space

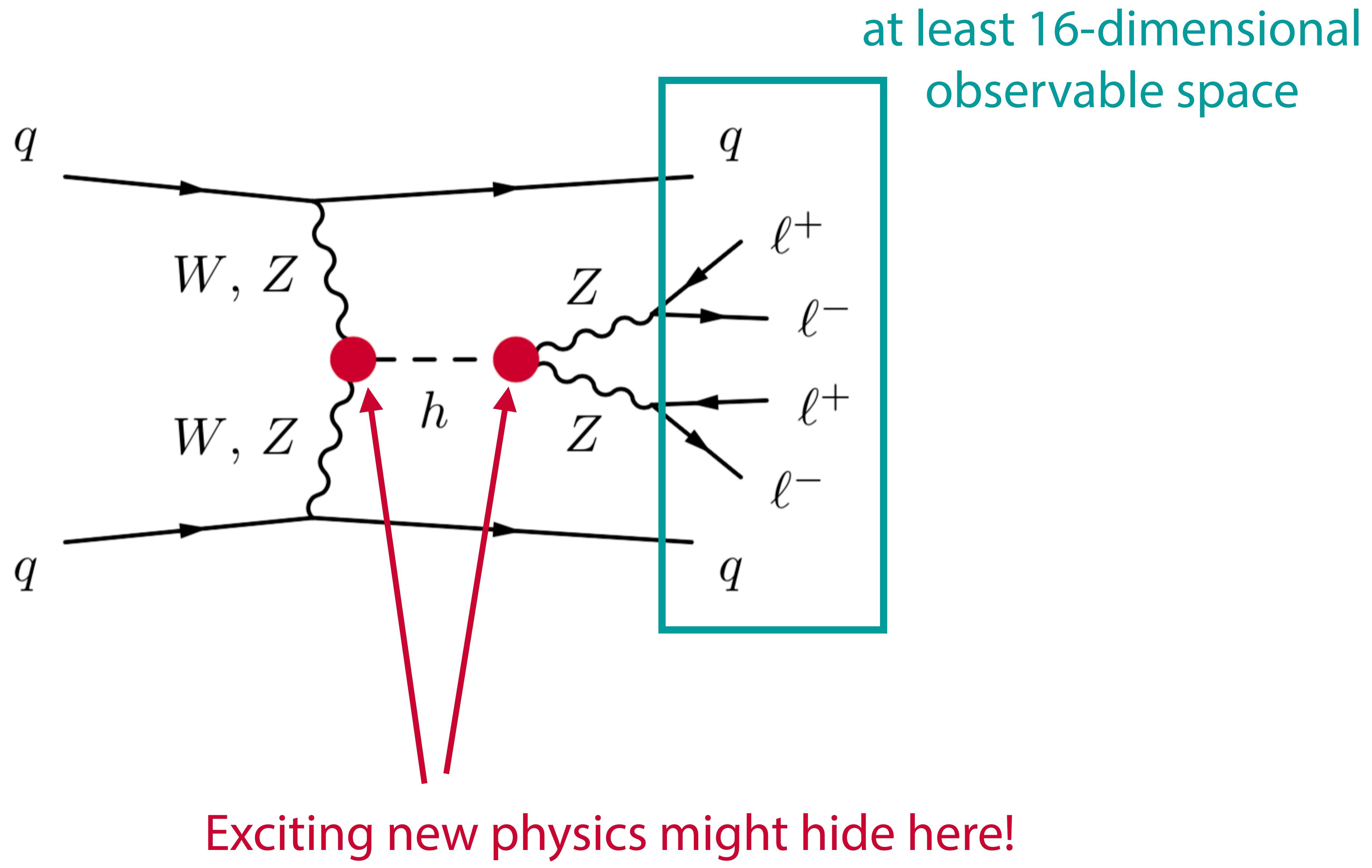


Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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

Proof of concept: Higgs production in weak boson fusion

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020]



We parameterize it with two EFT coefficients:

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

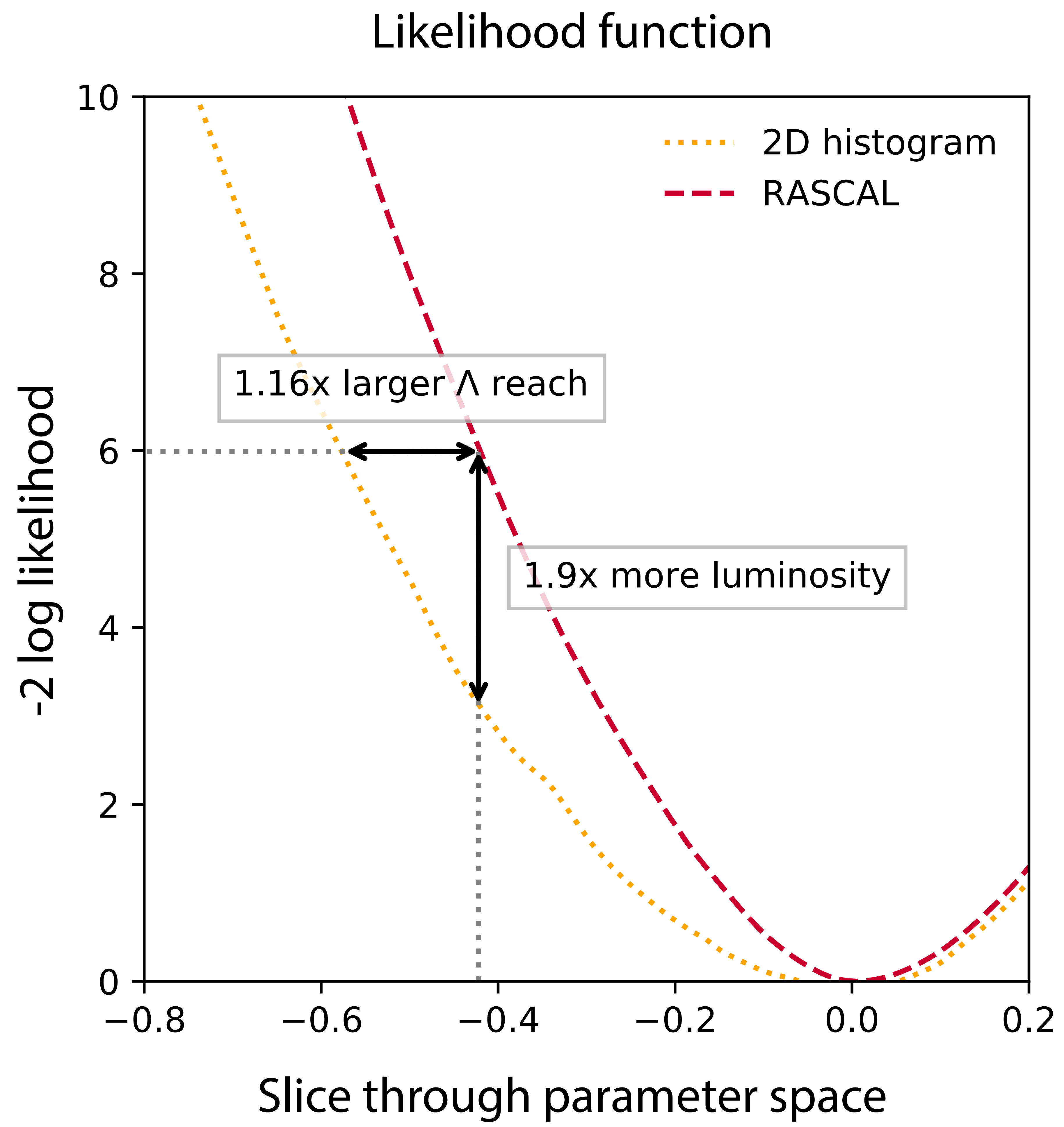
Goal: constrain the two EFT parameters

- 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

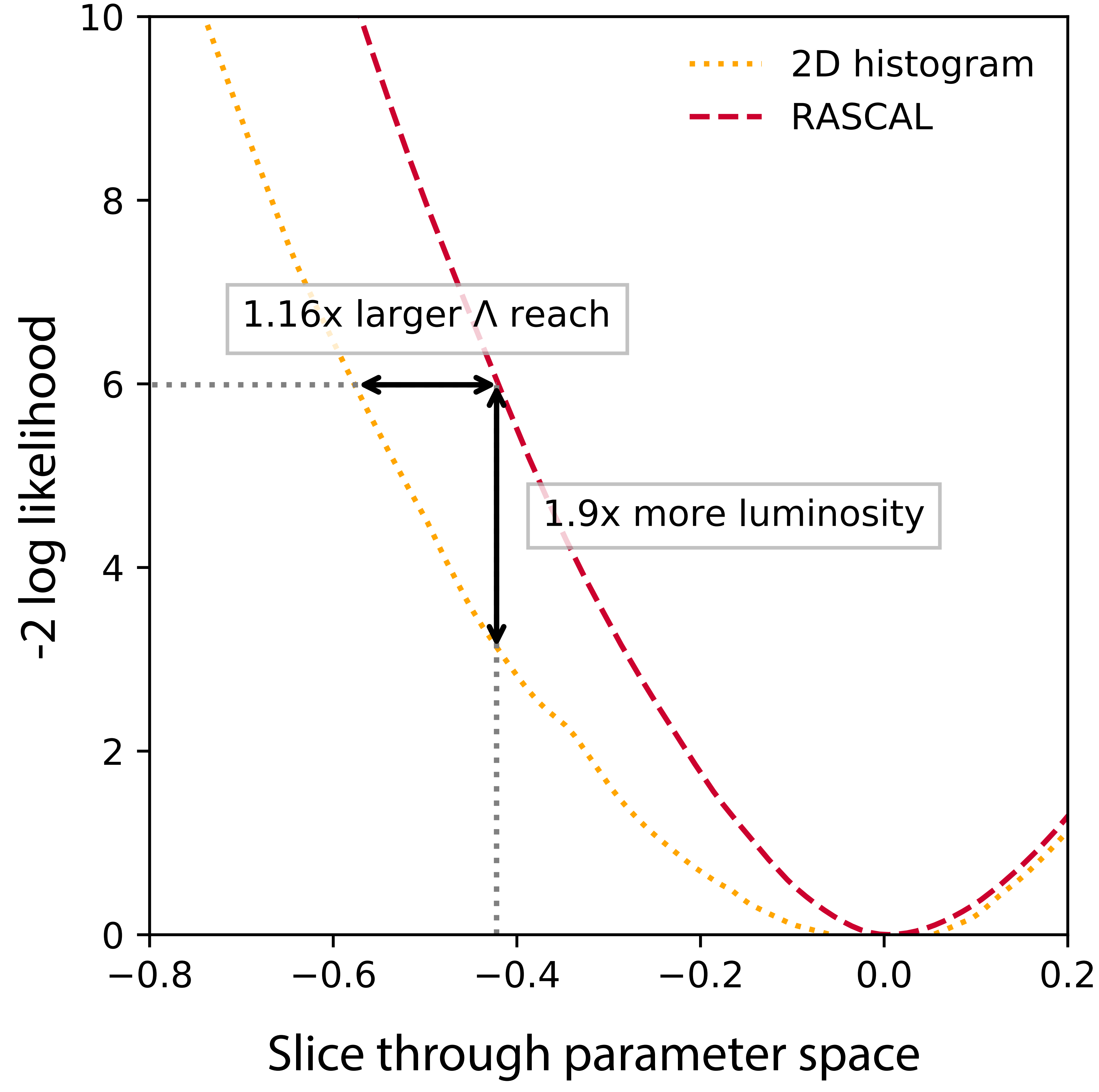
Better sensitivity to new physics



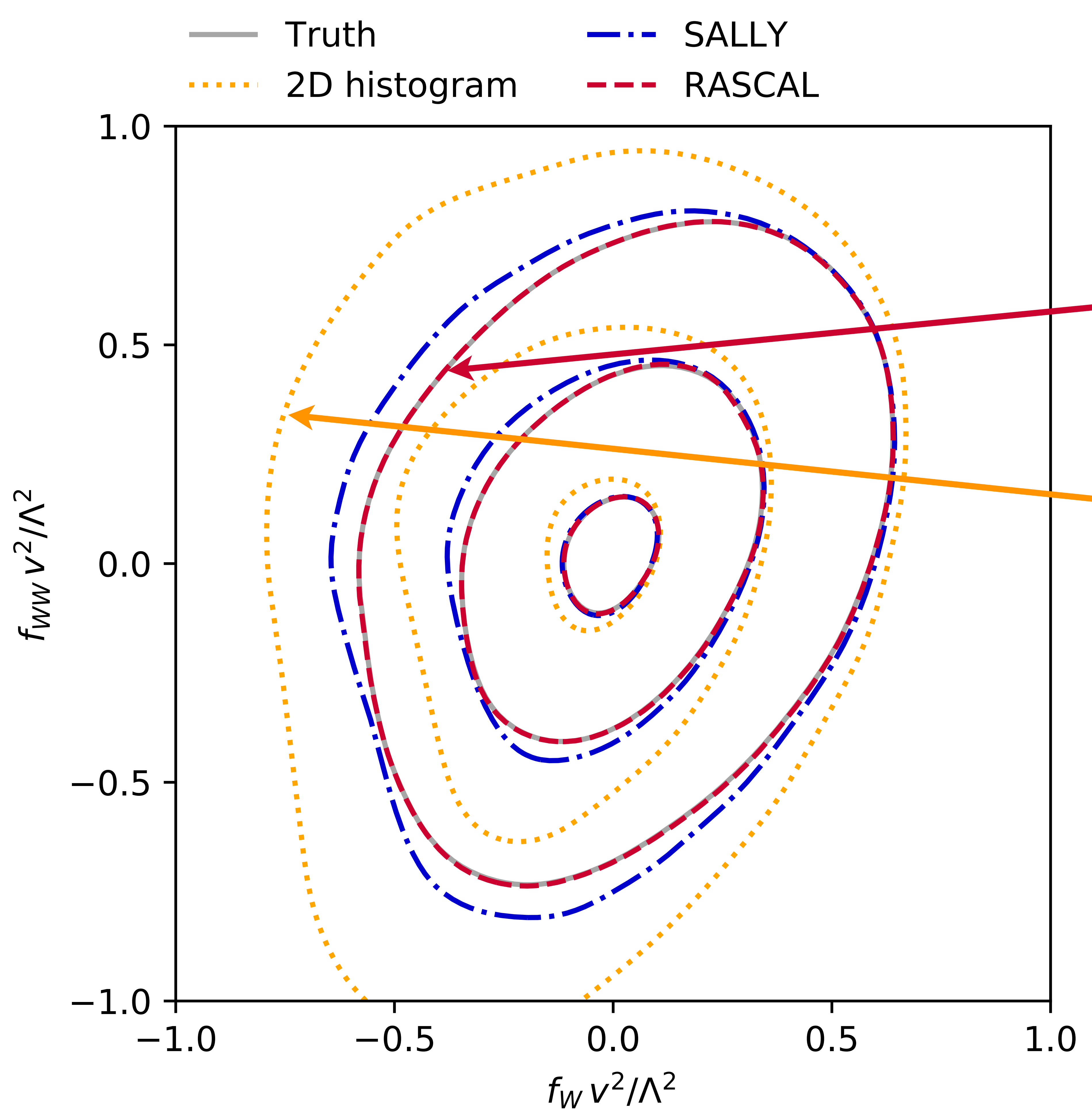
Results are based on 36 observed events, assuming SM

Better sensitivity to new physics

Likelihood function



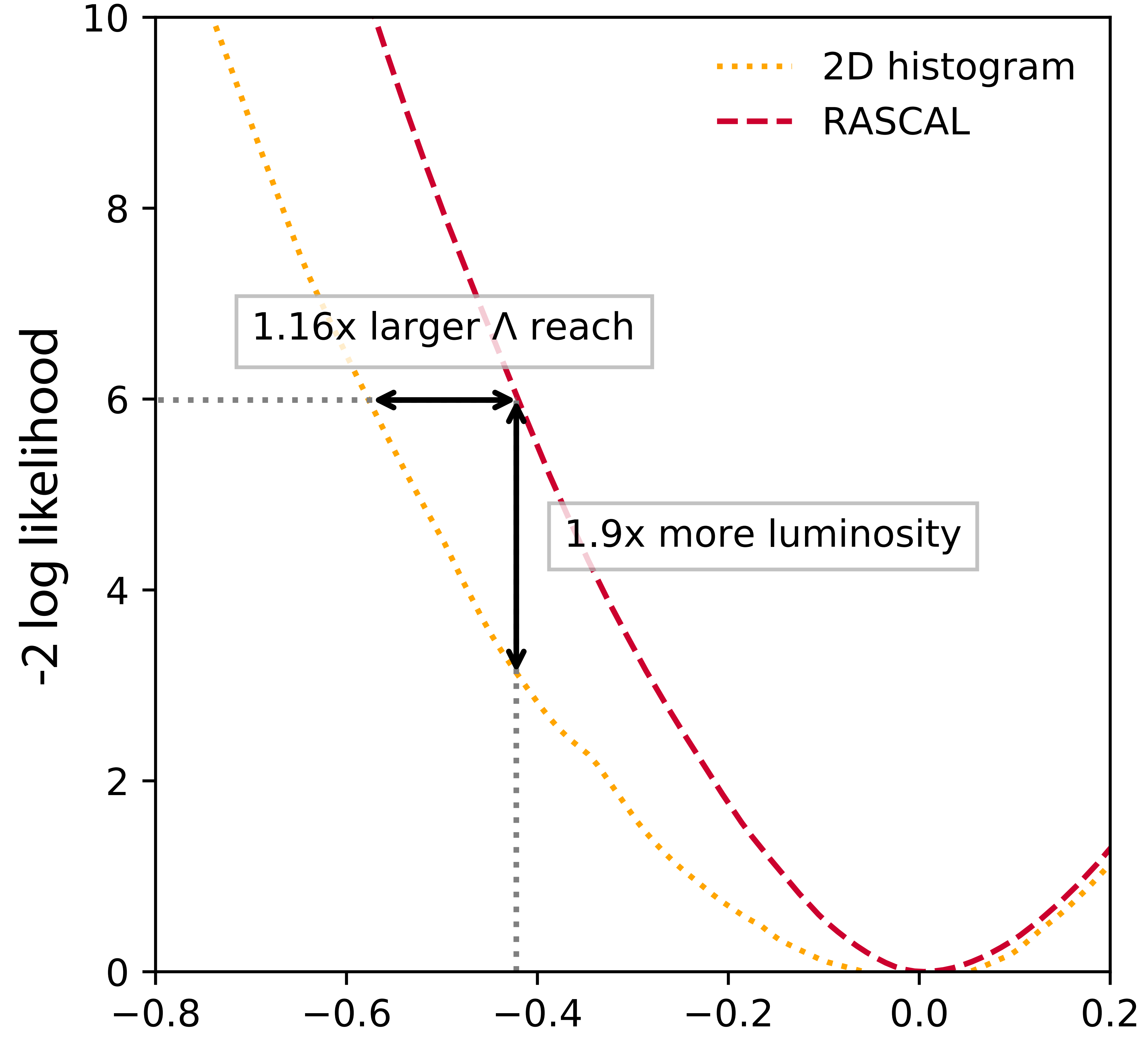
Expected exclusion limits at 68%, 95%, 99.7% CL



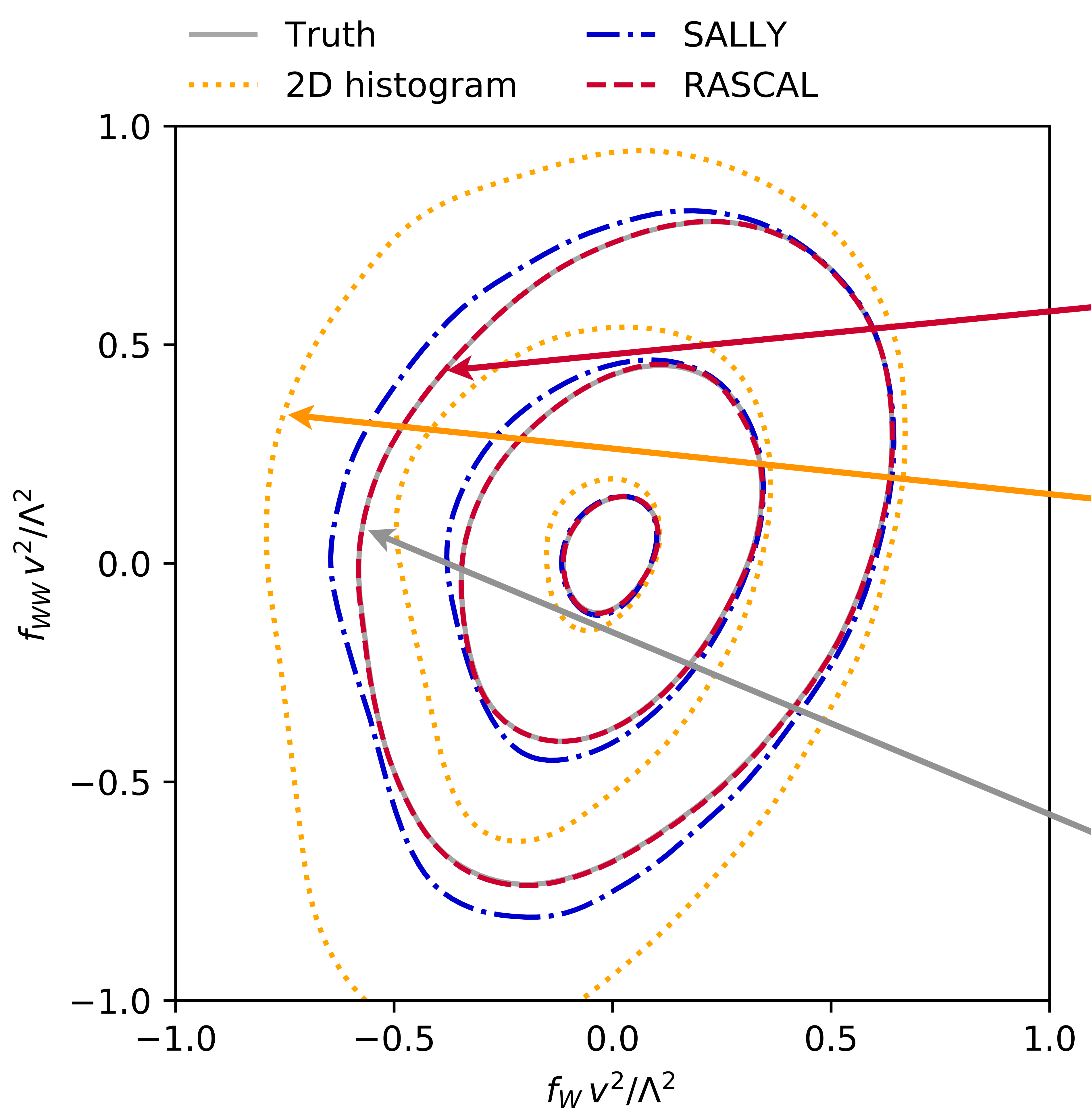
RASCAL and SALLY enables stronger limits than 2D histogram

