“Mining gold” from the simulator to improve likelihood-free inference

Johann Brehmer

New York University

#LFIweek

CCA, Flatiron Institute — March 18, 2019

Kyle Cranmer

Gilles Louppe

Juan Pavez

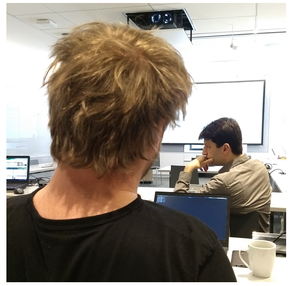
Felix Kling

Irina Espejo

Zubair Bhatti

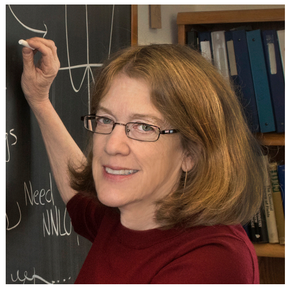
Markus Stoye

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!

NYU

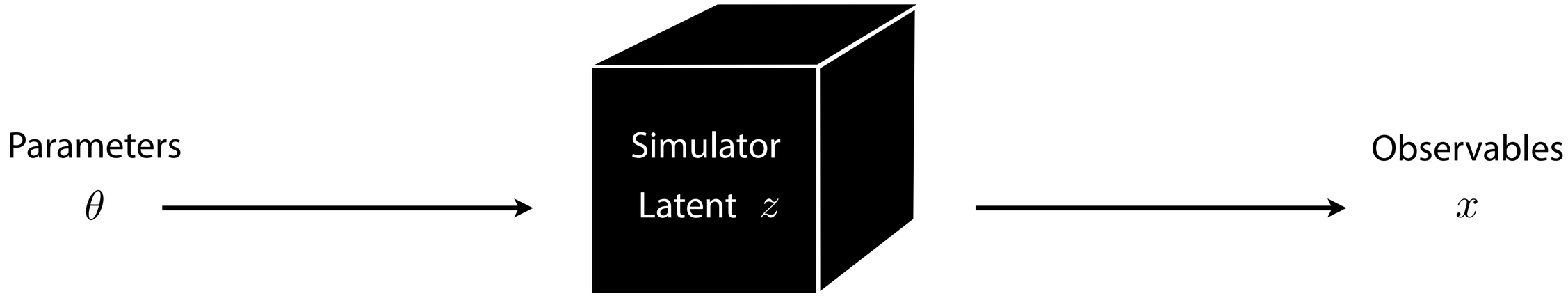
MSDSE
Moore - Sloan
Data Science Environments

diana hep

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

The SCAILFIN Project
scailfin.github.io

Likelihood-free inference / implicit models

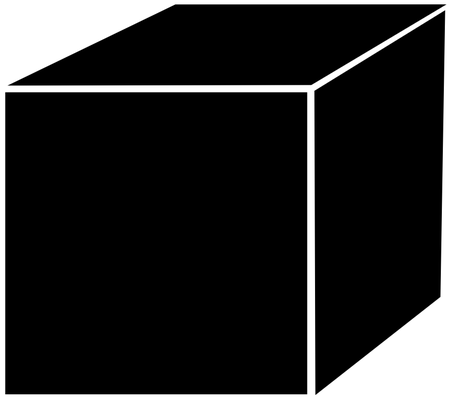


$$p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta)$$

Prediction:
- Well-understood mechanistic model
- Simulator can sample $x \sim p(x|\theta)$

Inference:
- Likelihood $p(x|\theta)$ is intractable
- Inference e.g. through estimator $\hat{p}(x|\theta)$

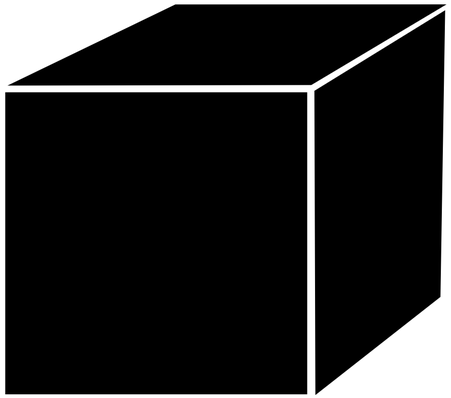
Opening the black box



“Mainstream LFI” treats simulator as generative black box: parameters in, samples out.

No one cares how the sausage is made.

Opening the black box



"Mainstream LFI" treats simulator as generative black box: parameters in, samples out.

No one cares how the sausage is made.

