

Constraining effective field theories with machine learning

Johann Brehmer

New York University

INFN Padova

June 6, 2019



Kyle Cranmer



Gilles Louppe



Juan Pavez



Felix Kling



Irina Espejo



Zubair Bhatti



Markus Stoye



Sid Mishra-Sharma



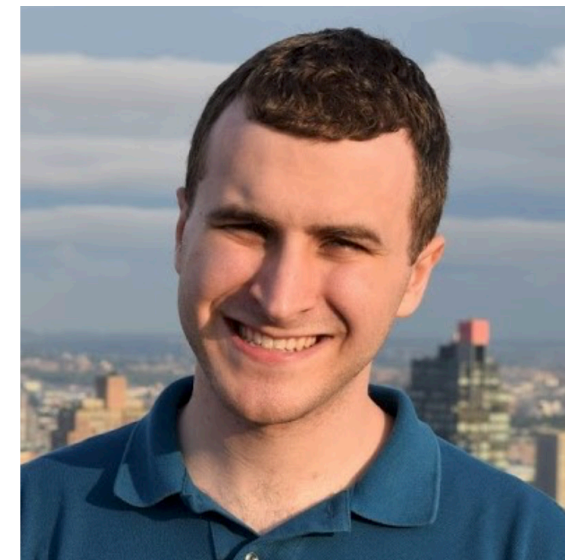
Joeri Hermans



Tilman Plehn



Sally Dawson



Sam Homiller



Josh Ruderman



Duccio Pappadopulo

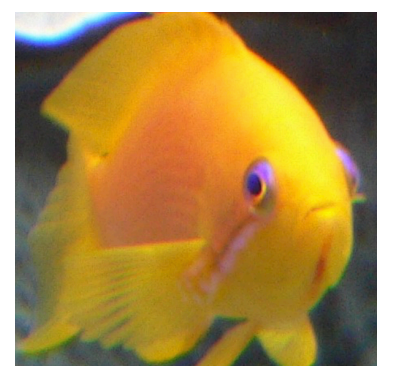


Marco Farina

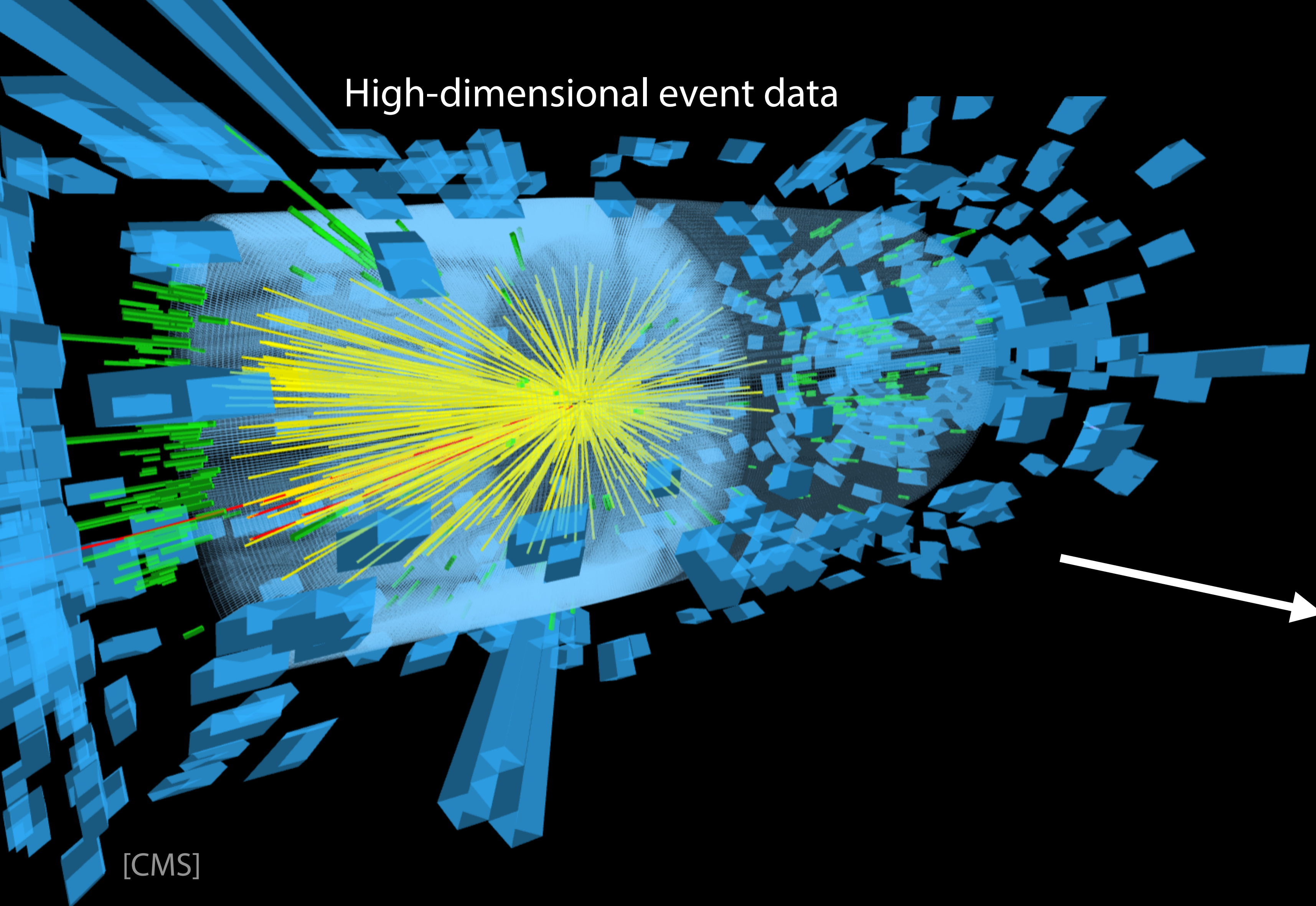
Thanks to Kyle, Gilles, Felix, Irina, and Sam for material and inspiration for slides!



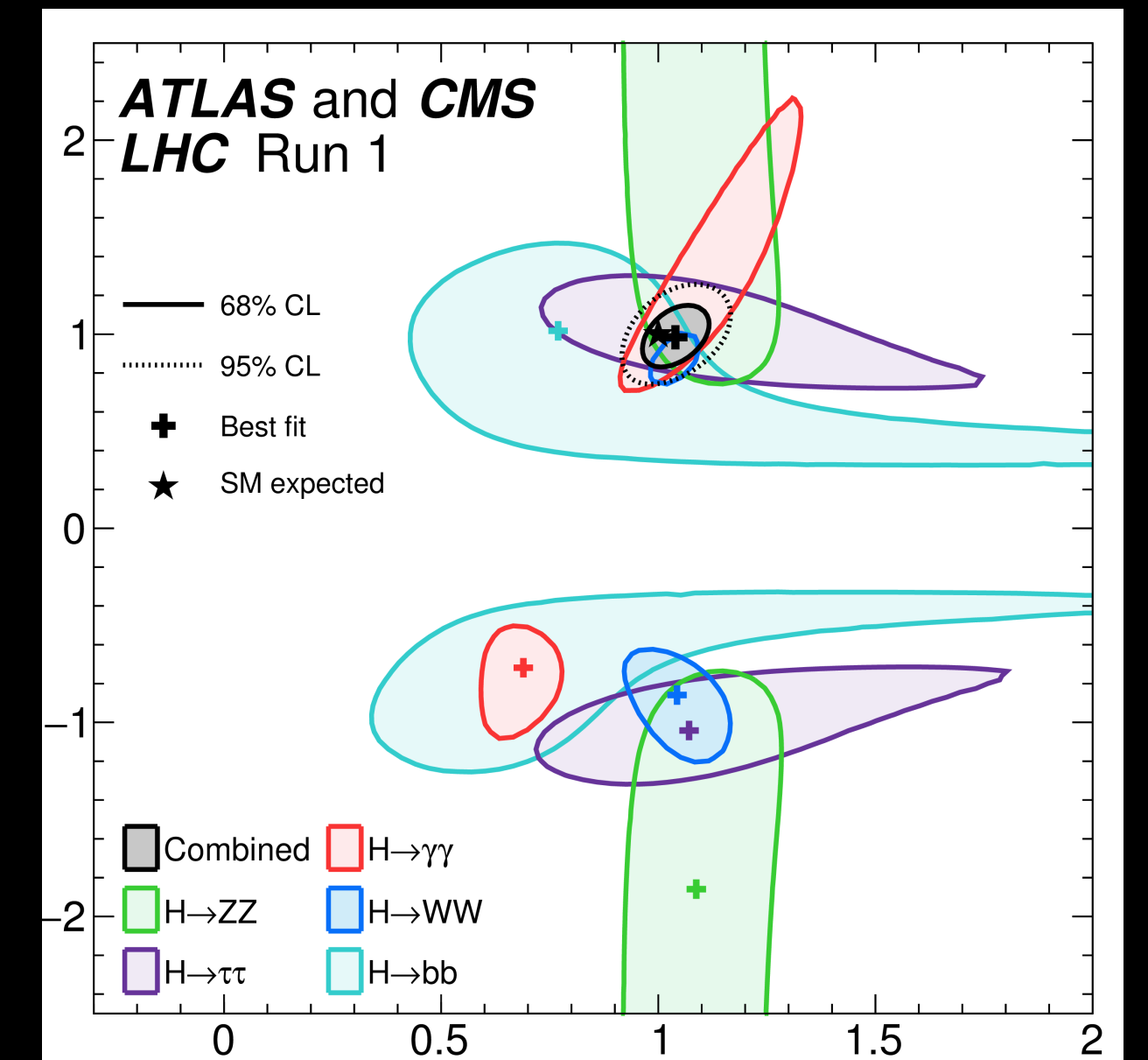
The SCAIFIN Project
scailfin.github.io



High-dimensional event data

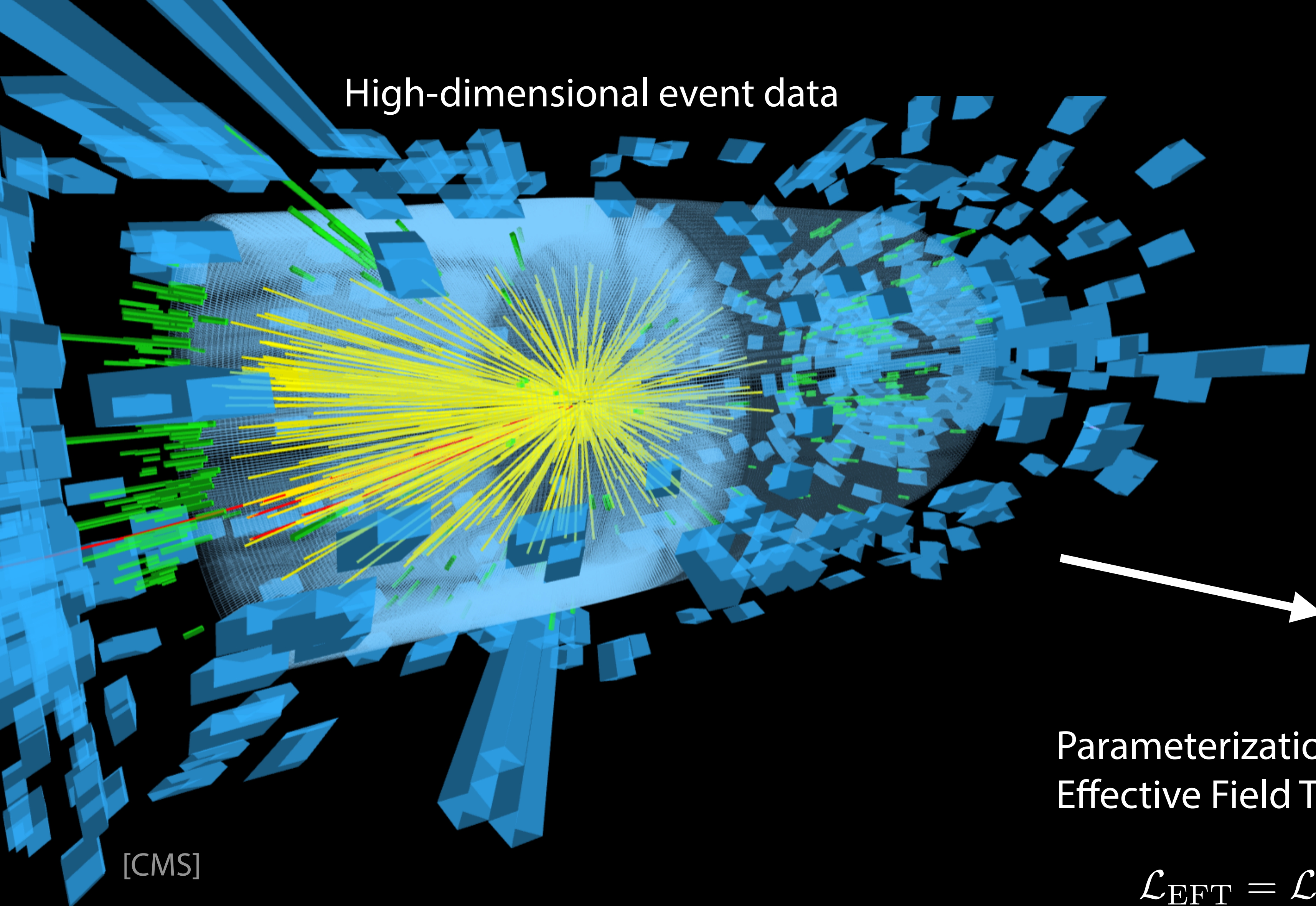


[CMS]

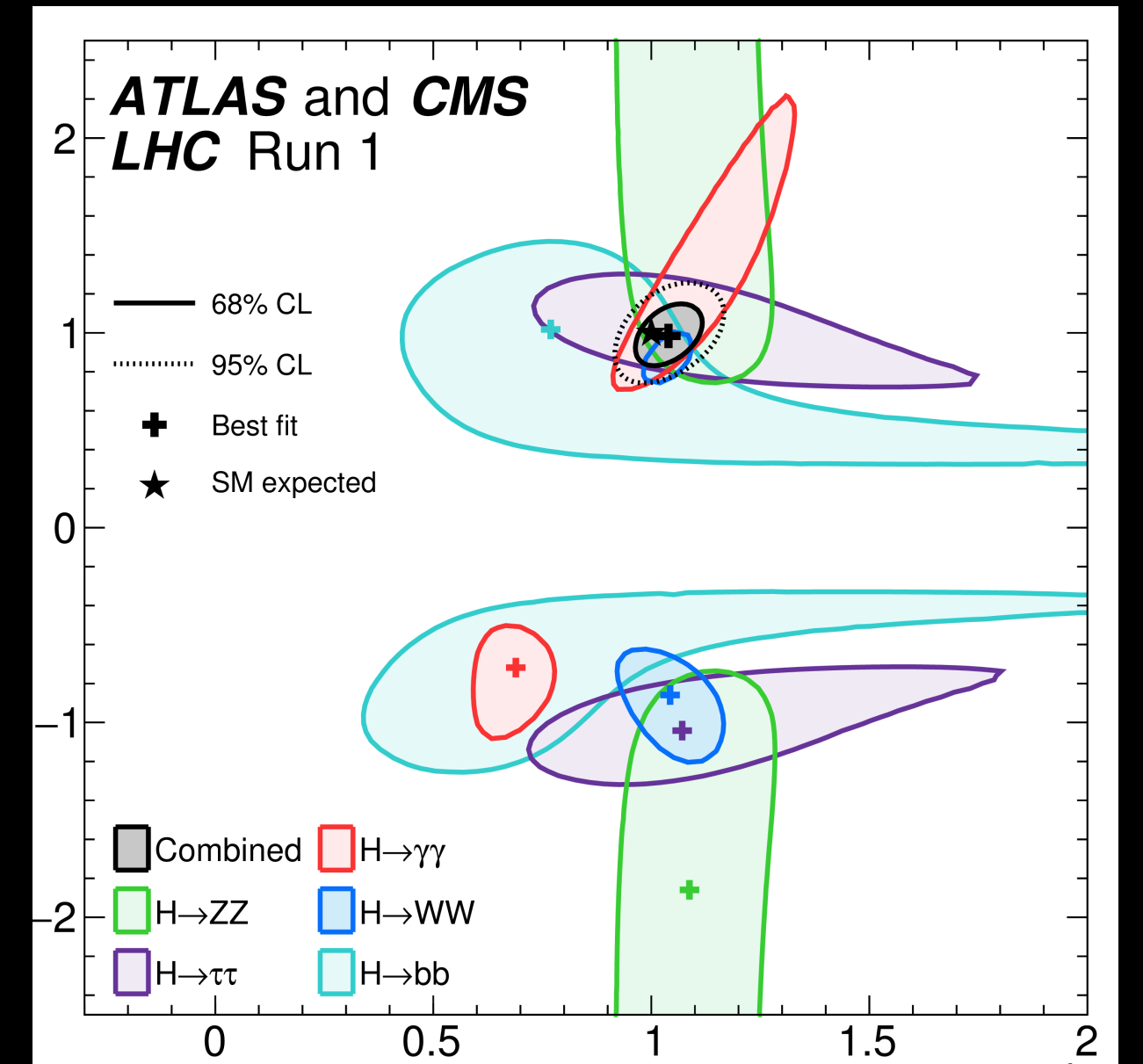


Precision constraints on indirect effects of new physics

High-dimensional event data



[CMS]



[ATLAS, CMS 1606.02266]

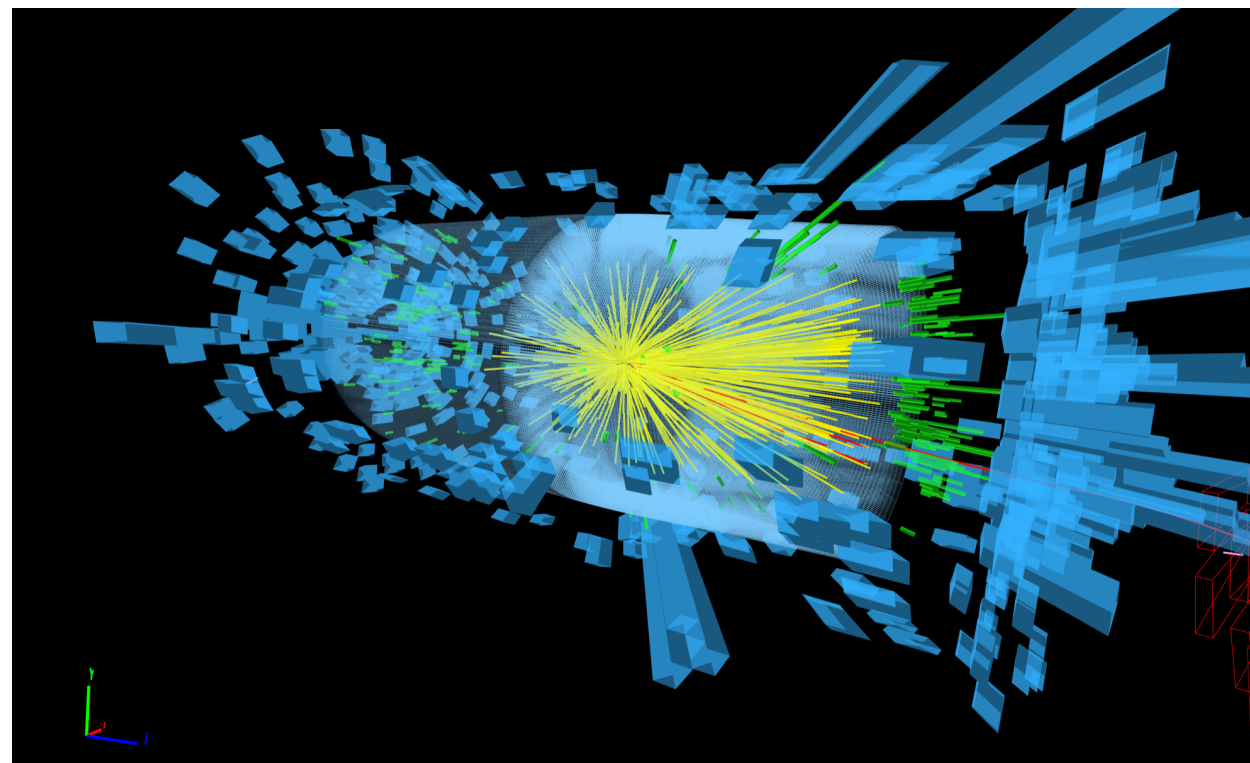
Precision constraints on indirect effects of new physics

Parameterization in Effective Field Theory:

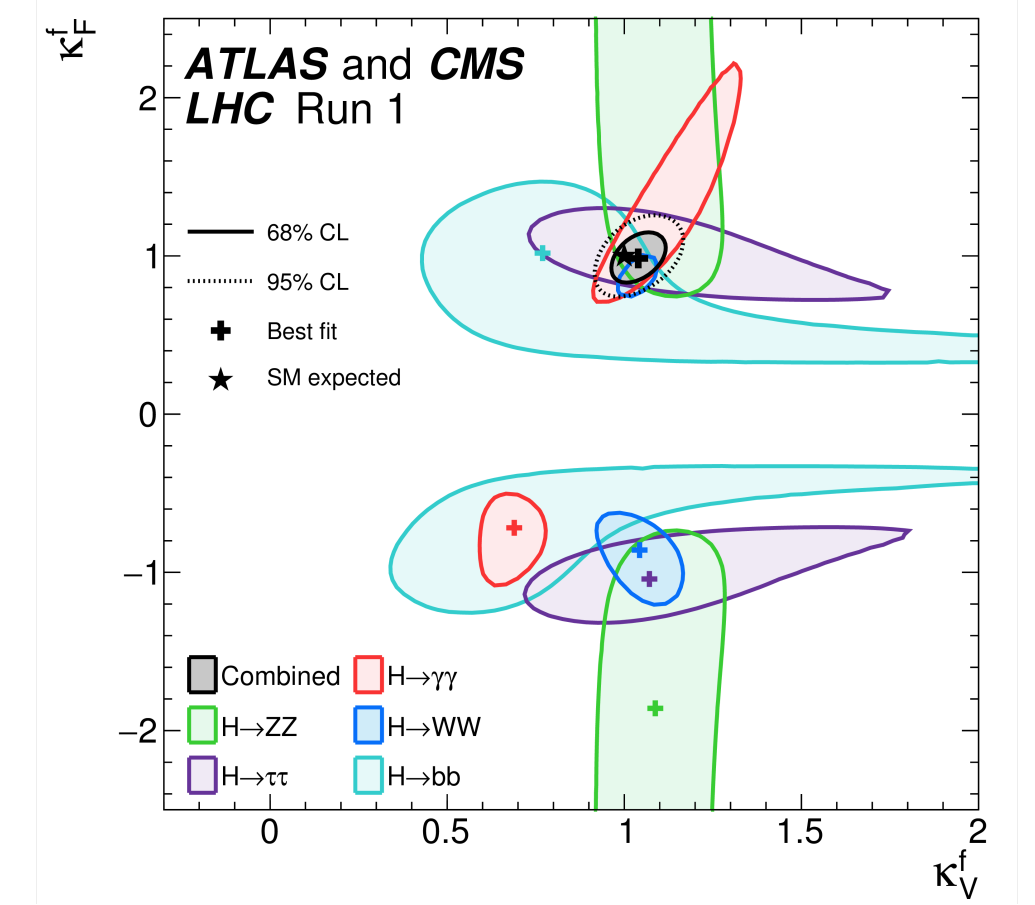
$$\mathcal{L}_{\text{EFT}} = \mathcal{L}_{\text{SM}} + \sum_i \frac{f_i}{\Lambda^2} \mathcal{O}_i + \dots$$

systematic expansion of new physics around Standard Model

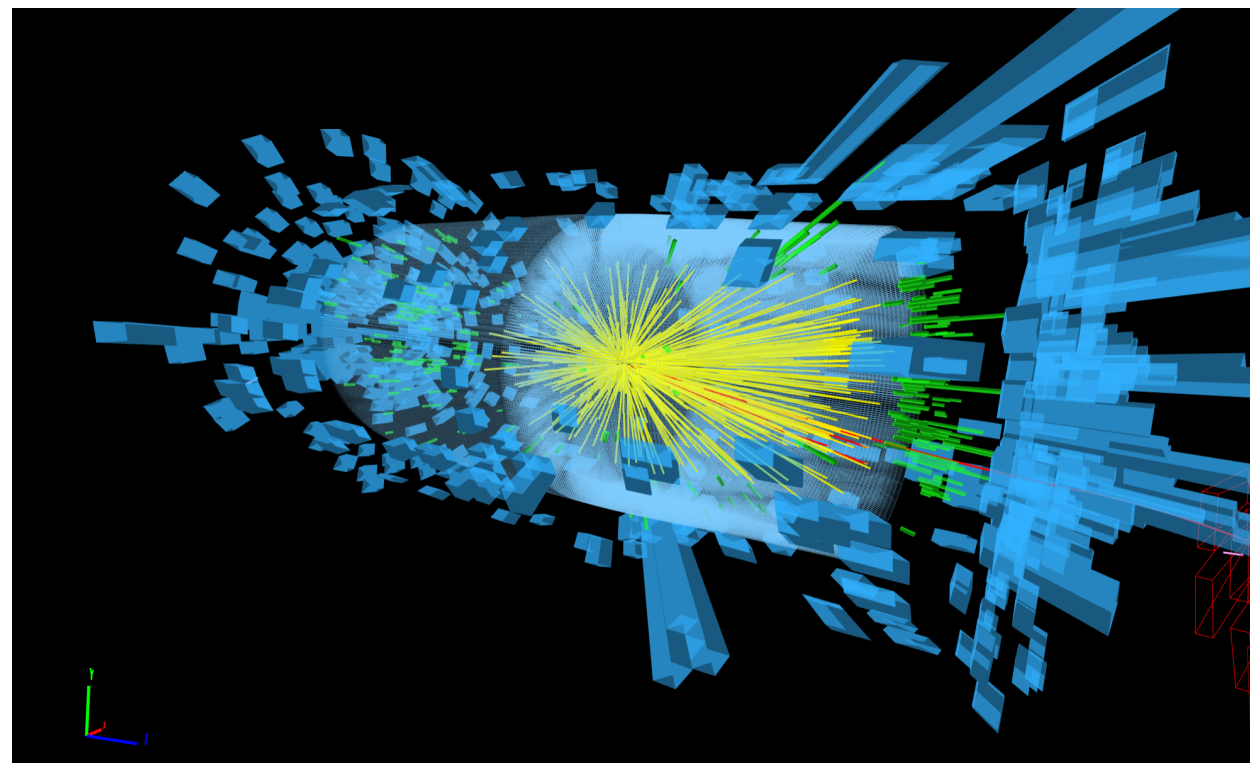
10s to 1000s "universal" parameters to measure



High-dimensional
event data x



Constraints on
parameters θ



High-dimensional event data x

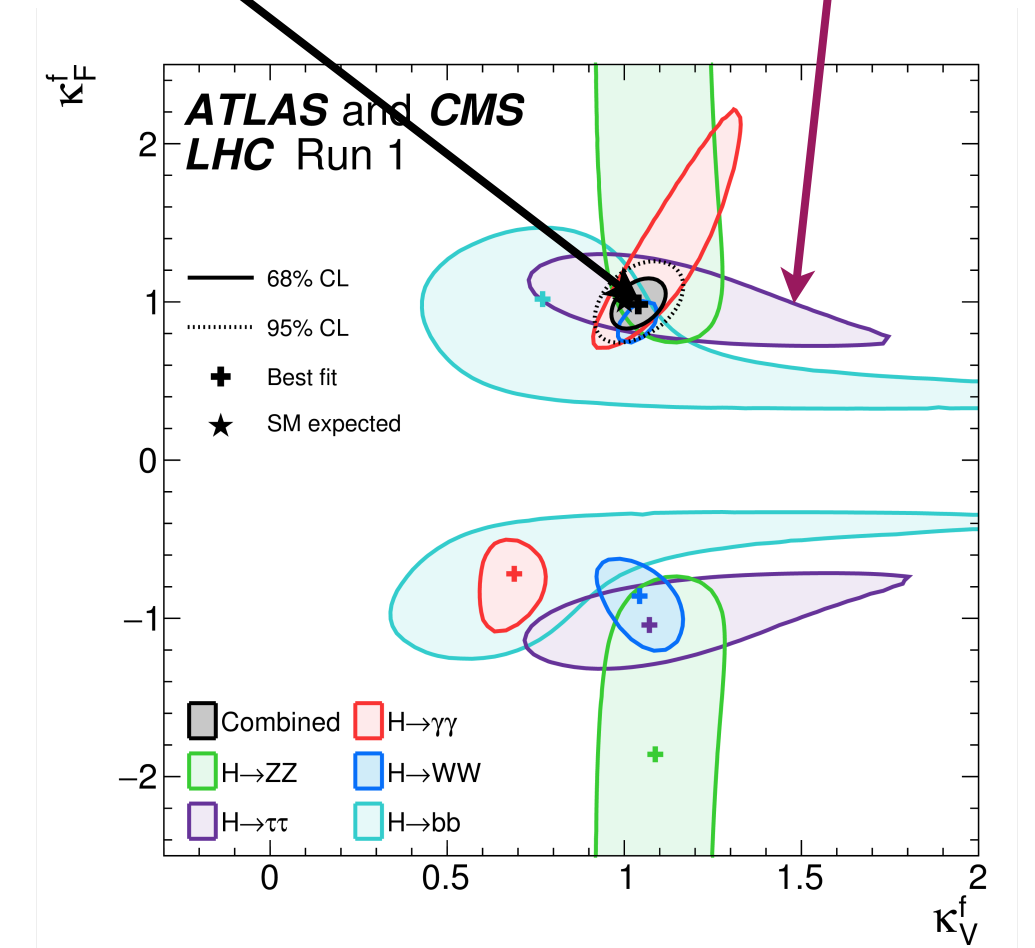


Likelihood function
 $p(x|\theta)$

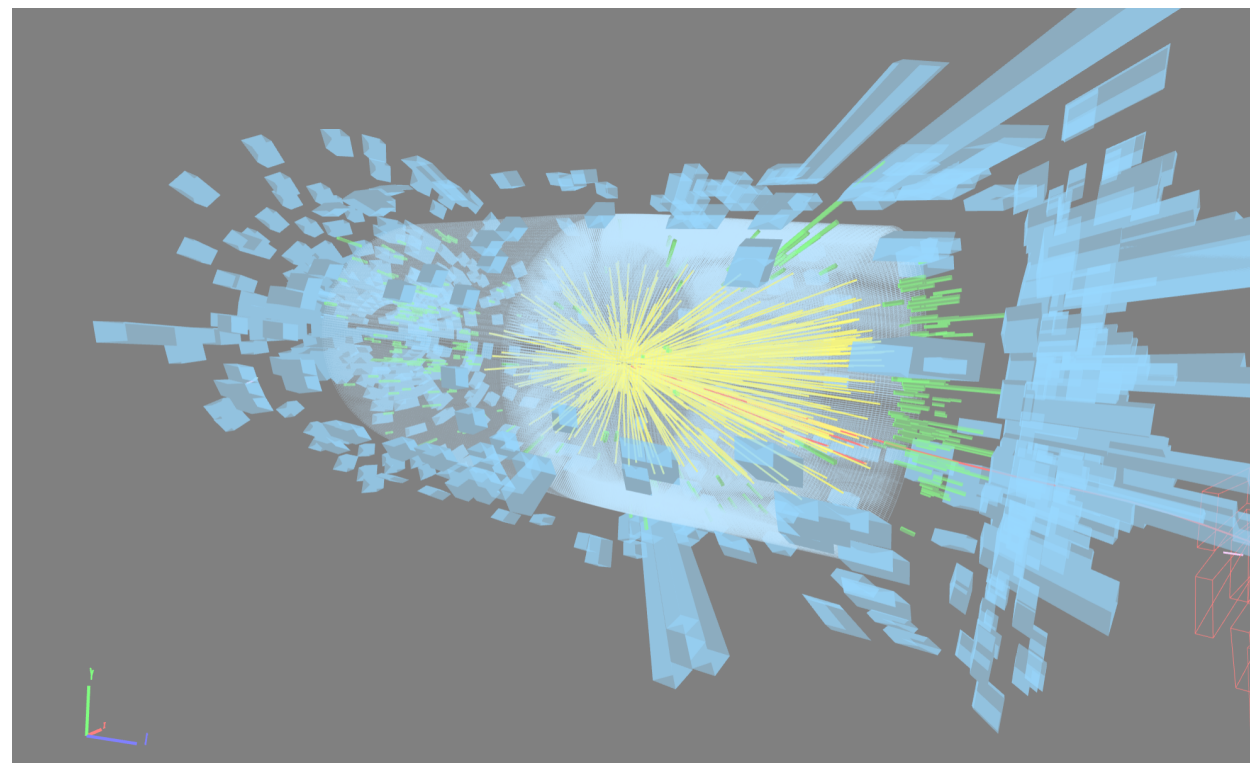


Maximum-likelihood estimator

Confidence limits based on likelihood ratio tests



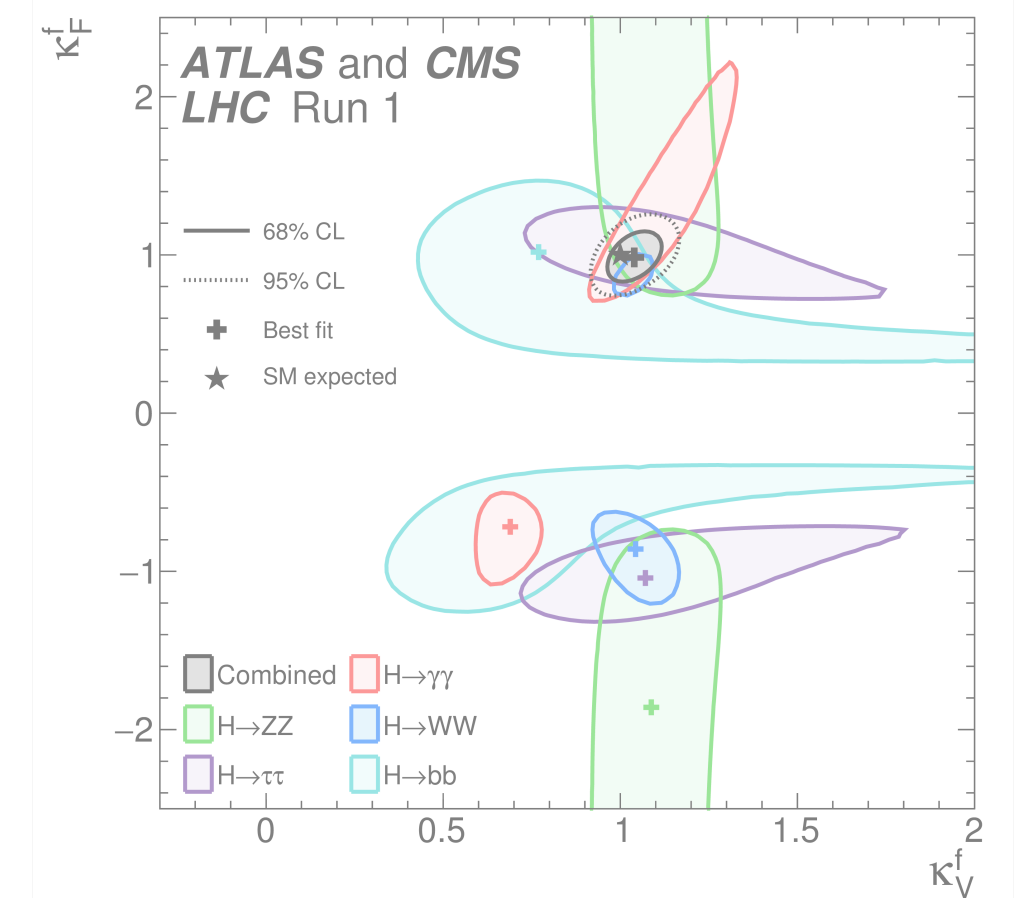
Constraints on parameters θ



High-dimensional event data x

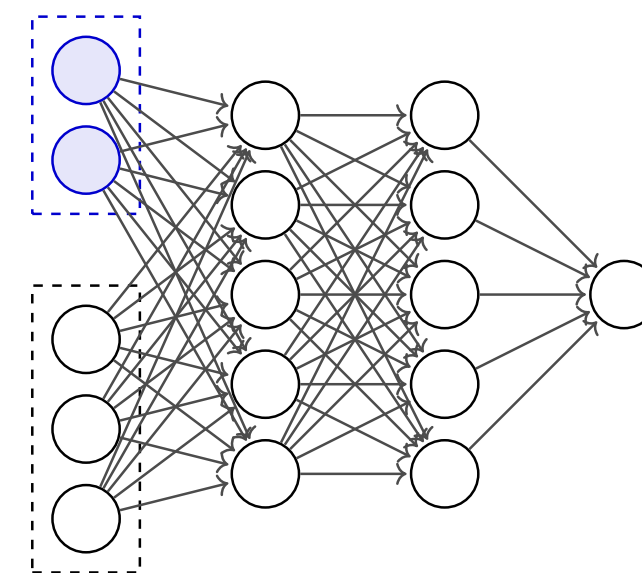
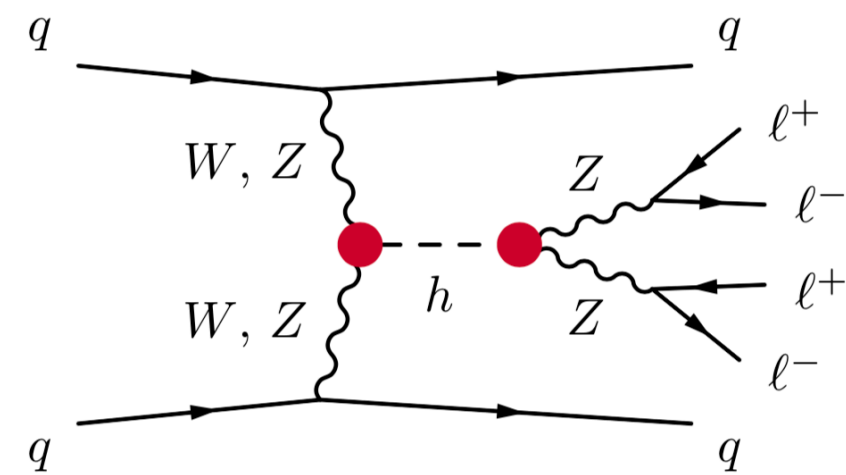


Likelihood function
 $p(x|\theta)$



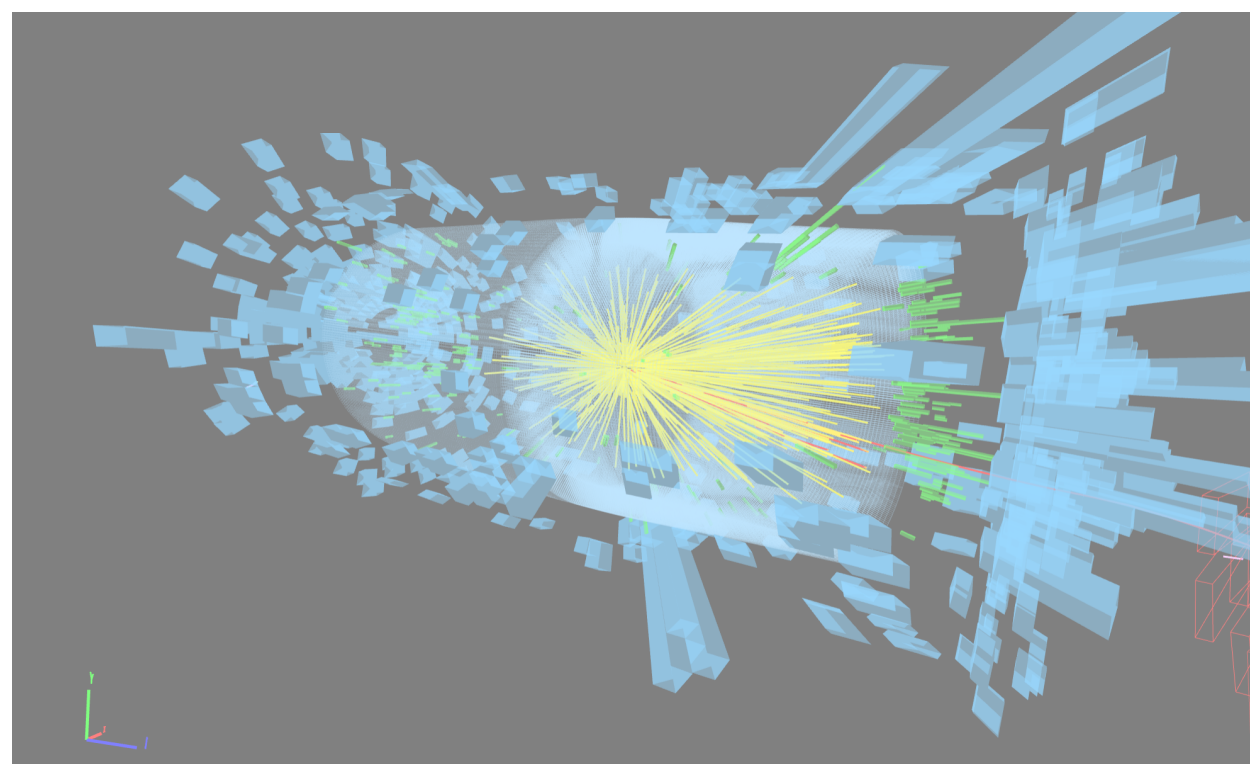
Constraints on parameters θ

Surprisingly, when we want to use high-dimensional data and have to deal with the detector response, we do not have a good way to calculate the likelihood.