Results are based on 36 observed events, assuming SM

Better sensitivity to new physics



Expected exclusion limits at 68%, 95%, 99.7% CL



RASCAL and SALLY enables stronger limits than 2D histogram

Limits from RASCAL indistinguishable from true likelihood

(usually we don't have that)

Results are based on 36 observed events, assuming SM

Constraining operators in ttH effectively

[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

- Pheno-level analysis of

  $$pp \to t\bar{t}\,h \to (b\ell^+)\,(\bar{b}\ell^-)\,(\gamma\gamma)\,E_T^{\text{miss}}$$

  with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

  $$\mathcal{O}_u = -\frac{1}{v^2}(H^\dagger H)(H^\dagger \bar{Q}_L)u_R\,,\ \mathcal{O}_G = \frac{g_s^2}{m_W^2}(H^\dagger H)G^a_{\mu\nu}G_a^{\mu\nu}\,,$$

  $$\mathcal{O}_{uG} = -\frac{4g_s}{m_W^2}y_u(H^\dagger \bar{Q}_L)\gamma^{\mu\nu}T_a u_R G^a_{\mu\nu}$$

Constraining operators in ttH effectively

[JB, F. Kling, I. Espejo, K. Cranmer 1907.10621]

- Pheno-level analysis of

$$pp \to t\bar{t}\,h \to (b\ell^+)\,(\bar{b}\ell^-)\,(\gamma\gamma)\,E_T^{\text{miss}}$$

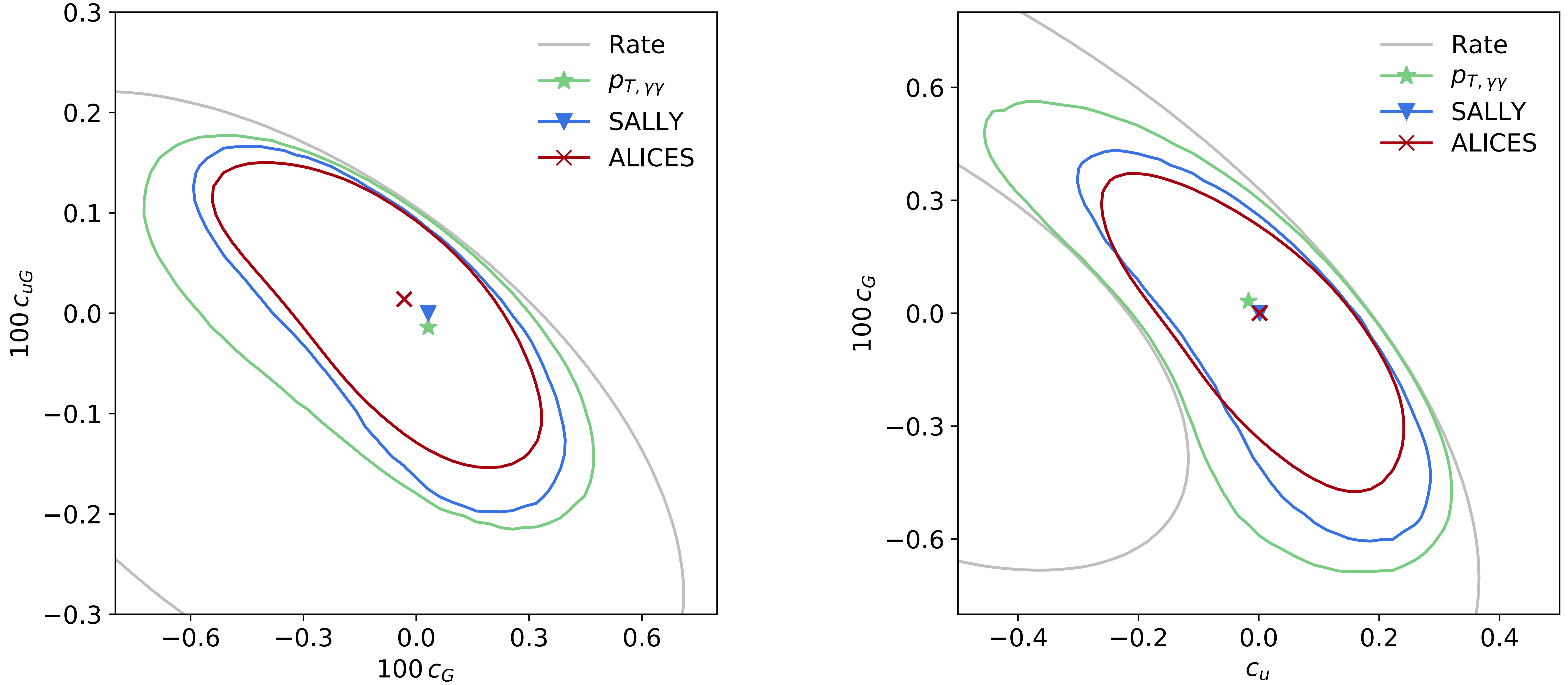
with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

$$\mathcal{O}_u = -\frac{1}{v^2}(H^\dagger H)(H^\dagger \bar{Q}_L)u_R\,,\ \mathcal{O}_G = \frac{g_s^2}{m_W^2}(H^\dagger H)G^a_{\mu\nu}G_a^{\mu\nu}\,,$$

$$\mathcal{O}_{uG} = -\frac{4g_s}{m_W^2}y_u(H^\dagger \bar{Q}_L)\gamma^{\mu\nu}T_a u_R G^a_{\mu\nu}$$

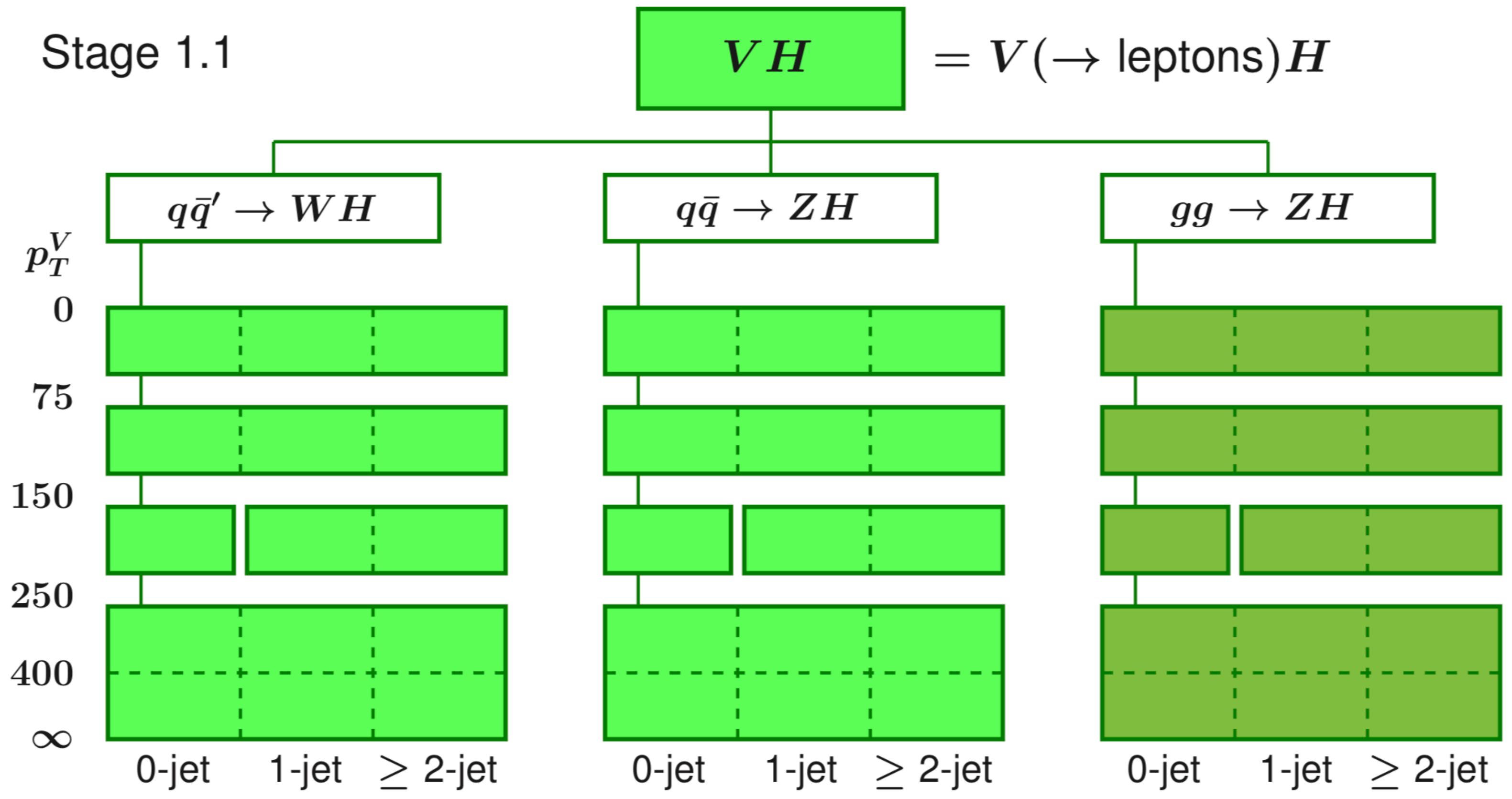
- New inference techniques improve expected HL-LHC limits compared to histogram baseline:



Benchmarking STXS in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]



- Let's check! How much information on

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$

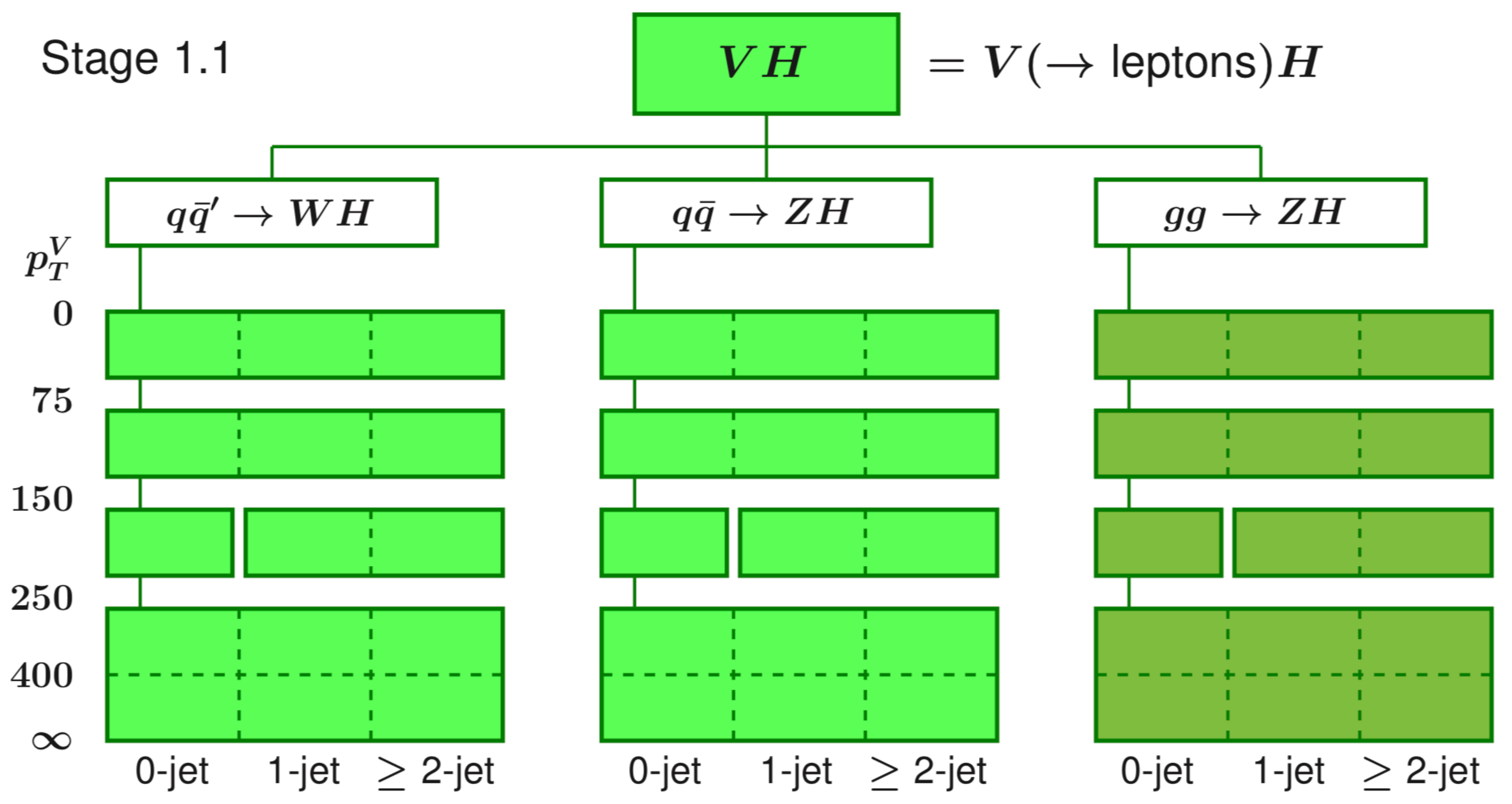
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

can we extract from $pp \to WH \to \ell\nu\ b\bar{b}$?

Benchmarking STXS in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]

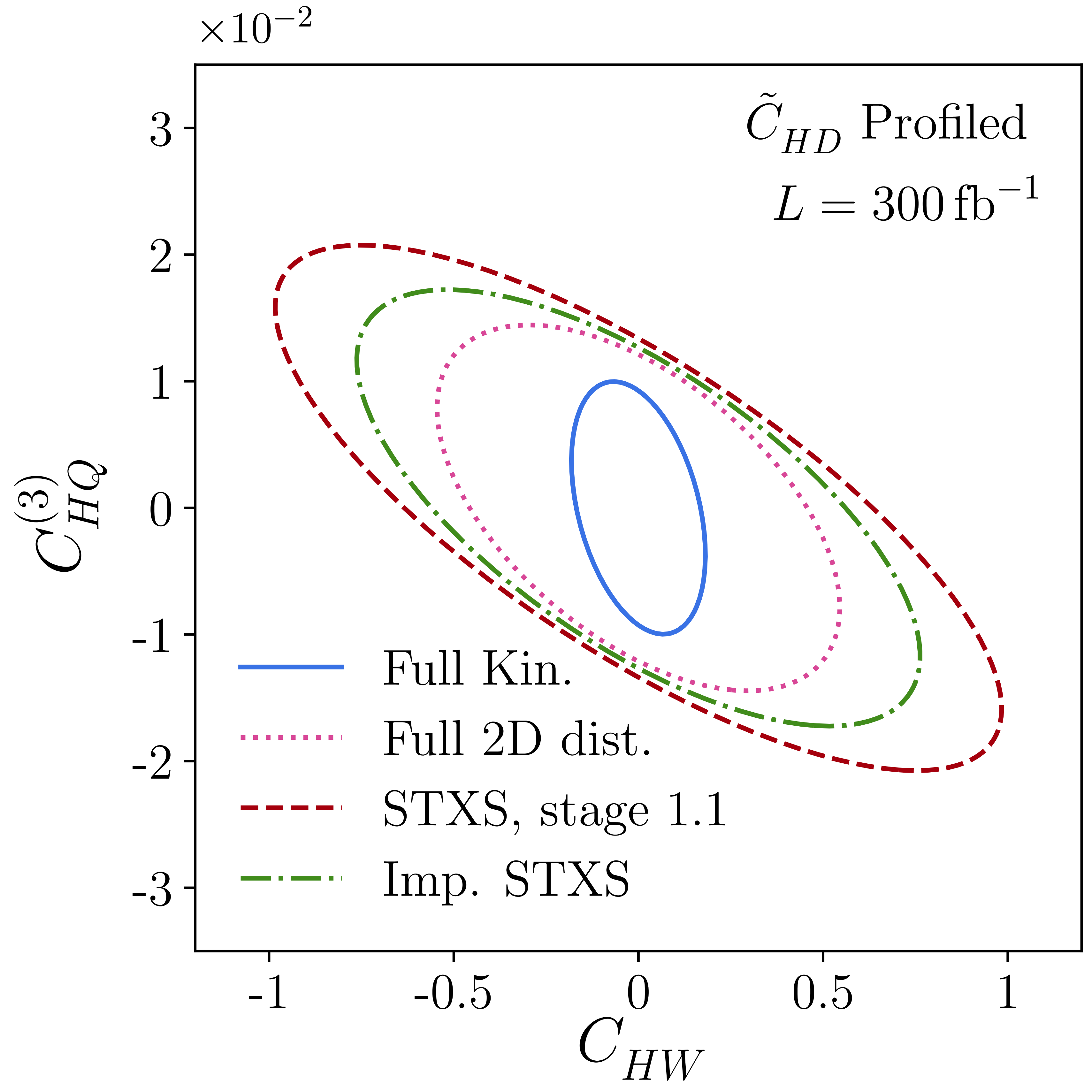


- Let's check! How much information on

$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$
$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i \overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$?

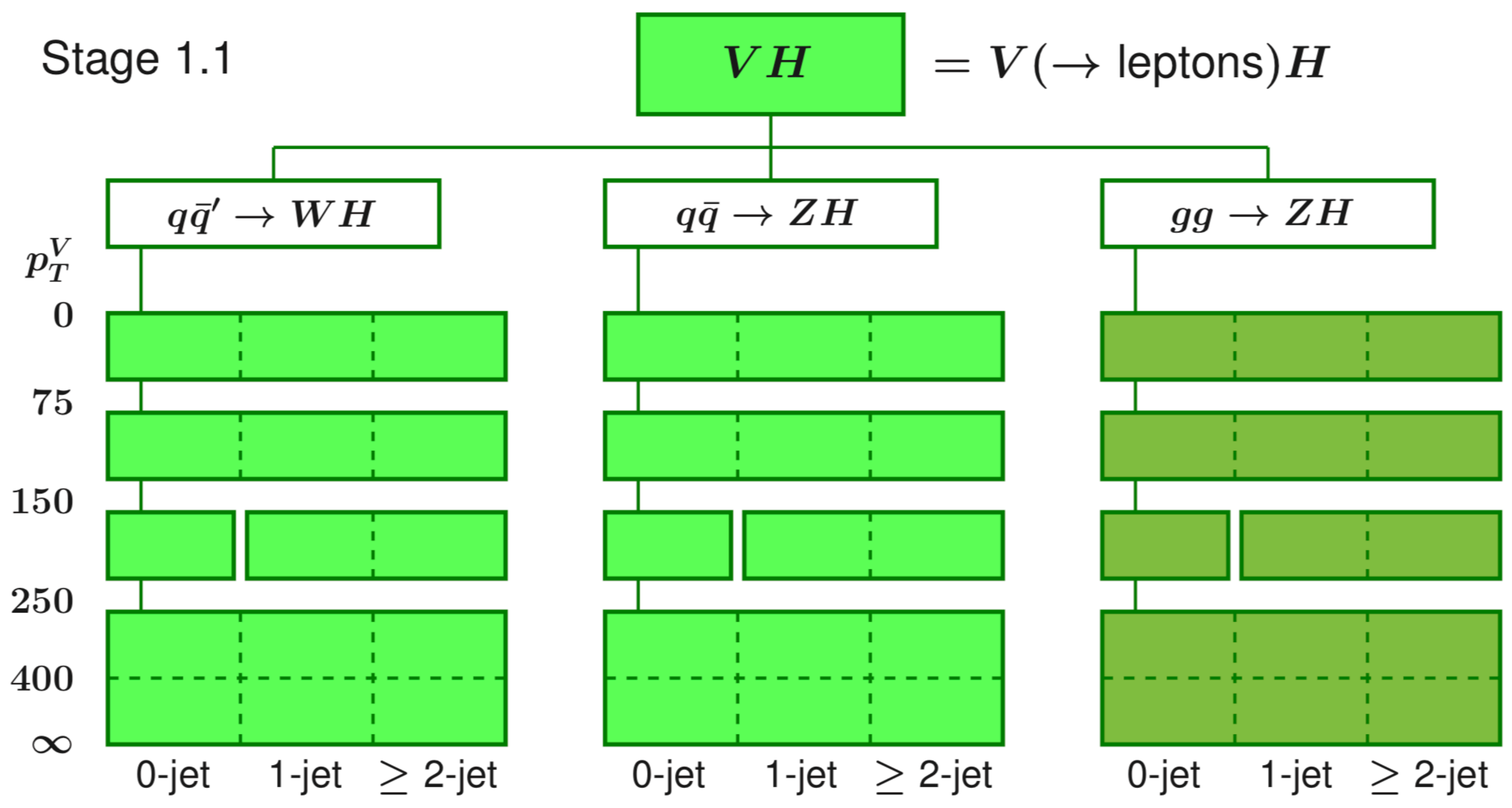
- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



Benchmarking STXS in WH

[JB, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

- Simplified Template Cross-Sections (STXS) define observable bins that are supposed to capture as much information on NP as possible
  [N. Berger et al. 1906.02754; HXSWG YR4]