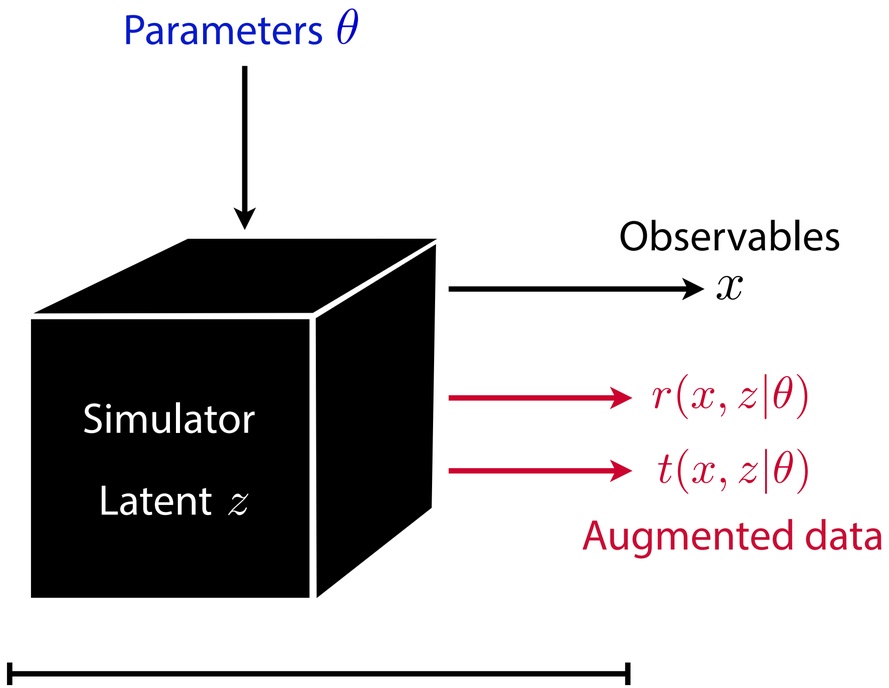
But in real-life problems, we have access to the simulator code and some understanding of the microscopic processes.

We can extract more simulation from the simulator and use it to improve inference.

“Mining gold” from the simulator

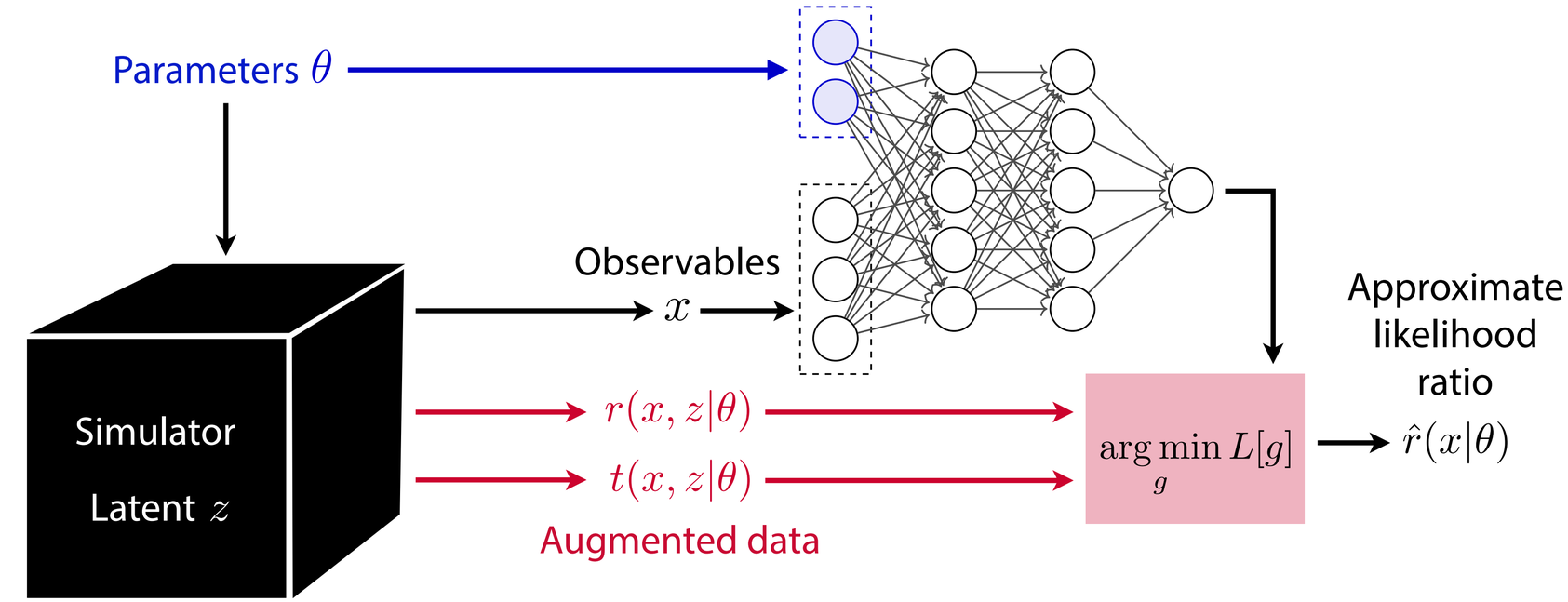
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