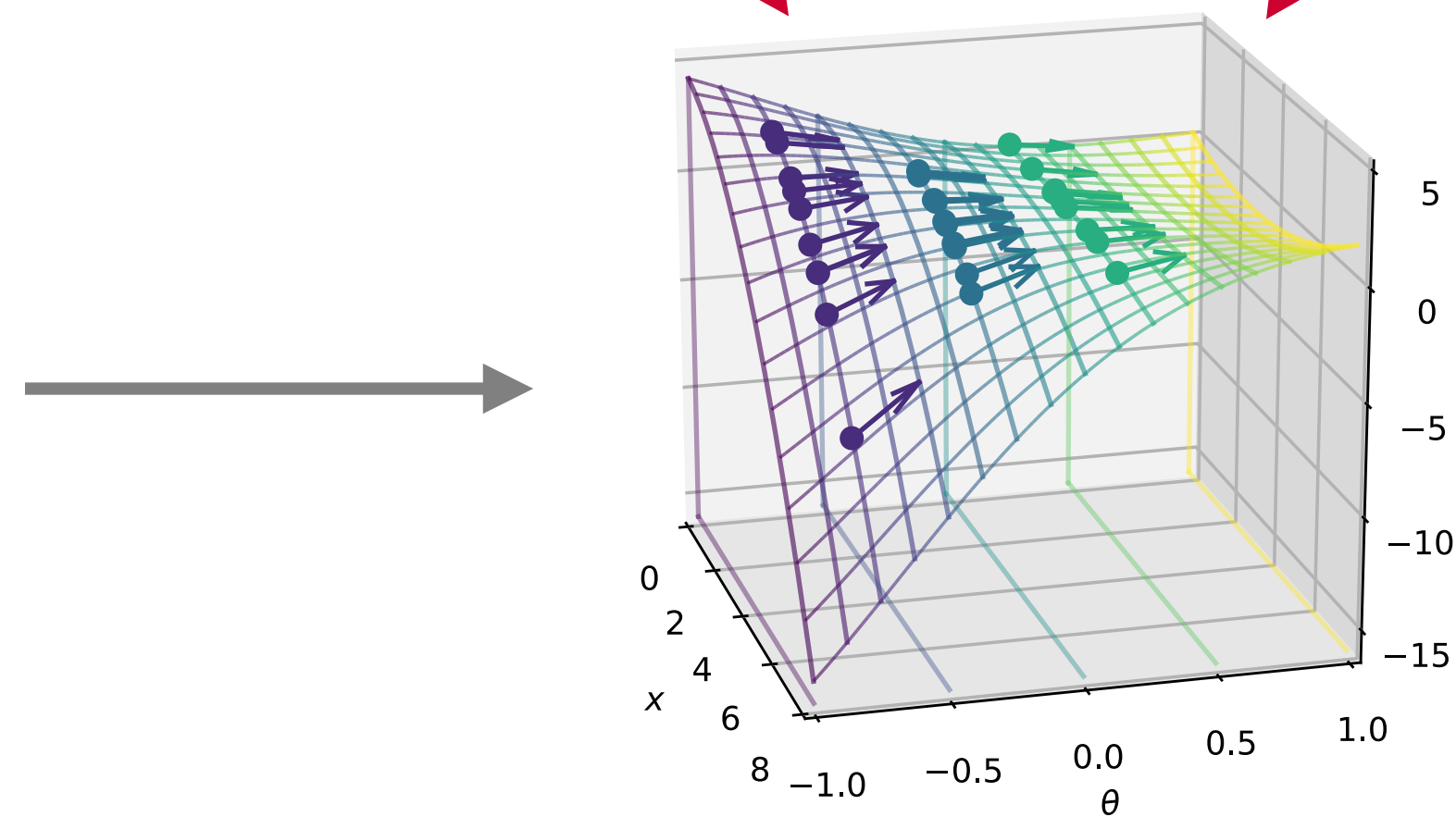


Physics insight:
matrix element information

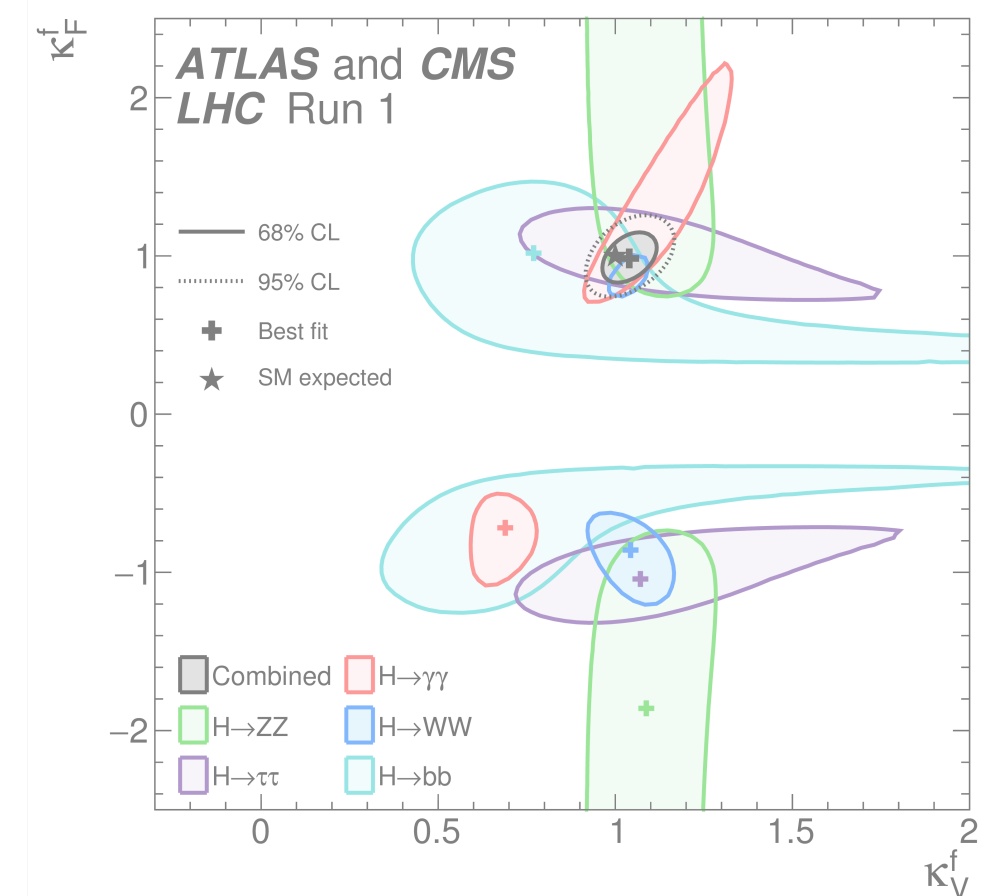
Machine learning



High-dimensional
event data x



Estimator of the
likelihood $p(x|\theta)$



Constraints on
parameters θ

LHC measurements as a likelihood-free inference problem

Modelling particle physics processes

Theory
parameters
 θ



Evolution

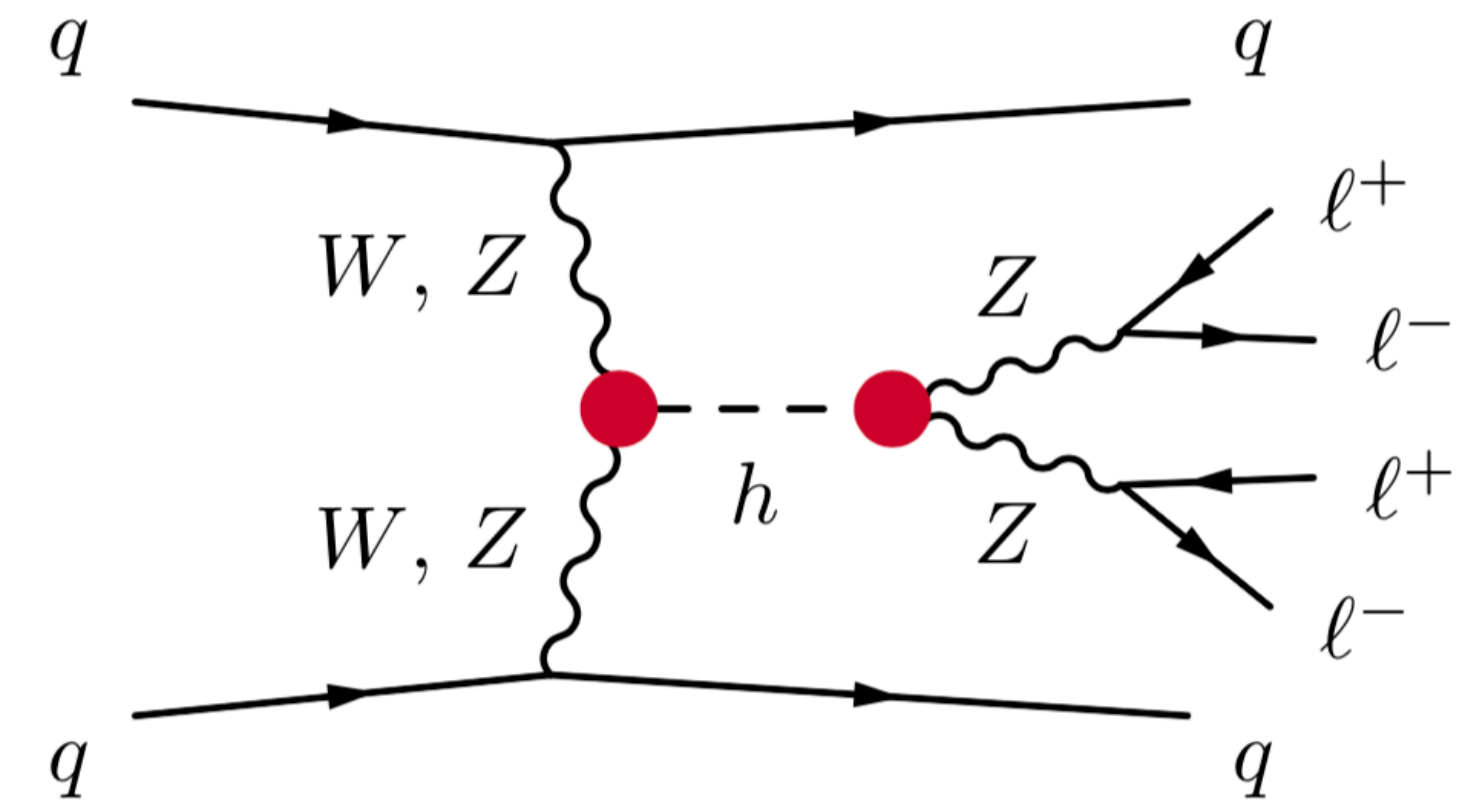
Modelling particle physics processes

Latent variables

Parton-level
momenta

Theory
parameters

z_p ← θ



Evolution

Modelling particle physics processes

Latent variables

Shower
splittings

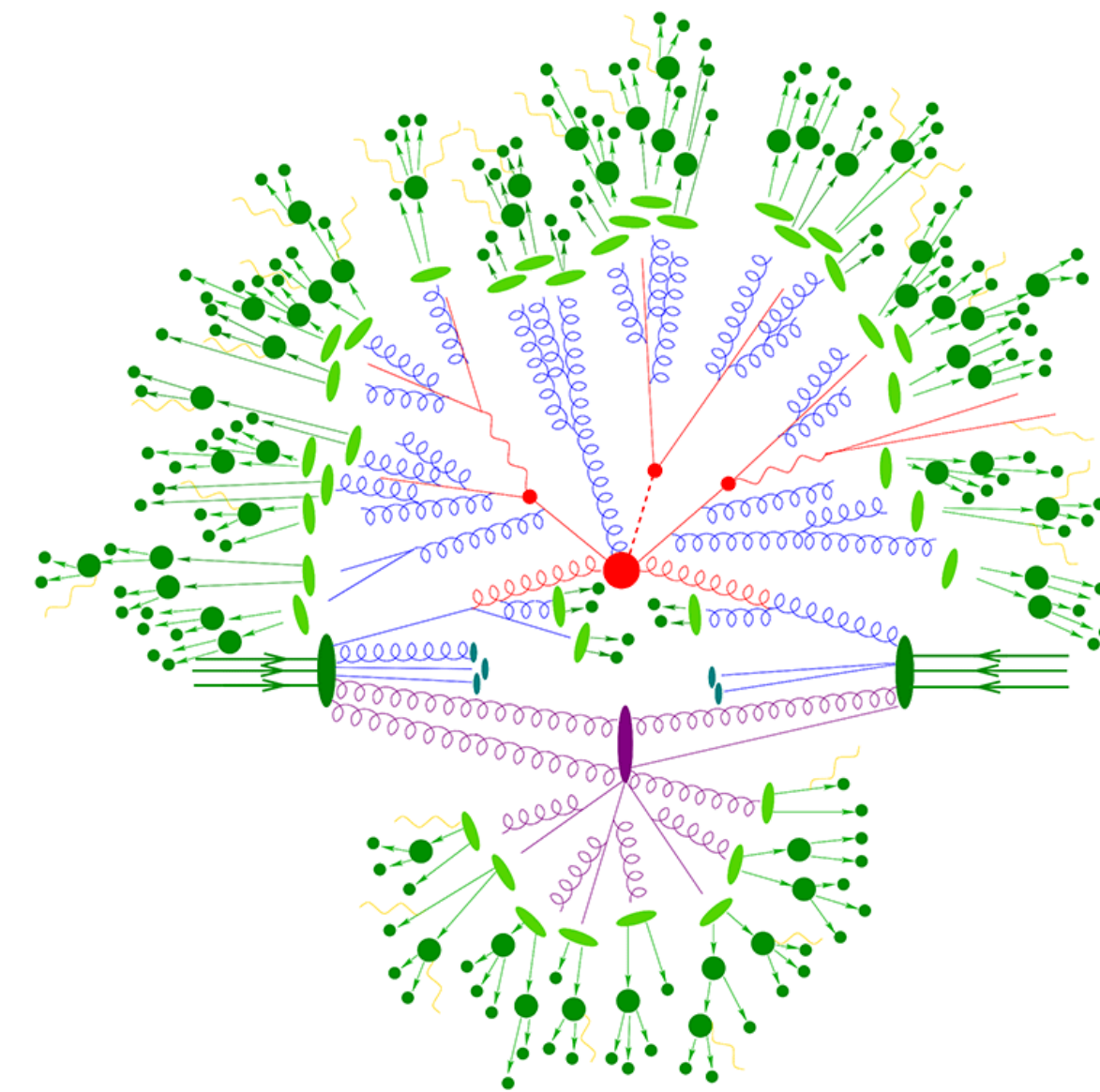
Parton-level
momenta

Theory
parameters

z_s

z_p

θ

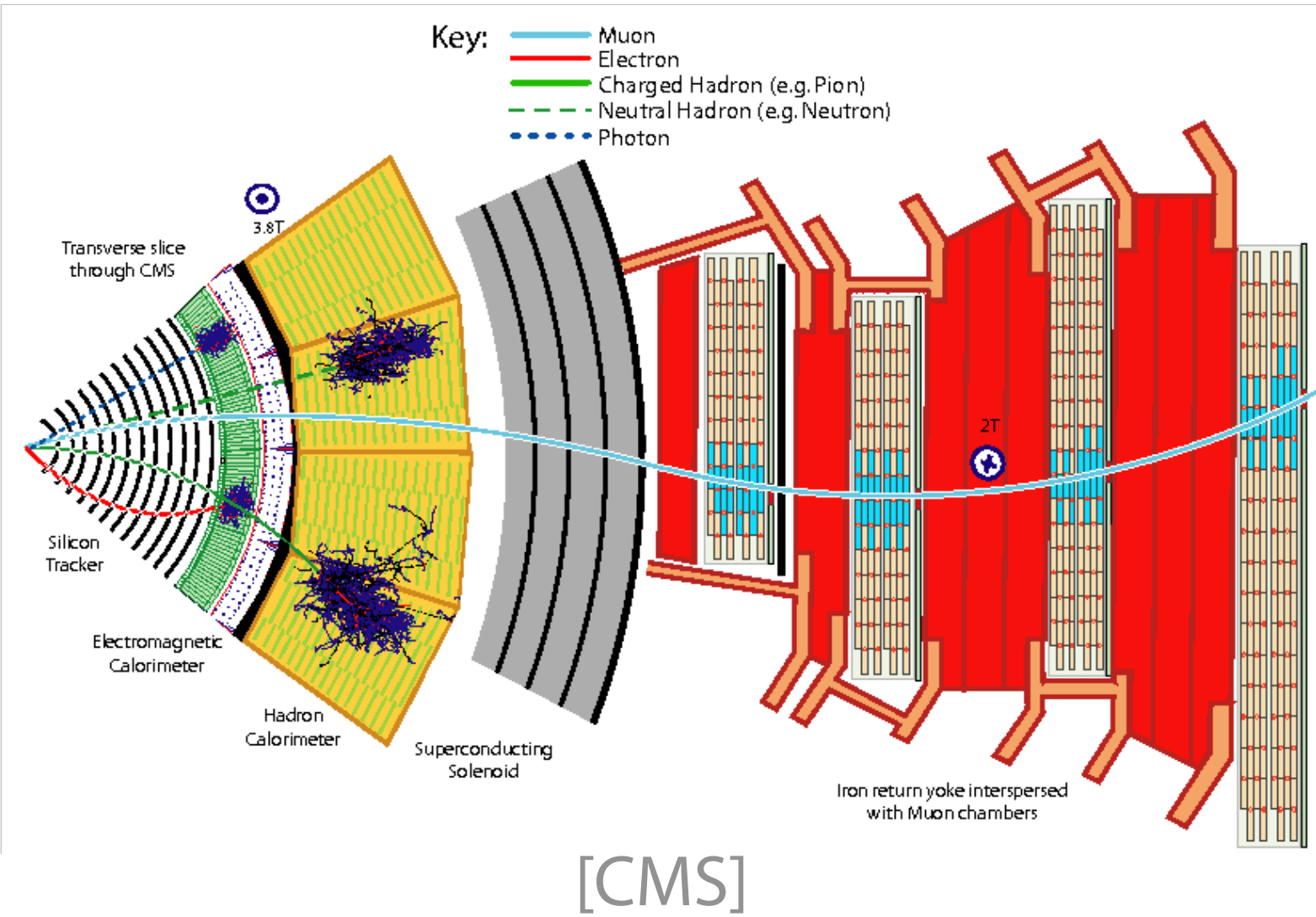
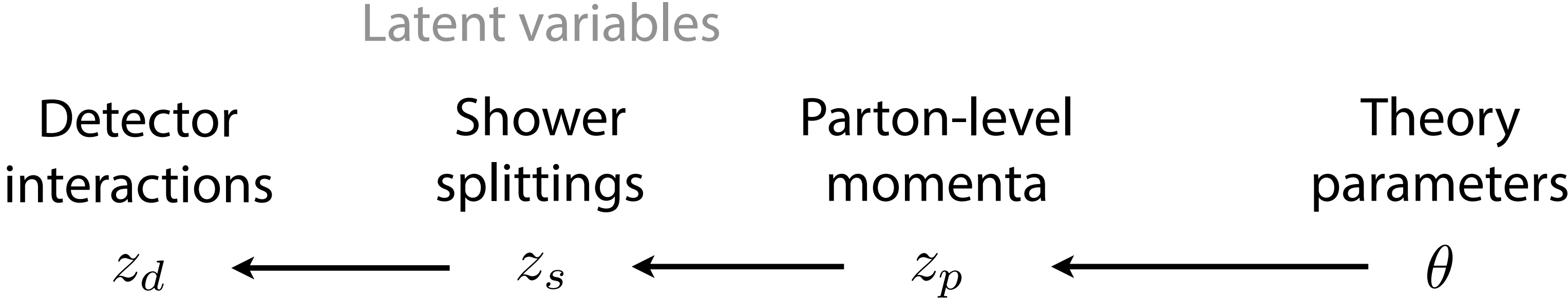


[F. Krauss]



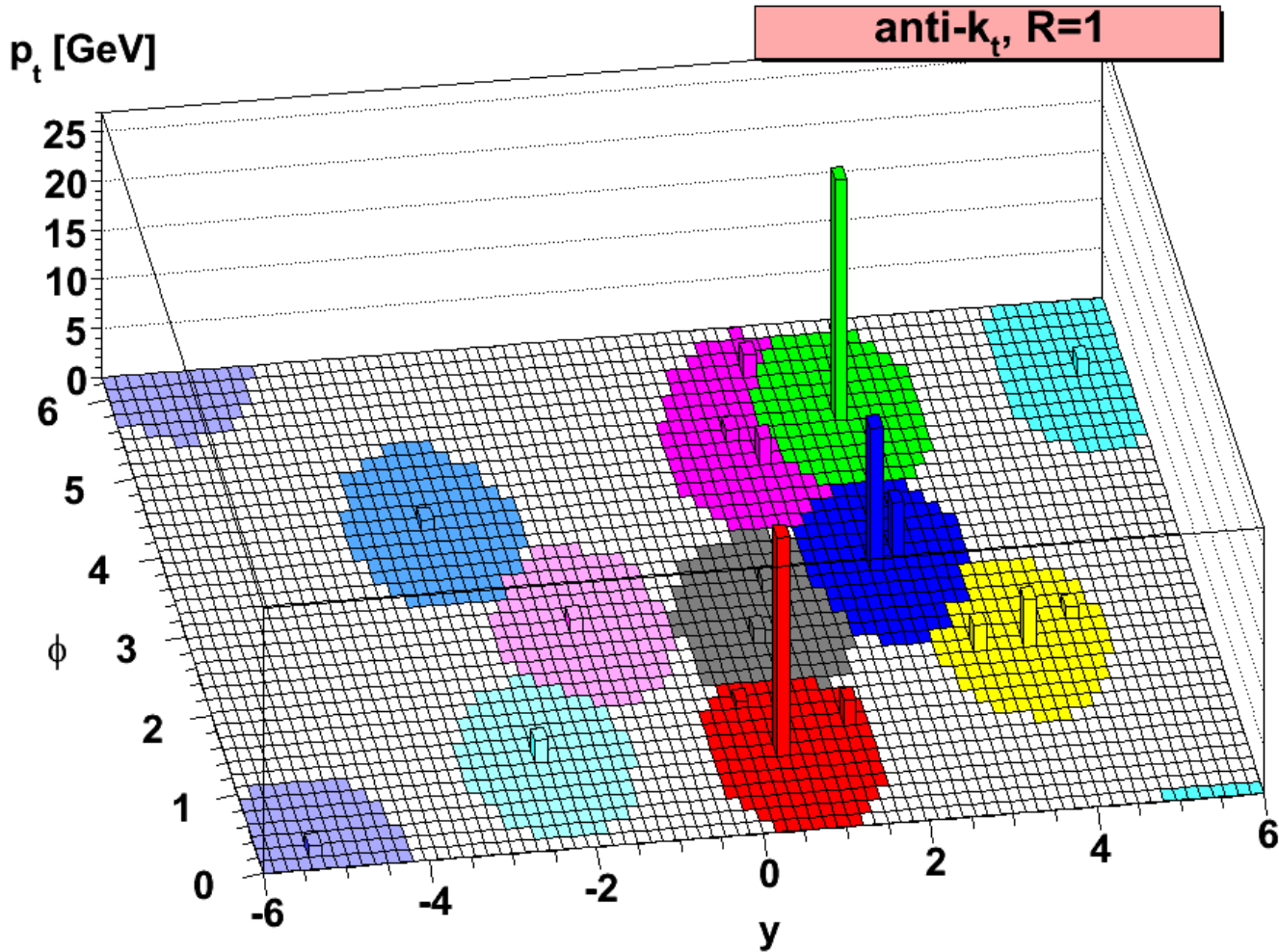
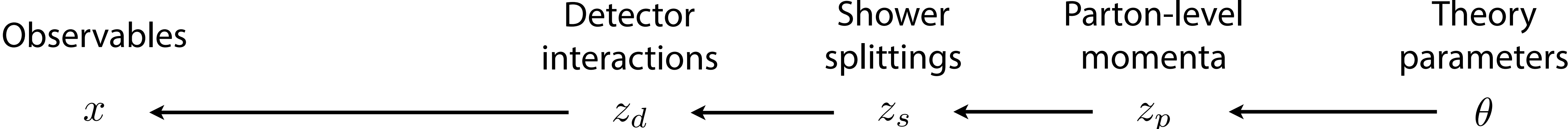
Evolution

Modelling particle physics processes



Modelling particle physics processes

Latent variables

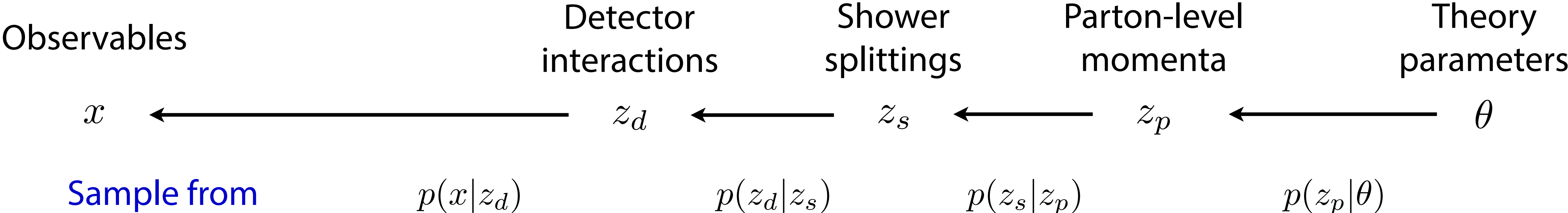


[M. Cacciari, G. Salam, G. Soyez 0802.1189]



Modelling particle physics processes

Latent variables



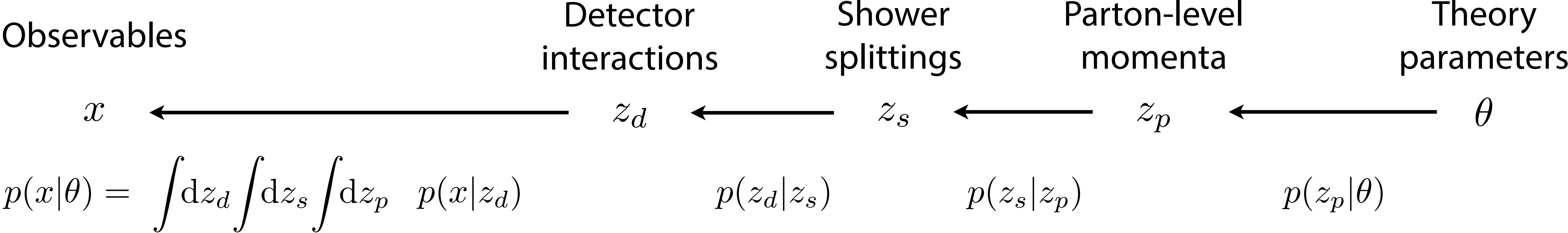
```
MADGRAPH5_aMC@NLO

*
*      * *      *
* * * * 5 * * * *
*      * *      *
*
```

← Prediction (simulation)

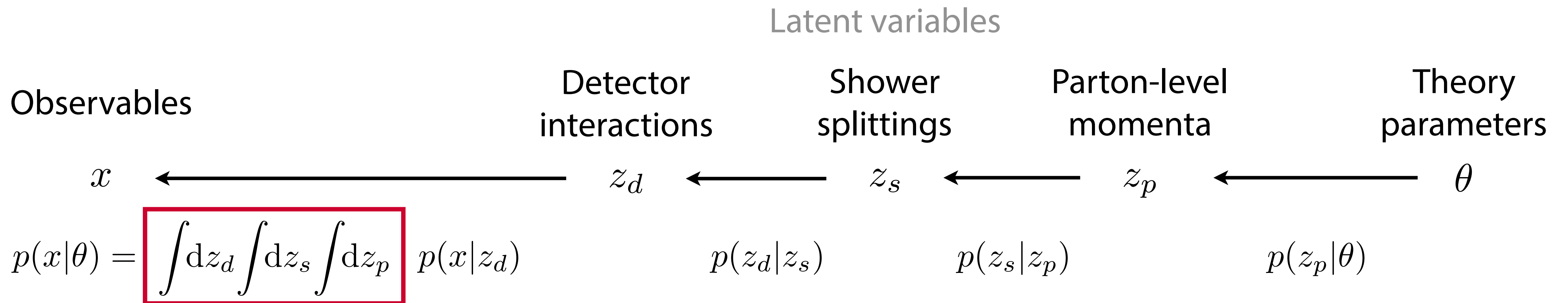
Modelling particle physics processes

Latent variables



Inference →

Modelling particle physics processes

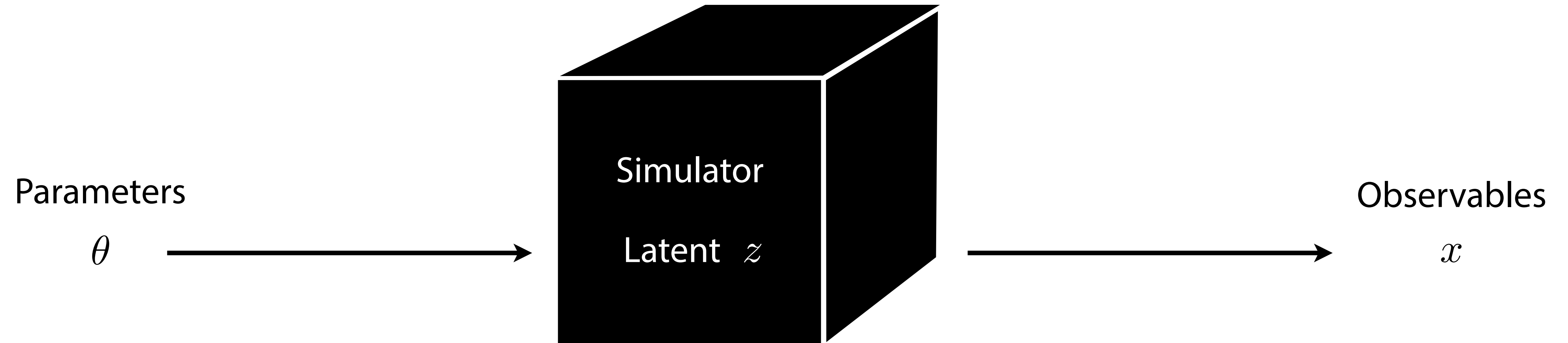


It's infeasible to calculate the integral over this enormous space!

(More subtle: We cannot sample from $p(z|x, \theta)$ efficiently.)

Inference

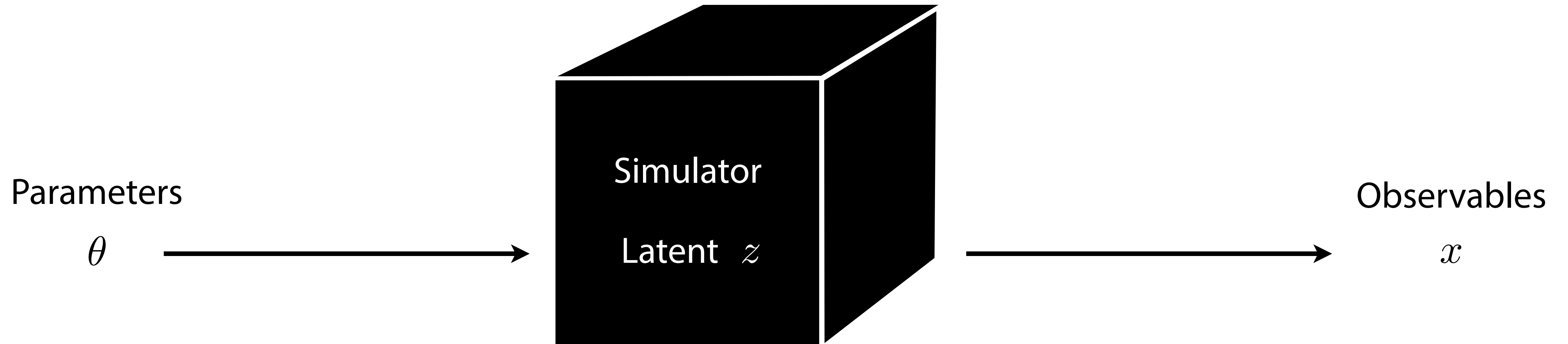
Likelihood-free inference / implicit models



Prediction:

- Well-understood mechanistic model
- Simulator can generate samples

Likelihood-free inference / implicit models



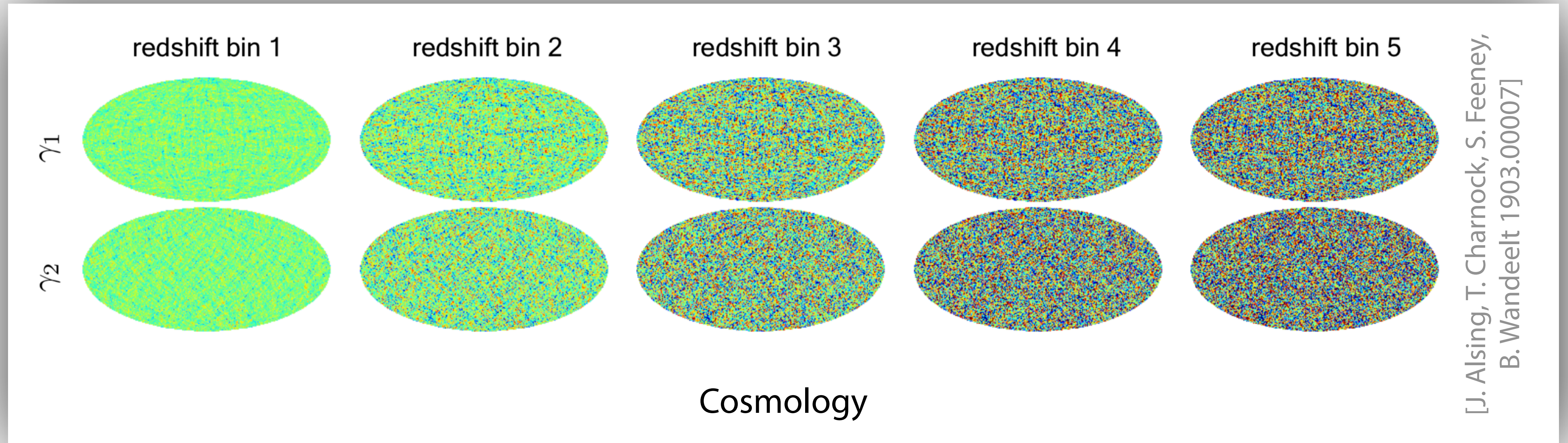
Prediction:

- Well-understood mechanistic model
- Simulator can generate samples

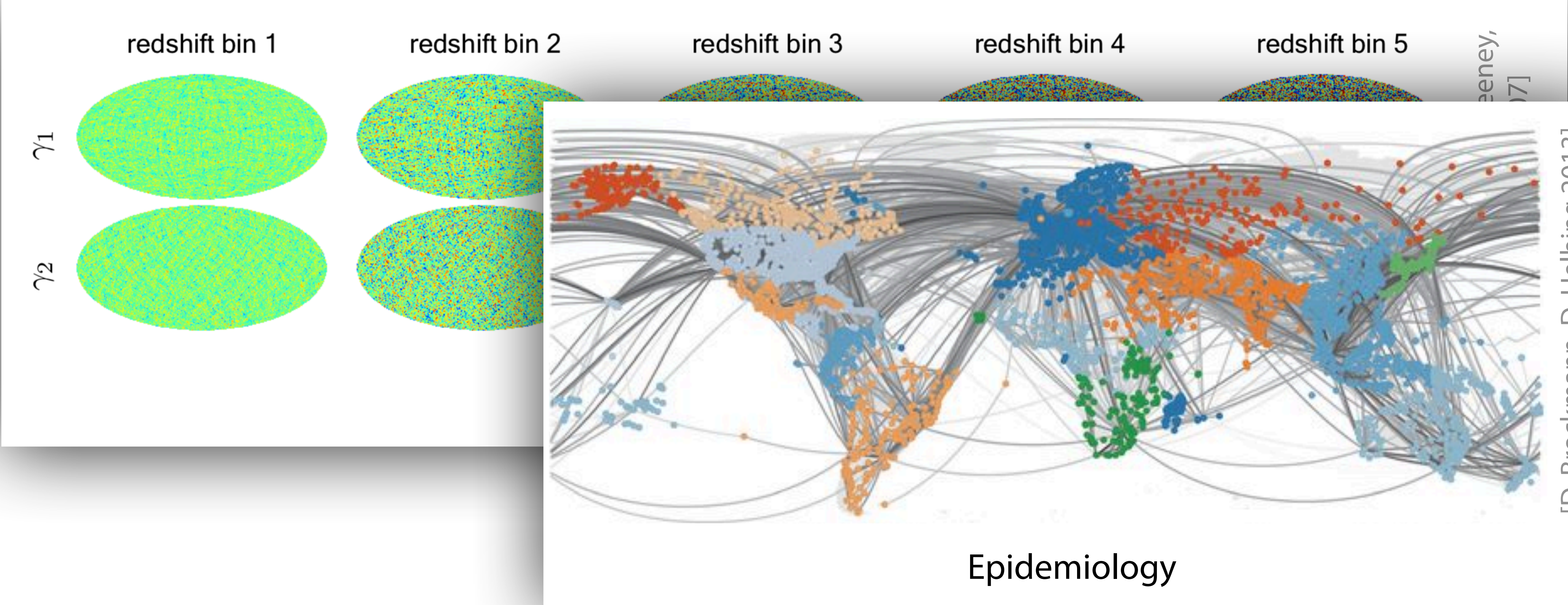
Inference:

- Likelihood function $p(x|\theta)$ is intractable
- Inference based on estimator $\hat{p}(x|\theta)$

A thriving research field

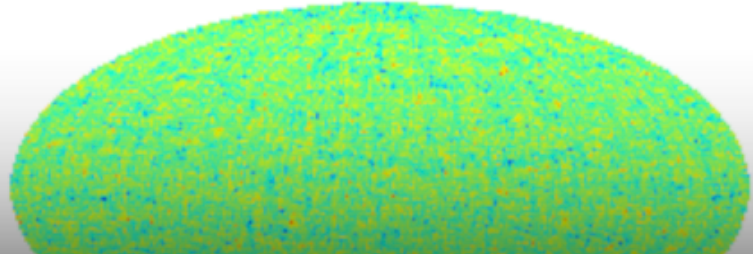


A thriving research field

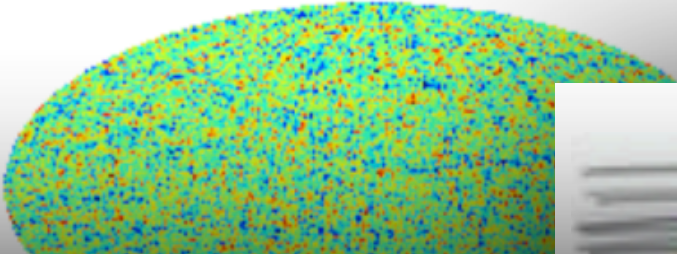


A thriving research field

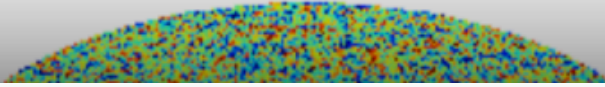
redshift bin 1



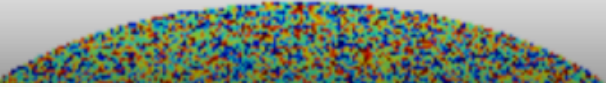
redshift bin 2



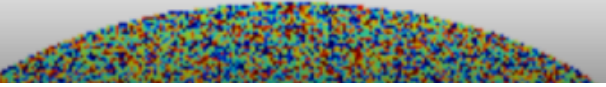
redshift bin 3



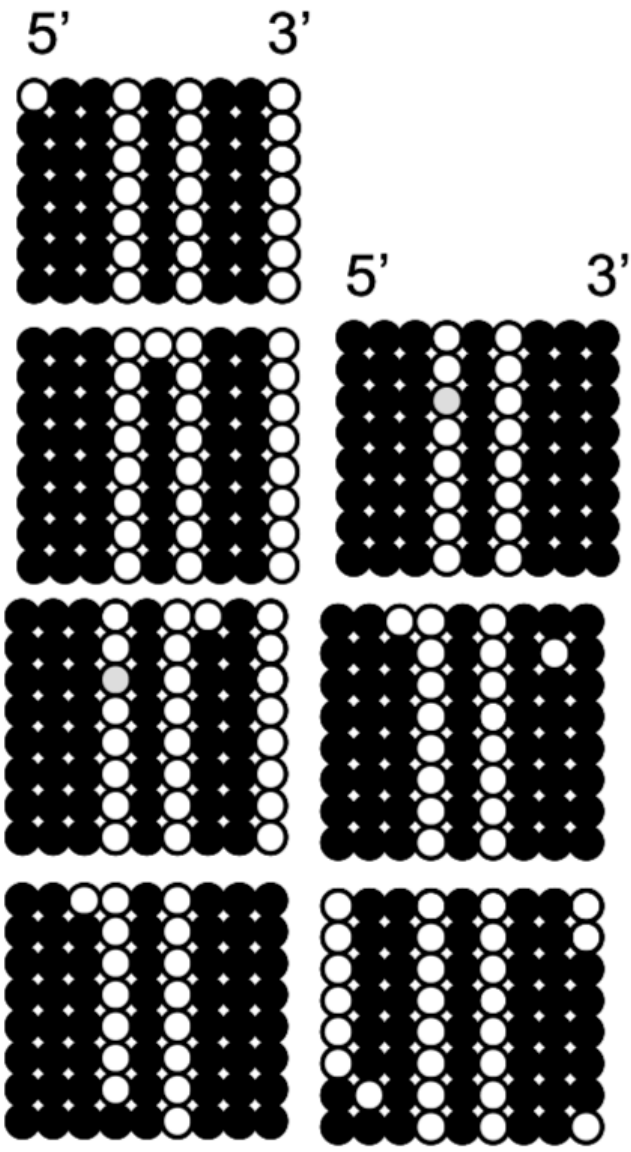
redshift bin 4



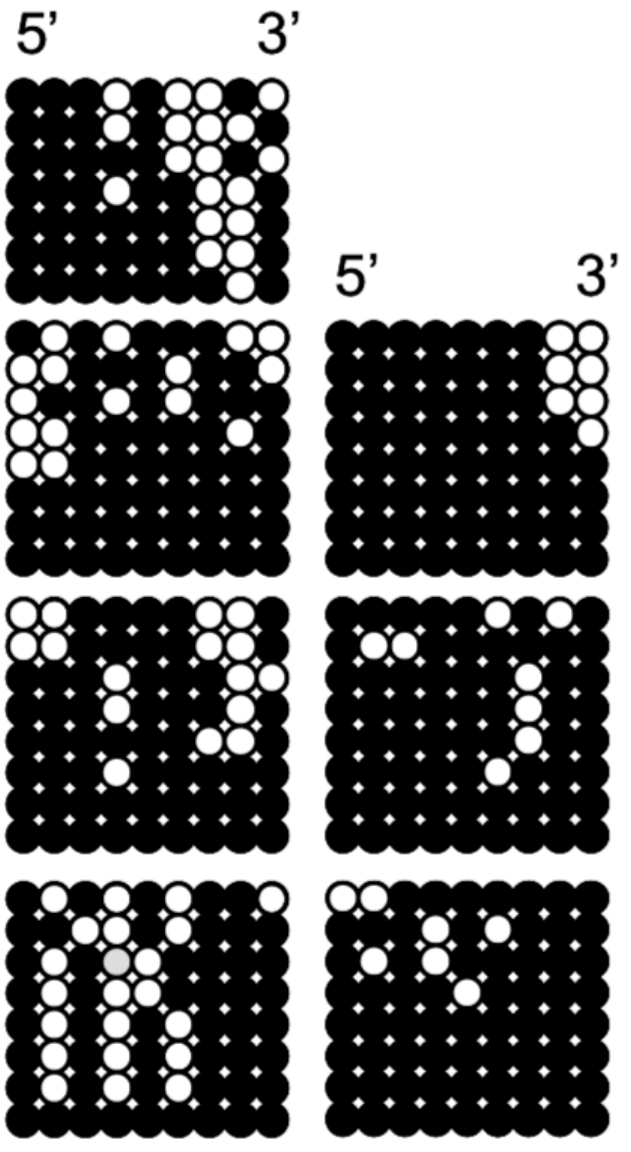
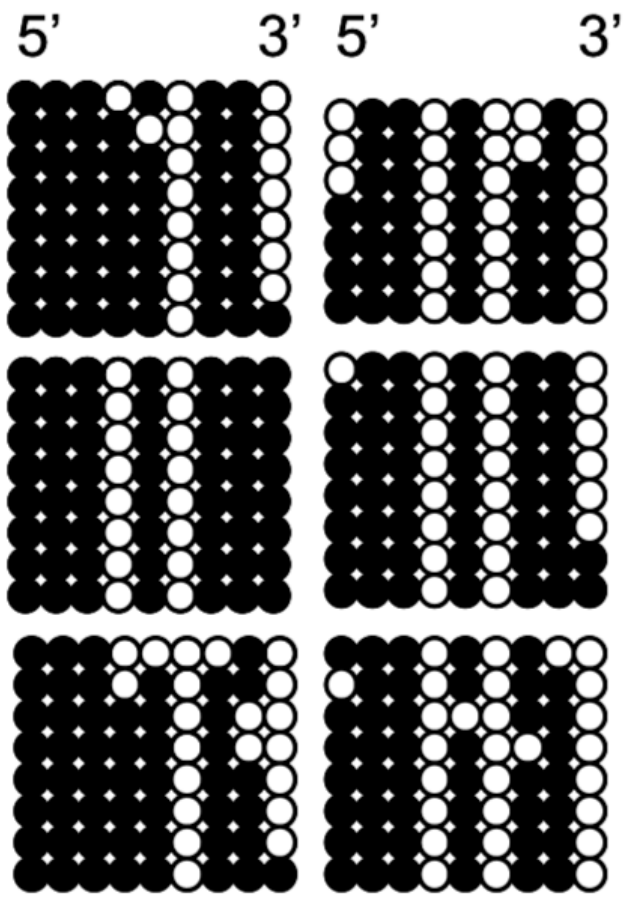
redshift bin 5



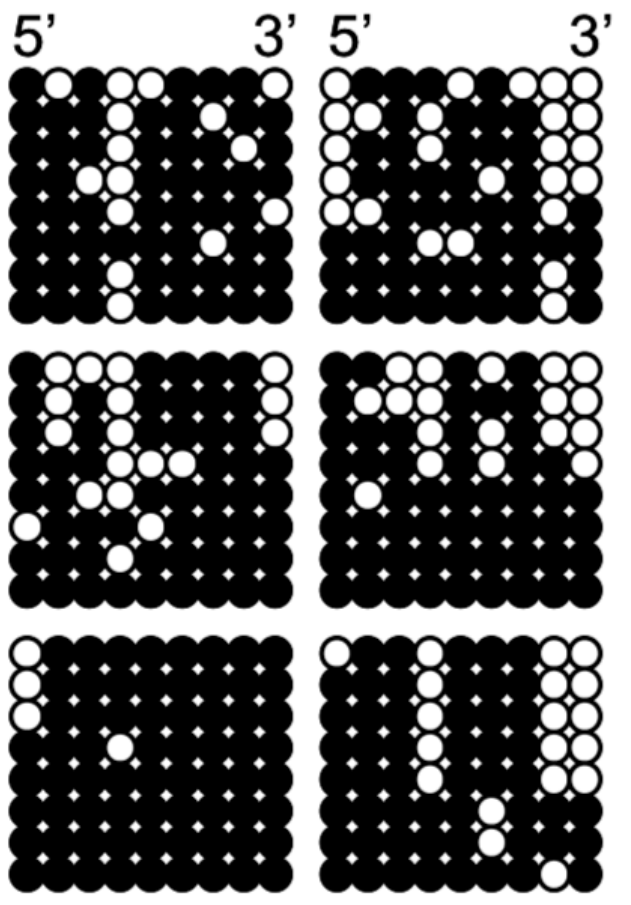
energy, [7]



Left Right
4cm tumor, 79-yr-old patient

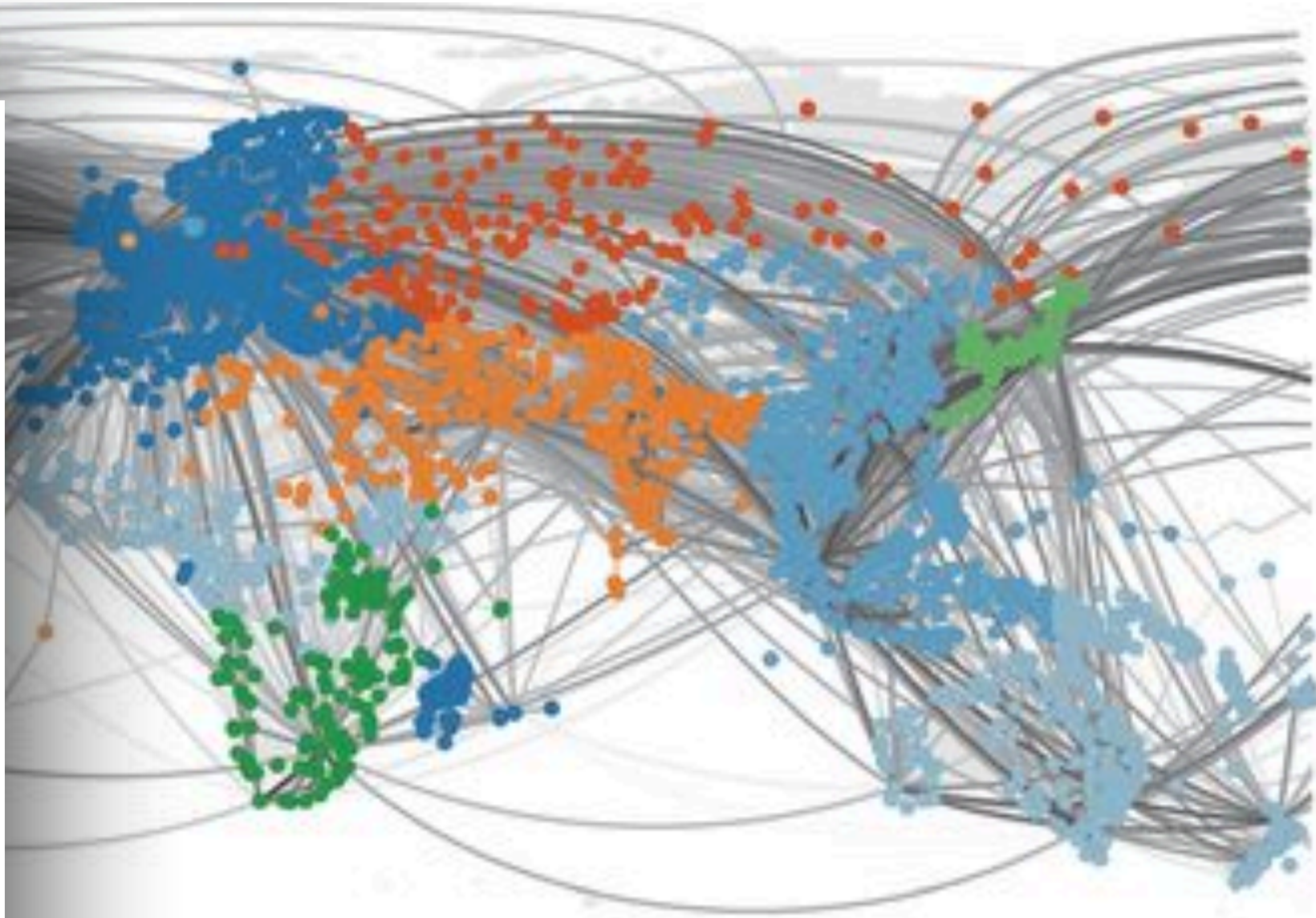


Left Right
8 cm tumor, 83-yr-old patient



Genetics

[K. Siegmund, P. Marjoram, D. Shibata 2008]

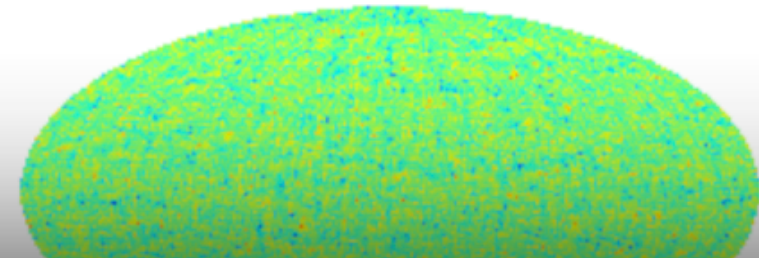


Epidemiology

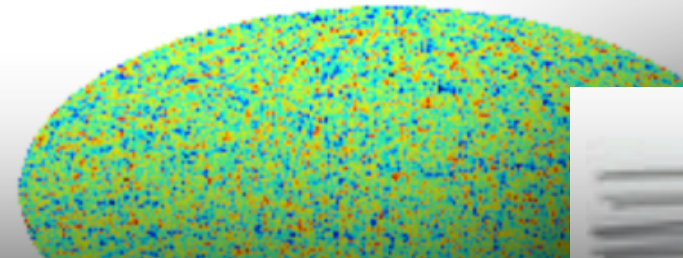
[D. Brockmann, D. Helbing 2013]

