Meticulous measurements with matrix elements and machine learning

Johann Brehmer
New York University

University of Oregon, ITS/CHEP joint seminar
January 14, 2019

Kyle Cranmer

Gilles Louppe

Juan Pavez

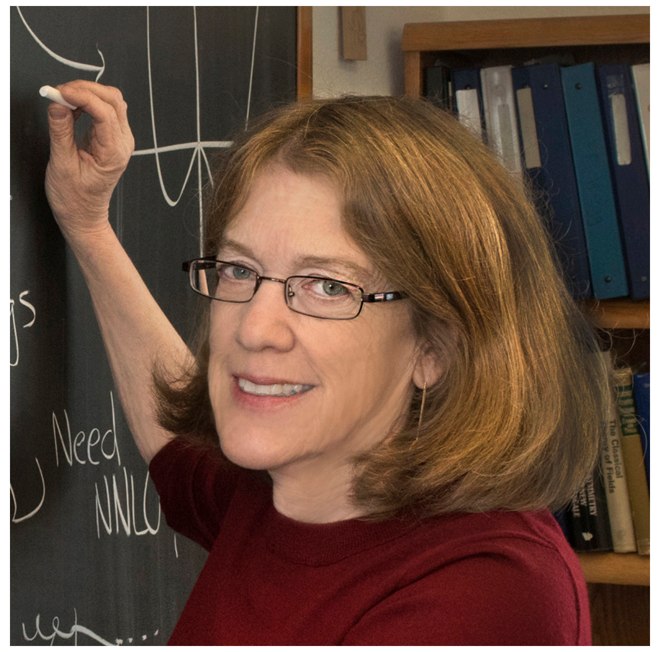
Felix Kling

Irina Espejo

Zubair Bhatti

Markus Stoye

Tilman Plehn



Sally Dawson

Sam Homiller

Josh Rudermann

Duccio Pappadopulo

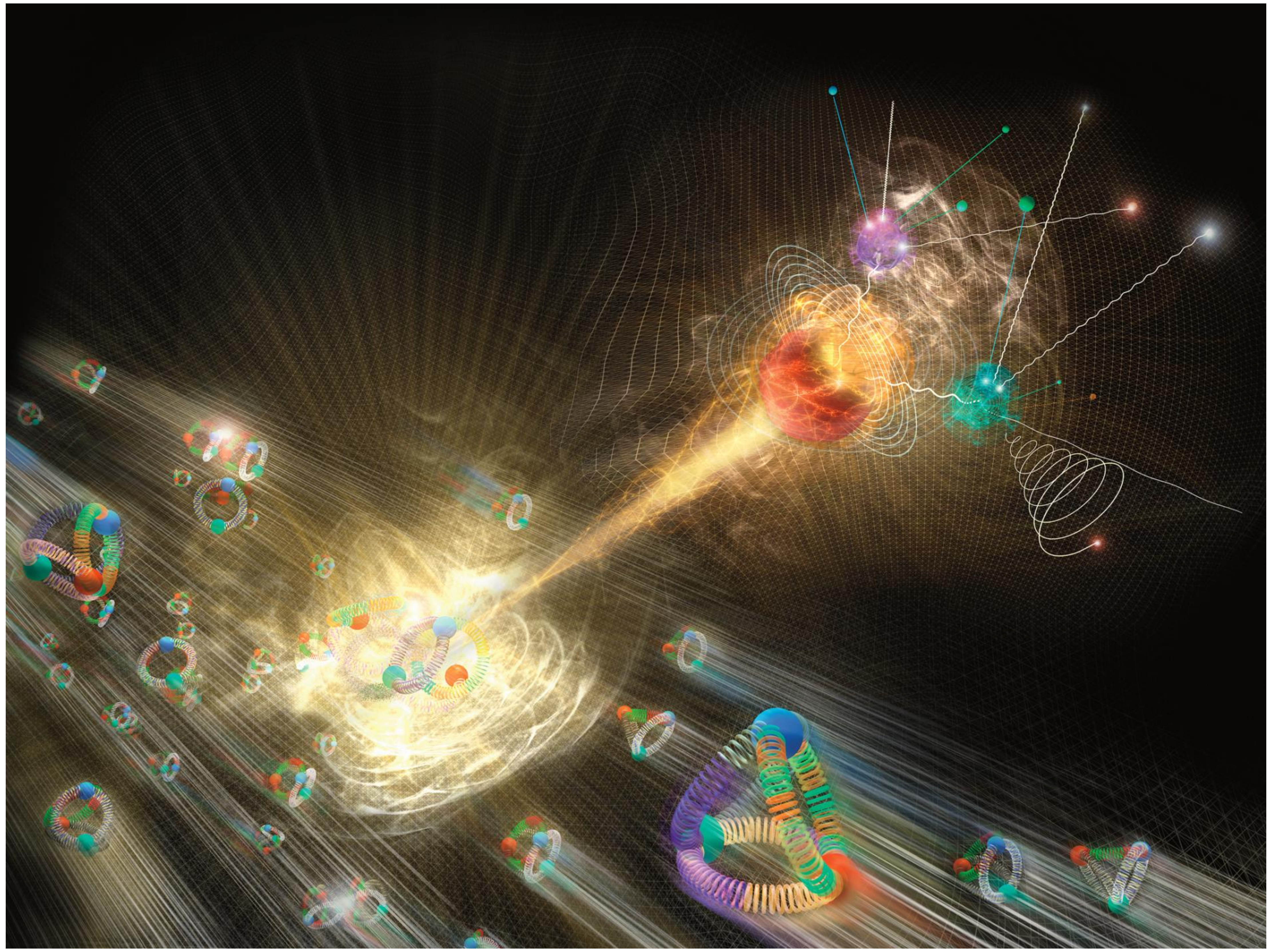
Marco Farina

The legacy of the LHC

- LHC: so far, not so good at discovering unexpected new particles…
  but great at making Higgs bosons!

The legacy of the LHC

- LHC: so far, not so good at discovering unexpected new particles…
  but great at making Higgs bosons!



[National Geographic]

The legacy of the LHC

- LHC: so far, not so good at discovering unexpected new particles…
  but great at making Higgs bosons!

- Measurements of Higgs properties & electroweak interactions ~~might~~ will help us understand…
  - the hierarchy problem (why is gravity so weak?)
  - fermion masses (why is the electron so light?)
  - matter dominance (why are we?)
  - vacuum stability (why is anything like it is?)

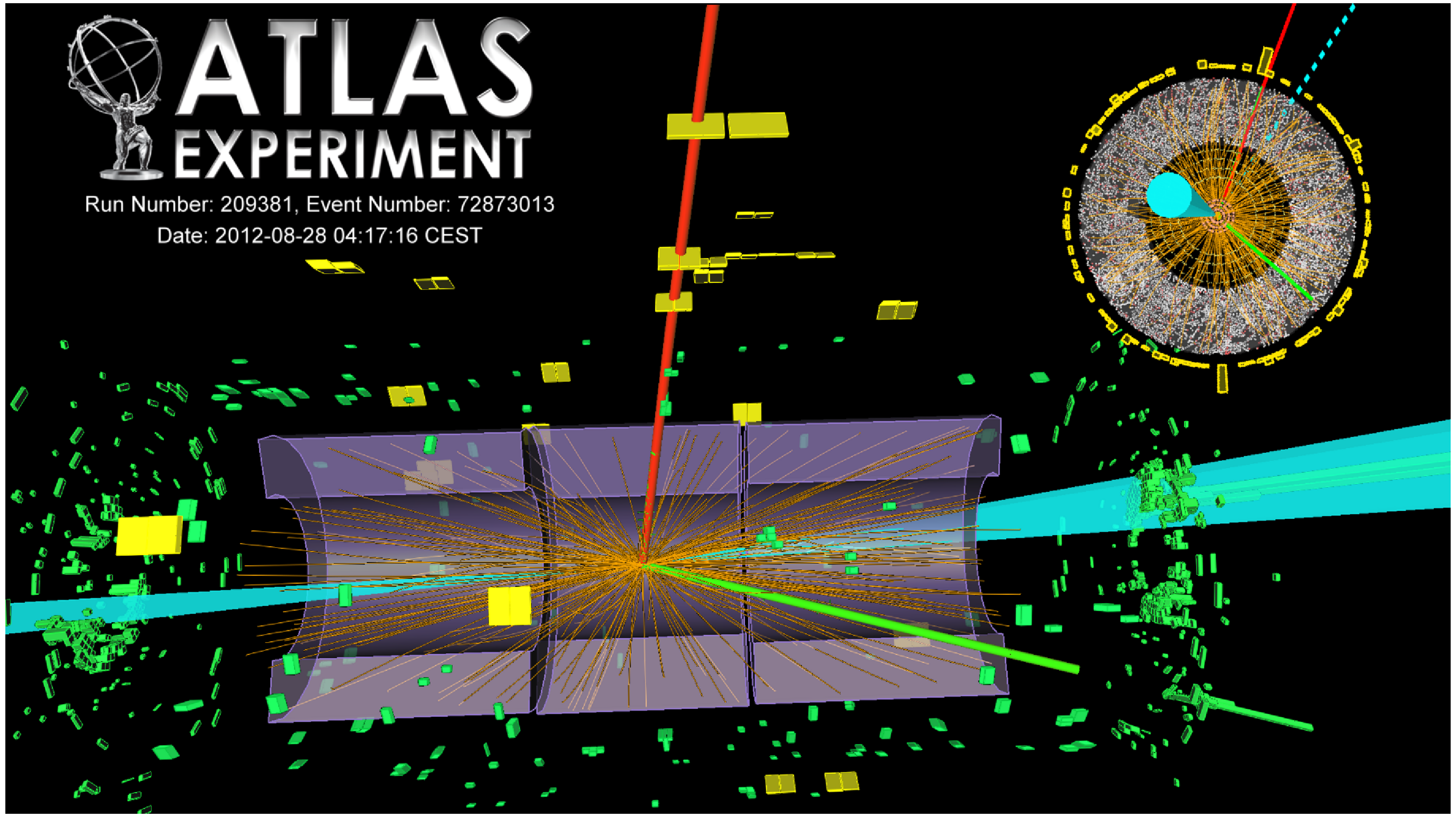
Old challenges and a new solution

- These measurements are difficult for established analysis strategies!
  - Many parameters

$$S = \int \mathrm{d}^4 x \left[\mathcal{L}_{\mathrm{SM}} + \frac{f_{\phi,2}}{\Lambda^2} \, \frac{1}{2} \, \partial_\mu (\phi^\dagger \phi) \, \partial^\mu (\phi^\dagger \phi) + \frac{f_{\phi,3}}{\Lambda^2} \, \frac{1}{3} (\phi^\dagger \phi)^3 \right.$$
$$+ \frac{f_{GG}}{\Lambda^2} \, (\phi^\dagger \phi) \, G^a_{\mu\nu} \, G^{\mu\nu\, a} - \frac{f_{BB}}{\Lambda^2} \, \frac{g'^2}{4} \, (\phi^\dagger \phi) \, B_{\mu\nu} \, B^{\mu\nu} - \frac{f_{WW}}{\Lambda^2} \, \frac{g^2}{4} \, (\phi^\dagger \phi) \, W^a_{\mu\nu} \, W^{\mu\nu\, a}$$
$$+ \frac{f_B}{\Lambda^2} \, \frac{\mathrm{i} g'}{2} \, (D^\mu \phi)^\dagger \, D^\nu \phi \, B_{\mu\nu} + \frac{f_W}{\Lambda^2} \, \frac{\mathrm{i} g}{2} \, (D^\mu \phi)^\dagger \, \sigma^a \, D^\nu \phi \, W^a_{\mu\nu}$$
$$+ \frac{f_\ell}{\Lambda^2} \, (\phi^\dagger \phi) \, \overline{L}_L \phi \, \ell_R + \frac{f_u}{\Lambda^2} \, (\phi^\dagger \phi) \, \overline{Q}_L \tilde{\phi} \, u_R + \frac{f_d}{\Lambda^2} \, (\phi^\dagger \phi) \, \overline{Q}_L \phi \, d_R$$
$$\left. + \frac{f_{G\widetilde{G}}}{\Lambda^2} \, (\phi^\dagger \phi) \, G^a_{\mu\nu} \, \widetilde{G}^{\mu\nu\, a} - \frac{f_{B\widetilde{B}}}{\Lambda^2} \, \frac{g'^2}{4} (\phi^\dagger \phi) \, B_{\mu\nu} \, \widetilde{B}^{\mu\nu} - \frac{f_{W\widetilde{W}}}{\Lambda^2} \, \frac{g^2}{4} (\phi^\dagger \phi) \, W^a_{\mu\nu} \, \widetilde{W}^{\mu\nu\, a} \right]$$

Old challenges and a new solution

- These measurements are difficult for established analysis strategies!
  - Many parameters
  - Many observables



[ATLAS 1501.04943]

Old challenges and a new solution

- These measurements are difficult for established analysis strategies!
  - Many parameters
  - Many observables
  - Subtle kinematic effects

Old challenges and a new solution

- These measurements are difficult for established analysis strategies!
  - Many parameters
  - Many observables
  - Subtle kinematic effects

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

Old challenges and a new solution

- These measurements are difficult for established analysis strategies!
  - Many parameters
  - Many observables
  - Subtle kinematic effects
  - We don't have a good way of calculating the likelihood function when observations are high-dimensional and we have to deal with the detector response

Old challenges and a new solution

- These measurements are difficult for established analysis strategies!
  - Many parameters
  - Many observables
  - Subtle kinematic effects
  - We don't have a good way of calculating the likelihood function when observations are high-dimensional and we have to deal with the detector response

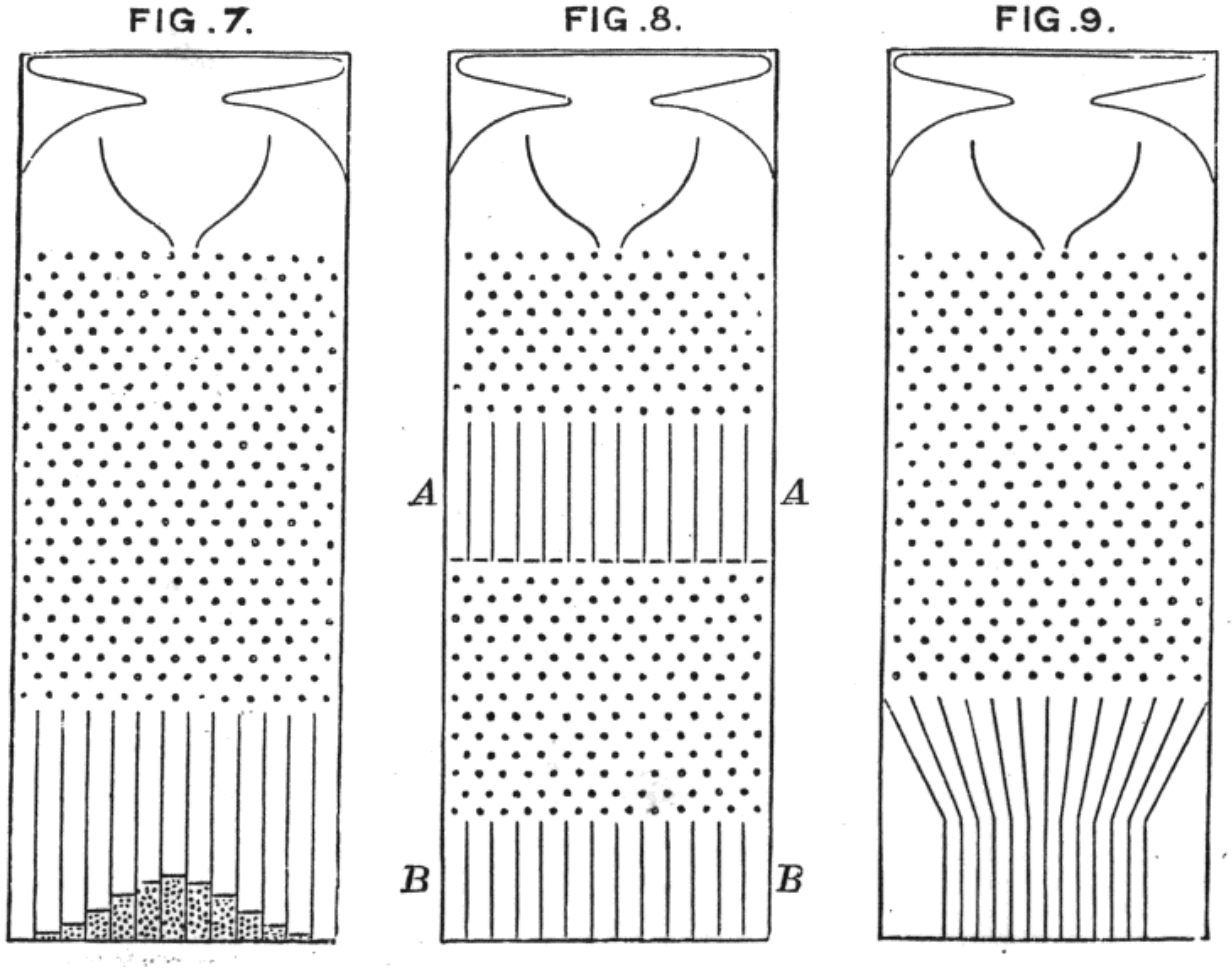
- This talk:
  1. Why can't we calculate the likelihood?
  2. Why has that not stopped us before?
  3. A new approach that can improve measurements
  4. An example
  5. Automization

Likelihood-free inference

The Galton board

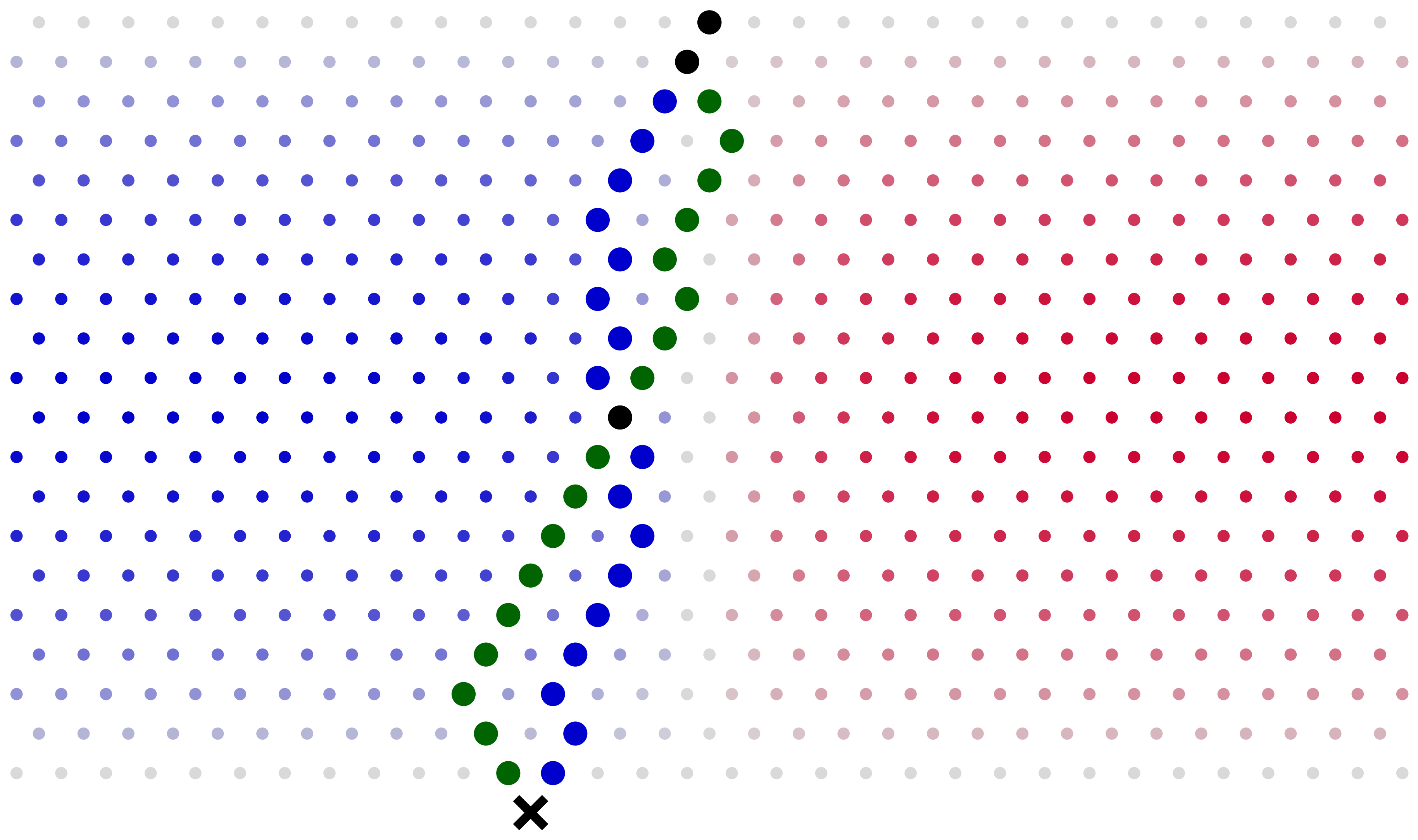
[galtonboard.com]

The Galton board



[F. Galton 1889]

Probabilities from integrating trajectories



Probability of ending in bin x : $p(x) = \int \mathrm{d}z \; p(x, z)$

Sum over
all trajectories
("latent variables")

Probability of
each path z
from start to x

The generalized Galton board

What if probability to go left at a nail is not always 0.5, but some (known) function of some parameters θ ?

- Prediction: given θ, generate samples of observations $\{x_i\}$.

  Simple: just drop balls!

The generalized Galton board

What if probability to go left at a nail is not always 0.5, but some (known) function of some parameters θ ?

- Prediction: given θ, generate samples of observations $\{x_i\}$.

  Simple: just drop balls!

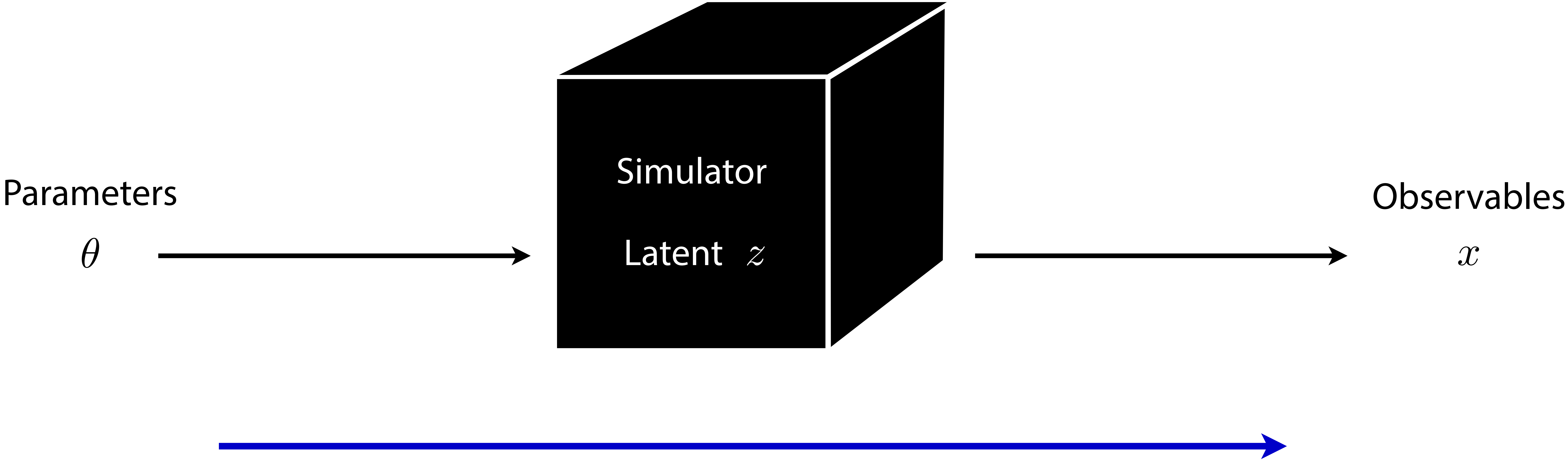
- Inference: given observations $\{x_i\}$, what are the most likely values for θ ?

  "Easy" problem if we can evaluate likelihood

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

  But the number of possible paths z can be huge, and it becomes impossible to calculate the integral!

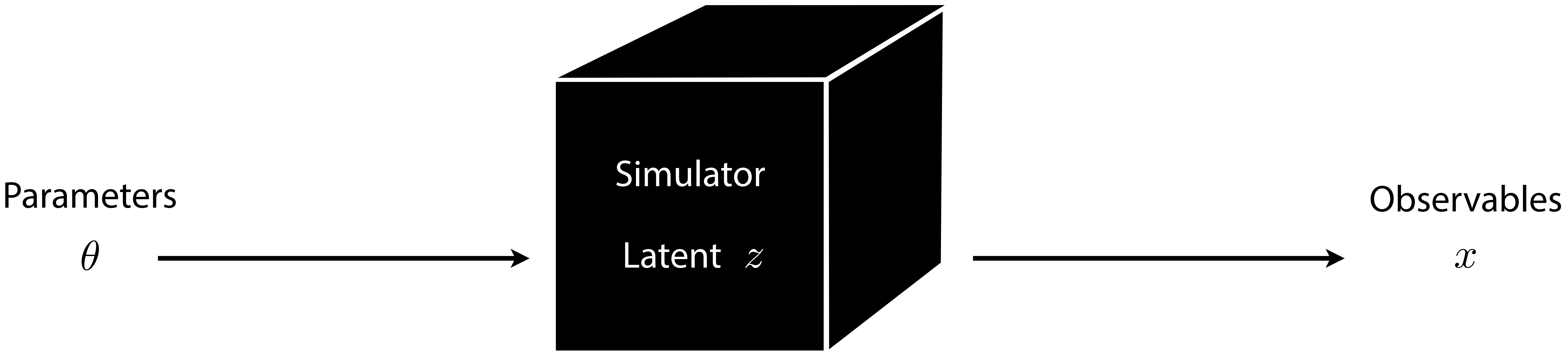
“Likelihood-free inference”



Prediction:

- Well-understood mechanistic model
- Simulator can generate samples

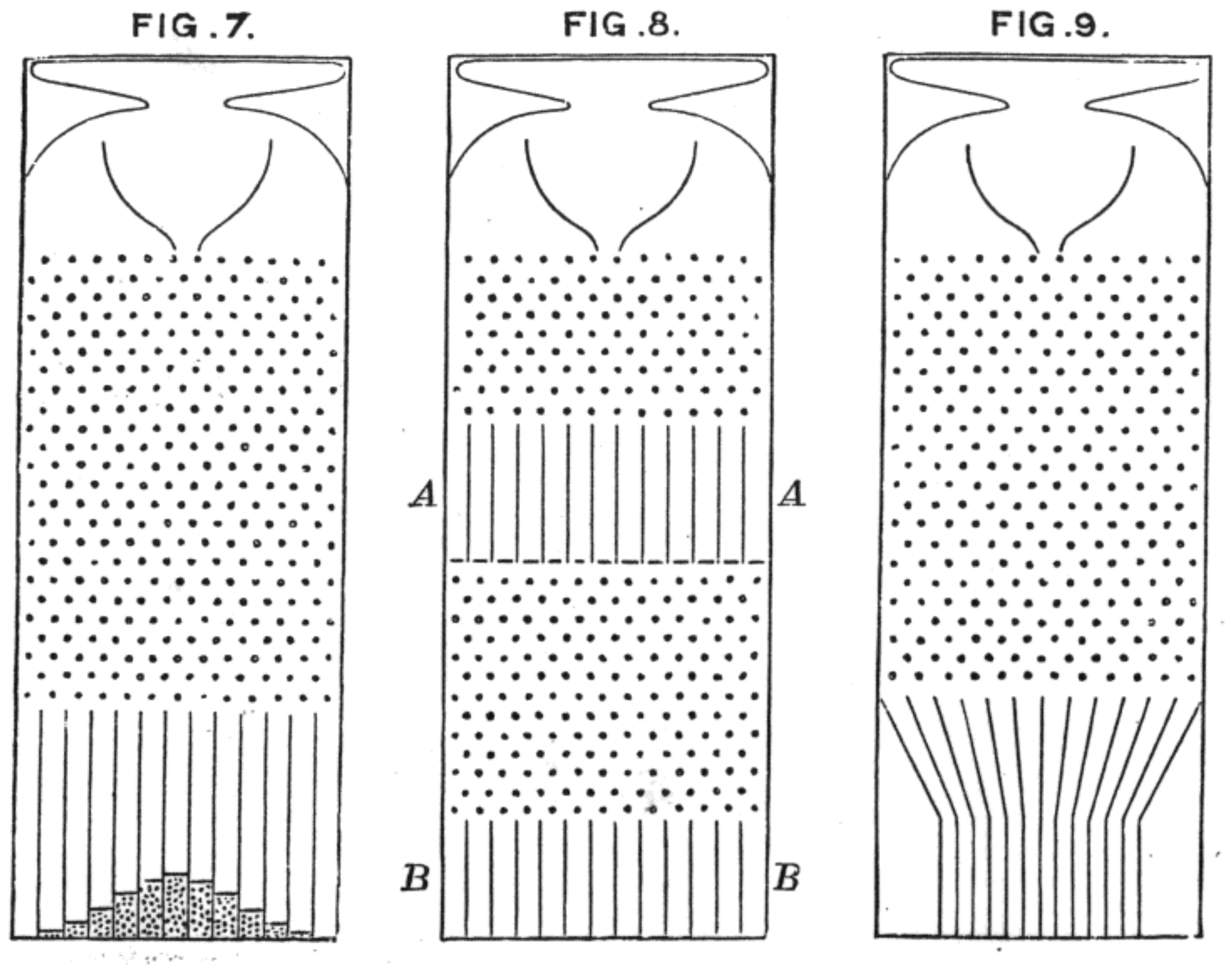
"Likelihood-free inference"



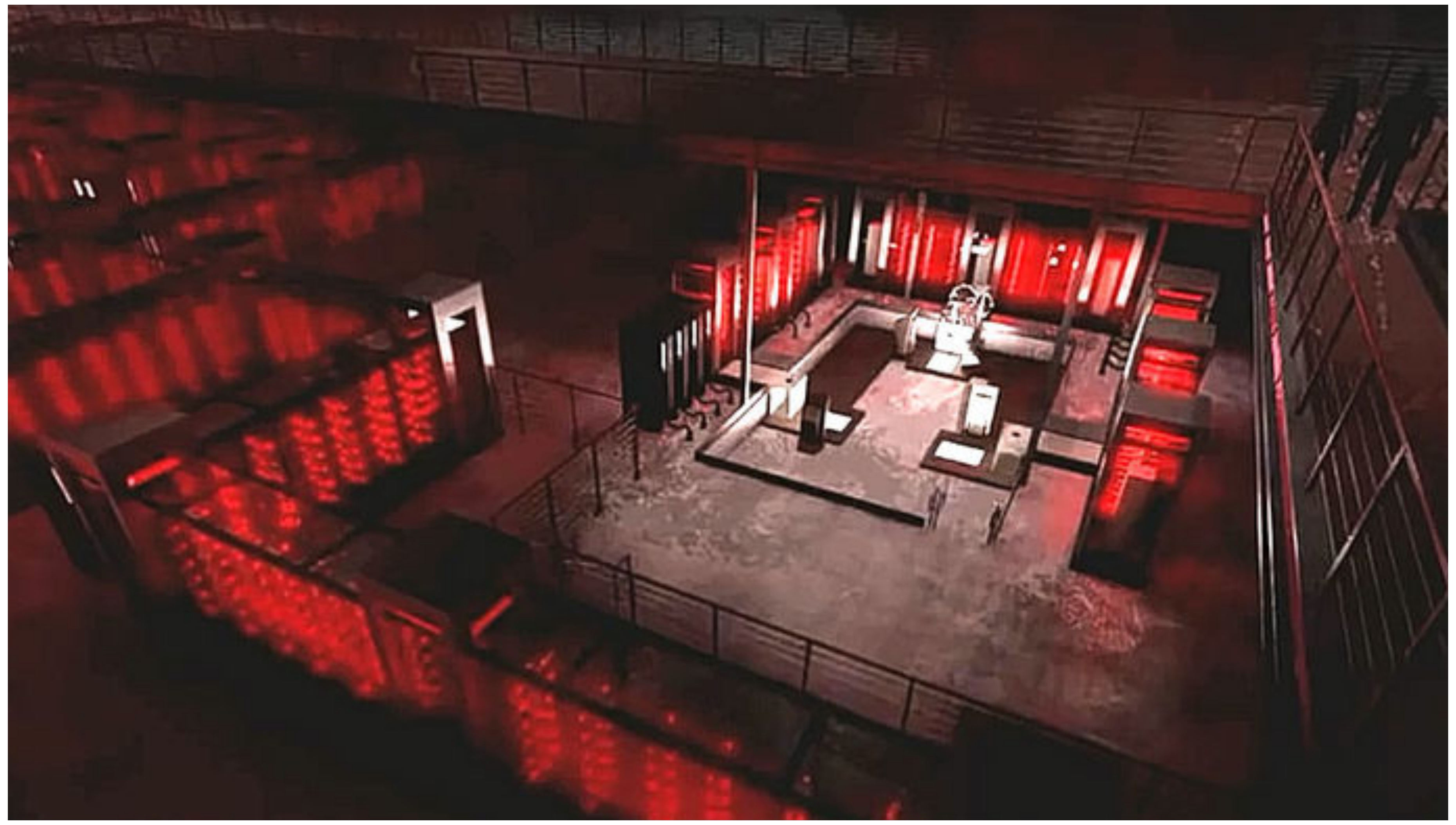
Prediction:
- Well-understood mechanistic model
- Simulator can generate samples

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

Galton board: metaphor for simulator-based science



[F. Galton 1889]

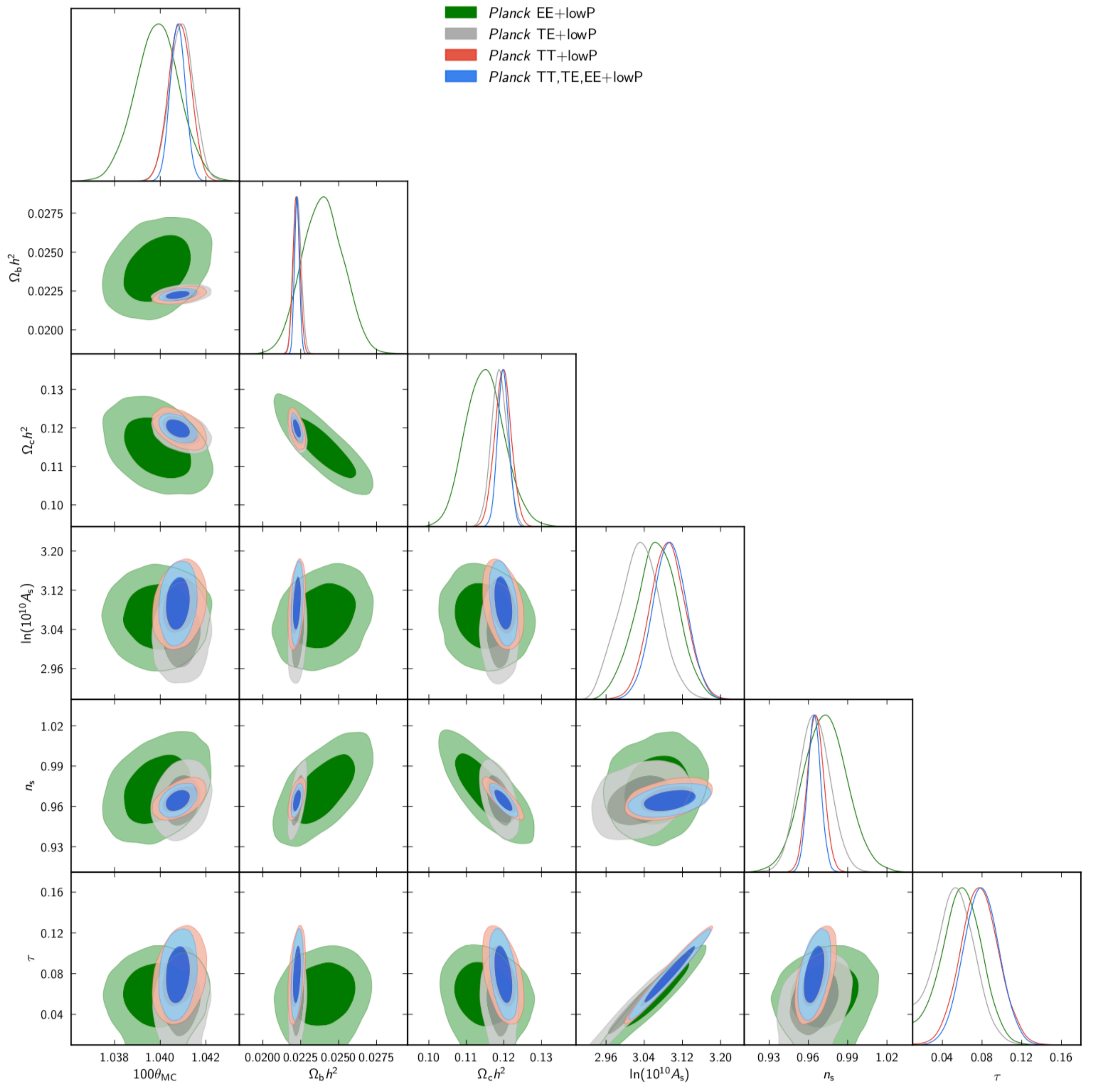


[HBO 2018]