- Let's check! How much information on

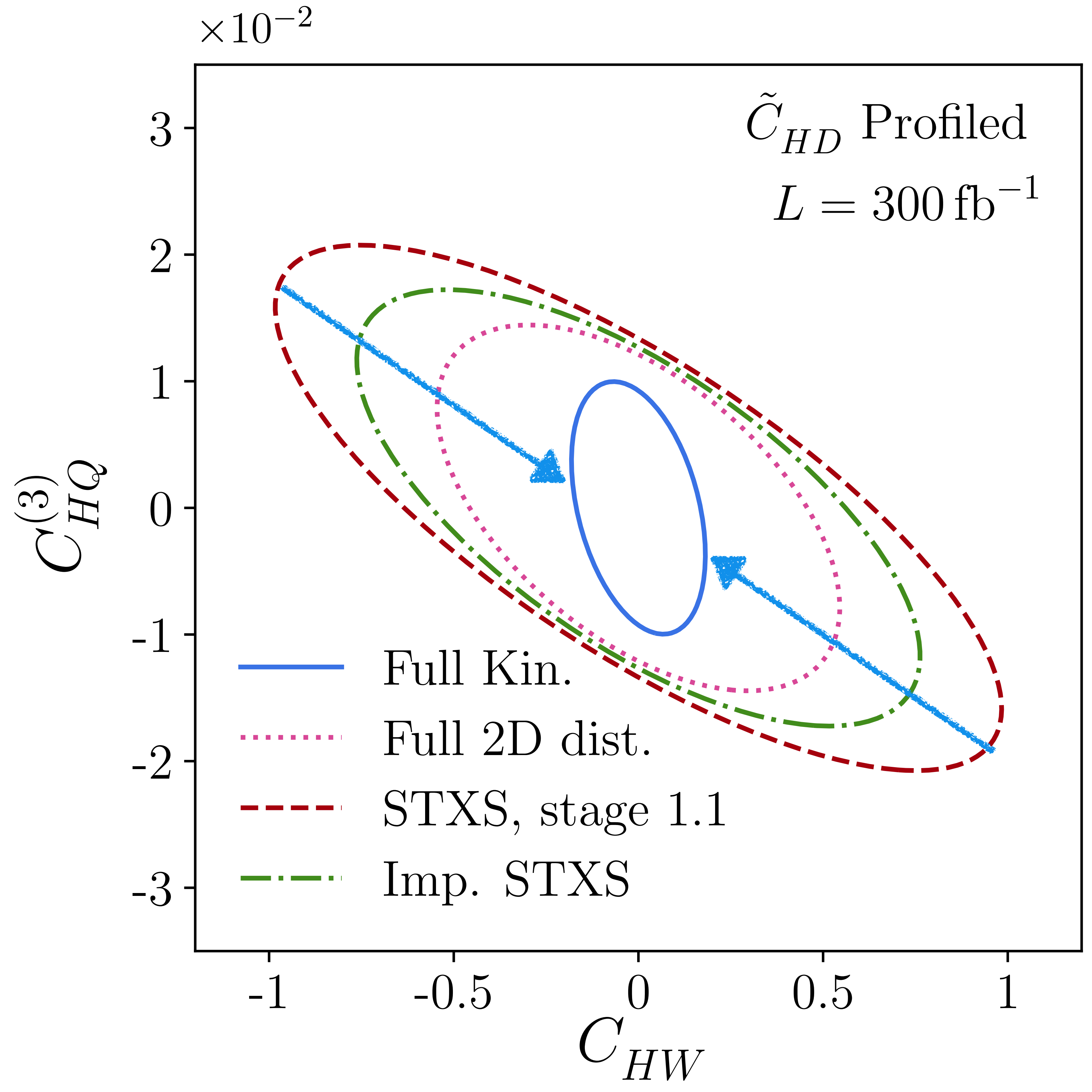
$$\tilde{\mathcal{O}}_{HD} = \mathcal{O}_{H\Box} - \frac{\mathcal{O}_{HD}}{4} = (\phi^\dagger\phi)\Box(\phi^\dagger\phi) - \frac{1}{4}(\phi^\dagger D^\mu\phi)^*(\phi^\dagger D_\mu\phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger\phi W^a_{\mu\nu}W^{\mu\nu a}$$

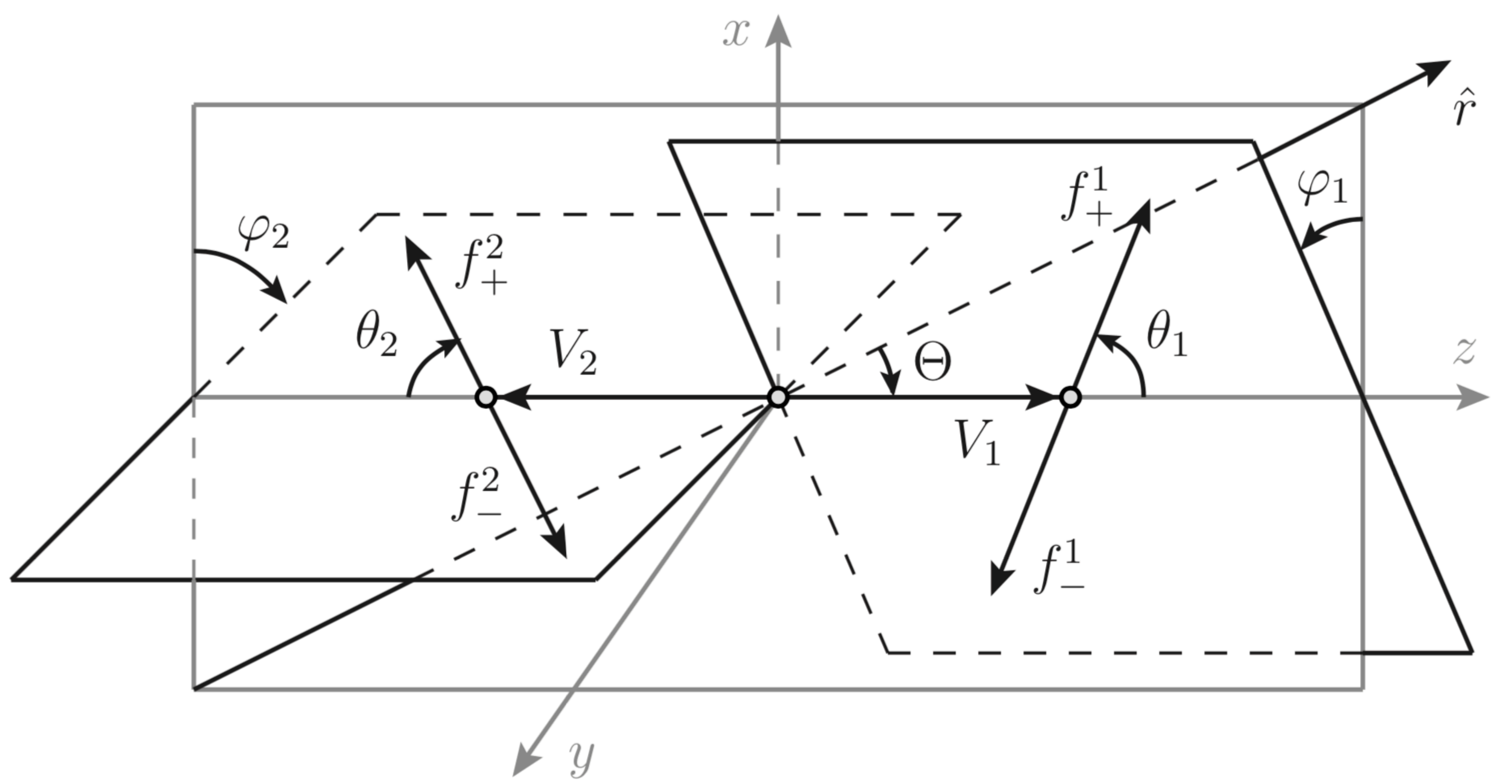
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i \overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

can we extract from $pp \to WH \to \ell\nu\, b\bar{b}$?

- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



Diboson production



- In inclusive observables, the interference between SM and new physics amplitudes vanishes

  ⇒ Reduced sensitivity to new physics

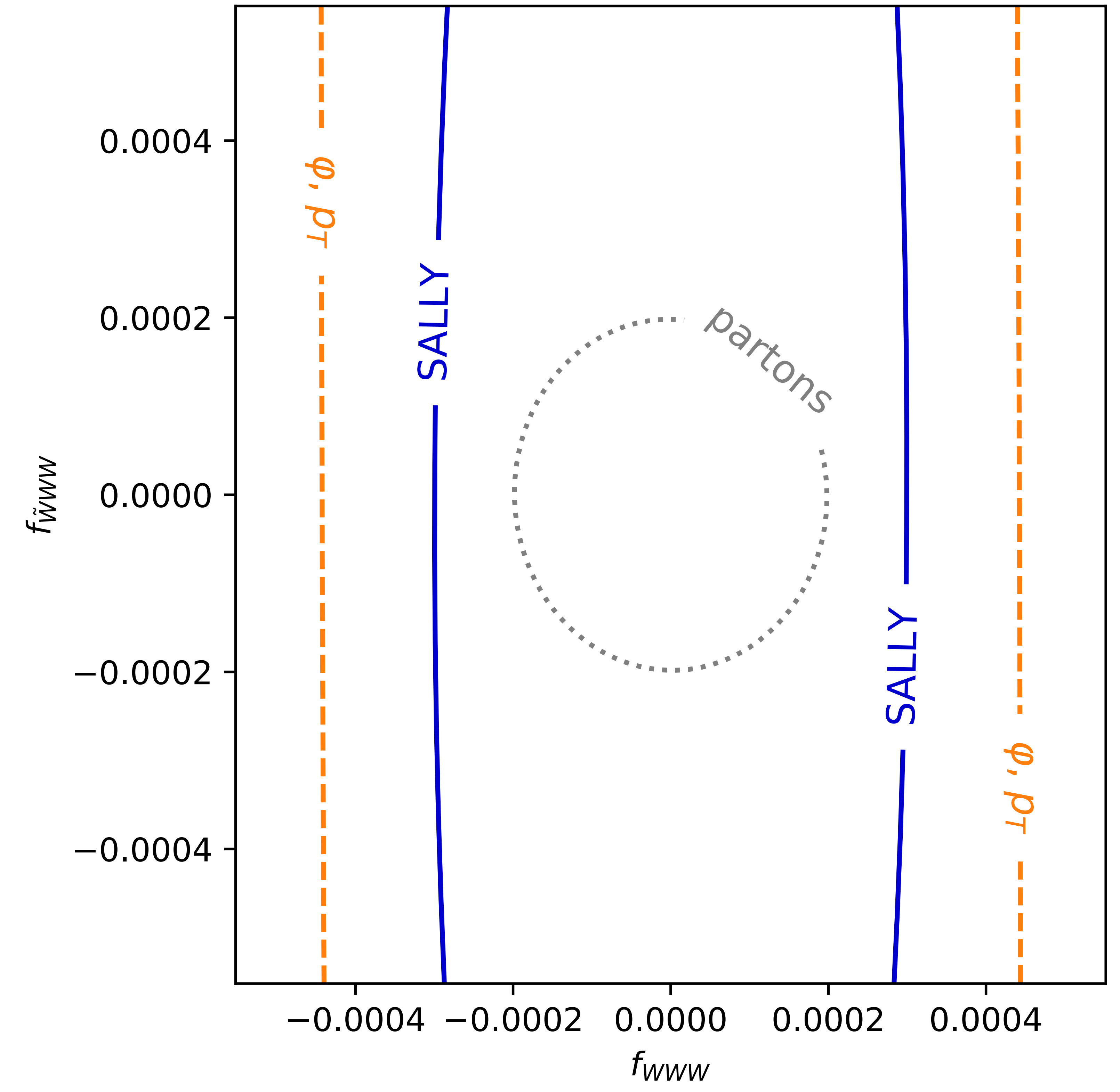
- "Diboson interference resurrection": an angular variable φ can be constructed to be sensitive to this interference
  [G. Panico, F. Riva, A. Wulzer 1708.07823;
  A. Azatov, D. Barducci, E. Venturini 1901.04821]

- We test the ML approach in EFT measurements in $W\gamma \to \ell\nu\ \gamma$
  [JB, K. Cranmer, M. Farina, F. Kling, D. Pappadopulo, J. Ruderman in progress]

  New: WZ→ ℓℓ ℓν by Chen, Glioti, Panico, Wulzer arXiv:2007.10356

- Preliminary results: we can extract more information when we analyze events with SALLY than with histograms of φ and standard observables

Very preliminary results



Conclusion

Likelihood fits in the data space are the gold standard for statistical inference

- RECAST and likelihood publishing are technical solutions that address model dependence and the theory-experiment interface
- STXS a good step, but more differential information can lead to large gain in sensitivity

Properties we want

- Ability to be fully differential
- Exploit highest fidelity simulation (QCD, detector simulation) without approximations that introduce additional systematic errors
- Clear statistical motivation and compatibility with traditional combined analyses
- Scalability in terms of channels and parameters

The approach I presented (implemented in MadMiner) achieves these goals

References

Opinionated review
K. Cranmer, JB, G. Louppe:
"The frontier of simulation-based inference"
[1911.01429]

Do It Yourself (for LHC physics)
JB, F. Kling, I. Espejo, K. Cranmer:
"MadMiner: Machine learning—based inference for particle physics"
[CSBS, 1907.10621, https://github.com/diana-hep/madminer]

LHC HXSWG YR4 STXS
JB, S. Dawson, S. Homiller, F. Kling, T. Plehn:
"Benchmarking simplified template cross sections in WH production"
[JHEP, 1908.06980]

Use in Astro: Strong lensing
JB, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer
"Mining for Dark Matter Substructure: Inferring subhalo population properties from strong lenses with machine learning"
[ApJ, 1909.02005]

Original works

JB, K. Cranmer, G. Louppe, J. Pavez:
"A guide to constraining Effective Field Theories with machine learning"
[PRD, 1805.00020]

JB, G. Louppe, J. Pavez, K. Cranmer:
"Mining gold from implicit models to improve likelihood-free inference"
[PNAS, 1805.12244]

Follow-up with incremental improvements
M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez:
"Likelihood-free inference with an improved cross-entropy estimator"
[NeurIPS workshop, 1808.00973]

An incomplete wrap-up of simulation-based inference methods

Method	Approximations	Upfront cost	Eval
Summary statistics:			
Likelihood for summary stats (standard histograms)	Reduction to summary stats	Fast	Fast
Approximate Bayesian Computation	Reduction to summary stats	Depends	Depends
Matrix elements:			
Matrix Element Method	Transfer fns	Fast	Slow
Optimal Observables	Transfer fns, optimal only locally	Fast	Slow
Neural networks:			
Neural likelihood	NN	Needs many samples	Fast
Neural posterior	NN	Needs many samples	Fast
Neural likelihood ratio	NN	Needs many samples	Fast
Neural networks + matrix elements:			
Neural likelihood (ratio) + gold mining (RASCAL etc)	NN	Needs less samples	Fast
Neural optimal observables (SALLY)	NN, optimal only locally	Needs less samples	Fast

Mining gold: A family of new inference techniques

Method	Simulate	Extract $r(x,z)$	Extract $t(x,z)$	NN estimates	Asympt. exact	Generative
ROLR	$\theta_0 \sim \pi(\theta), \theta_1$	✓		$\hat{r}(x\|\theta_0, \theta_1)$	✓	
CASCAL	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
ALICE	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
RASCAL	$\theta_0 \sim \pi(\theta), \theta_1$	✓	✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
ALICES	$\theta_0 \sim \pi(\theta), \theta_1$	✓	✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
SCANDAL	$\theta \sim \pi(\theta)$		✓	$\hat{p}(x\|\theta)$	✓	✓
SALLY	θ_{ref}		✓	$\hat{t}(x\|\theta_{\text{ref}})$	in local approx.	
SALLINO	θ_{ref}		✓	$\hat{t}(x\|\theta_{\text{ref}})$	in local approx.	

Mining gold: A family of new inference techniques

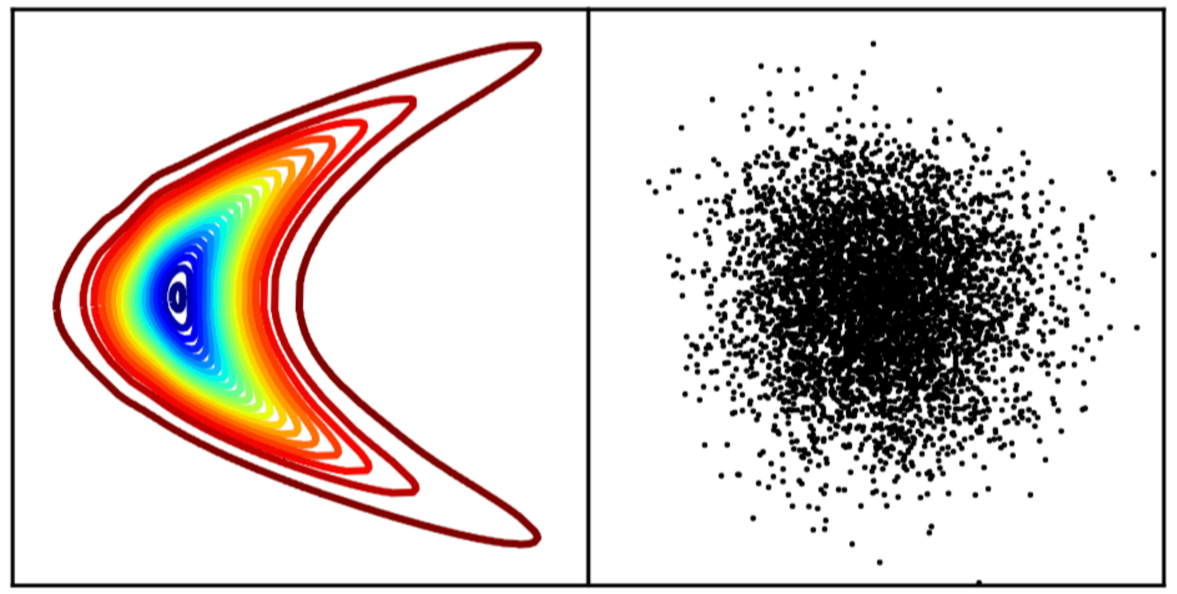
Method	Simulate	Extract $r(x,z)$	Extract $t(x,z)$	NN estimates	Asympt. exact	Generative
ROLR	$\theta_0 \sim \pi(\theta), \theta_1$	✓		$\hat{r}(x\|\theta_0, \theta_1)$	✓	
CASCAL	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
ALICE	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
RASCAL	$\theta_0 \sim \pi(\theta), \theta_1$	✓	✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
ALICES	$\theta_0 \sim \pi(\theta), \theta_1$	✓	✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
SCANDAL	$\theta \sim \pi(\theta)$		✓	$\hat{p}(x\|\theta)$	✓	✓
SALLY	θ_{ref}		✓	$\hat{t}(x\|\theta_{\text{ref}})$	in local approx.	
SALLINO	θ_{ref}		✓	$\hat{t}(x\|\theta_{\text{ref}})$	in local approx.	

Performance gains with cross-entropy-based loss (ALICE, ALICES)

[M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez 1808.00973]

Mining gold: A family of new inference techniques

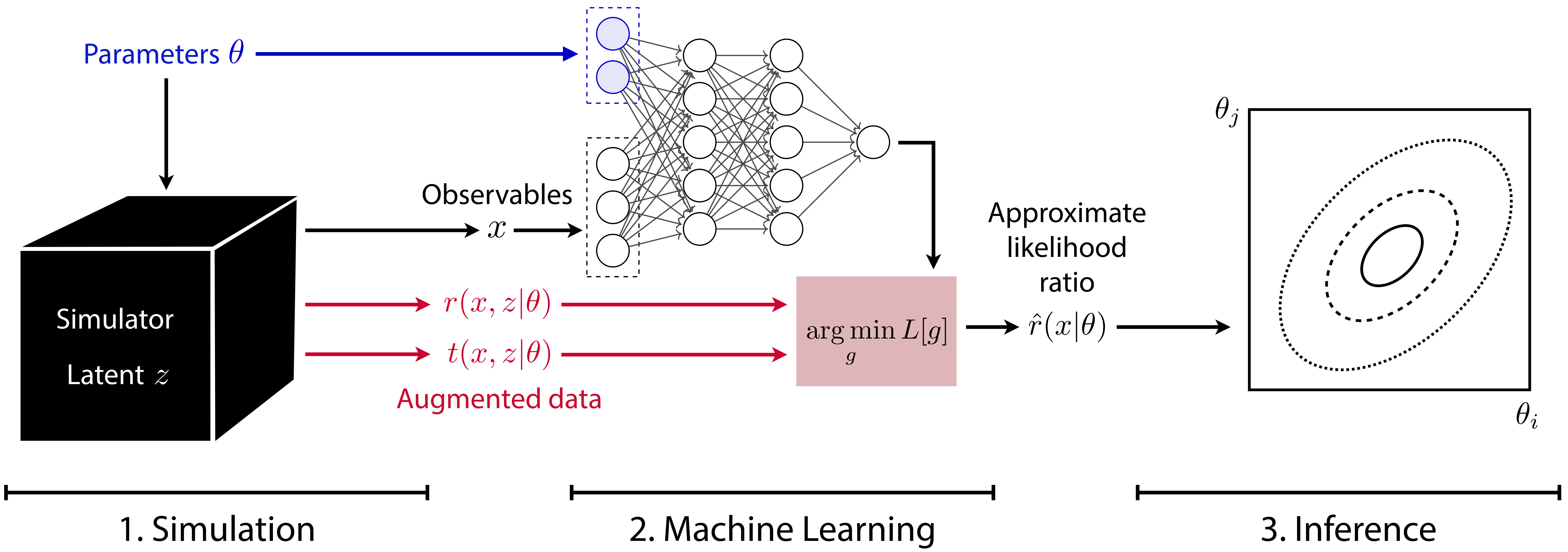
Method	Simulate	Extract $r(x,z)$	Extract $t(x,z)$	NN estimates	Asympt. exact	Generative
ROLR	$\theta_0 \sim \pi(\theta), \theta_1$	✓		$\hat{r}(x\|\theta_0, \theta_1)$	✓	
CASCAL	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
ALICE	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
RASCAL	$\theta_0 \sim \pi(\theta), \theta_1$	✓	✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
ALICES	$\theta_0 \sim \pi(\theta), \theta_1$	✓	✓	$\hat{r}(x\|\theta_0, \theta_1)$	✓	
SCANDAL	$\theta \sim \pi(\theta)$		✓	$\hat{p}(x\|\theta)$	✓	✓
SALLY	θ_{ref}		✓	$\hat{t}(x\|\theta_{\text{ref}})$	in local approx.	
SALLINO	θ_{ref}		✓	$\hat{t}(x\|\theta_{\text{ref}})$	in local approx.	