A thriving research field

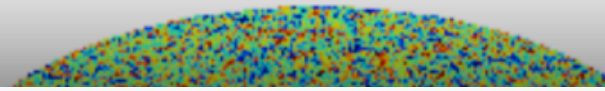
redshift bin 1



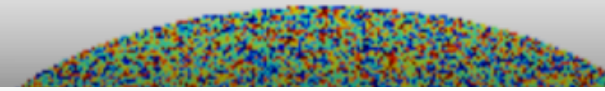
redshift bin 2



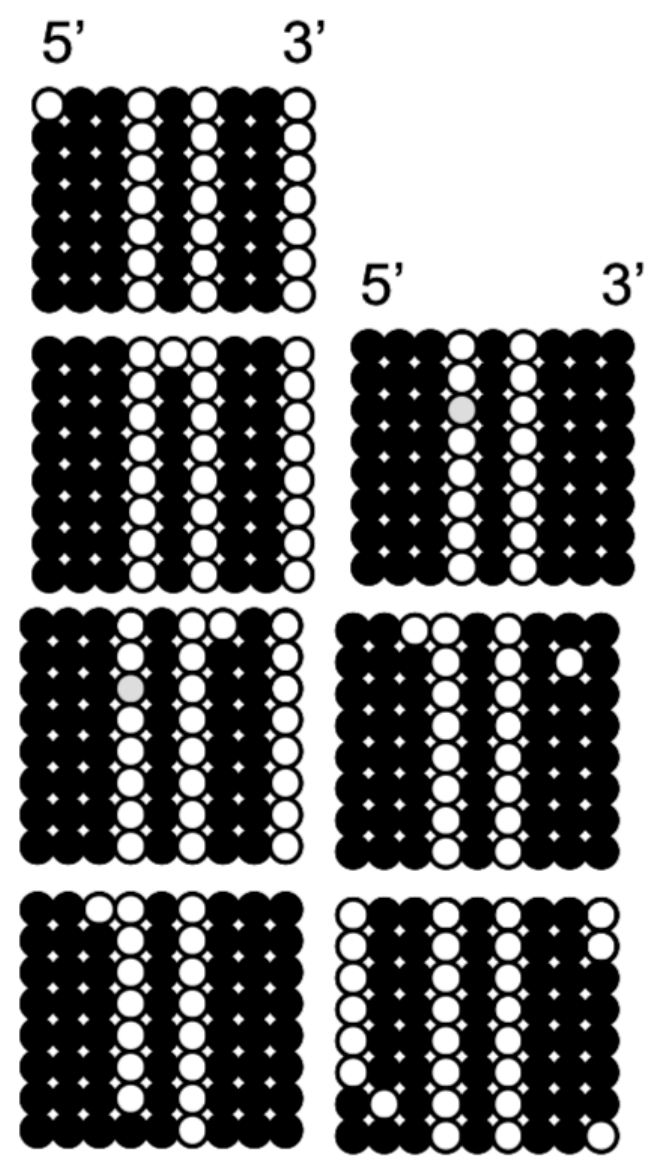
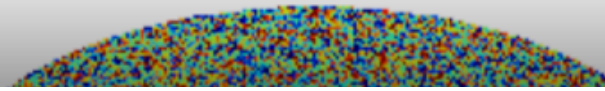
redshift bin 3



redshift bin 4

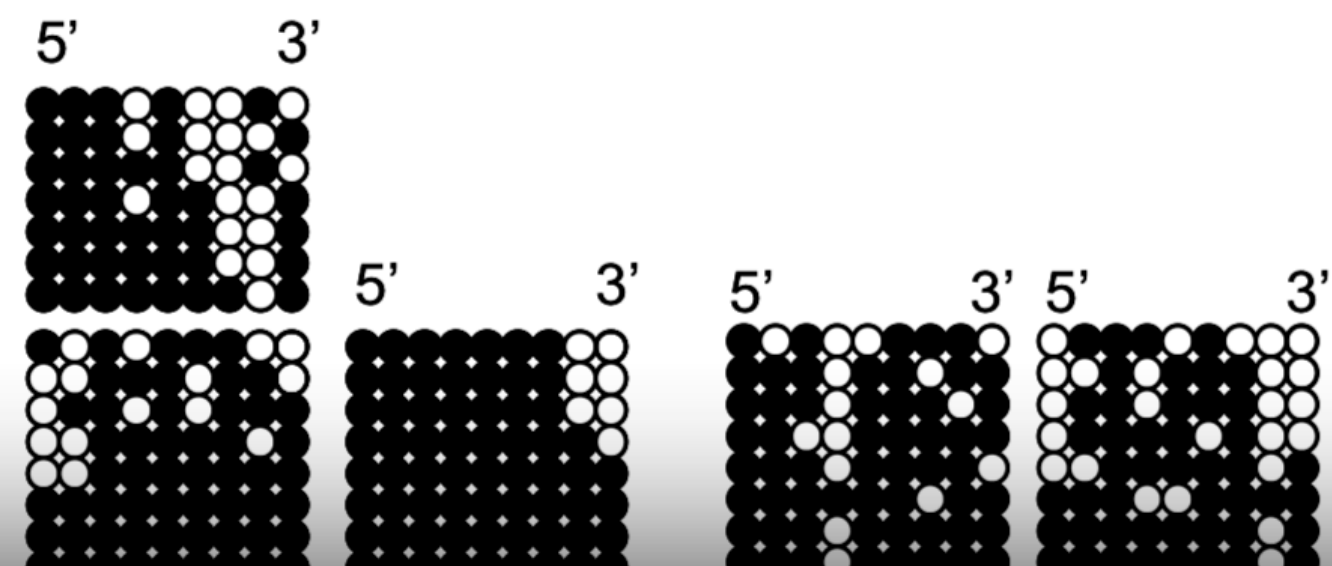
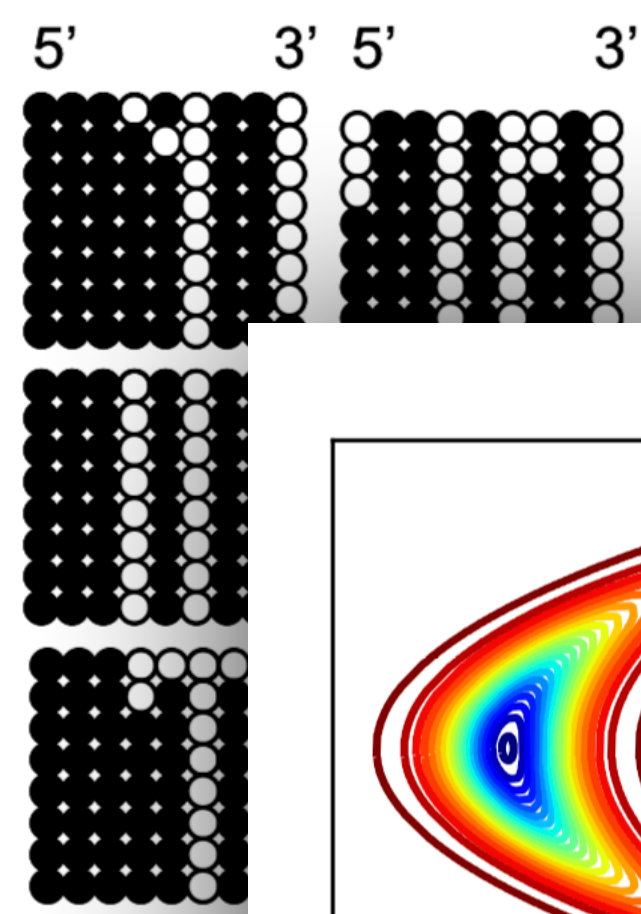


redshift bin 5

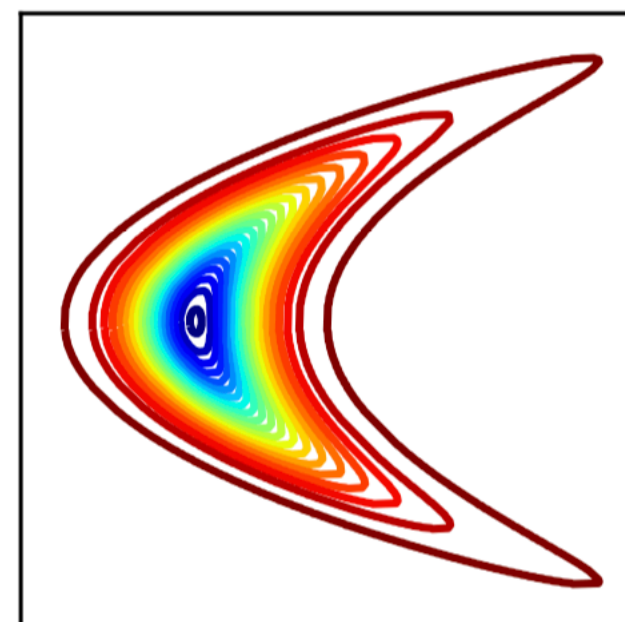


Left

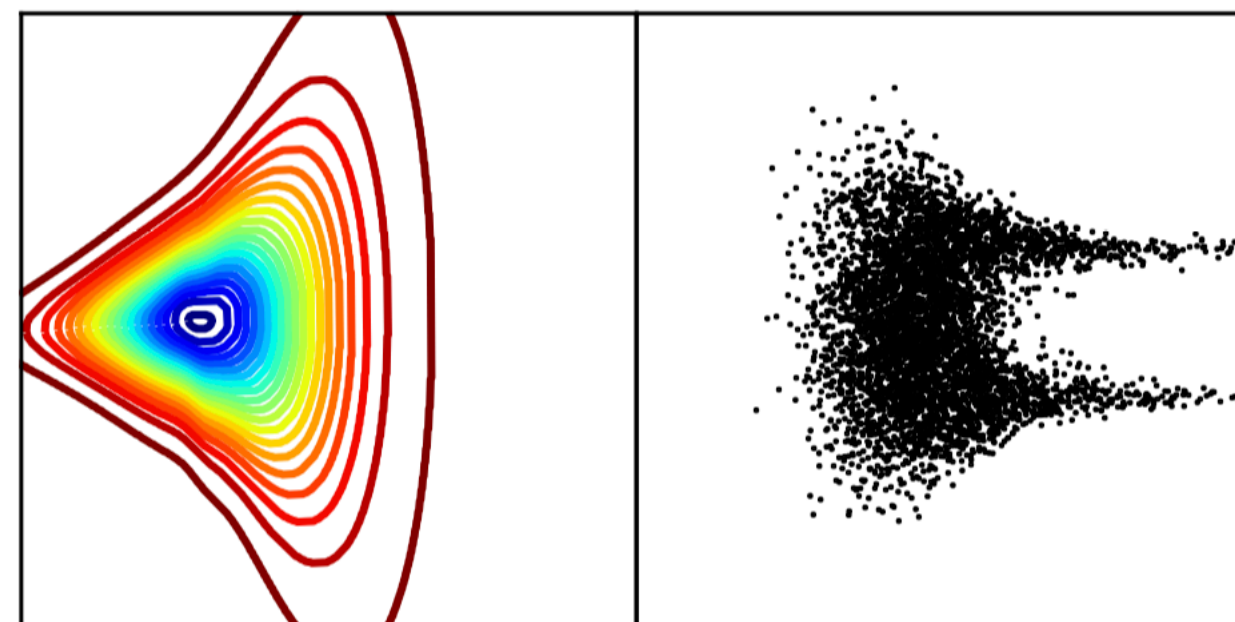
4cm tumor, 79-yr-old pa



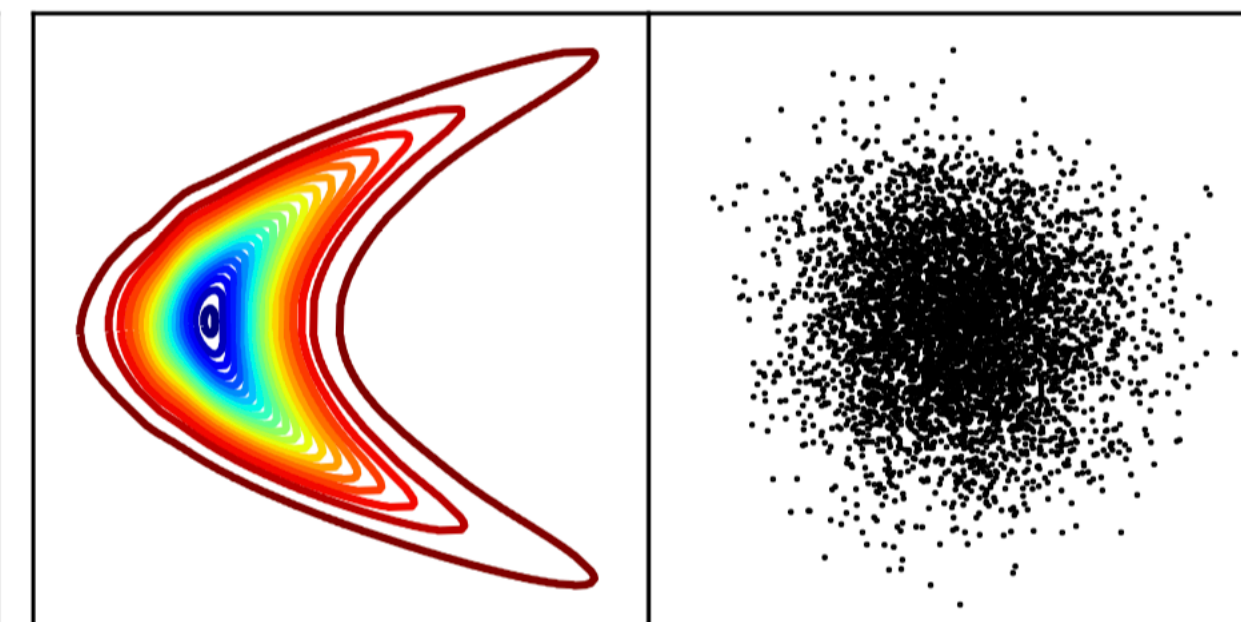
Shibata 2008]



(a) Target density



(b) MADE with Gaussian conditionals



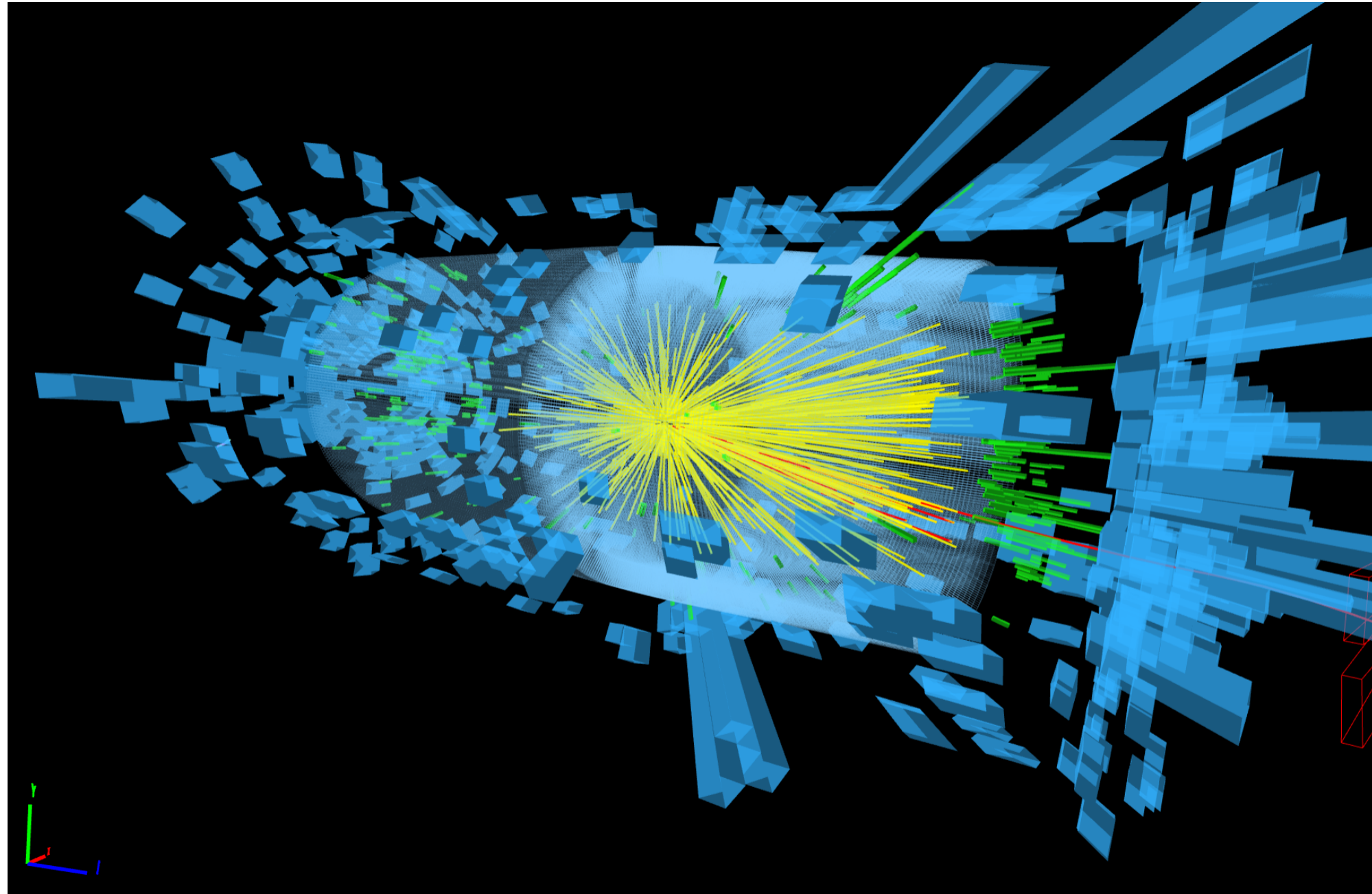
(c) MAF with 5 layers

[G. Papamakarios, T. Pavlakou,
I. Murray 1705.07057]

Computer Science / Machine Learning / Statistics

Why has that not stopped us before?

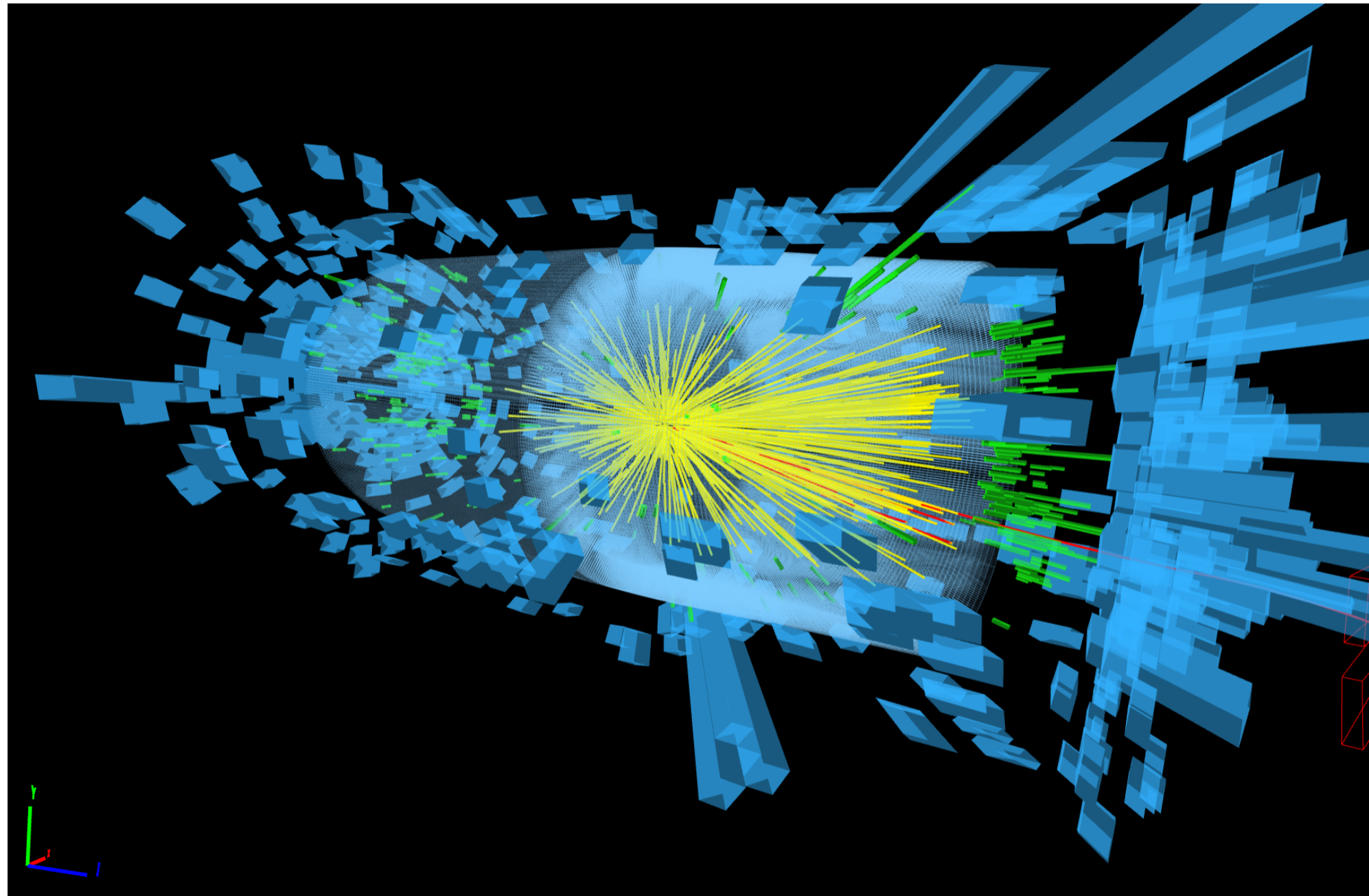
Solve it with summary statistics



High-dimensional event data x

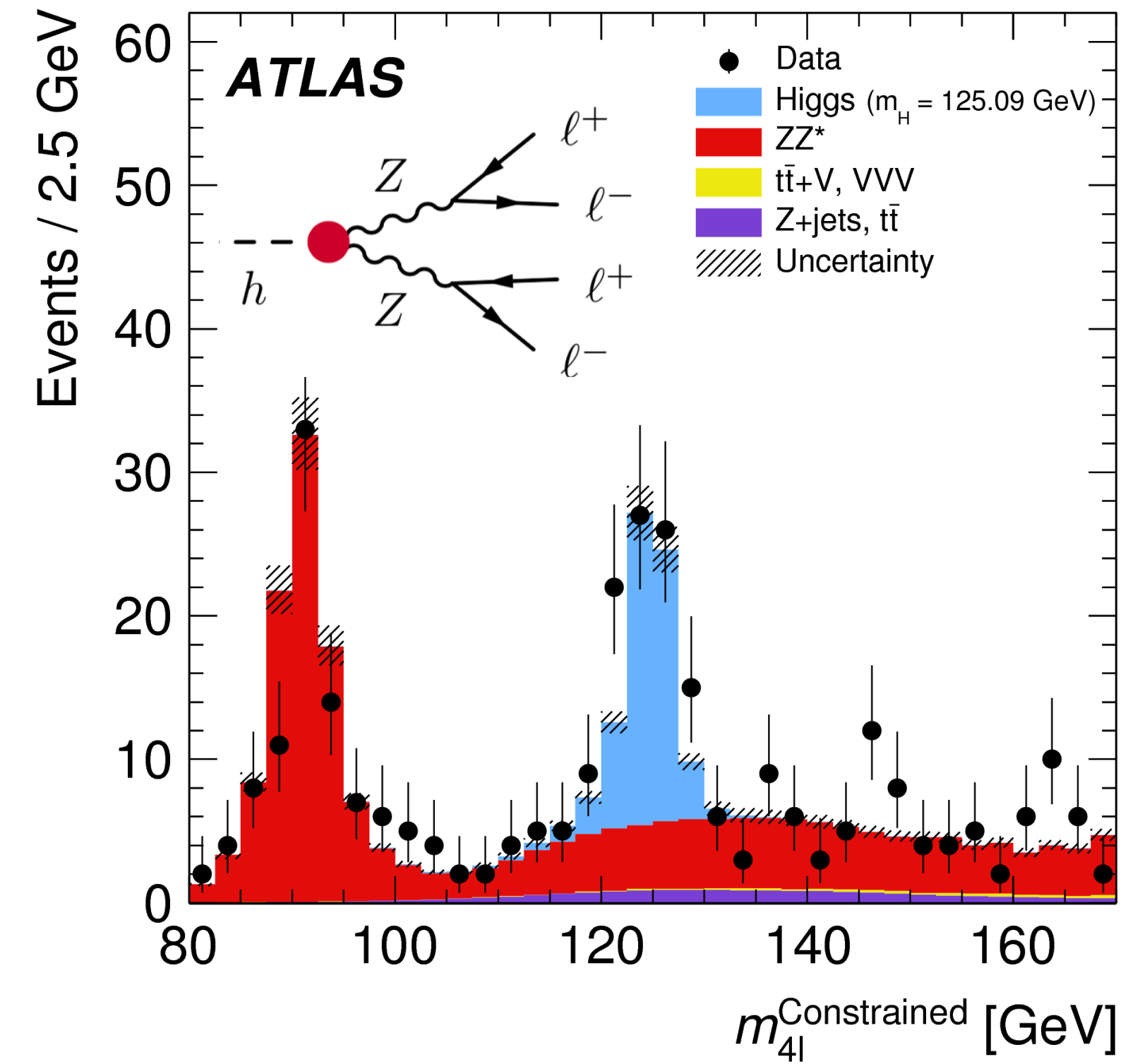
$p(x|\theta)$ cannot be calculated

Solve it with summary statistics



High-dimensional event data x

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



[ATLAS 1712.02304]

One or two summary statistics x'

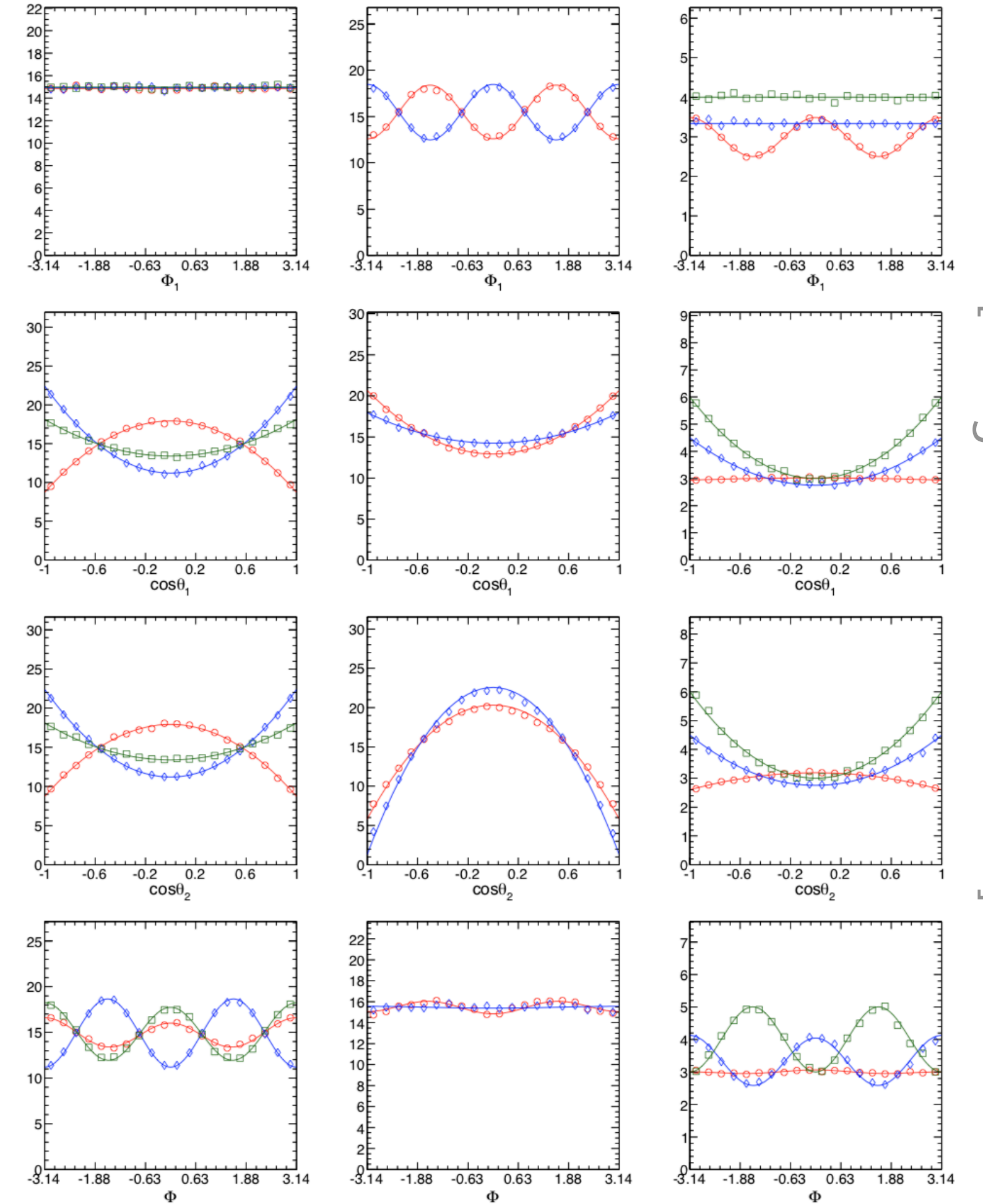
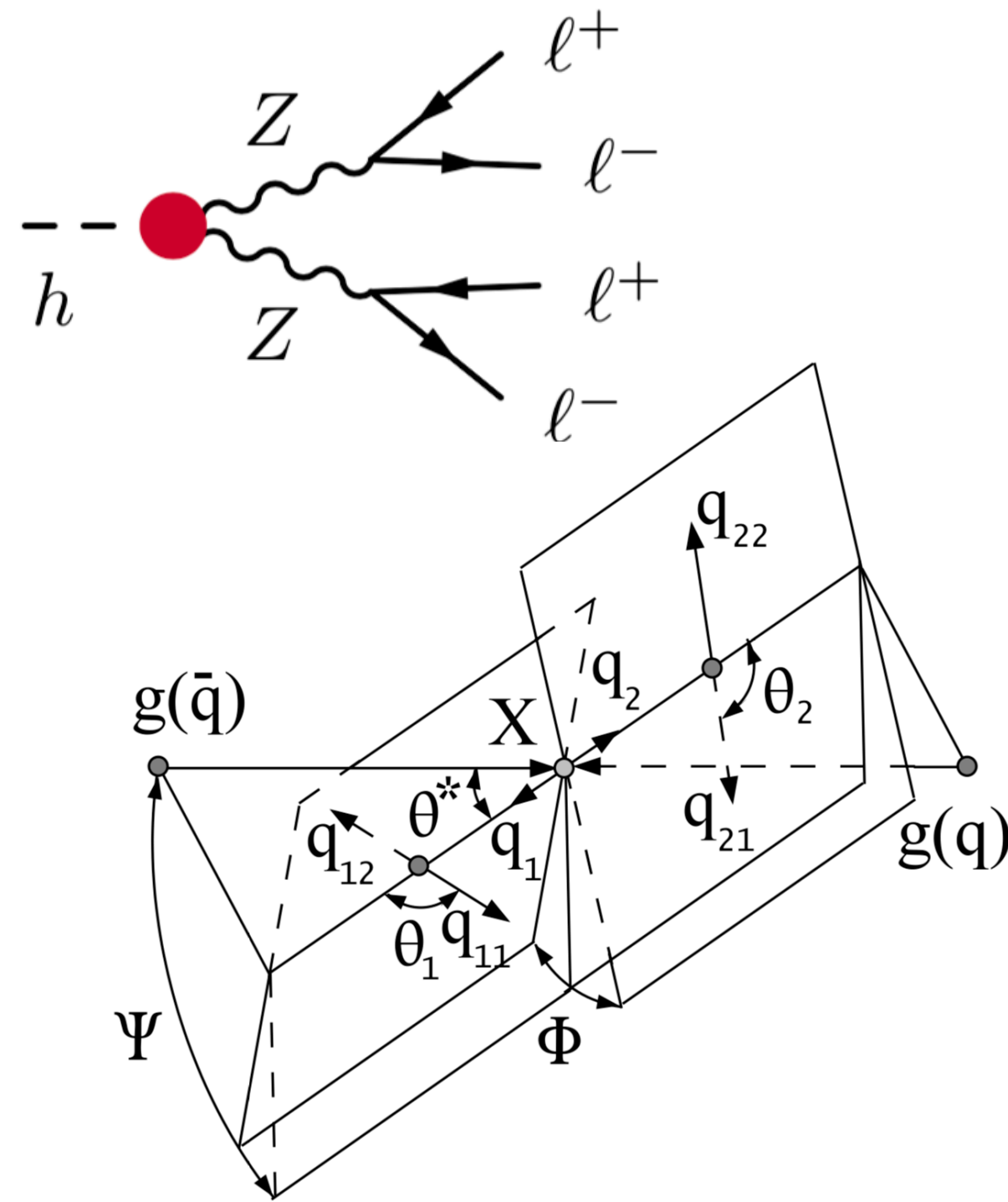
$p(x'|\theta)$ can be estimated
with histograms, KDE, ...

Summary statistics for EFT measurements?

- Choosing summary statistics x' is difficult and problem-dependent
- Often there is no single good standard variable — compressing to any x' loses information!
[JB, K. Cranmer, F. Kling, T. Plehn 1612.05261;
JB, F. Kling, T. Plehn, T. Tait 1712.02350]
- Ideally: analyze high-dimensional x including all correlations
("fully differential cross section")

Summary statistics for EFT measurements?

- Choosing summary statistics x' is difficult and problem-dependent
- Often there is no single good standard variable — compressing to any x' loses information!
 [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261;
 JB, F. Kling, T. Plehn, T. Tait 1712.02350]
- Ideally: analyze high-dimensional x including all correlations (“fully differential cross section”)

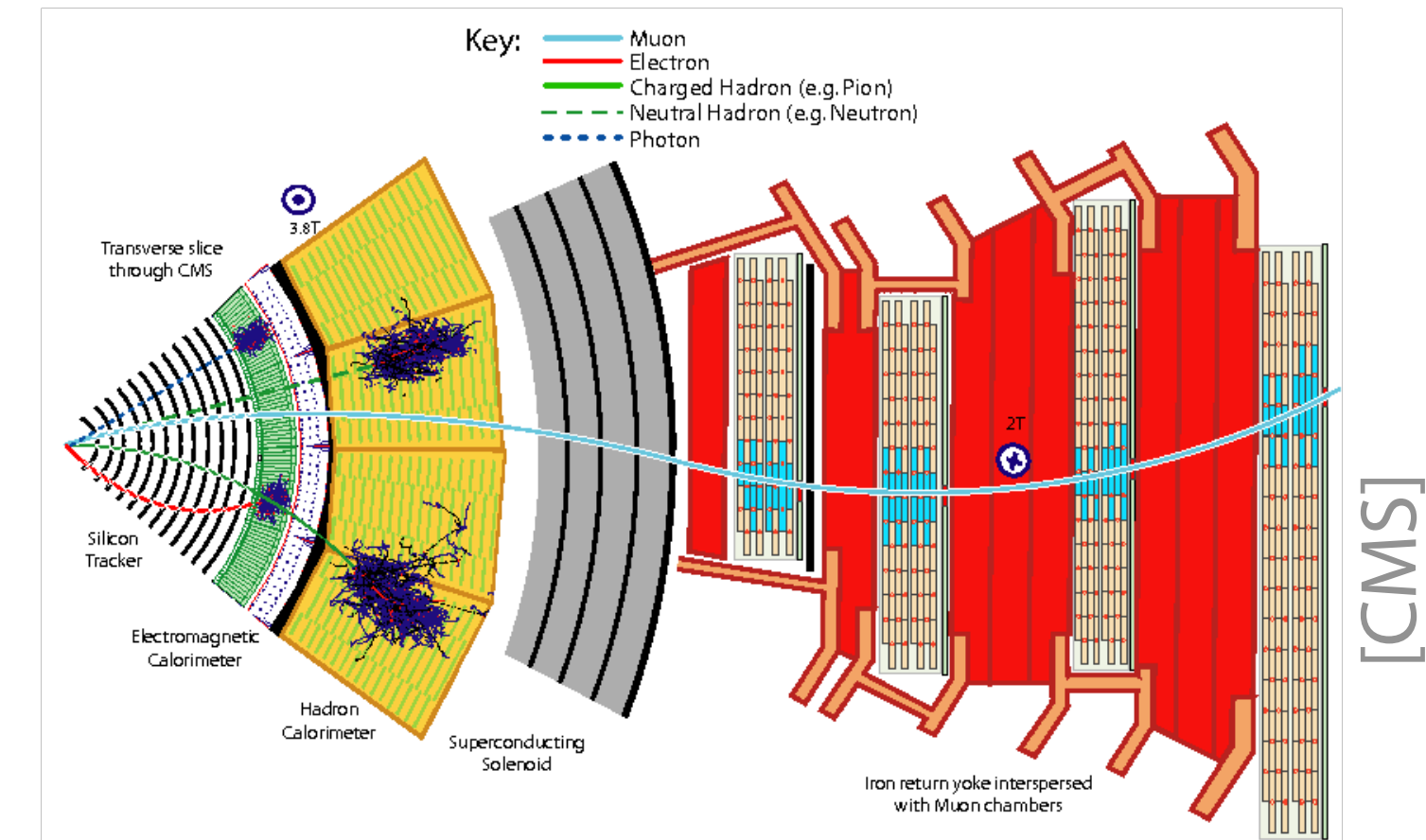


[Bolognesi et al. 1208.4018]

Solve it by approximating the integral

- Problem: high-dim. integral over **shower / detector trajectories**

$$p(x|\theta) = \int dz_d \int dz_s \int dz_p p(x|z_d) p(z_d|z_s) p(z_s|z_p) p(z_p|\theta)$$



Solve it by approximating the integral

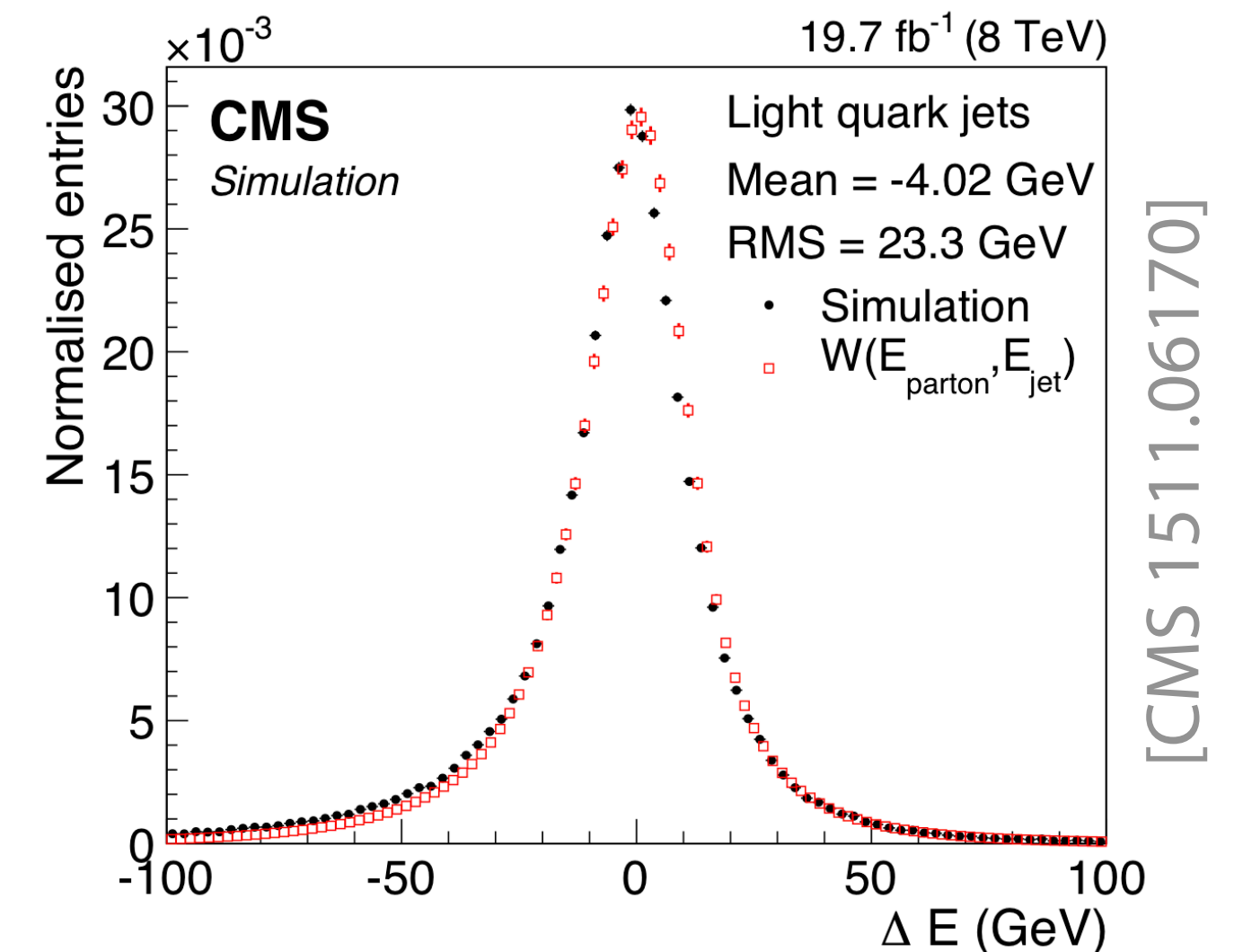
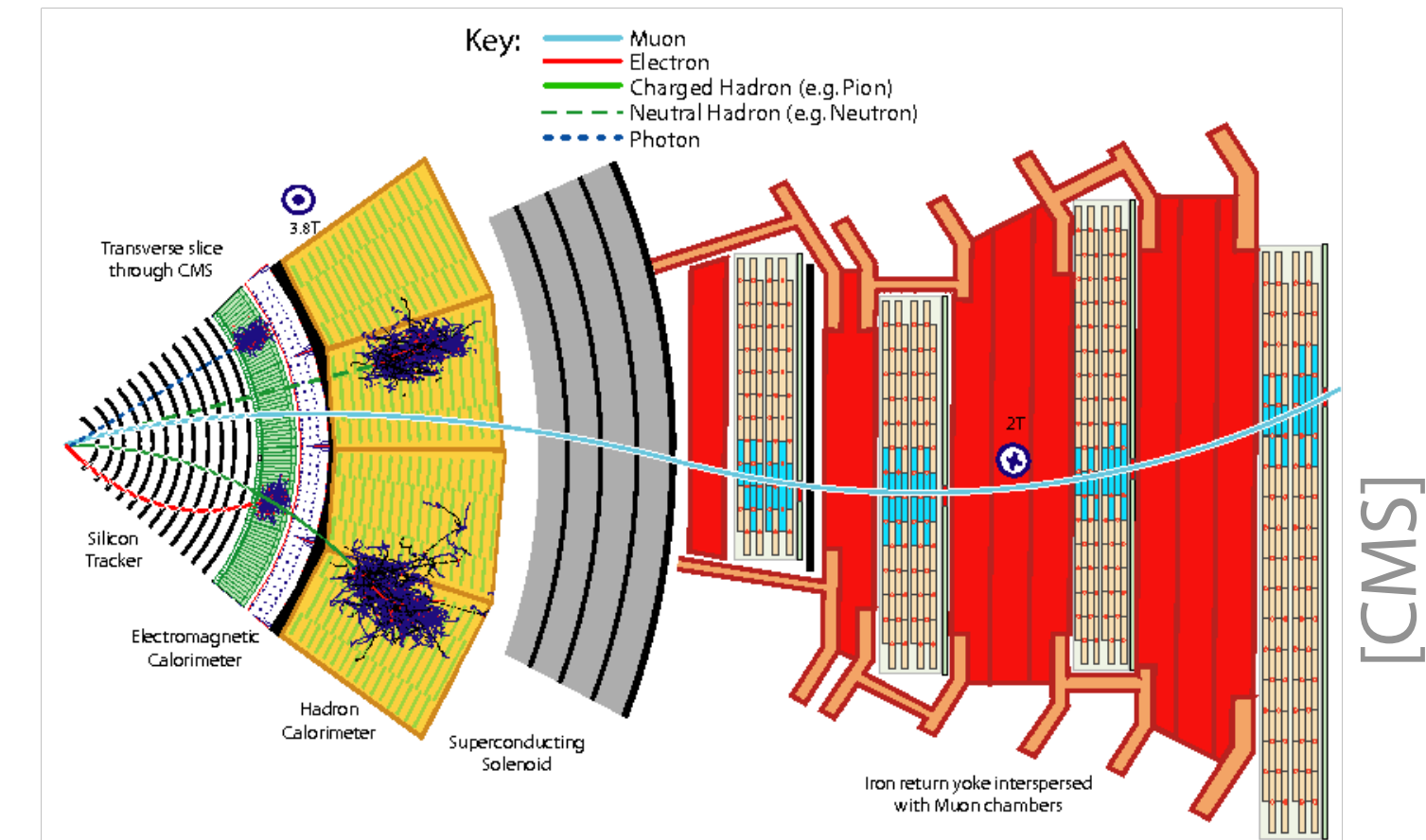
- Problem: high-dim. integral over **shower / detector trajectories**

$$p(x|\theta) = \int dz_d \int dz_s \int dz_p p(x|z_d) p(z_d|z_s) p(z_s|z_p) p(z_p|\theta)$$

- Matrix Element Method (MEM): [K. Kondo 1988]