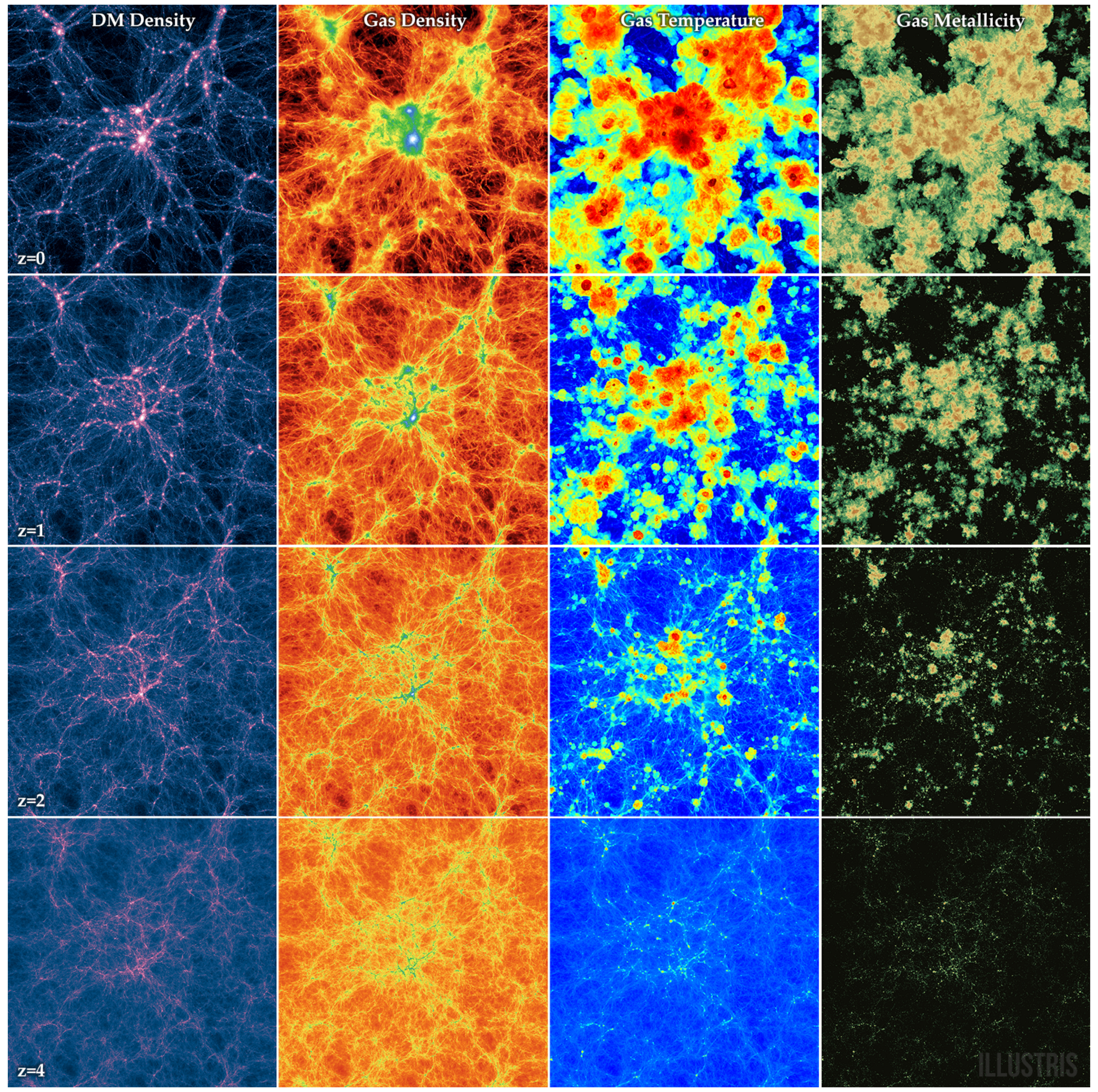
Galton board device	$\longrightarrow$	Computer simulation
Parameters θ	$\longrightarrow$	Model parameters θ
Bins x	$\longrightarrow$	Observables x
Path z	$\longrightarrow$	Latent variables z (stochastic execution trace through simulator)

Cosmological N-body simulations

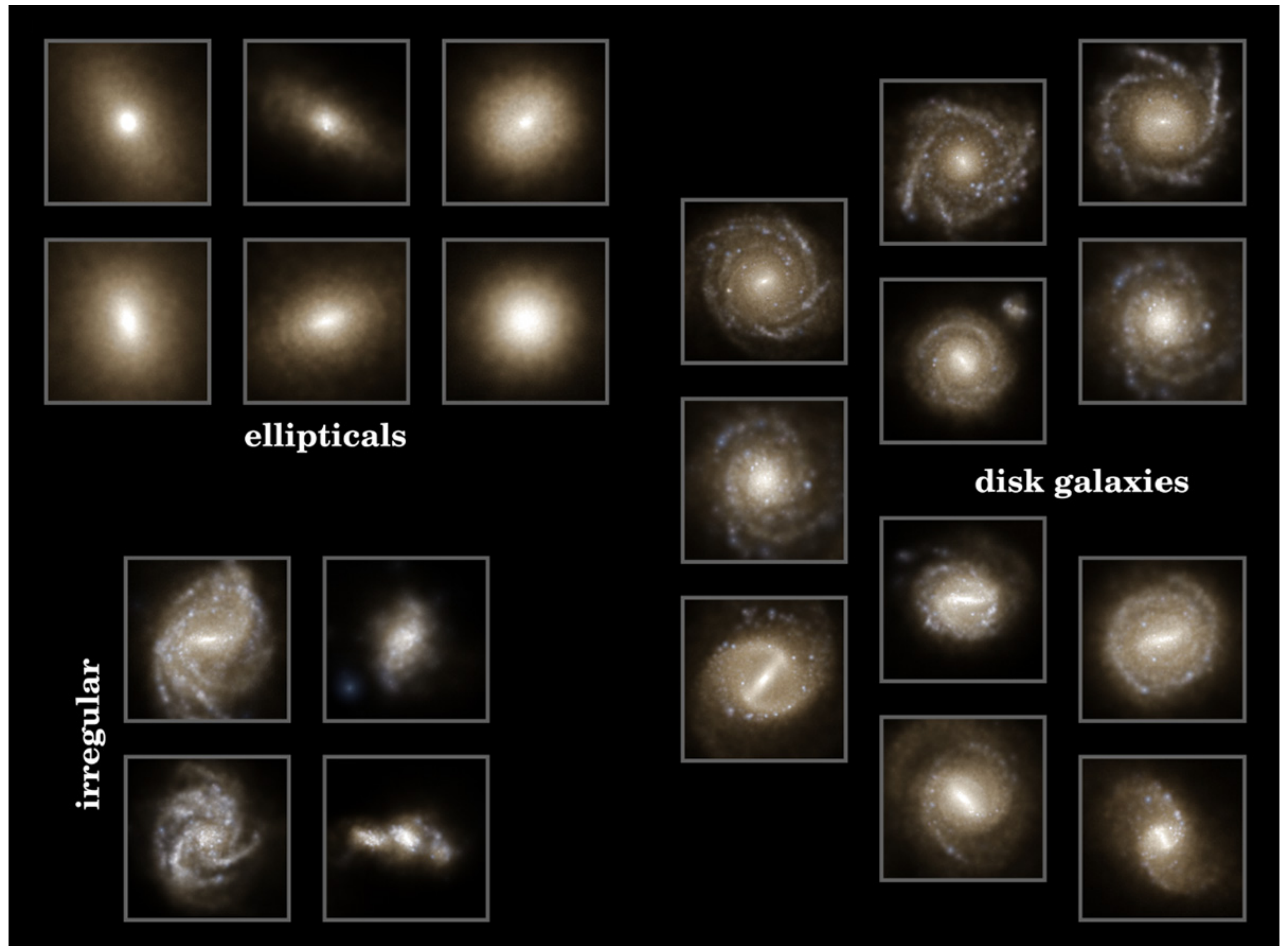
Parameters θ → **Simulator** (Latent z) → Observables x



[Planck 1502.01589]

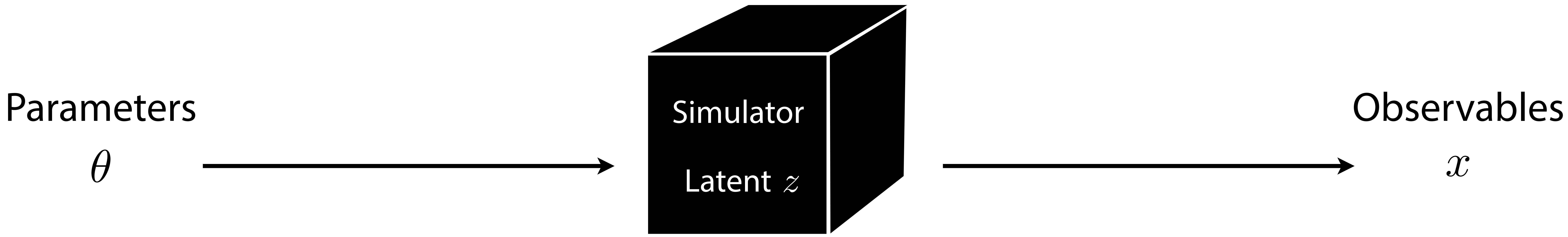


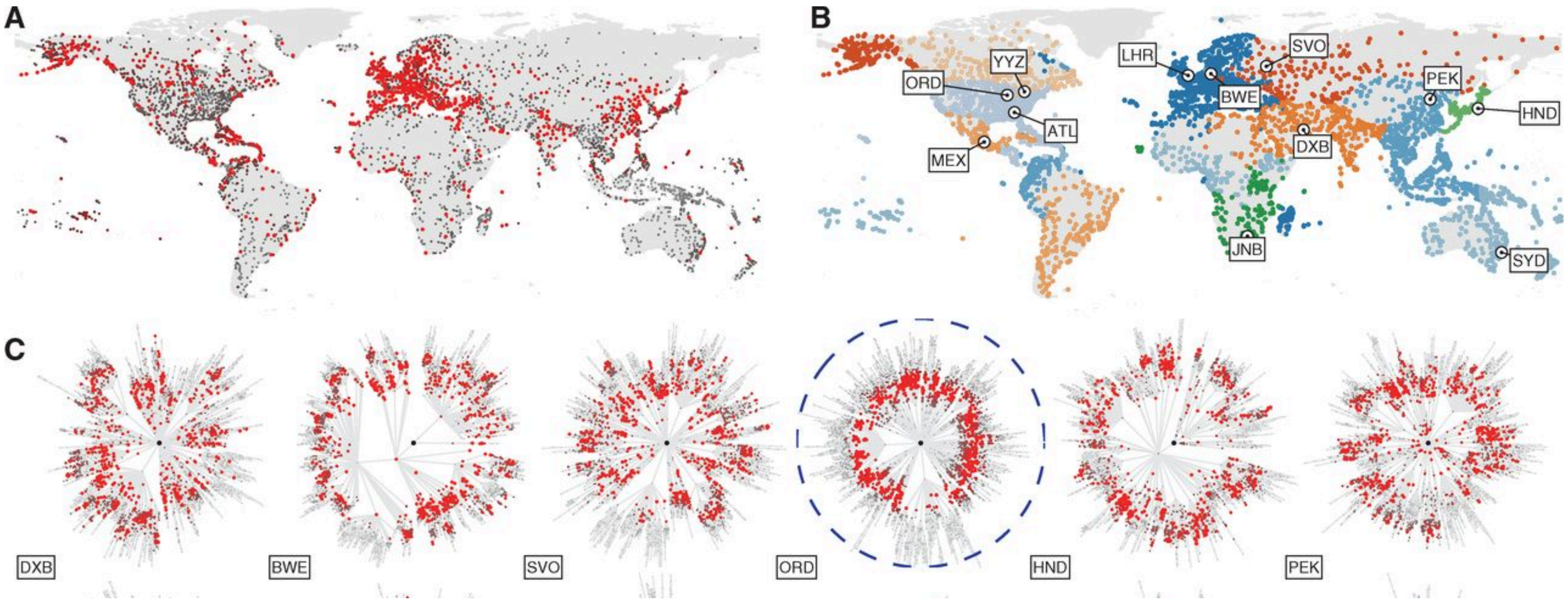
[Illustris 1405.2921]



[Illustris 1405.2921]

Epidemiology





[D. Brockmann, D. Helbing 2013]

Particle physics processes

Theory
parameters
θ

Evolution

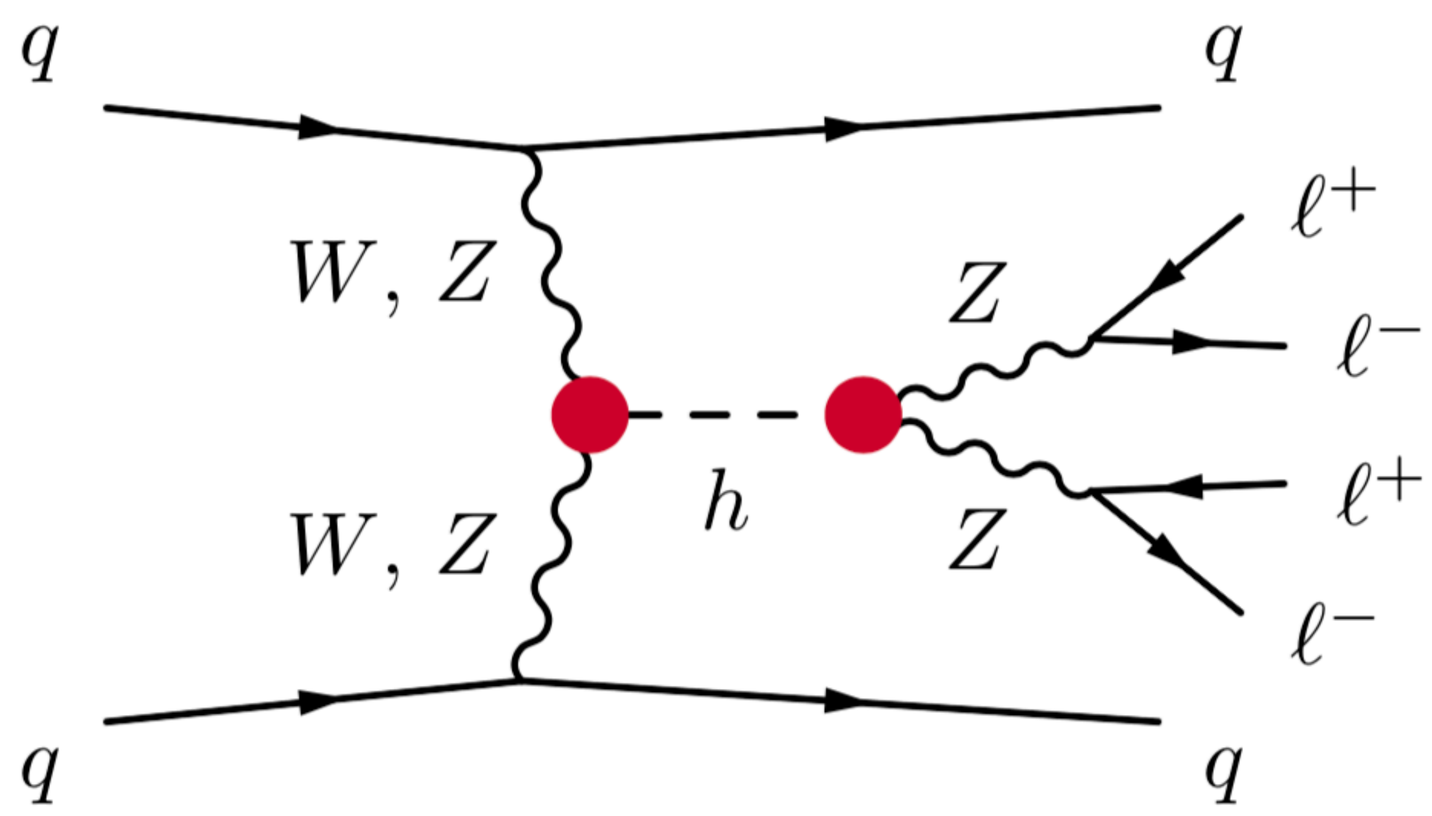
Particle physics processes

Latent variables

Parton-level momenta

Theory parameters

$$z_p \longleftarrow \theta$$



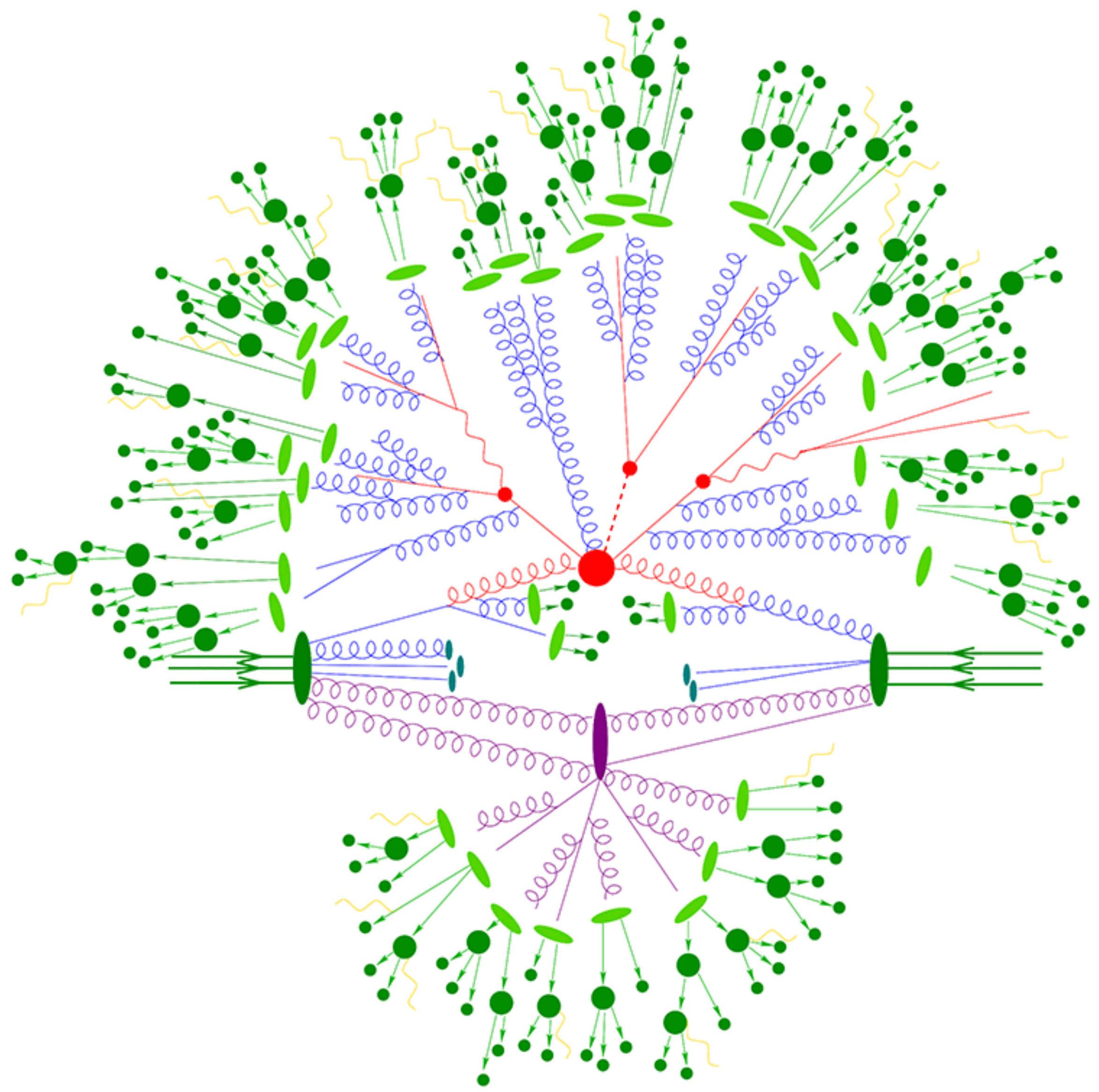
Evolution

Particle physics processes

Latent variables

Shower splittings	Parton-level momenta	Theory parameters
z_s	z_p	θ

$$z_s \longleftarrow z_p \longleftarrow \theta$$



[F. Krauss]

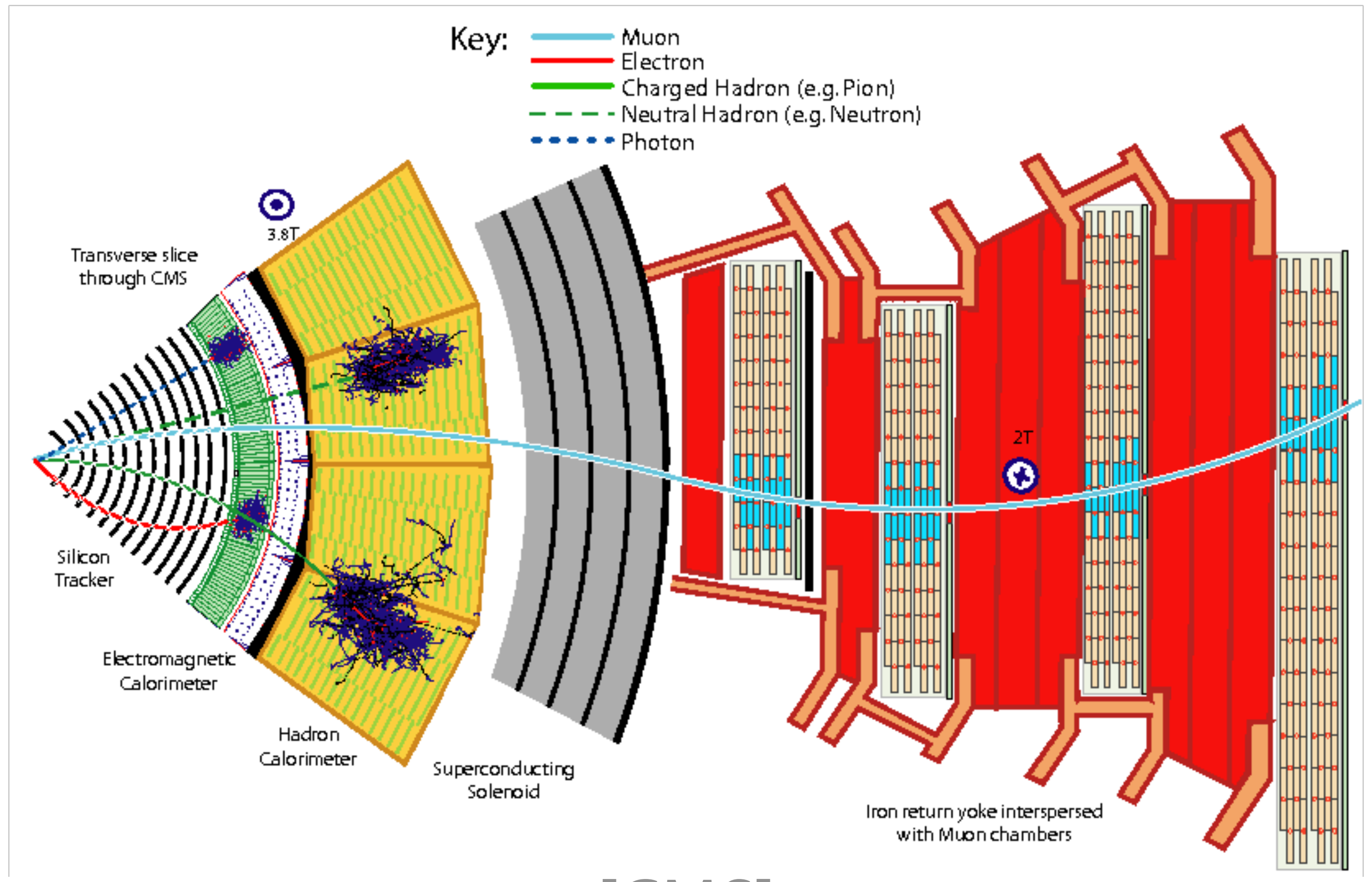
Evolution

Particle physics processes

Latent variables

Detector interactions	Shower splittings	Parton-level momenta	Theory parameters

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



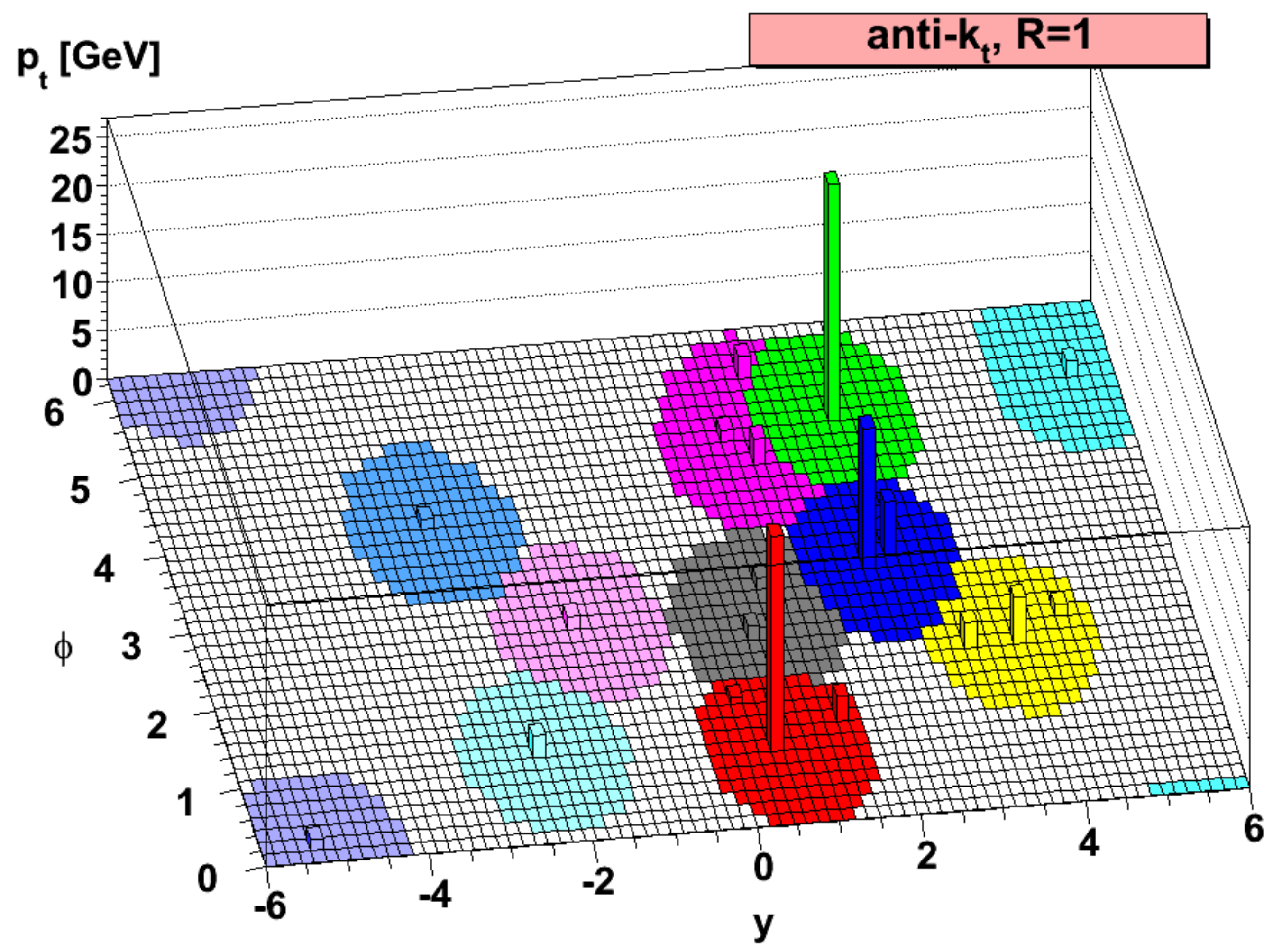
[CMS]

Evolution

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



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

$\longleftarrow$ Evolution

Particle physics processes

Latent variables

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

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

Sample from $p(x|z_d)$ $p(z_d|z_s)$ $p(z_s|z_p)$ $p(z_p|\theta)$

Prediction (simulation)

Particle physics processes

Latent variables

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters

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

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

$\longrightarrow$

Inference

Particle physics processes

Latent variables

Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters

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

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

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

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

$\longrightarrow$

Inference

Why has that not stopped us so far?

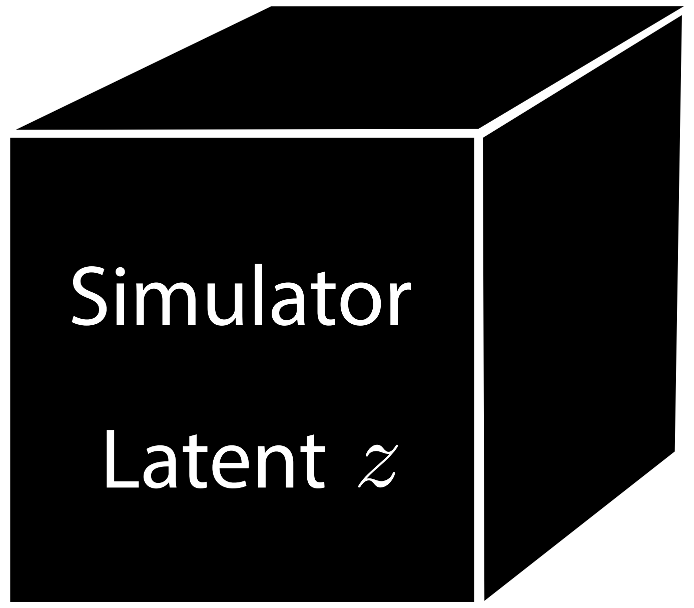
Solve it by histogramming summary statistics

- Typical physicists' solution:

Solve it by histogramming summary statistics

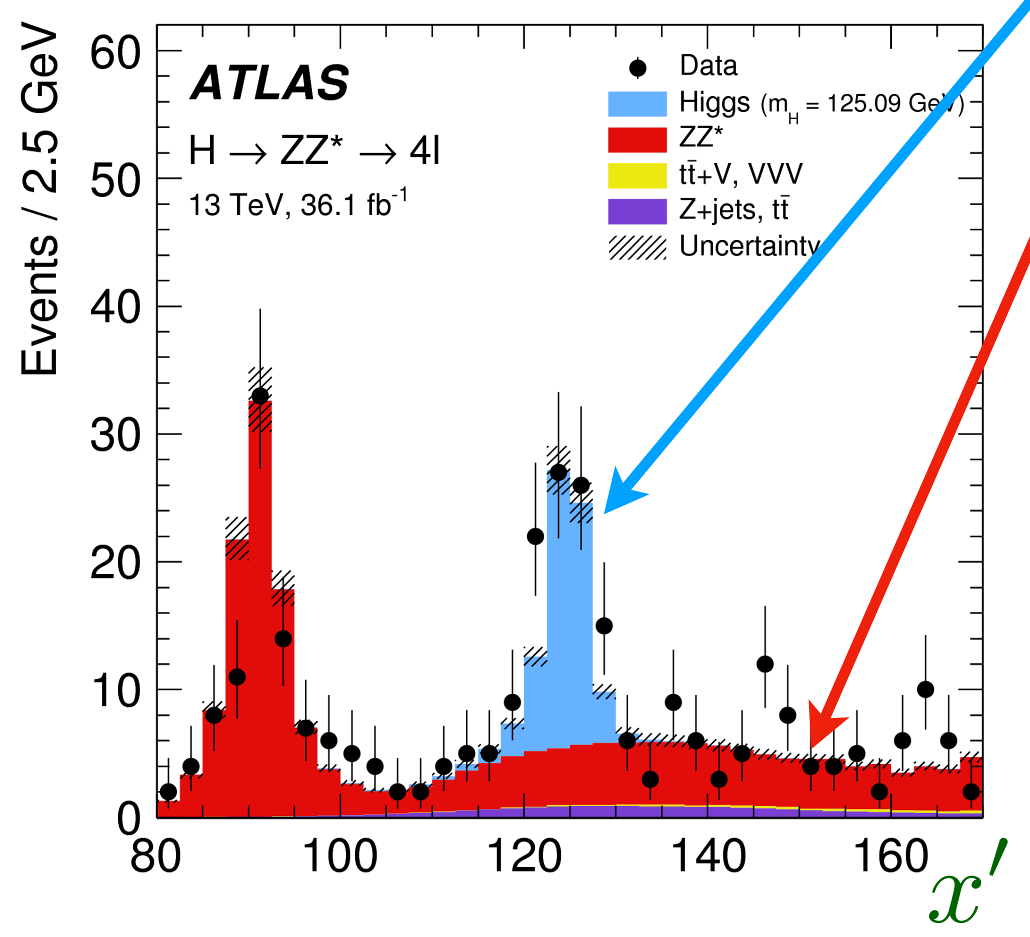
- Typical physicists' solution:

Parameters θ $\longrightarrow$



$\longrightarrow$ Observables x $\longrightarrow$ Summary statistics x'

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



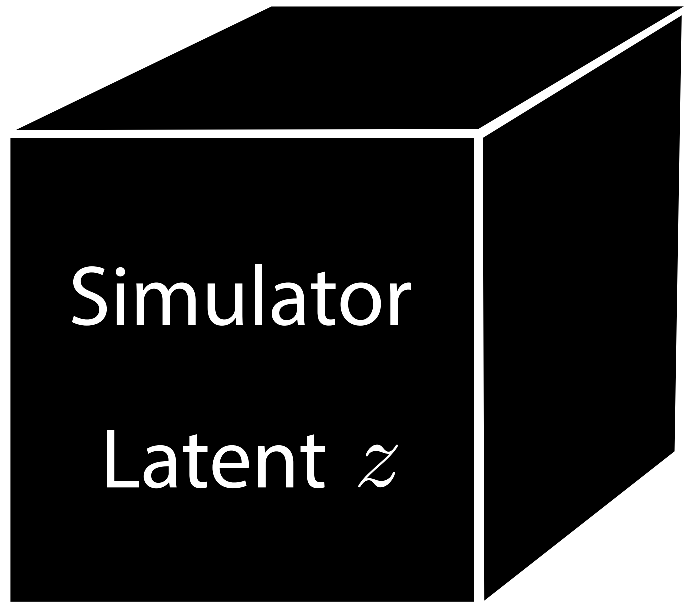
[ATLAS 1712.02304]

Run simulator for different θ, fill histograms

Solve it by histogramming summary statistics

- Typical physicists' solution:

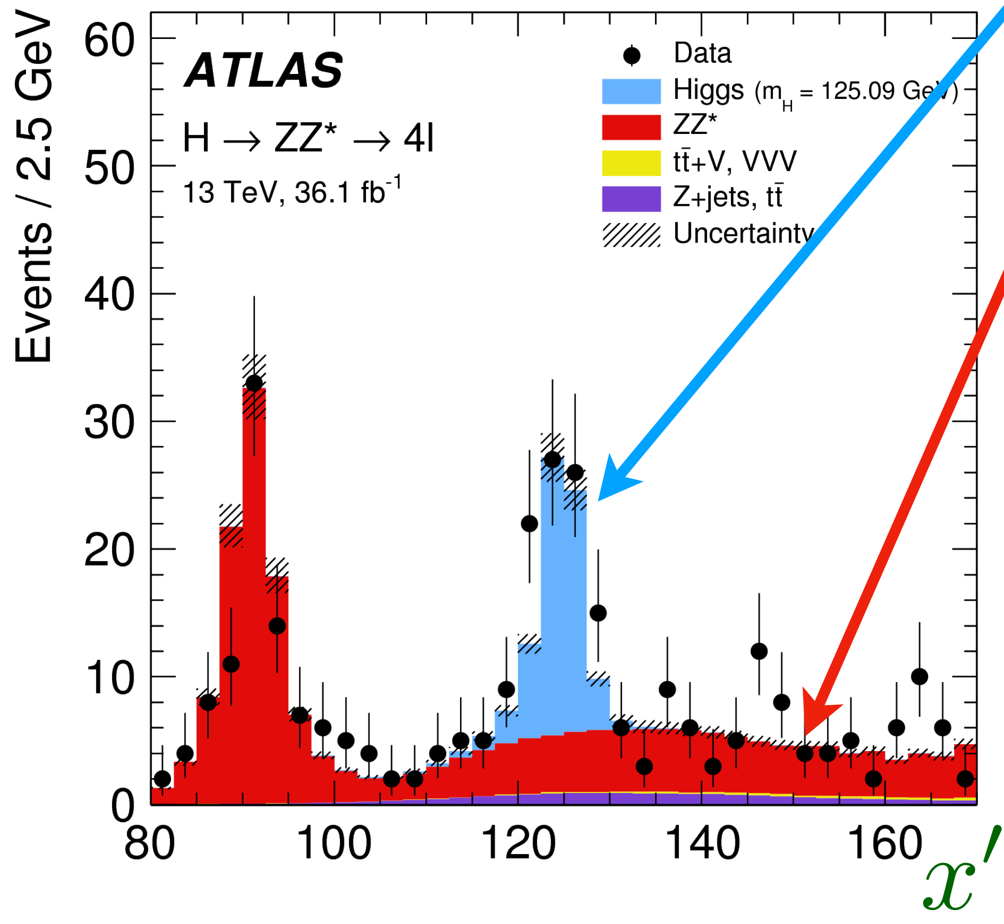
Parameters θ $\longrightarrow$



$\longrightarrow$ Observables x $\longrightarrow$ Summary statistics x'