Combination with state-of-the-art conditional neural density estimators, e.g. normalizing flows

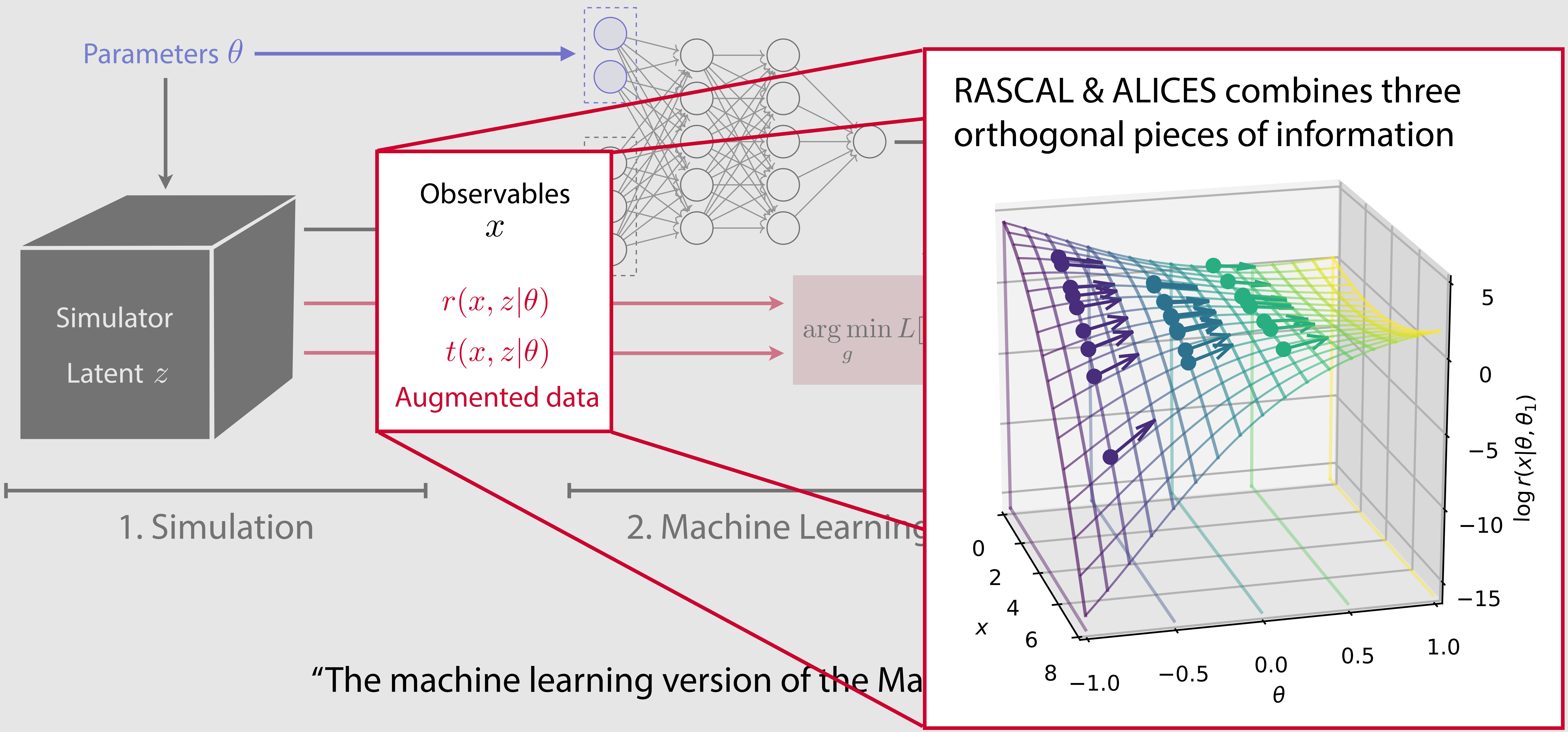
[everything by G. Papamakarios: G. Papamakarios, T. Pavlakou, I. Murray 1705.07057; G. Papamakarios, D. Sterratt, I. Murray 1805.07226; …]

Putting the pieces together: RASCAL & ALICES



"The machine learning version of the Matrix Element Method"

Putting the pieces together: RASCAL & ALICES

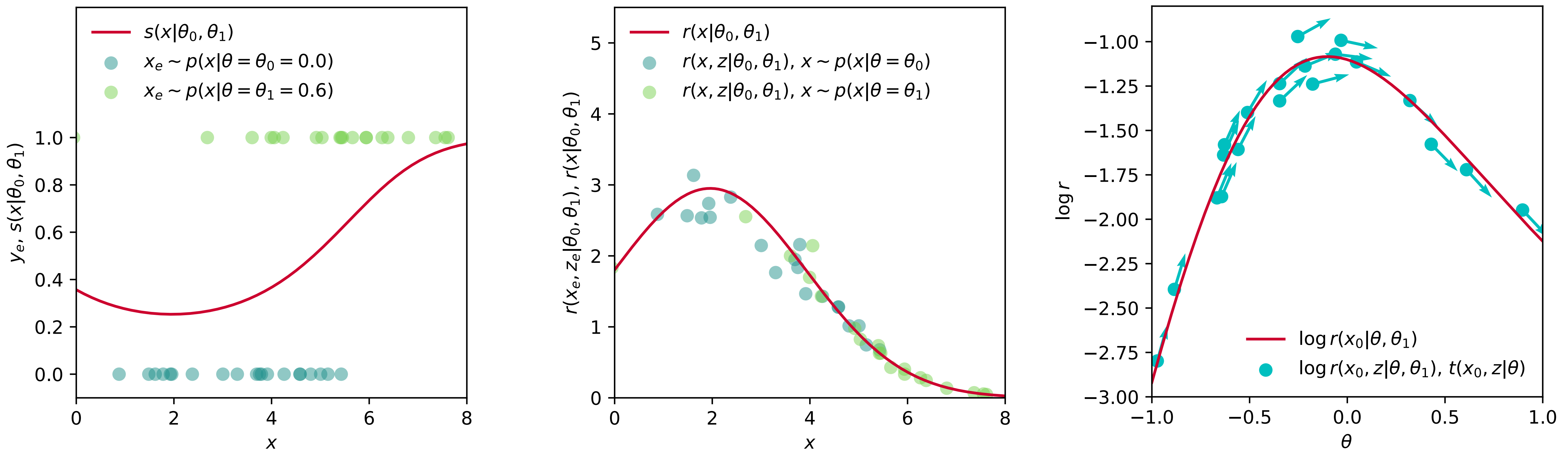


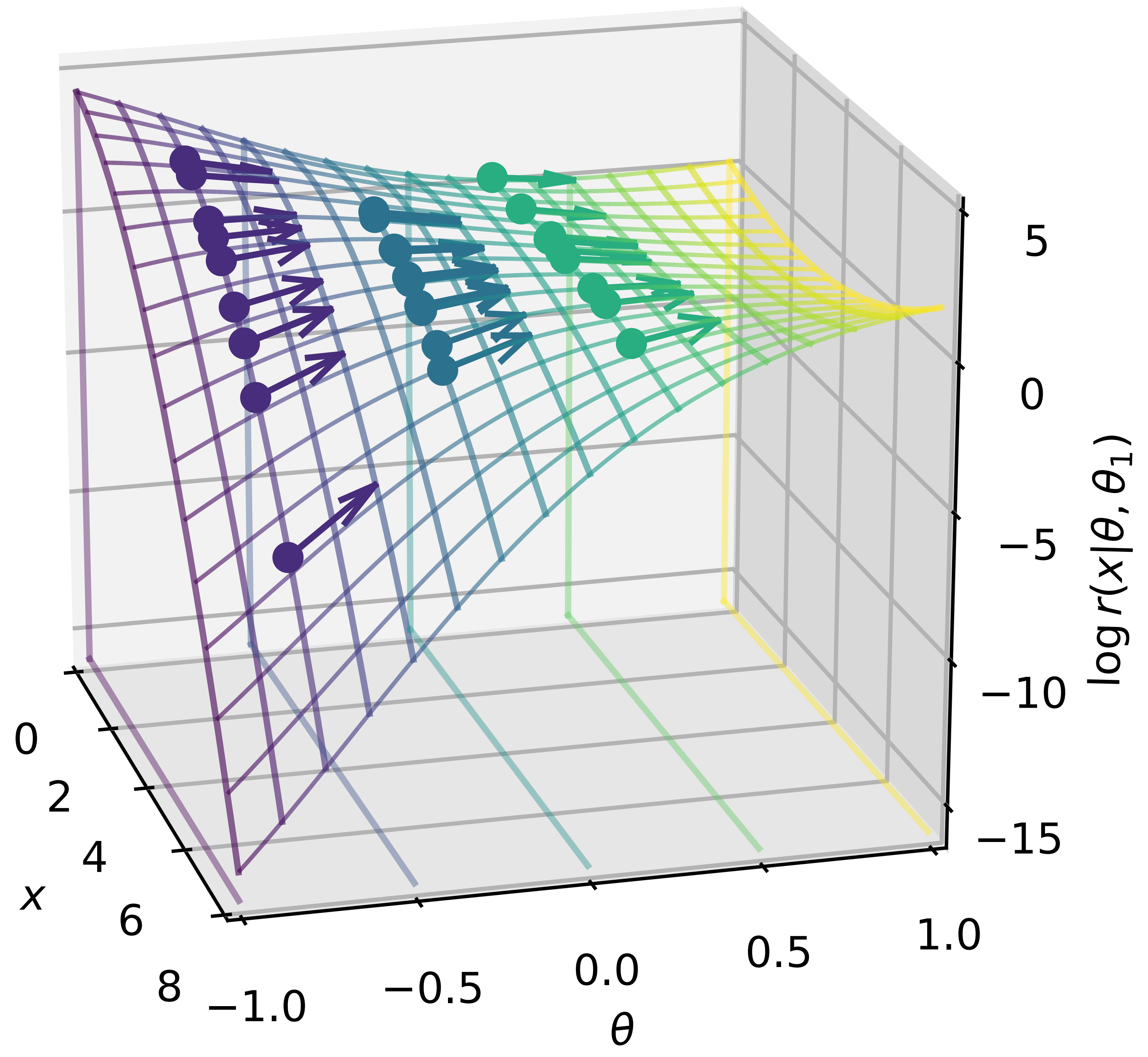
"The machine learning version of the Ma

Gold mining: augmenting the training data

The augmented training data converts supervised classification into supervised regression with lower variance

- improvement in training efficiency

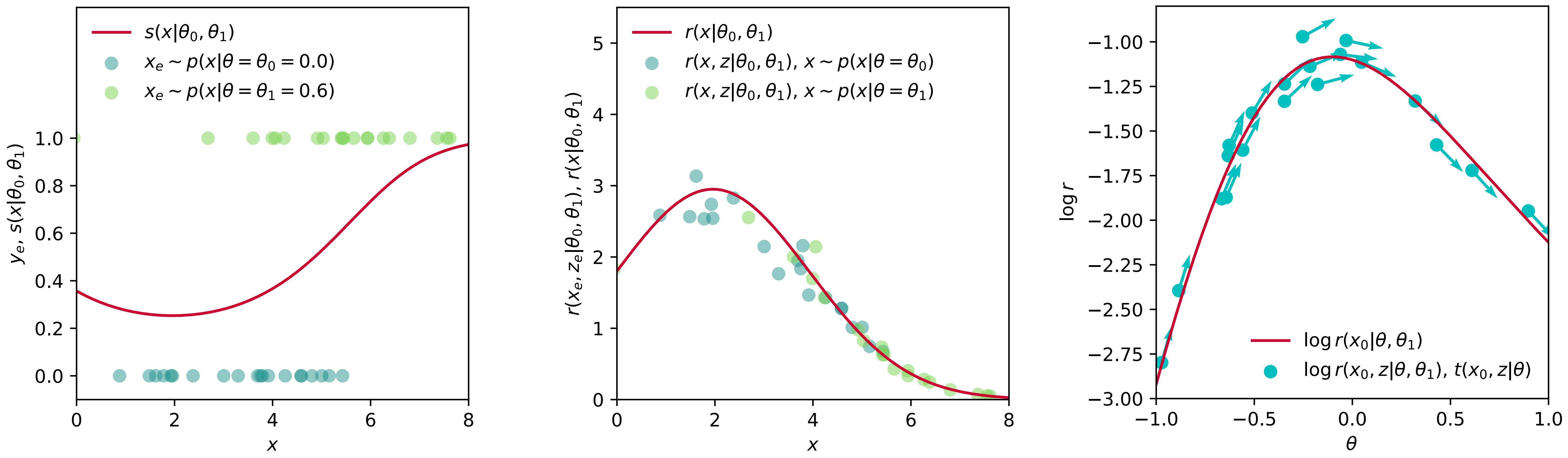


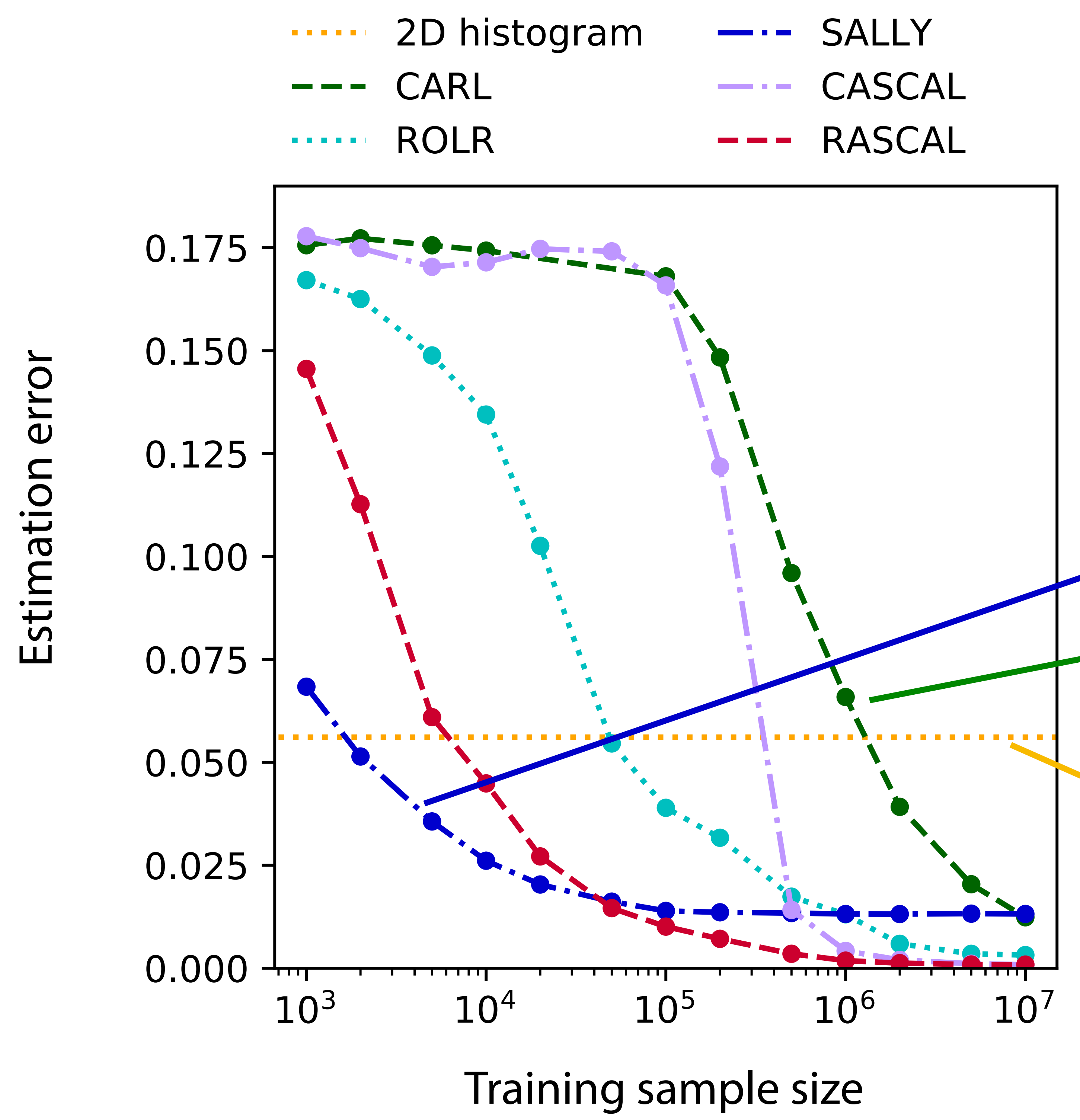


Gold mining: augmenting the training data

The augmented training data converts supervised classification into supervised regression with lower variance