- approximate **shower + detector effects** into **transfer function** $\hat{p}(x|z_p)$
- explicitly calculate remaining integral

$$\hat{p}(x|\theta) = \int dz_p \hat{p}(x|z_p) p(z_p|\theta)$$



Solve it by approximating the integral

- Problem: high-dim. integral over **shower / detector trajectories**

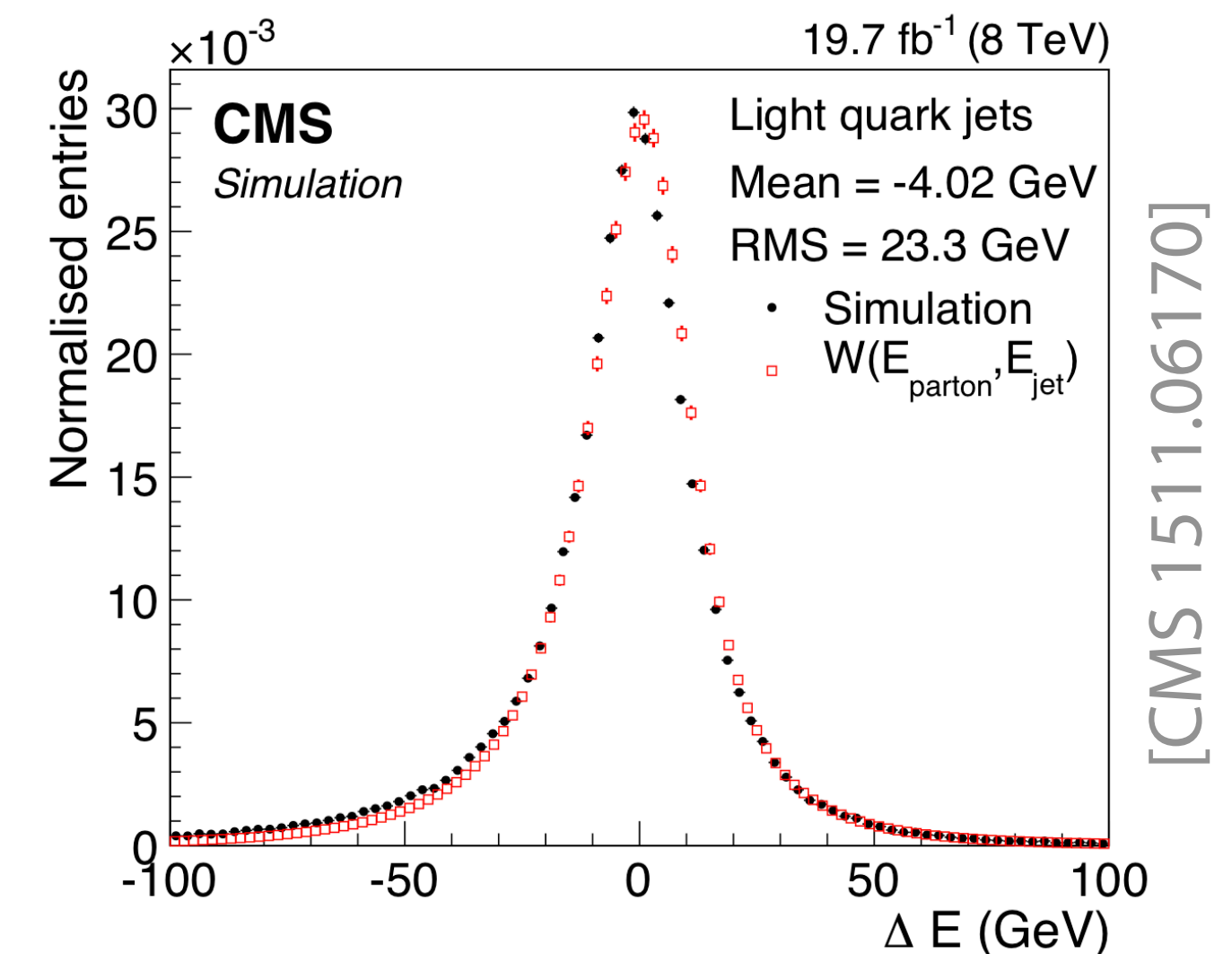
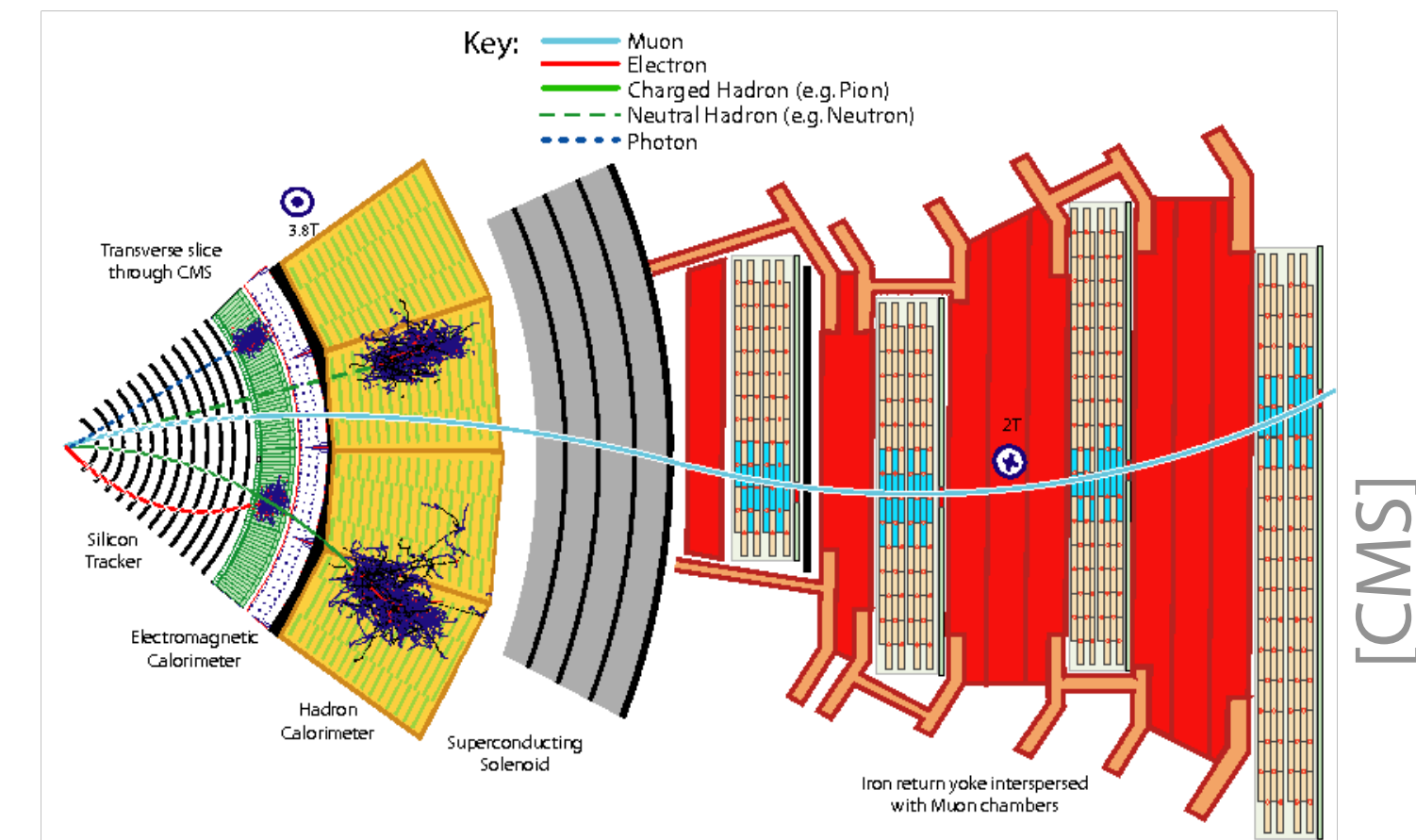
$$p(x|\theta) = \int dz_d \int dz_s \int dz_p p(x|z_d) p(z_d|z_s) p(z_s|z_p) p(z_p|\theta)$$

- Matrix Element Method (MEM): [K. Kondo 1988]
 - approximate **shower + detector effects** into **transfer function** $\hat{p}(x|z_p)$
 - explicitly calculate remaining integral

$$\hat{p}(x|\theta) = \int dz_p \hat{p}(x|z_p) p(z_p|\theta)$$

⇒ Uses matrix-element information, no summary statistics necessary, but:

- ad-hoc transfer functions (what about extra radiation?)
- evaluation still requires calculating an expensive integral



Solve it with machine learning classifiers

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

Solve it with machine learning classifiers

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

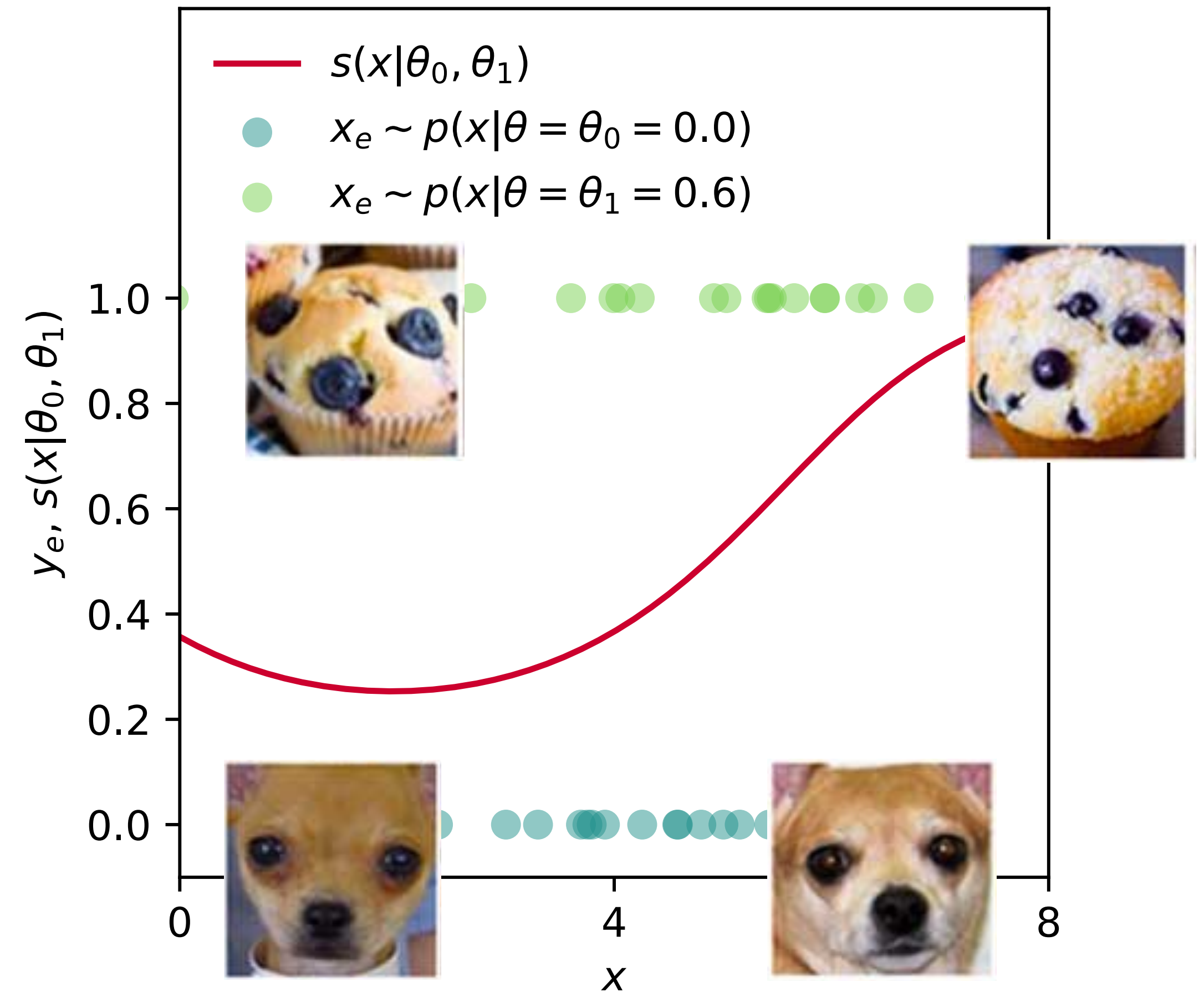


[M. Yao, idea for analogy: K. Cranmer]

Solve it with machine learning classifiers

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

- Train neural network (BDT, ...) to tell $x \sim p(x|\theta_0)$ from $x \sim p(x|\theta_1)$
- Classifier output $\hat{s}(x)$ is closer to 0 for θ_0 -like events (closer to 1 for θ_1 -like events)



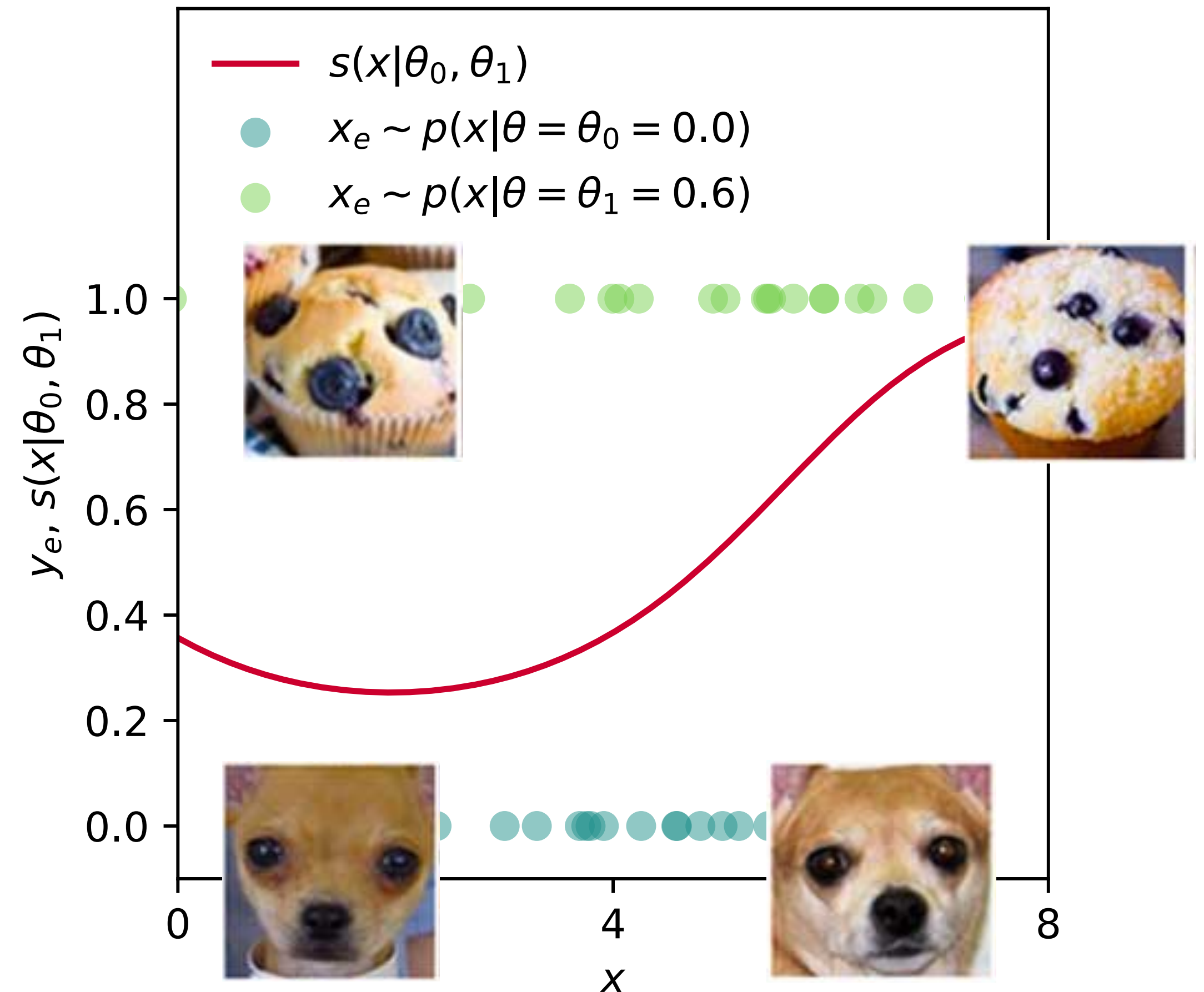
Solve it with machine learning classifiers

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

- Train neural network (BDT, ...) to tell $x \sim p(x|\theta_0)$ from $x \sim p(x|\theta_1)$
- Classifier output $\hat{s}(x)$ is closer to 0 for θ_0 -like events (closer to 1 for θ_1 -like events)
- CARL: Transform classifier output function $\hat{s}(x)$ into estimator for the likelihood ratio

$$r(x) \equiv p(x|\theta_0)/p(x|\theta_1)$$

(calibrate estimator with histograms of NN output)

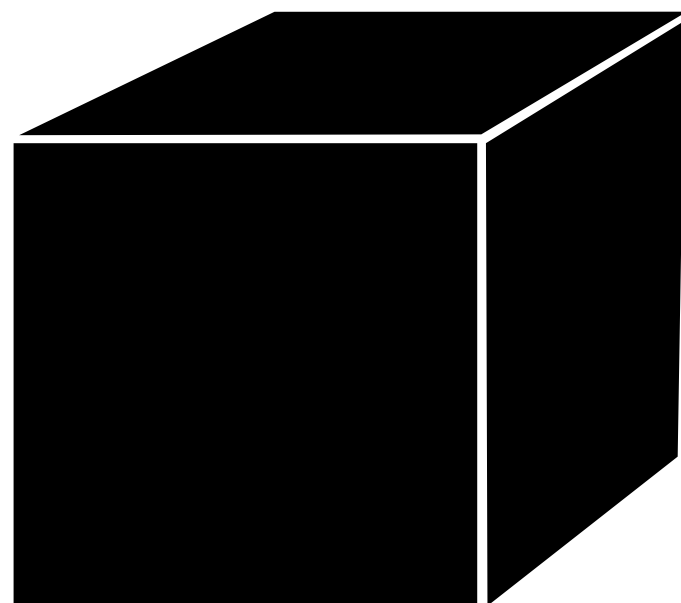


⇒ No summary statistics necessary, very fast evaluation... but may require large training samples

An incomplete list of likelihood-free inference methods

Treat simulator as black box:

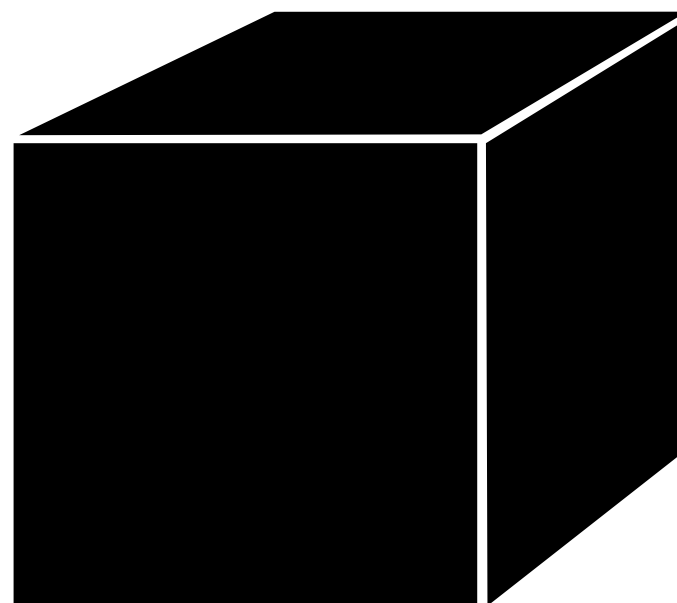
- Histograms of observables,
Approximate Bayesian Computation
Rely on summary statistics
- Machine learning techniques
Density networks, CARL, autoregressive models,
normalizing flows, ...



An incomplete list of likelihood-free inference methods

Treat simulator as black box:

- Histograms of observables, Approximate Bayesian Computation
Rely on summary statistics
- Machine learning techniques
Density networks, CARL, autoregressive models, normalizing flows, ...



Use physics insights (matrix elements):

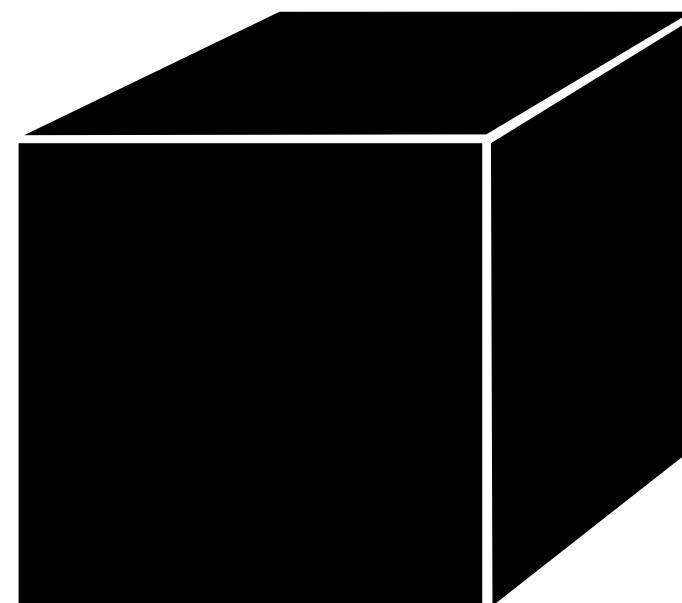
- Matrix Element Method, Optimal Observables, Shower Deconstruction, Event Deconstruction
Neglect or approximate shower + detector, explicitly calculate z integral



An incomplete list of likelihood-free inference methods

Treat simulator as black box:

- Histograms of observables, Approximate Bayesian Computation
Rely on summary statistics
- Machine learning techniques
Density networks, CARL, autoregressive models, normalizing flows, ...



Use physics insights (matrix elements):

- Matrix Element Method, Optimal Observables, Shower Deconstruction, Event Deconstruction
Neglect or approximate shower + detector, explicitly calculate z integral

- **Mining gold from the simulator**
Leverage matrix-element information + machine learning

New!



**A new approach:
“Mining gold” from the simulator**

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

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

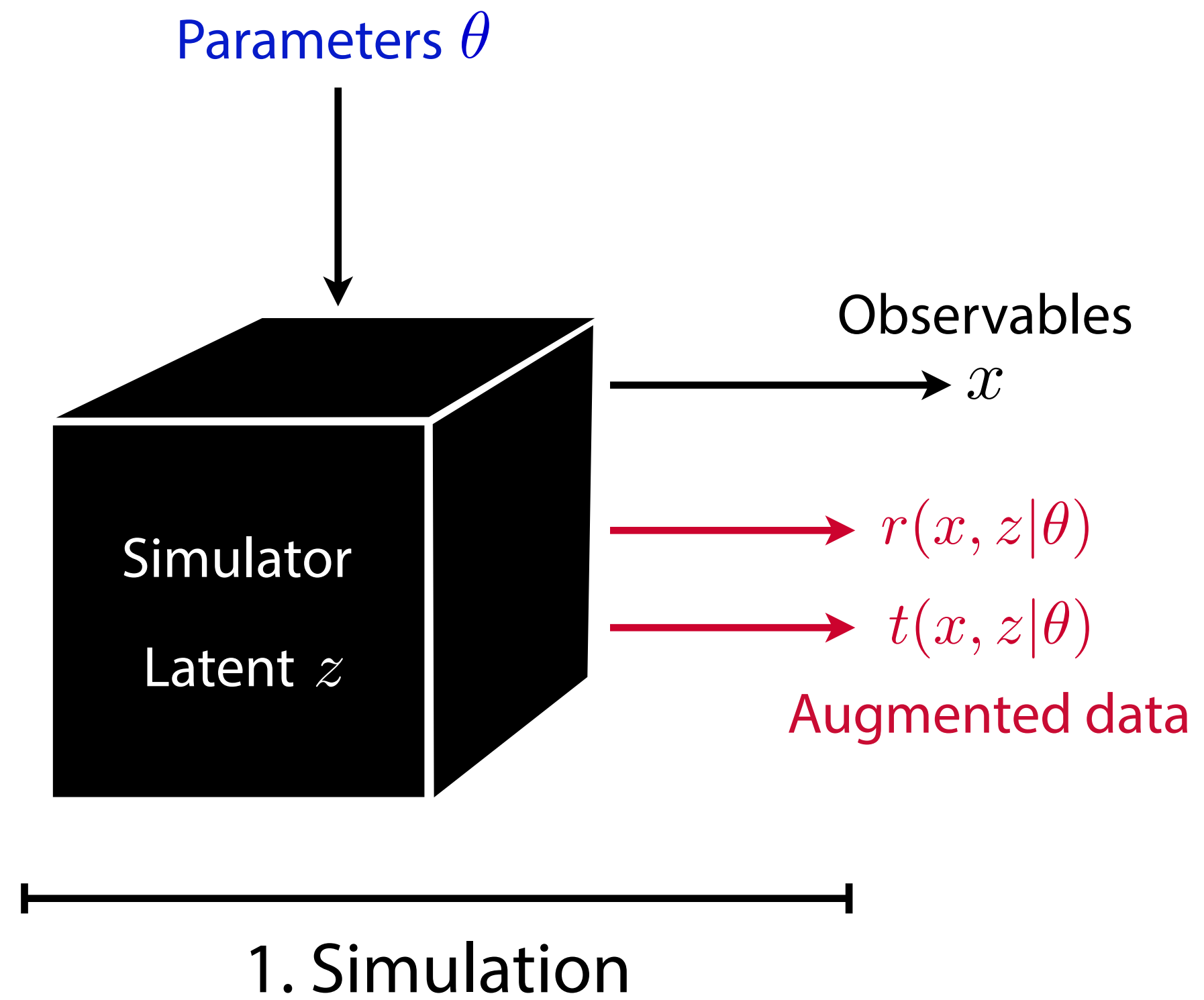
- in microseconds?

amortized inference: train once, then always evaluate fast

- requiring less training examples than established machine learning methods?

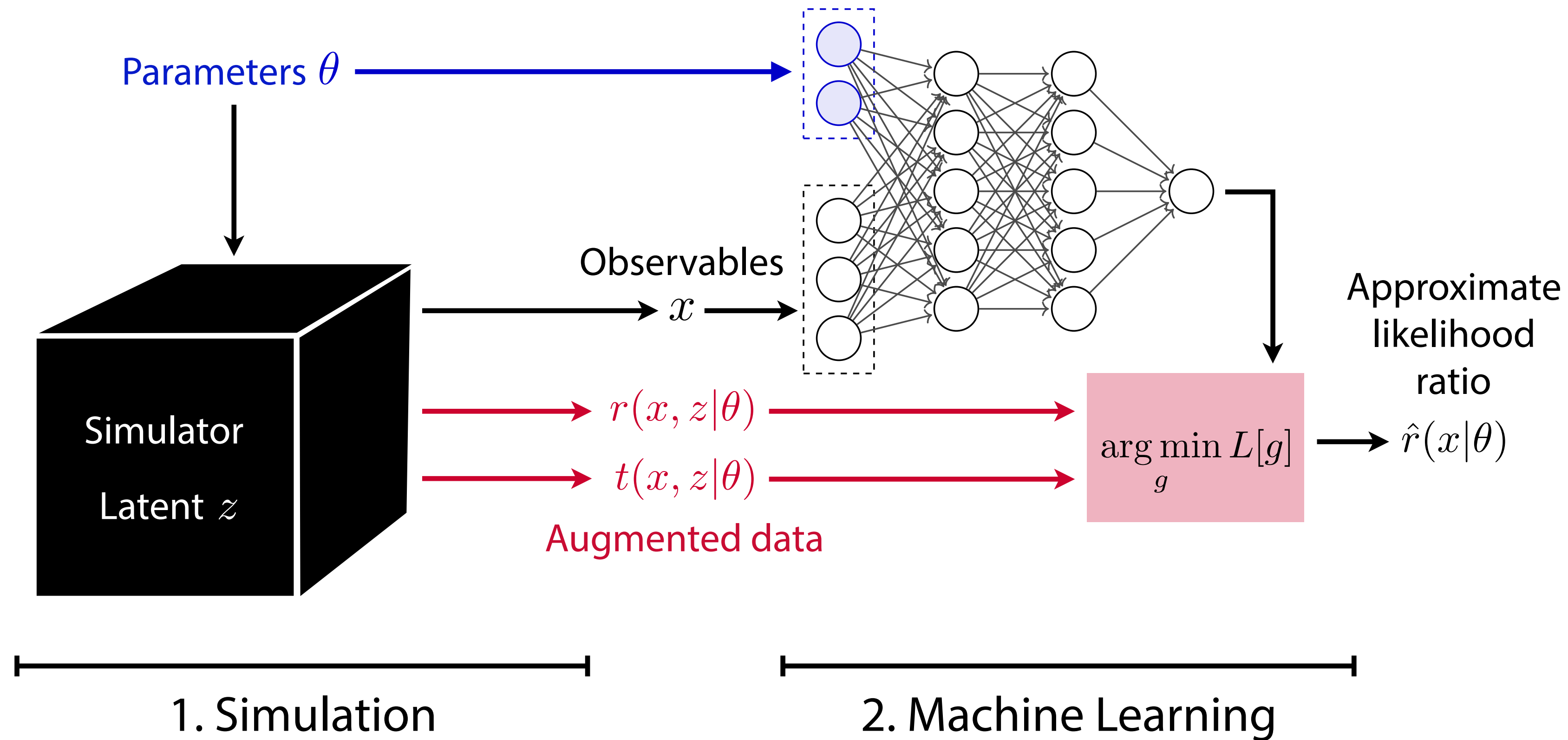
using matrix element information: "ML version of MEM"

Bird's-eye view



“Mining gold”: Extract additional information from simulator

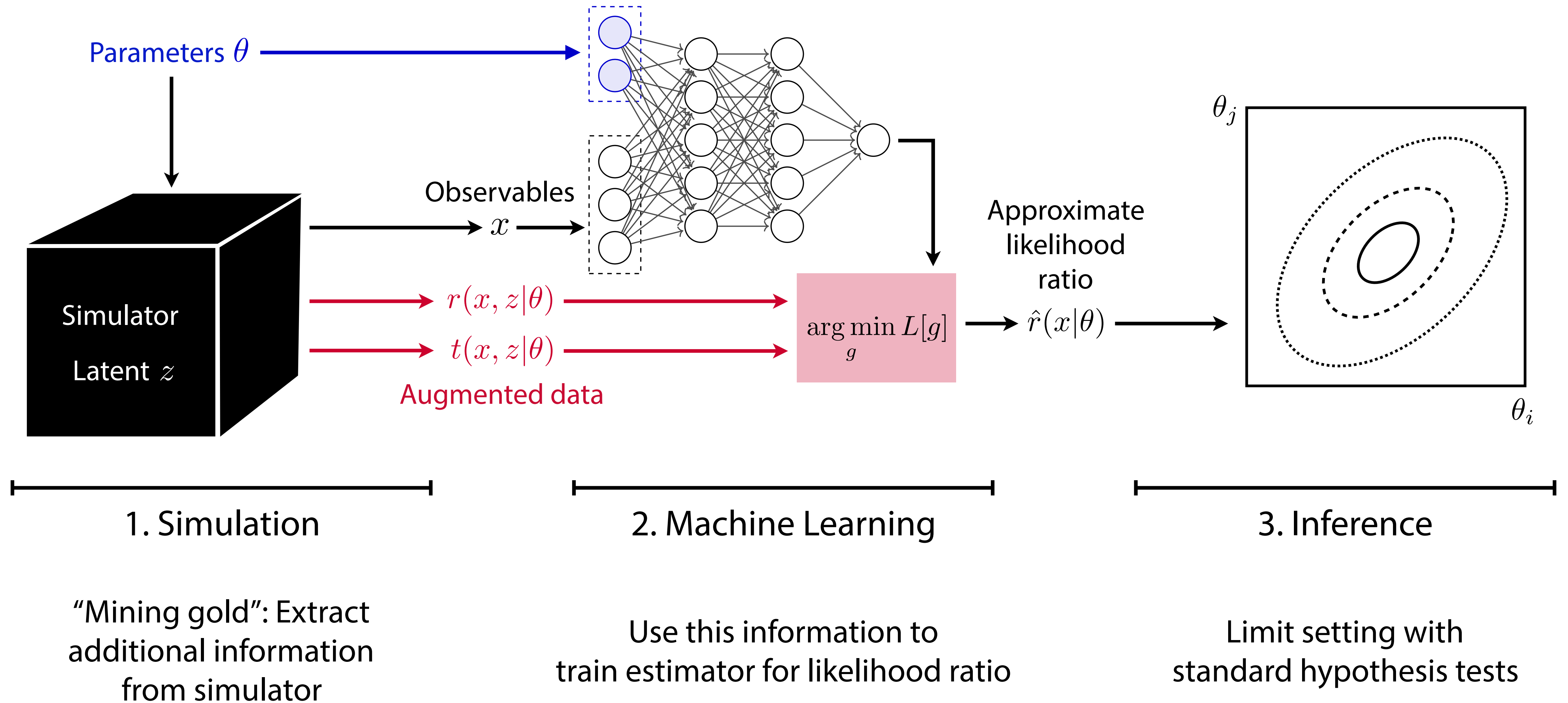
Bird's-eye view

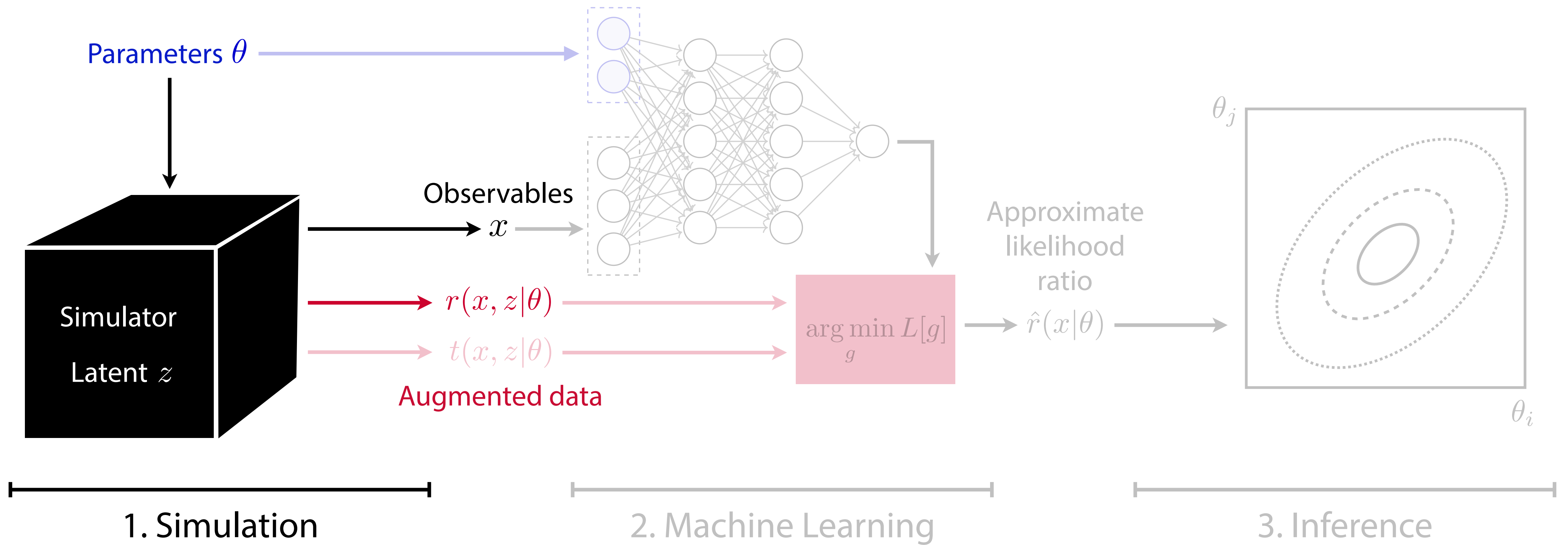


"Mining gold": Extract additional information from simulator

Use this information to train estimator for likelihood ratio

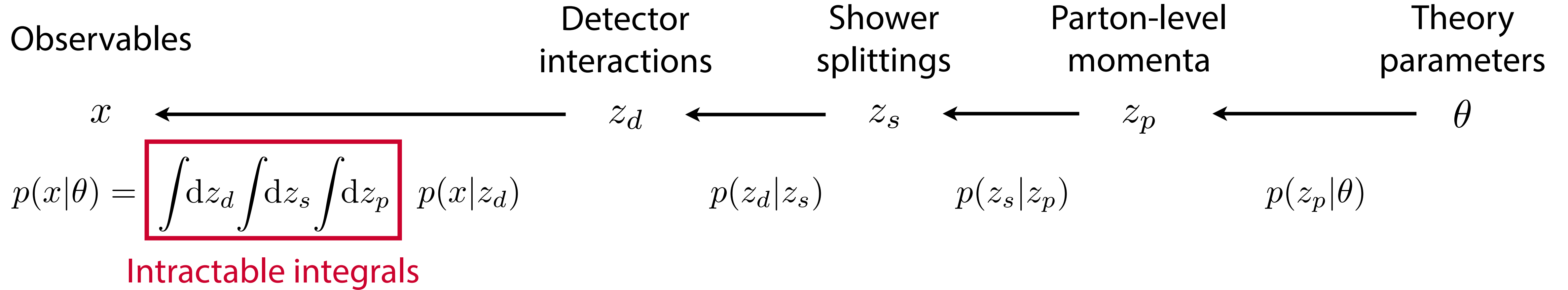
Bird's-eye view



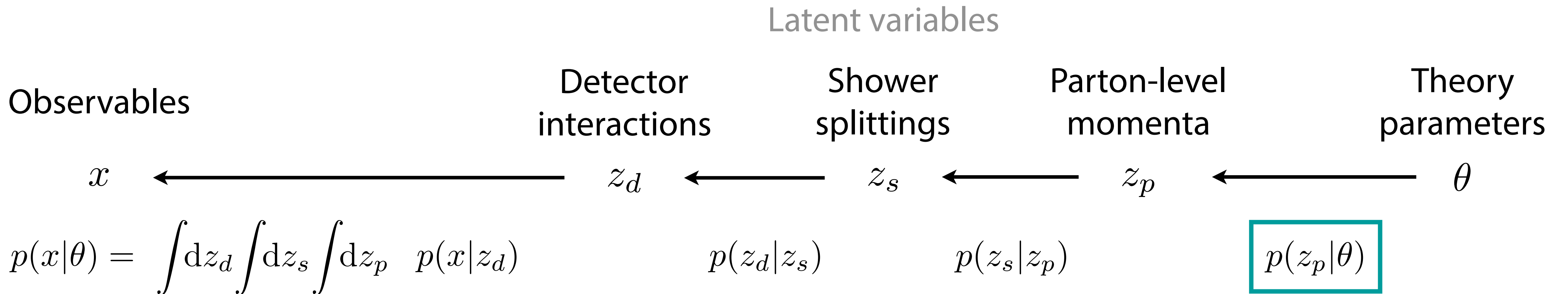


Mining gold from the simulator

Latent variables



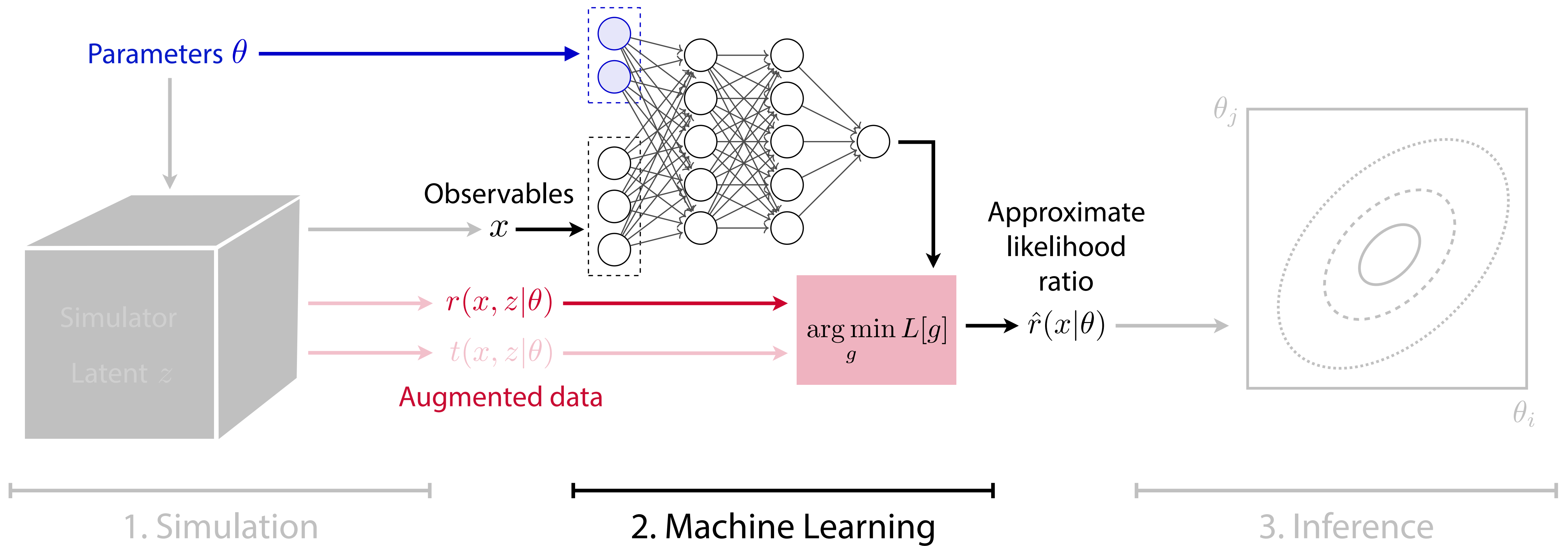
Mining gold from the simulator



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**

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



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 this simulated event, including all intermediate states, for θ_0 compared to θ_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**

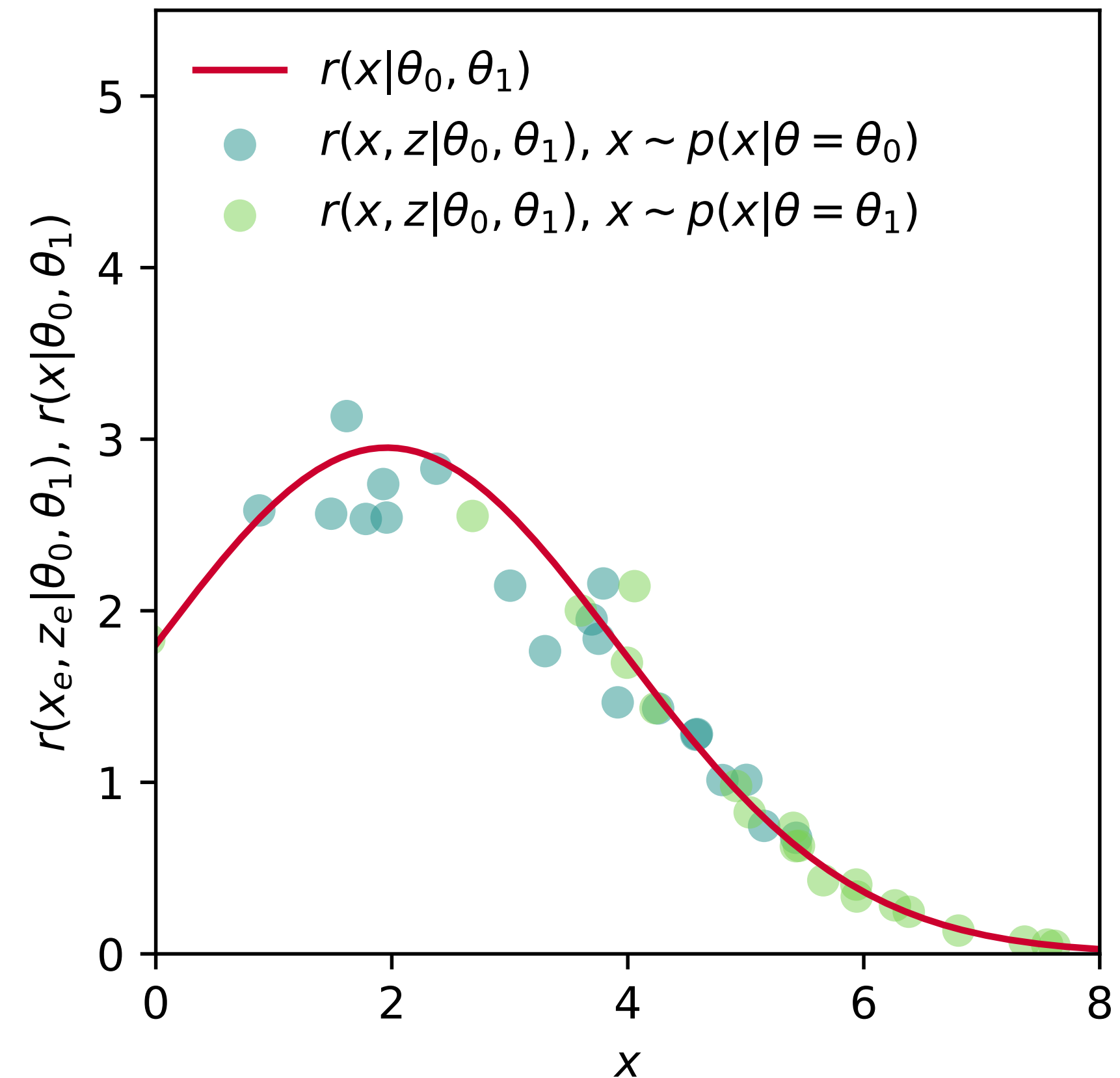
$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)}$$



$r(x, z|\theta_0, \theta_1)$ 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**

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

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

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

It is minimized by

$$r(x|\theta_0, \theta_1) = \arg \min_{\hat{r}(x|\theta_0, \theta_1)} L_r[\hat{r}(x|\theta_0, \theta_1)]!$$

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

We want the **likelihood ratio function**

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

Machine learning = applied calculus of variations

So to get a good estimator of the likelihood ratio, we need to minimize a functional numerically:

The diagram illustrates the process of finding a good estimator of the likelihood ratio. It features three main components with arrows pointing to a central equation:

- Variational family**: An arrow points from this text to the term $\hat{r}(x|\theta_0, \theta_1)$ in the equation.
- Extremization**: An arrow points from this text to the $\arg \min$ operator in the equation.
- Functional with integral**: An arrow points from this text to the functional $L_r[\hat{r}(x|\theta_0, \theta_1)]$ in the equation.

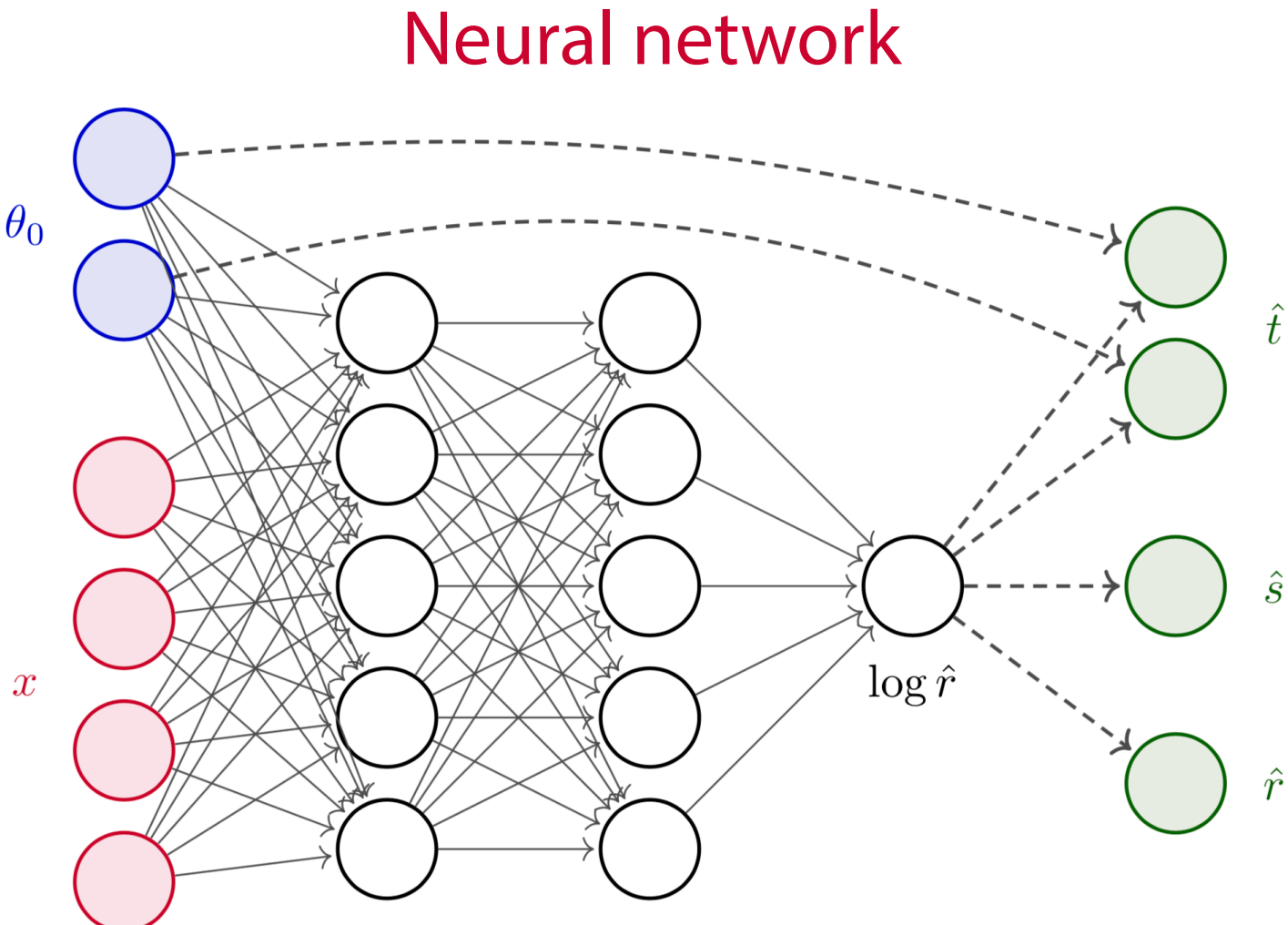
$$r(x|\theta_0, \theta_1) = \arg \min_{\hat{r}(x|\theta_0, \theta_1)} L_r[\hat{r}(x|\theta_0, \theta_1)]$$

Machine learning = applied calculus of variations

So to get a good estimator of the likelihood ratio, we need to minimize a functional numerically:

$$r(x|\theta_0, \theta_1) \approx \arg \min_{\hat{r}} L_r[\hat{r}(x|\theta_0, \theta_1)]$$

This is where machine learning comes in!