Bird's-eye view



"Mining gold": Extract additional information from simulator

Bird's-eye view



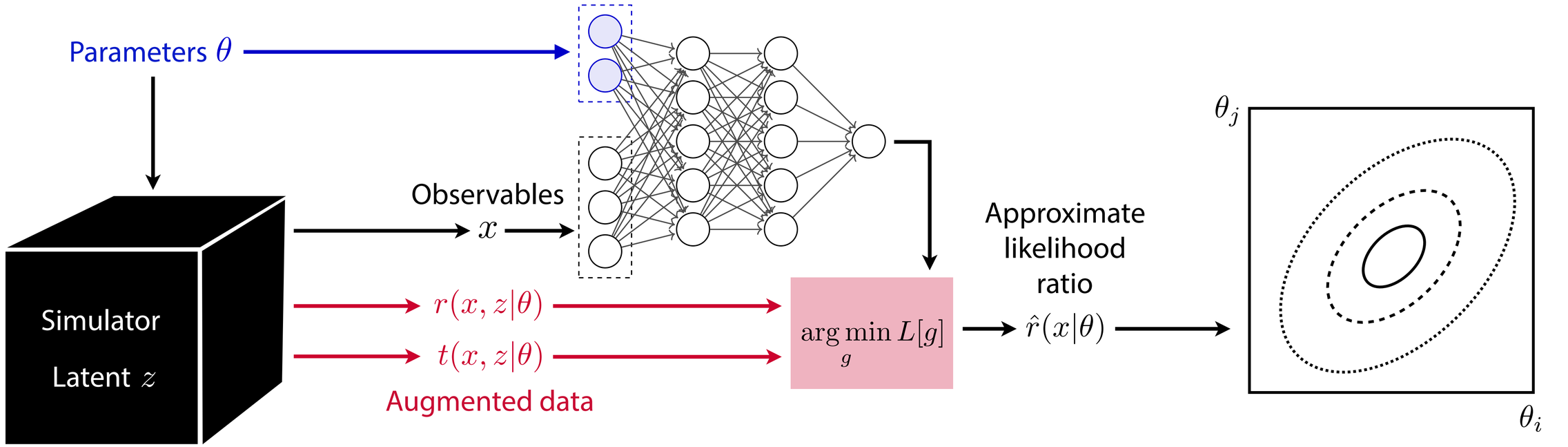
1. Simulation

"Mining gold": Extract additional information from simulator

2. Machine Learning

Use this information to train estimator for likelihood ratio

Bird's-eye view



1. Simulation

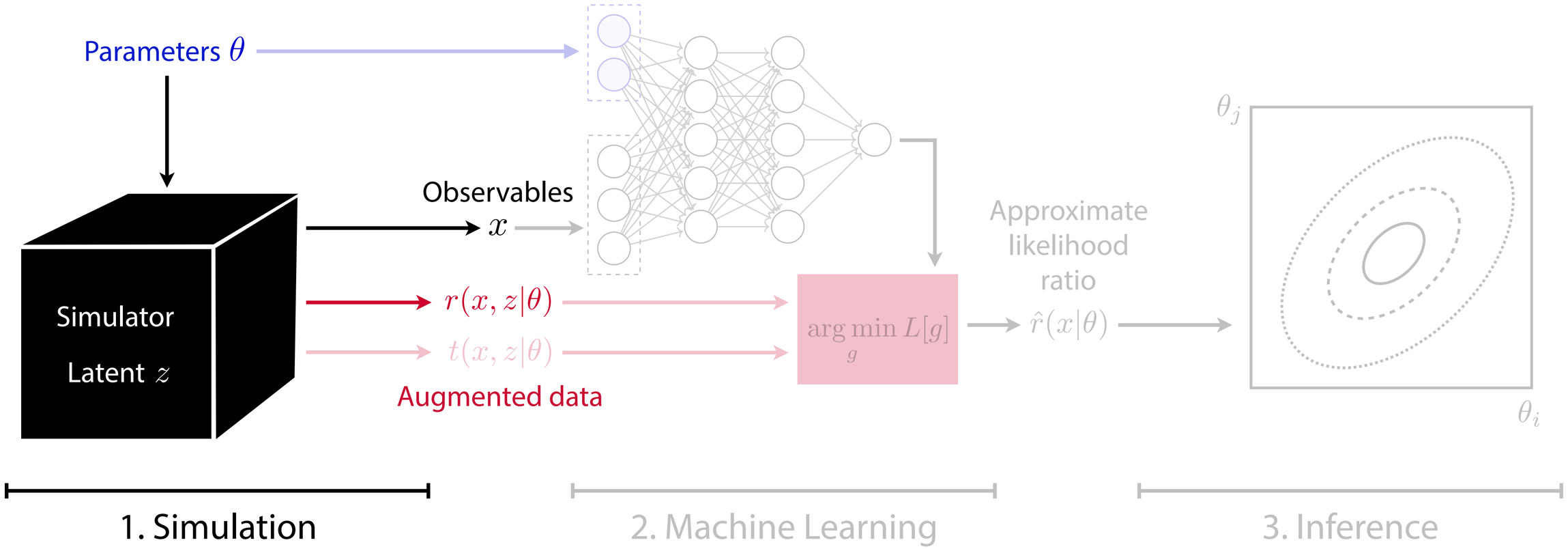
"Mining gold": Extract additional information from simulator