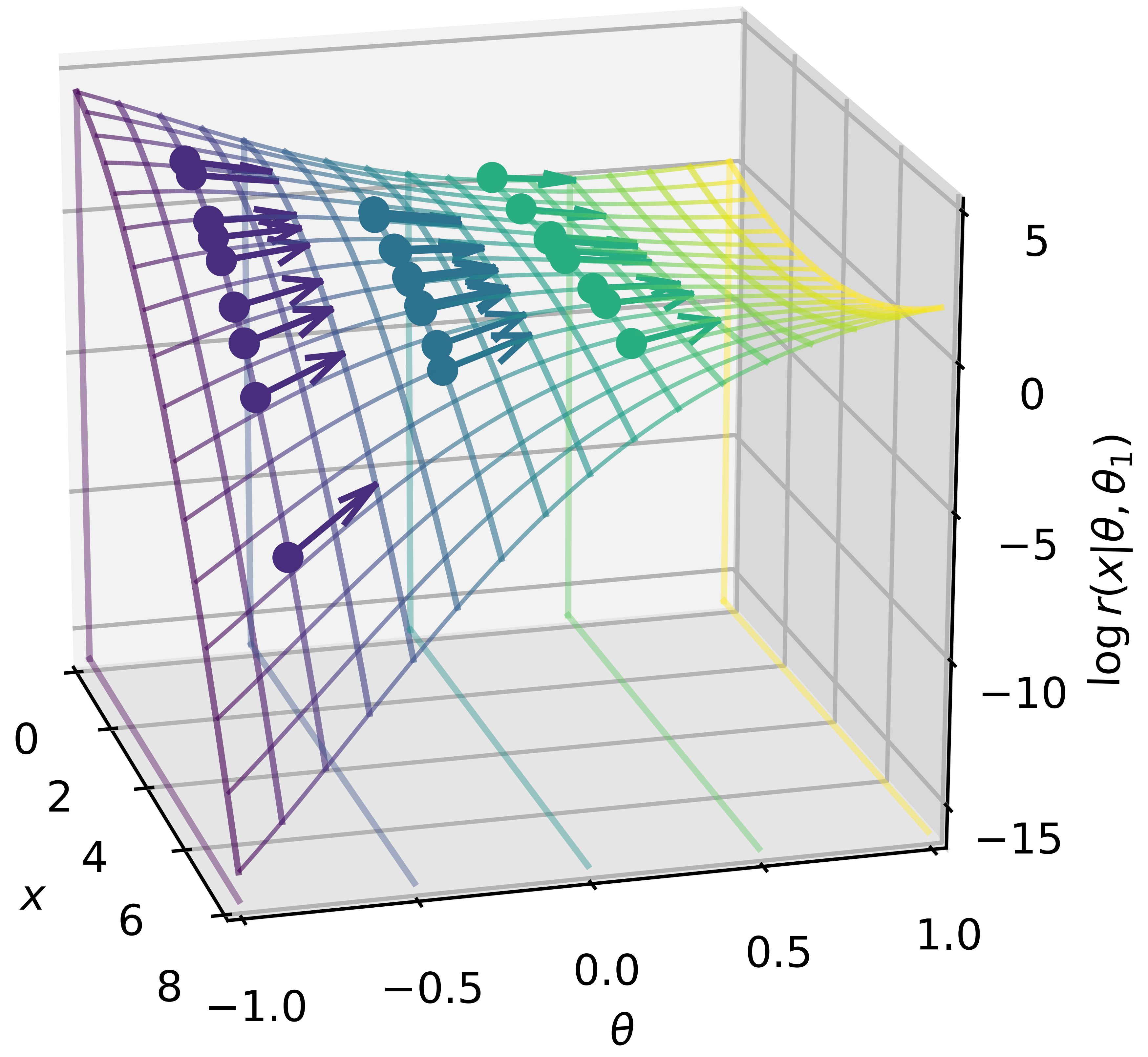
- improvement in training efficiency





New techniques require less data than without augmented data

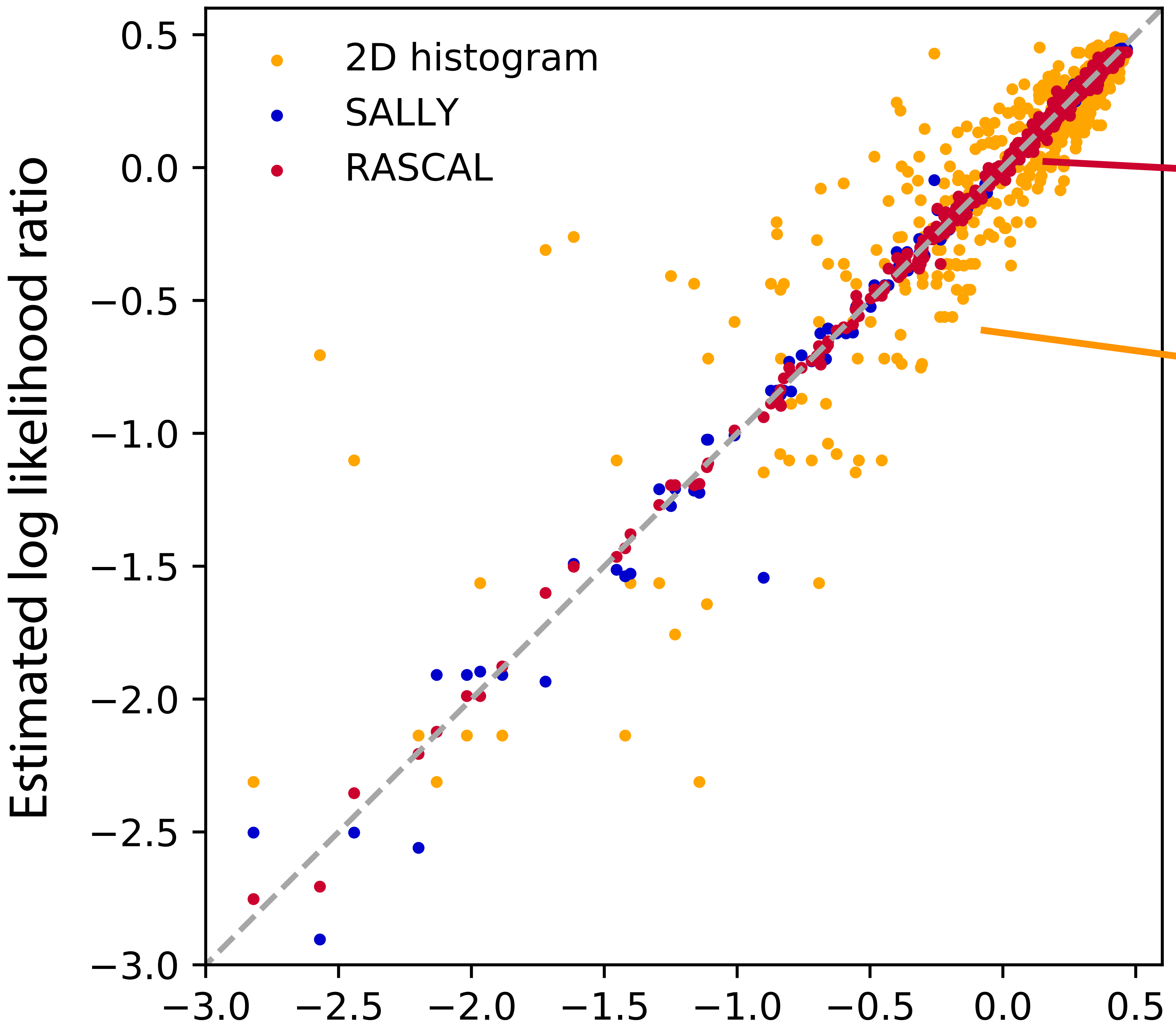
Traditional Approach no NN



Brehmer, Louppe, Pavez, KC, PNAS (2019), arXiv:1805.12244

More precise likelihood ratio estimates with less training data

Each point = 1 event

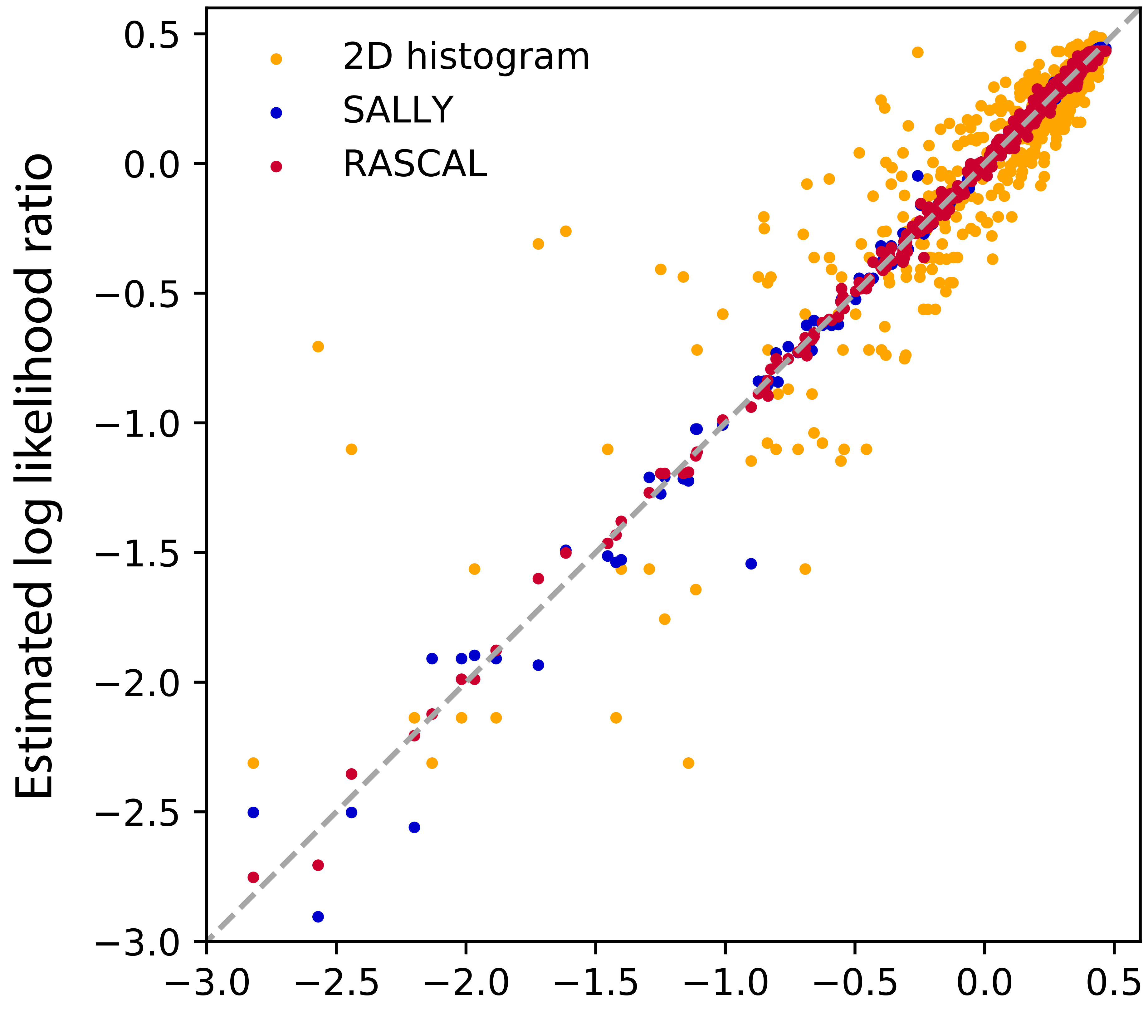


True log likelihood ratio
(usually don't have this)

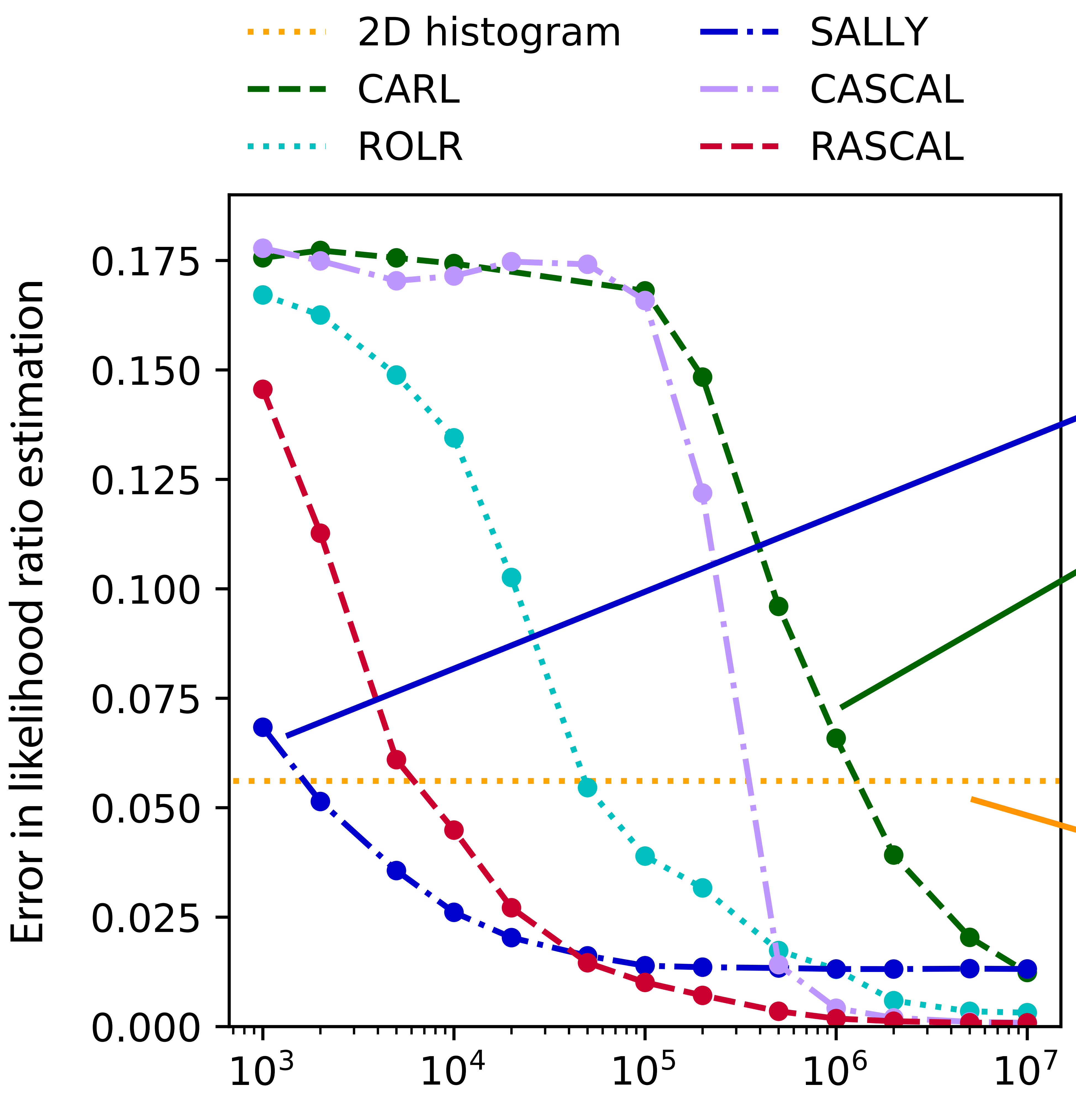
RASCAL
gives more precise likelihood ratio estimates than the
2D histogram

More precise likelihood ratio estimates with less training data

Each point = 1 event



True log likelihood ratio
(usually don't have this)



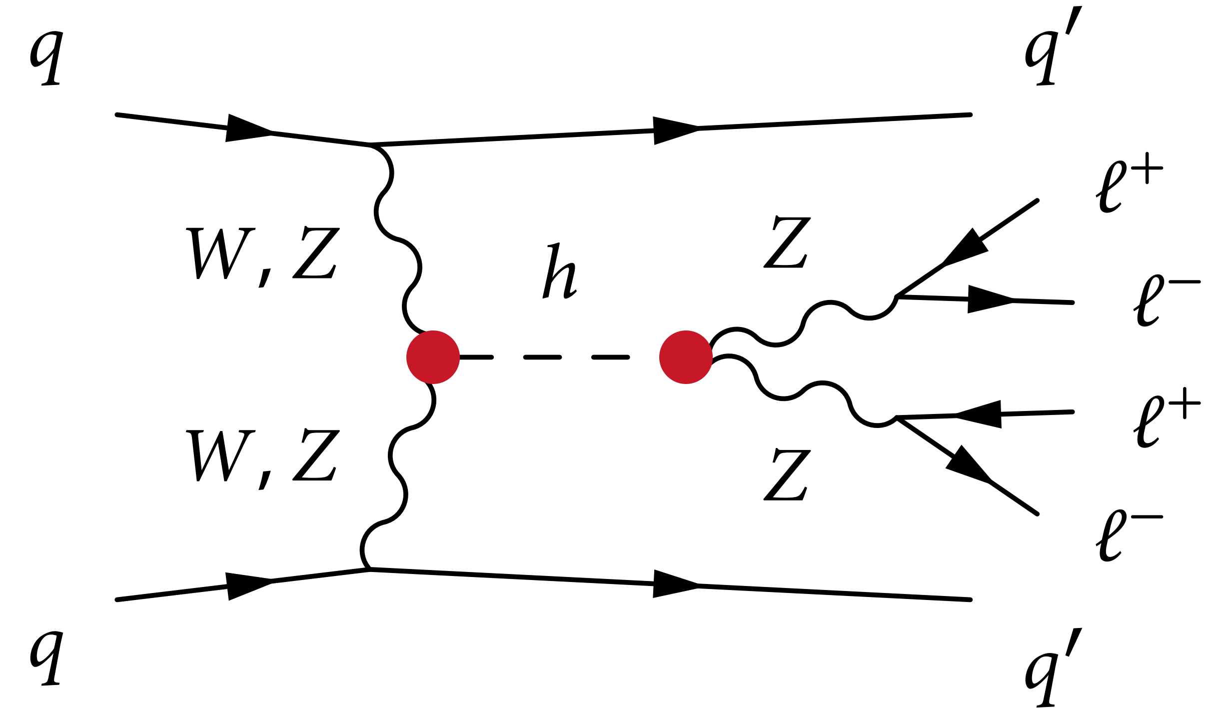
SALLY and RASCAL require less data than generic ML method

[CARL, K. Cranmer, G. Louppe, J. Pavez 1506.02169]

Baseline: 2D histogram (based on large training sample)

Challenge for EFT

Let θ denote the coefficients of higher dimensional operators in the Lagrangian, x be high-dimensional data associated to an event, and $p(x \mid \theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma}{d\theta}$ be the distribution for the data



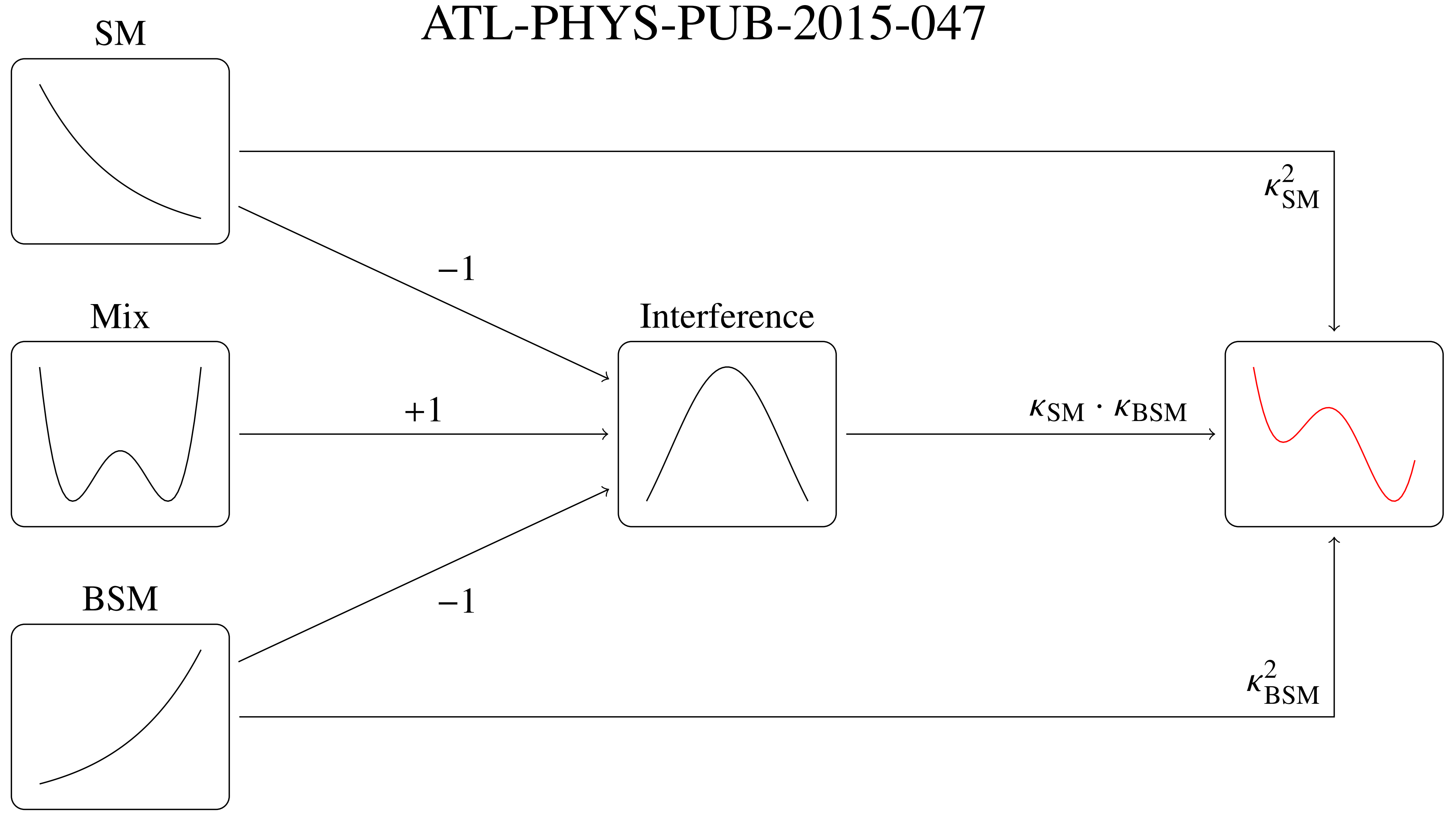
- we want to compare any two points in EFT parameter space
- evaluate the **likelihood ratio** $r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$

Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

- It would be very computationally expensive (infeasible) to generate samples for every value of θ and estimate $p(x \mid \theta)$ with histograms. Small changes mean we need a lot of MC events!
- Ideally we could directly estimate the **score** $t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$

EFT Embedded in a vector space

ATL-PHYS-PUB-2015-047



Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

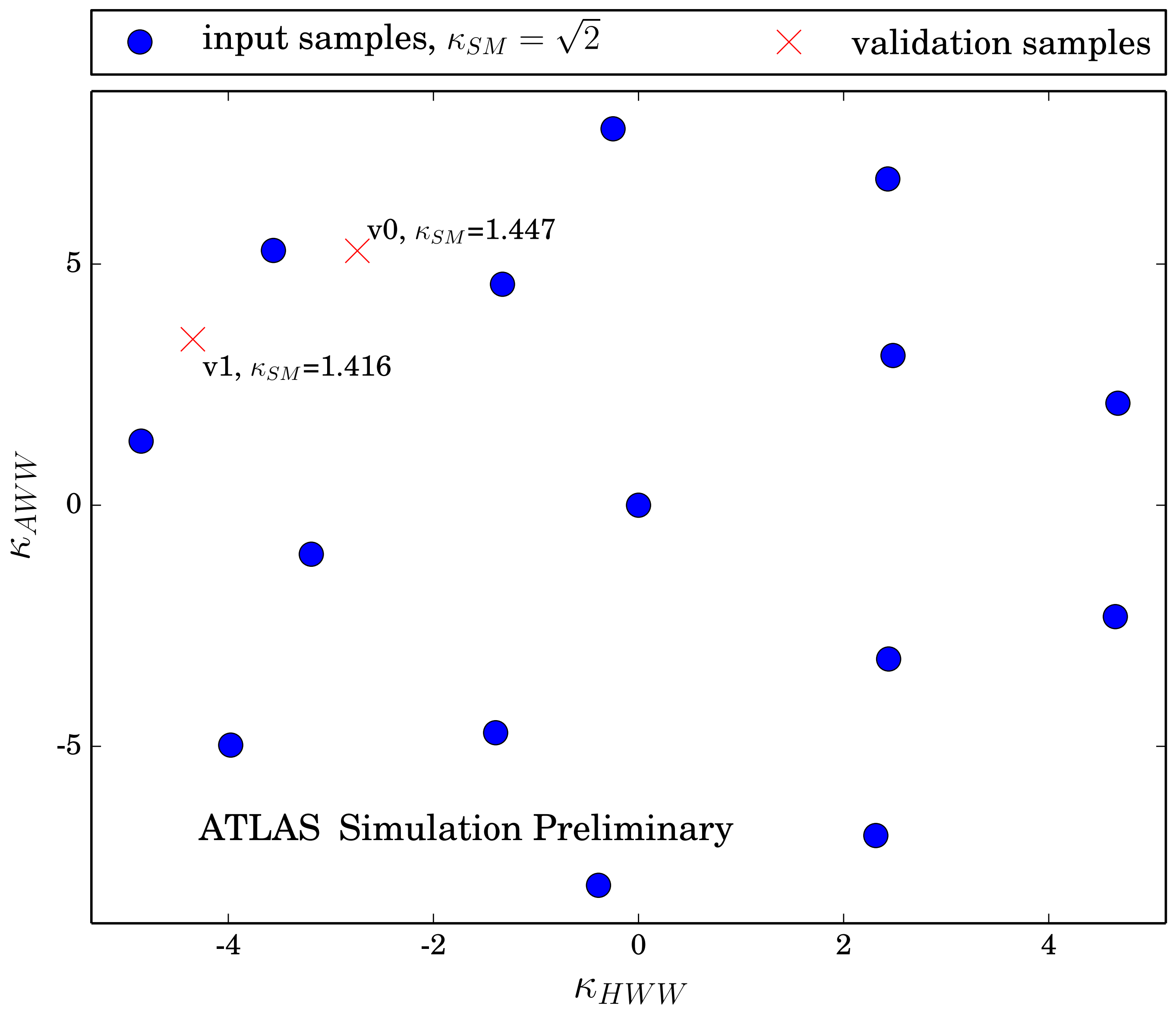
But there is a trick:

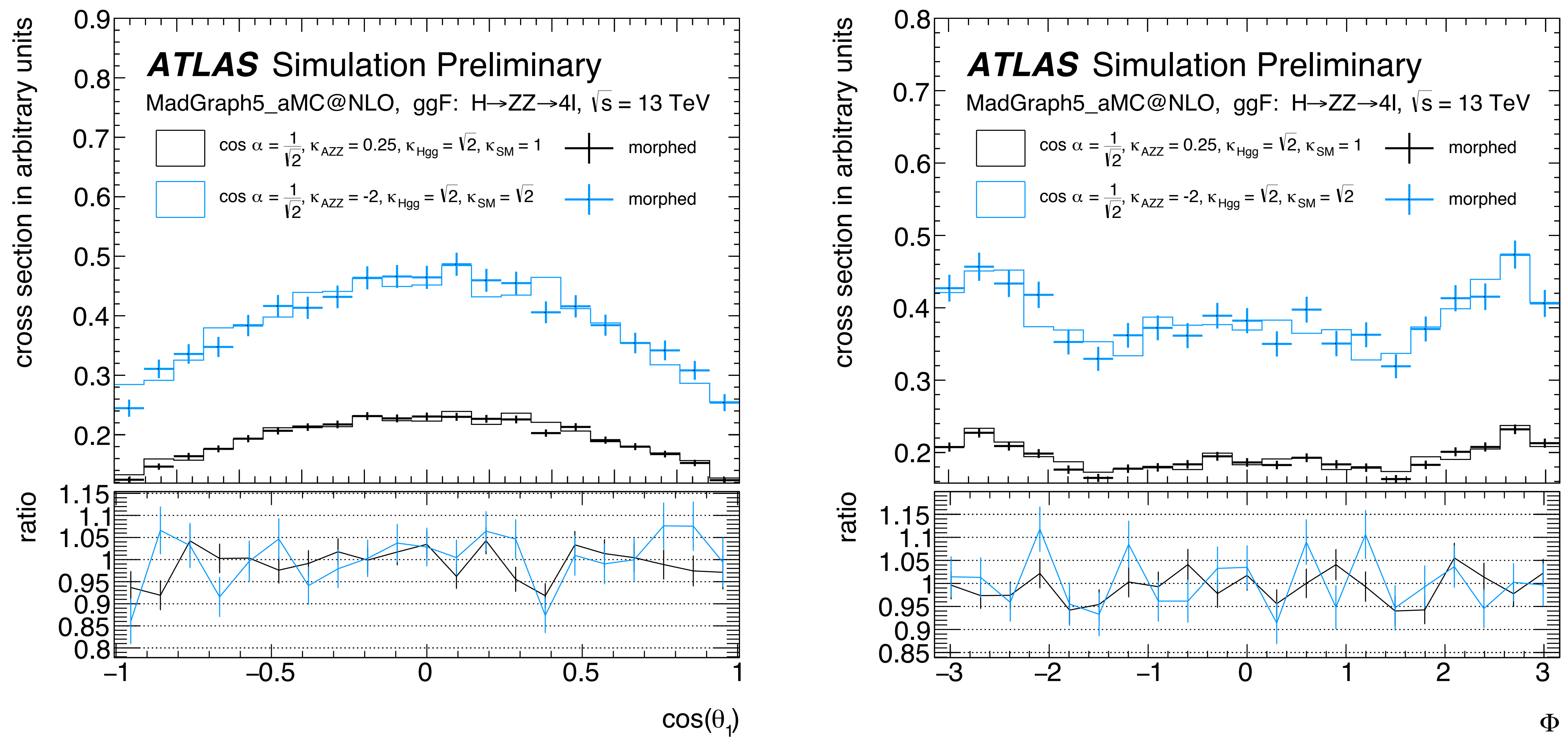
Simple example:

$$|g_1 M_{SM} + g_2 M_{BSM}|^2 = g_1^2 |M_{SM}|^2 + 2 g_1 g_2 Re\left[M_{SM}^* M_{BSM}\right] + g_2^2 |M_{BSM}|^2$$

3-d vector space, distribution for any point in this space is linear mixture of distribution for 3 basis samples!