Neural network

Stochastic gradient descent

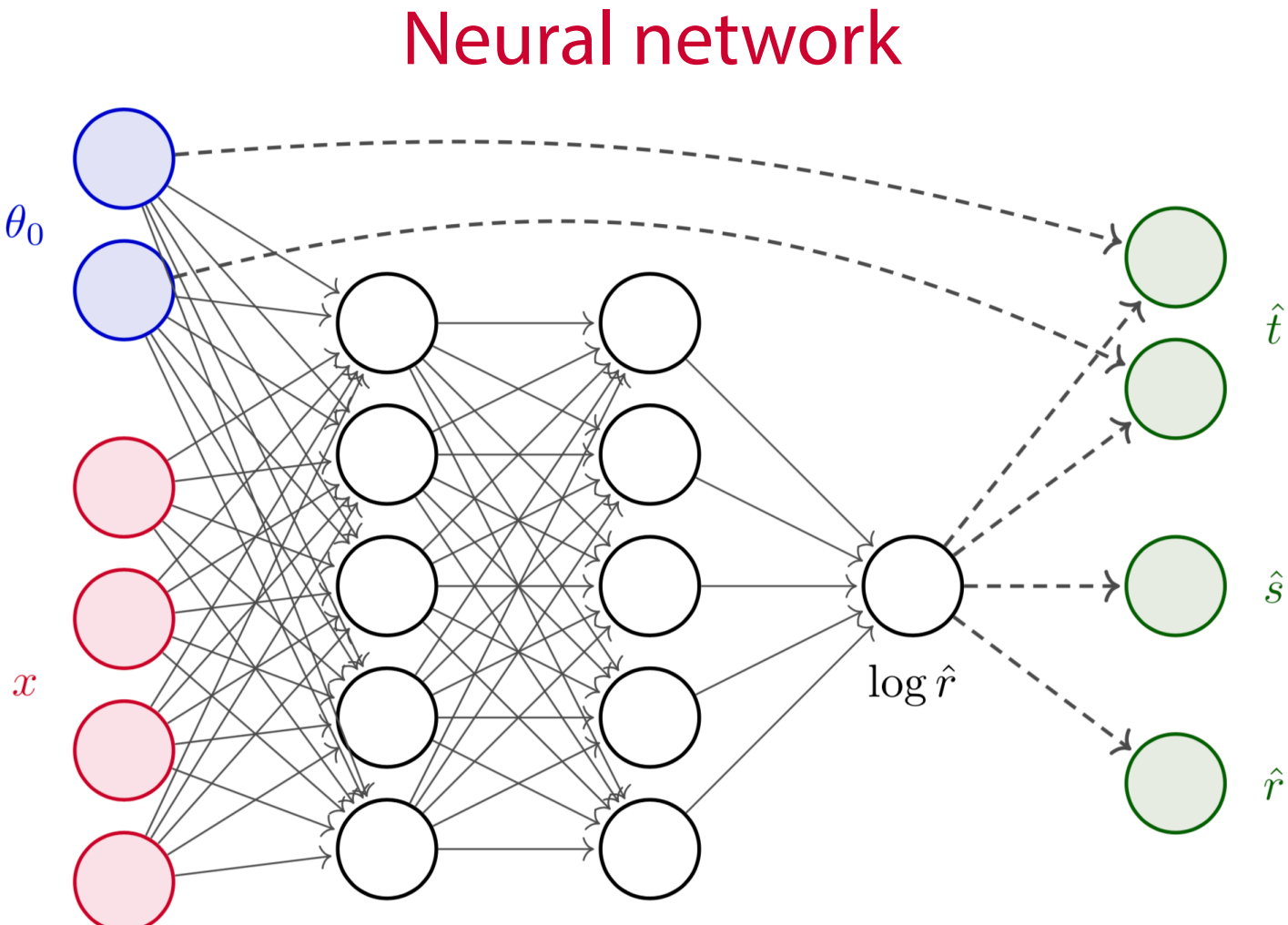
Loss function with finite sum over samples

Machine learning = applied calculus of variations

So to get a good estimator of the likelihood ratio, we need to minimize a functional numerically:

$$r(x|\theta_0, \theta_1) = \underset{\hat{r}(x|\theta_0, \theta_1)}{\operatorname{arg\,min}} L_r[\hat{r}(x|\theta_0, \theta_1)]$$

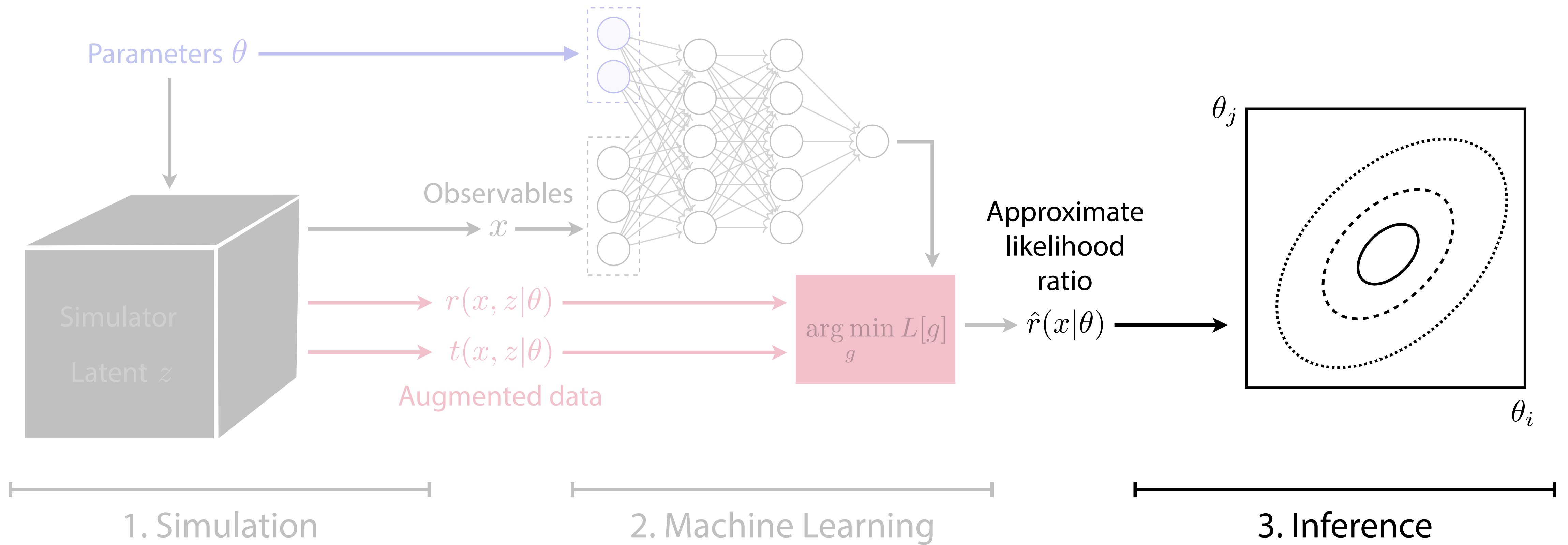
This is where machine learning comes in!

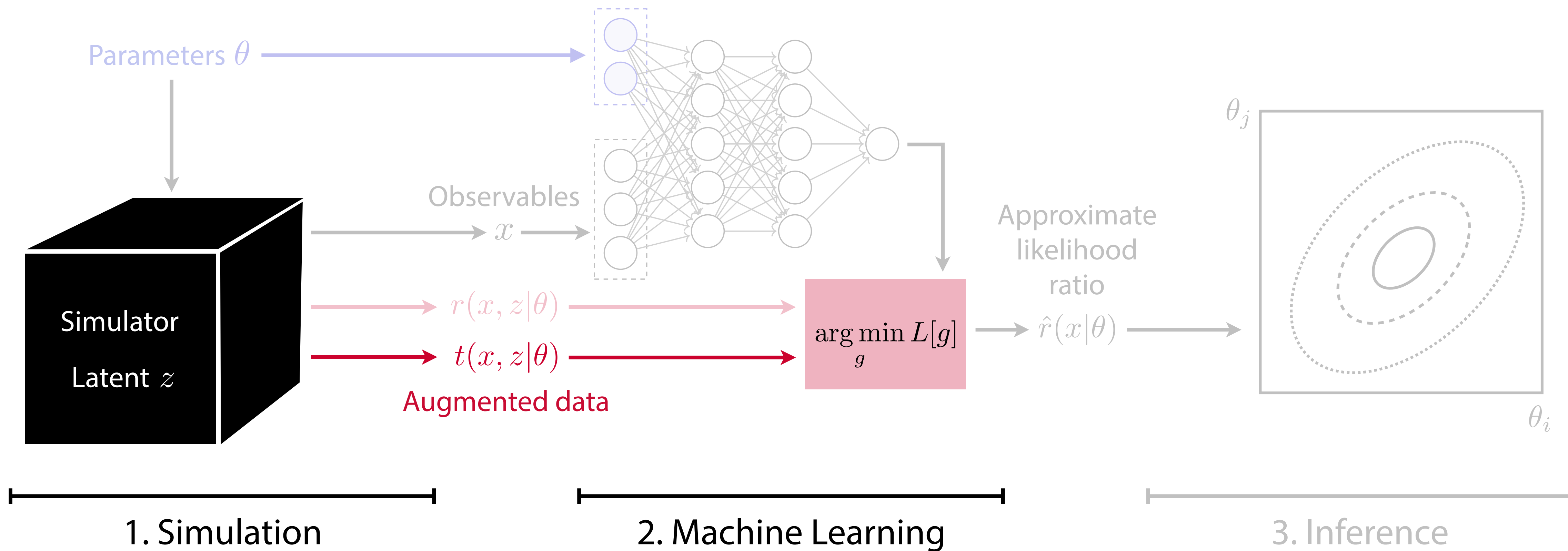


Stochastic gradient descent

Loss function with finite sum over samples

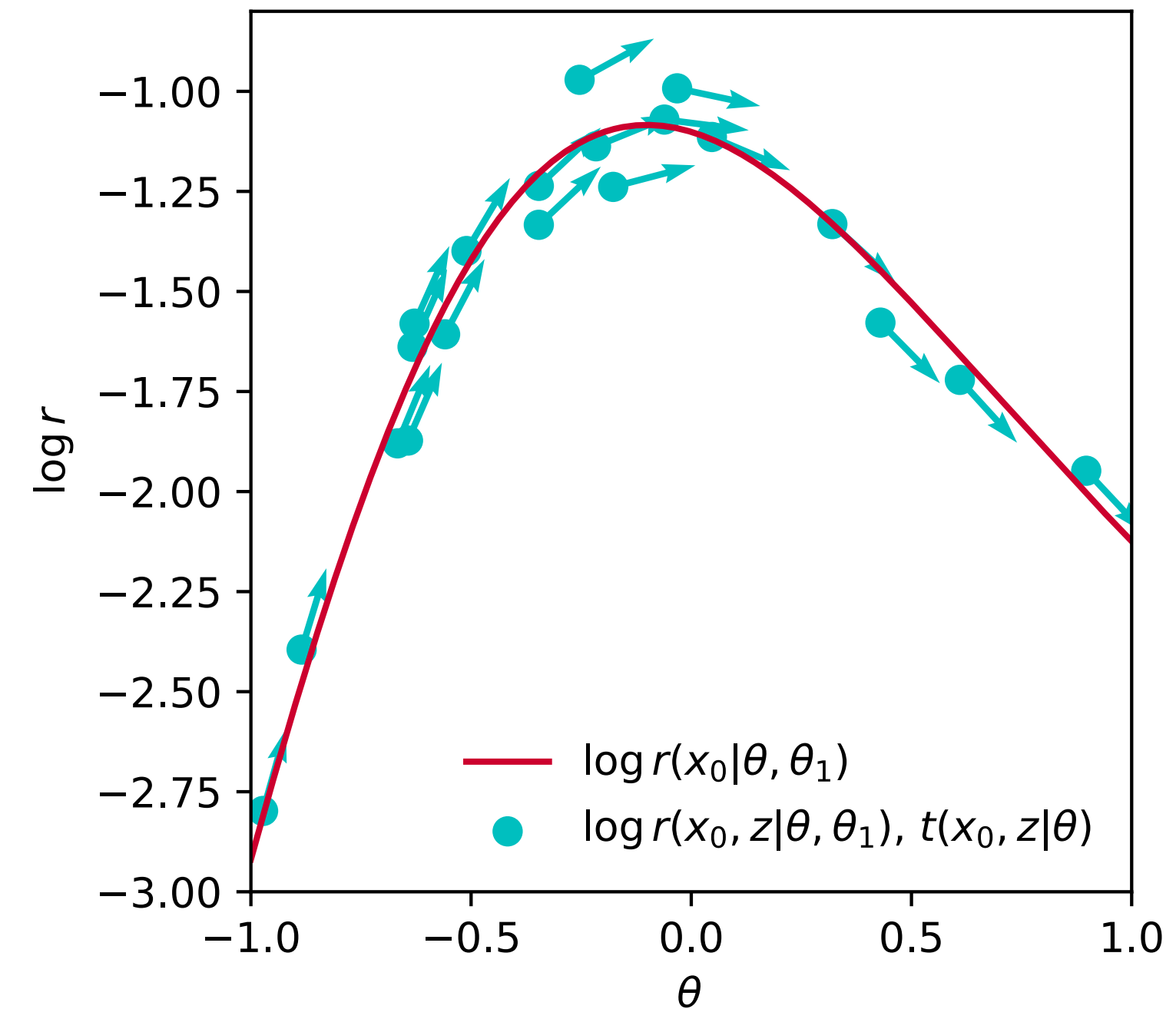
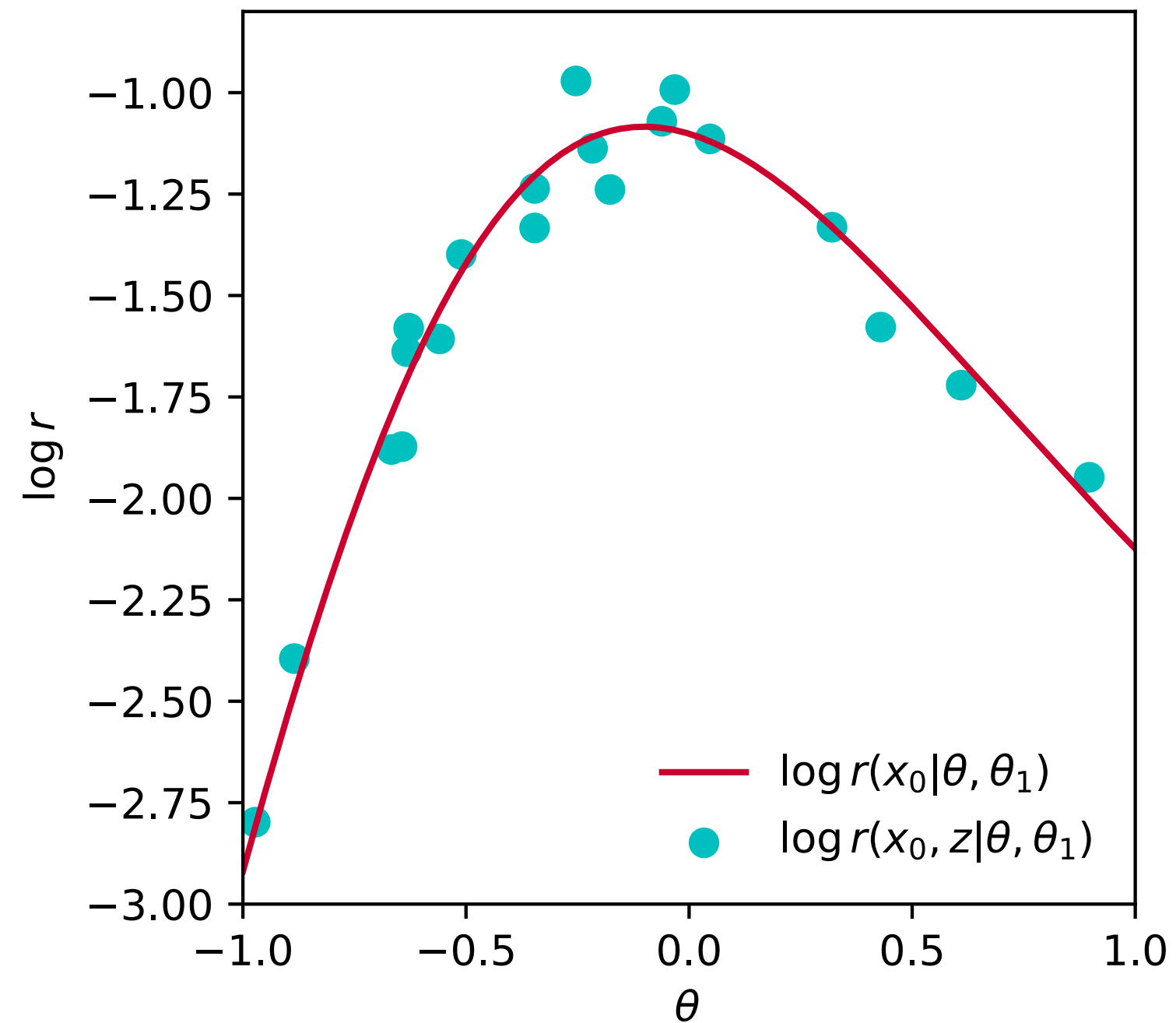
A sufficiently expressive neural network efficiently trained in this way with enough data will learn the likelihood ratio function $r(x|\theta_0, \theta_1)$!





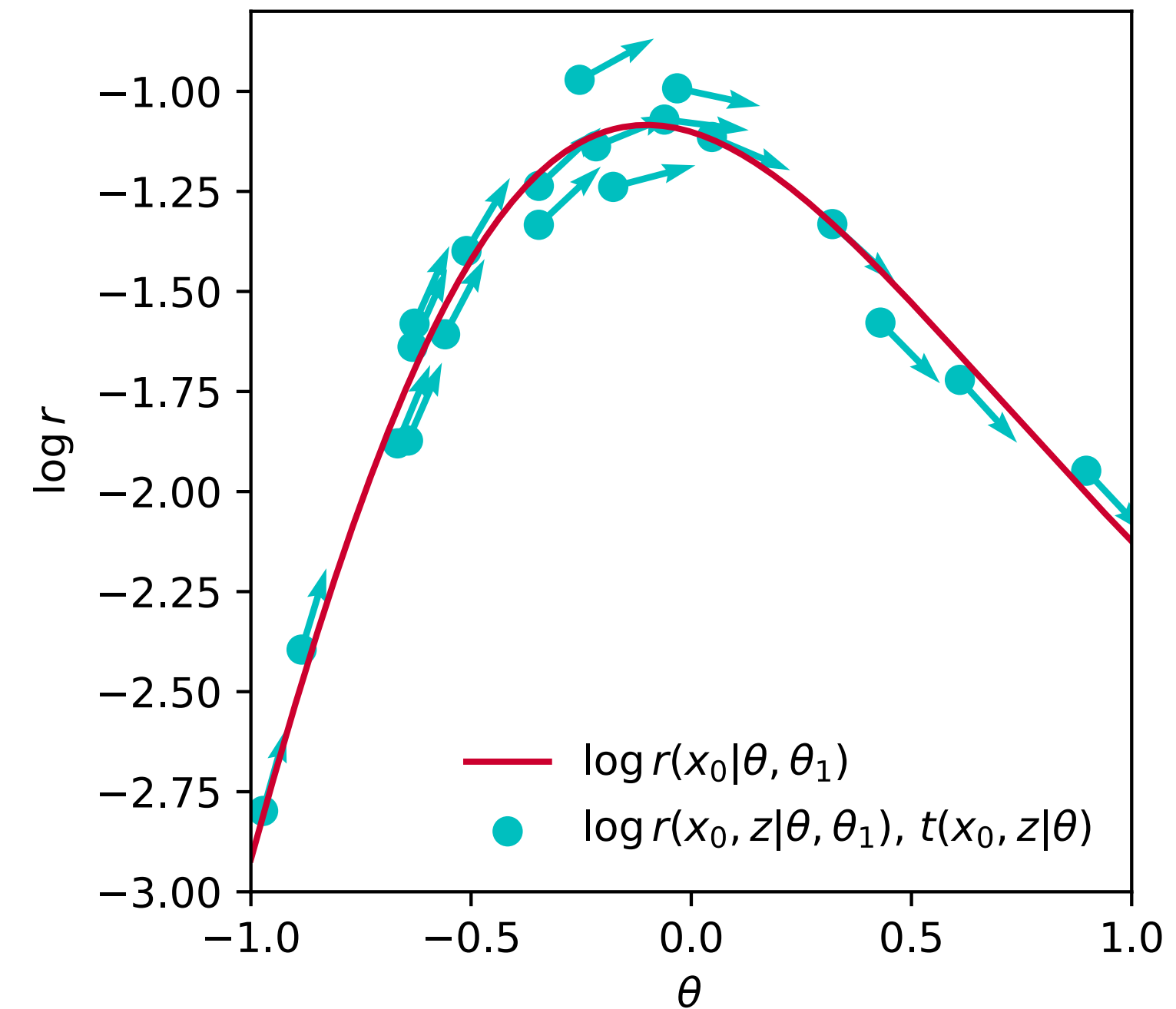
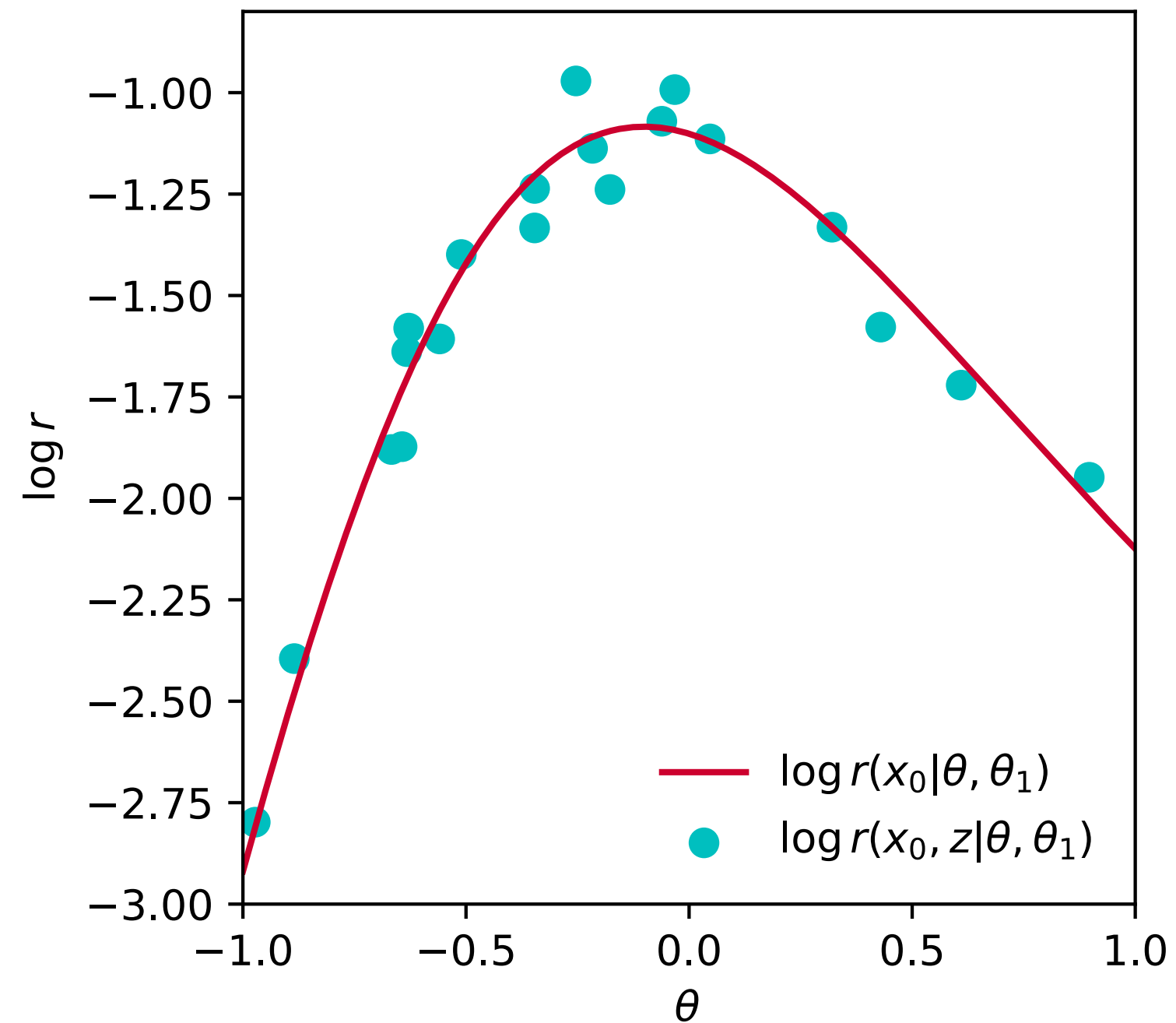
One more piece: the score

- Knowing derivative often helps fitting:



One more piece: the score

- Knowing derivative often helps fitting:



- In our case, the relevant quantity is the **score** $t(x|\theta_0) \equiv \nabla_{\theta} \log p(x|\theta) \Big|_{\theta_0}$.
- The score itself is intractable. But...

Learning the score

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

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

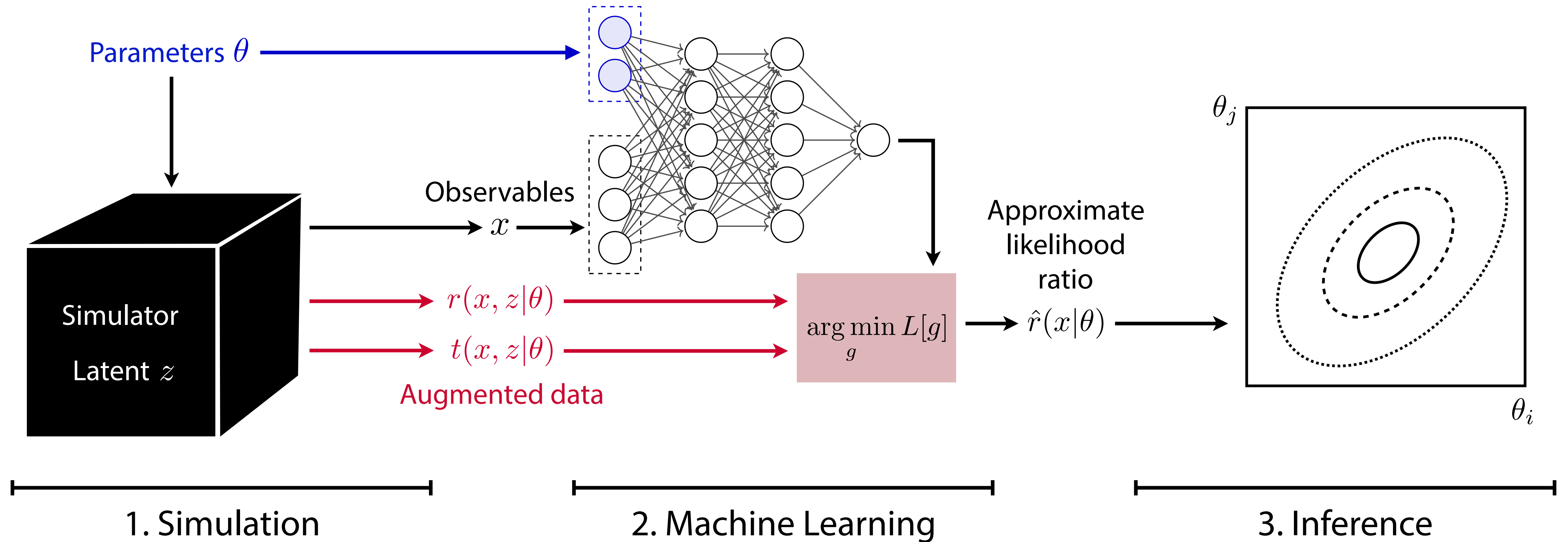
$$L_t[\hat{t}(x|\theta_0)] = \int dx \int dz p(x, z|\theta_0) \left[(\hat{t}(x|\theta_0) - t(x, z|\theta_0))^2 \right].$$

One can show it is minimized by

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

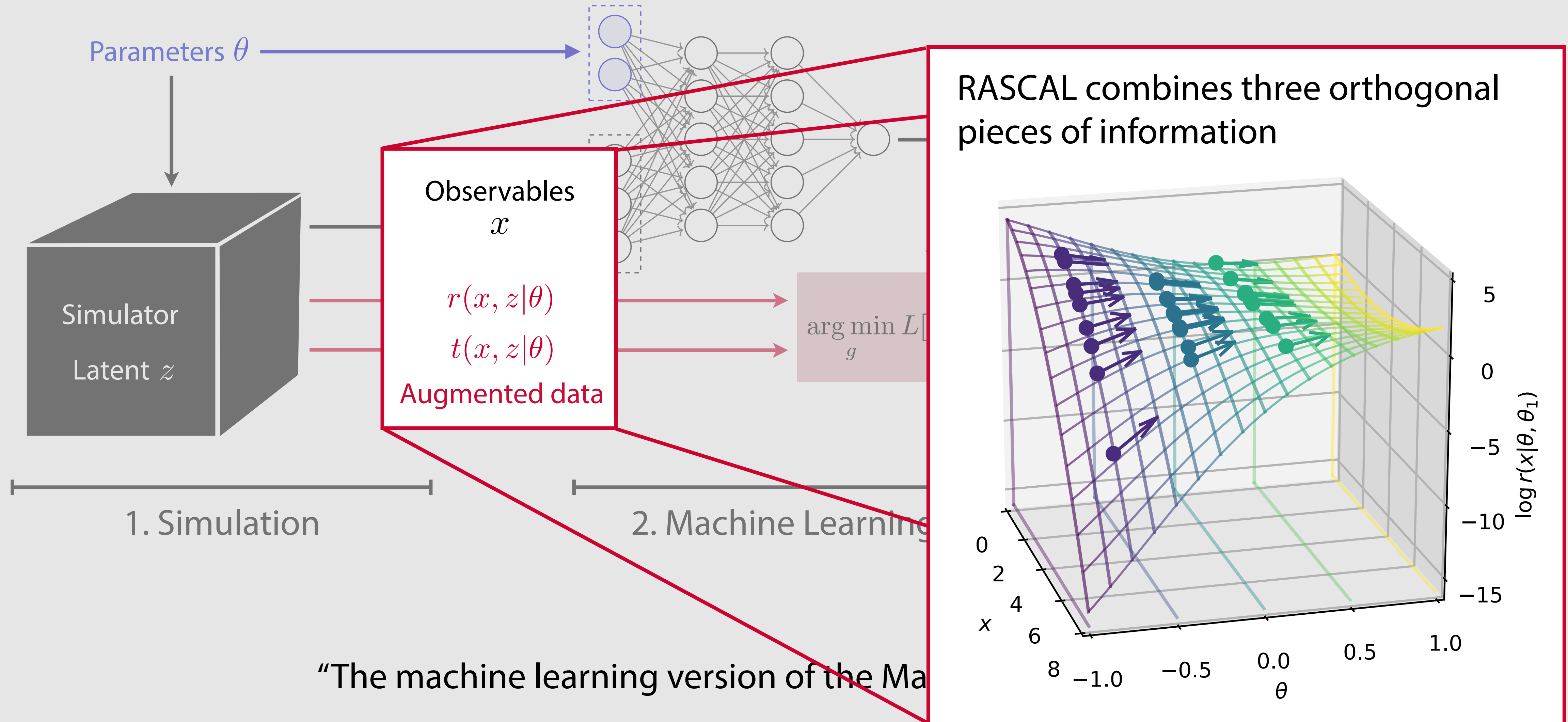
Again, we implement this minimization through machine learning.

Putting the pieces together: RASCAL (Ratio and score approximate likelihood ratio)



“The machine learning version of the Matrix Element Method”

Putting the pieces together: RASCAL (Ratio and score approximate likelihood ratio)



Learning optimal observables

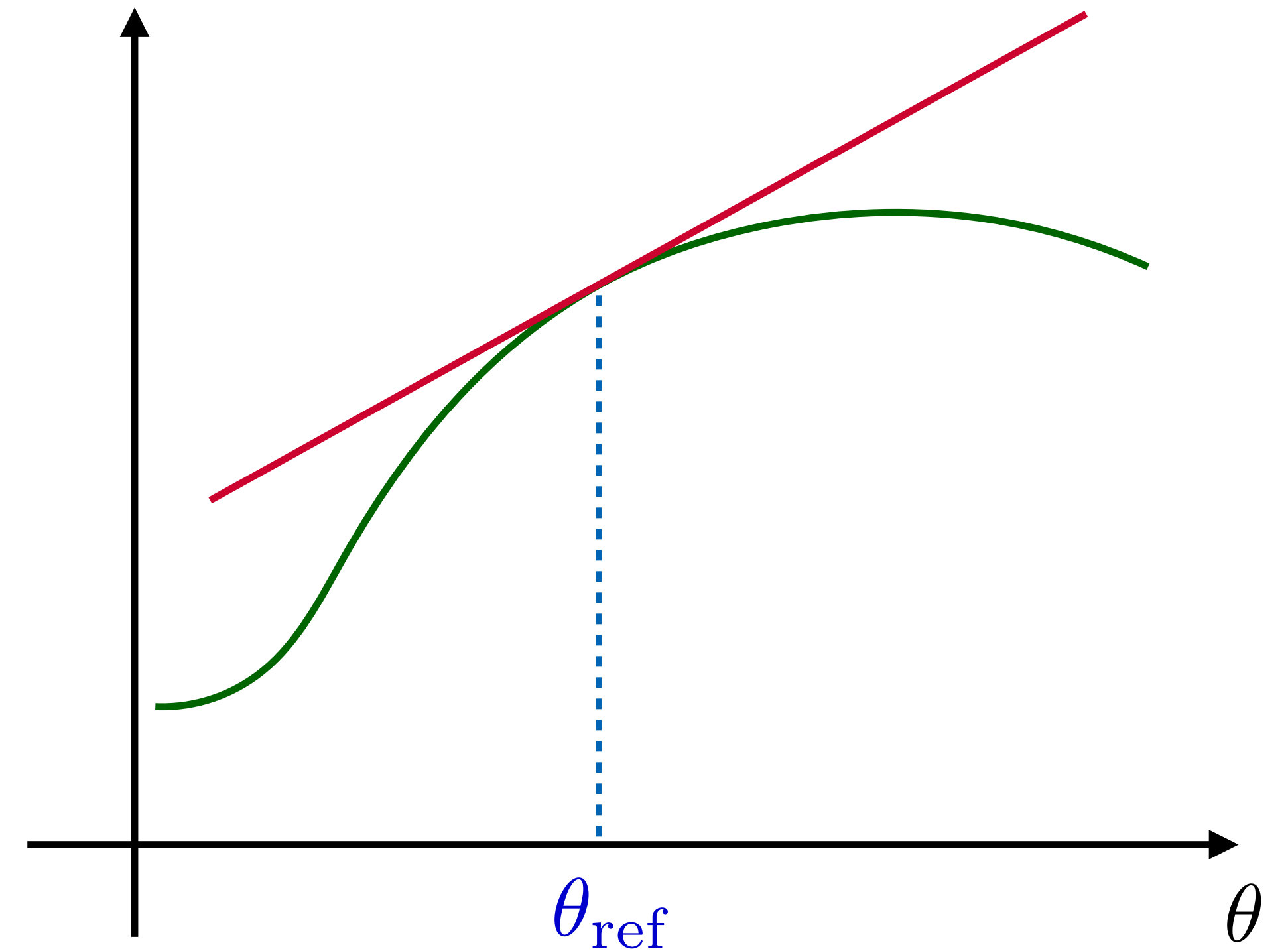
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

The local model

[see also J. Alsing, B. Wandelt 1712.00012; J. Alsing, B. Wandelt, S. Freney 1801.01497; P. de Castro, T. Dorigo 1806.04743; J. Alsing, B. Wandelt 1903.01473]

Taylor expansion of $\log p(x|\theta)$ around θ_{ref} :

$$\begin{aligned} \log p(x|\theta) &= \log p(x|\theta_{\text{ref}}) \\ &+ \underbrace{\nabla_{\theta} \log p(x|\theta) \Big|_{\theta_{\text{ref}}}}_{\equiv t(x|\theta_{\text{ref}})} \cdot (\theta - \theta_{\text{ref}}) \\ &+ \mathcal{O}((\theta - \theta_{\text{ref}})^2) \end{aligned}$$



The local model

[see also J. Alsing, B. Wandelt 1712.00012; J. Alsing, B. Wandelt, S. Freney 1801.01497; P. de Castro, T. Dorigo 1806.04743; J. Alsing, B. Wandelt 1903.01473]

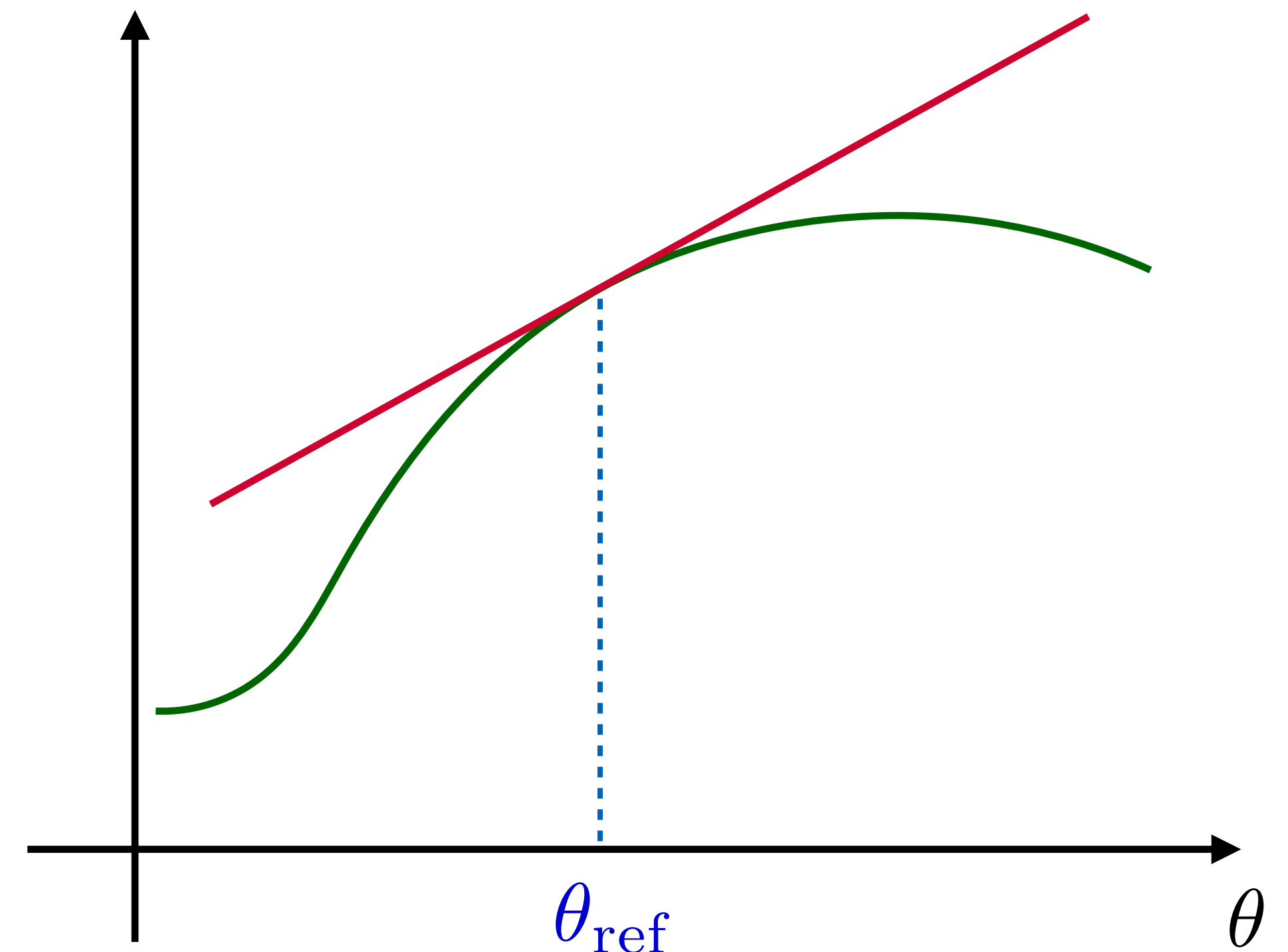
Taylor expansion of $\log p(x|\theta)$ around θ_{ref} :

$$\begin{aligned}\log p(x|\theta) &= \log p(x|\theta_{\text{ref}}) \\ &+ \underbrace{\nabla_{\theta} \log p(x|\theta) \Big|_{\theta_{\text{ref}}}}_{\equiv t(x|\theta_{\text{ref}})} \cdot (\theta - \theta_{\text{ref}}) \\ &+ \mathcal{O}((\theta - \theta_{\text{ref}})^2)\end{aligned}$$

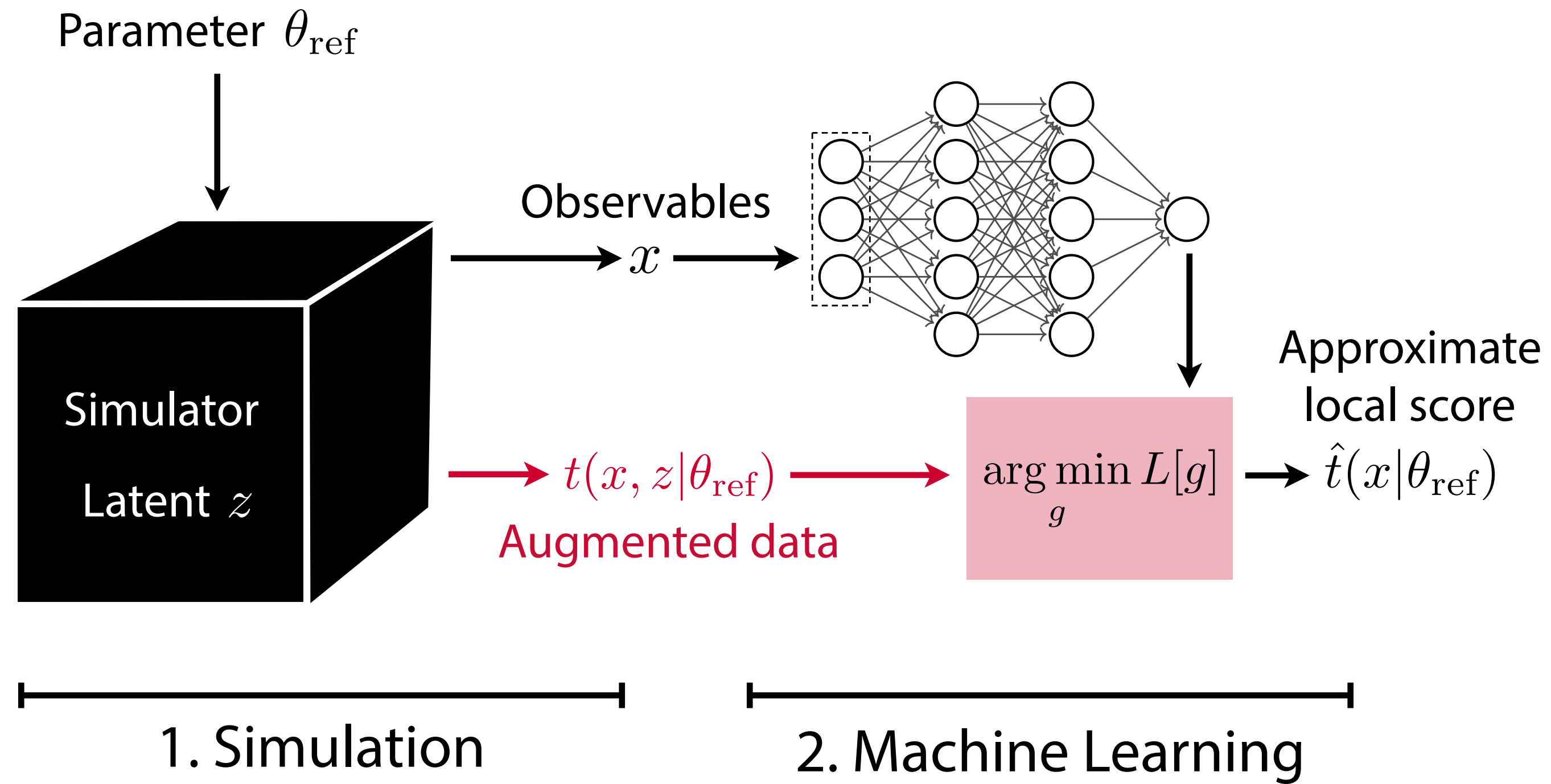
In the neighborhood of θ_{ref} (e.g. close to the SM):

- the **score vector** $t(x|\theta_{\text{ref}})$ is the sufficient statistics
- knowing $t(x|\theta_{\text{ref}})$ is just as powerful as knowing the full function $\log p(x|\theta)$
- $t(x|\theta_{\text{ref}})$ is the most powerful observable

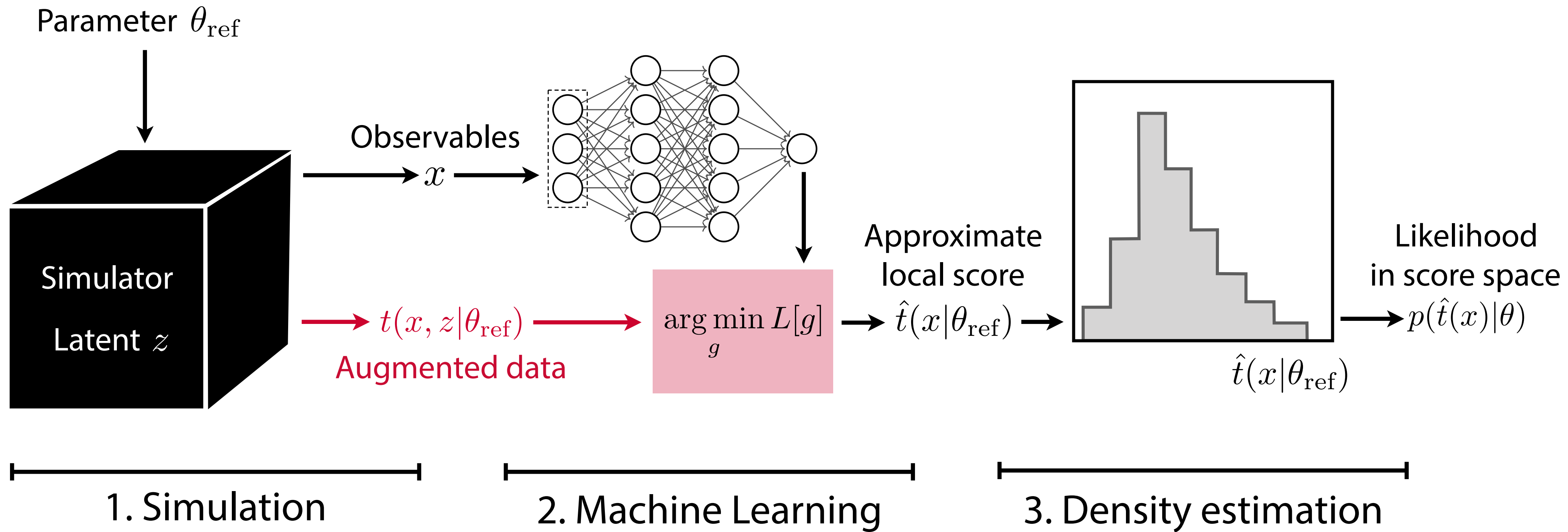
The score itself is intractable. But we can use the same trick as for the likelihood ratio!