Run simulator for different θ, fill histograms

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



[ATLAS 1712.02304]

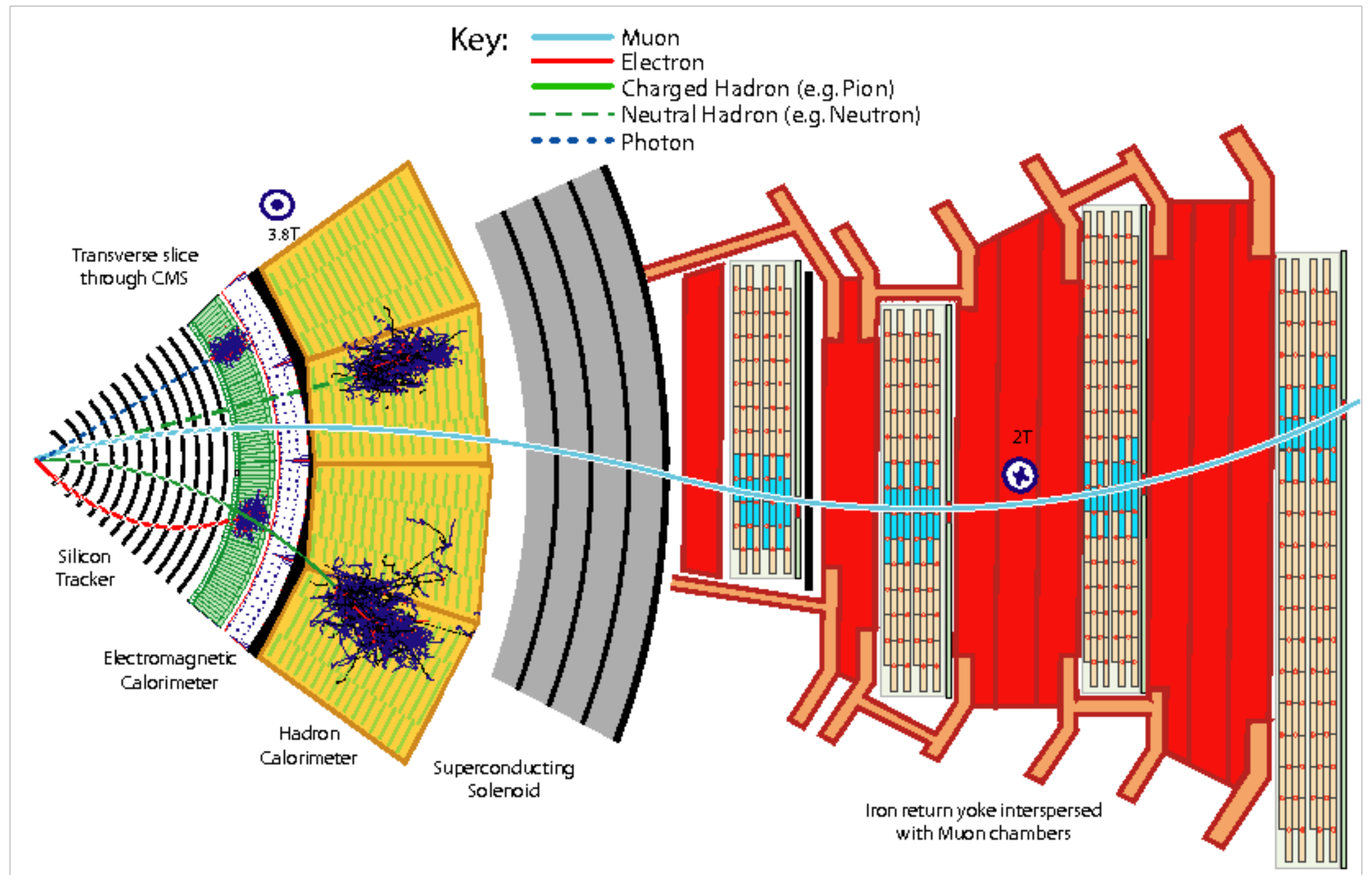
- How to choose x' ? Standard variables often lose information
  [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261; JB, F. Kling, T. Plehn, T. Tait 1712.02350]

- "Curse of dimensionality": Histograms don't scale to high-dimensional x

Solve it by approximating the integral

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

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



[CMS]

Solve it by approximating the integral

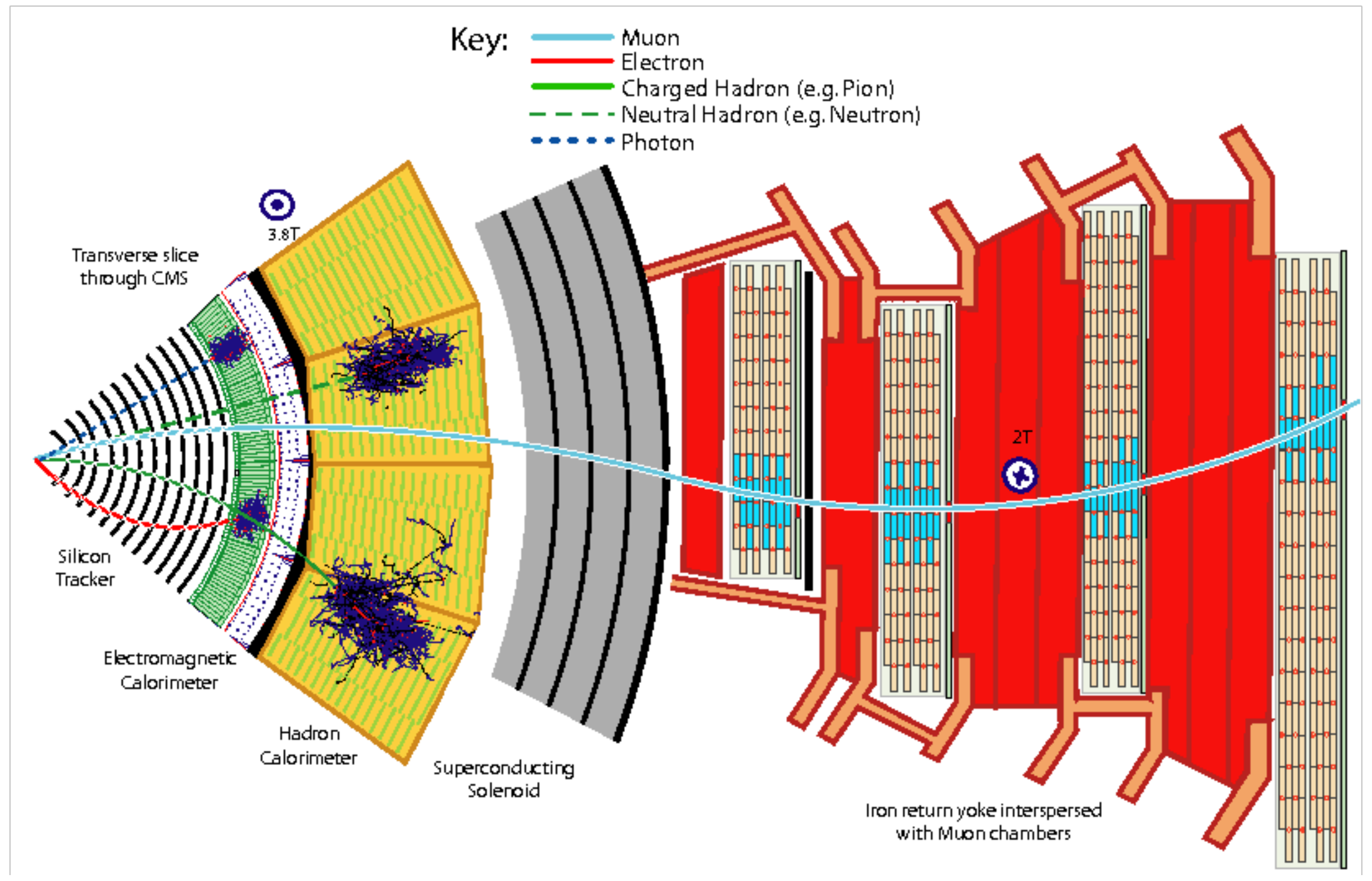
- Problem: high-dim. integral over shower / detector trajectories

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

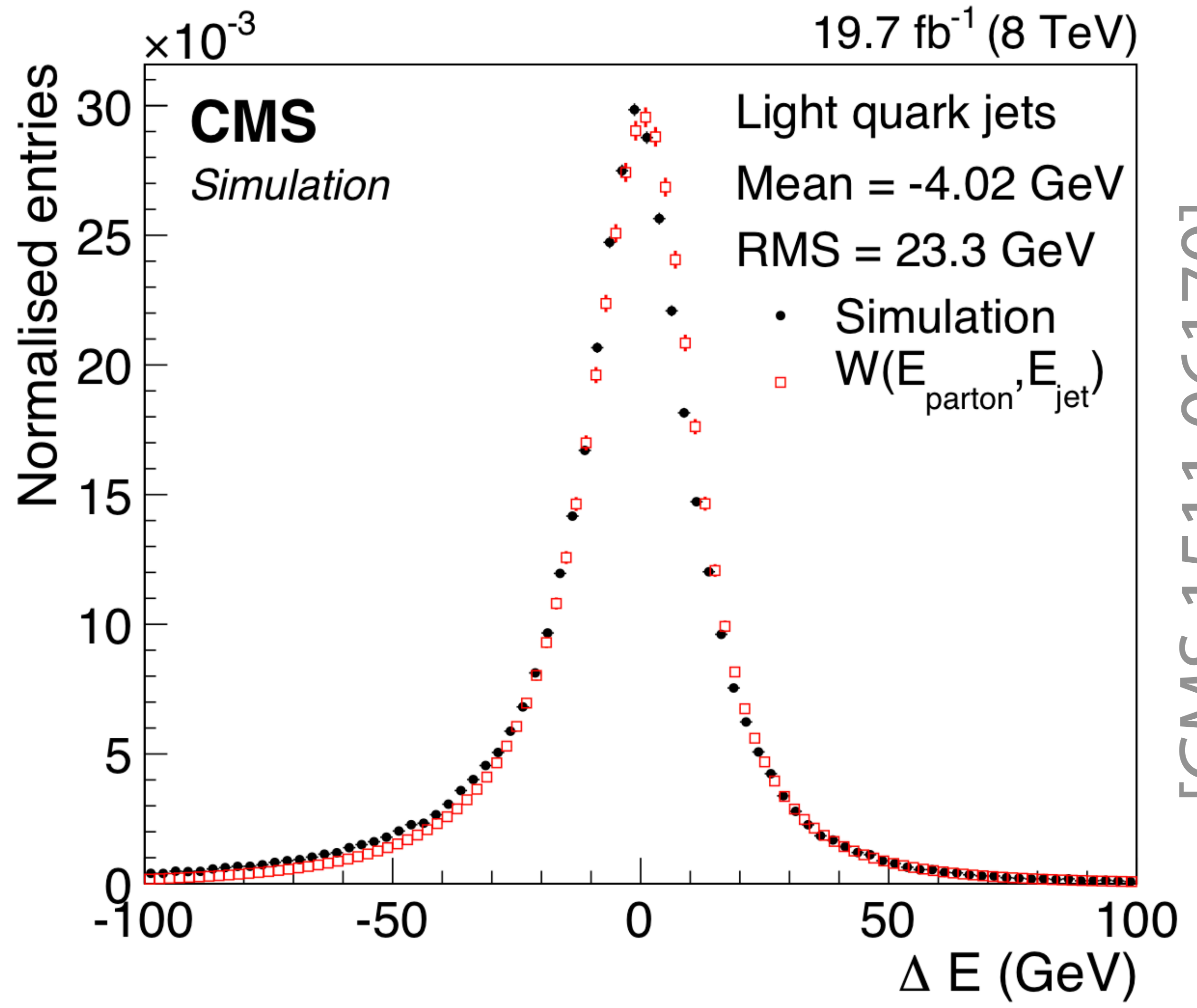
- Matrix Element Method: [K. Kondo 1988]
  - approximate shower + detector effects into transfer function $\hat{p}(x|z_p)$
  - explicitly calculate remaining integral

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

  - Shower / Event Deconstruction [D. E. Soper, M. Spannowsky 1102.3480] extend explicit calculation to the shower



[CMS]

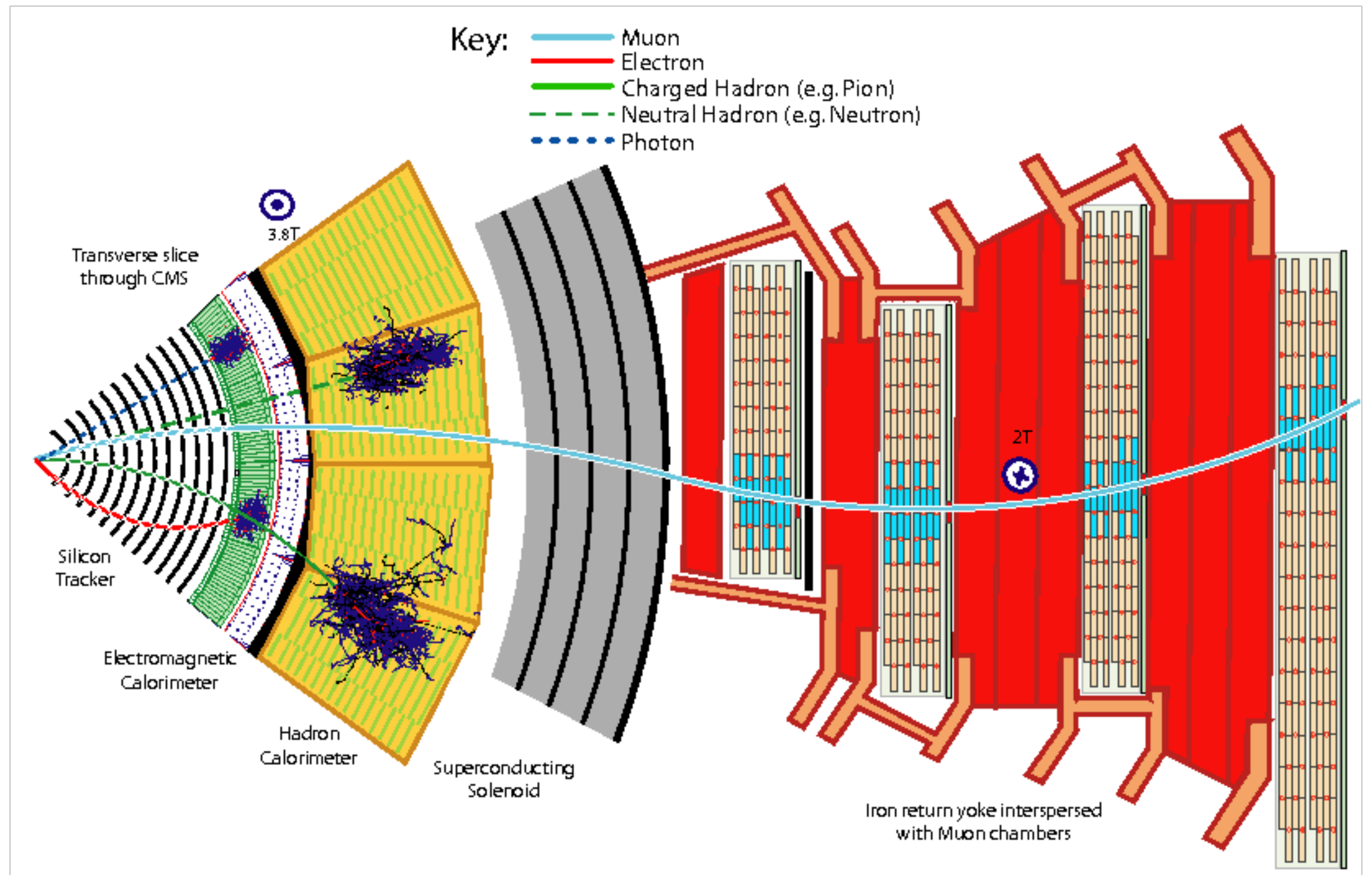


[CMS 1511.06170]

Solve it by approximating the integral

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

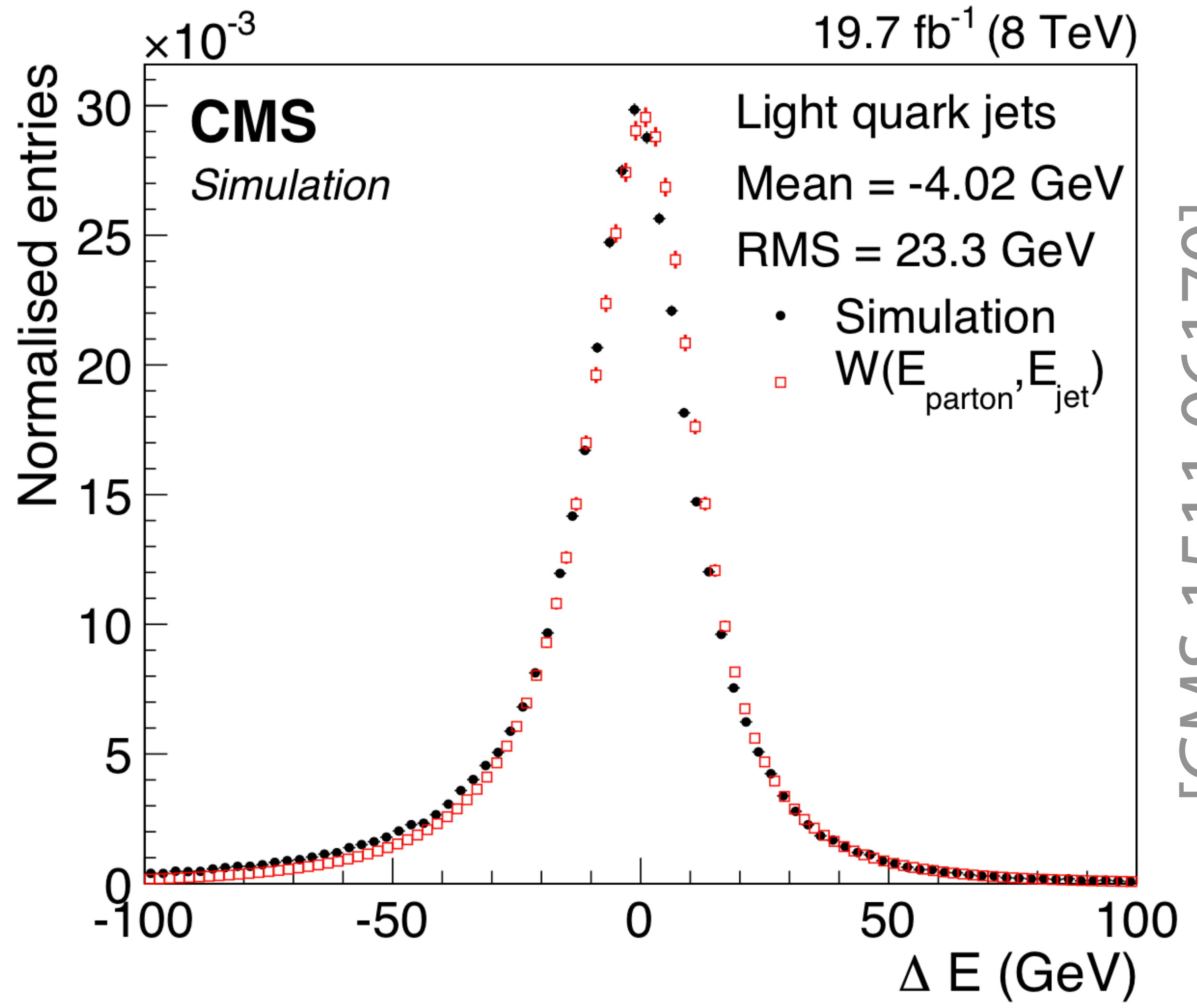
$$p(x|\theta) = \int dz_d \int dz_s \int dz_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$



[CMS]

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

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



[CMS 1511.06170]

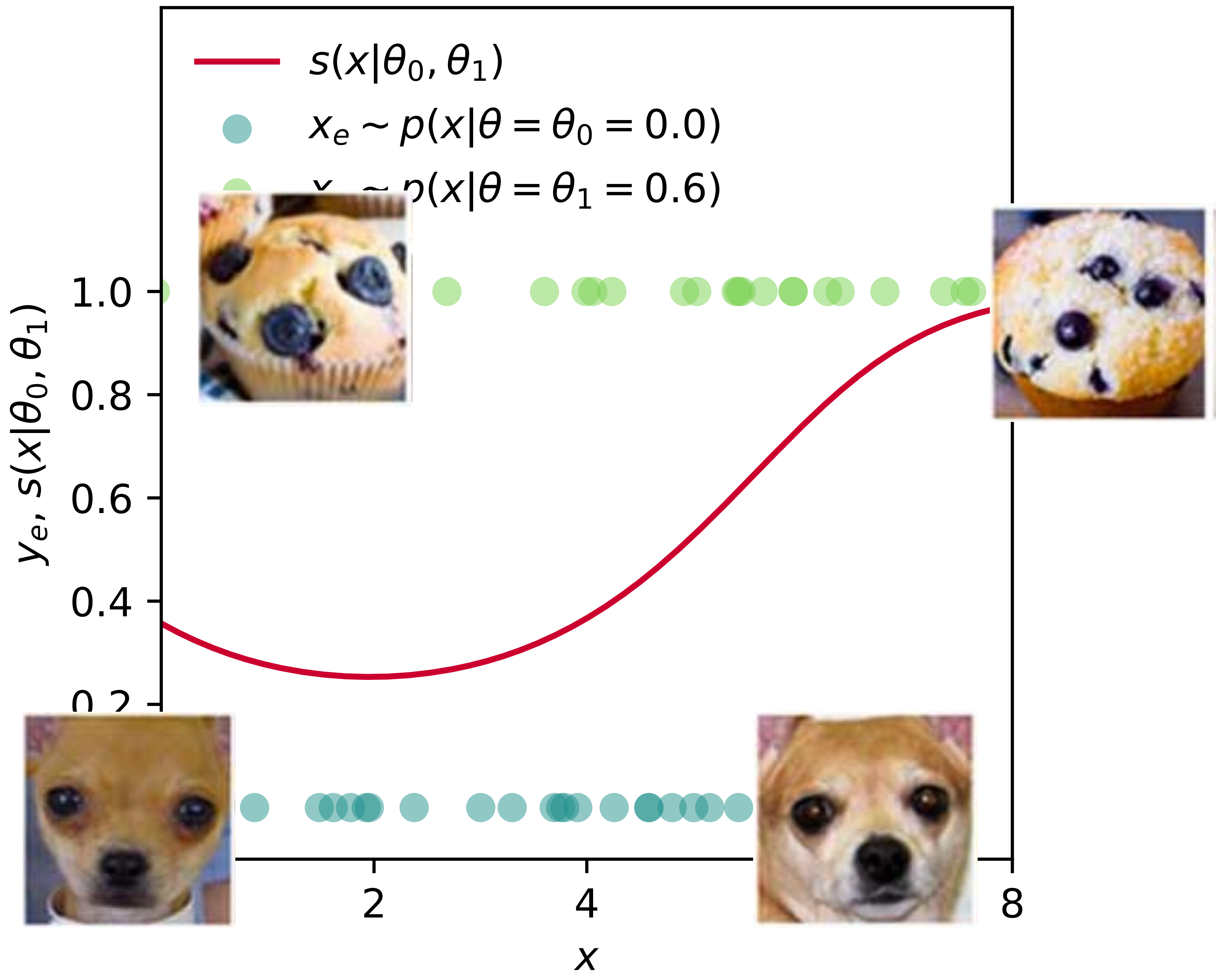
  - Shower / Event Deconstruction [D. E. Soper, M. Spannowsky 1102.3480] extend explicit calculation to the shower

⇒ 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]

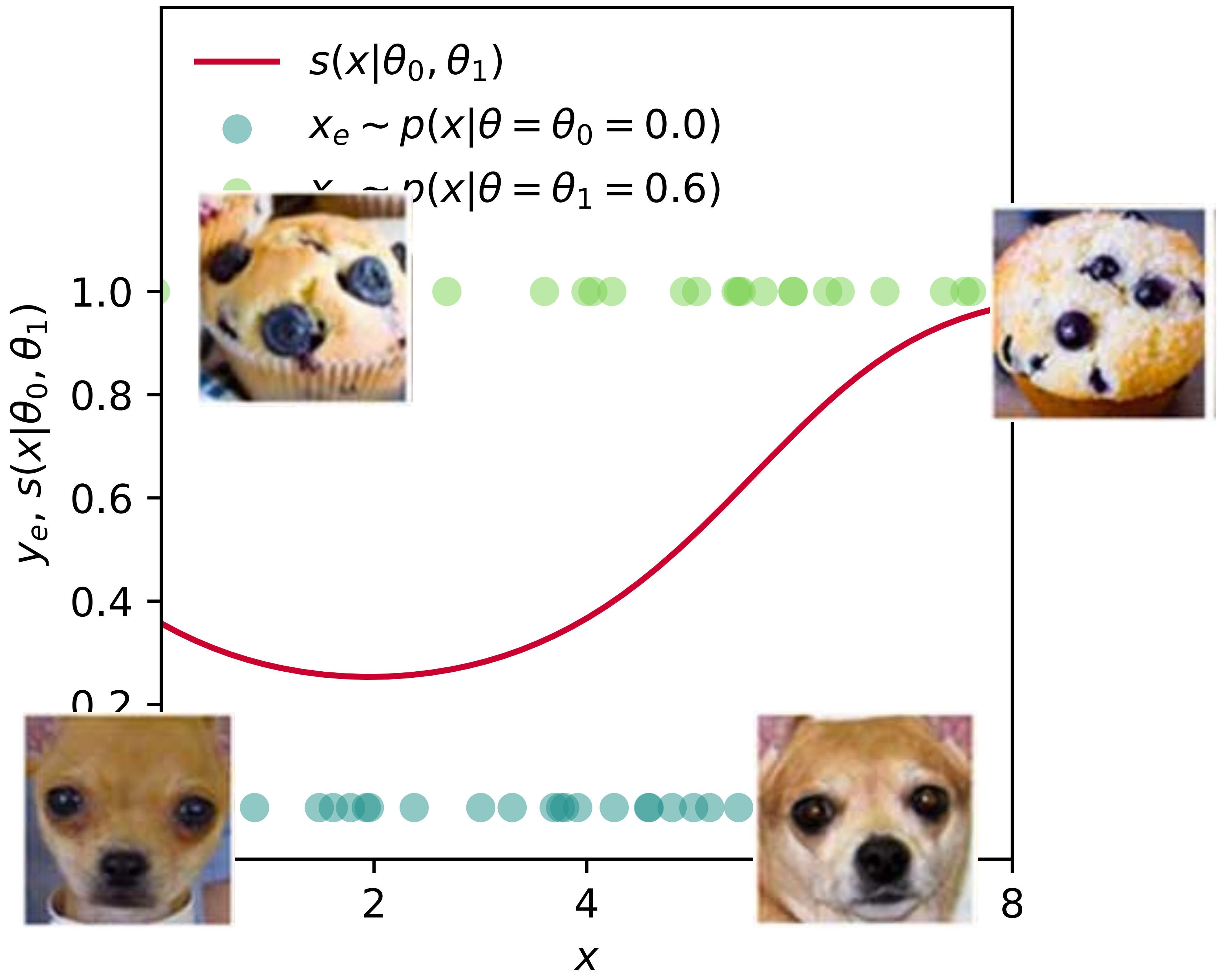
- Train neural network (BDT, ...) to tell $x \sim p(x|\theta_0)$ from $x \sim p(x|\theta_1)$
  - Classifier output $\hat{s}(x)$ is closer to 0 for θ_0-like events (closer to 1 for θ_1-like events)



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)



- Optimal classifier (i.e. minimizing the cross-entropy) converges to

$$s(x) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)} = \frac{1}{1 + \frac{p(x|\theta_0)}{p(x|\theta_1)}} \equiv \frac{1}{1 + r(x|\theta)}$$

- CARL: Turn (calibrated) classifier output $\hat{s}(x)$ into estimator for the likelihood ratio $r(x) \equiv p(x|\theta_0)/p(x|\theta_1)$:

$$\hat{r}(x) = \frac{1 - \hat{s}(x)}{\hat{s}(x)}$$

$\Rightarrow$ No summary statistics necessary, very fast evaluation… but may require large training samples

What if we could estimate the likelihood…

- for high-dimensional measurements, 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 ad-hoc transfer functions

- in microseconds?

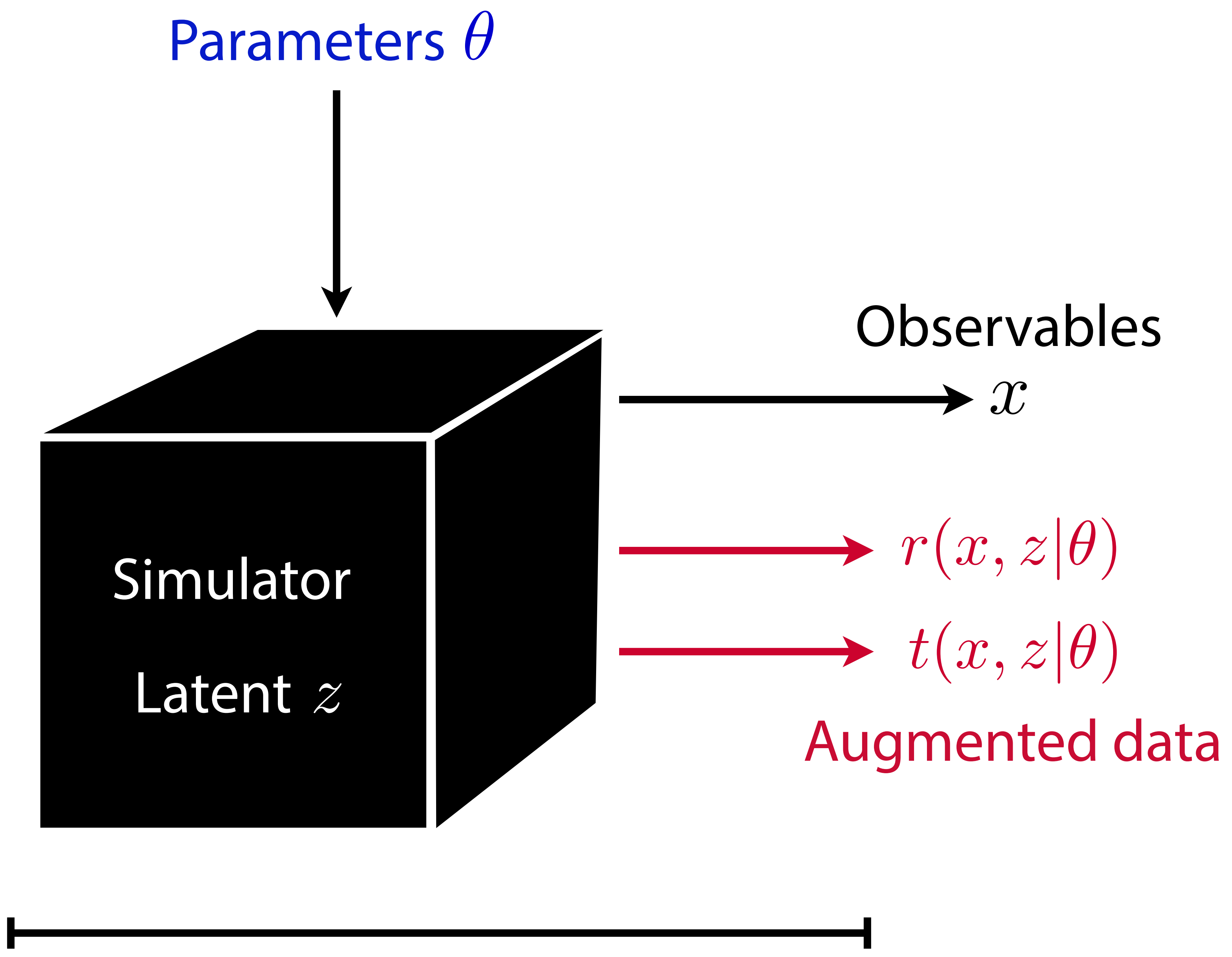
  amortized inference: train once, evaluate fast

- requiring less statistics than established machine learning methods?

  using matrix element information

A new approach:
Mining gold from the simulator

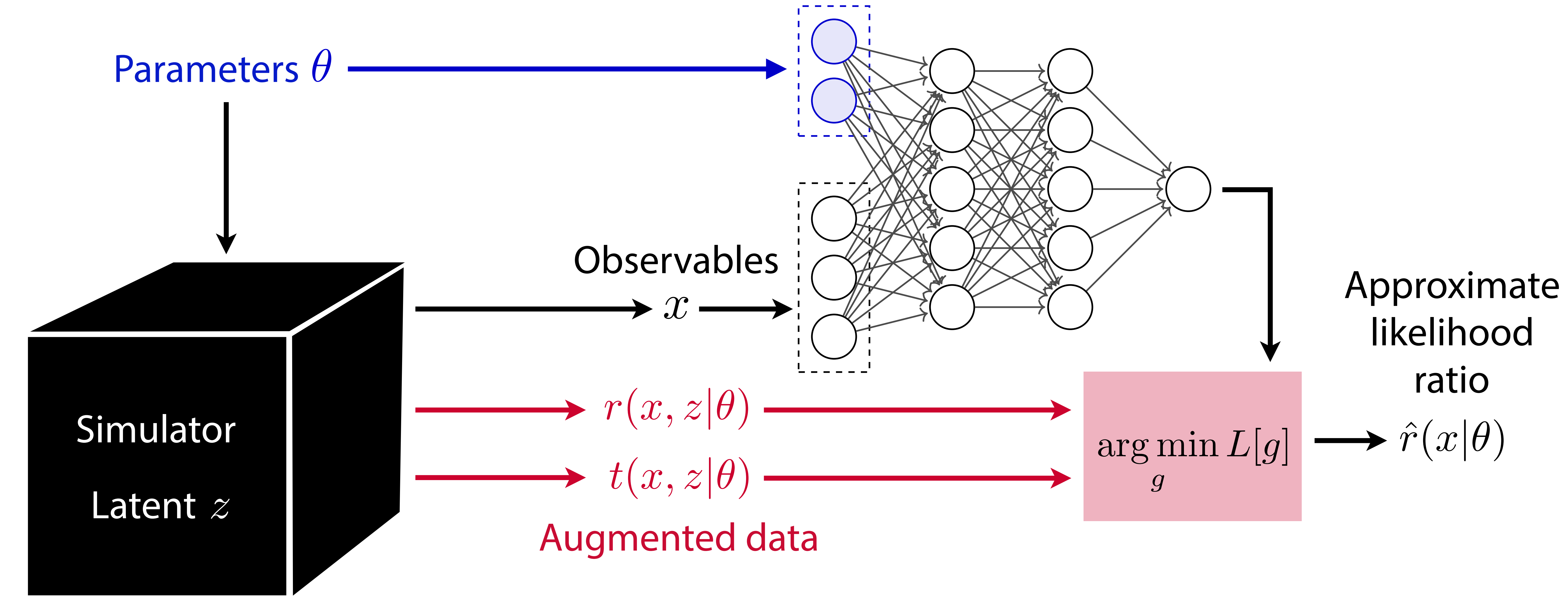
Bird's-eye view



1. Simulation

"Mining gold": Extract additional information from simulator

Bird’s-eye view



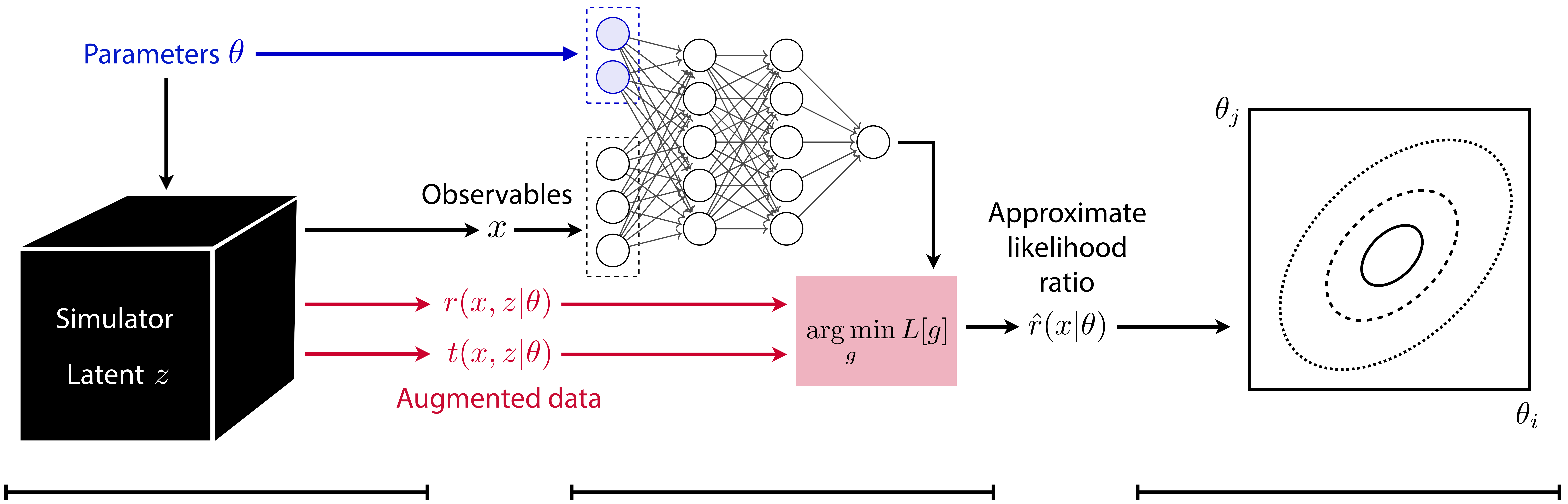
1. Simulation

2. Machine Learning

“Mining gold”: Extract additional information from simulator

Use this information to train estimator for likelihood ratio

Bird's-eye view



1. Simulation

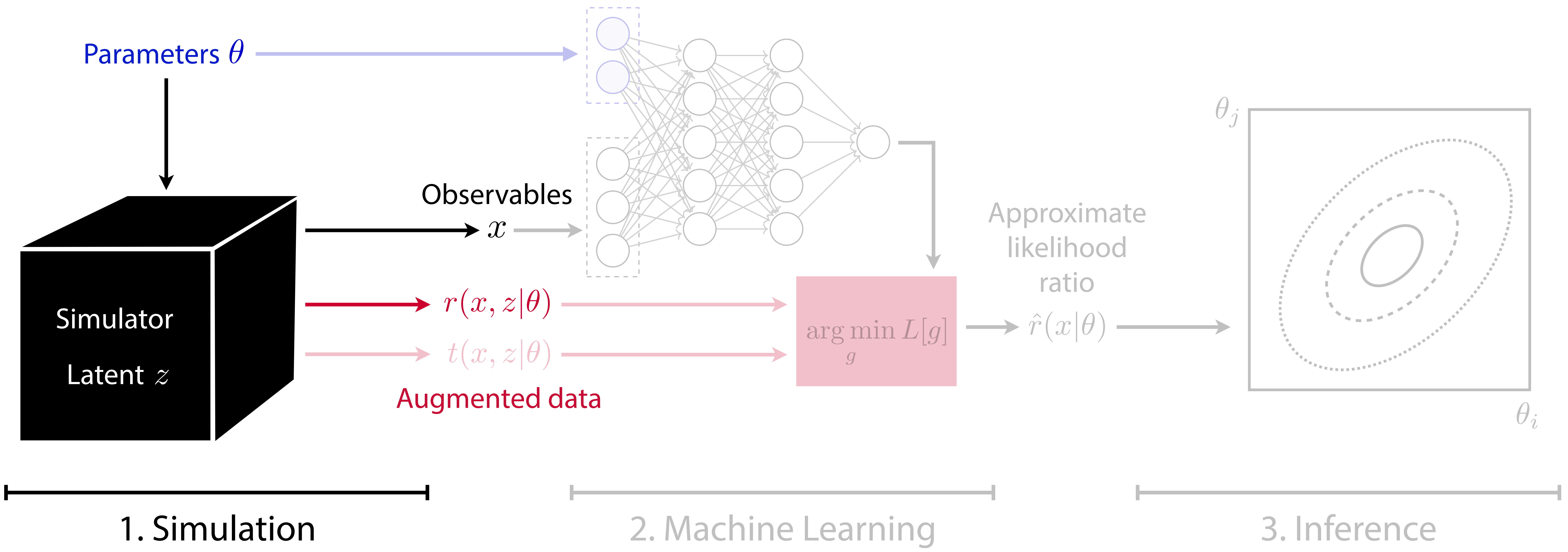
"Mining gold": Extract additional information from simulator