2. Machine Learning

Use this information to train estimator for likelihood ratio

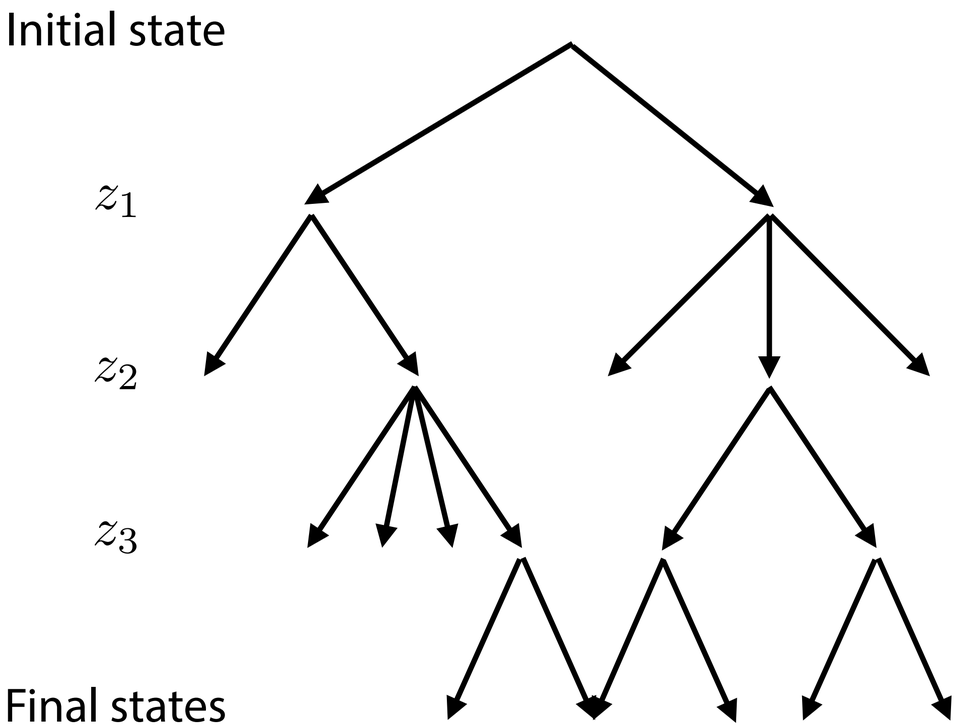
3. Inference

Frequentist: hypothesis tests
Bayesian: MCMC, VI


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

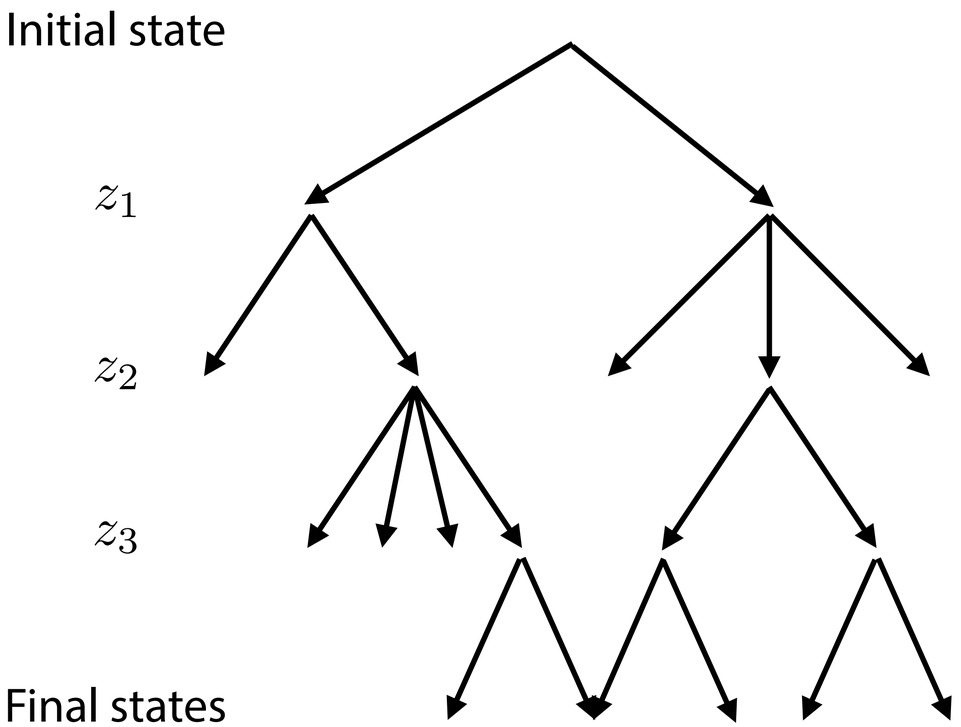