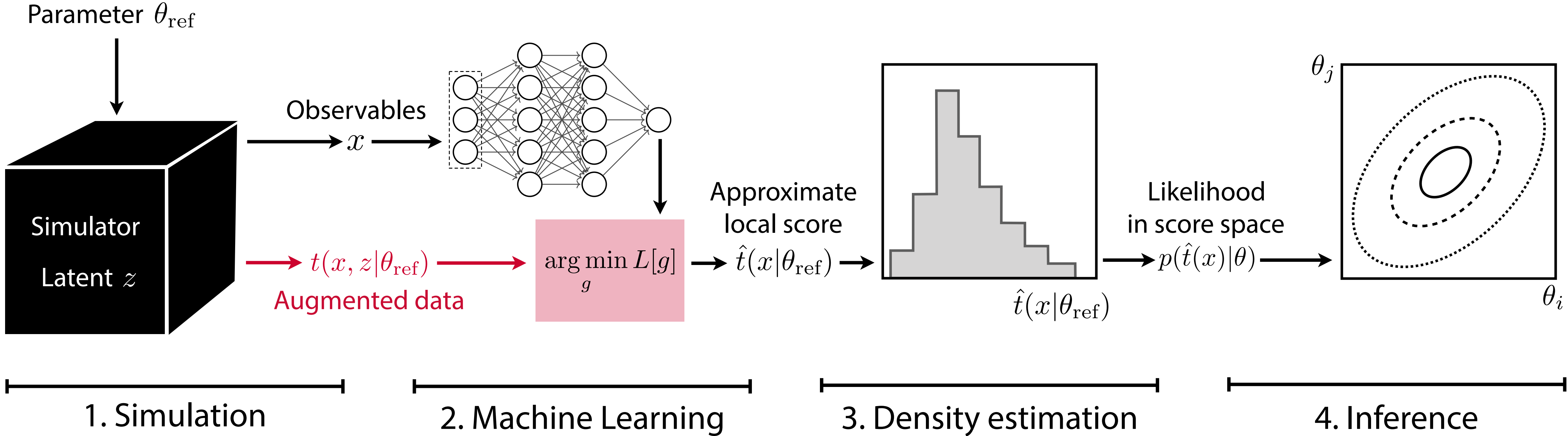
SALLY (Score approximates likelihood locally)



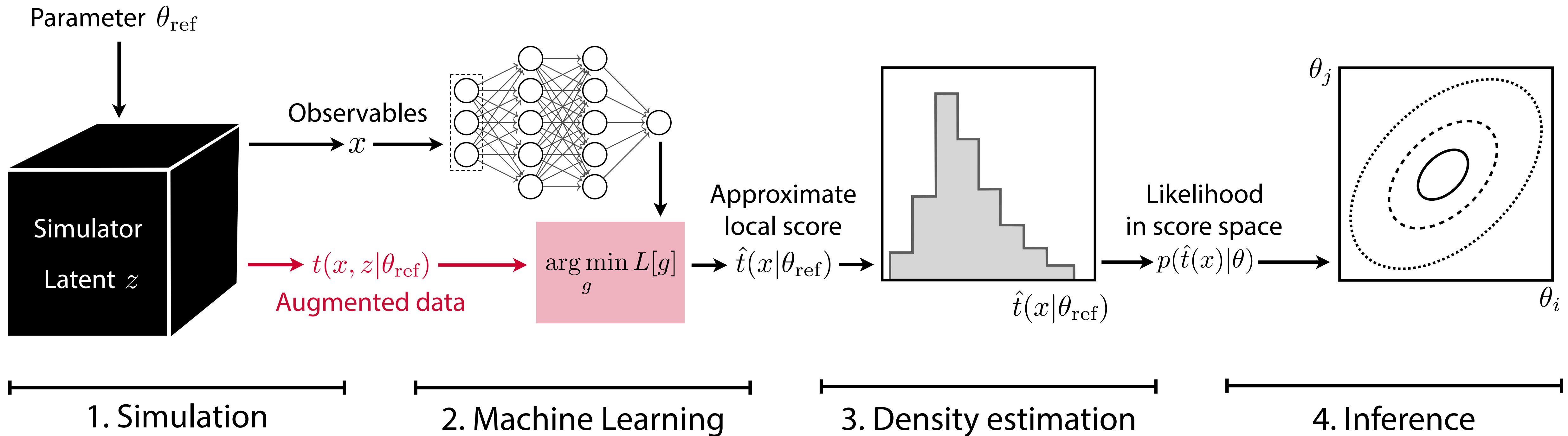
SALLY (Score approximates likelihood locally)



SALLY (Score approximates likelihood locally)



SALLY (Score approximates likelihood locally)



“The machine learning version of Optimal Observables”:

- Simpler & more robust than RASCAL
- Just as powerful close to θ_{ref} , but can lead to suboptimal limits further away

A family of new inference techniques

Method	Simulate	Extract		NN estimates	Asympt. exact	Generative
		$r(x, z)$	$t(x, z)$			
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.	

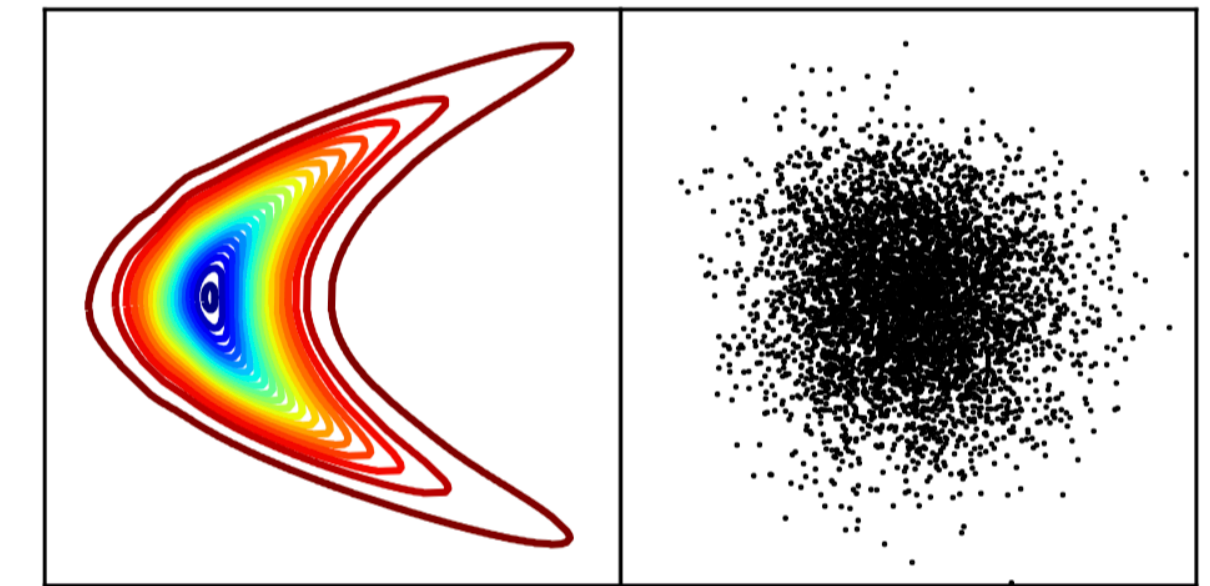
A family of new inference techniques

Method	Simulate	Extract		NN estimates	Asympt. exact	Generative
		$r(x, z)$	$t(x, z)$			
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
 [M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez 1808.00973]

A family of new inference techniques

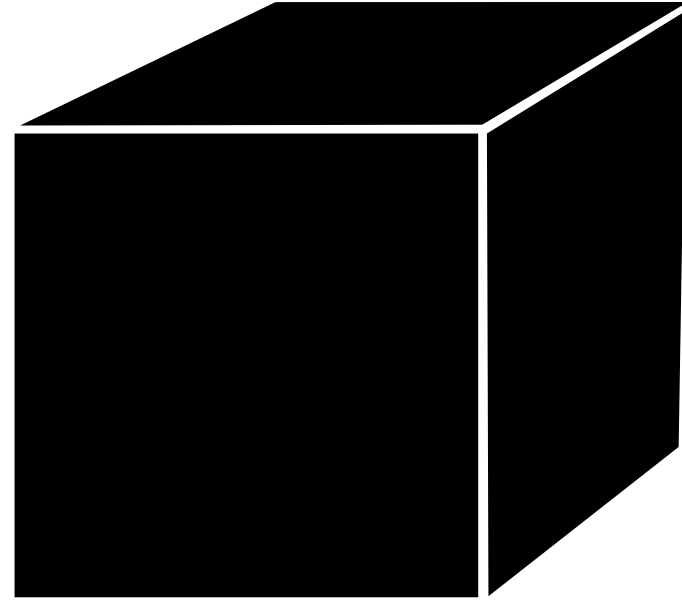
Method	Simulate	Extract		NN estimates	Asympt. exact	Generative
		$r(x, z)$	$t(x, z)$			
ROLR	$\theta_0 \sim \pi(\theta), \theta_1$	✓		$\hat{r}(x \theta_0, \theta_1)$	✓	
CASCAL	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x \theta_0, \theta_1)$	✓	
ALICE	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x \theta_0, \theta_1)$	✓	
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; ...]

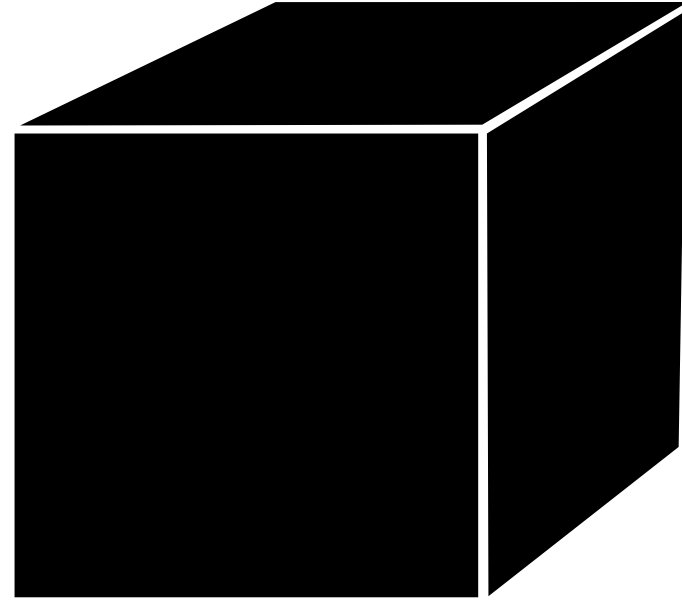
Systematics



Don't fully trust the simulator?

- Nuisance parameters to model systematic uncertainties
- Methods learn dependence both on parameters of interest and nuisance parameters. Then we can construct profile likelihood and “nuisance-hardened” score
[J. Alsing, B. Wandelt 1903.01473;
see also P. de Castro, T. Dorigo 1806.04743]
- Alternatively: Robustness to nuisance with adversarial training
[G. Louppe, M. Kagan, K. Cranmer 1611.01046]

Systematics



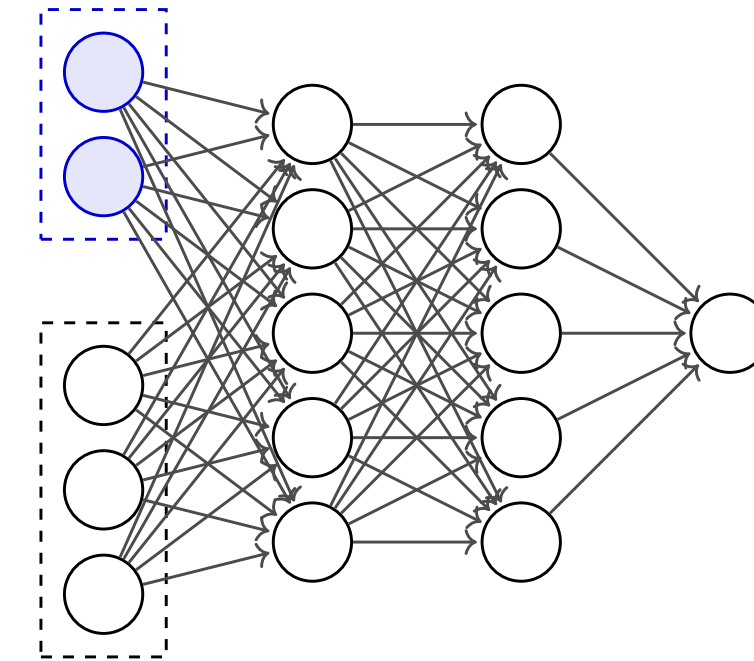
Don't fully trust the simulator?

- Nuisance parameters to model systematic uncertainties
- Methods learn dependence both on parameters of interest and nuisance parameters. Then we can construct profile likelihood and “nuisance-hardened” score

[J. Alsing, B. Wandelt 1903.01473;
see also P. de Castro, T. Dorigo 1806.04743]

- **Alternatively: Robustness to nuisance with adversarial training**

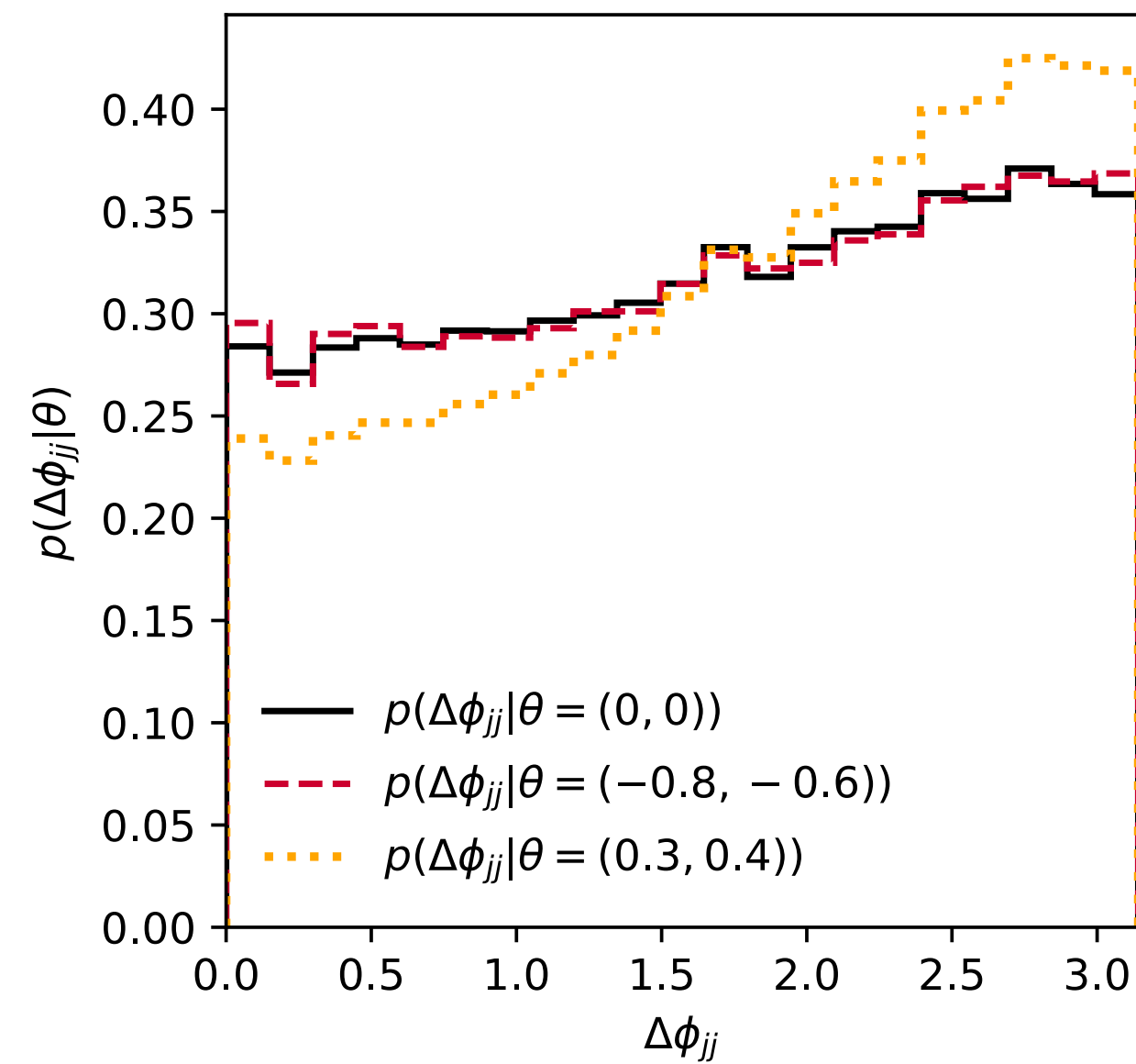
[G. Louppe, M. Kagan, K. Cranmer 1611.01046]



Don't blindly trust the neural network?

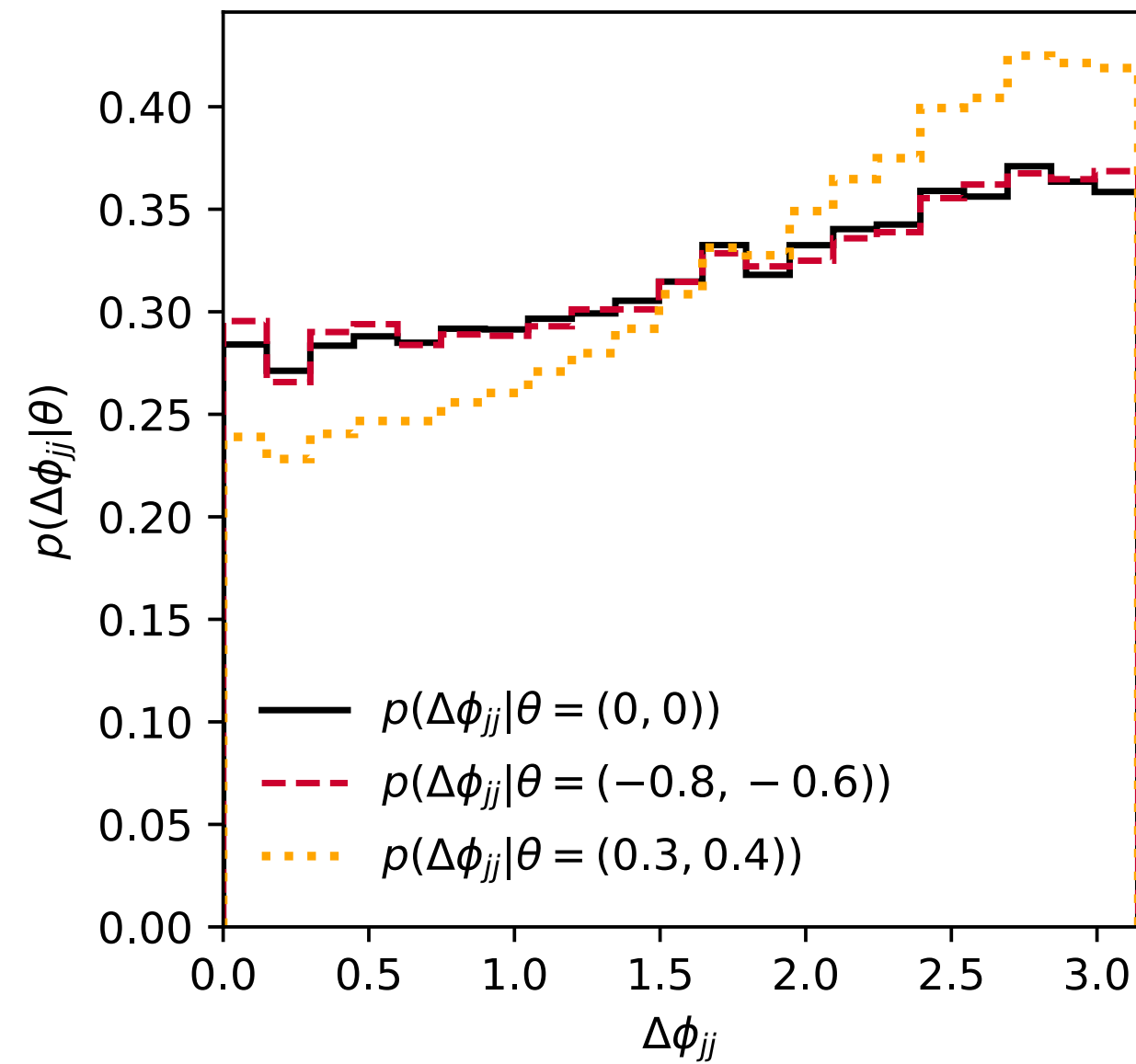
- Diagnostic cross checks: known expectation values, “critic” tests
- Calibration / Neyman construction with toys: badly trained network can lead to suboptimal limits, but not to wrong limits

Perfect match for EFT measurements

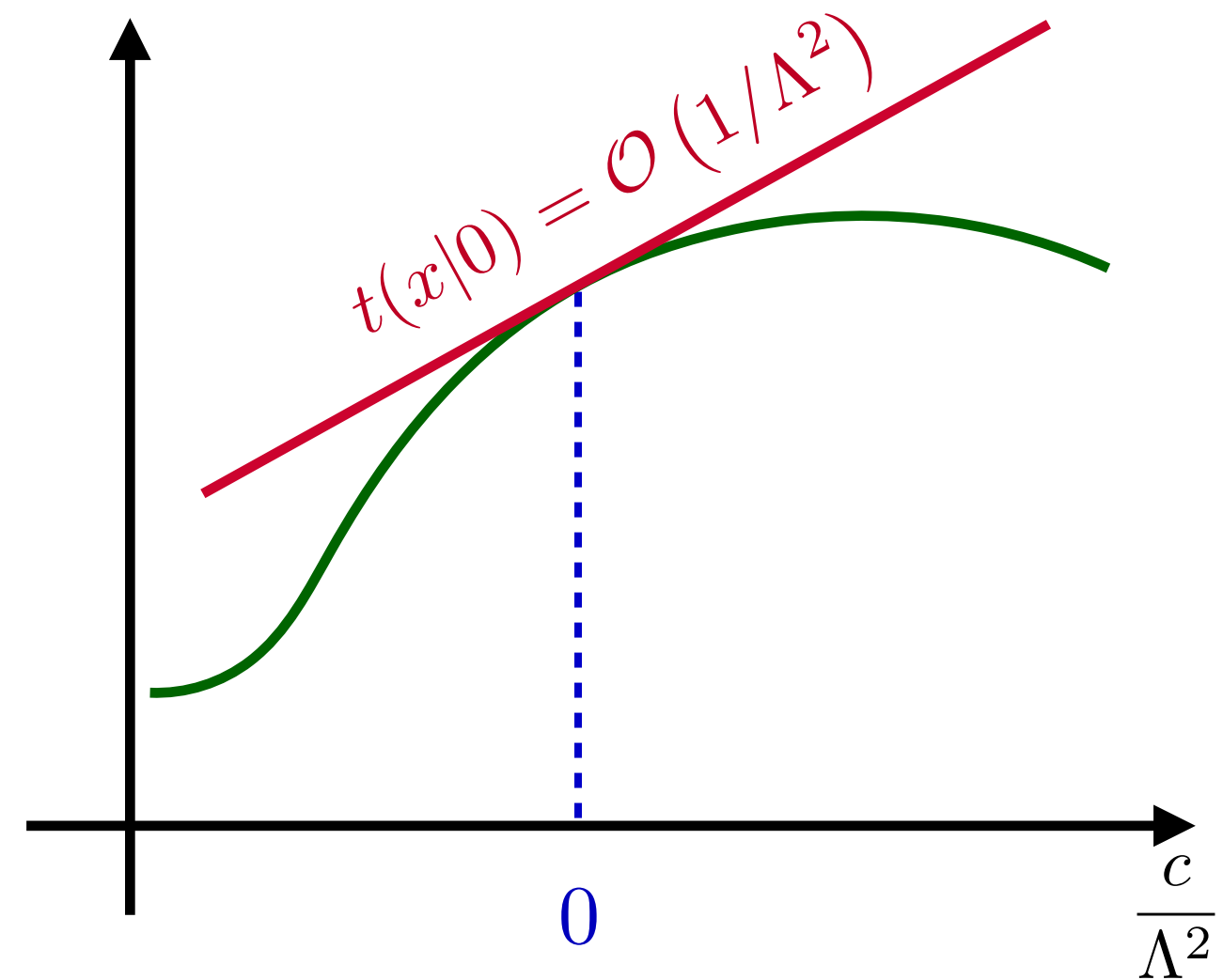


- Good for subtle kinematic effects
(Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)

Perfect match for EFT measurements

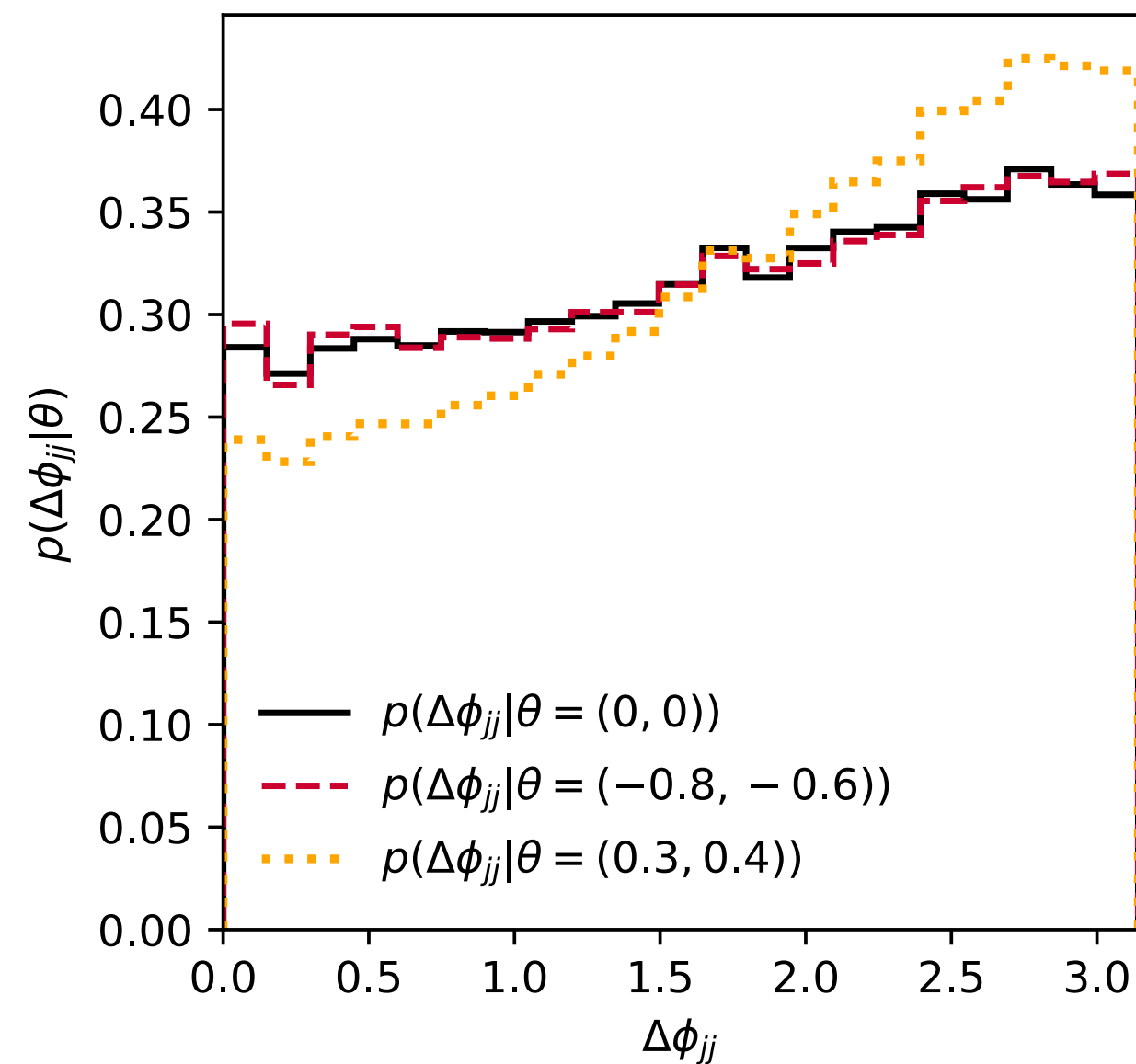


- Good for subtle kinematic effects
(Subtle point: Large overlap of kinematic distributions reduces variance of joint likelihood ratio / joint score)

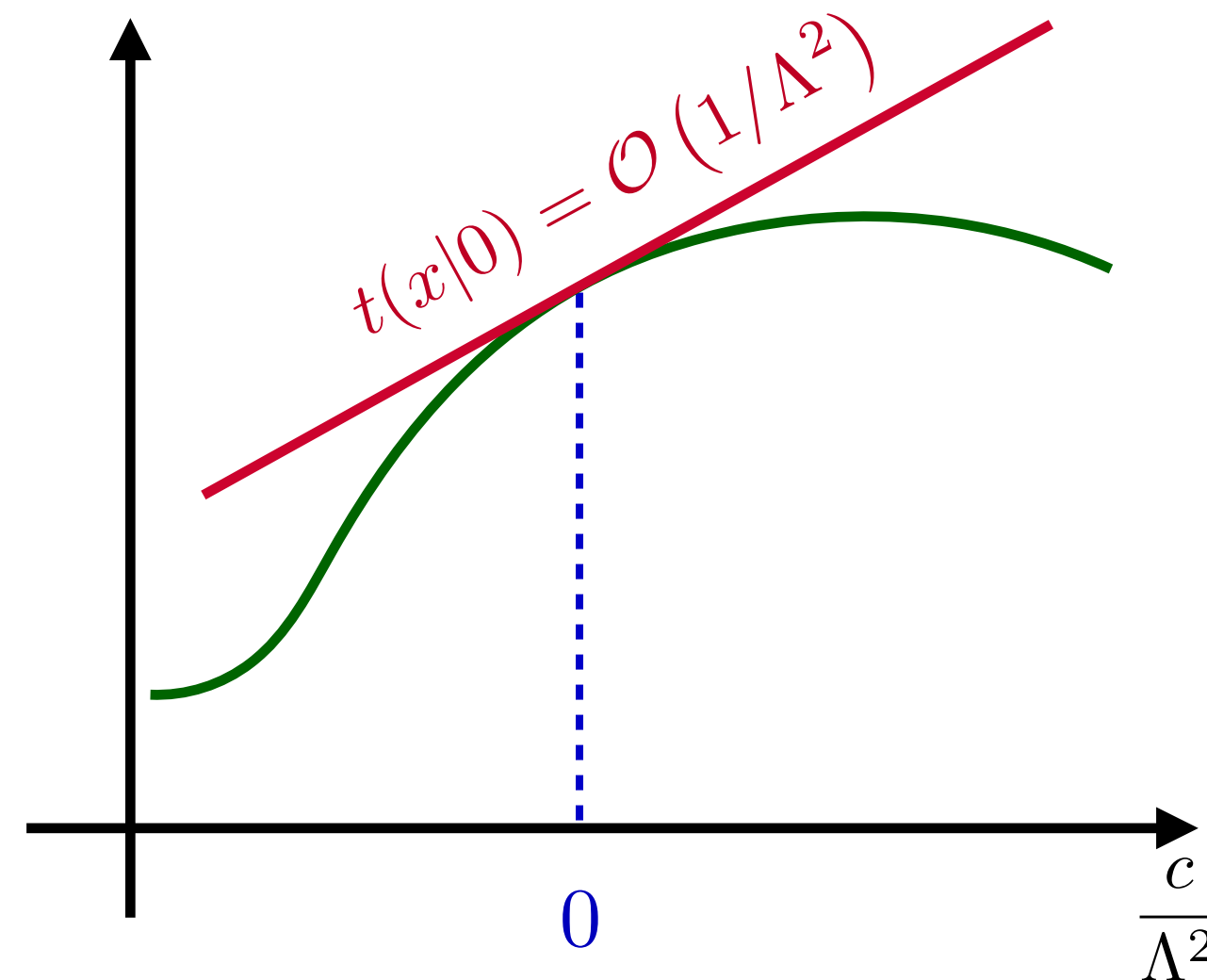


- Interference effects can be isolated using SALLY at the SM (SMALLY?)

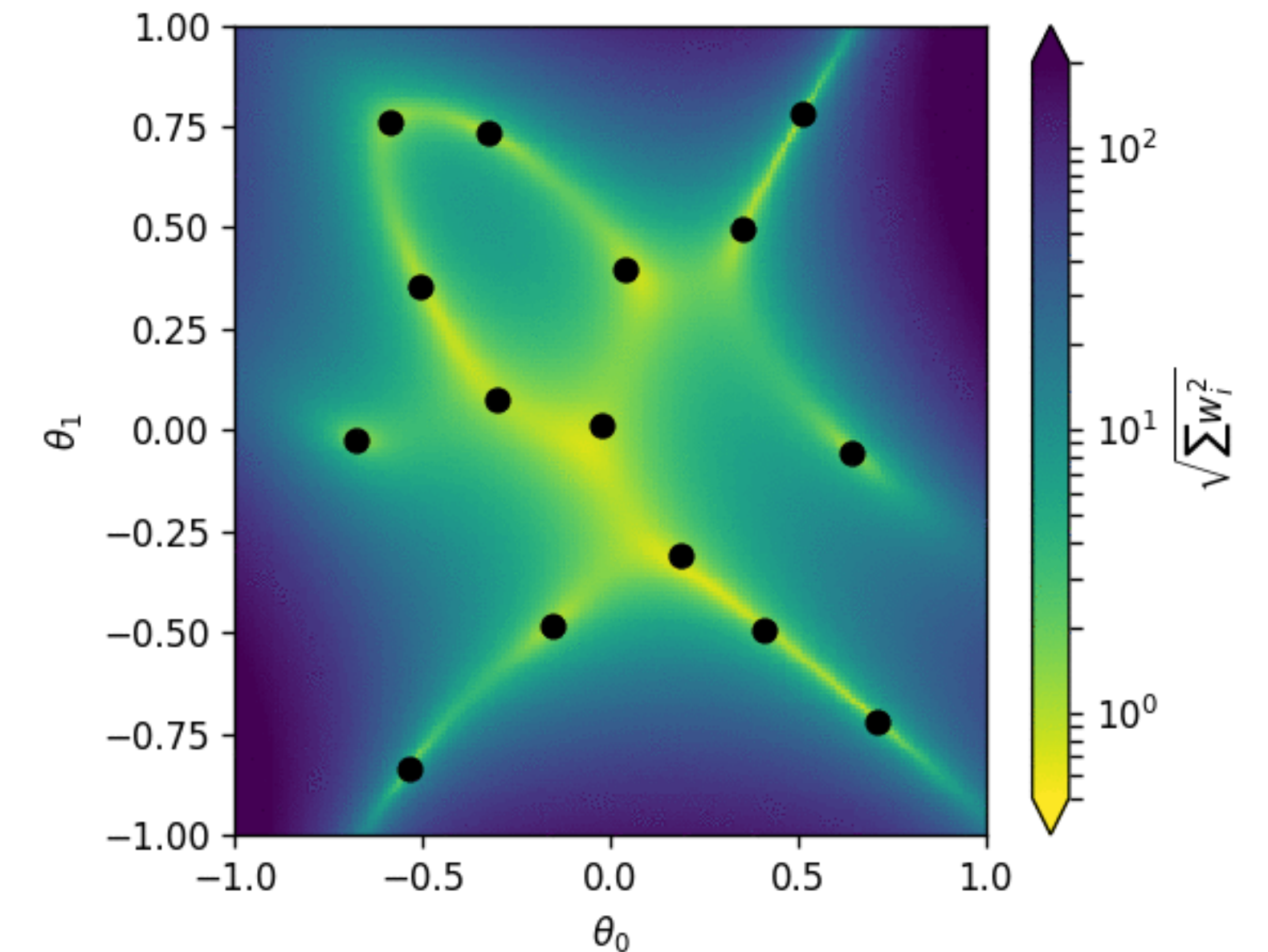
Perfect match for EFT measurements



- 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 (SMALLY?)



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

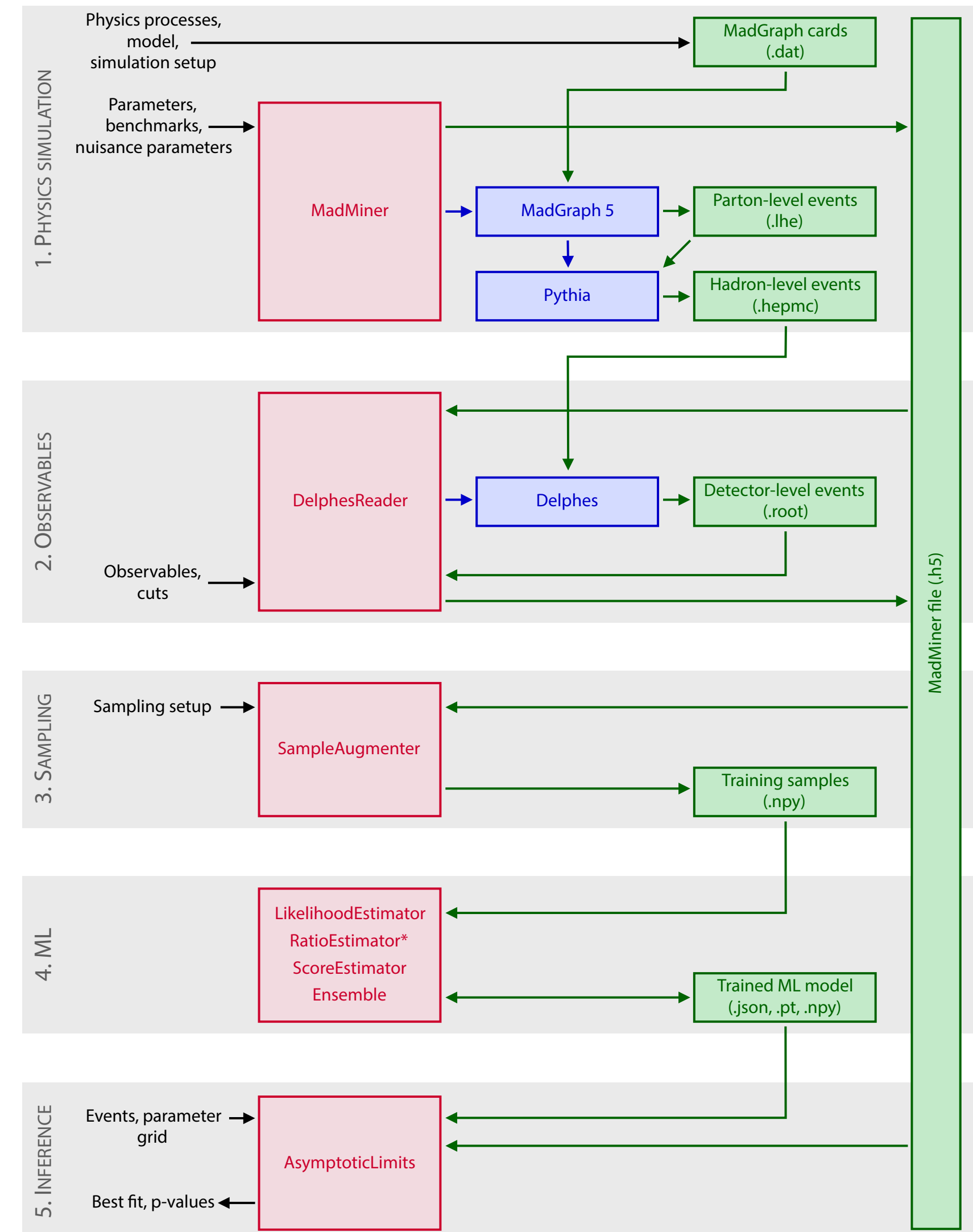
MadMiner

[JB, F. Kling, I. Espejo, K. Cranmer doi: 10.5281/zenodo.1489147]

Automization

We are developing **MadMiner**, which makes it straightforward to apply the new techniques to LHC problems

- Out of the box: Pheno-level analyses
 - MadGraph, Pythia, Delphes
 - 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

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

Mining gold from MadGraph to improve limit setting in particle physics.

Note that this is an early development version. Do not expect anything to be stable. If you have any questions, please contact us at johann.brehmer@nyu.edu.

Note: Consider working with the `future` branch, where we develop a refactored and cleaned up version of MadMiner. The API and even the file format for trained neural networks is different from the `master` version (v0.2.x). We expect these changes to make it into MadMiner v0.3.0.

pypi package 0.2.8
docs passing
build error
docker pulls 2k
launch binder
code style black
License MIT
DOI 10.5281/zenodo.1489147

Introduction

Particle physics processes are usually modelled with complex Monte-Carlo simulations of the hard process, parton shower,

MadMiner latest

Docs » MadMiner [Edit on GitHub](#)

Search docs

MadMiner

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

An inference toolkit for LHC measurements

Note that this is a development version. Do not rely on anything being stable. If you have any questions, please contact us at johann.brehmer@nyu.edu.