(real examples need more basis samples)





EFT Decomposition

$$d\sigma \propto \left\| \underbrace{\left(\mathcal{M}^p_{\text{SM}} + \sum_i \frac{f_i}{\Lambda^2}\mathcal{M}^p_i\right)}_{\text{production}} \underbrace{\left(\mathcal{M}^d_{\text{SM}} + \sum_j \frac{f_j}{\Lambda^2}\mathcal{M}^d_j\right)}_{\text{decay}} \right\|^2$$

Express EFT as a mixture:

$$p(x|\theta) = \sum_c w_c(\theta) p_c(x)$$

$w_c(\theta)$ are polynomials

$\nabla_\theta \log p(x|\theta)$ is now possible!

Process	Number of components for n operators					
	$\mathcal{O}(\Lambda^0)$	$\mathcal{O}(\Lambda^{-2})$	$\mathcal{O}(\Lambda^{-4})$	$\mathcal{O}(\Lambda^{-6})$	$\mathcal{O}(\Lambda^{-8})$	Σ
hV / WBF production	1	n	$\frac{n(n+1)}{2}$			$\frac{(n+1)(n+2)}{2}$
$h \to VV$ decay	1	n	$\frac{n(n+1)}{2}$			$\frac{(n+1)(n+2)}{2}$
Production + decay	1	n	$\frac{n(n+1)}{2}$	$\binom{n+2}{3}$	$\binom{n+3}{4}$	$\binom{n+4}{4}$

Table 1: Number of components c as given in Eq. (6) for different processes, sorted by their suppression by the EFT cutoff scale Λ.

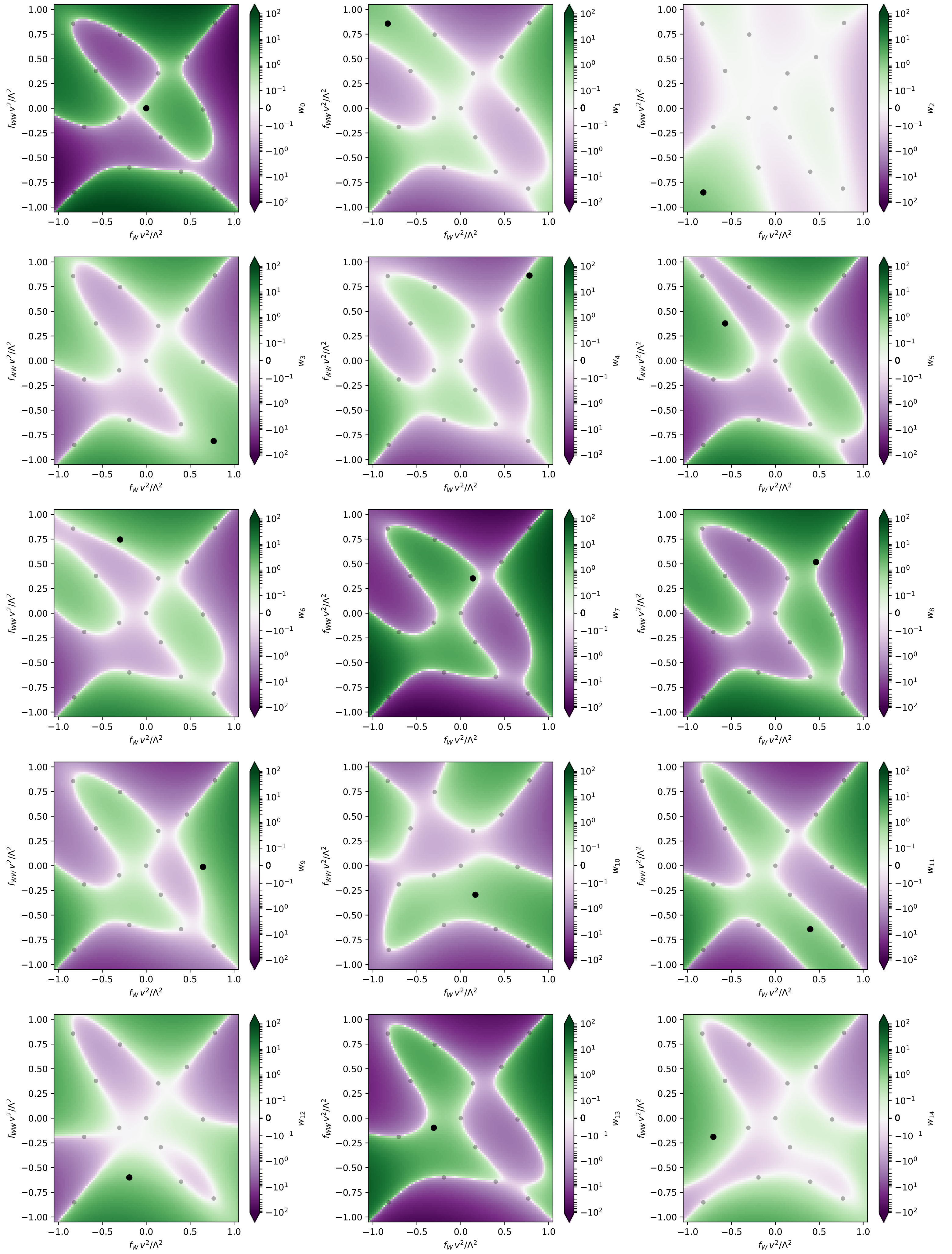
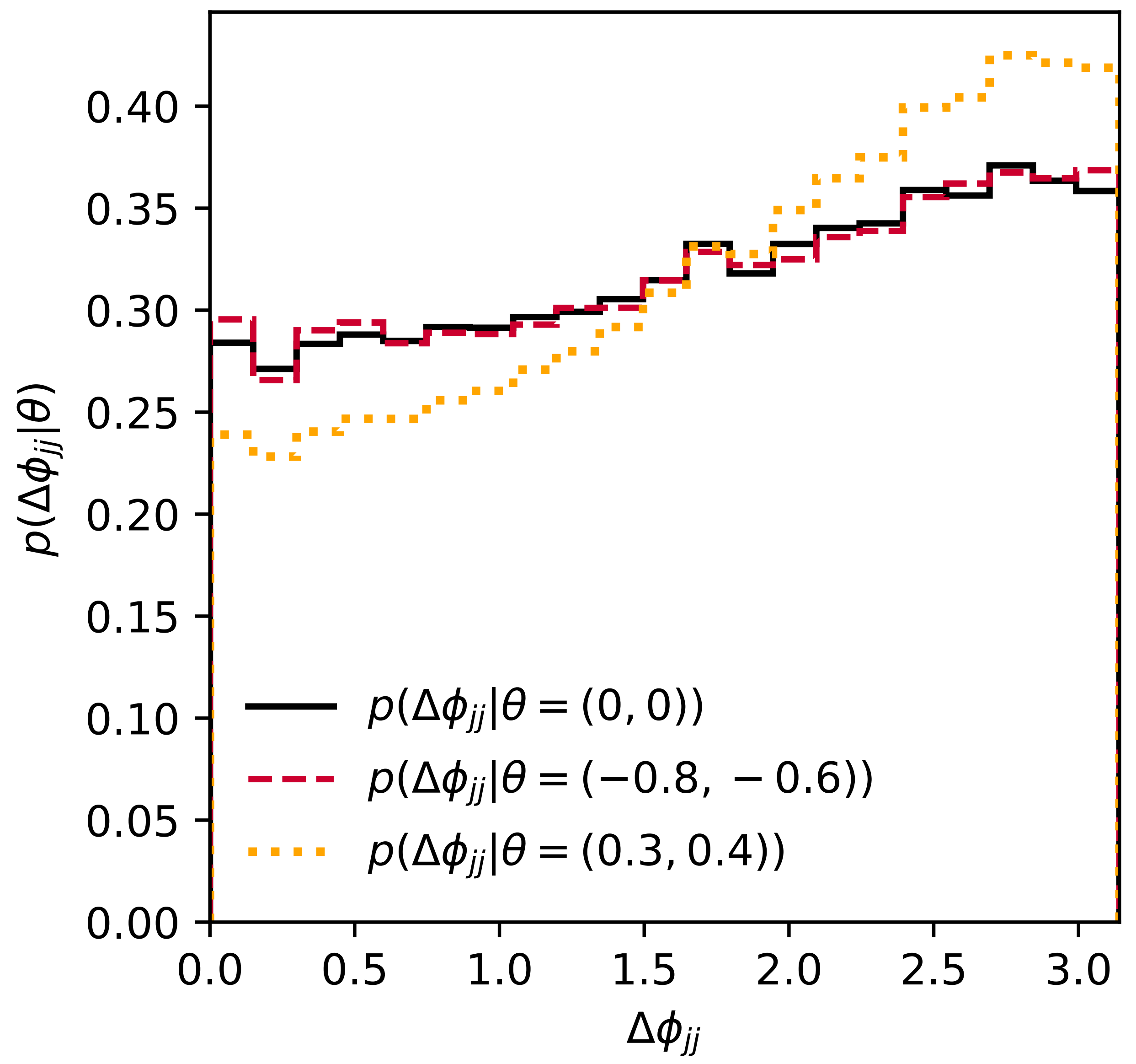


Figure 13: Morphing weights $w_i(\boldsymbol{\theta})$ for basis points distributed over the full relevant parameter space.

For 2 BSM operators affecting VBF Higgs production and decay, we need a 15-D vector space

For 5 BSM operators we need 126-D vector space

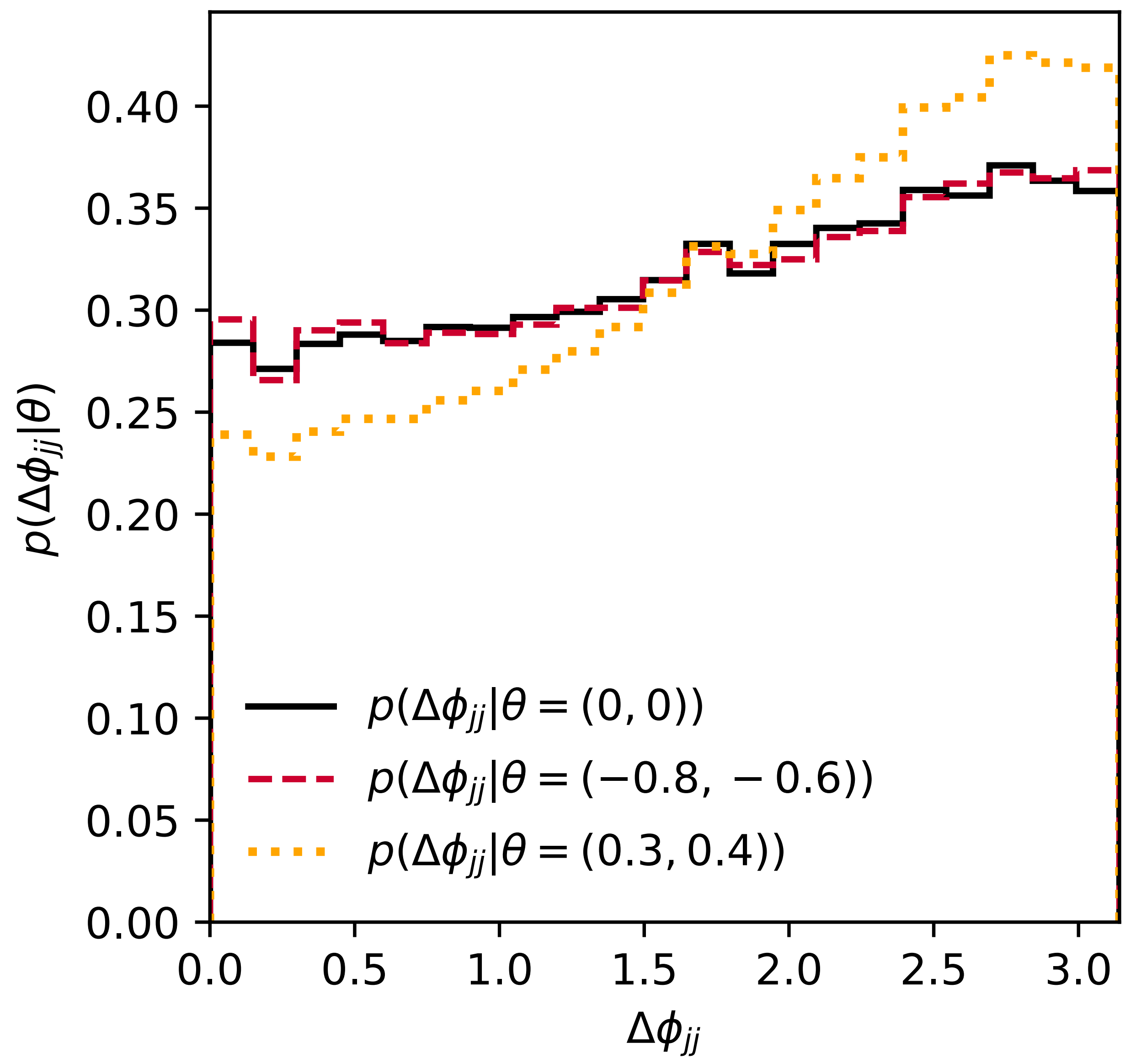
Perfect match for EFT measurements

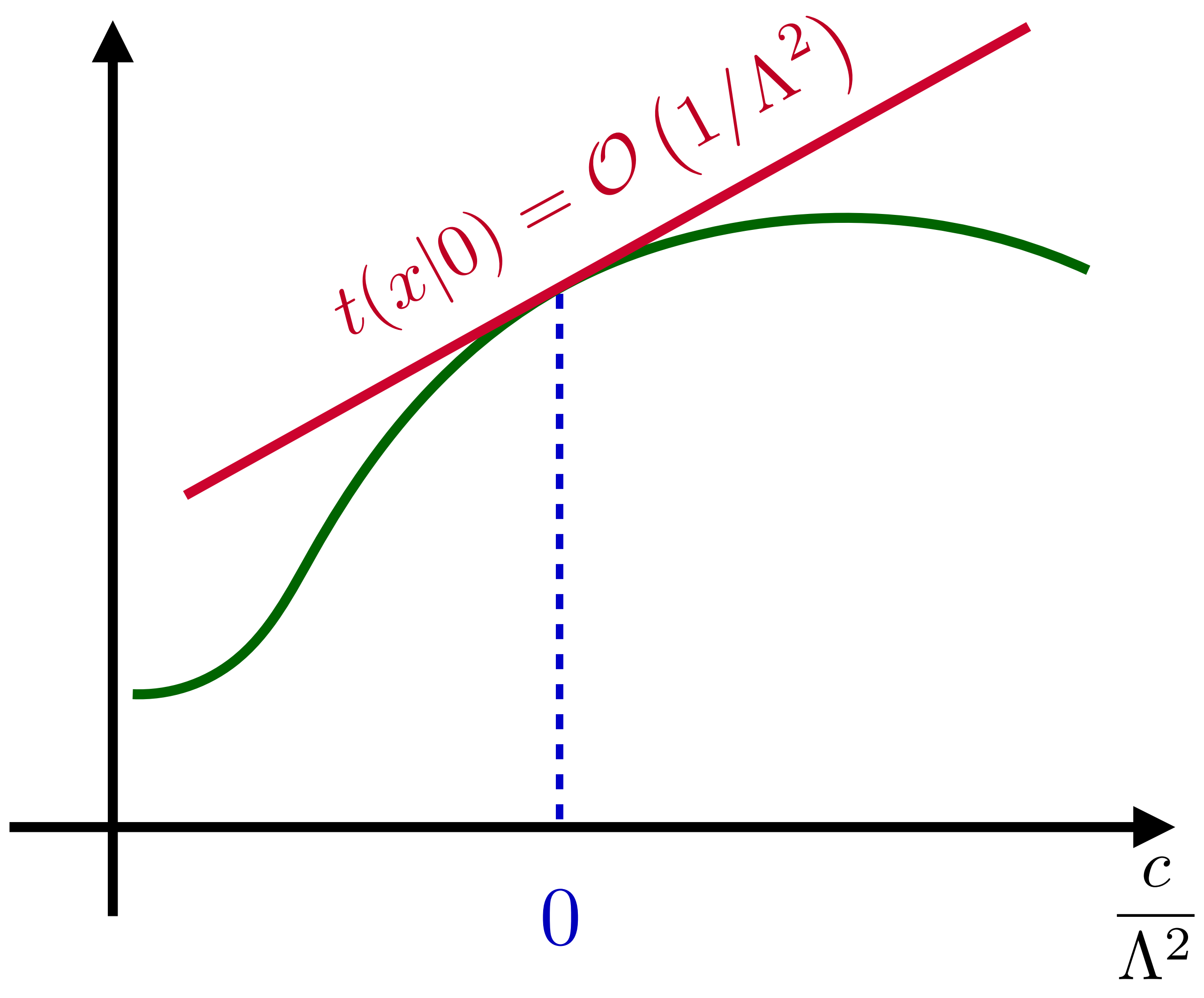


- Good for subtle kinematic effects

  (Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)

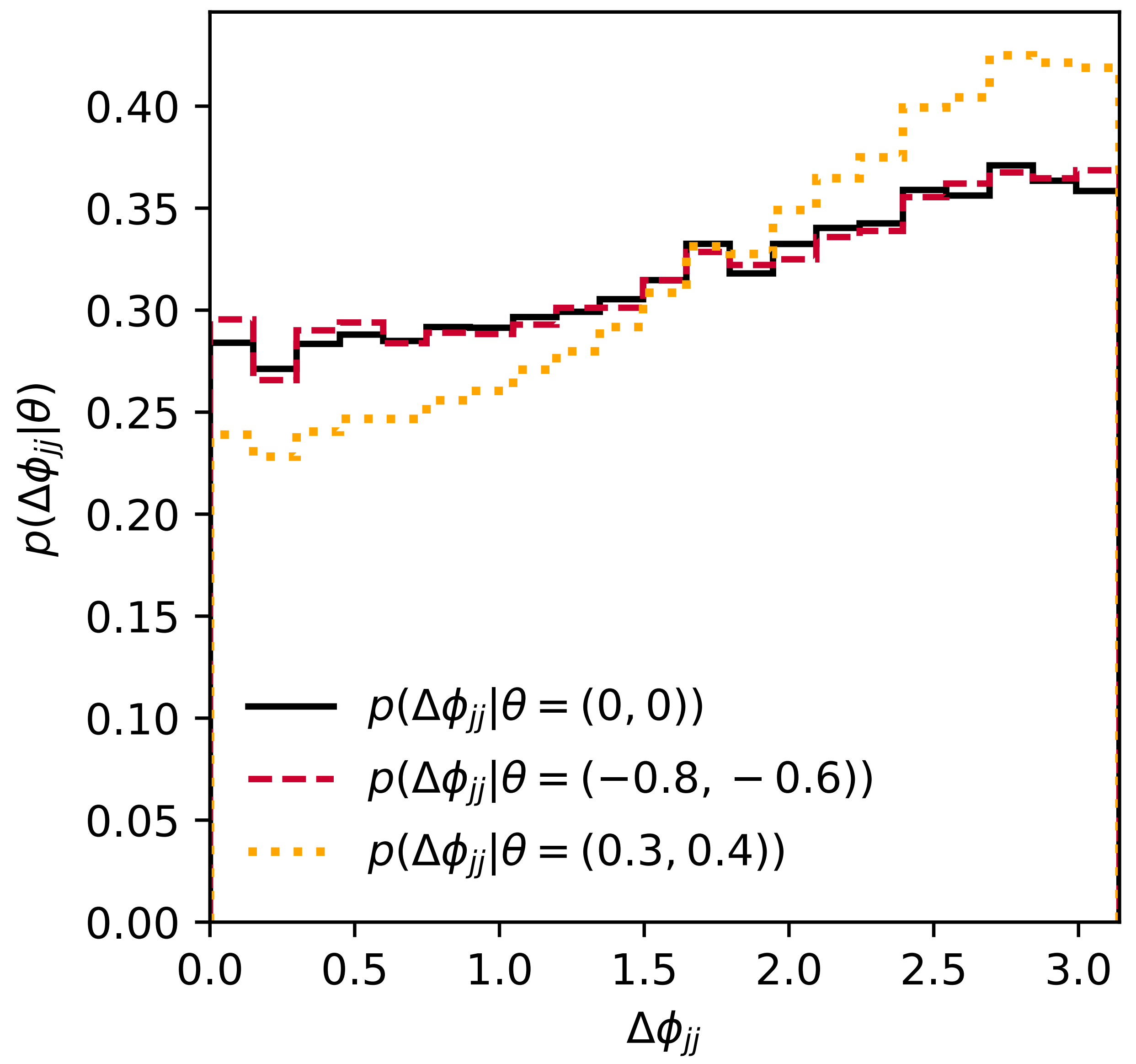
Perfect match for EFT measurements



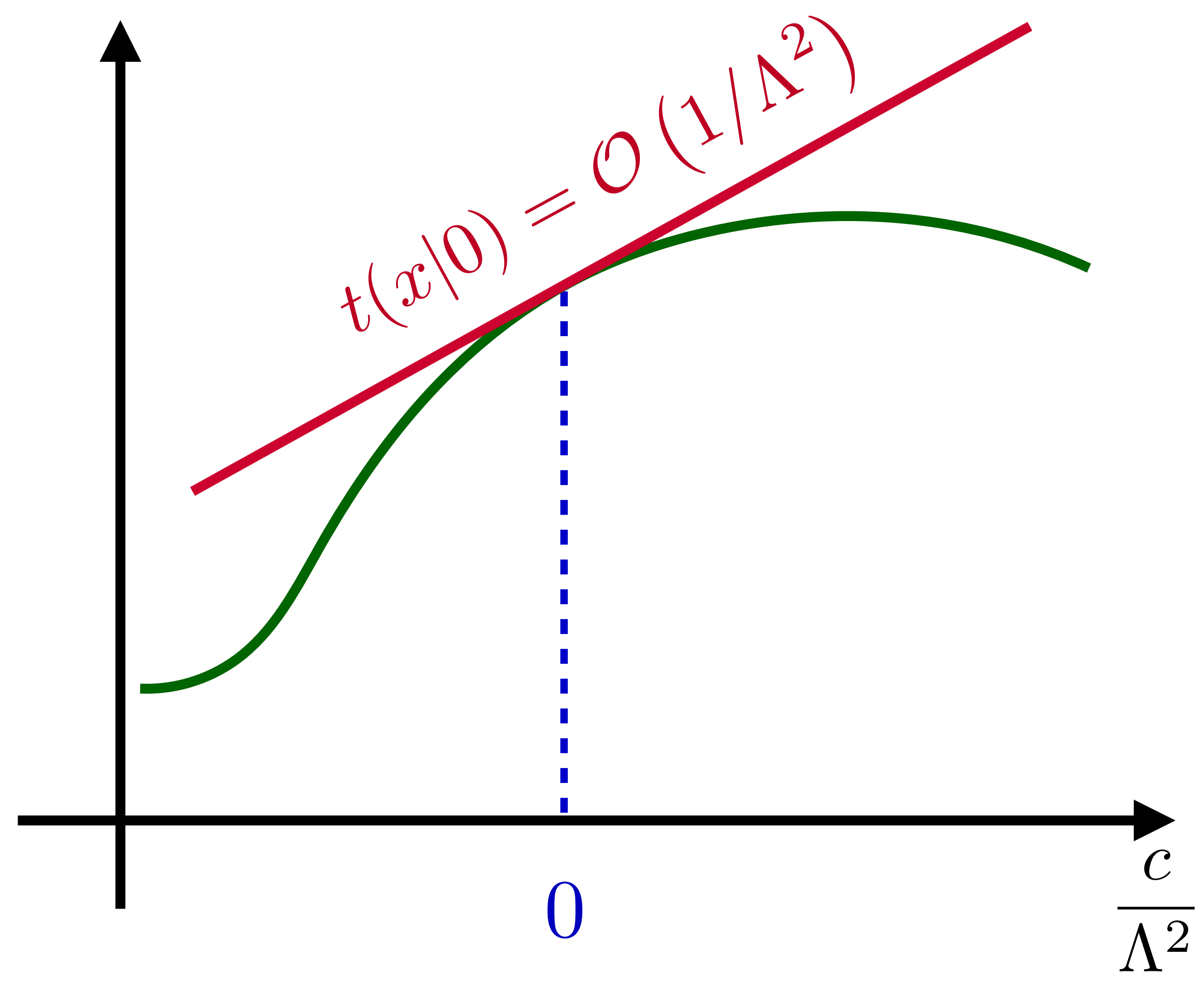


- Good for subtle kinematic effects
  (Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)
- Interference effects can be isolated using SALLY at the SM