2. Machine Learning

Use this information to train estimator for likelihood ratio

3. Inference

Limit setting with standard hypothesis tests



Parameters θ
Observables
x
Simulator
Latent z
$r(x, z|\theta)$
$t(x, z|\theta)$
Augmented data
$\arg\min_g L[g]$
Approximate likelihood ratio
$\hat{r}(x|\theta)$
θ_j
θ_i
1. Simulation
2. Machine Learning
3. Inference

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

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

Intractable integrals

Mining gold from the simulator

Latent variables

Observables		Detector interactions	Shower splittings	Parton-level momenta	Theory parameters

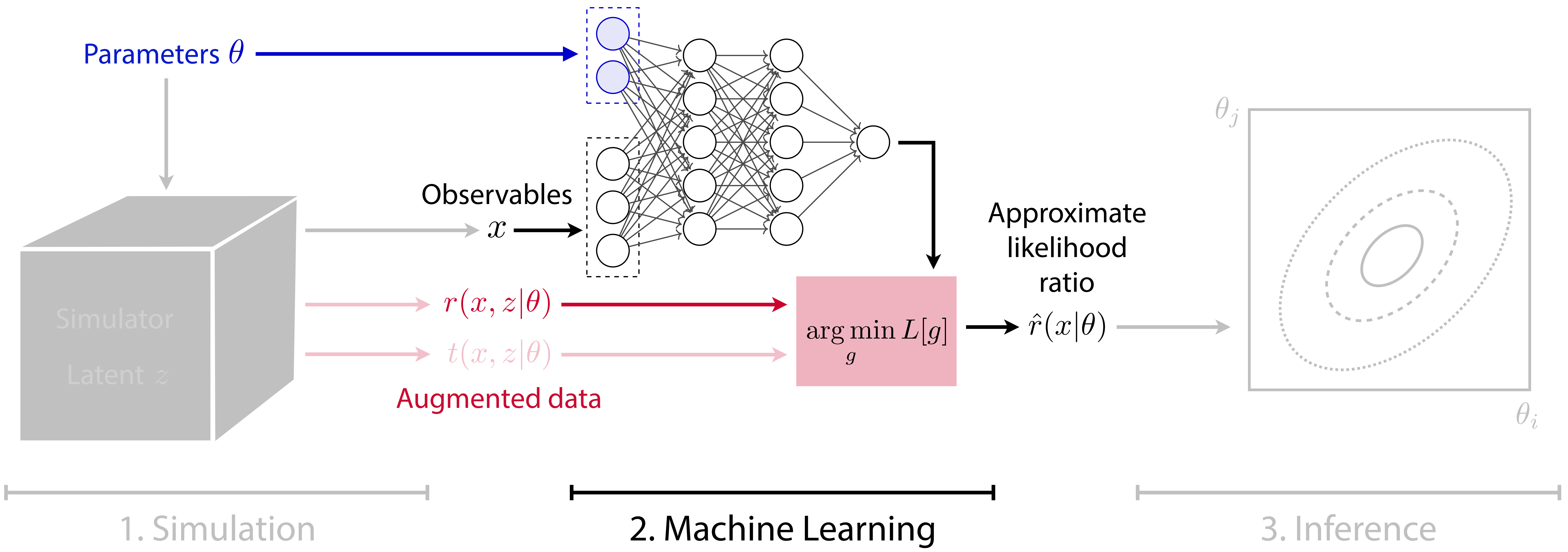
$$x \longleftarrow z_d \longleftarrow z_s \longleftarrow z_p \longleftarrow \theta$$

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

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

⇒ For each generated event, we can calculate the joint likelihood ratio conditional on its specific evolution:

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



Parameters θ
Observables
x
Simulator
Latent z
$r(x, z|\theta)$
$t(x, z|\theta)$
Augmented data
$\arg\min_g L[g]$
Approximate likelihood ratio
$\hat{r}(x|\theta)$
θ_j
θ_i
1. Simulation
2. Machine Learning
3. Inference

The value of gold

We can calculate the joint likelihood ratio

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

$$\downarrow$$

We want the likelihood ratio function

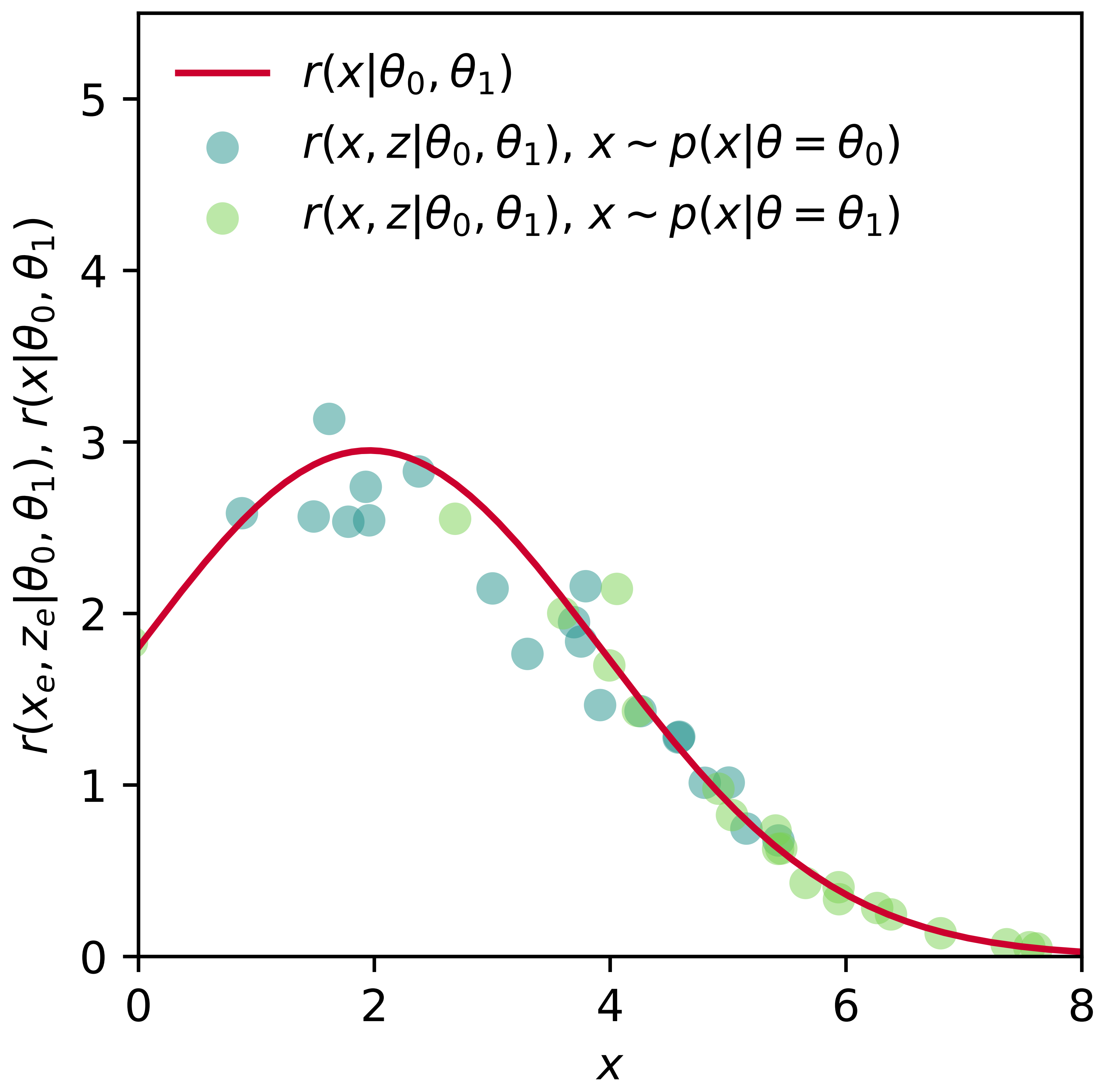
$$r(x|\theta_0, \theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

The value of gold

We can calculate the joint likelihood ratio

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

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

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

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

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

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

It is minimized by

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

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

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 variatios

We can get a precise estimator of the likelihood ratio by numerically mimimizing a functional:

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

Machine learning = applied calculus of variatios

We can get a precise estimator of the likelihood ratio by numerically mimimizing a functional:

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

We do this through machine learning:

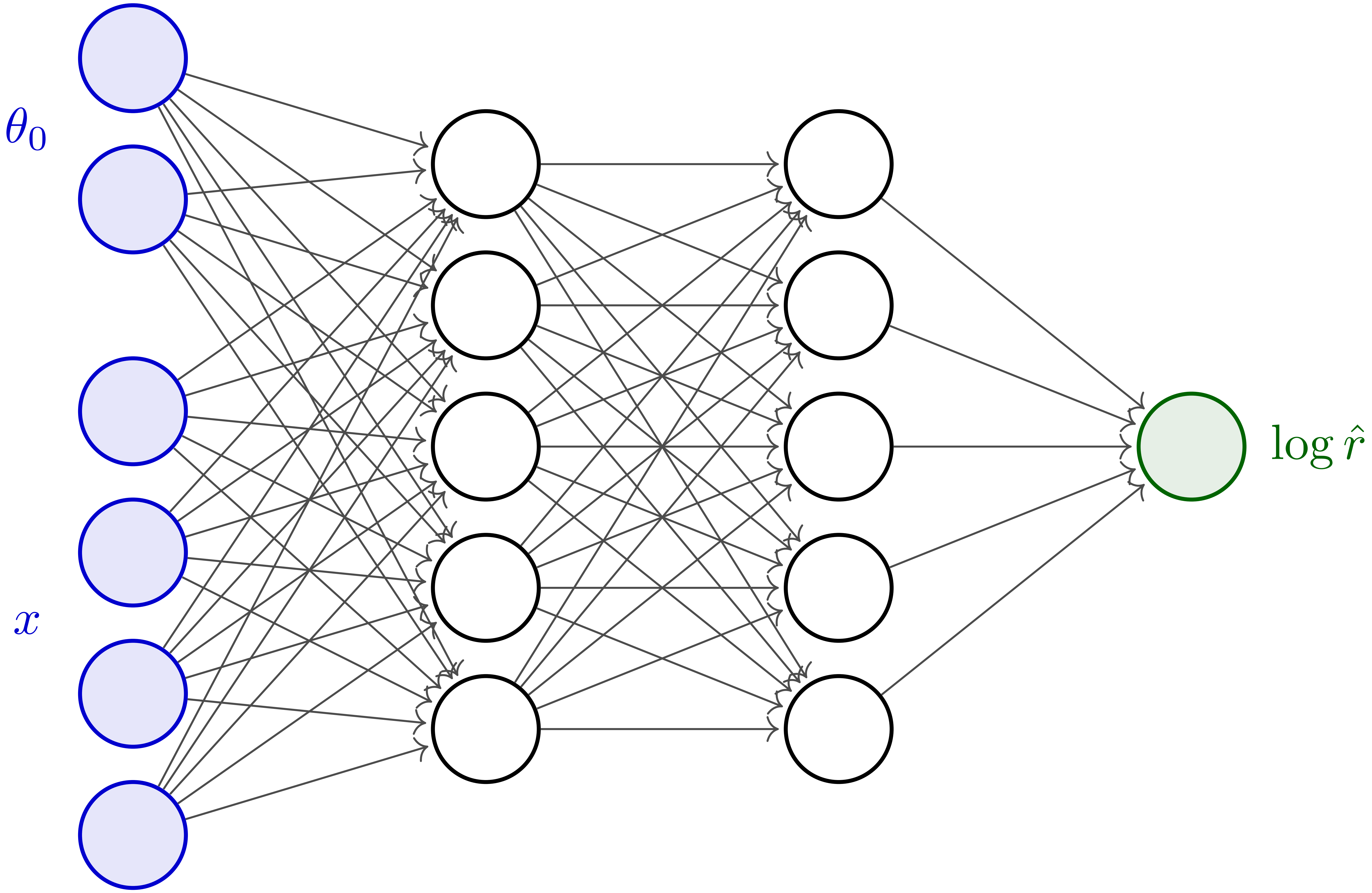
- Functional L_r $\longrightarrow$ Loss function

$$\hat{L}_r[\hat{r}(x|\theta_0,\theta_1)] = \frac{1}{N} \sum_{(x_i,z_i)\sim p(x,z|\theta_1)} \left[\hat{r}(x_i|\theta_0,\theta_1) - r(x_i,z_i|\theta_0,\theta_1)\right]^2$$

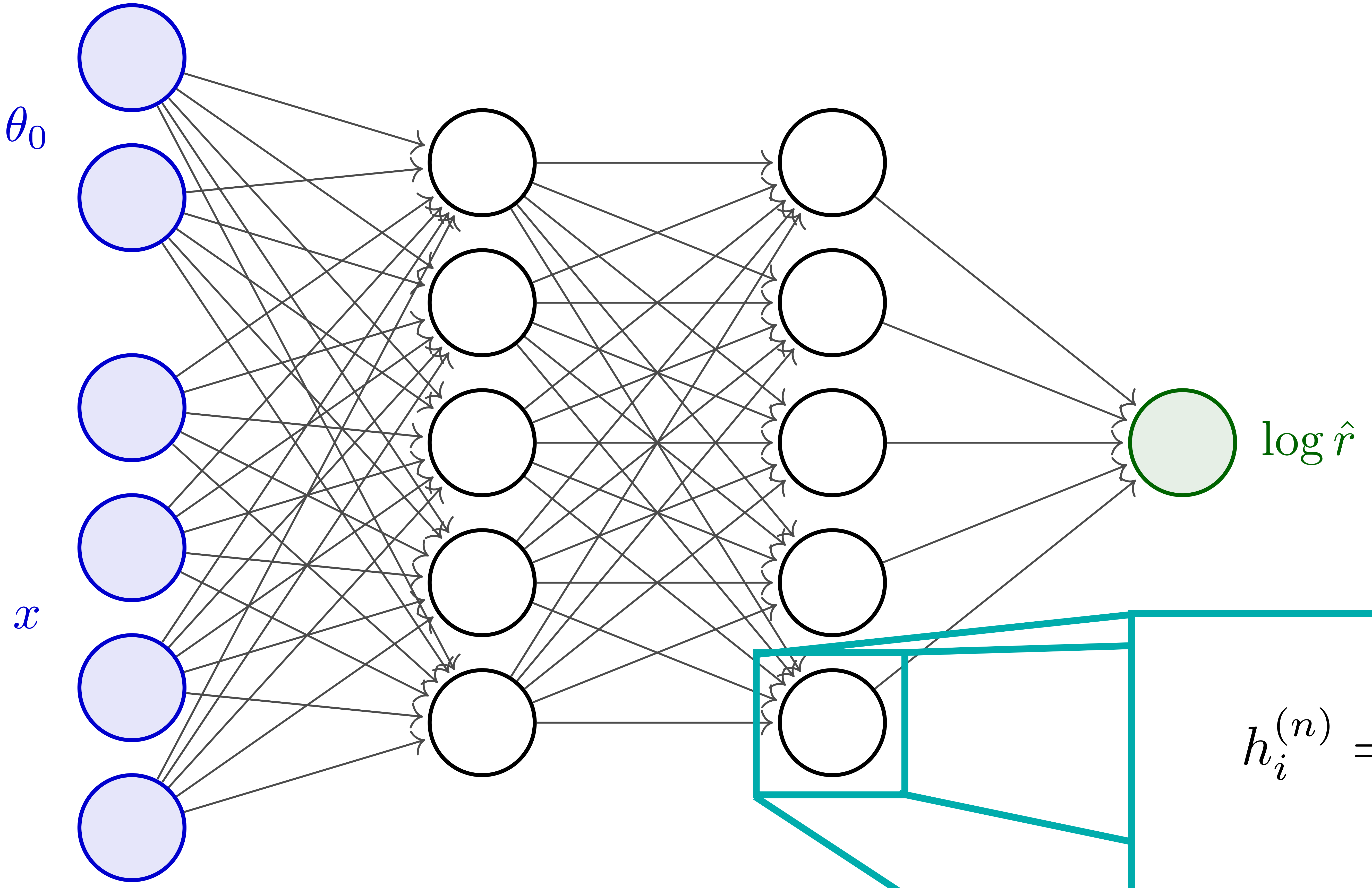
- Variational family $\hat{r}(x|\theta_0,\theta_1)$ $\longrightarrow$ Flexible parametric function (e.g. neural network)
- Exact minimization $\longrightarrow$ Numerical optimization algorithm (e.g. stochastic gradient descent)

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

Neural networks = universal function approximators

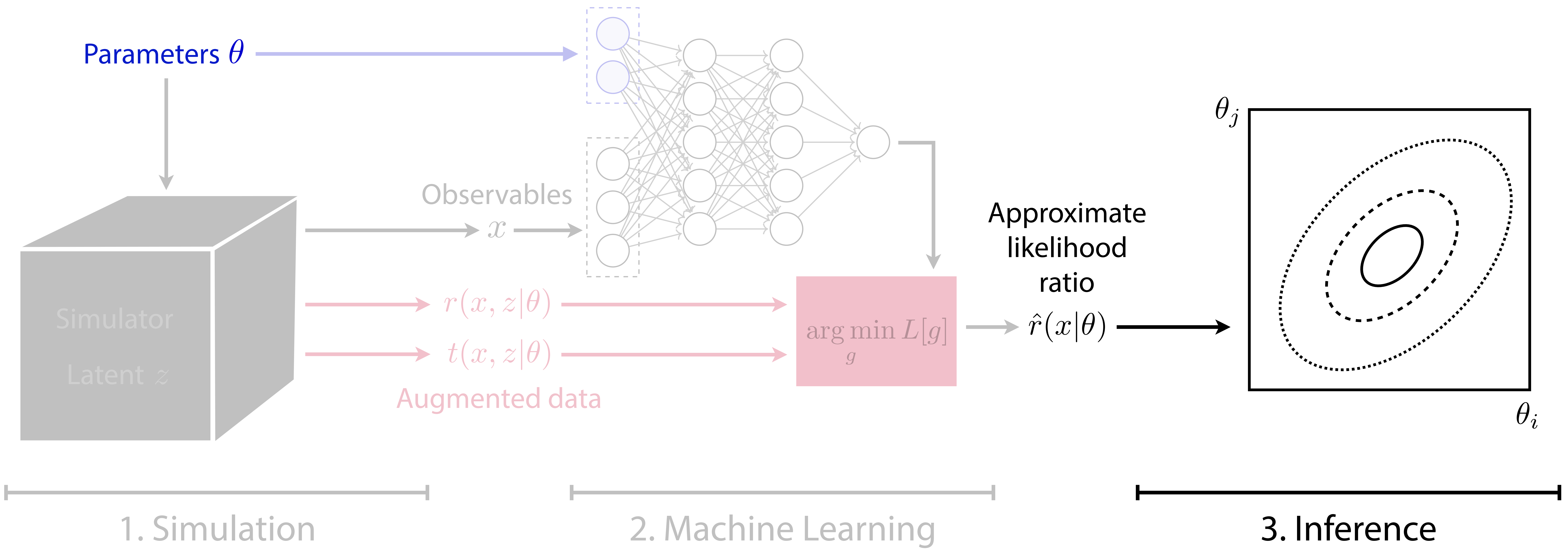


Neural networks = universal function approximators



$$h_i^{(n)} = f\left(\sum_k w_{ik}^{(n)}\, h_k^{(n-1)} + b_i^{(n)}\right)$$

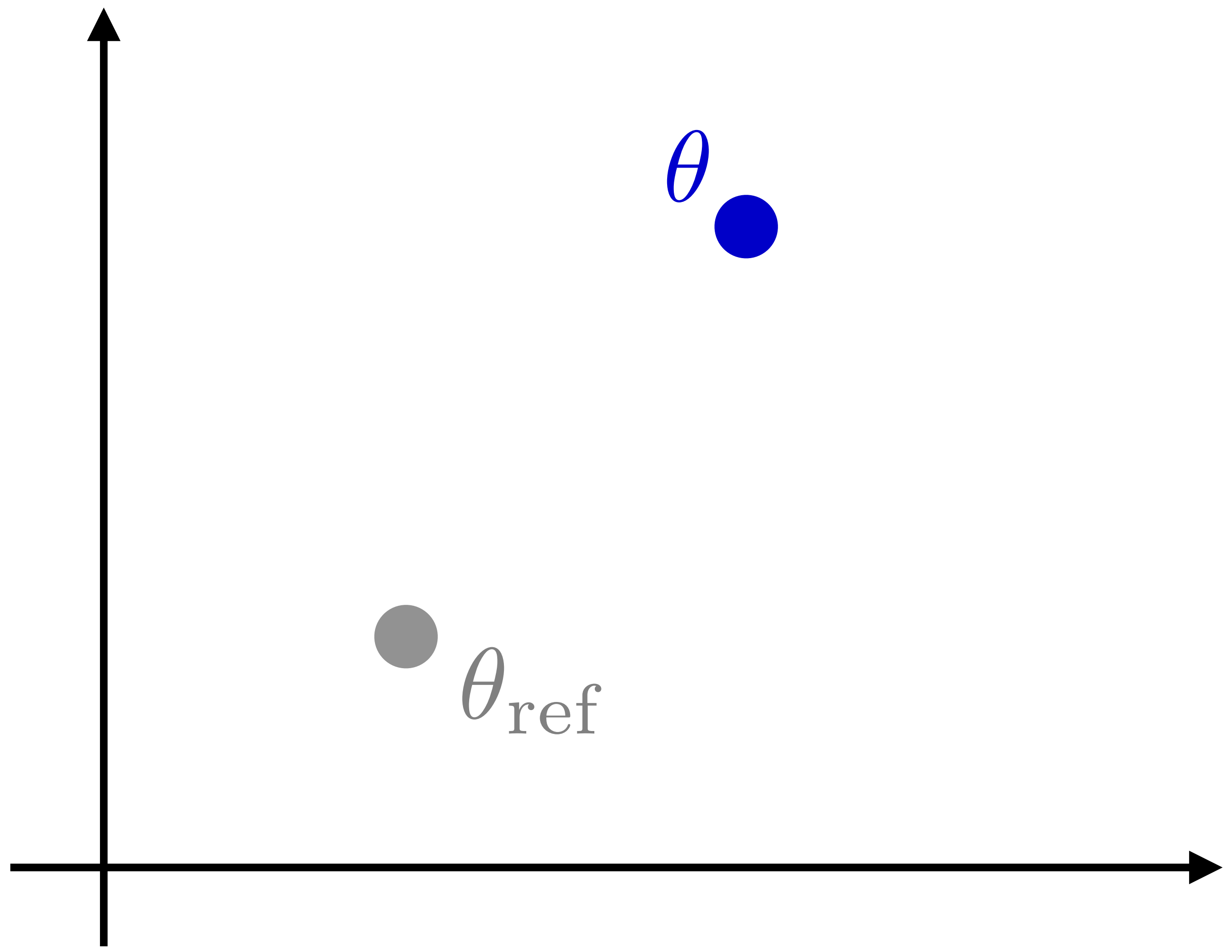
where the weights $w_{ik}^{(n)}$, $b_i^{(n)}$ are parameters “trained” by the optimizer


Parameters θ
Observables
x
$r(x, z|\theta)$
$t(x, z|\theta)$
Augmented data
Simulator
Latent z
$\arg\min_g L[g]$
Approximate likelihood ratio
$\hat{r}(x|\theta)$
θ_j
θ_i
1. Simulation
2. Machine Learning
3. Inference

Limit setting (frequentist)

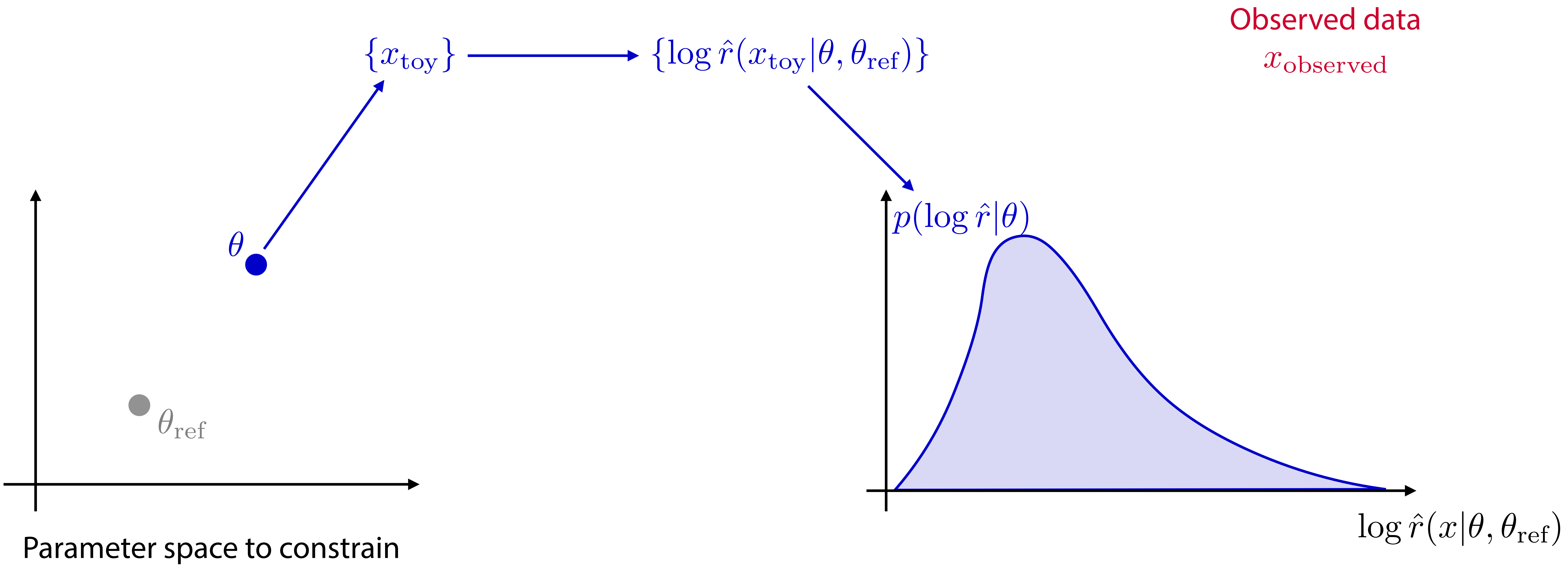
Observed data

x_{observed}

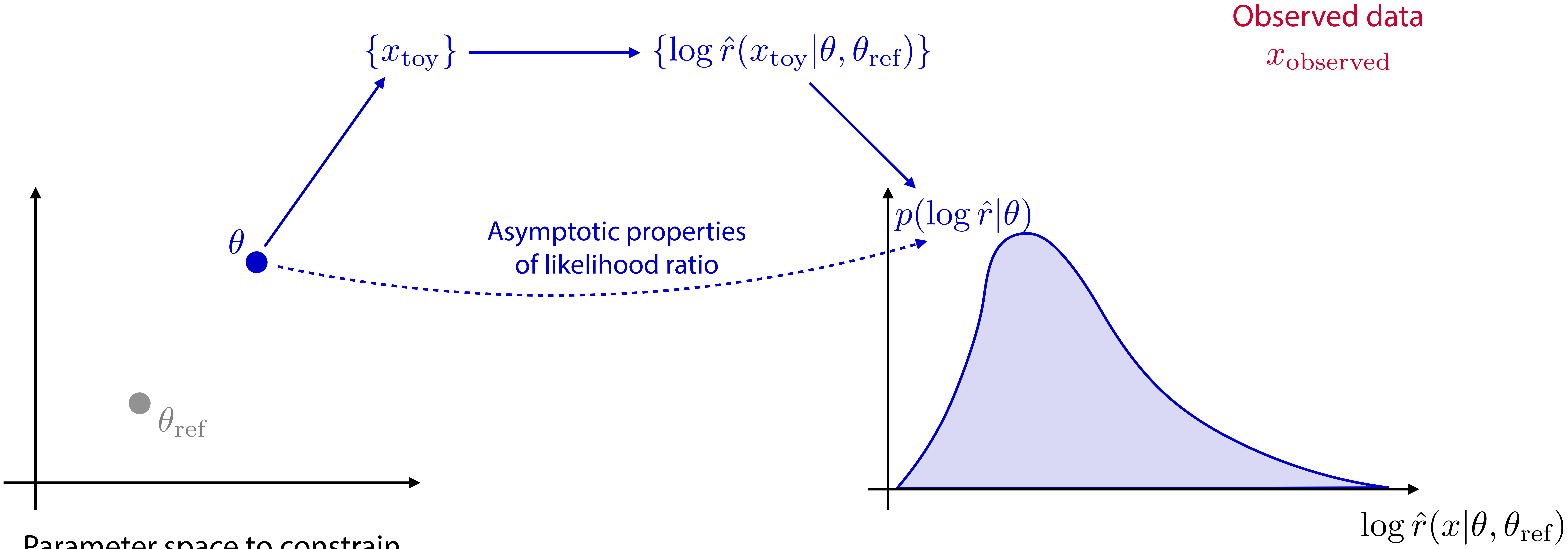


Parameter space to constrain

Limit setting (frequentist)

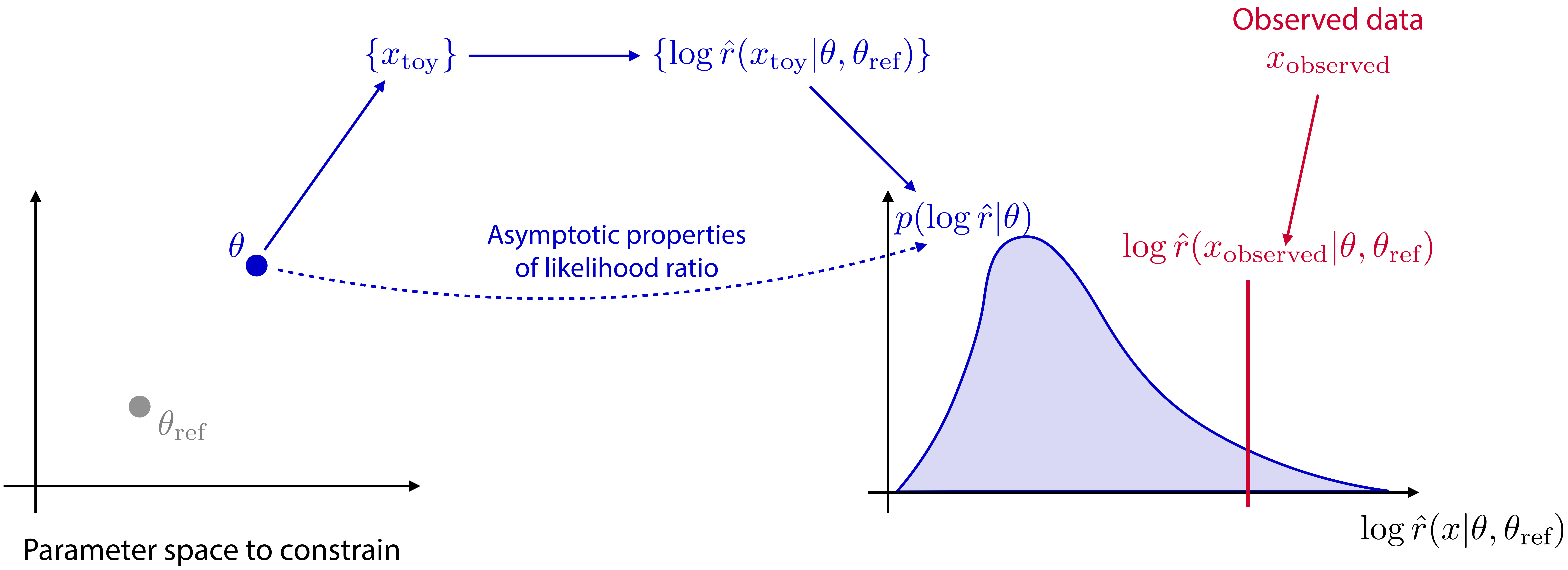


Limit setting (frequentist)