Sites

- Introduction to MadMiner
- Getting started
- Using MadMiner
- References

Reference

madminer.core module
 madminer.delphes module
 madminer.fisherinformation module
 madminer.lhe module
 madminer.ml module
 madminer.morphing module
 madminer.plotting module

Repository and tutorials:
github.com/johannbrehmer/madminer

Documentation:
madminer.readthedocs.io

Search projects

Help Donate Login Register

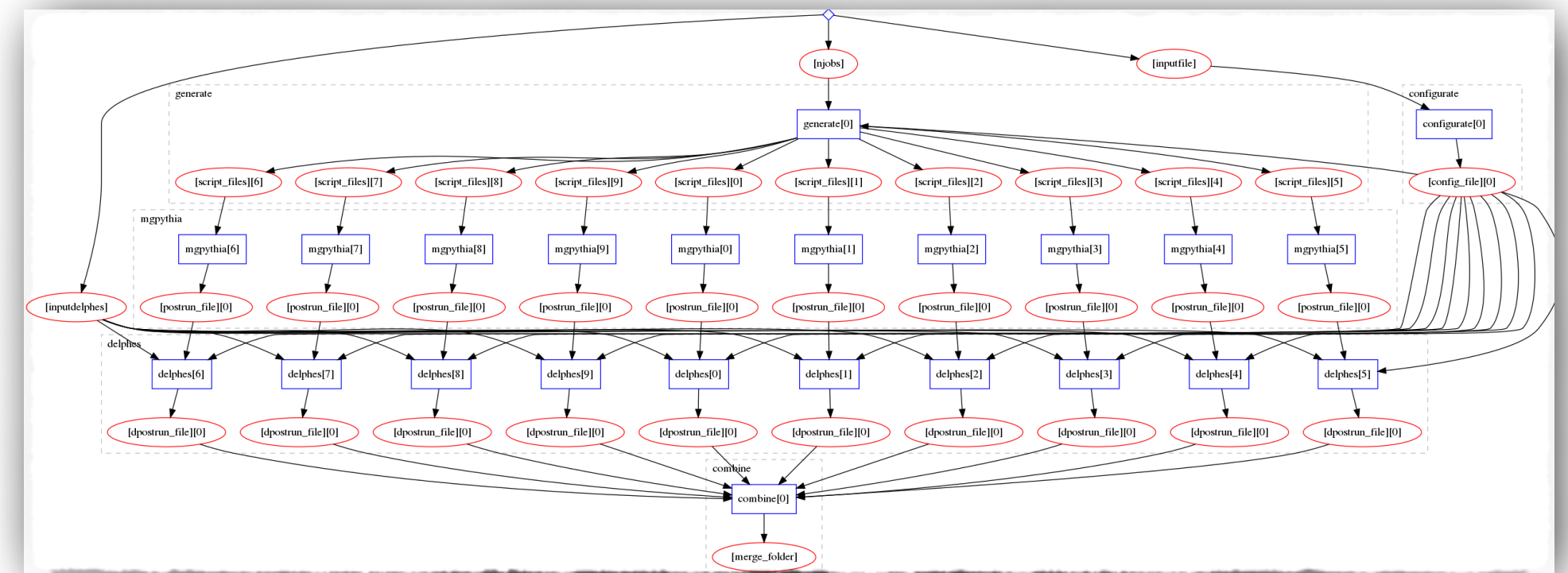
madminer 0.2.8

pip install madminer

Latest version

Last released: Feb 28, 2019

Installation:
[pip install madminer](https://pip.pypa.io/en/stable/cli/commands/install/)

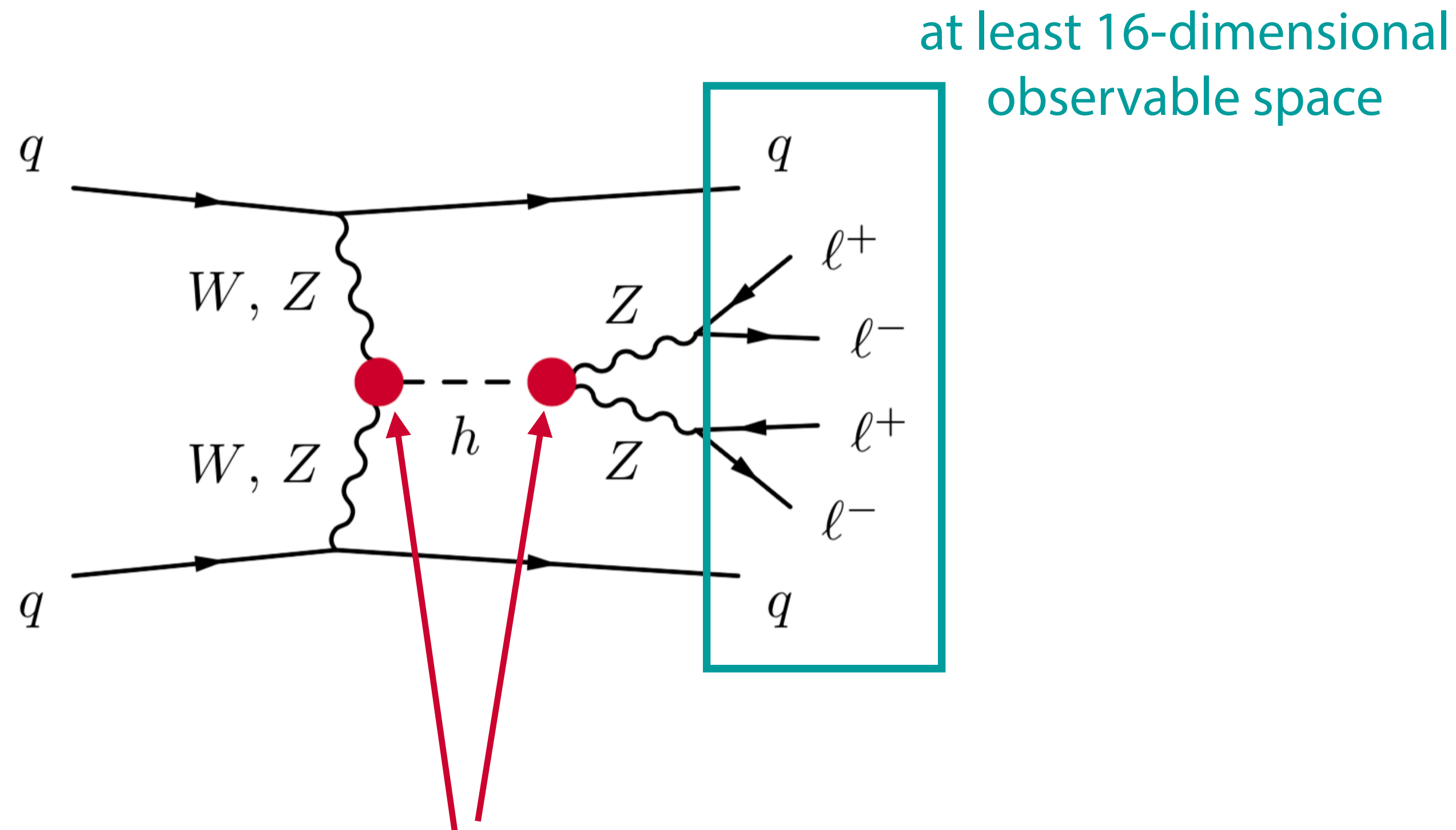


Deployment with Docker, yadage, REANA:
github.com/irinaespejo/workflow-madminer

In the wild

Proof of concept: Higgs production in weak boson fusion

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



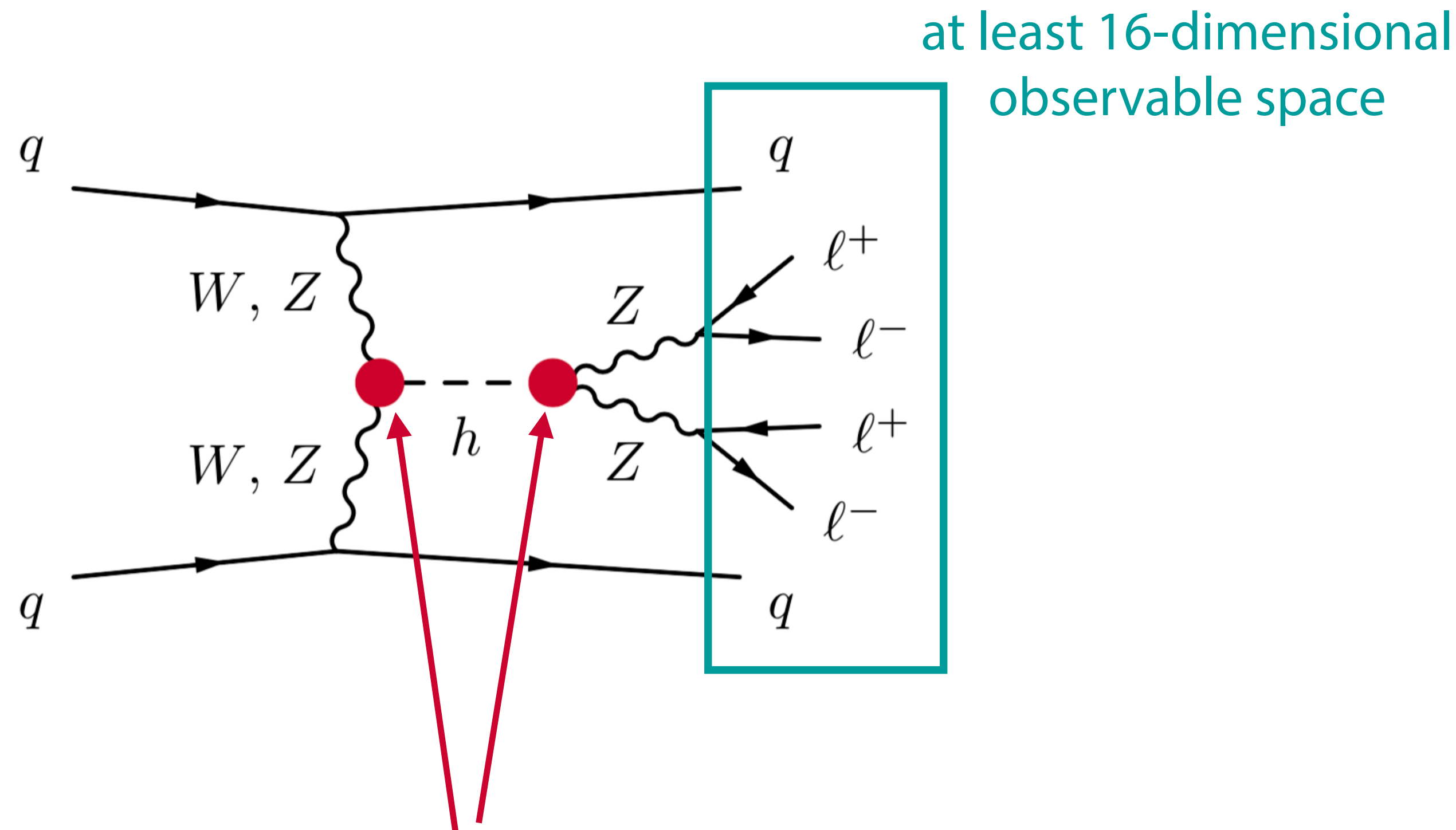
Exciting new physics might hide here!

We parameterize it with two EFT coefficients:

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

Proof of concept: Higgs production in weak boson fusion

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



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

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

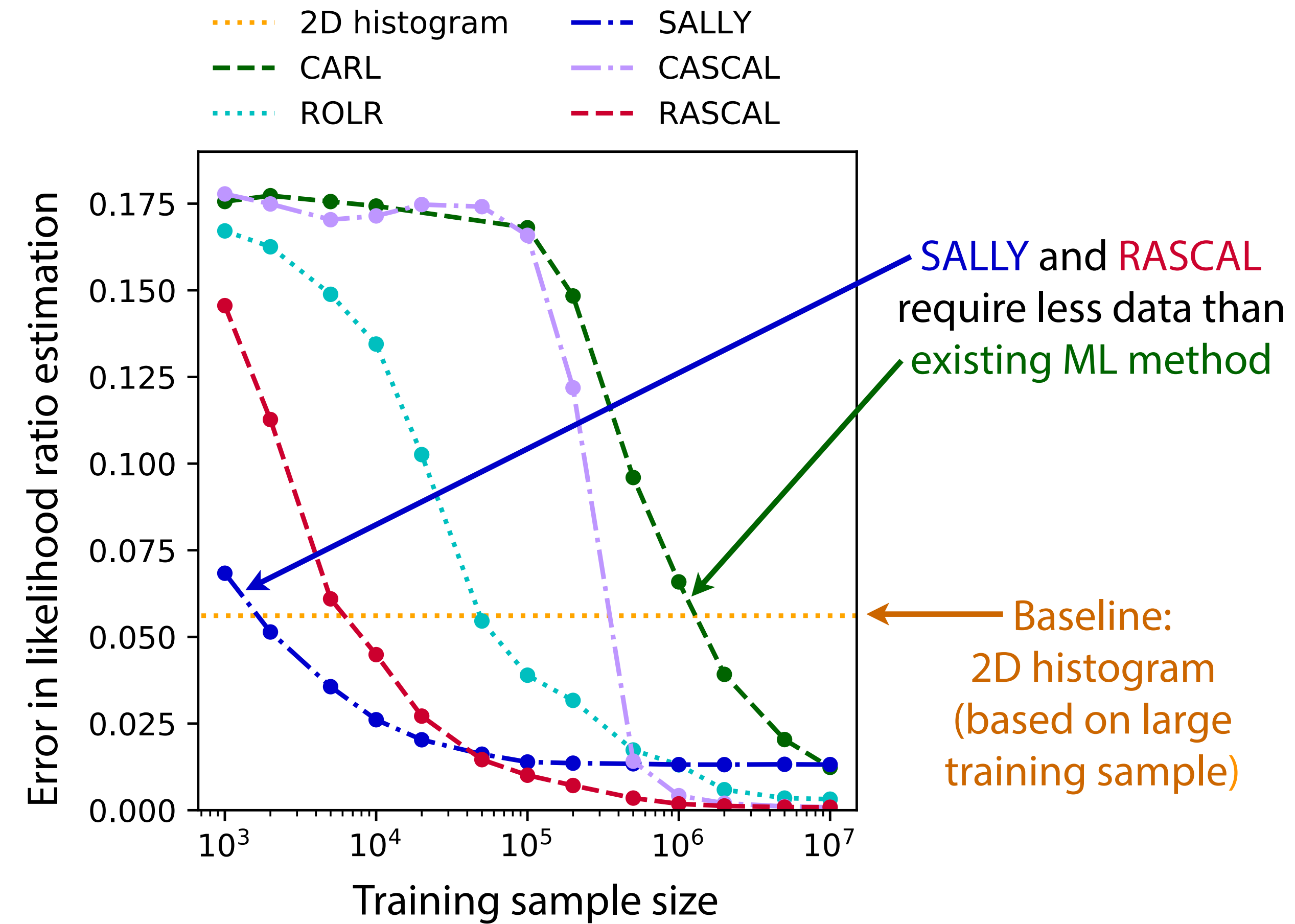
Goal: constrain the **two EFT parameters**

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

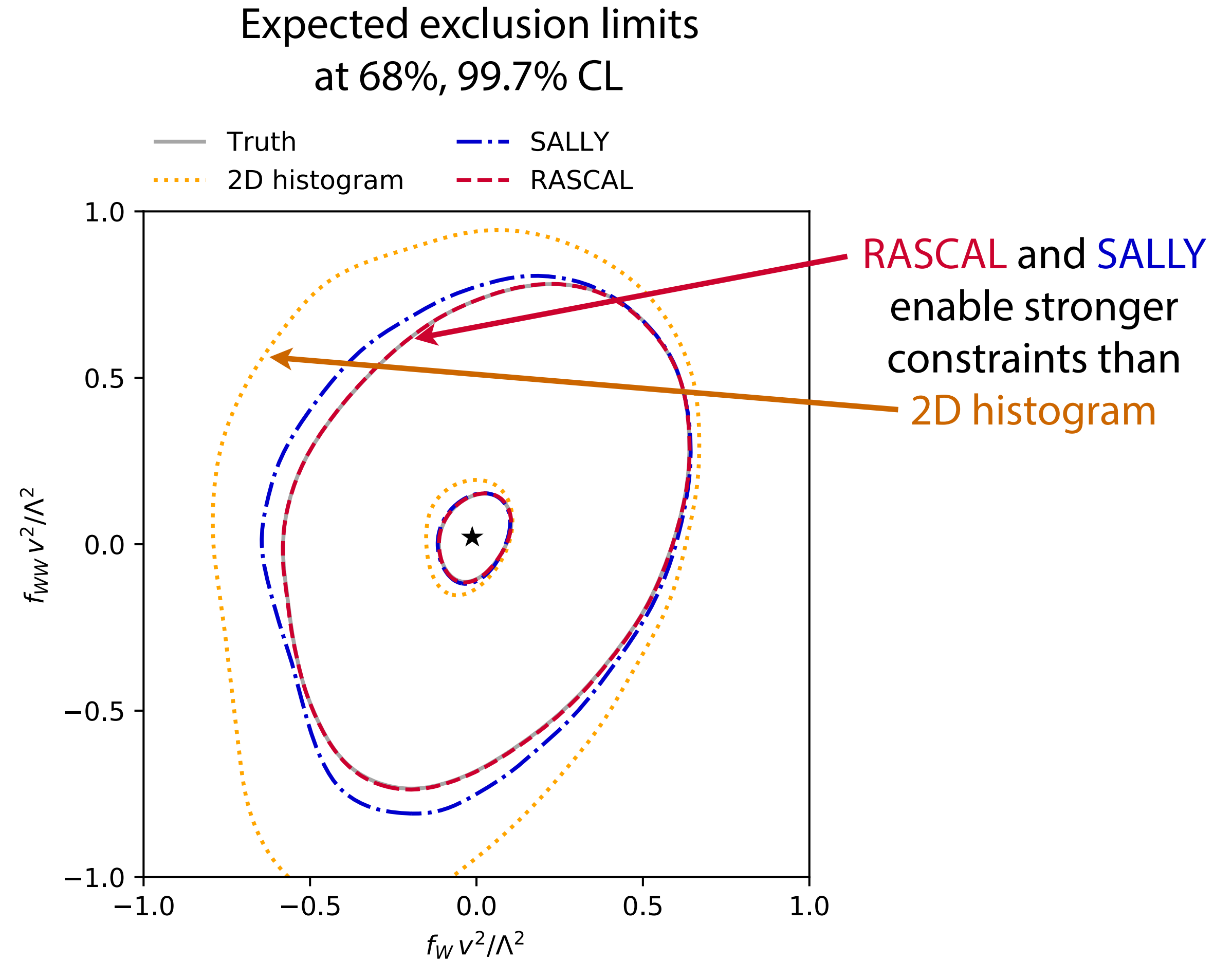
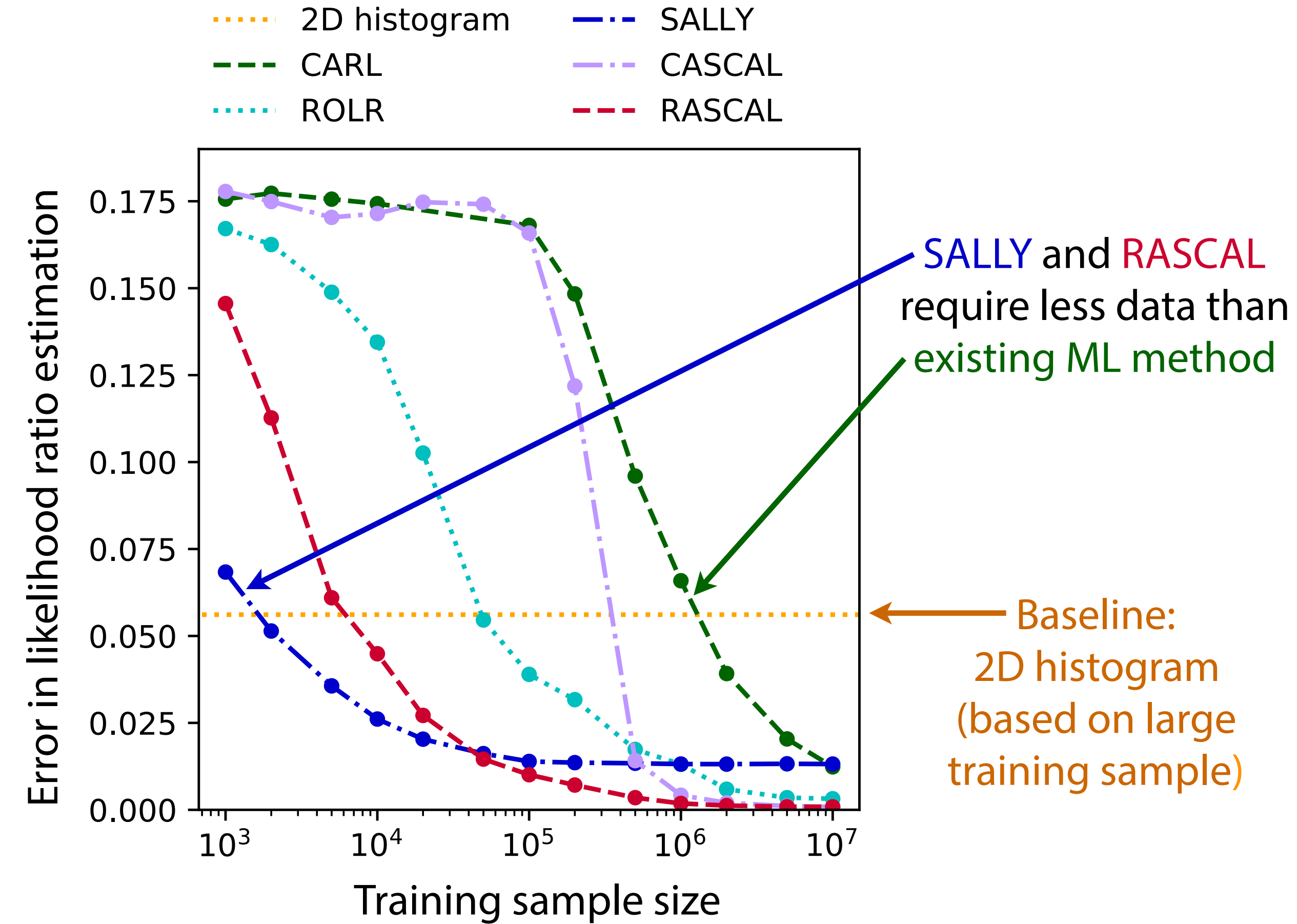
Two scenarios:

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

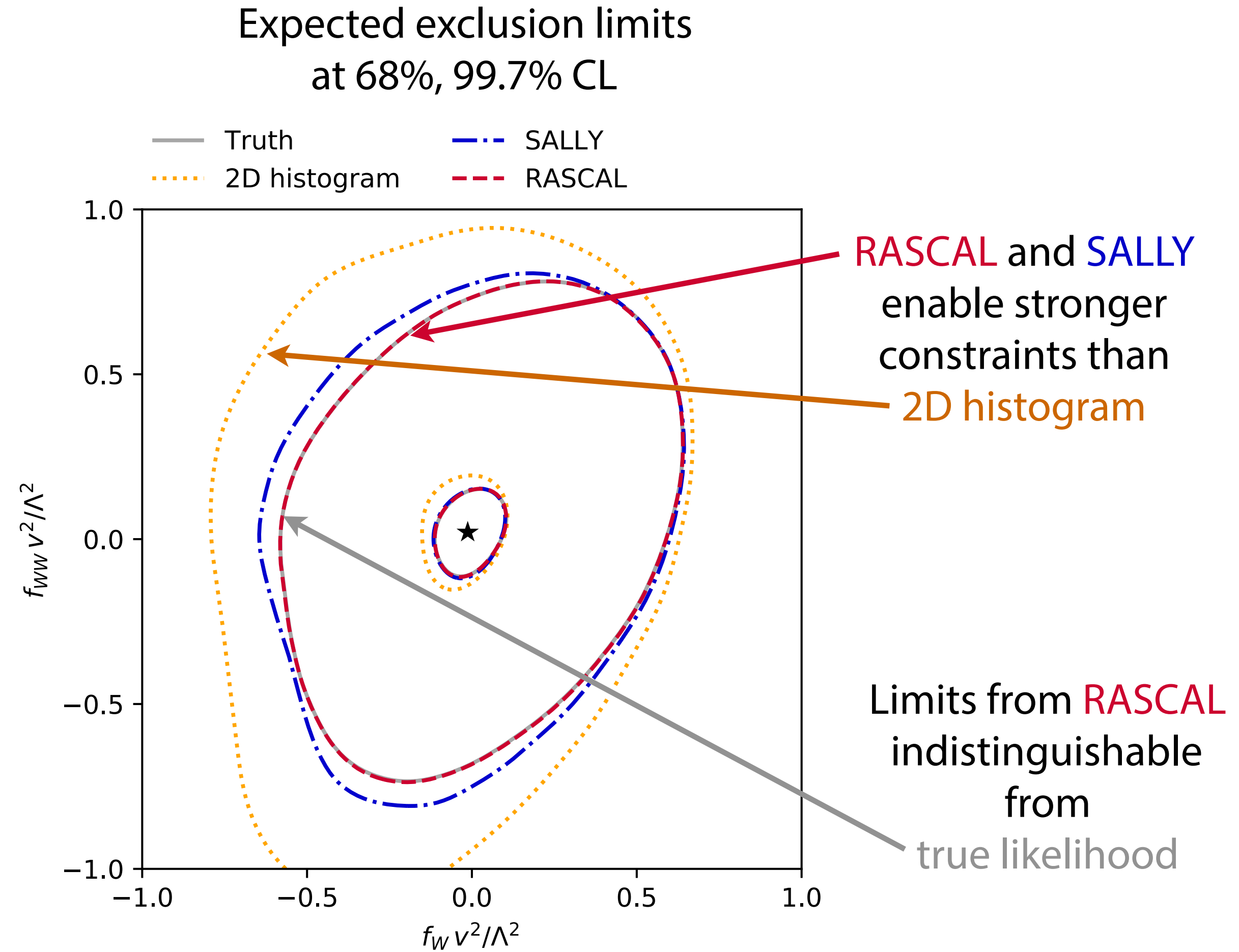
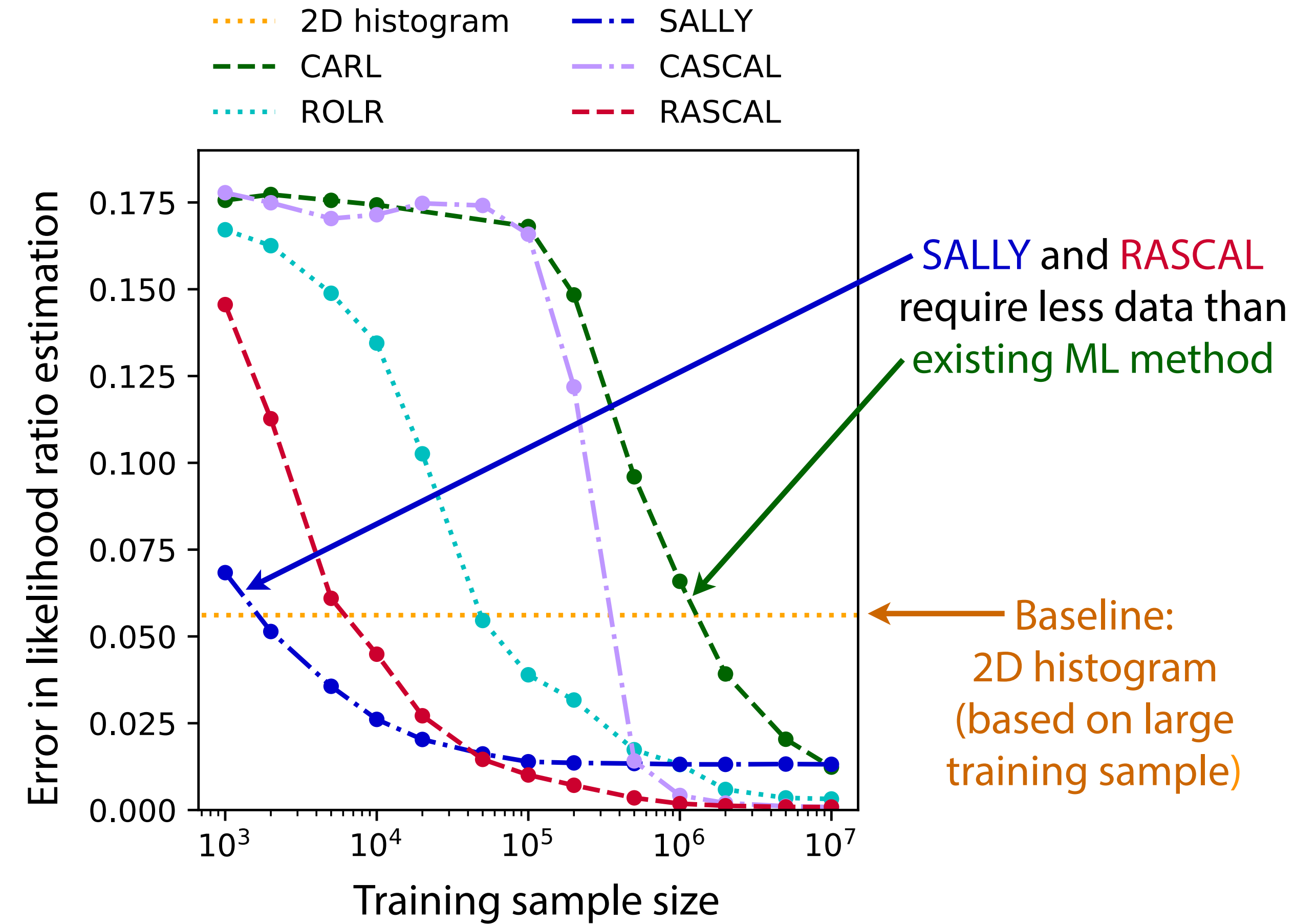
Proof of concept: Stronger constraints with less training data



Proof of concept: Stronger constraints with less training data



Proof of concept: Stronger constraints with less training data



Diboson production

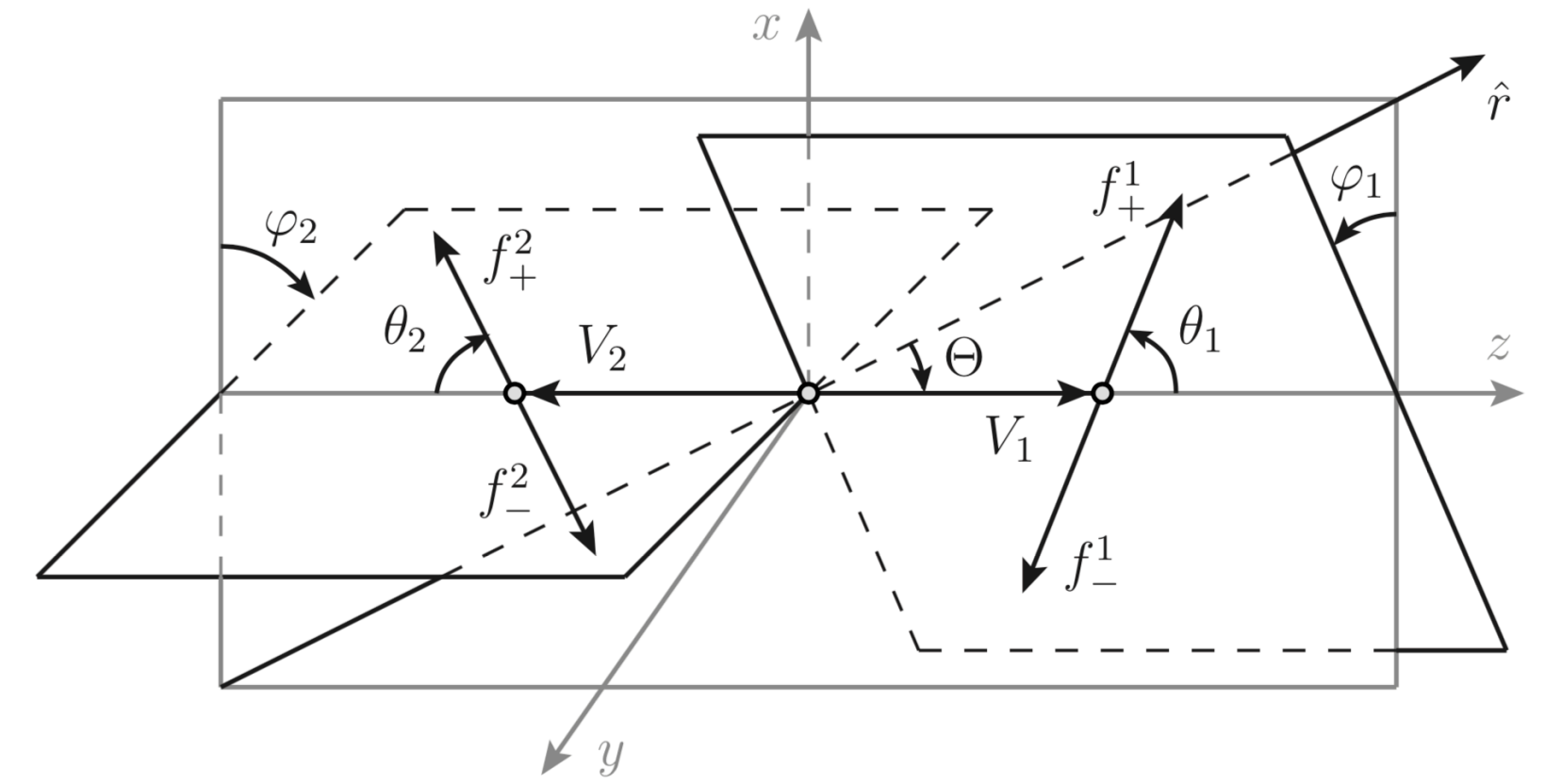
- In inclusive observables, the interference between SM and new physics amplitudes vanishes

⇒ 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]



Diboson production

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

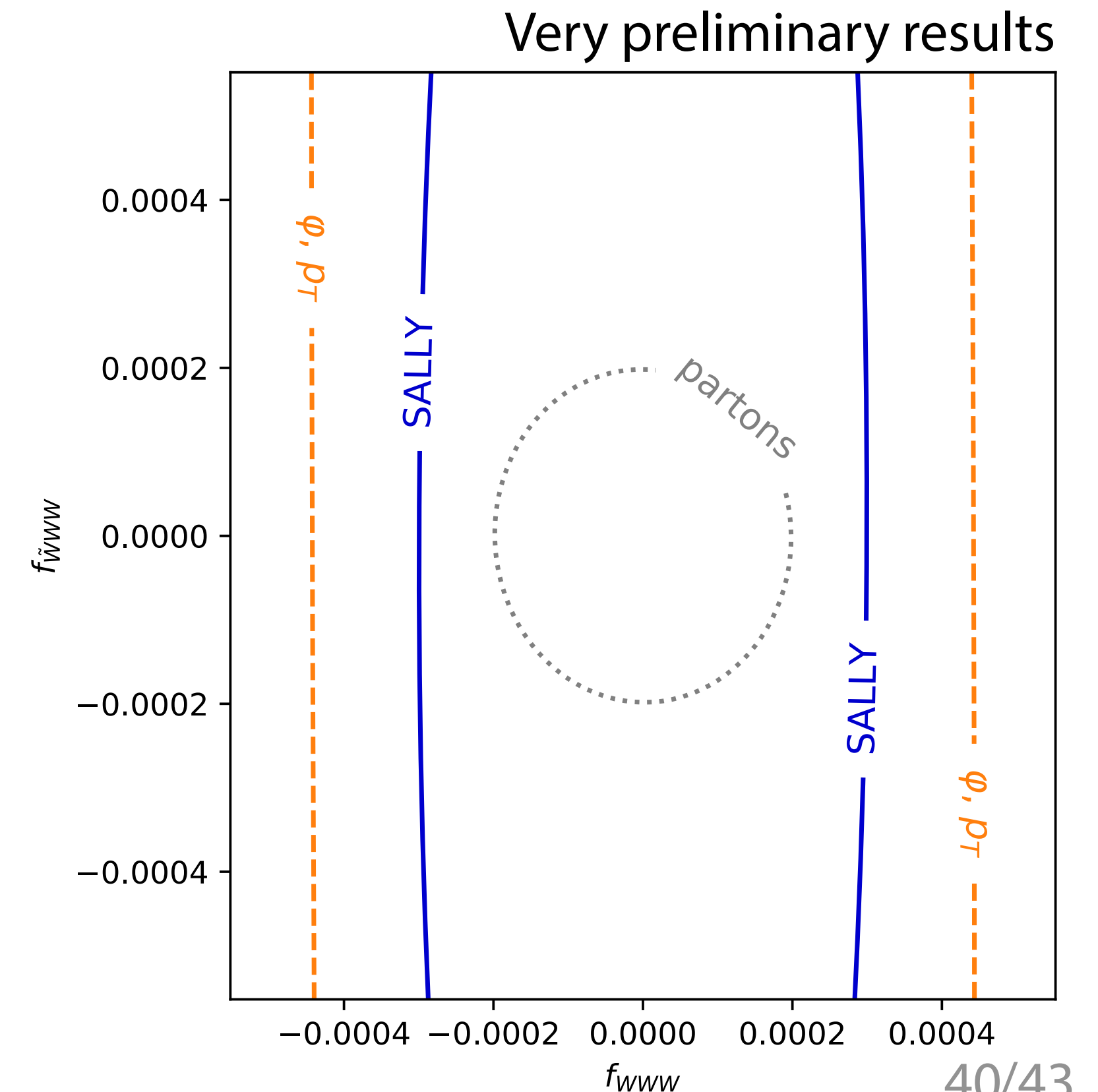
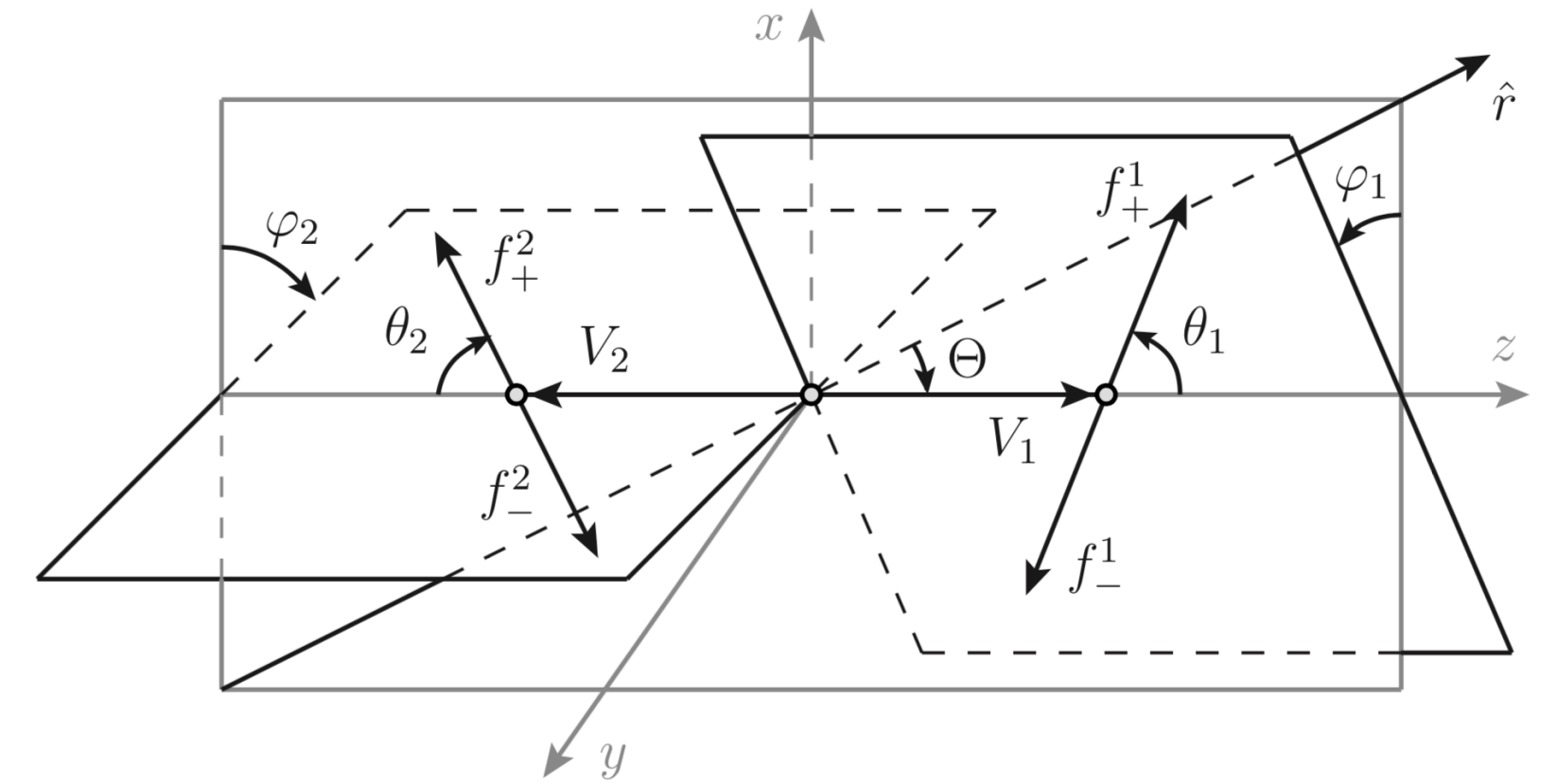
⇒ Reduced sensitivity to new physics

- “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 $W\gamma \rightarrow l\nu\gamma$
[JB, K. Cranmer, M. Farina, F. Kling, D. Pappadopulo, J. Ruderman in progress]

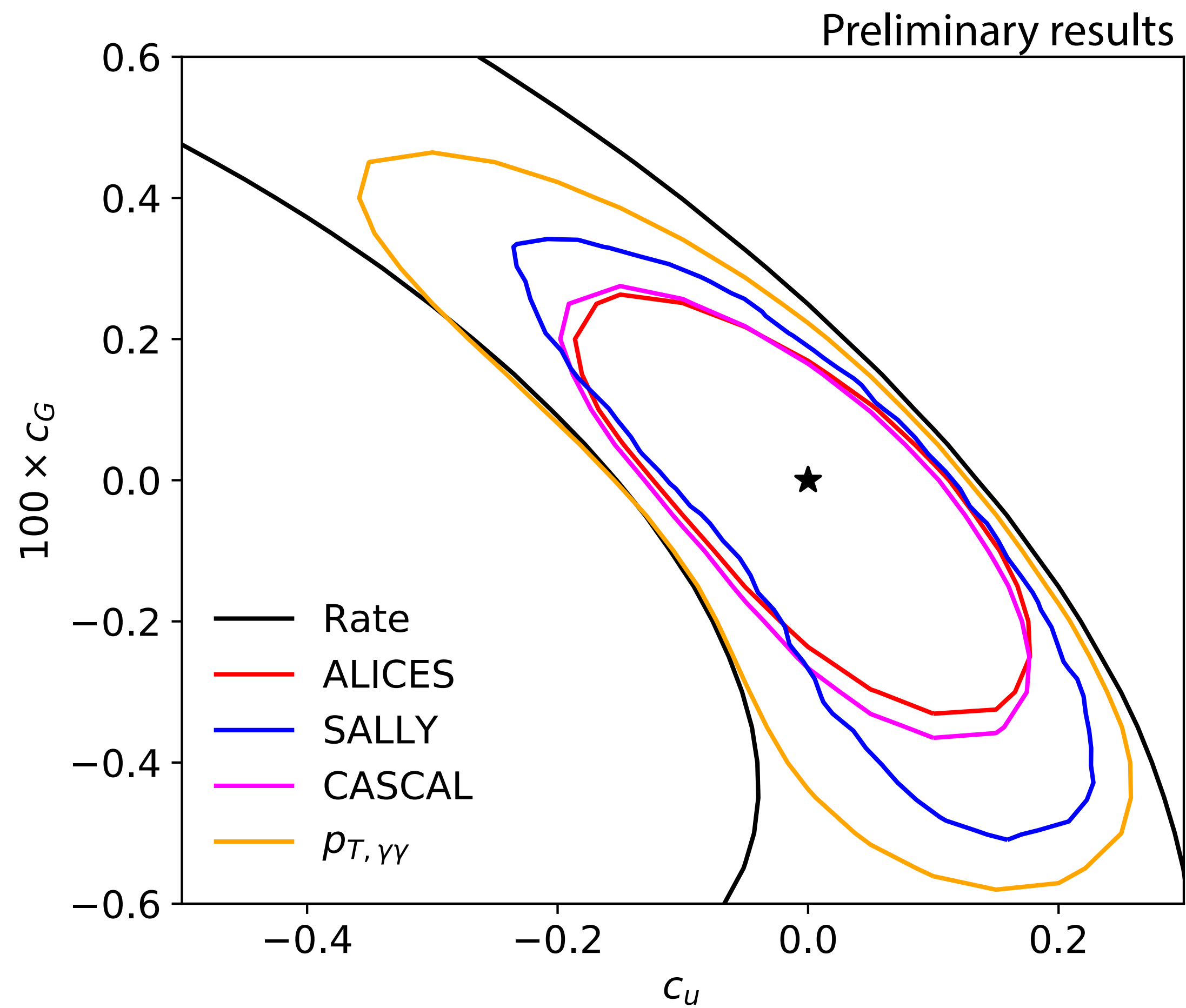
- Preliminary results: we can extract more information when we **analyze events with SALLY** than with **histograms of φ and standard observables**



Higgs measurements

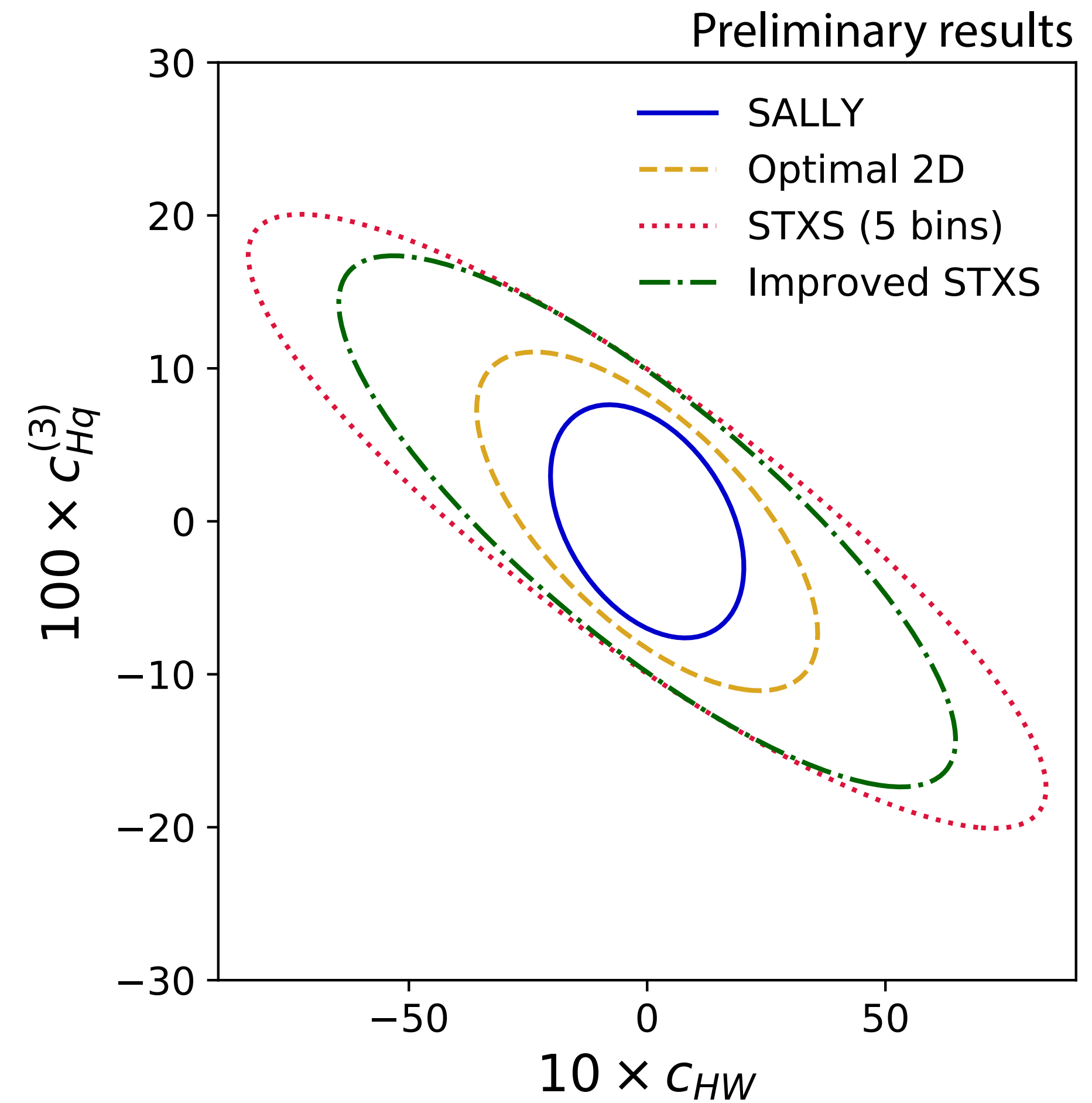
$$t\bar{t}h \rightarrow (bl^+\nu)(bl^-\bar{\nu})(\gamma\gamma)$$

[JB, F. Kling, I. Espejo, K. Cranmer in progress]