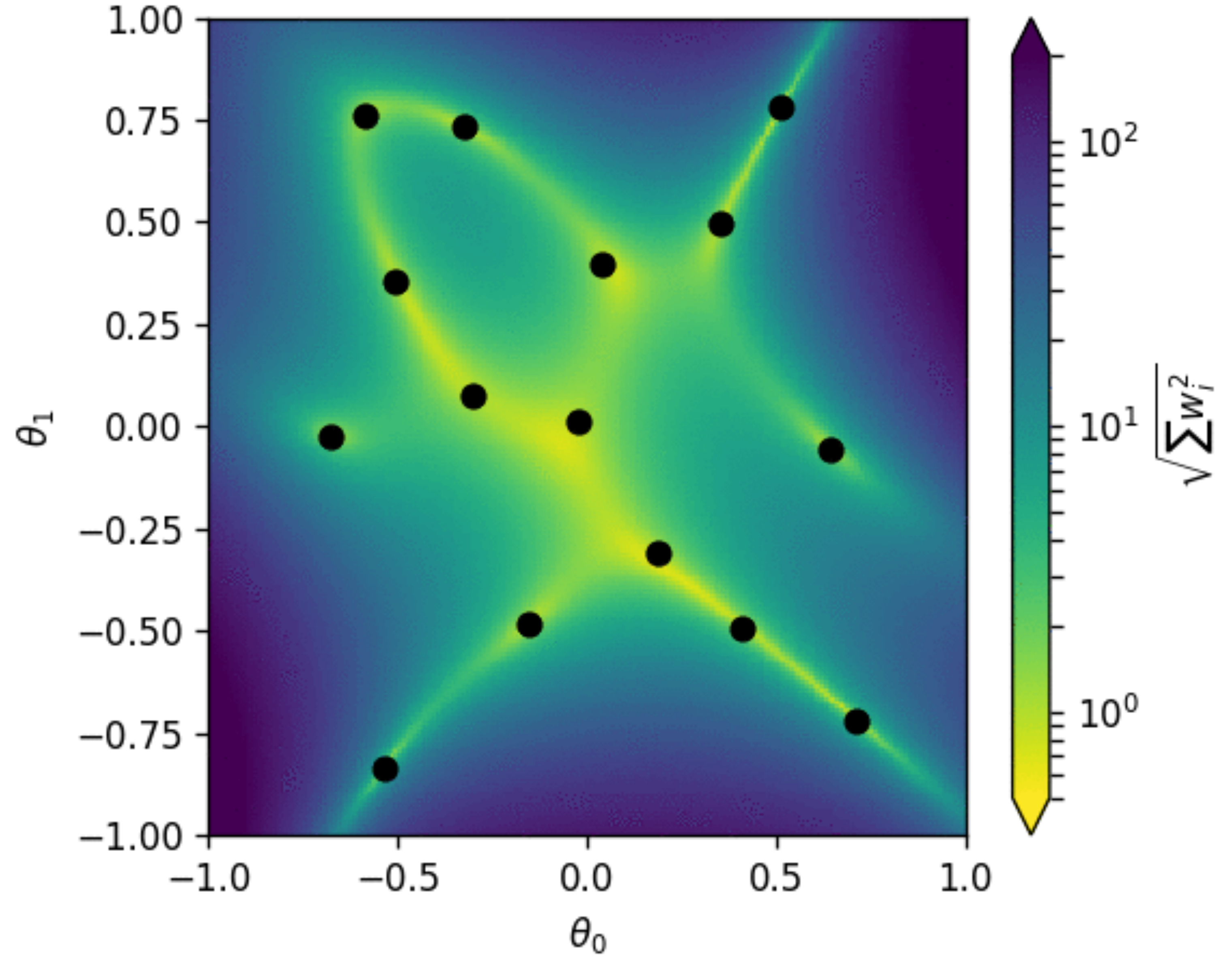
Perfect match for EFT measurements



- Good for subtle kinematic effects
  (Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)



- Interference effects can be isolated using SALLY at the SM



- Morphing techniques allow fast reweighting to any parameter points
  [e.g. ATL-PHYS-PUB-2015-047]

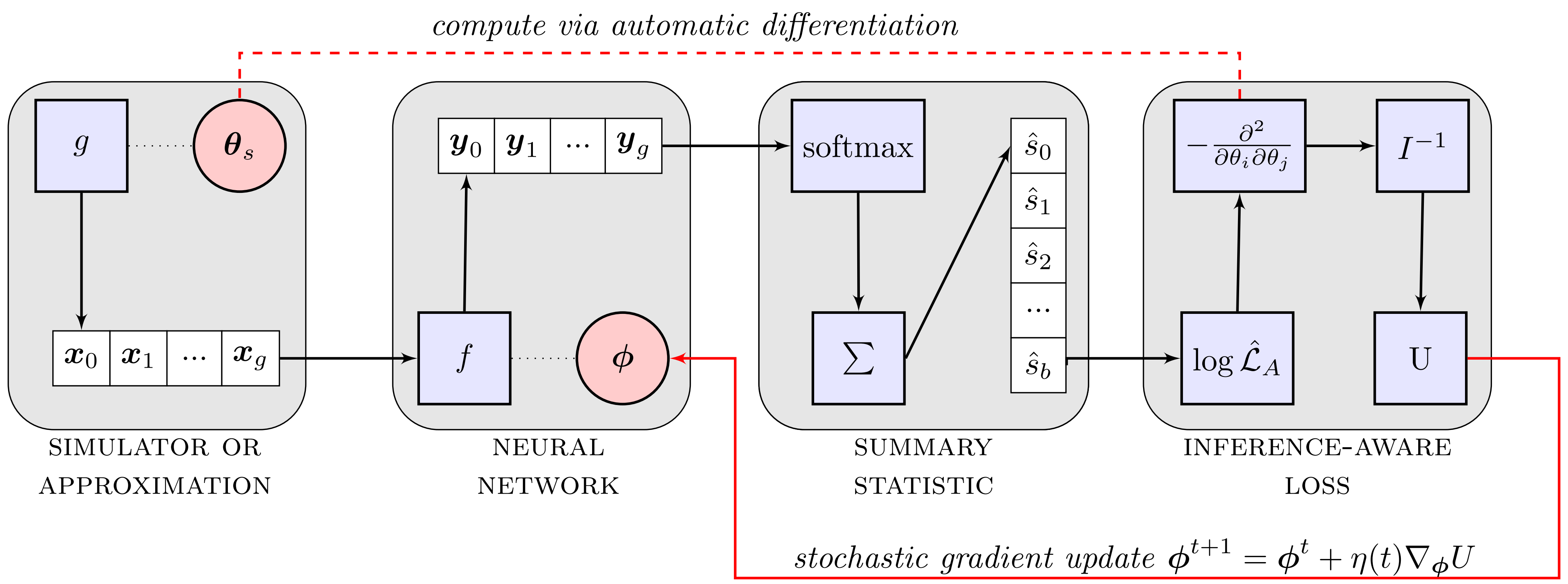
End-to-end optimization with autodiff

- With tools like MadMiner the objective is to learn a likelihood ratio, which is known optimal properties for measurements etc.
- In INFERNO and Neo the inference objective is directly optimized

INFERNO: Inference-Aware Neural Optimisation

Pablo de Castro
INFN - Sezione di Padova
pablo.de.castro@cern.ch

Tommaso Dorigo
INFN - Sezione di Padova
tommaso.dorigo@cern.ch



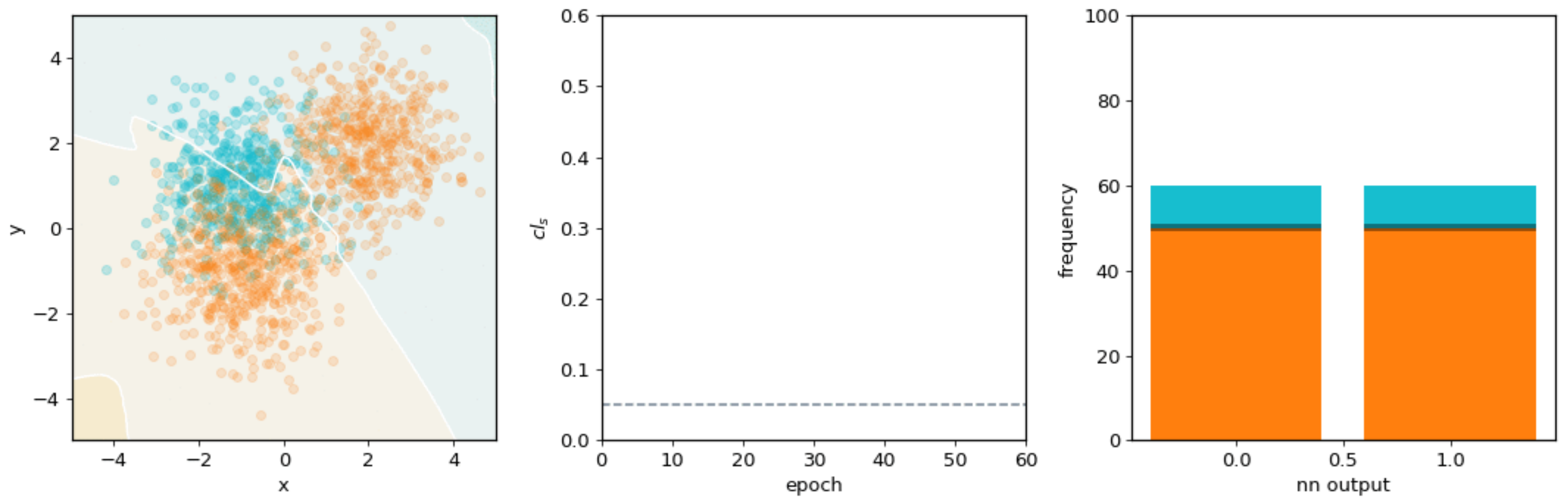
Kyle Cranmer @KyleCranmer · 19h
Take note! Here is a nice example of differentiable programming. It shows end-to-end optimization of a NN for event categorization wrt. final statistical analysis (using pyhf). Requires running gradients through results of maximum likelihood with fixed-point differentiation 👏

Nathan Simpson @ CERN
@phi_nate

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich_ !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)



10:58 AM · Mar 5, 2020 · Twitter Web App

https://github.com/pyhf/neos

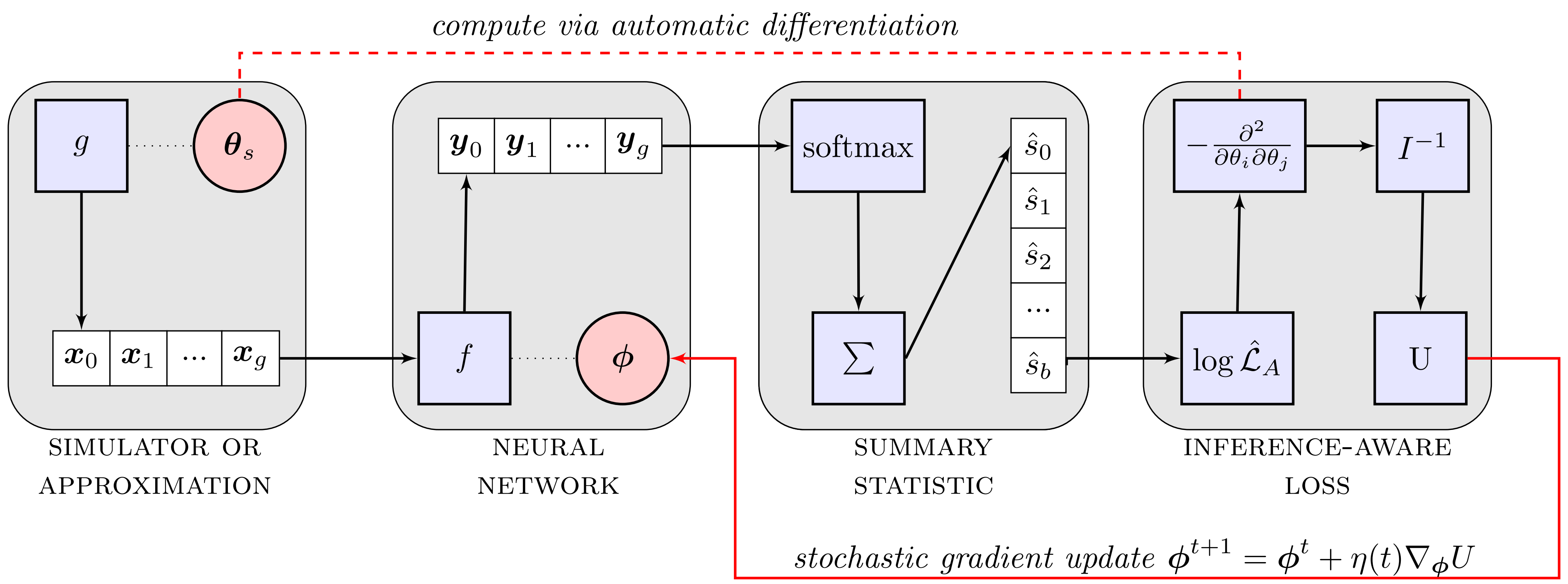
End-to-end optimization with autodiff

- With tools like MadMiner the objective is to learn a likelihood ratio, which is known optimal properties for measurements etc.
- In INFERNO and Neo the inference objective is directly optimized

INFERNO: Inference-Aware Neural Optimisation

Pablo de Castro
INFN - Sezione di Padova
pablo.de.castro@cern.ch

Tommaso Dorigo
INFN - Sezione di Padova
tommaso.dorigo@cern.ch



Kyle Cranmer @KyleCranmer · 19h

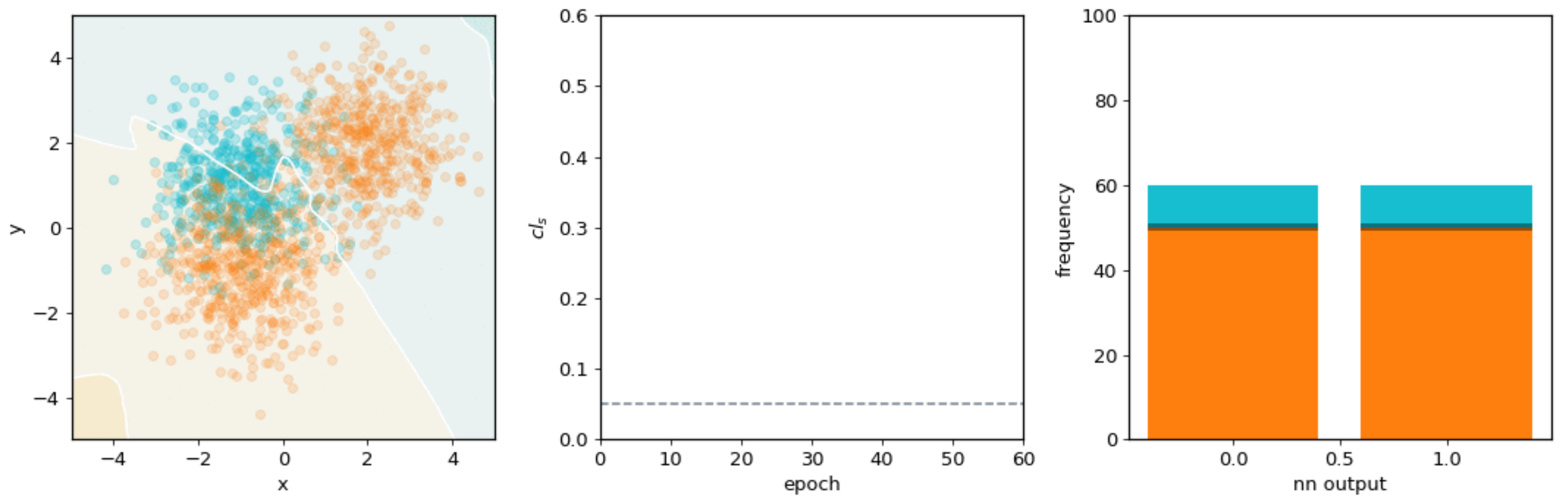
Take note! Here is a nice example of differentiable programming. It shows end-to-end optimization of a NN for event categorization wrt. final statistical analysis (using pyhf). Requires running gradients through results of maximum likelihood with fixed-point differentiation 👏

Nathan Simpson @ CERN
@phi_nate

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich_ !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

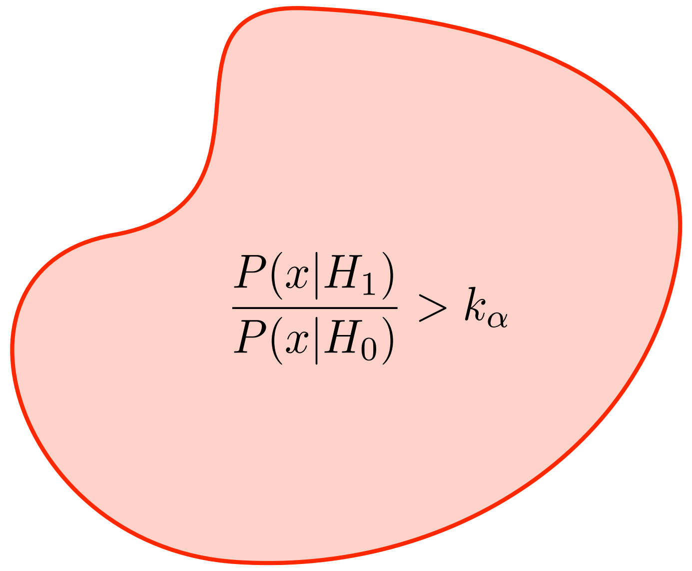
try it yourself at github.com/pyhf/neos! :)



10:58 AM · Mar 5, 2020 · Twitter Web App

https://github.com/pyhf/neos

Recap on Likelihood Ratios



For signal vs. background searches:

- **Neyman-Pearson Lemma**: optimal hypothesis test given by **likelihood ratio** (basis of Higgs search)
- Likelihood ratio $\frac{p(x|\theta_0)}{p(x|\theta_1)}$ also used for exclusion contours

For estimates of parameters $\hat{\theta}$

- **Cramér-Rao bound** states $\mathrm{cov}[\hat{\theta}|\theta_0]_{ij} \geq I^{-1}_{ij}(\theta_0)$ where I_{ij} is the **Fisher-information matrix** (Hessian of log-likelihood)
- Motivates **Information Geometry** as a phenomenological tool
- Maximum-likelihood (asymptotically) saturates the bound