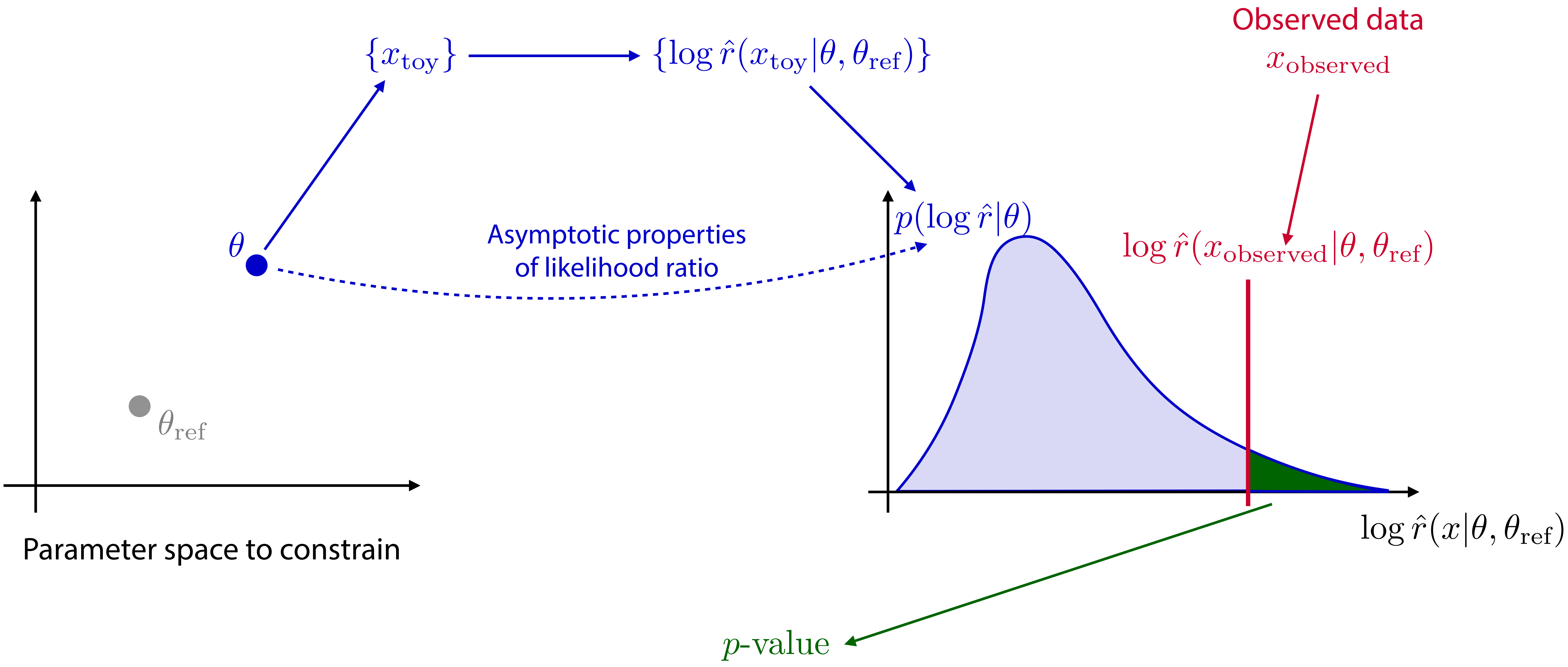


Parameter space to constrain

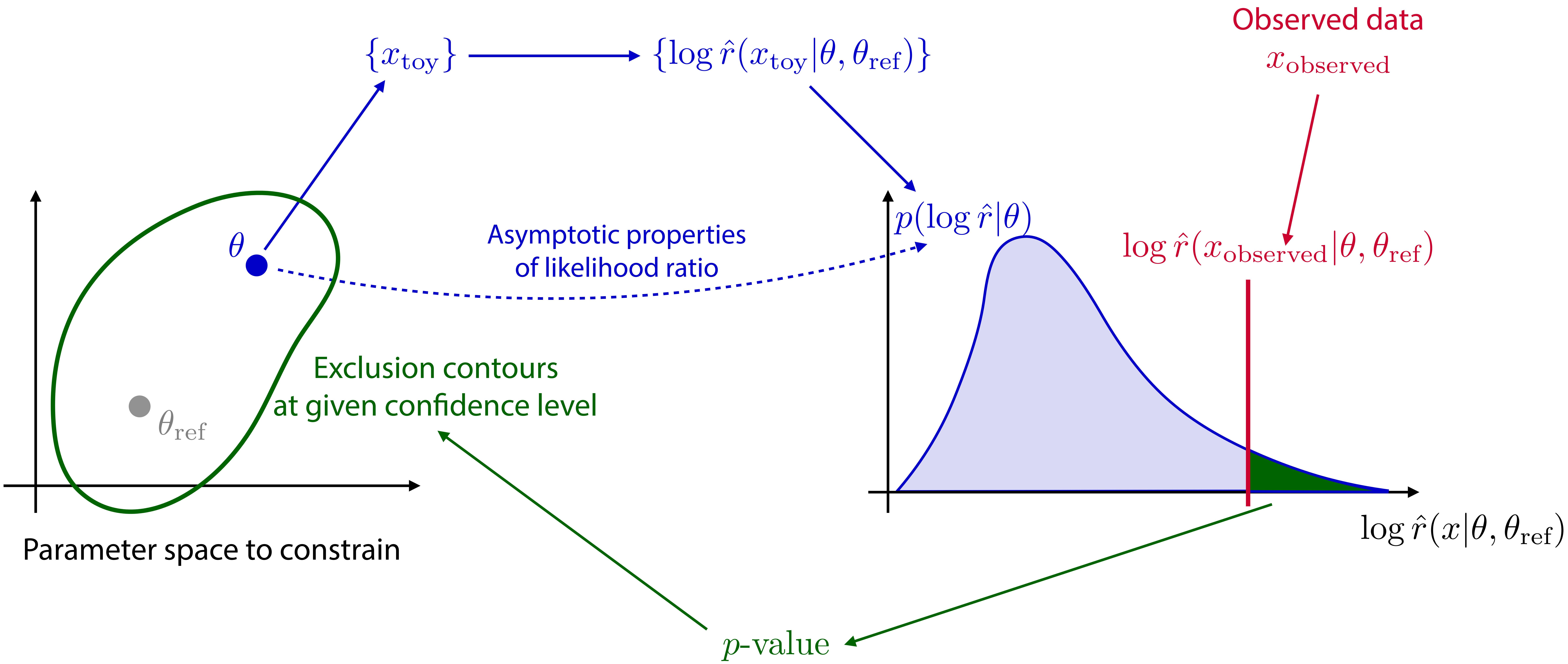
Limit setting (frequentist)



Limit setting (frequentist)



Limit setting (frequentist)



The likelihood ratio is the most powerful test statistic

IX. *On the Problem of the most Efficient Tests of Statistical Hypotheses.*

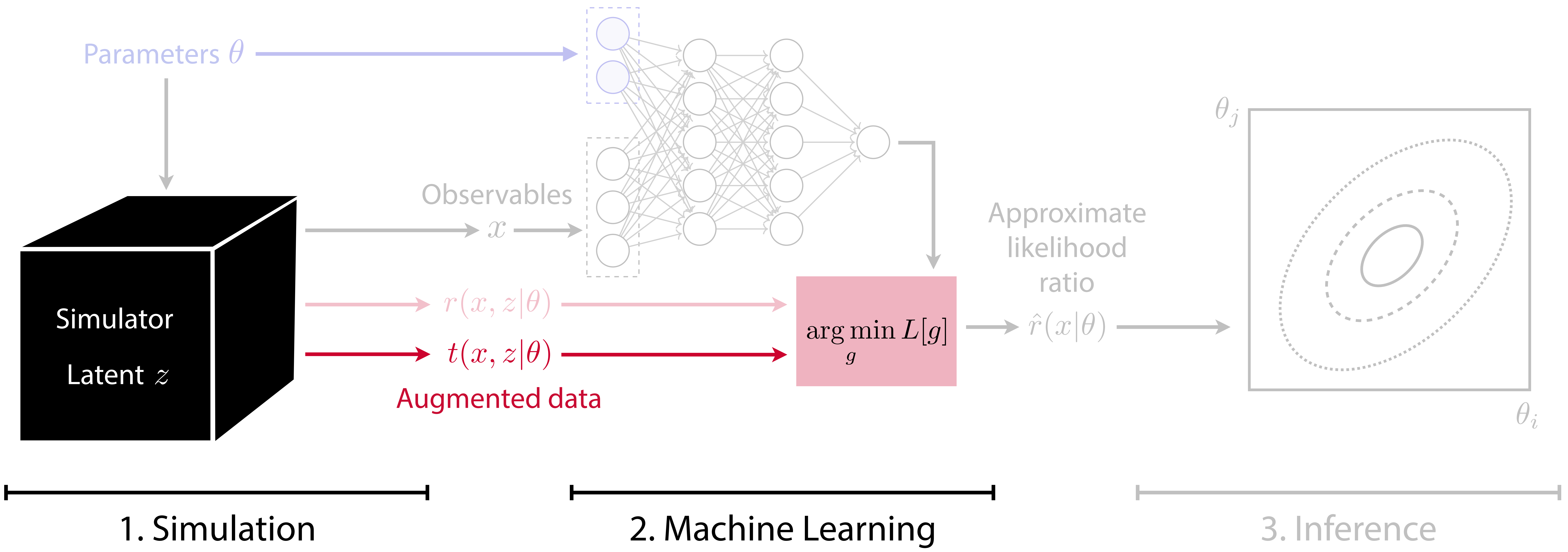
By J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

(*Communicated by* K. PEARSON, *F.R.S.*)

(Received August 31, 1932.—Read November 10, 1932.)

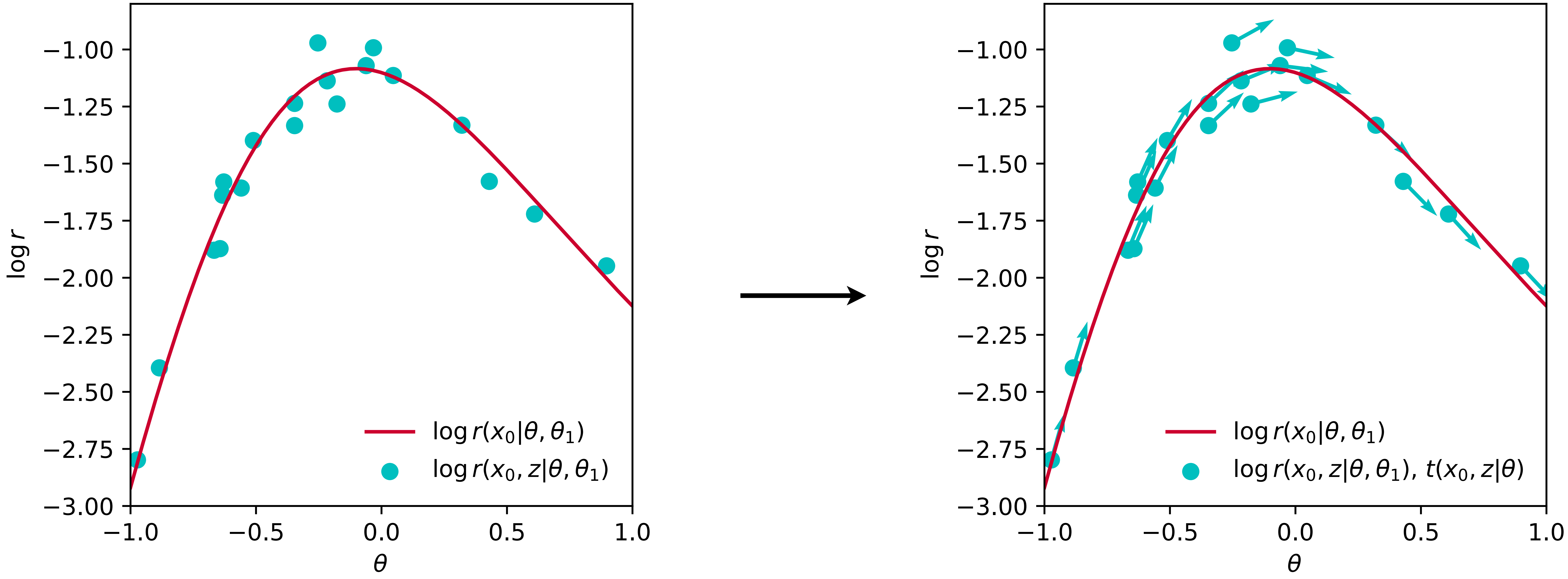
CONTENTS.

		PAGE.
I.	Introductory	289
II.	Outline of General Theory	293
III.	Simple Hypotheses :—	


Parameters θ
Observables
x
Simulator
Latent z
$r(x, z|\theta)$
$t(x, z|\theta)$
Augmented data
$\arg\min_g L[g]$
Approximate likelihood ratio
$\hat{r}(x|\theta)$
θ_j
θ_i
1. Simulation
2. Machine Learning
3. Inference

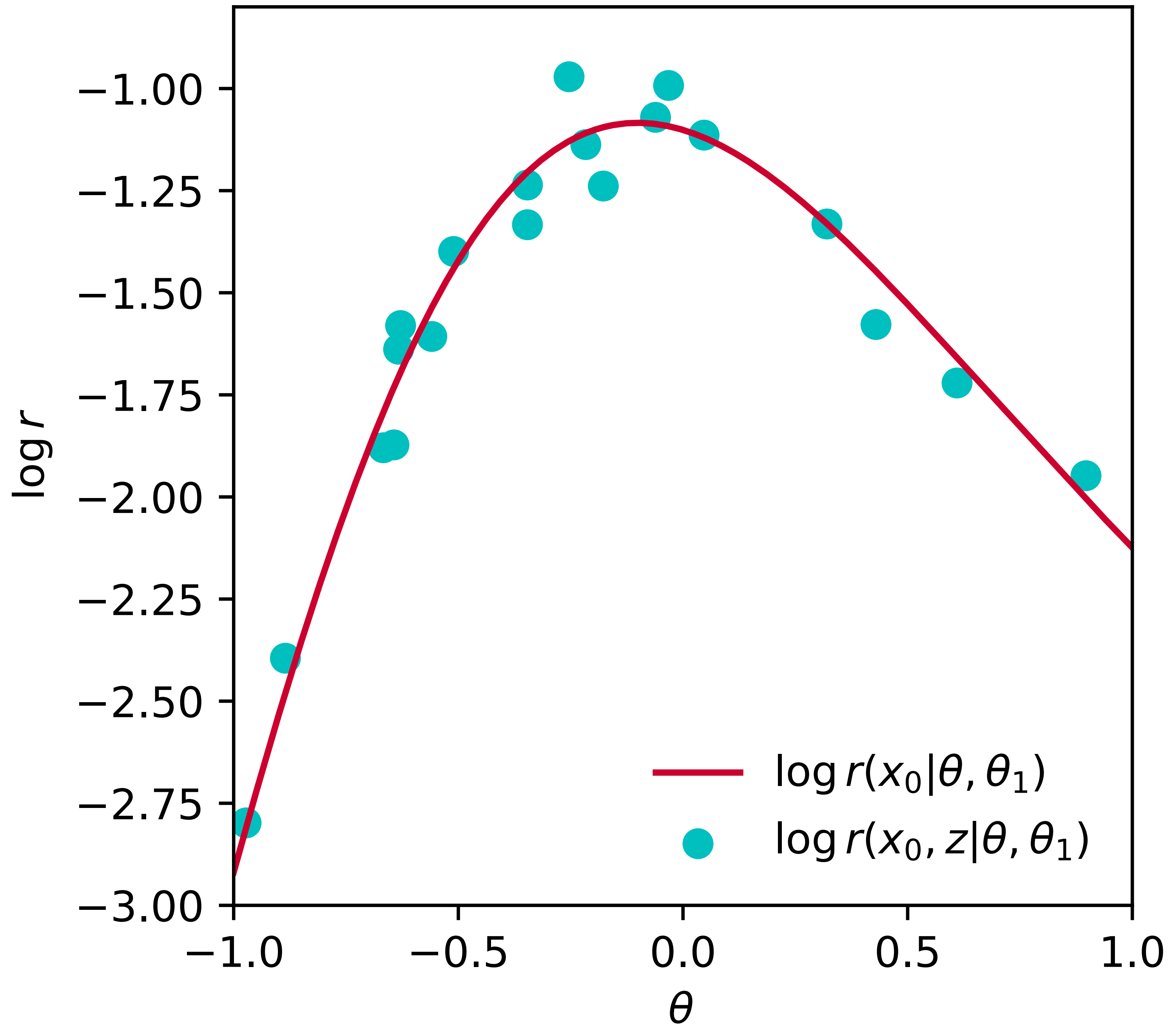
One more piece: the score

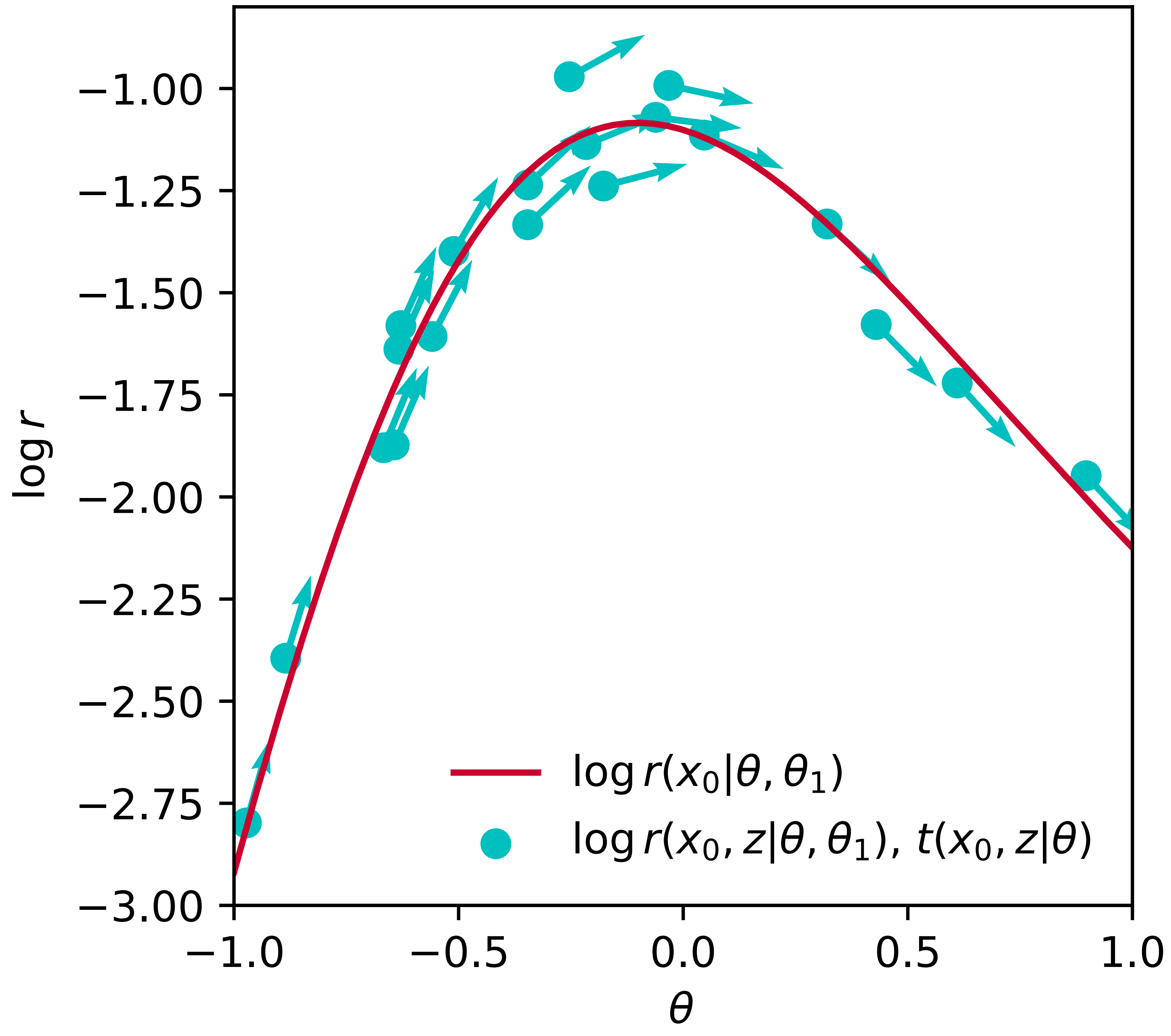
- Knowing derivative often helps fitting:



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

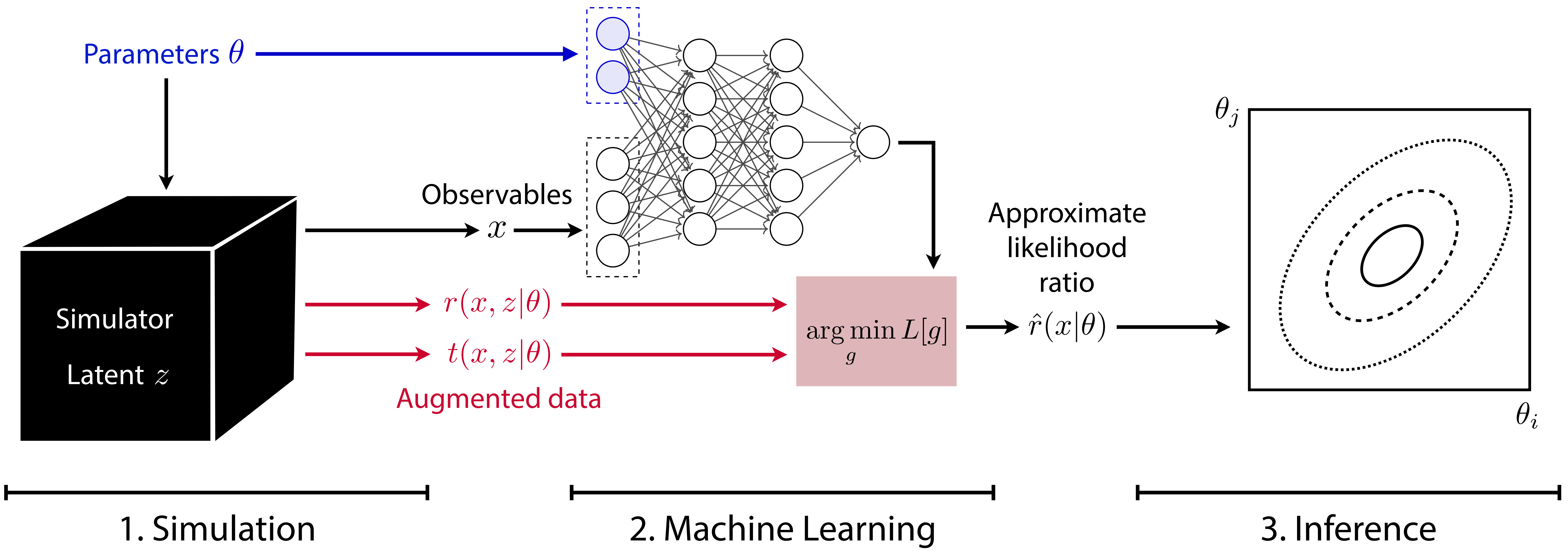
$$L_t[\hat{t}(x|\theta_0)] = \int \mathrm{d}x \int \mathrm{d}z \;\; p(x, z|\theta_0) \; \left[\left(\hat{t}(x|\theta_0) - t(x, z|\theta_0)\right)^2\right].$$

One can show it is minimized by

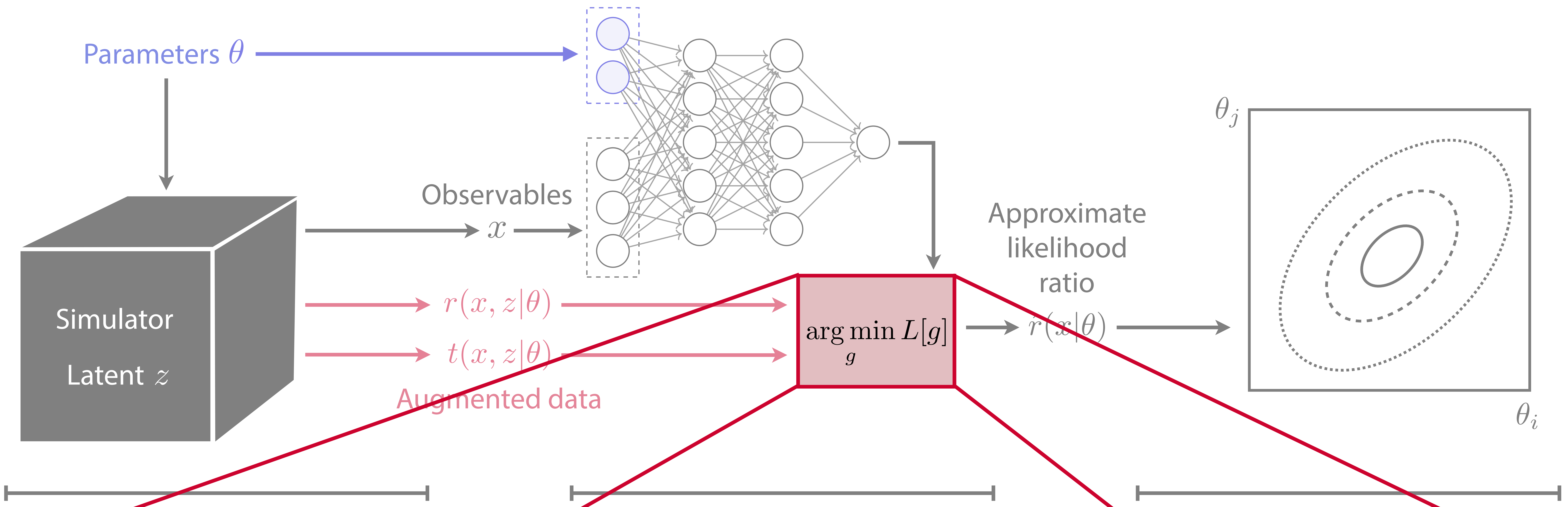
$$t(x|\theta_0) = \underset{\hat{t}(x|\theta_0)}{\arg\min}\, L_t[\hat{t}(x|\theta_0)]\,.$$

Again, we implement this minimization through machine learning.

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



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

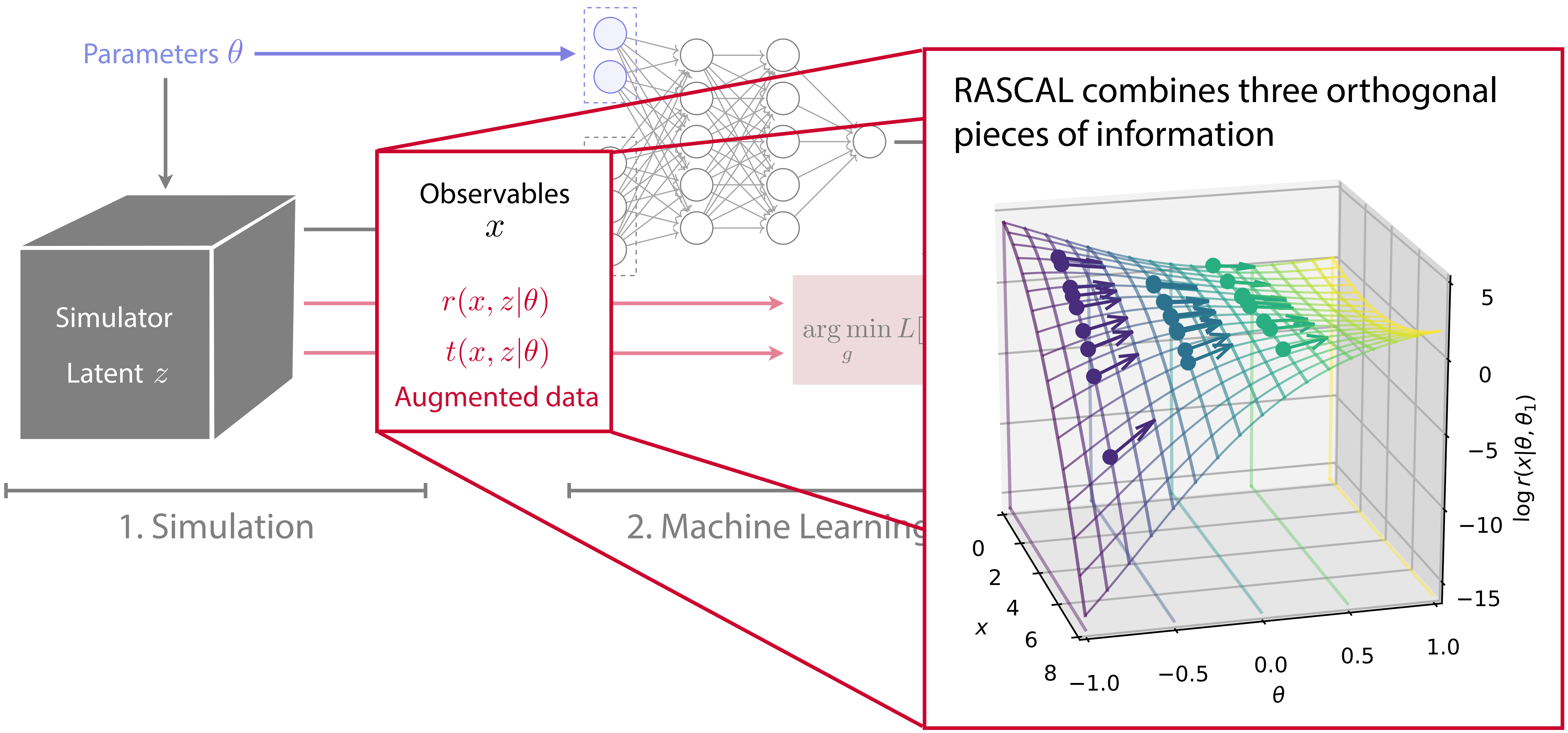


Combining ratio and score information:

- If ratio estimator is differentiable, we can calculate the estimated score $\hat{t}(x|\theta_0) = \nabla_{\theta_0} \log \hat{r}(x|\theta_0, \theta_1)$
- Minimize combined loss function:

$$L_{\mathrm{RASCAL}}[\hat{r}(x|\theta_0,\theta_1)] = L_r[\hat{r}(x|\theta_0,\theta_1)] + \alpha\, L_t[\nabla_{\theta_0} \log \hat{r}(x|\theta_0,\theta_1)]$$

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



Isn’t there a simpler way?

- or -

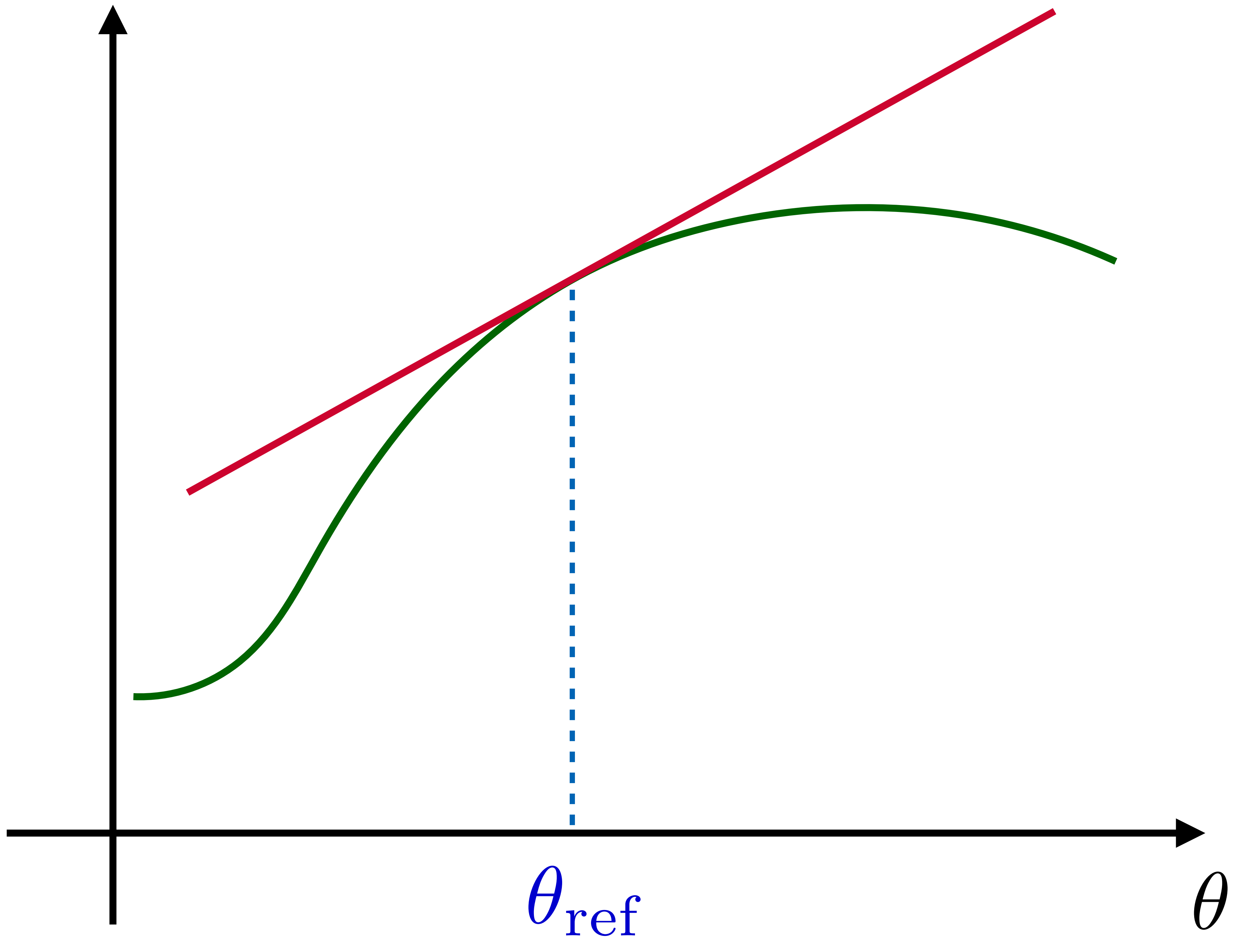
Can the machine tell me the most powerful observables?

The local model

[see also J. Alsing, B. Wandelt 1712.00012; J. Alsing, B. Wandelt. S. Freeney 1801.01497]

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

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

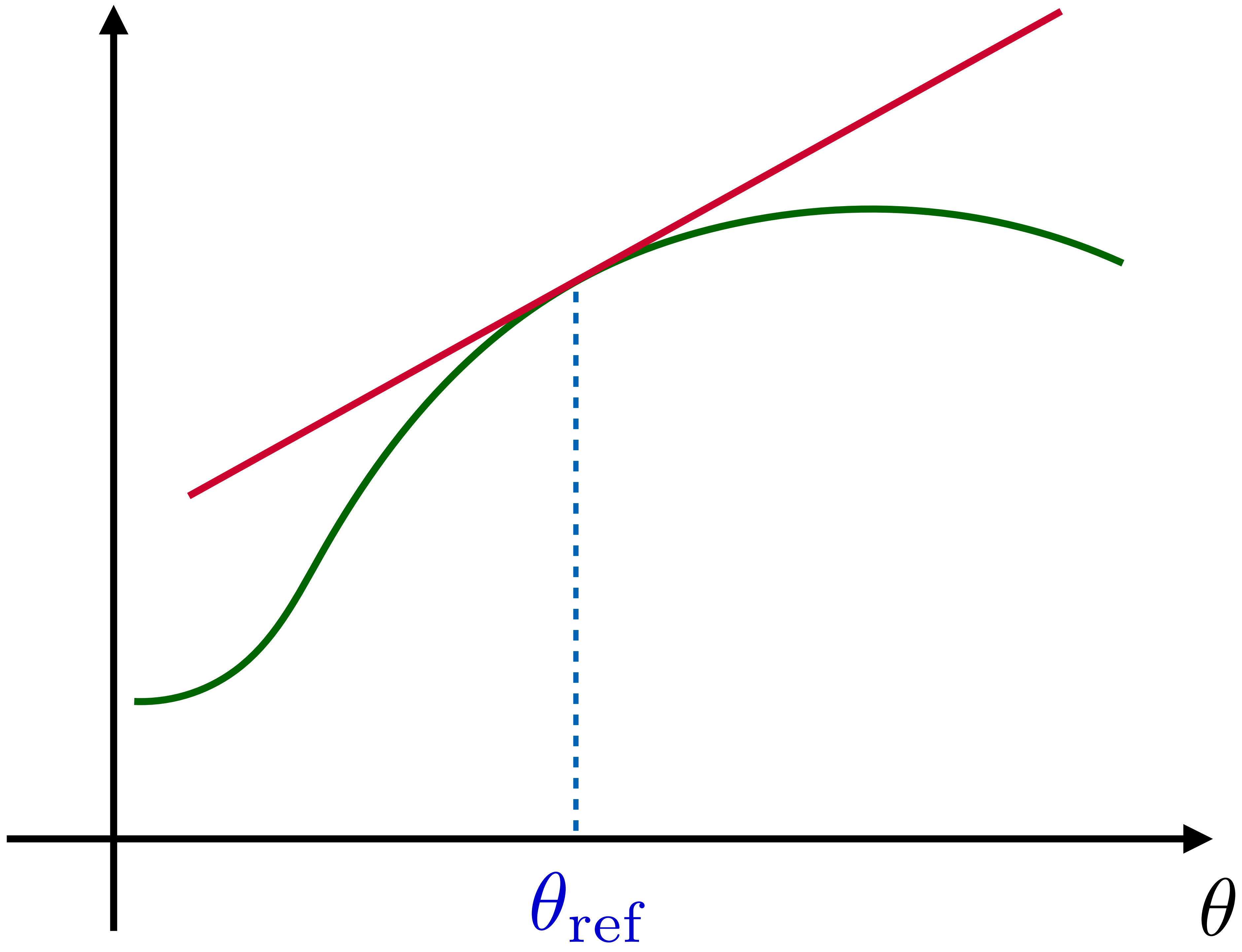


The local model

[see also J. Alsing, B. Wandelt 1712.00012;
J. Alsing, B. Wandelt. S. Freeney 1801.01497]