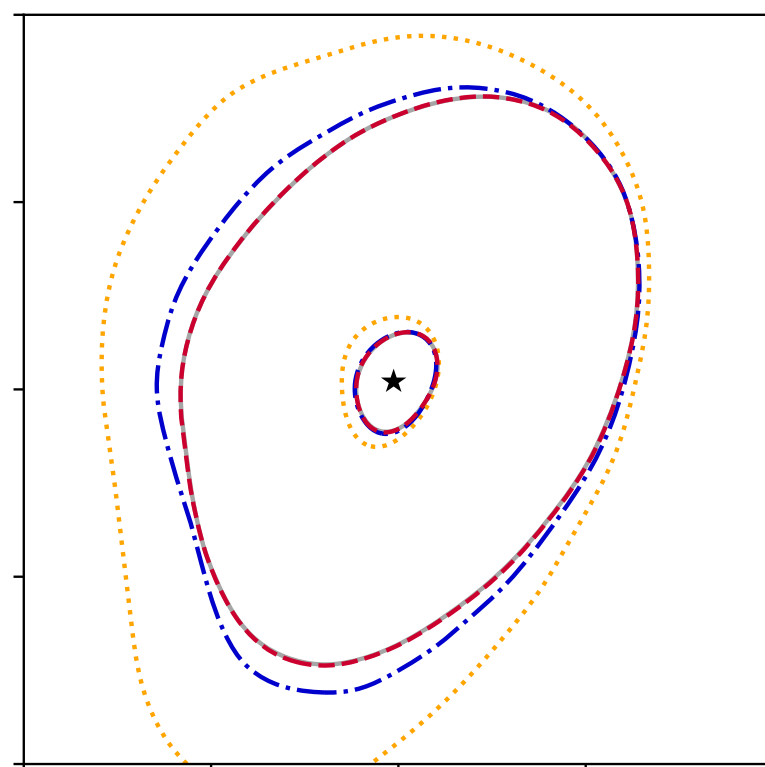
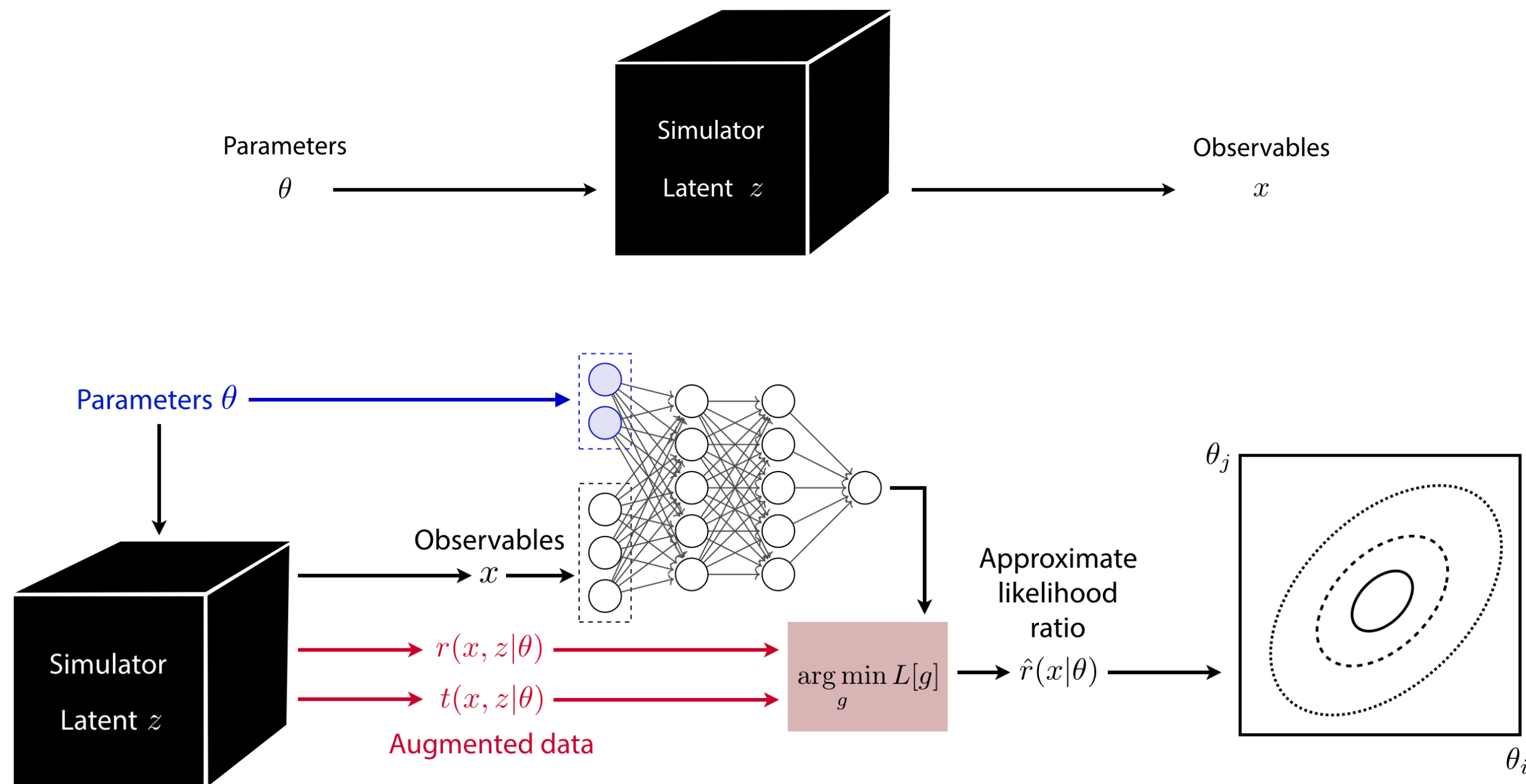


$$Wh \rightarrow (\ell\nu)(b\bar{b})$$

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



A new approach to LHC measurements



- LHC measurements with high-dimensional observations have “likelihood-free” structure
- New multivariate inference techniques: Leverage information in matrix elements + power of machine learning to...
 - estimate the full likelihood function
 - learn optimal summary statistics
- First tests show potential to substantially increase sensitivity to new physics

References



Kyle Cranmer



Gilles Louppe



Juan Pavez



Markus Stoye



Felix Kling

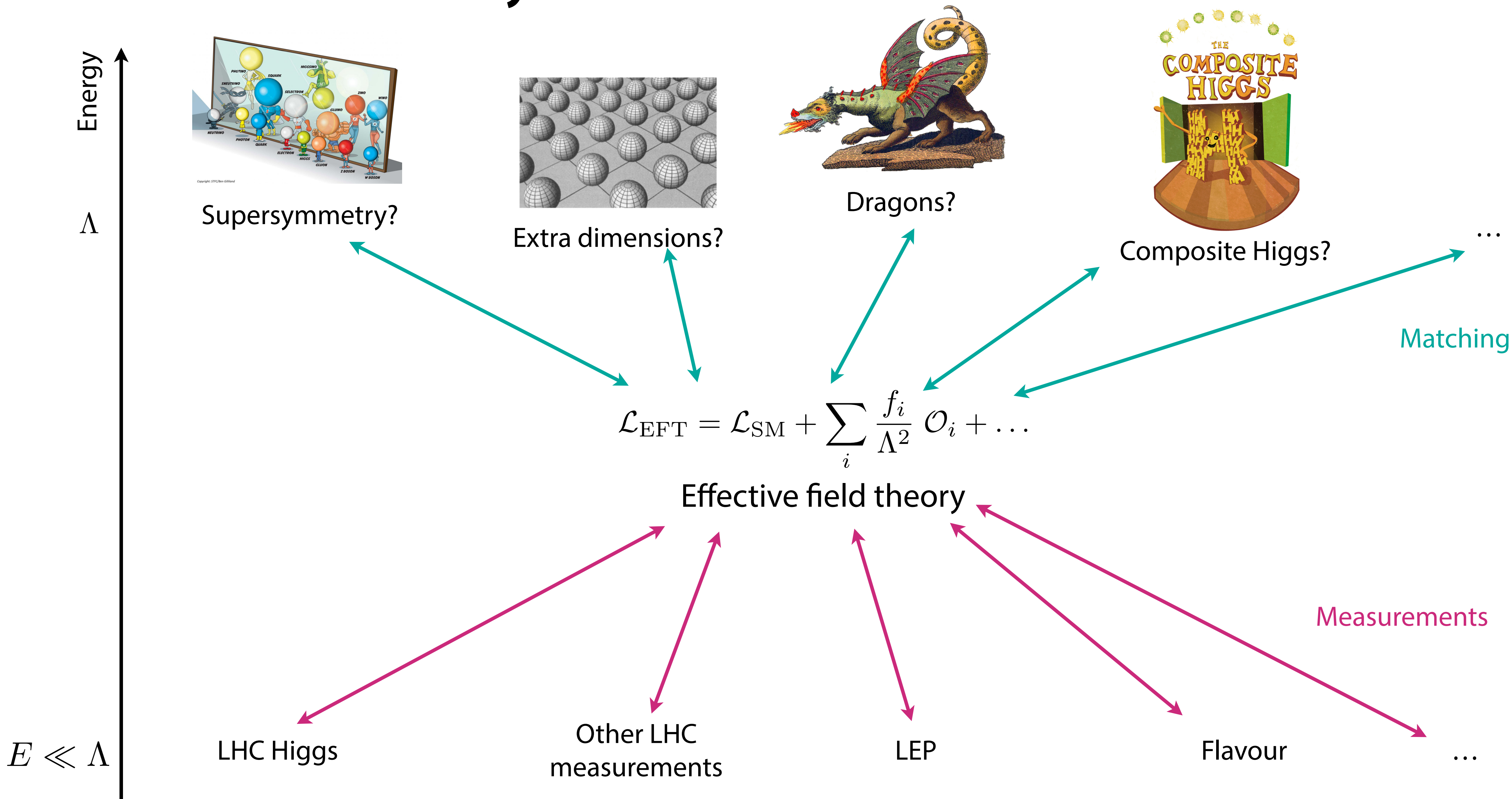


Irina Espejo

- JB, KC, GL, JP:** Constraining Effective Field Theories with machine learning [PRL, 1805.00013]
- JB, KC, GL, JP:** A guide to constraining Effective Field Theories with machine learning [PRD, 1805.00020]
- JB, GL, JP, KC:** Mining gold from implicit models to improve likelihood-free inference [1805.12244]
- MS, JB, GL, JP, KC:** Likelihood-free inference with an improved cross-entropy estimator [1808.00973]
- JB, KC, IE, FK, GL, JP:** Effective LHC measurements with matrix elements and machine learning [ACAT, 1906.01578]
- JB, FK, IE, KC:** MadMiner: An inference toolkit for particle physics [doi: 10.5281/zenodo.1489147]

Bonus material

Effective field theory



SMEFT (Standard Model Effective Field Theory)

[W. Buchmuller, D. Wyler 85;

B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek 1008.4884; ...]

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

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

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

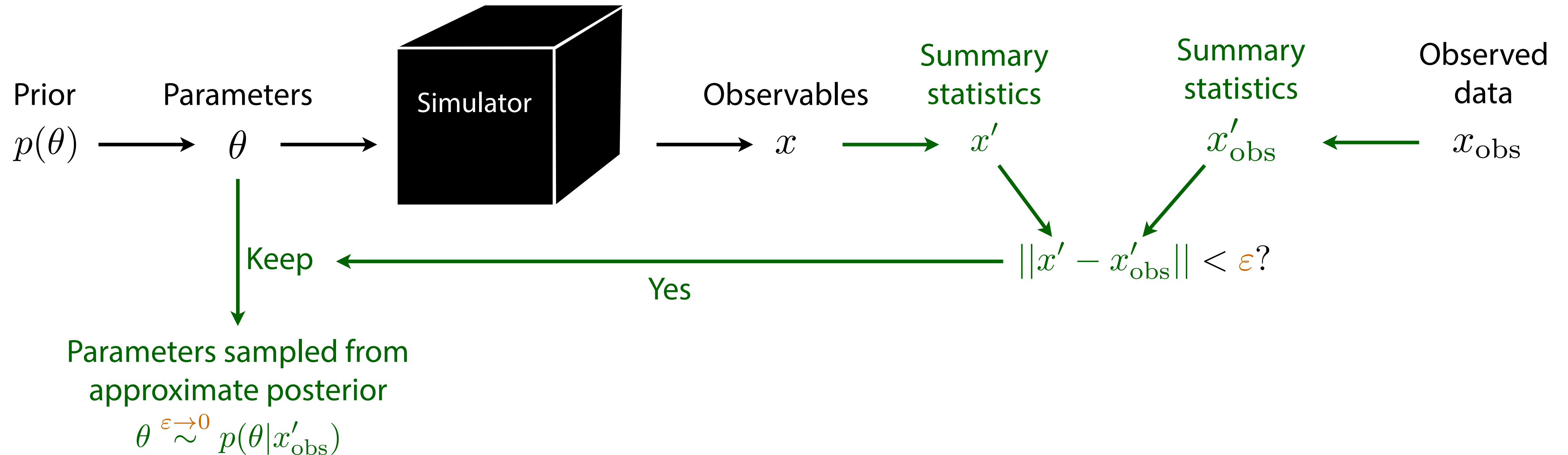
Higher-order terms

- suppressed by additional factors of E^2 / Λ^2

$\mathcal{O}_{\phi,1} = (D_\mu \phi)^\dagger \phi \phi^\dagger D^\mu \phi$	$\mathcal{O}_{GG} = (\phi^\dagger \phi) G_{\mu\nu}^a G^{\mu\nu a}$
$\mathcal{O}_{\phi,2} = \frac{1}{2} \partial_\mu (\phi^\dagger \phi) \partial^\mu (\phi^\dagger \phi)$	$\mathcal{O}_{BB} = -\frac{g'^2}{4} (\phi^\dagger \phi) B_{\mu\nu} B^{\mu\nu}$
$\mathcal{O}_{\phi,3} = \frac{1}{3} (\phi^\dagger \phi)^3$	$\mathcal{O}_{WW} = -\frac{g^2}{4} (\phi^\dagger \phi) W_{\mu\nu}^a W^{\mu\nu a}$
$\mathcal{O}_{\phi,4} = (\phi^\dagger \phi) (D_\mu \phi)^\dagger D^\mu \phi$	$\mathcal{O}_{BW} = -\frac{g g'}{4} (\phi^\dagger \sigma^a \phi) B_{\mu\nu} W^{\mu\nu a}$
	$\mathcal{O}_B = \frac{i g'}{2} (D^\mu \phi)^\dagger D^\nu \phi B_{\mu\nu}$
	$\mathcal{O}_W = \frac{i g}{2} (D^\mu \phi)^\dagger \sigma^a D^\nu \phi W_{\mu\nu}^a$

Approximate Bayesian Computation (ABC)

[D. Rubin 1984]



- How to choose x' ?
- How to choose ϵ ?
- No tractable posterior
- Need to run new simulations for new data or new prior

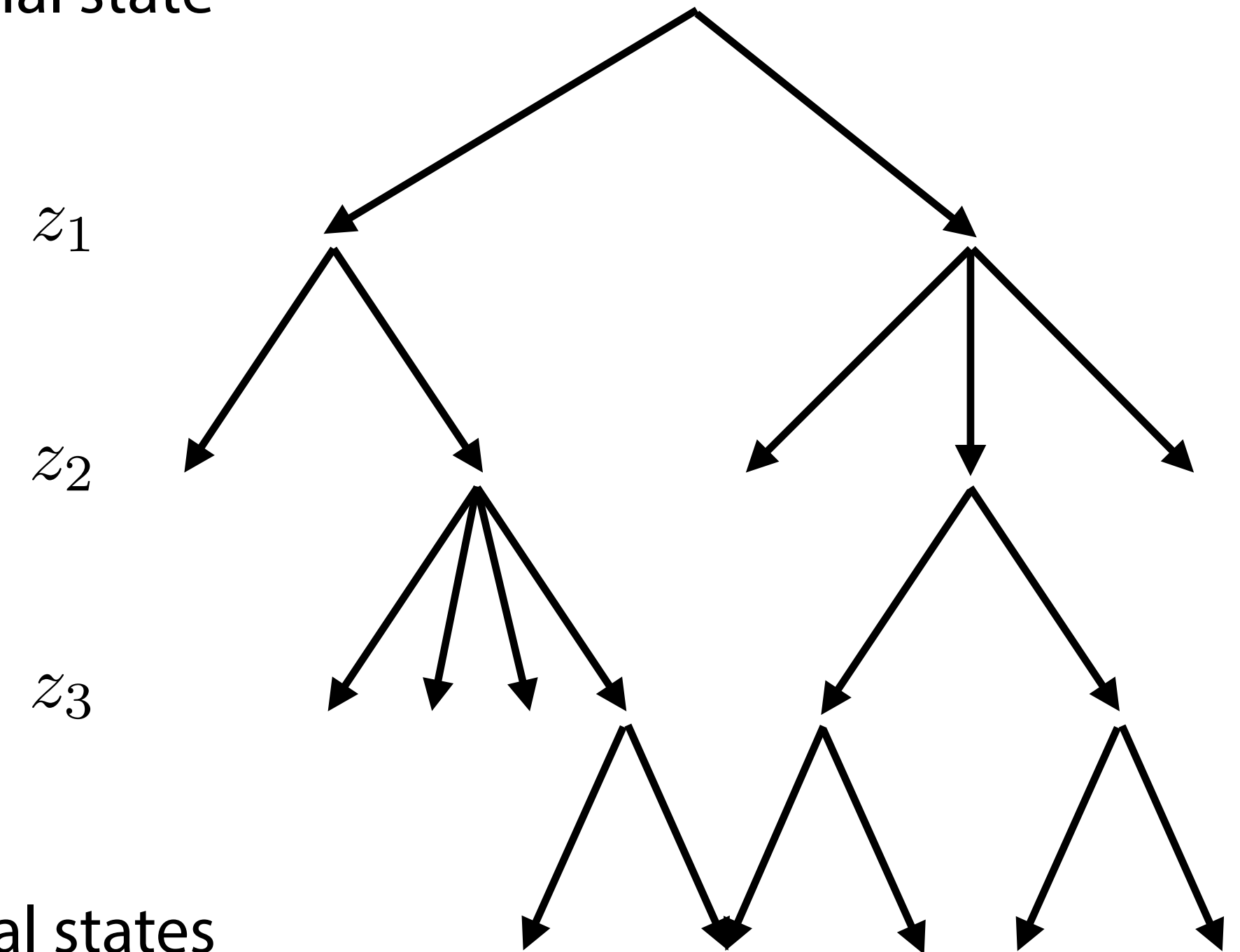
“Curse of dimensionality”
Precision vs efficiency tradeoff

Extracting the joint likelihood ratio from any simulation

- Computer simulation typically evolve along a tree-like structure of successive random branchings
- The probabilities of each branching $p_i(z_i|z_{i-1}, \theta)$ are often clearly defined in the code:

```
if random() > 0.1 + 2.5 * model_parameter:  
    do_one_thing()  
else:  
    do_another_thing()
```

Initial state



Final states

Extracting the joint likelihood ratio from any simulation

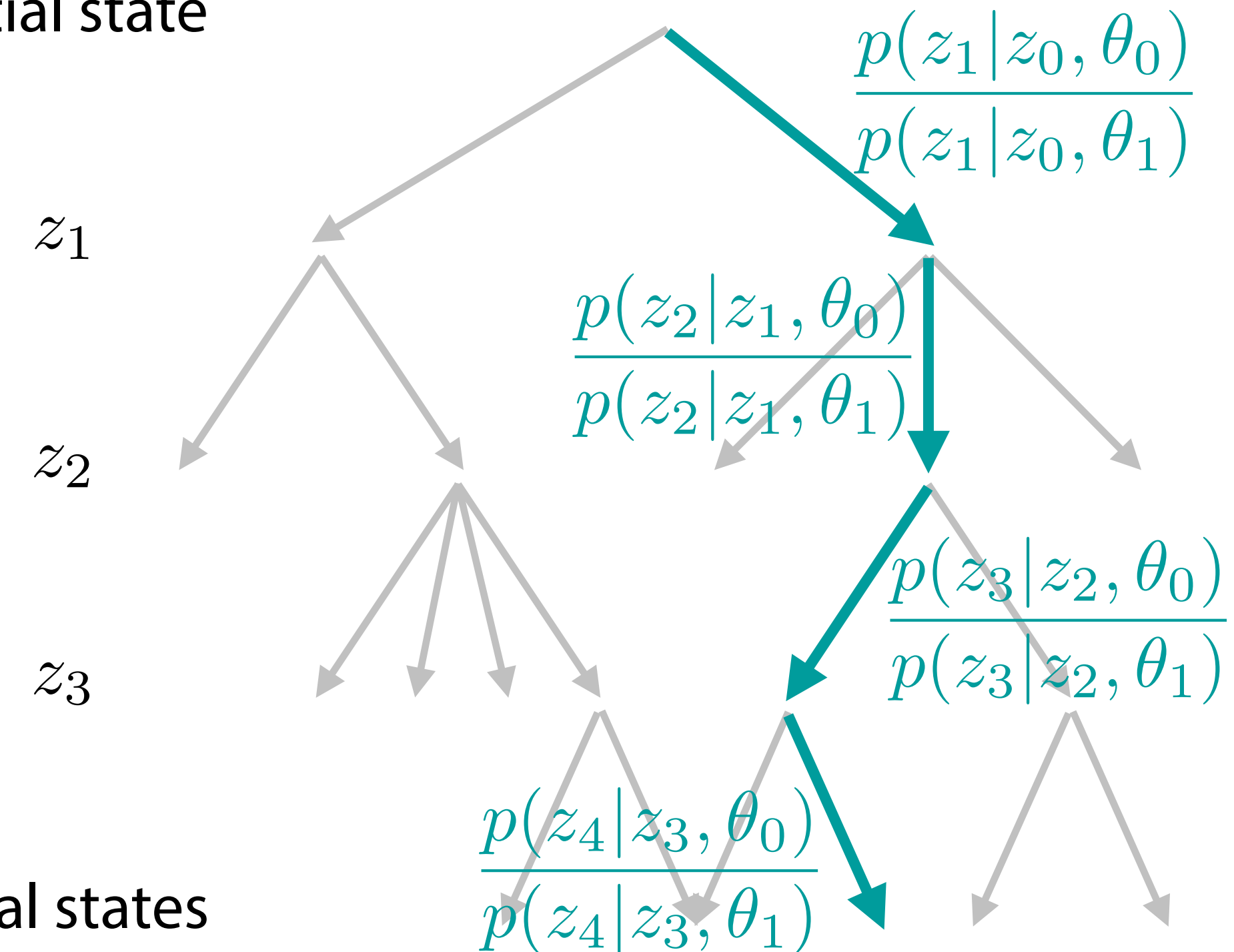
- Computer simulation typically evolve along a tree-like structure of successive random branchings
- The probabilities of each branching $p_i(z_i|z_{i-1}, \theta)$ are often clearly defined in the code:

```
if random() > 0.1 + 2.5 * model_parameter:  
    do_one_thing()  
else:  
    do_another_thing()
```

- For each run of the simulator, we can calculate the probability of the chosen path for different values of the parameters, and the “joint likelihood ratio”:

$$r(x, z|\theta_0, \theta_1) = \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} = \prod_i \frac{p(z_i|z_{i-1}, \theta_0)}{p(z_i|z_{i-1}, \theta_1)}$$

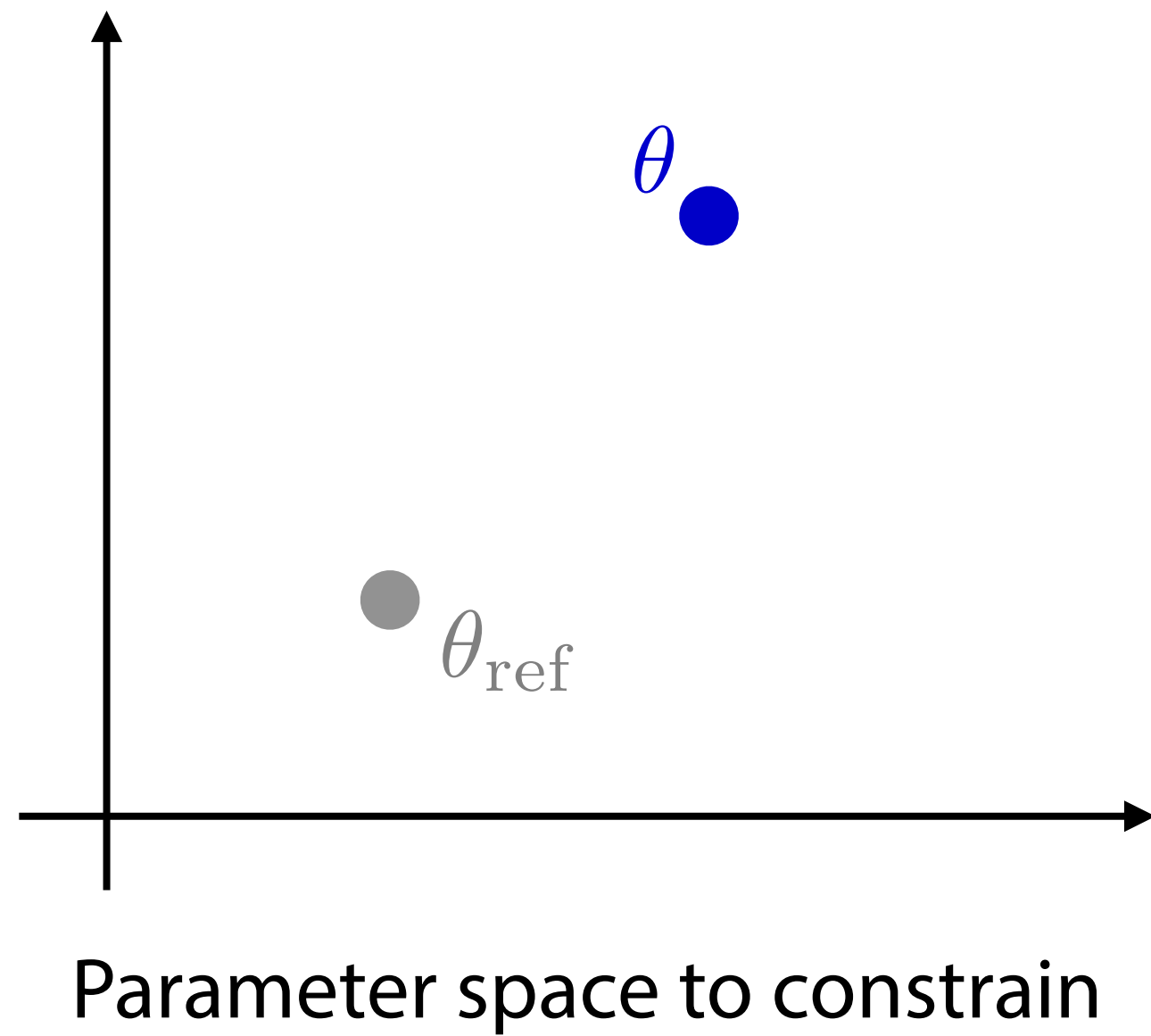
Initial state



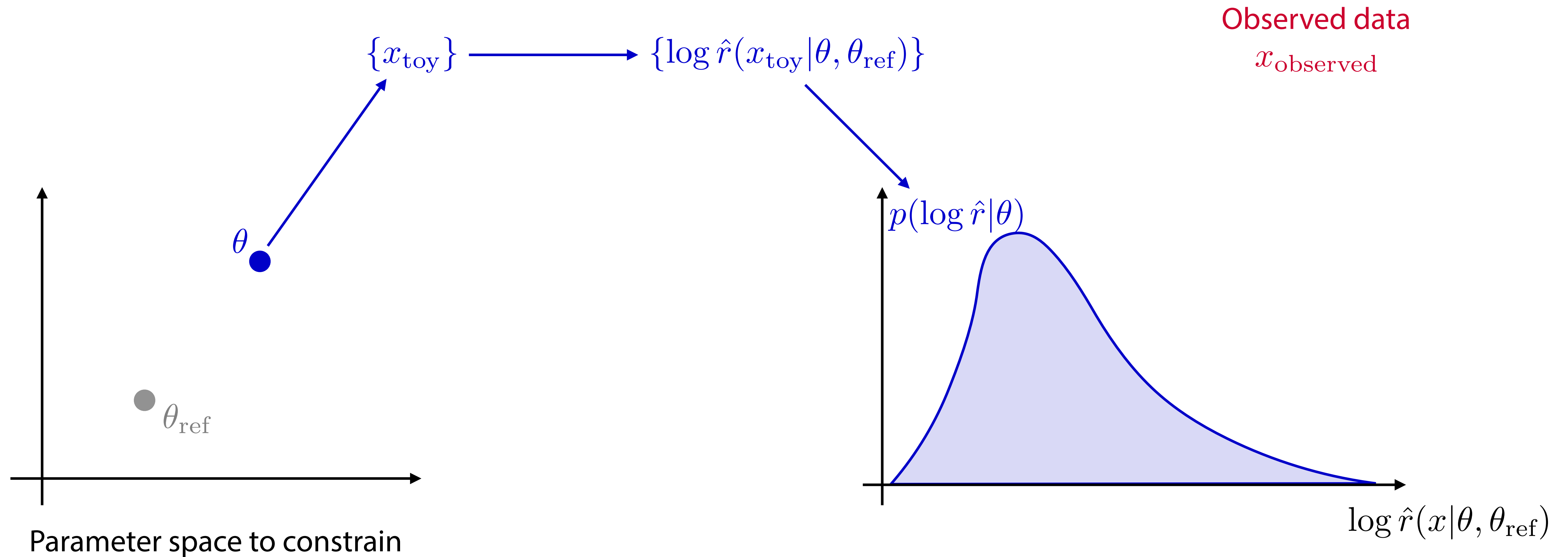
Limit setting (frequentist, standard ATLAS / CMS practice)

Observed data

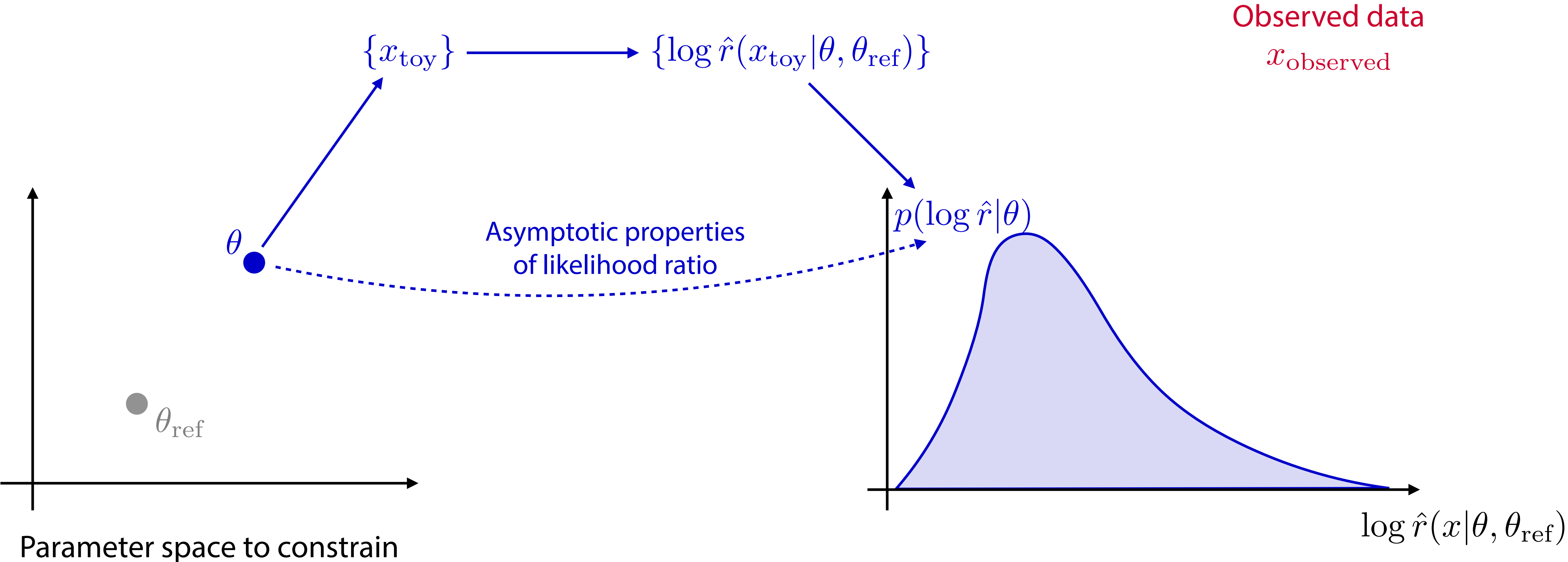
x_{observed}



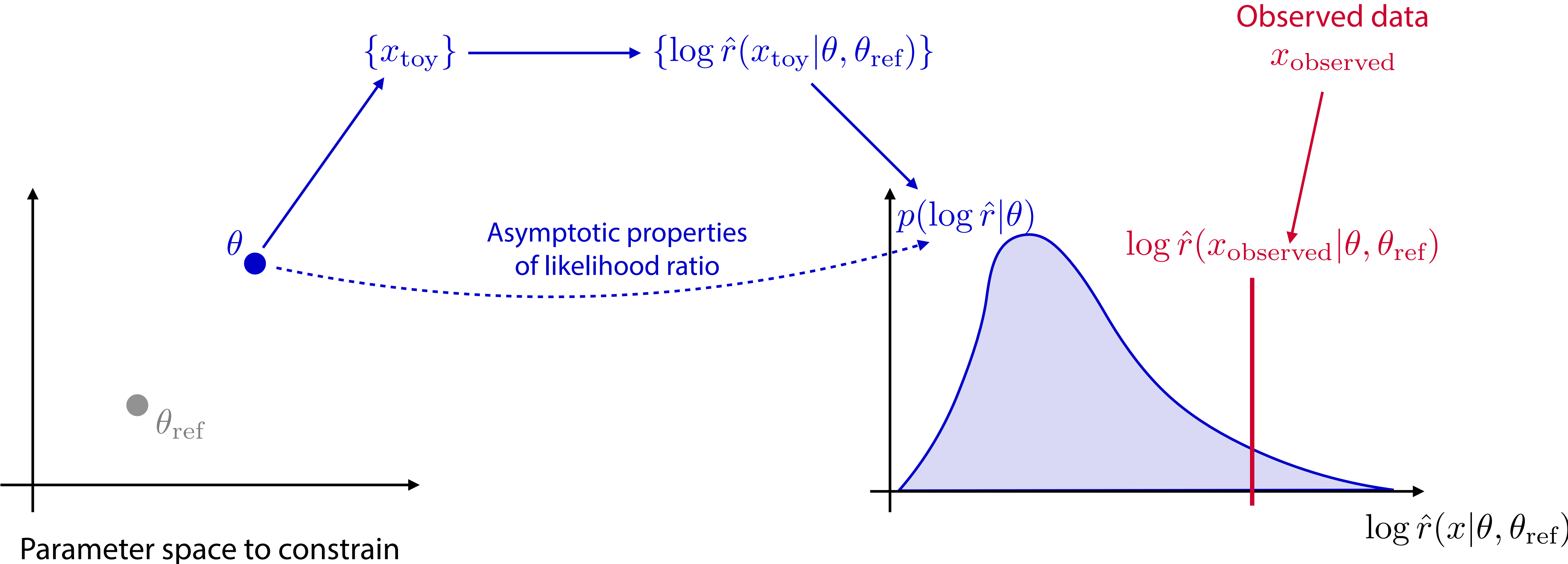
Limit setting (frequentist, standard ATLAS / CMS practice)



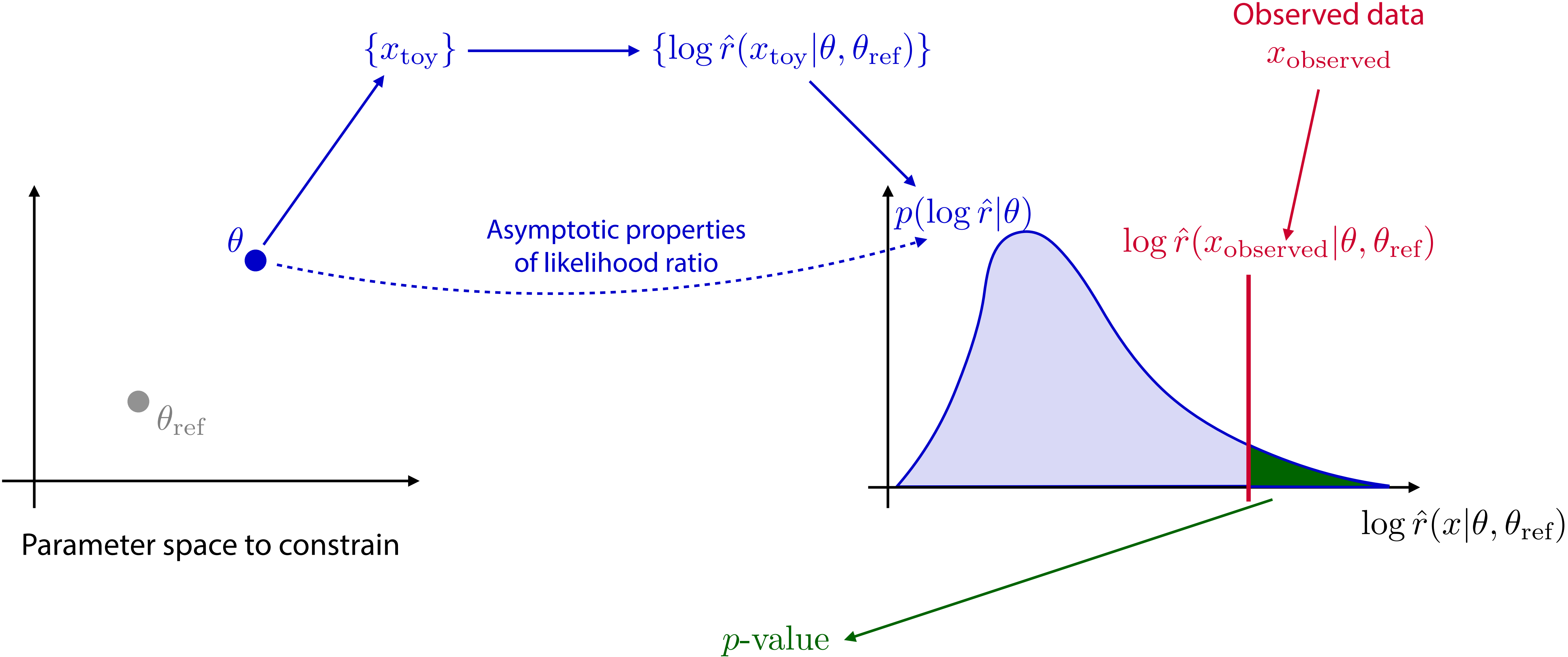
Limit setting (frequentist, standard ATLAS / CMS practice)



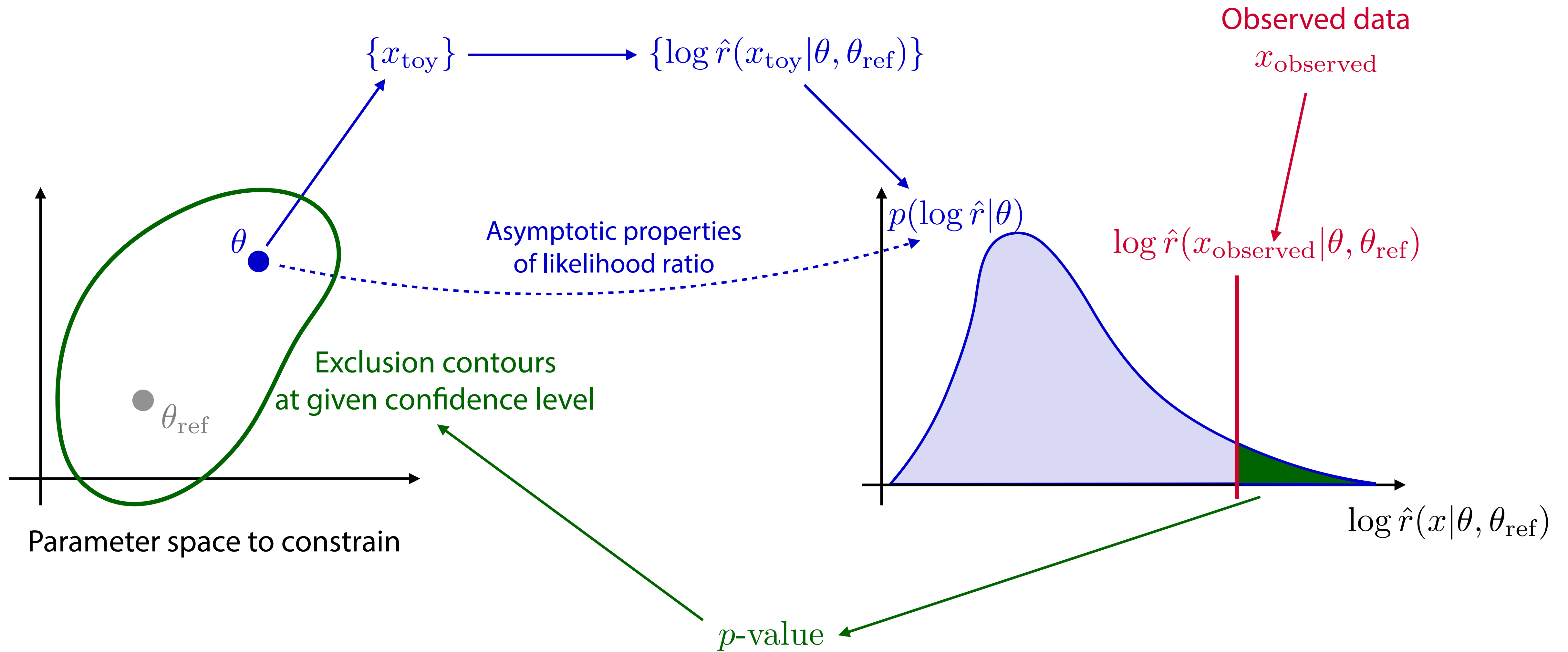
Limit setting (frequentist, standard ATLAS / CMS practice)



Limit setting (frequentist, standard ATLAS / CMS practice)



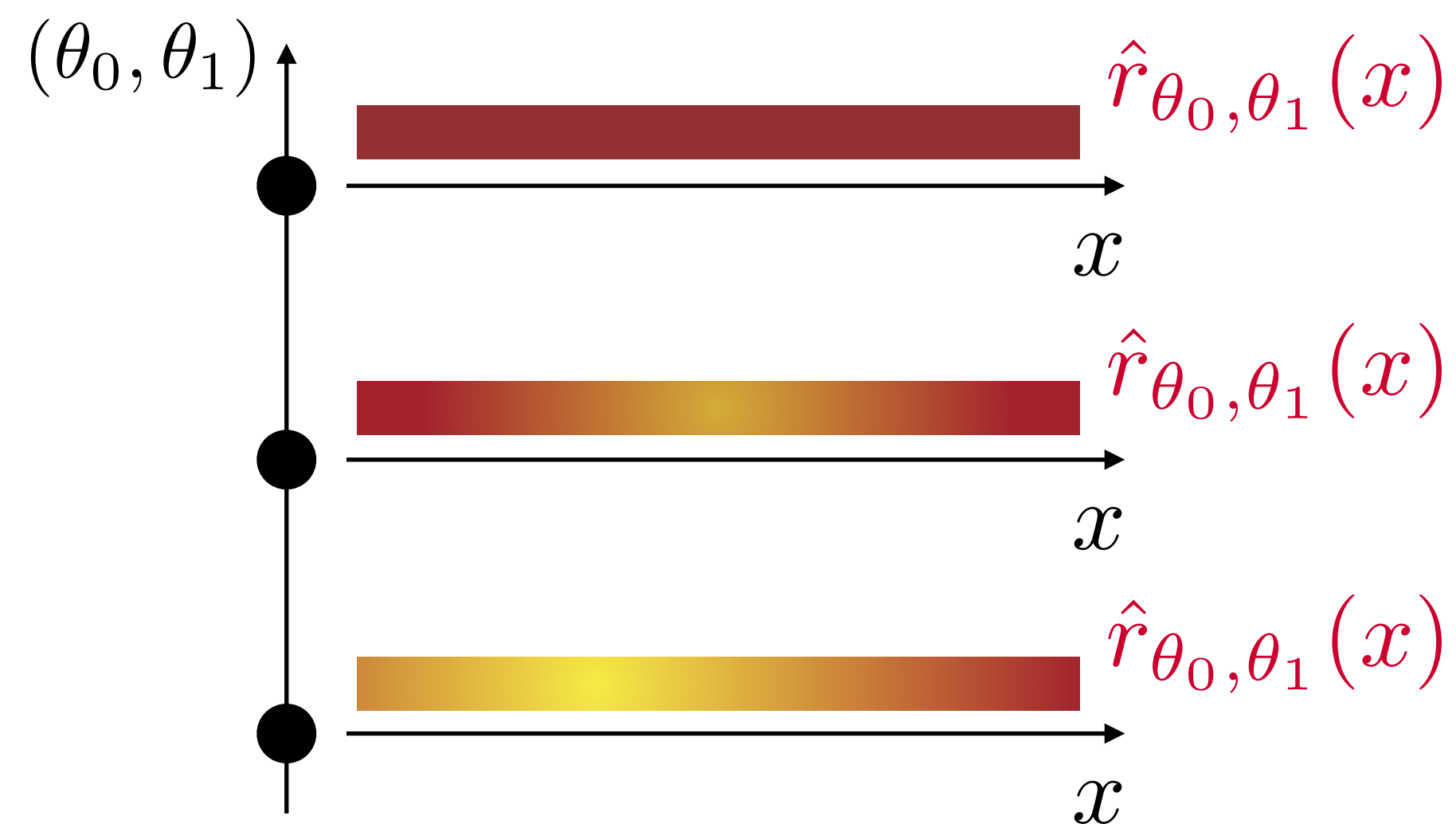
Limit setting (frequentist, standard ATLAS / CMS practice)



Two types of likelihood ratio estimators

A) Point by point:

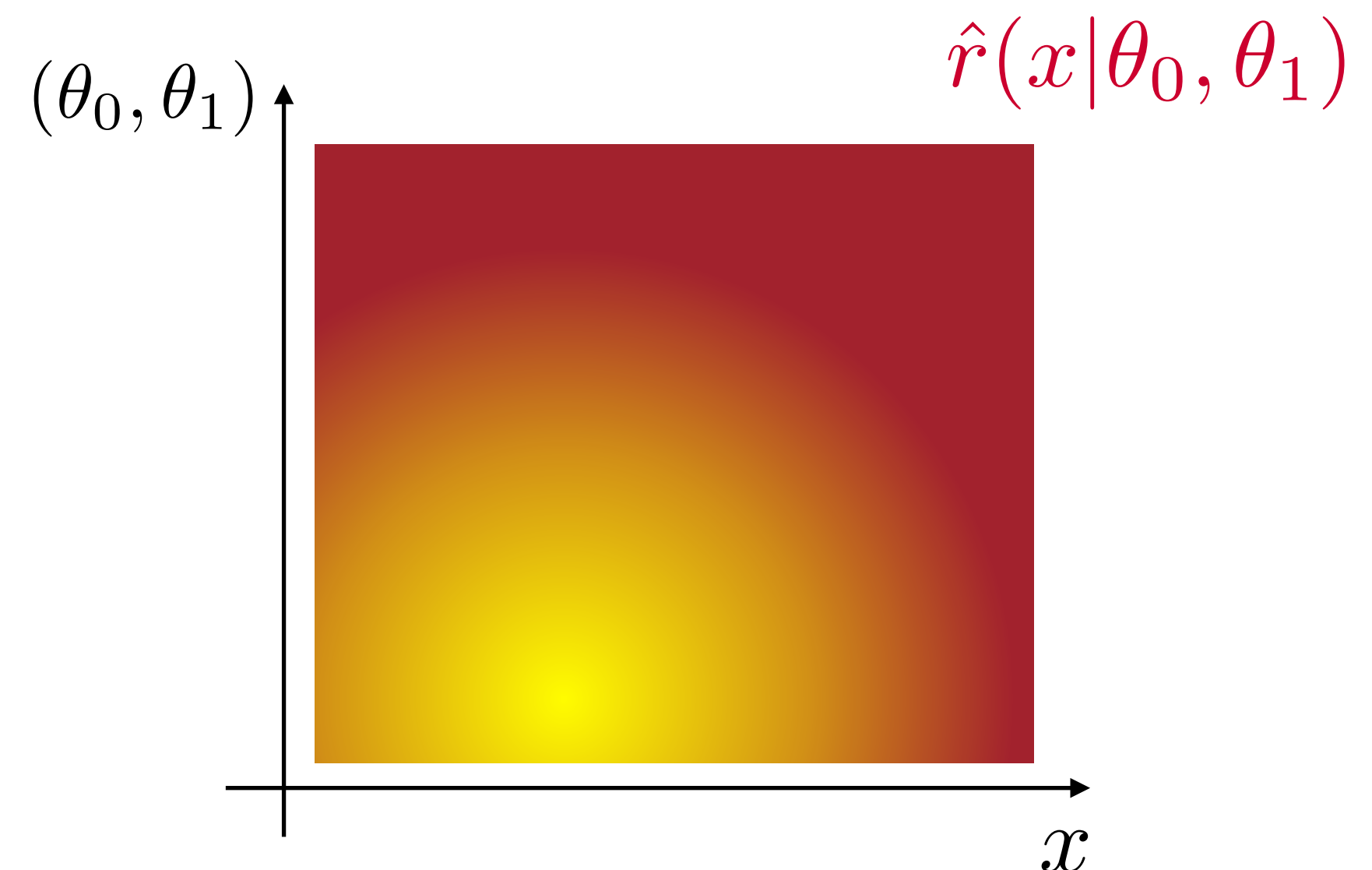
- first, define grid of parameter points $\{(\theta_0, \theta_1)\}$
- for each combination (θ_0, θ_1) , create separate estimator $\hat{r}_{\theta_0, \theta_1}(x)$
- final results can be interpolated between grid points



B) Parameterized:

[K. Cranmer, J. Pavez, G. Louppe 1506.02169;
P. Baldi et al. 1601.07913]

- create one estimator $\hat{r}(x|\theta_0, \theta_1)$ that is a function of θ_0 and θ_1
- no further interpolation necessary
- “borrows information” from close points



WBF Higgs to four leptons with detector effects