Note: $\nabla_\theta \log p(x|\theta)$ acts like a likelihood ratio locally

Cramér-Rao Bound

The minimum variance bound on an unbiased estimator is given by the Cramér-Rao bound:

$$\mathrm{cov}[\hat{\theta}|\theta_0]_{ij} \geq I^{-1}_{ij}(\theta_0)$$

Expected error of best-fit parameter

Inverse of Fisher information

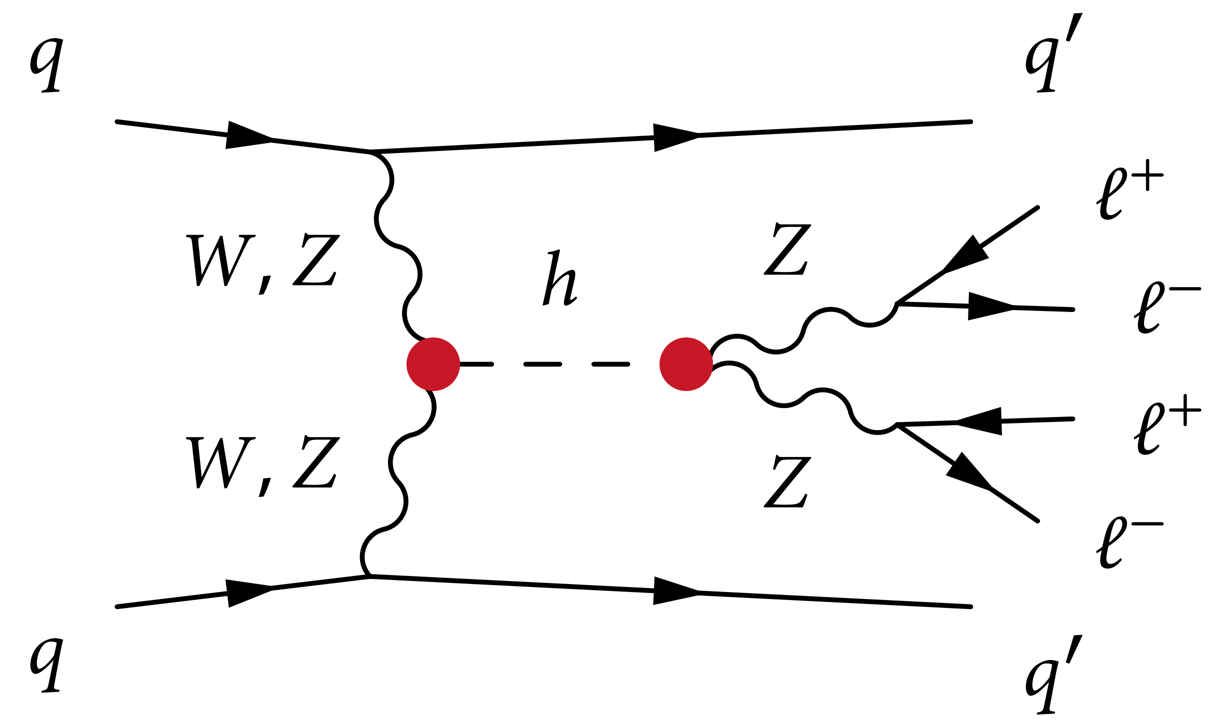
Fisher information matrix (is also a Riemannian metric!)

$$I_{ij}[\theta] = -\mathbb{E}\left[\frac{\partial^2 \log p(x|\theta)}{\partial\theta_i\,\partial\theta_j}\middle|\theta\right]$$

Maximum Likelihood Estimators *asymptotically* reach this bound

Challenge for EFT

Let θ denote the coefficients of higher dimensional operators in the Lagrangian, x be high-dimensional data associated to an event, and $p(x \mid \theta) = \dfrac{1}{\sigma(\theta)} \dfrac{d\sigma}{d\theta}$ be the distribution for the data



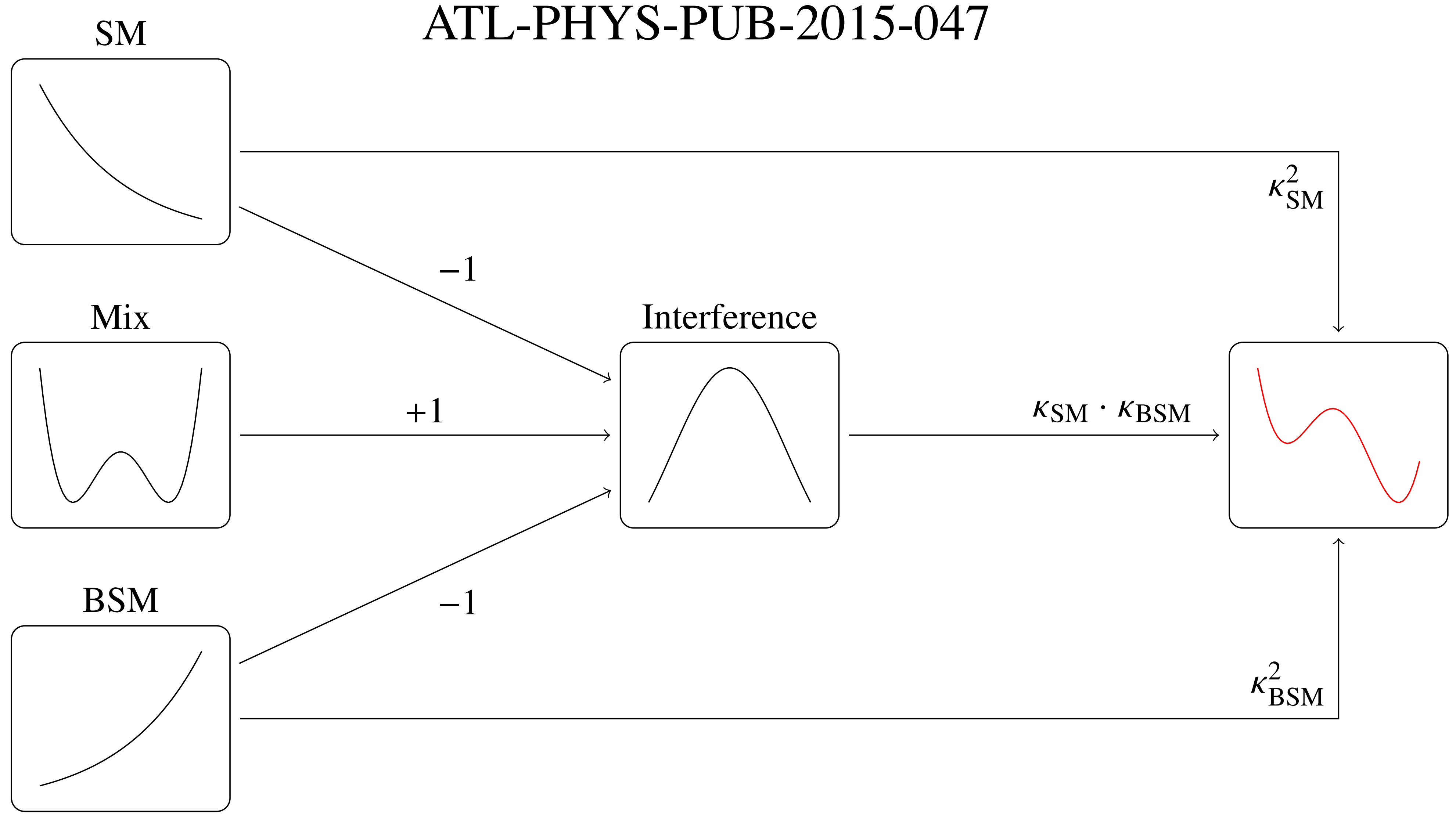
- we want to compare any two points in EFT parameter space
- evaluate the **likelihood ratio** $r(x|\theta_0, \theta_1) \equiv \dfrac{p(x|\theta_0)}{p(x|\theta_1)}$

Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

- It would be very computationally expensive (infeasible) to generate samples for every value of θ and estimate $p(x \mid \theta)$ with histograms. Small changes mean we need a lot of MC events!
- Ideally we could directly estimate the **score** $t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0}$

EFT Embedded in a vector space

ATL-PHYS-PUB-2015-047



Difficulty is that one changes the parameters of the EFT, the distributions p(x|θ) change due to interference.

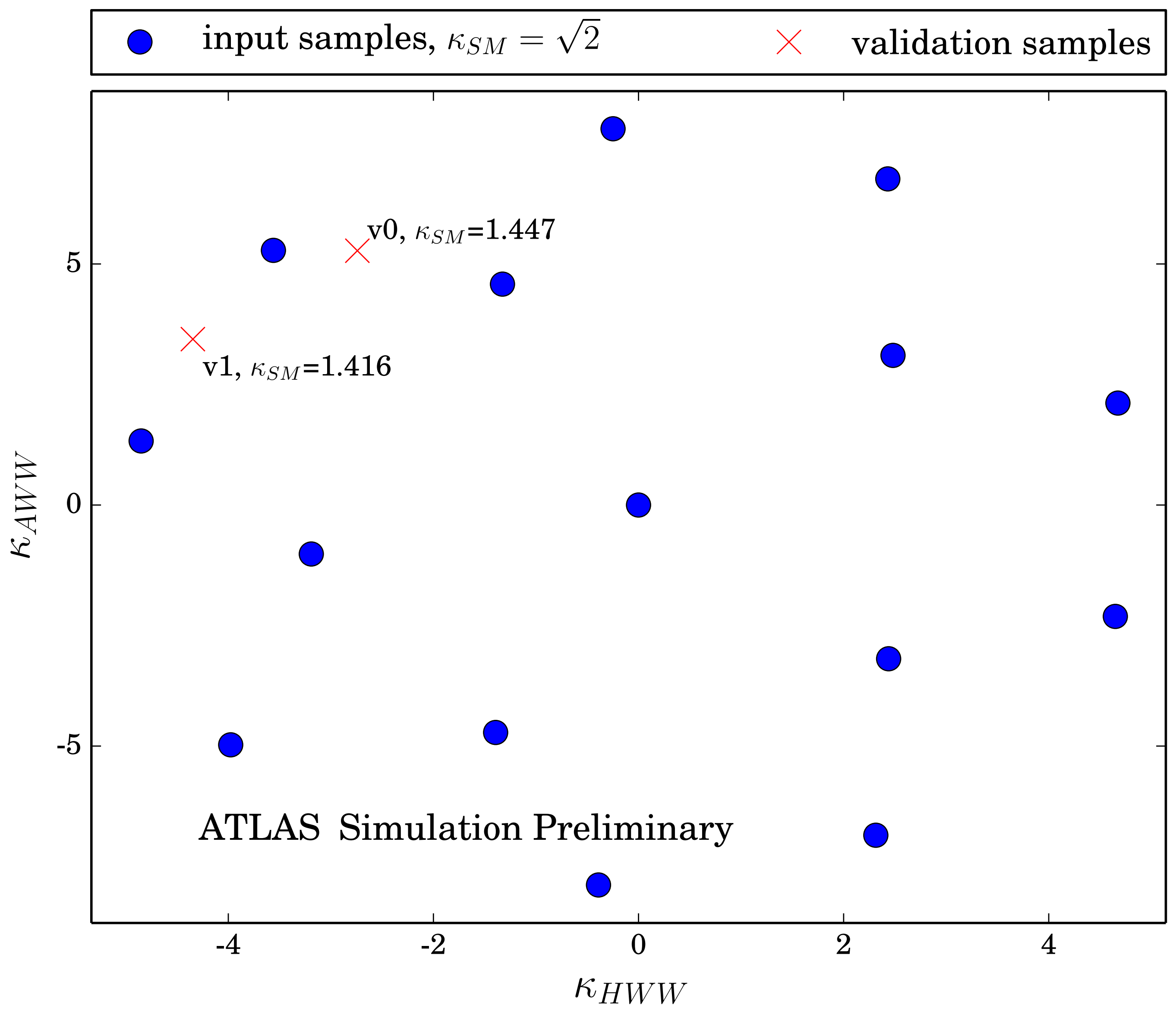
But there is a trick:

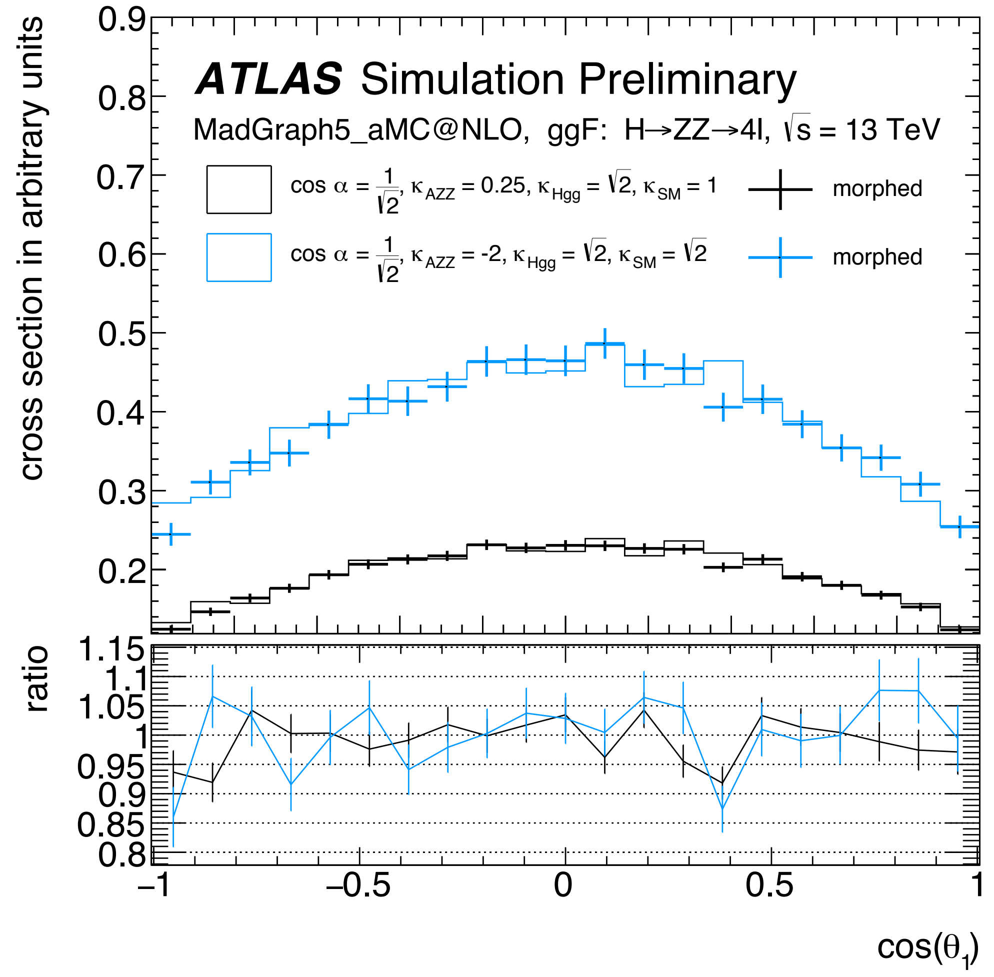
Simple example:

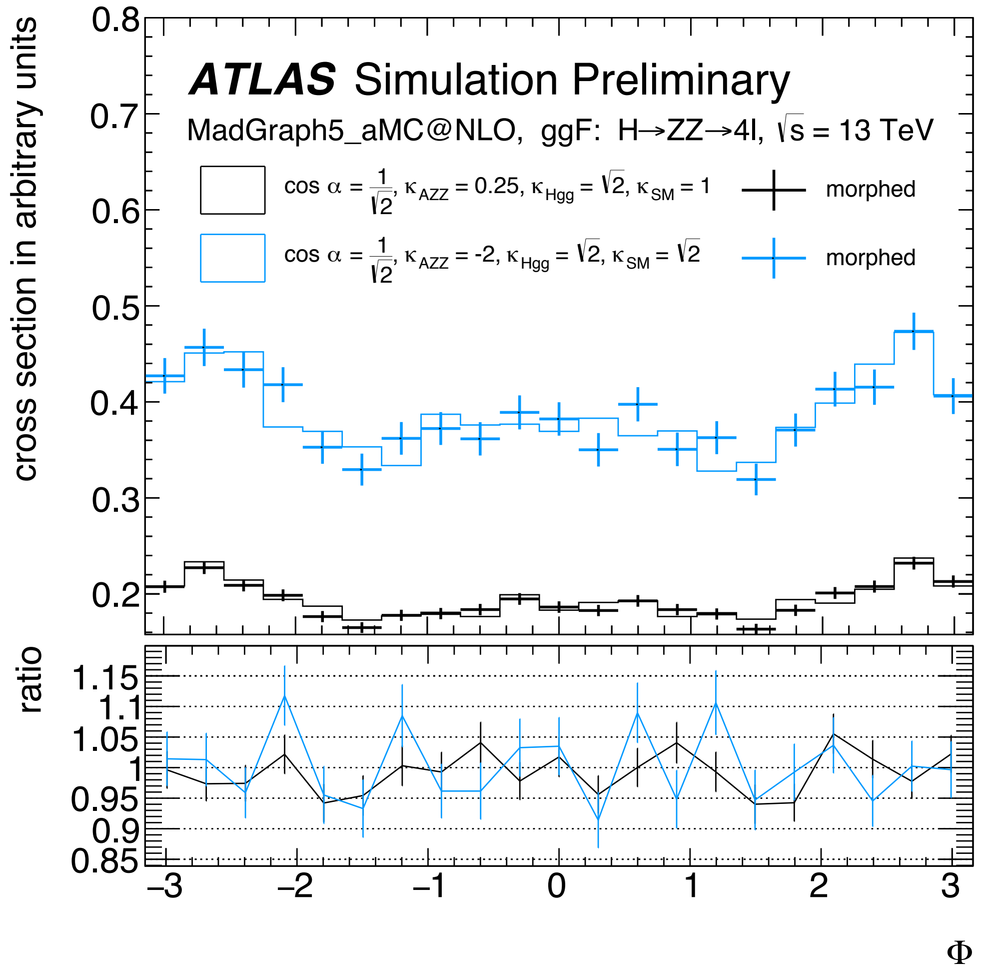
$$|g_1 M_{SM} + g_2 M_{BSM}|^2 = g_1^2 |M_{SM}|^2 + 2 g_1 g_2 Re\left[M_{SM}^* M_{BSM}\right] + g_2^2 |M_{BSM}|^2$$

3-d vector space, distribution for any point in this space is linear mixture of distribution for 3 basis samples!

(real examples need more basis samples)







EFT Decomposition

production | decay

$$d\sigma \propto \left|\left(\mathcal{M}^p_{\text{SM}} + \sum_i \frac{f_i}{\Lambda^2}\mathcal{M}^p_i\right)\left(\mathcal{M}^d_{\text{SM}} + \sum_j \frac{f_j}{\Lambda^2}\mathcal{M}^d_j\right)\right|^2$$

Express EFT as a mixture:

$$p(x|\theta) = \sum_c w_c(\theta) p_c(x)$$

$w_c(\theta)$ are polynomials

$\nabla_\theta \log p(x|\theta)$ is now possible!

Process	Number of components for n operators					
	$\mathcal{O}(\Lambda^0)$	$\mathcal{O}(\Lambda^{-2})$	$\mathcal{O}(\Lambda^{-4})$	$\mathcal{O}(\Lambda^{-6})$	$\mathcal{O}(\Lambda^{-8})$	$\sum$
hV / WBF production	1	n	$\frac{n(n+1)}{2}$			$\frac{(n+1)(n+2)}{2}$
$h \to VV$ decay	1	n	$\frac{n(n+1)}{2}$			$\frac{(n+1)(n+2)}{2}$
Production + decay	1	n	$\frac{n(n+1)}{2}$	$\binom{n+2}{3}$	$\binom{n+3}{4}$	$\binom{n+4}{4}$

Table 1: Number of components c as given in Eq. (6) for different processes, sorted by their suppression by the EFT cutoff scale Λ.

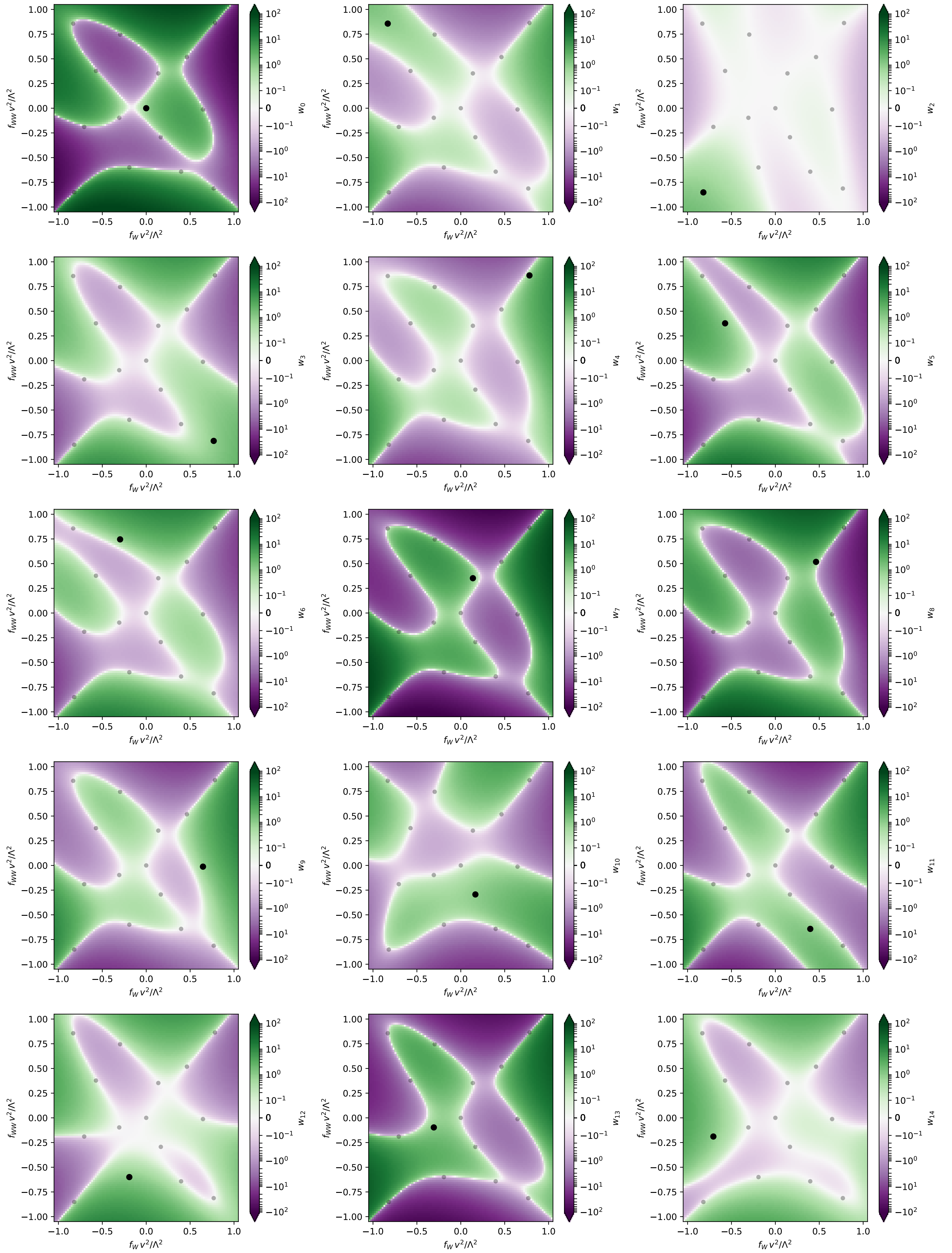


Figure 13: Morphing weights $w_i(\boldsymbol{\theta})$ for basis points distributed over the full relevant parameter space.

For 2 BSM operators affecting VBF Higgs production and decay, we need a 15-D vector space

For 5 BSM operators we need 126-D vector space