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

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

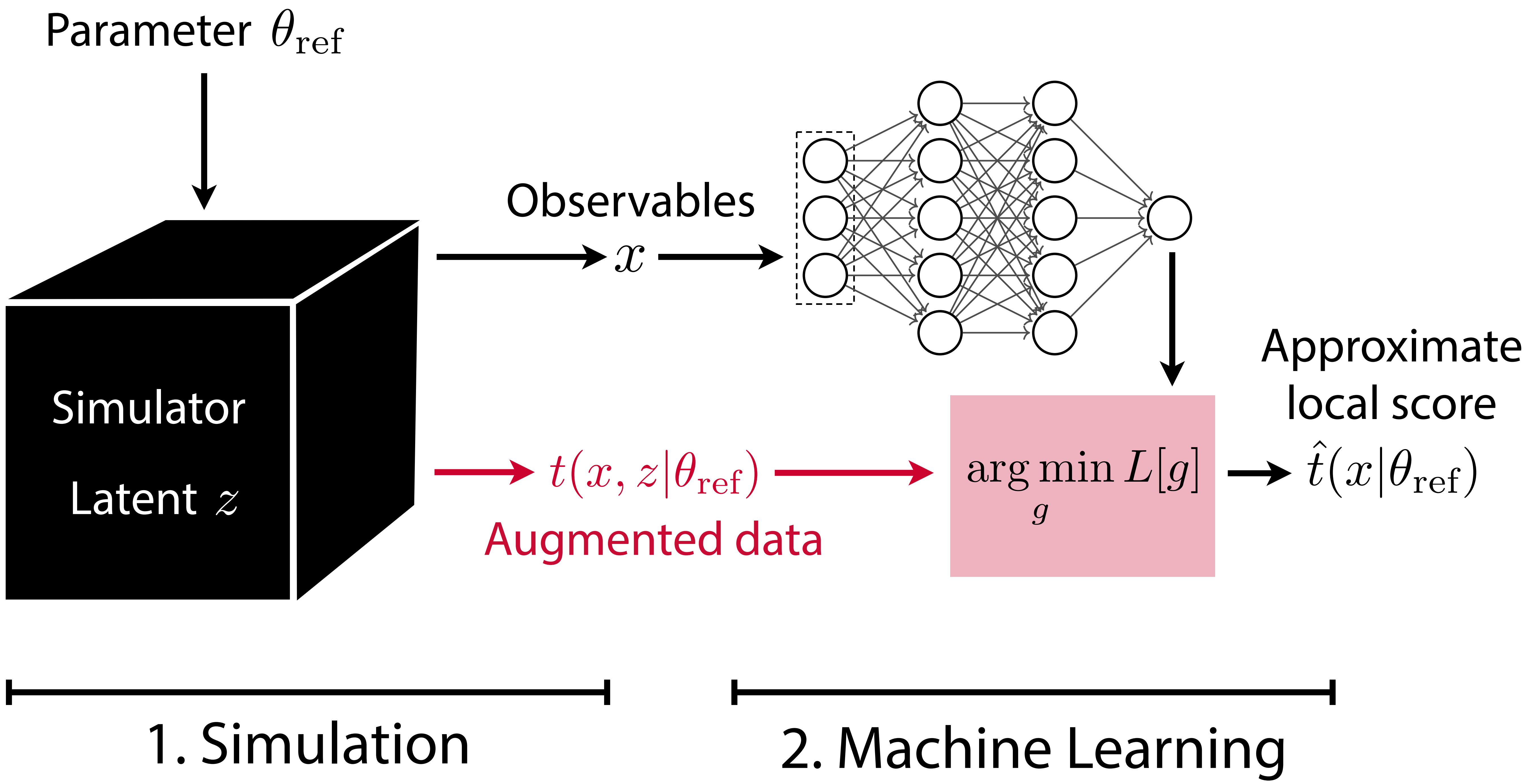


$\Rightarrow$ In the neighborhood of θ_{ref} (e.g. close to the SM), the score vector $t(x|\theta_{\text{ref}})$ is the sufficient statistics: it contains all information on θ

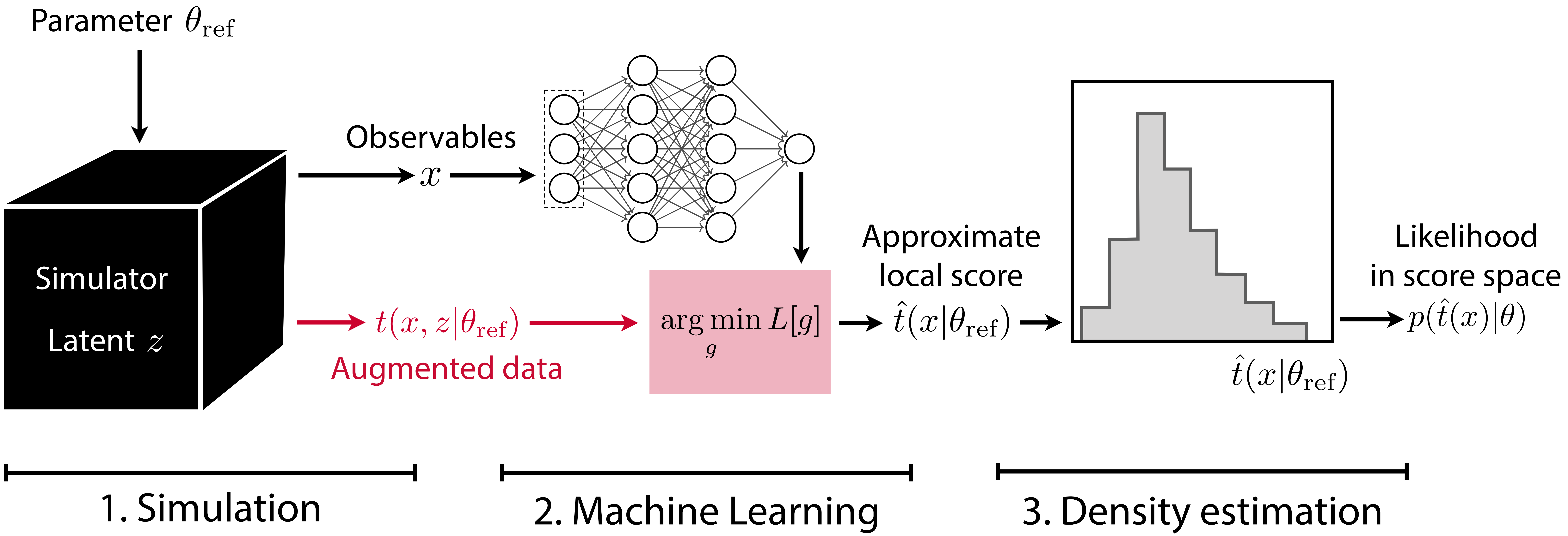
In this part of parameter space, knowing $t(x|\theta_{\text{ref}})$ is just as powerful as knowing the full function $\log p(x|\theta)$

$t(x|\theta_{\text{ref}})$ does not depend on θ — it's (locally) the most powerful observable

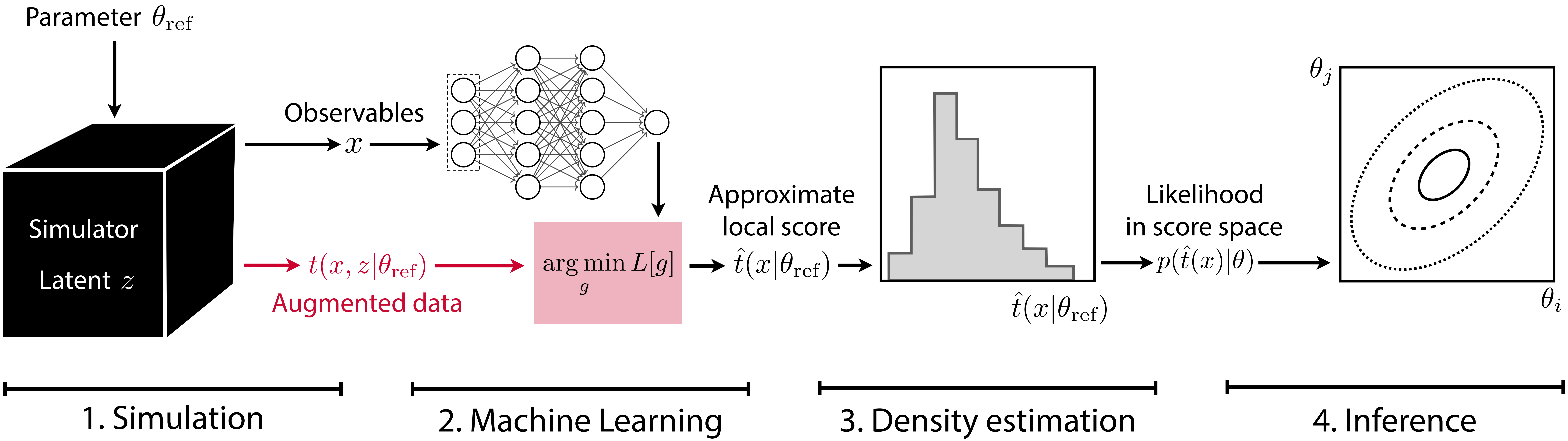
SALLY (Score approximates likelihood locally)



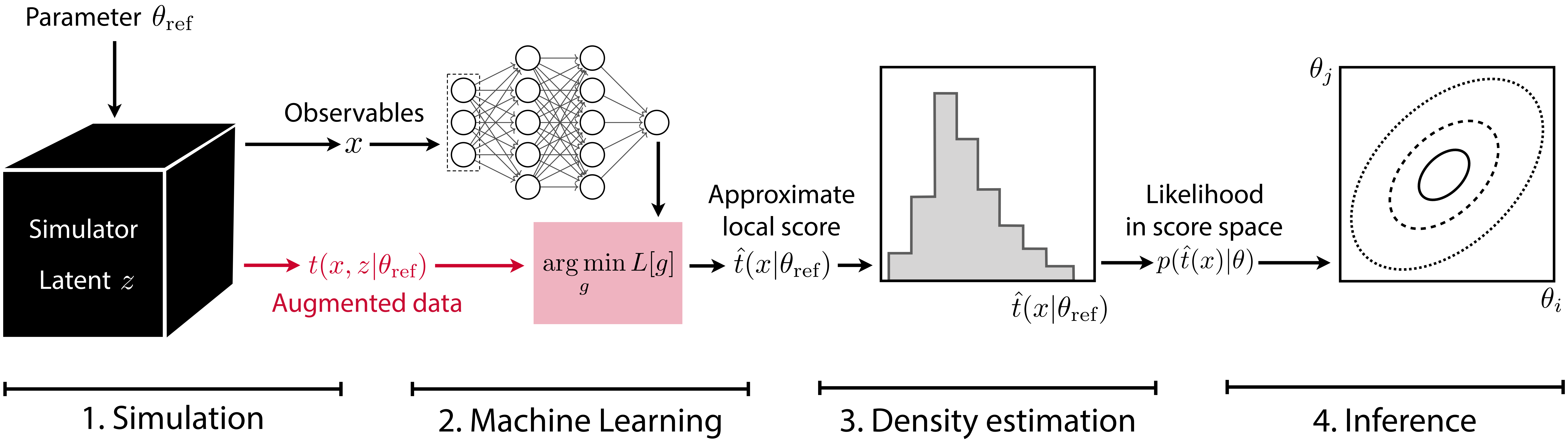
SALLY (Score approximates likelihood locally)



SALLY (Score approximates likelihood locally)



SALLY (Score approximates likelihood locally)

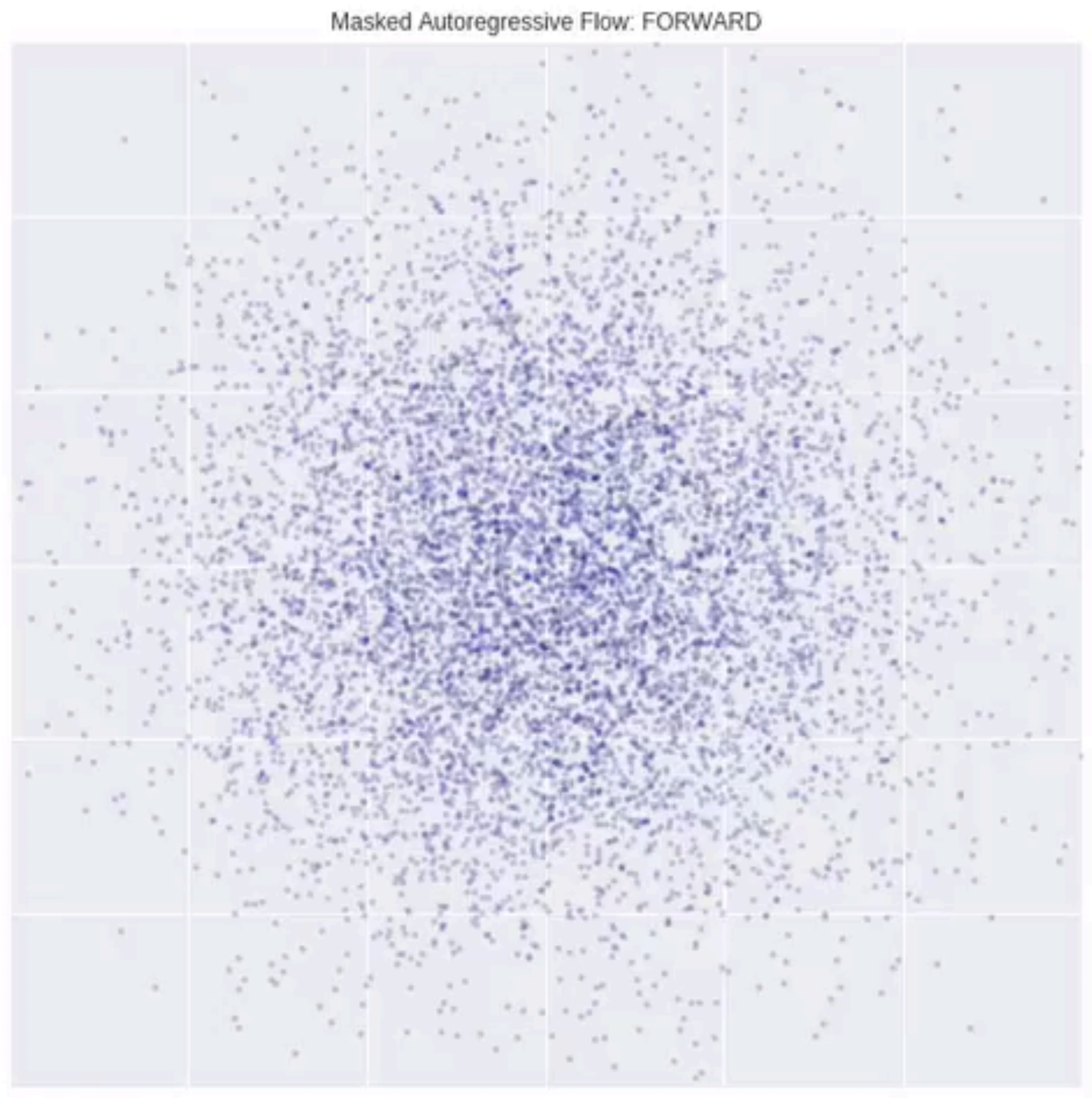


"The machine learning version of Optimal Observables":

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

There's more…

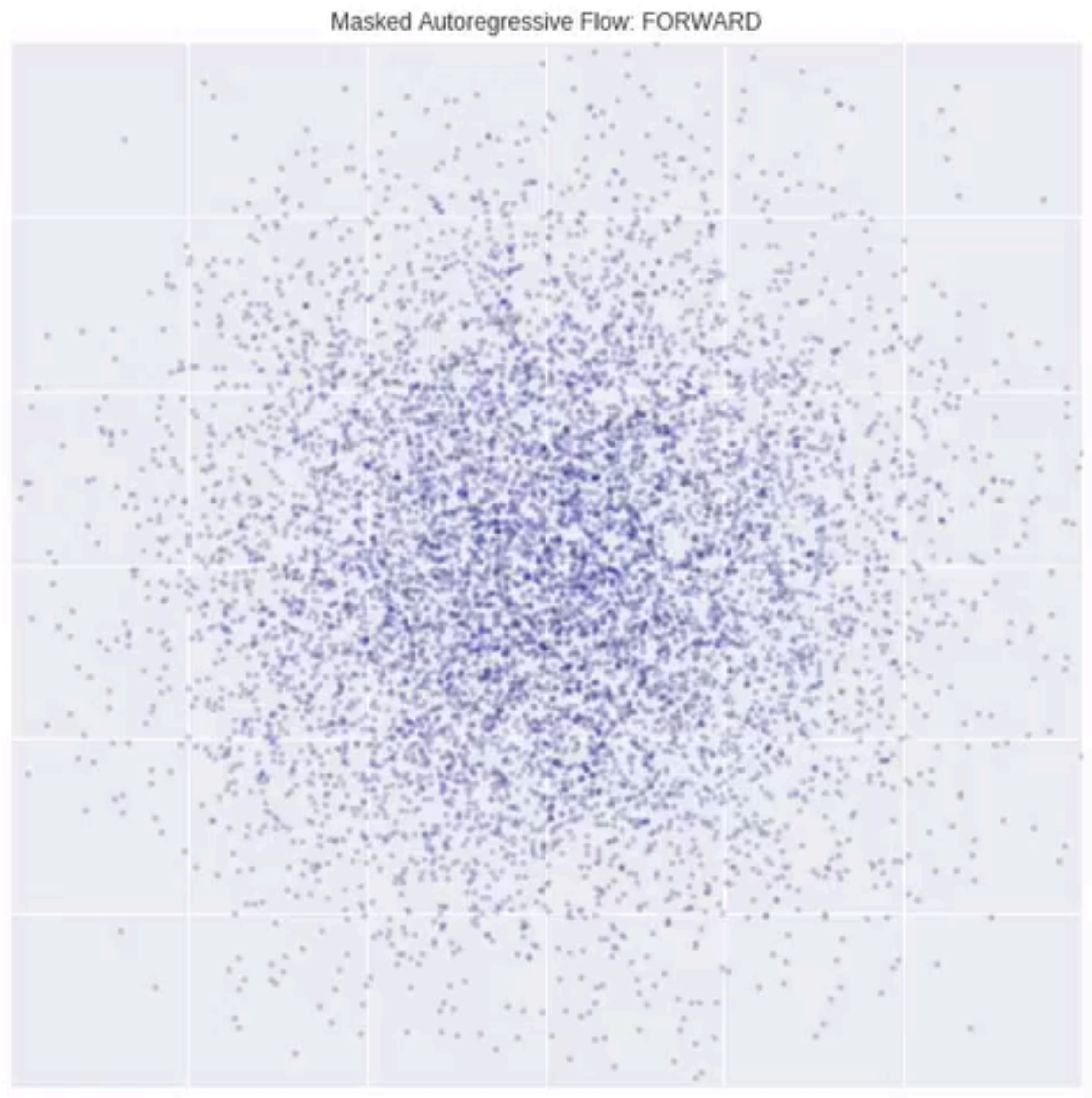
- Other loss functions also work
  - ALICE: based on cross entropy
  - SCANDAL: combine with neural density estimators, e.g. Masked Autoregressive Flows
    [G. Papamakarios, T. Pavlakou, I. Murray 1705.07057]



[Alex Mordvintsev]

There's more…

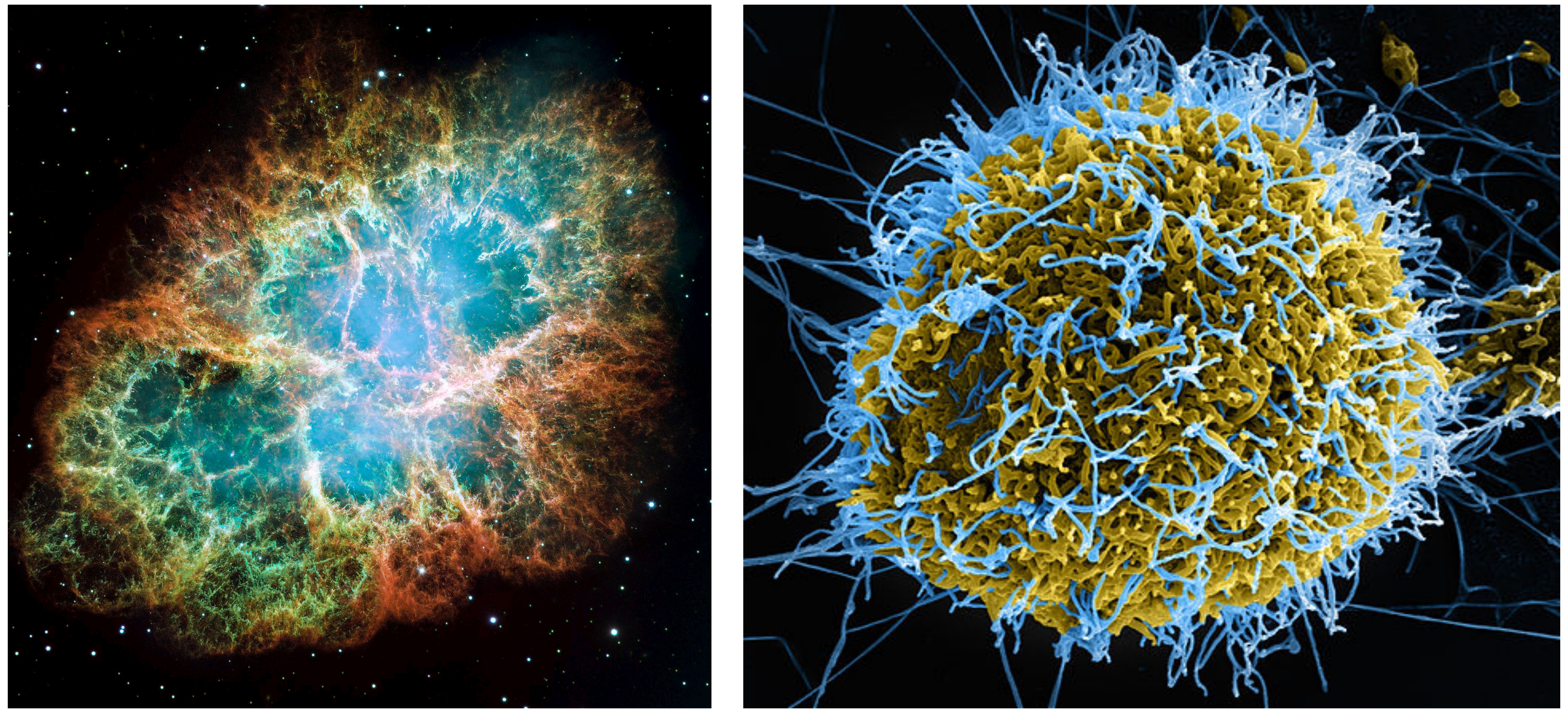
- Other loss functions also work
  - ALICE: based on cross entropy
  - SCANDAL: combine with neural density estimators, e.g. Masked Autoregressive Flows
    [G. Papamakarios, T. Pavlakou, I. Murray 1705.07057]



[Alex Mordvintsev]

- What if we don't fully trust the simulator?
  - Nuisance parameters to model systematic uncertainties
  - Learn robustness with adversarial training
    [G. Louppe, M. Kagan, K. Cranmer 1611.01046]

- More general than particle physics
  - Currently being adapted to cosmology, epidemiology



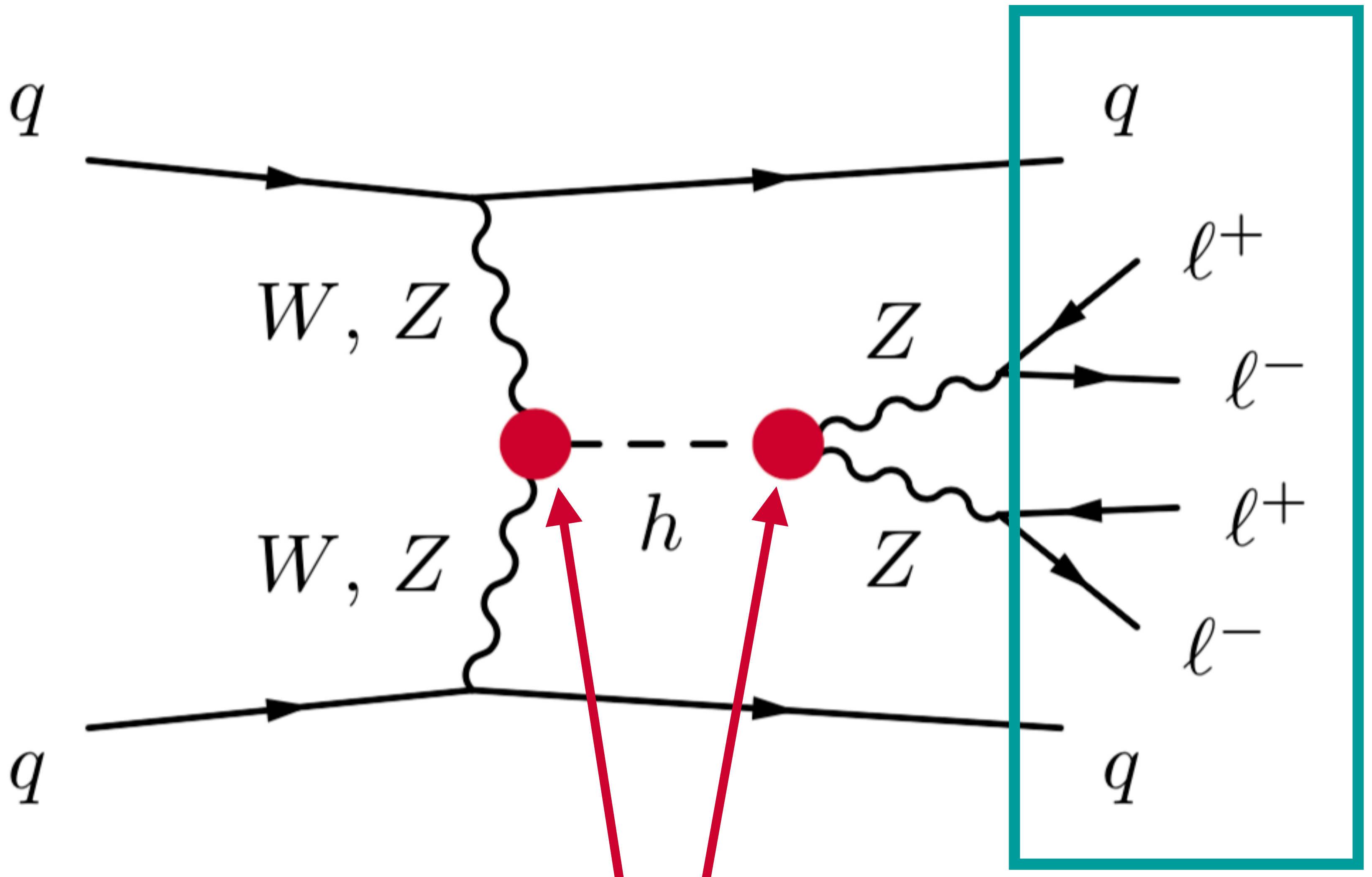
[NASA, NIAID]

EFT example

Proof of concept

Higgs production in weak boson fusion:

at least 16-dimensional
observable space

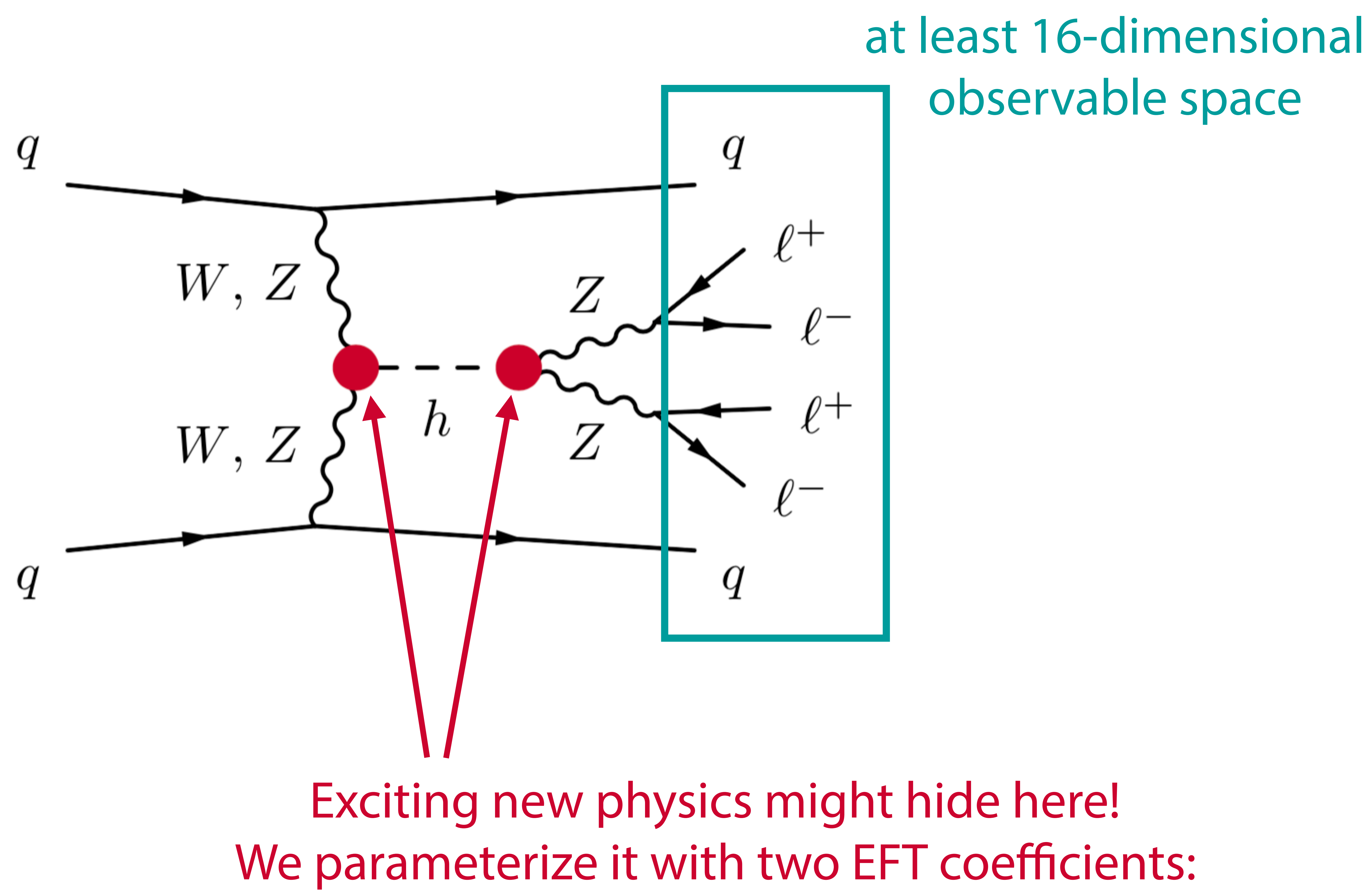


Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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

Proof of concept

Higgs production in weak boson fusion:

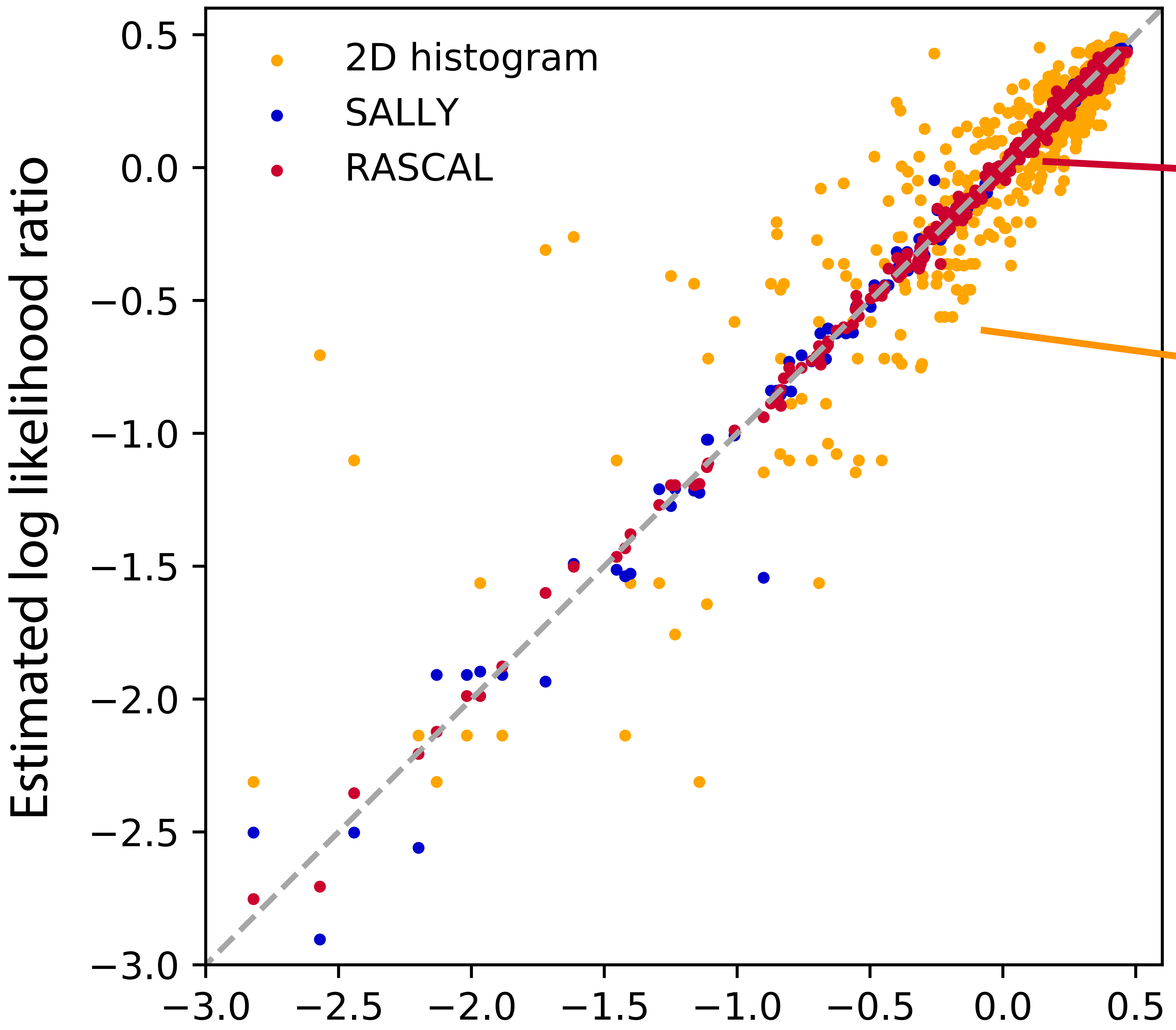


$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \boxed{\frac{f_W}{\Lambda^2}} \underbrace{\frac{ig}{2} \, (D^\mu \phi)^\dagger \, \sigma^a \, D^\nu \phi \, W^a_{\mu\nu}}_{\mathcal{O}_W} - \boxed{\frac{f_{WW}}{\Lambda^2}} \underbrace{\frac{g^2}{4} \, (\phi^\dagger \phi) \, W^a_{\mu\nu} \, W^{\mu\nu\, a}}_{\mathcal{O}_{WW}}$$

- Goal: constrain the two EFT parameters
  - new inference methods
  - baseline: 2d histogram analysis of jet momenta & angular correlations
- Two scenarios:
  - Simplified setup in which we can compare to true likelihood
  - "Realistic" simulation with approximate detector effects
- Simulation: MadGraph + MadMax
  [J. Alwall et al. 1405.0301; K. Cranmer, T. Plehn hep-ph/0605268; T. Plehn, P. Schichtel, D. Wiegand 1311.2591]

More precise likelihood ratio estimates with less training data

Each point = 1 event

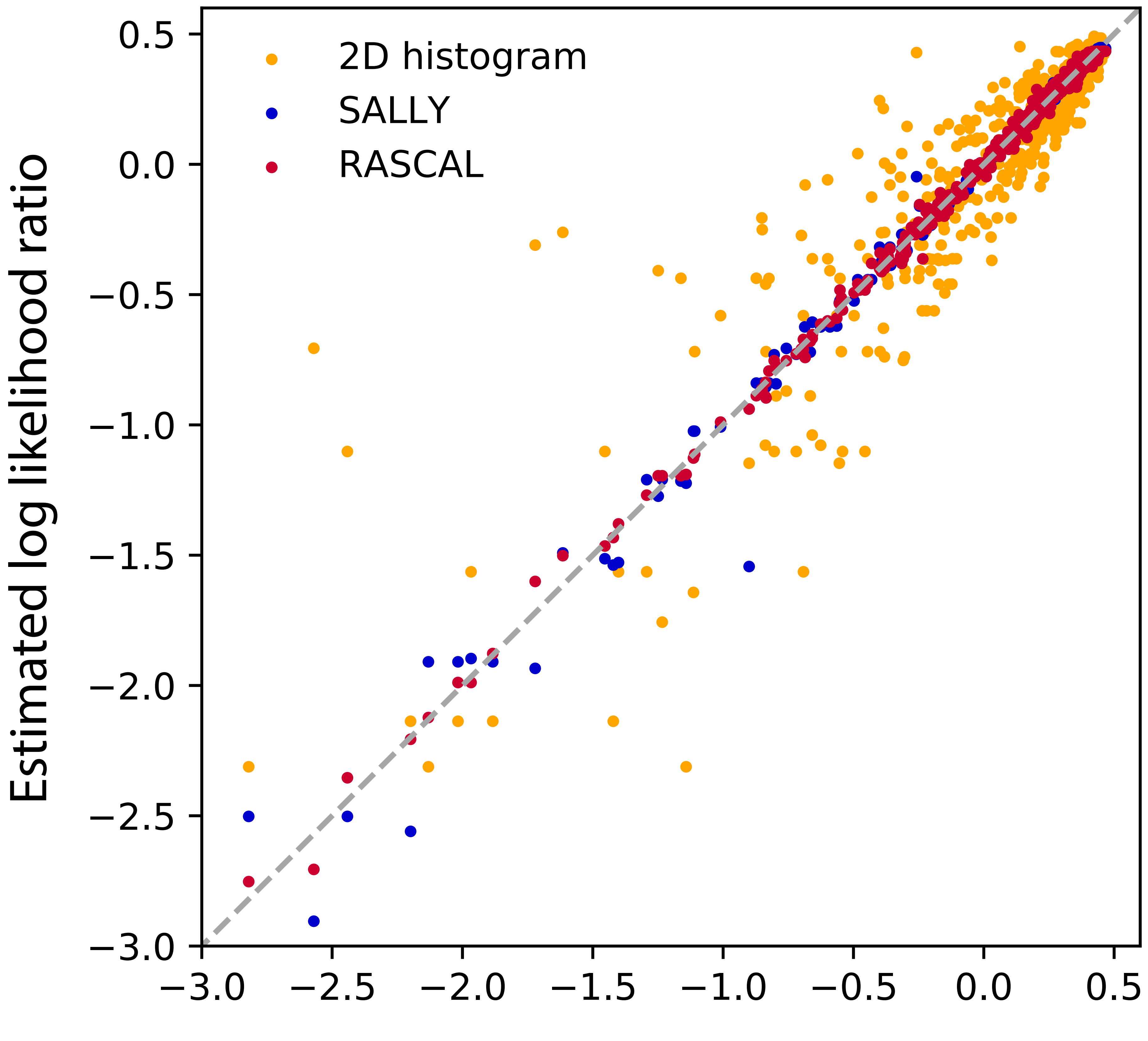


True log likelihood ratio
(usually don't have this)

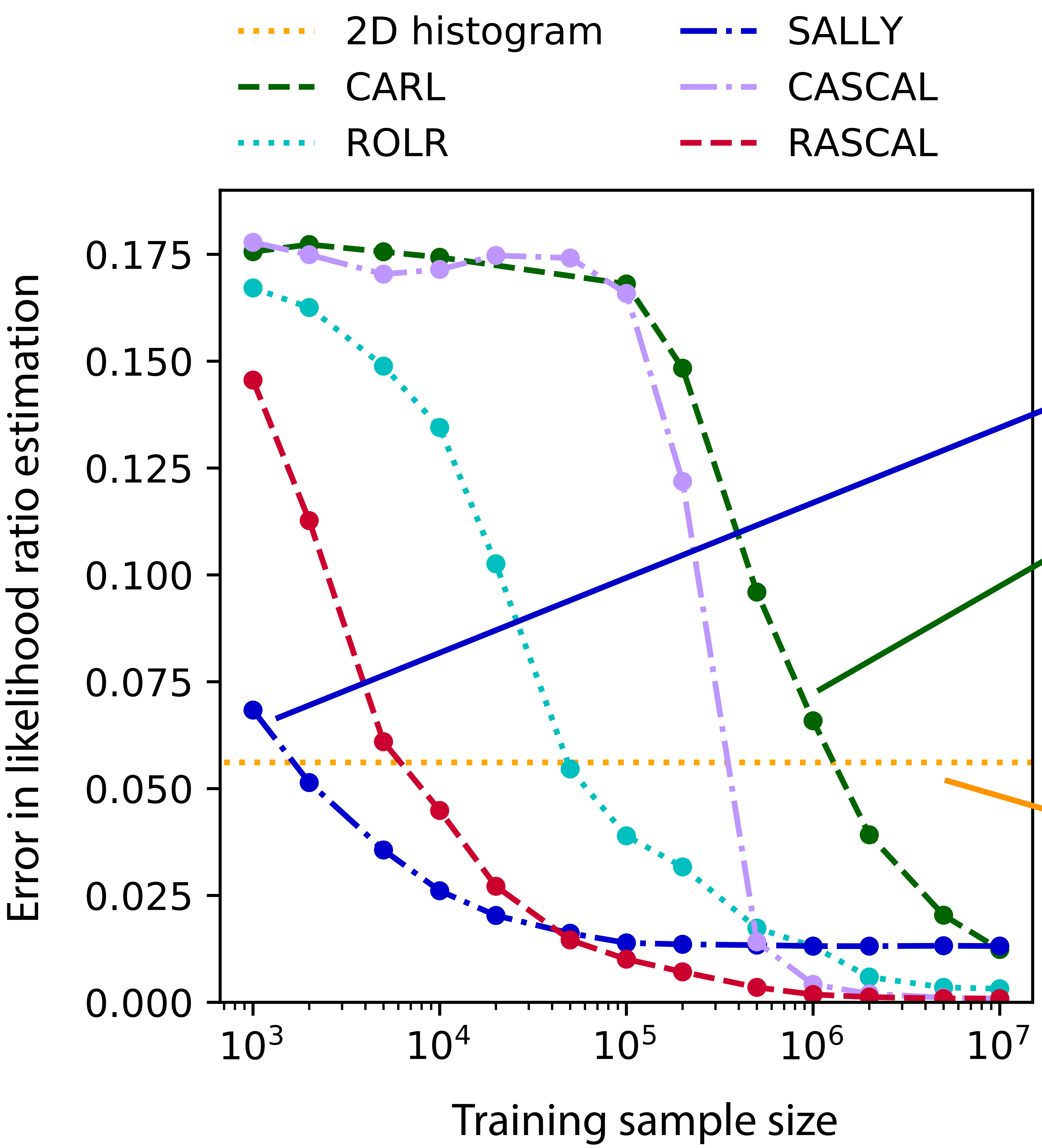
RASCAL
gives more precise likelihood ratio estimates than the
2D histogram

More precise likelihood ratio estimates with less training data

Each point = 1 event



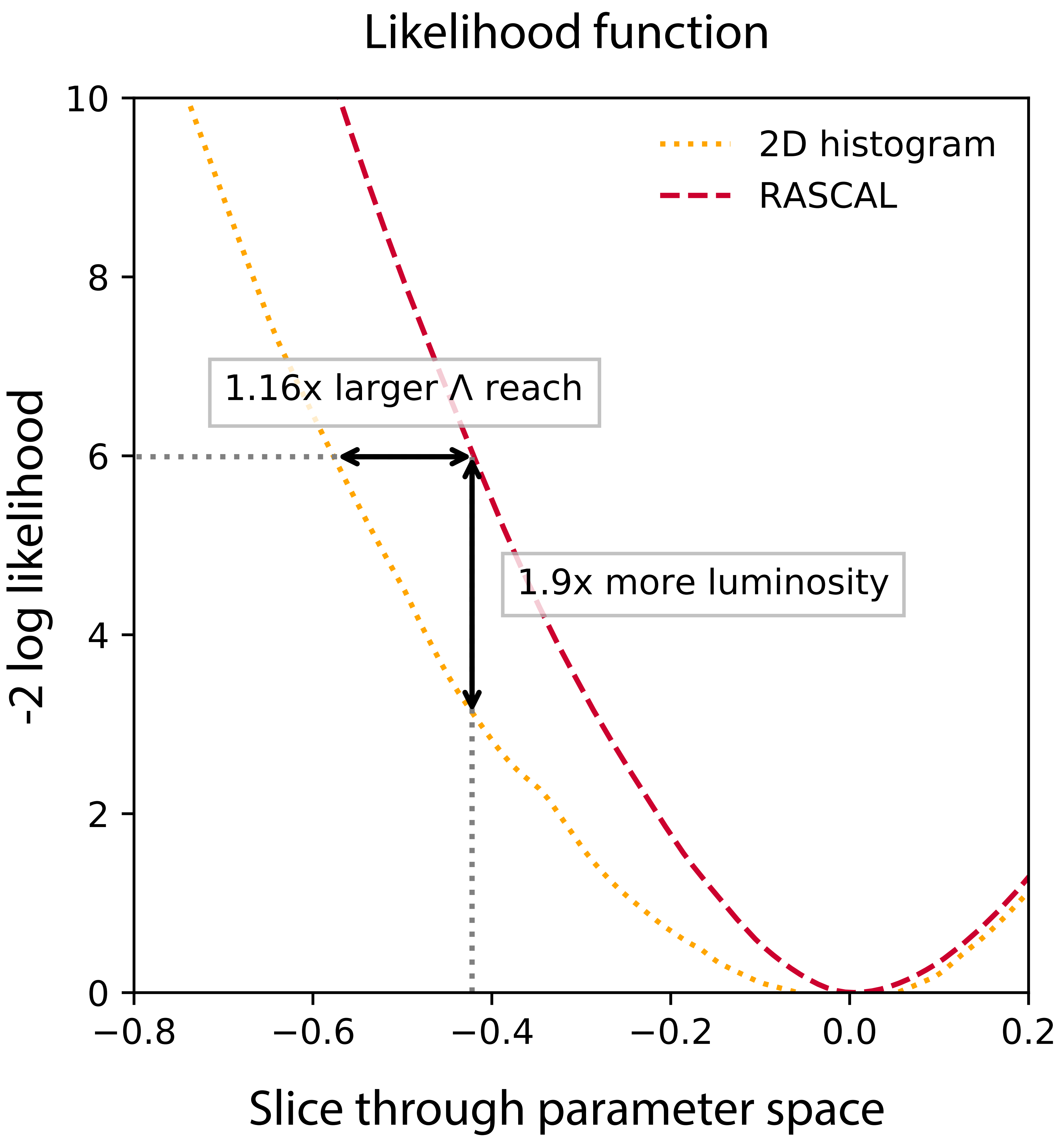
True log likelihood ratio
(usually don't have this)



SALLY and RASCAL require less data than generic ML method [CARL, K. Cranmer, G. Louppe, J. Pavez 1506.02169]

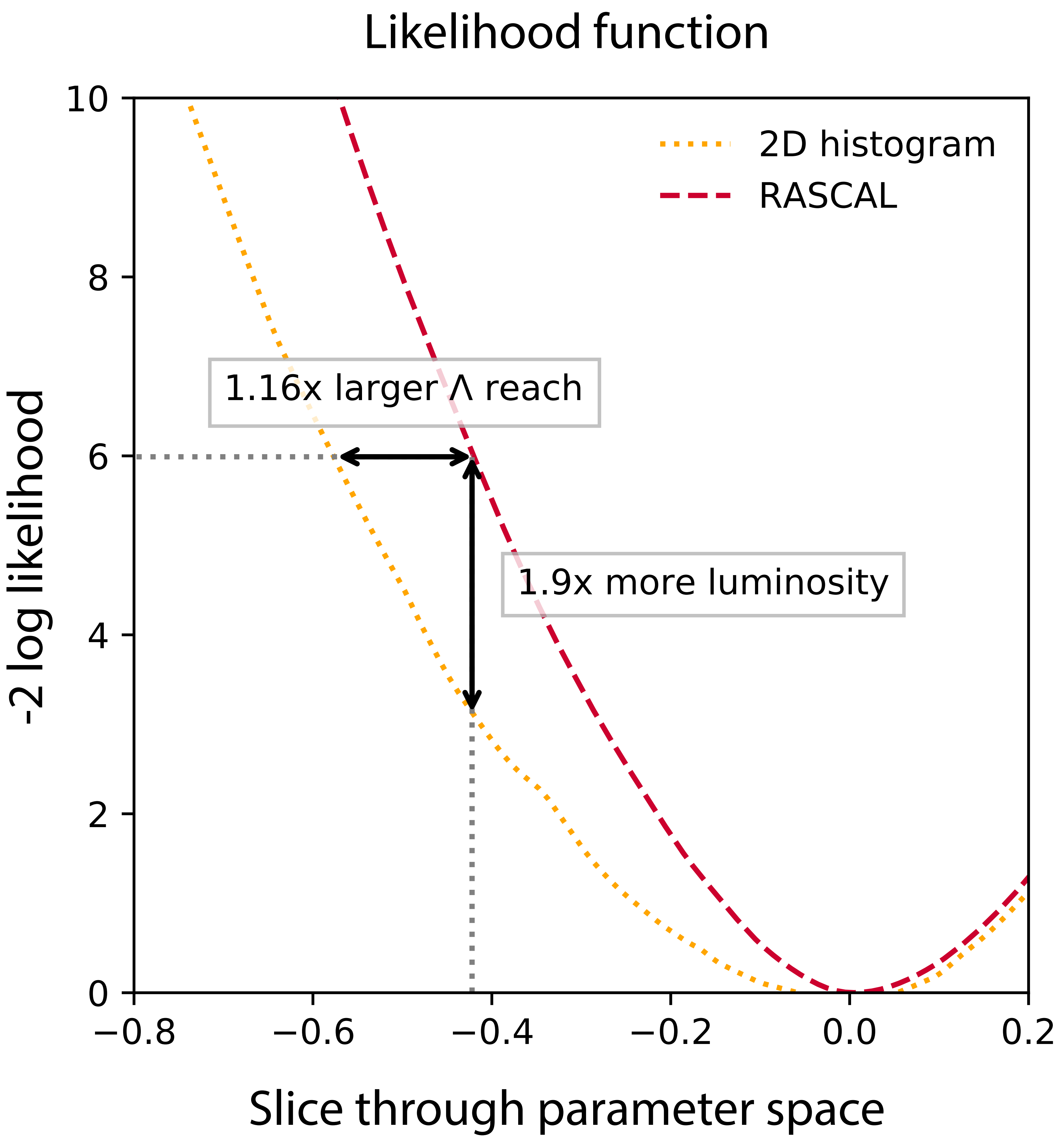
Baseline: 2D histogram (based on large training sample)

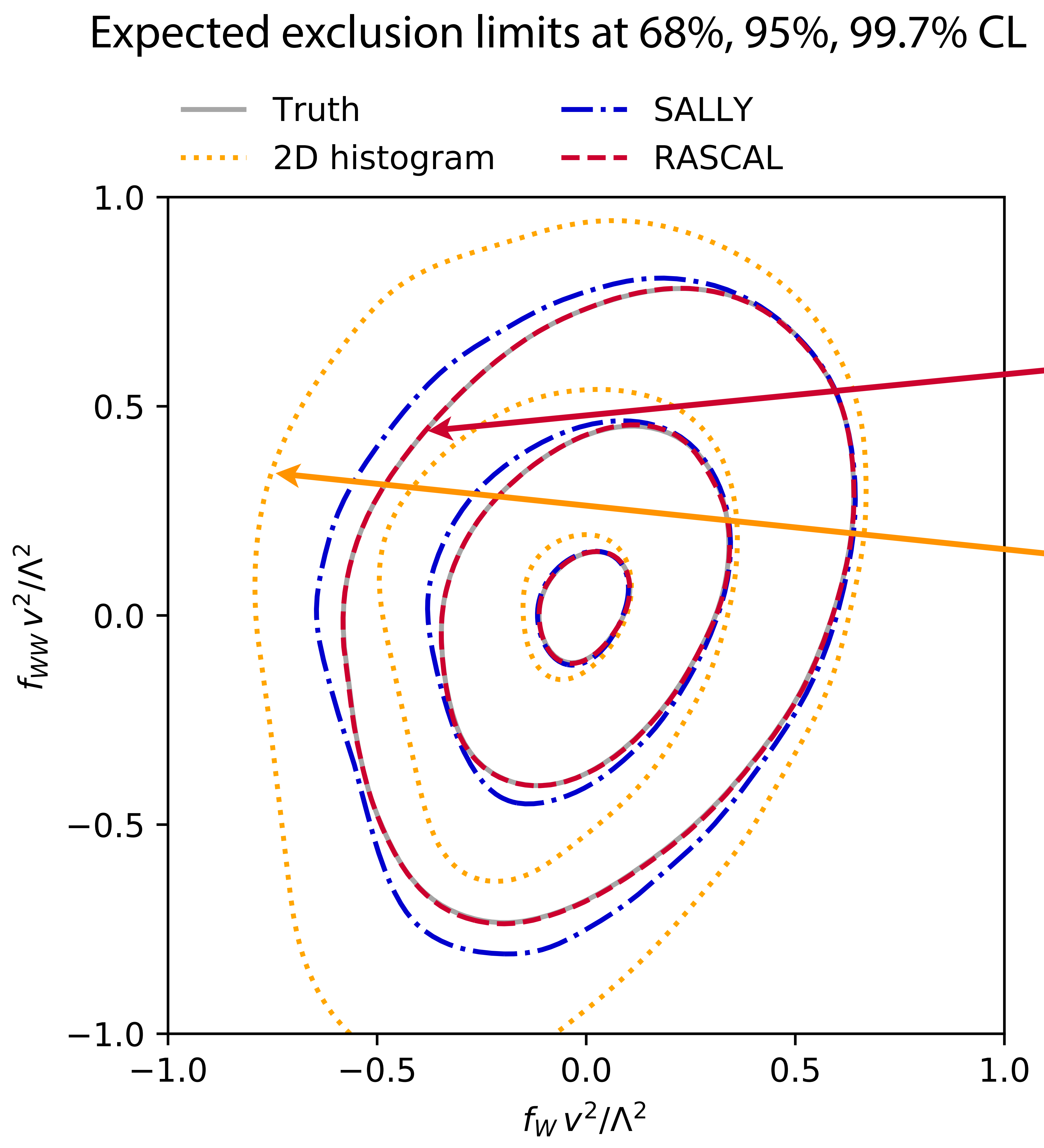
Better sensitivity to new physics



Results are based on 36 observed events, assuming SM

Better sensitivity to new physics

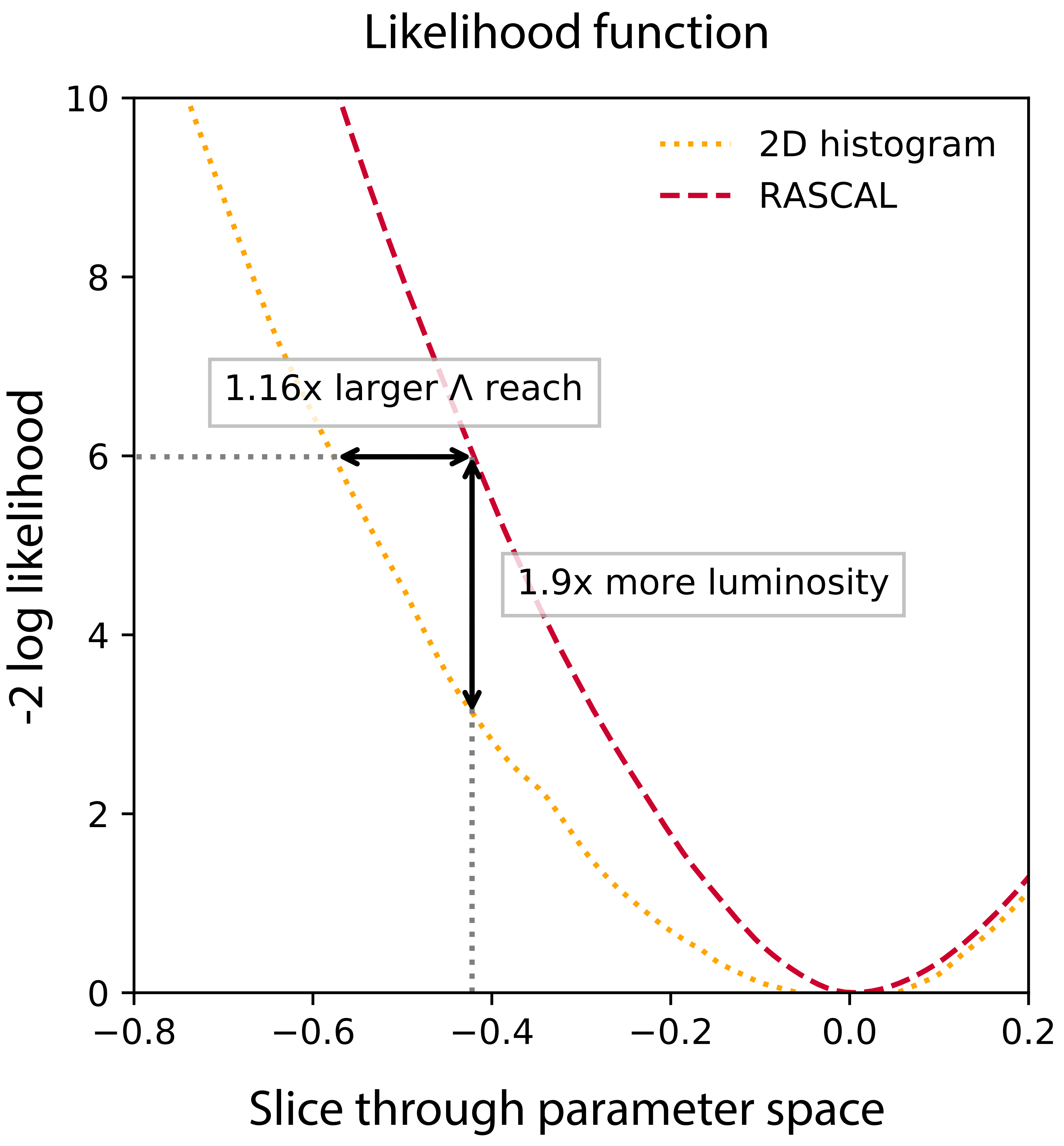




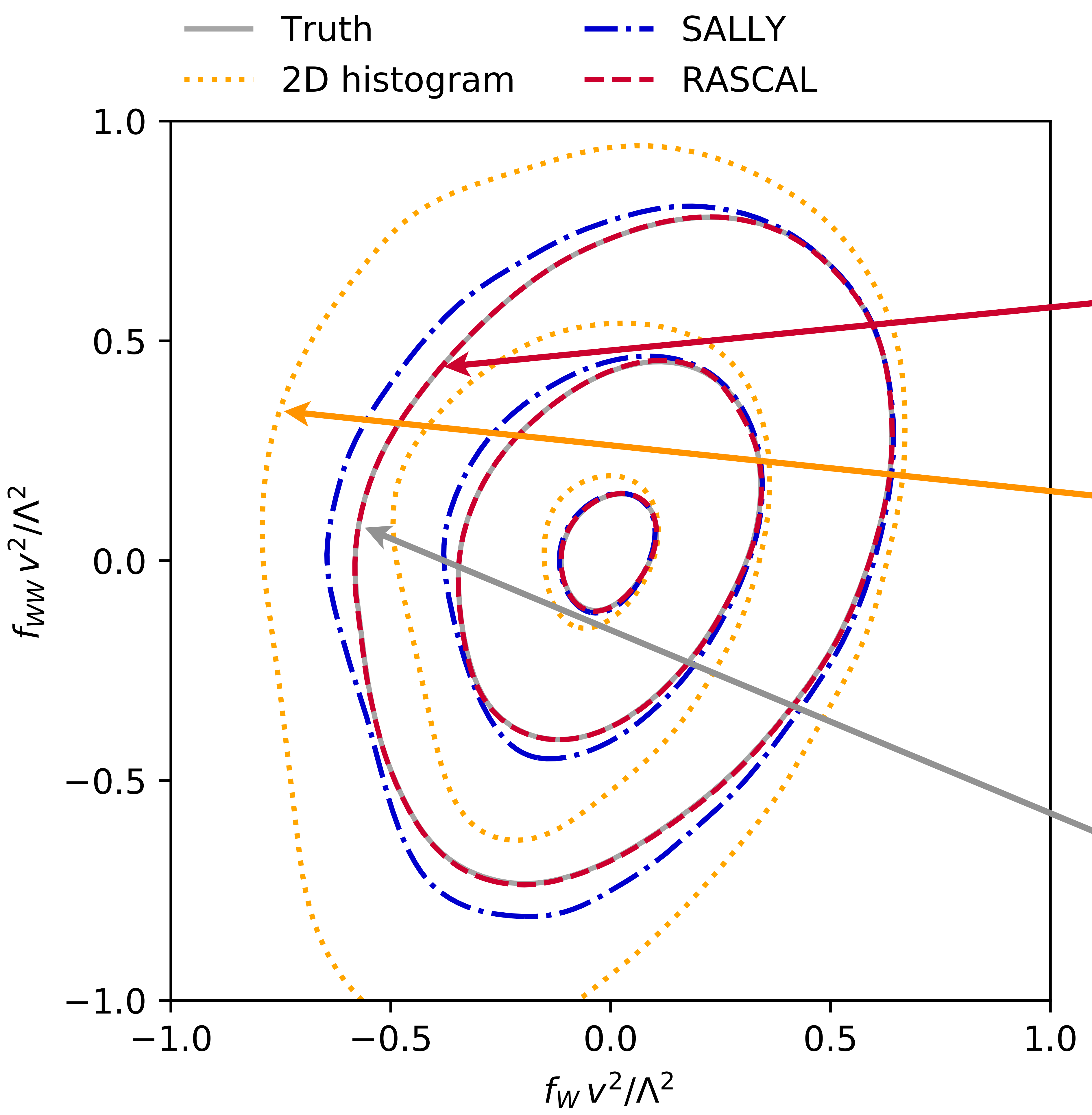
RASCAL and SALLY enables stronger limits than 2D histogram

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)

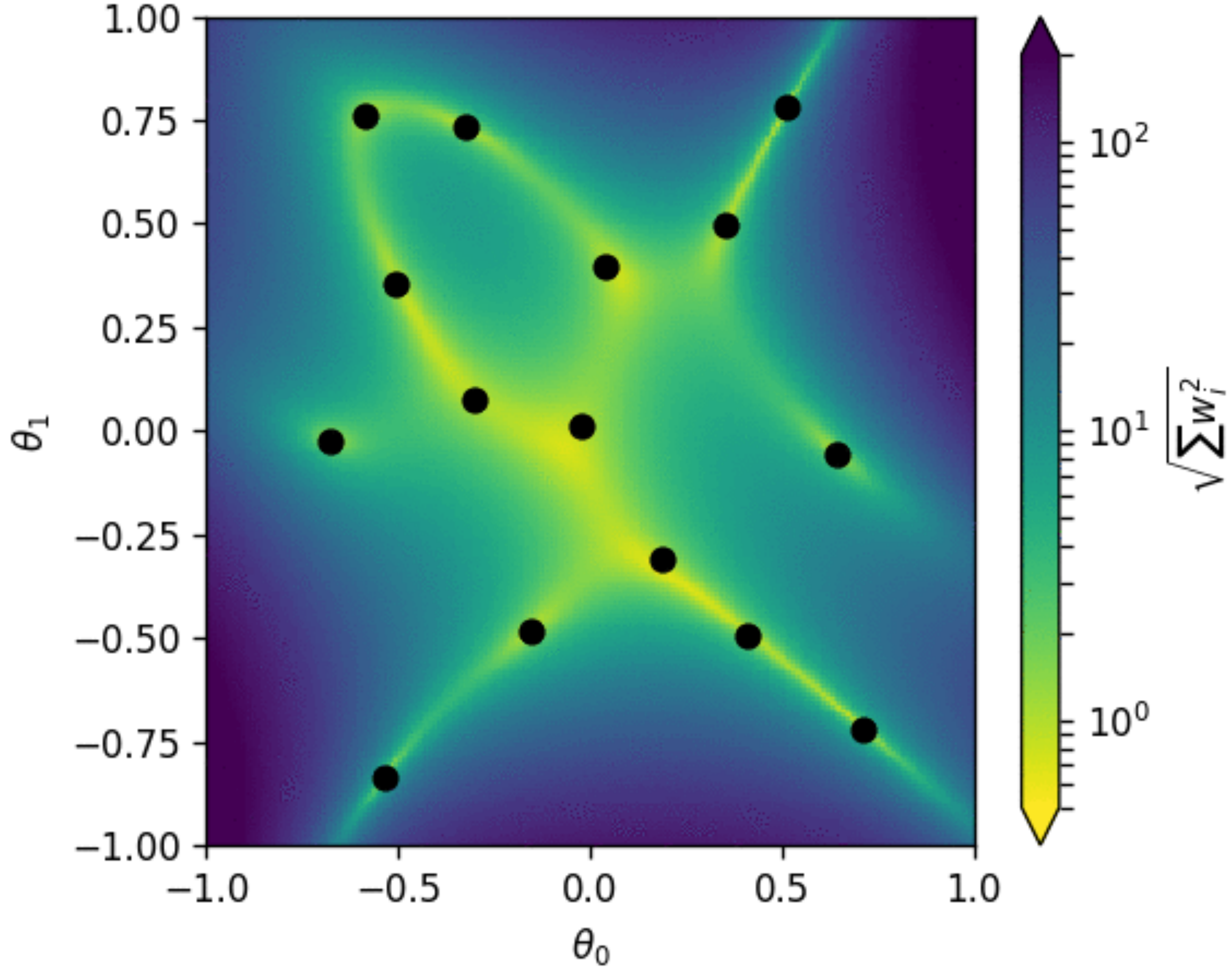
Results are based on 36 observed events, assuming SM

The MadMiner package

Can I use any of this?

Yes! To make that as painless as possible, we're working on the python package MadMiner:

- "Mining gold" from MadGraph + Pythia + detector simulation
- Parameter morphing (for EFT-like situations)
- Likelihood estimation with RASCAL and friends
- Fisher information calculation
- Systematic uncertainties (currently from PDF variation or scale variation)

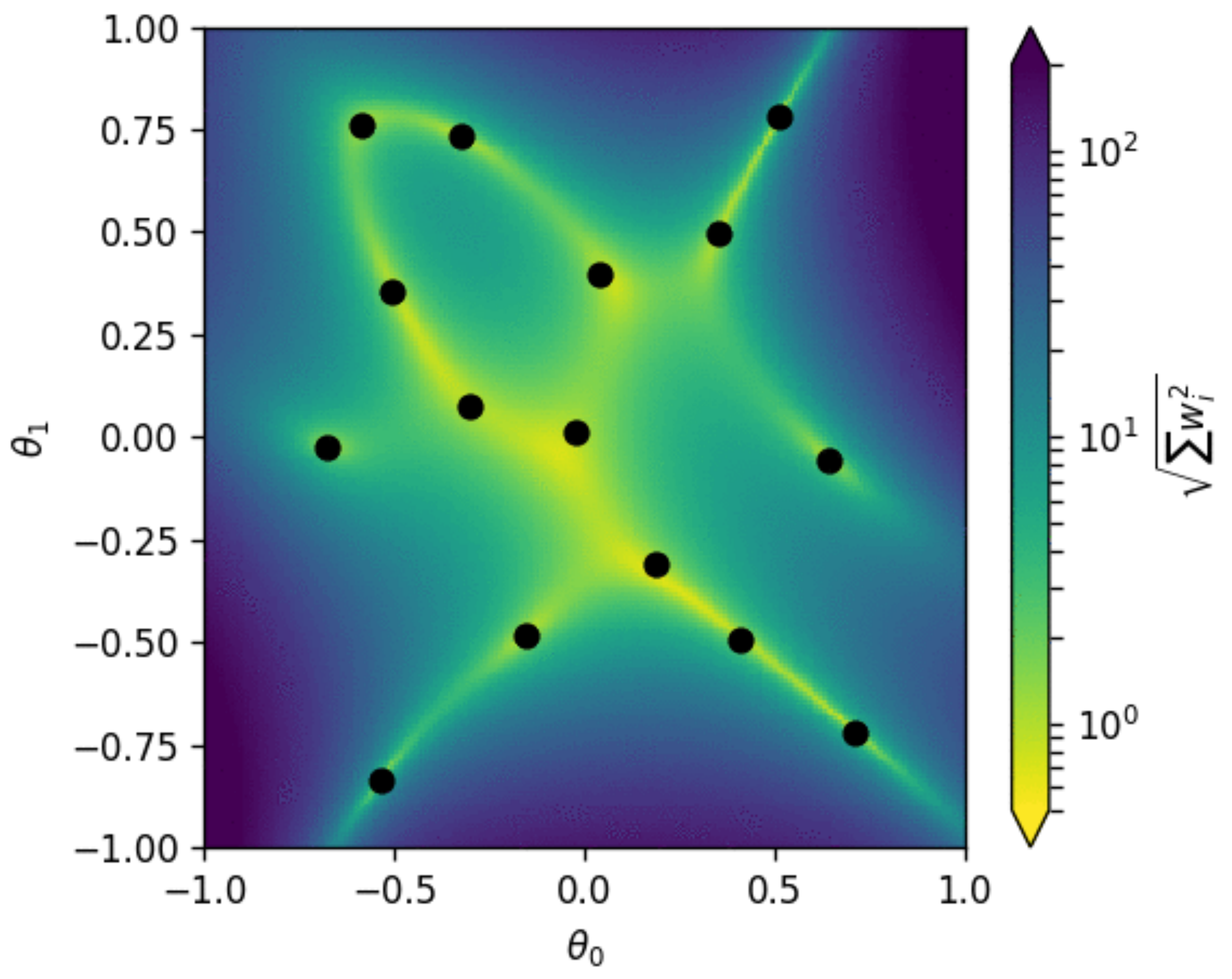


We need users / testers / collaborators!

Can I use any of this?

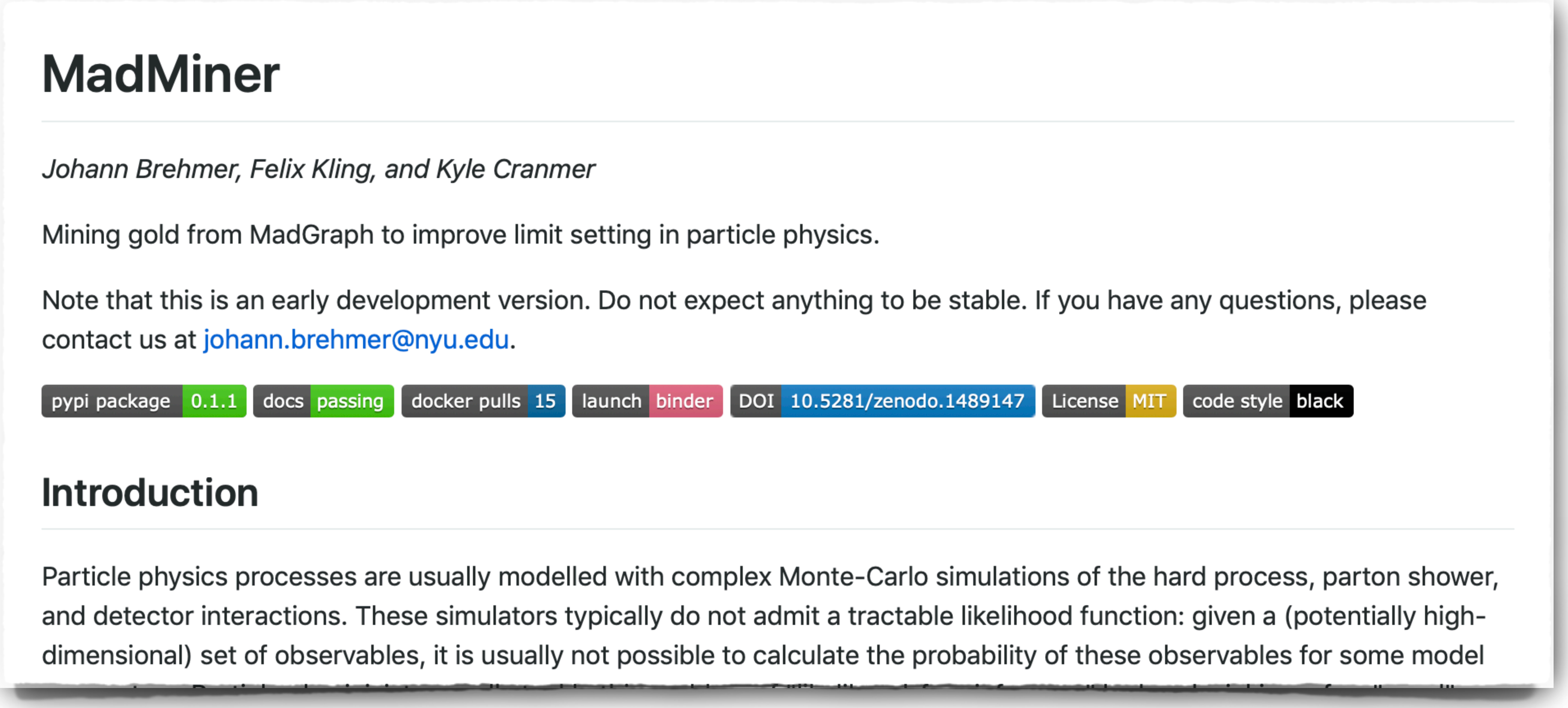
Yes! To make that as painless as possible, we're working on the python package MadMiner:

- "Mining gold" from MadGraph + Pythia + detector simulation
- Parameter morphing (for EFT-like situations)
- Likelihood estimation with RASCAL and friends
- Fisher information calculation
- Systematic uncertainties (currently from PDF variation or scale variation)



We need users / testers / collaborators!

MadMiner resources



MadMiner

Johann Brehmer, Felix Kling, 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.

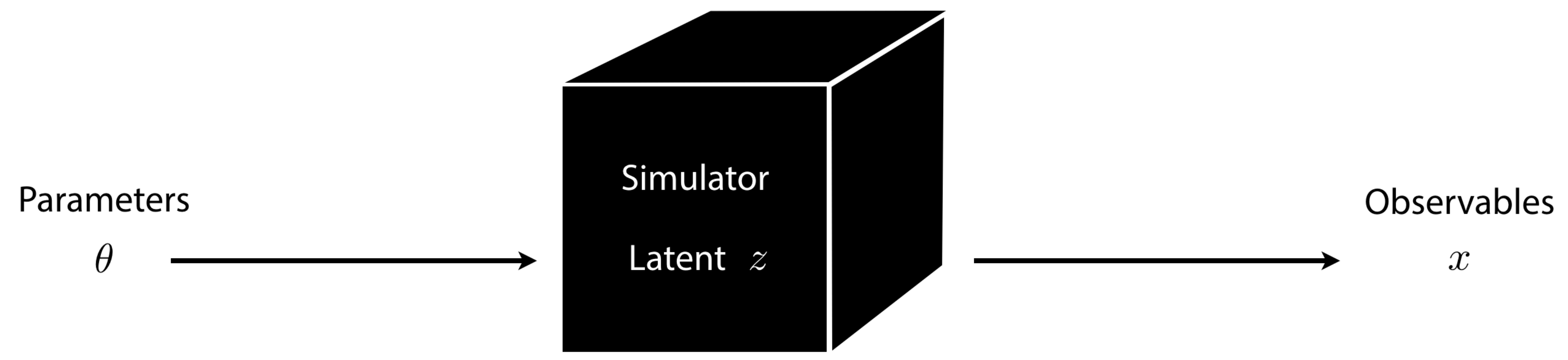
pypi package 0.1.1 | docs passing | docker pulls 15 | launch binder | DOI 10.5281/zenodo.1489147 | License MIT | code style black

Introduction

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

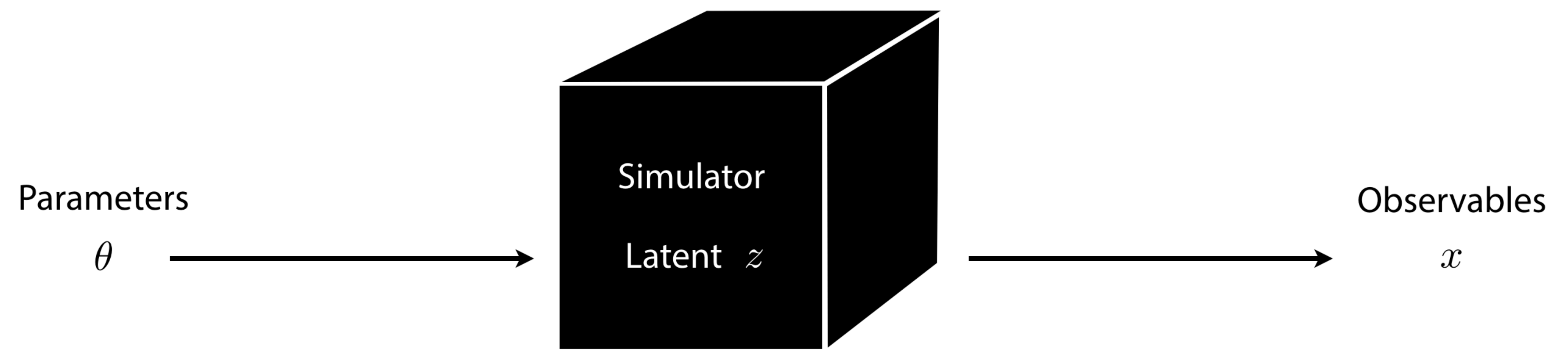
- Current version and tutorials: github.com/johannbrehmer/madminer
- Detailed documentation: madminer.readthedocs.io
- `pip install madminer`

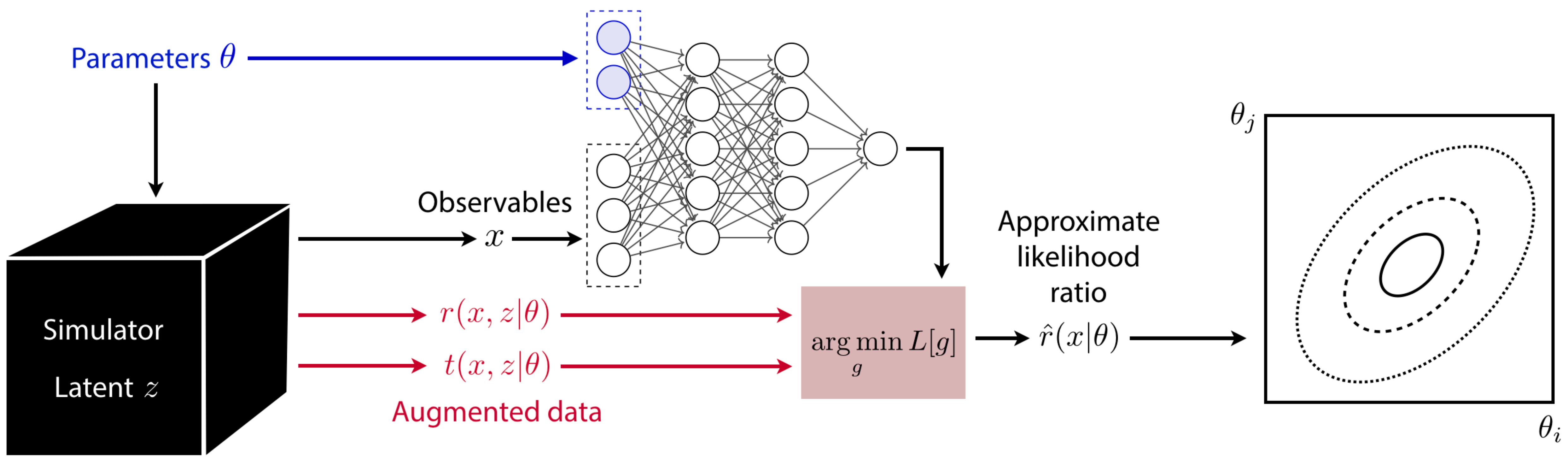
A new approach to simulator-based inference



- Many LHC analyses (and much of modern science) are based on simulations, “likelihood-free”

A new approach to simulator-based inference

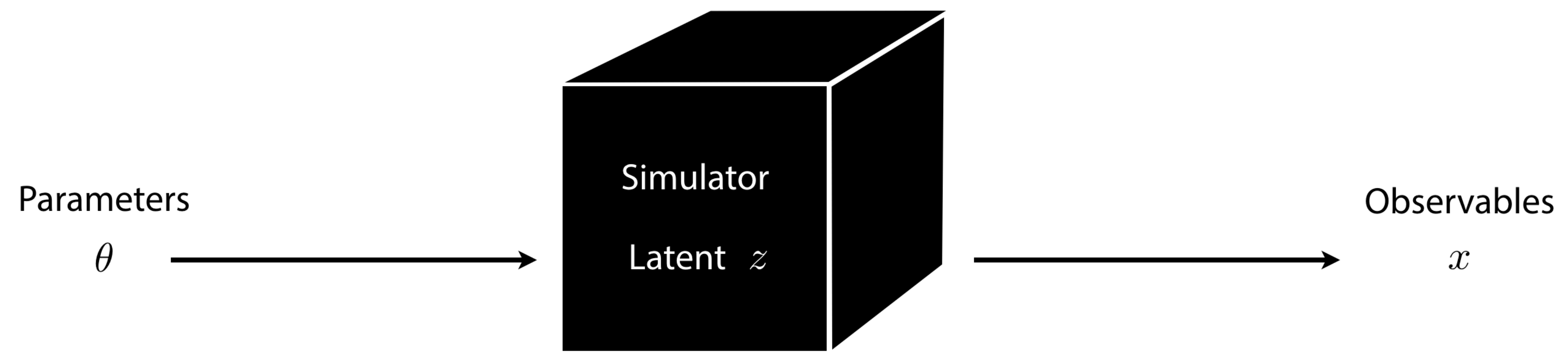


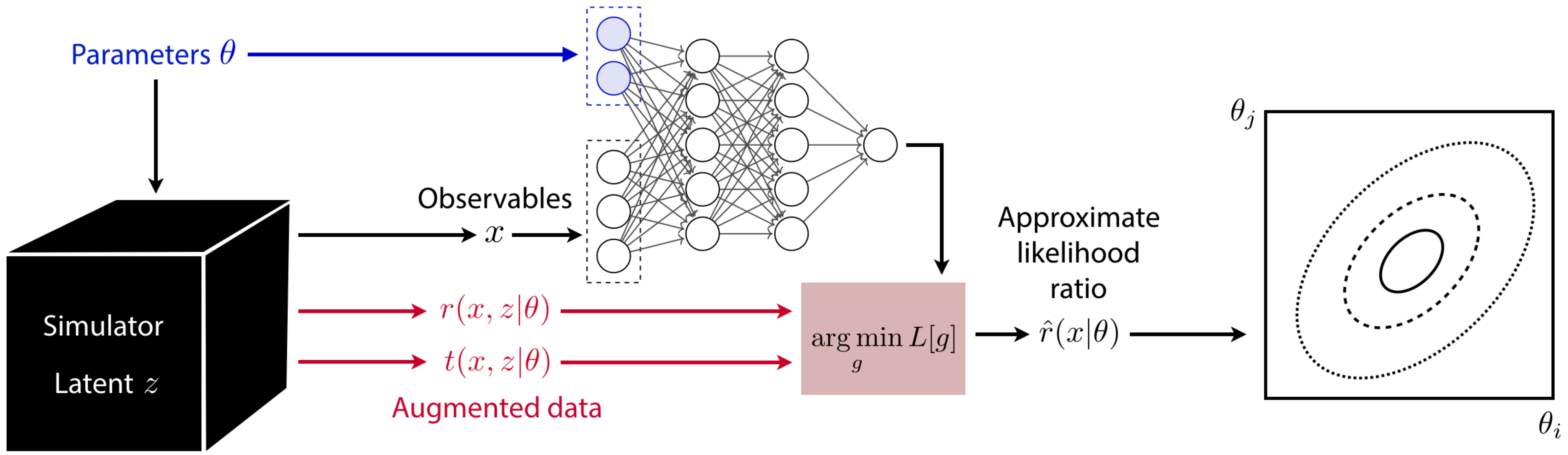


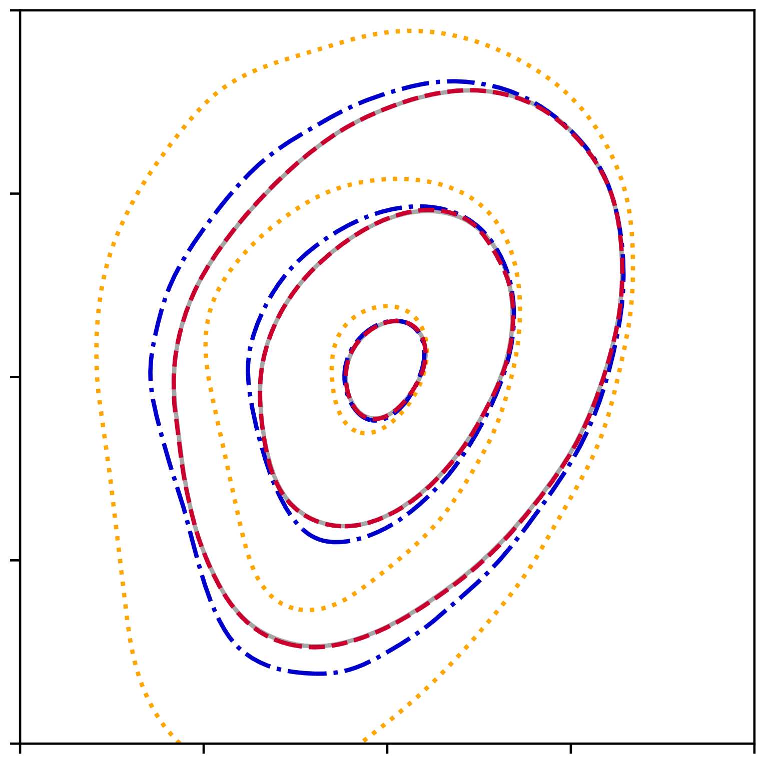
- Many LHC analyses (and much of modern science) are based on simulations, “likelihood-free”

- New multivariate inference techniques: Leverage information in matrix elements + power of machine learning

A new approach to simulator-based inference







- Many LHC analyses (and much of modern science) are based on simulations, "likelihood-free"
- New multivariate inference techniques: Leverage information in matrix elements + power of machine learning
- First application to LHC physics: Stronger EFT constraints with less simulations
- Automatization with new tool MadMiner

References

Kyle Cranmer

Gilles Louppe

Juan Pavez

Markus Stoye

Felix Kling

JB, KC, GL, JP: Constraining Effective Field Theories with Machine Learning [1805.00013]

JB, KC, GL, JP: A Guide to Constraining Effective Field Theories with Machine Learning [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, FK, KC: MadMiner [doi: 10.5281/zenodo.1489147]

Thanks to Kyle and Gilles for inspiring many slides

and to the Moore-Sloan Data Science Environment at NYU for their support!

Bonus material

Variational calculus

$$L[\hat{g}(x)] = \int \mathrm{d}x\, \mathrm{d}z\; p(x, z|\theta)\; |g(x, z) - \hat{g}(x)|^2$$

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

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

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

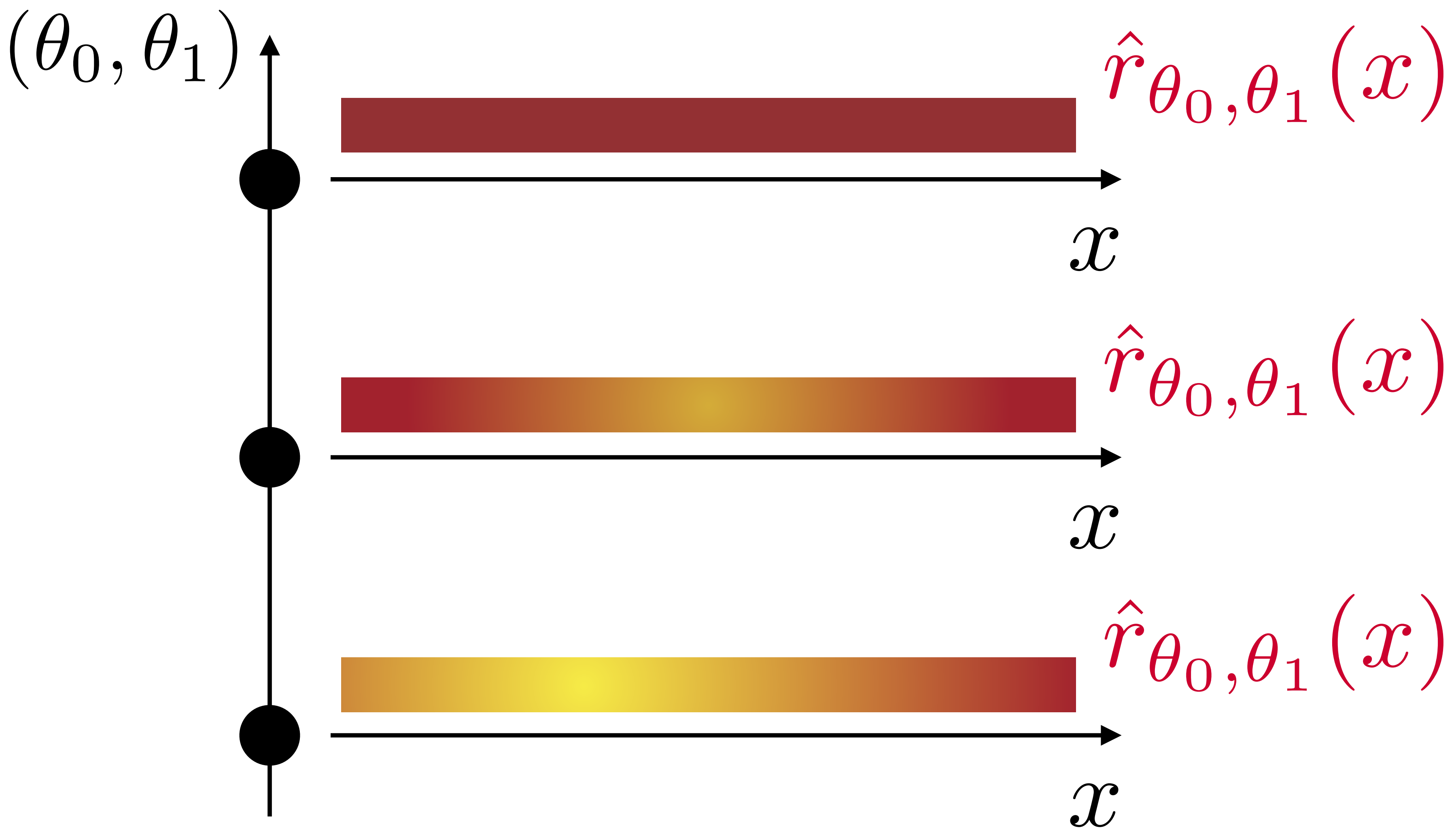
Two types of likelihood ratio estimators

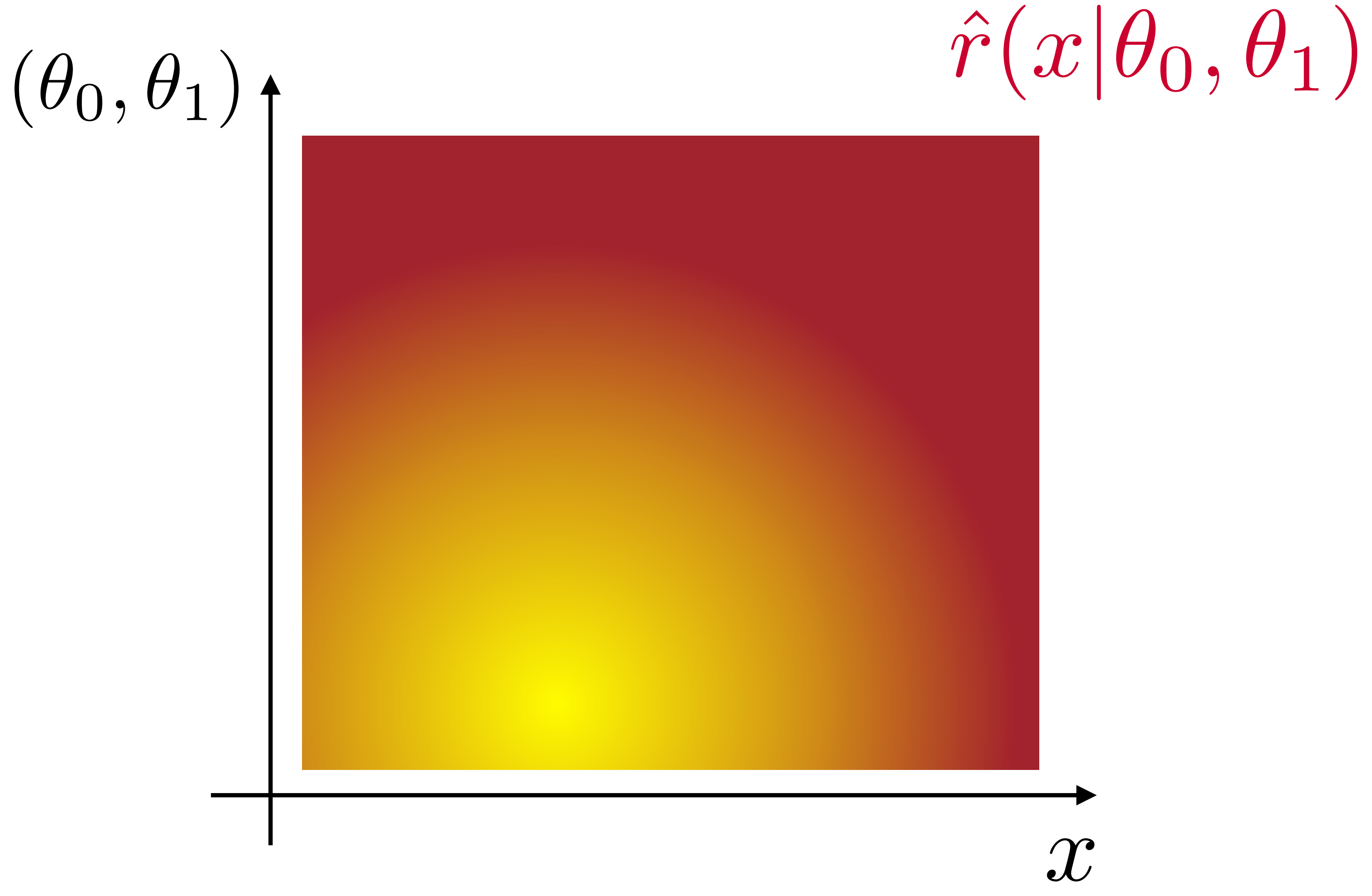
A) Point by point:

- 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

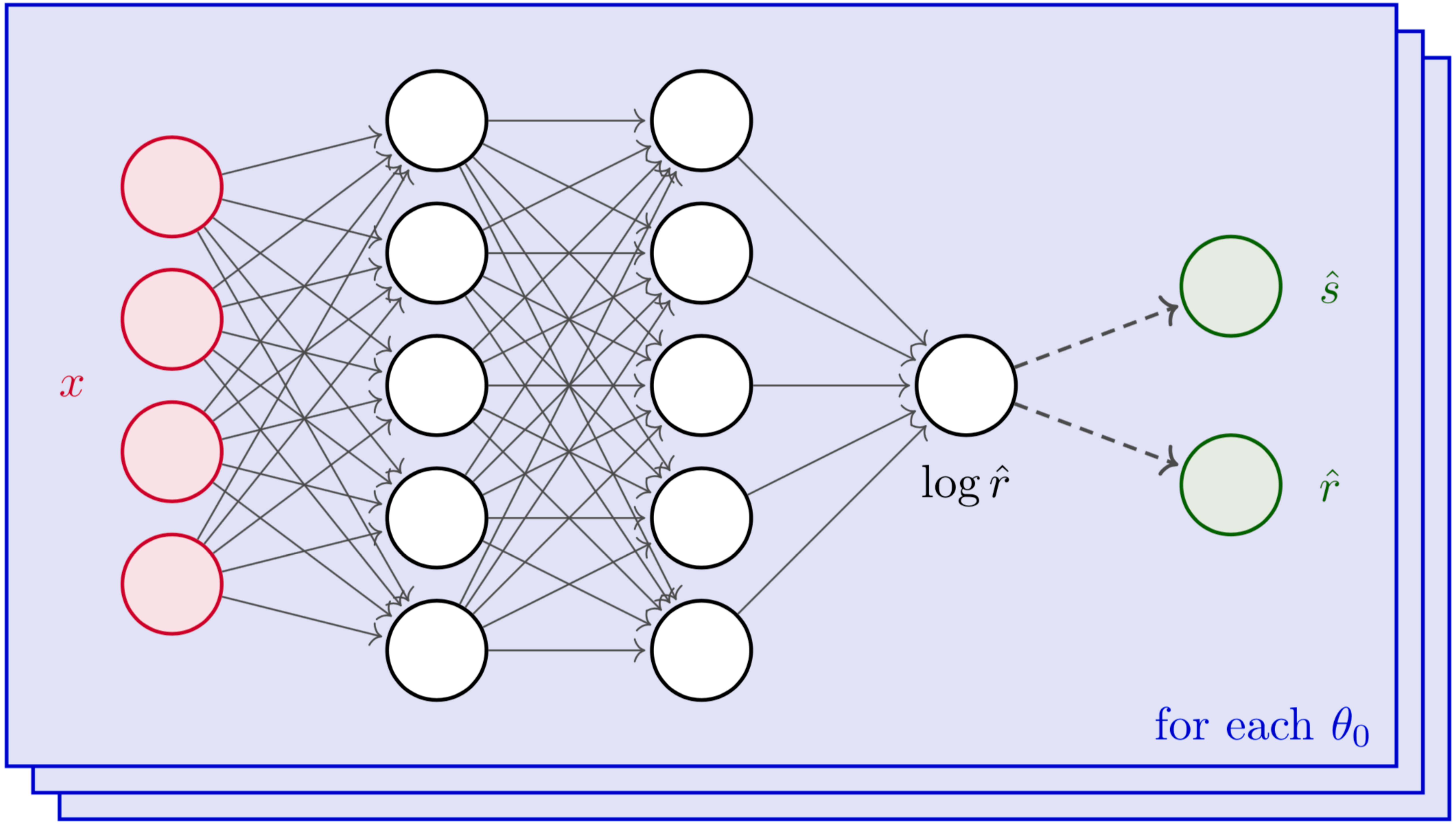
B) Parameterized: [P. Baldi et al. 1506.02169]

- 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

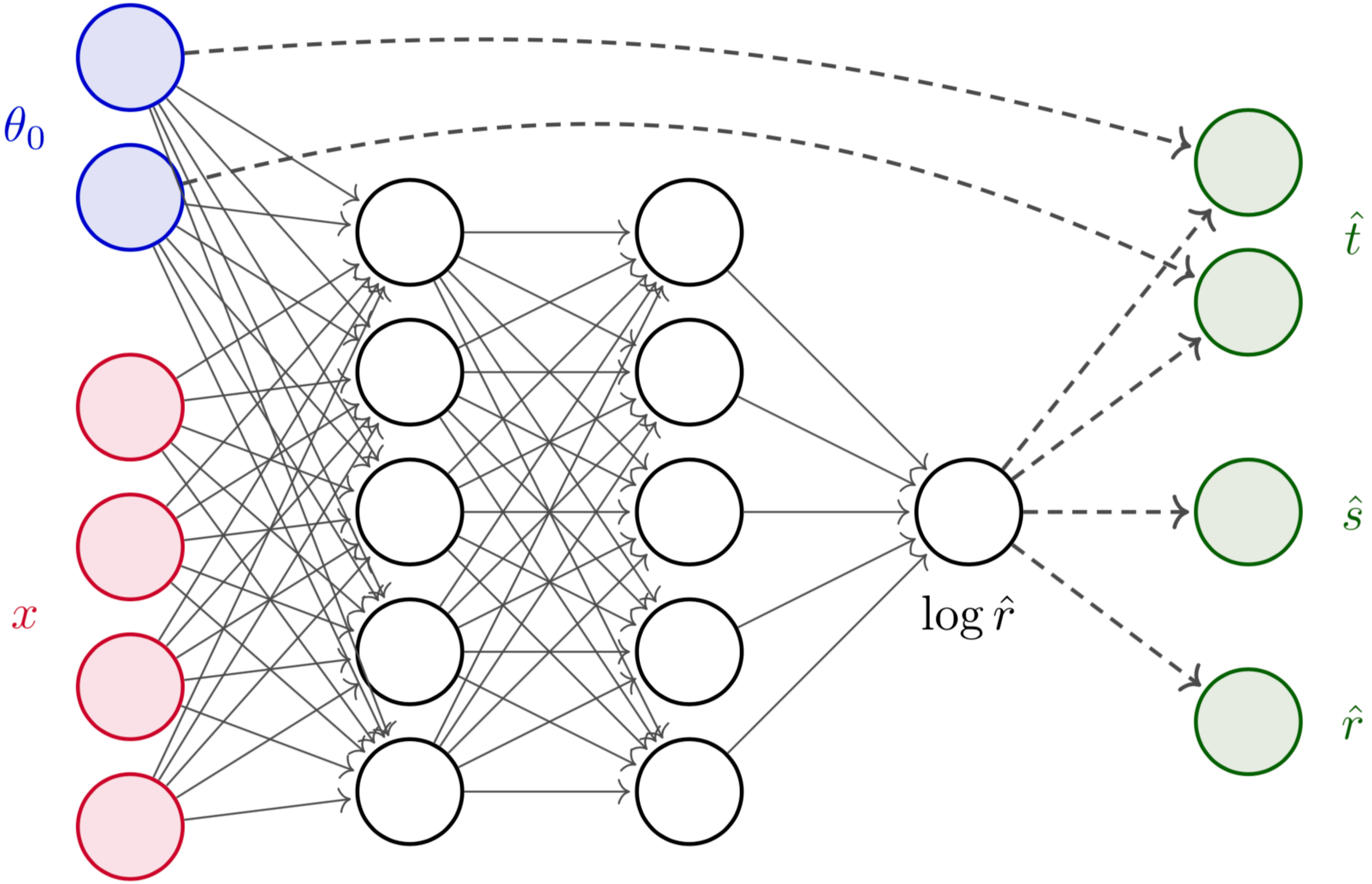




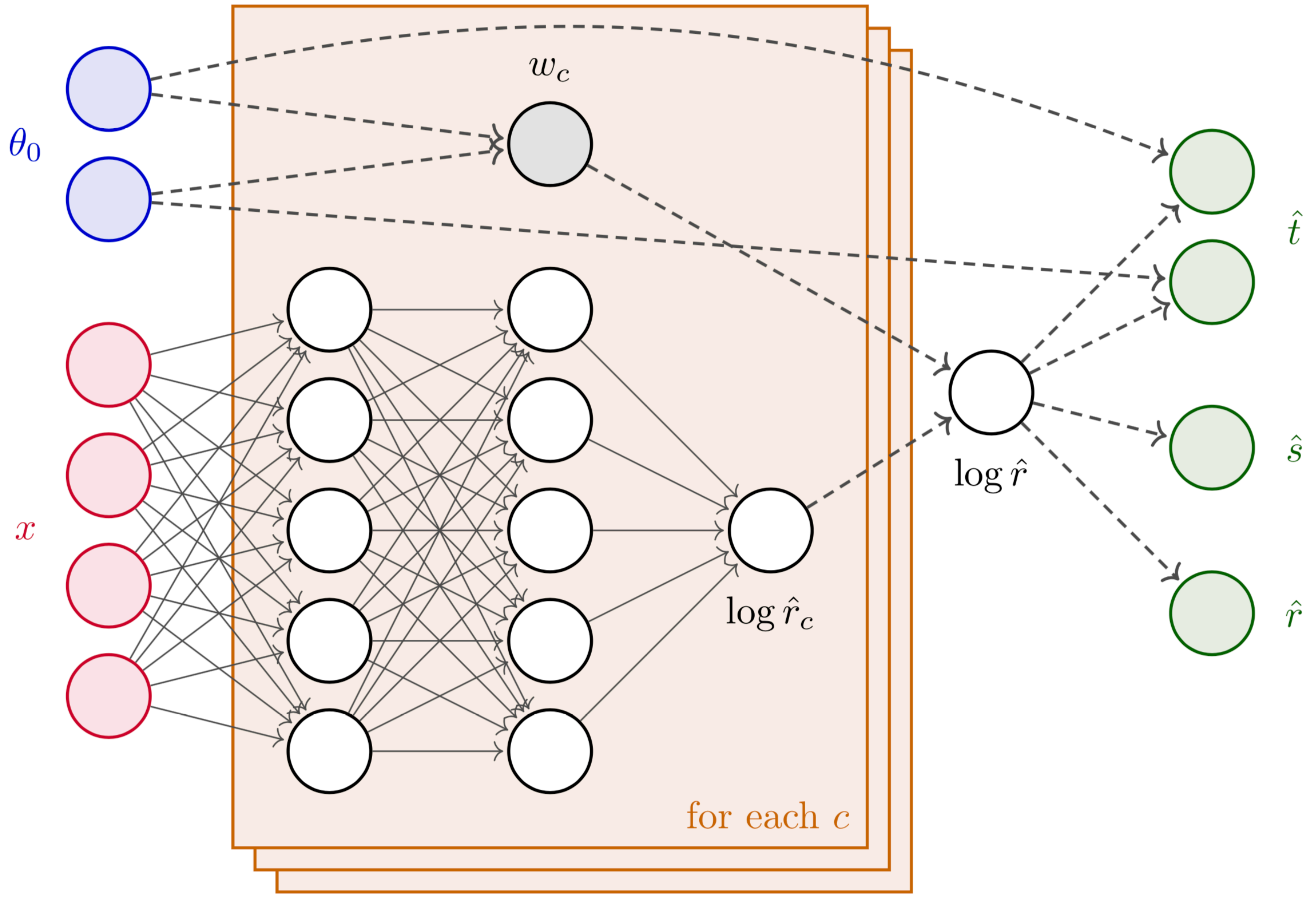
Point by point



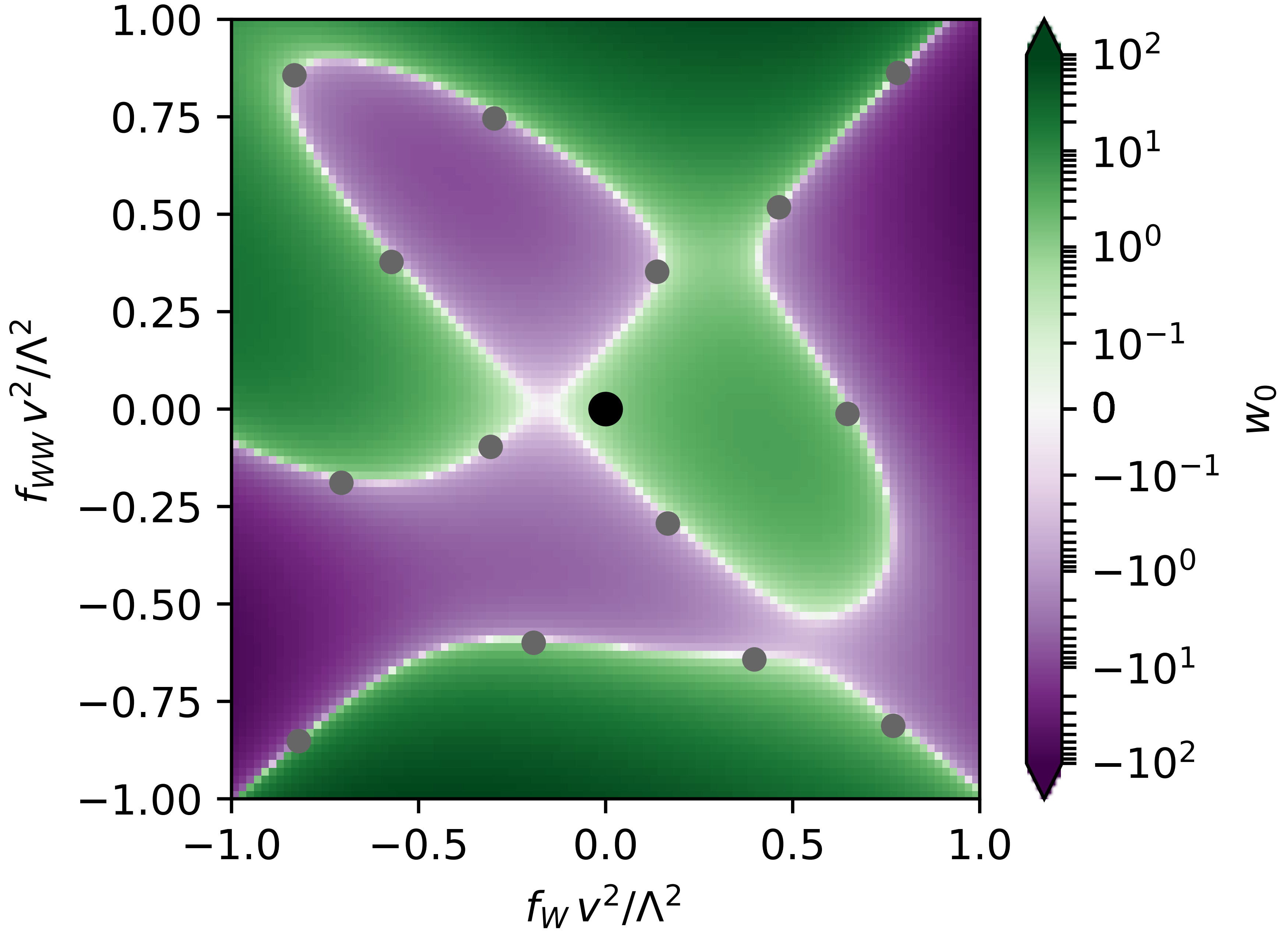
(Agnostic) parameterized estimators



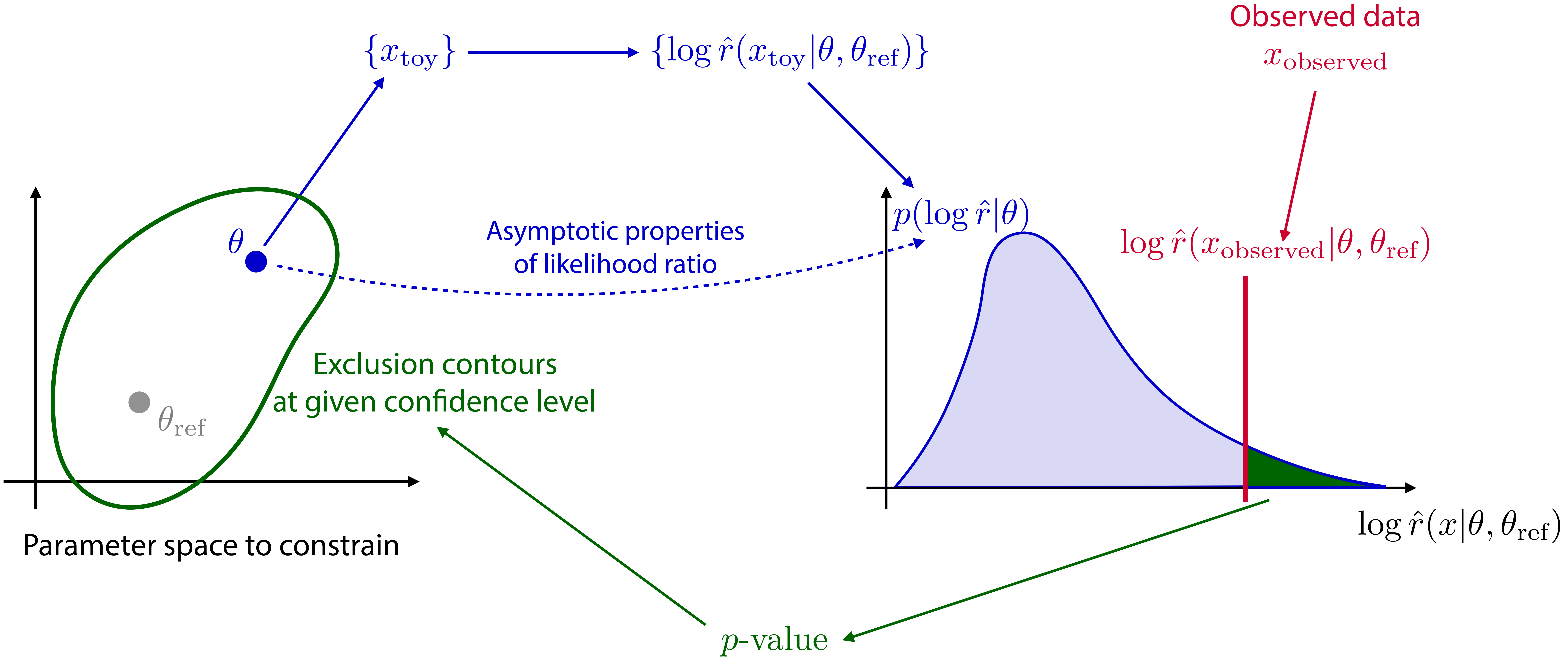
Morphing-aware parameterized estimators



$$\hat{r}(x|\theta_0, \theta_1) = \sum_c w_c(\theta_0)\, \hat{r}_c(x)$$



Limit setting (frequentist)



Detector effects