- Computer simulation typically evolve along a tree-like structure of successive random branchings



Mining gold from the simulator

- 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()
```



Mining gold from the simulator

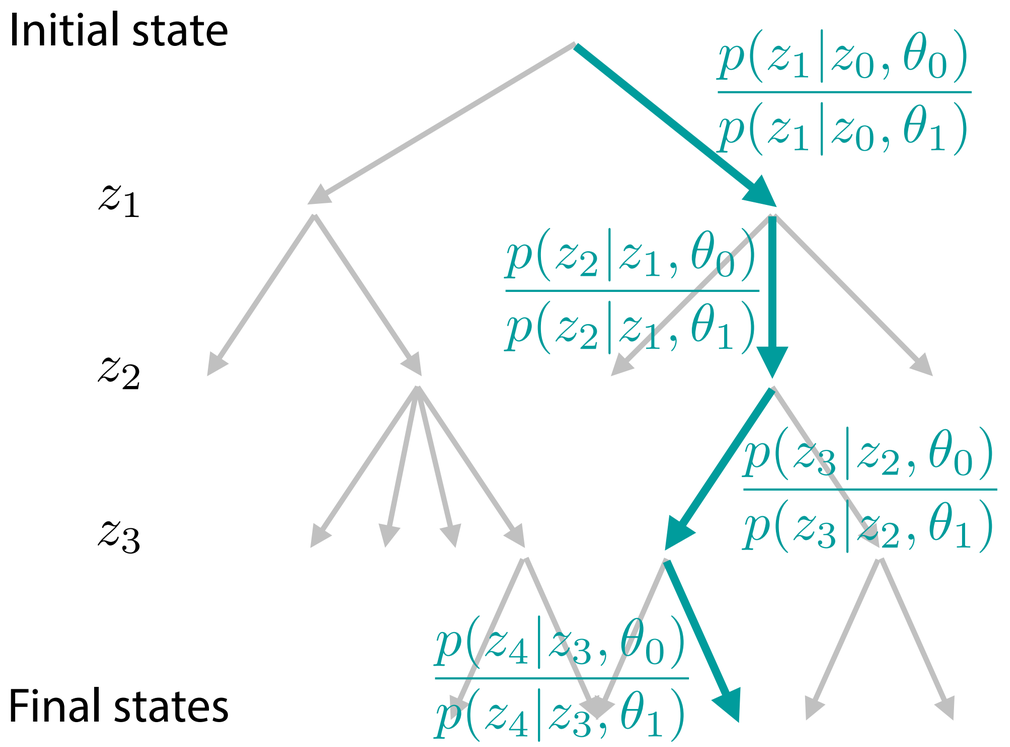
- Computer simulation typically evolve along a tree-like structure of successive random branchings

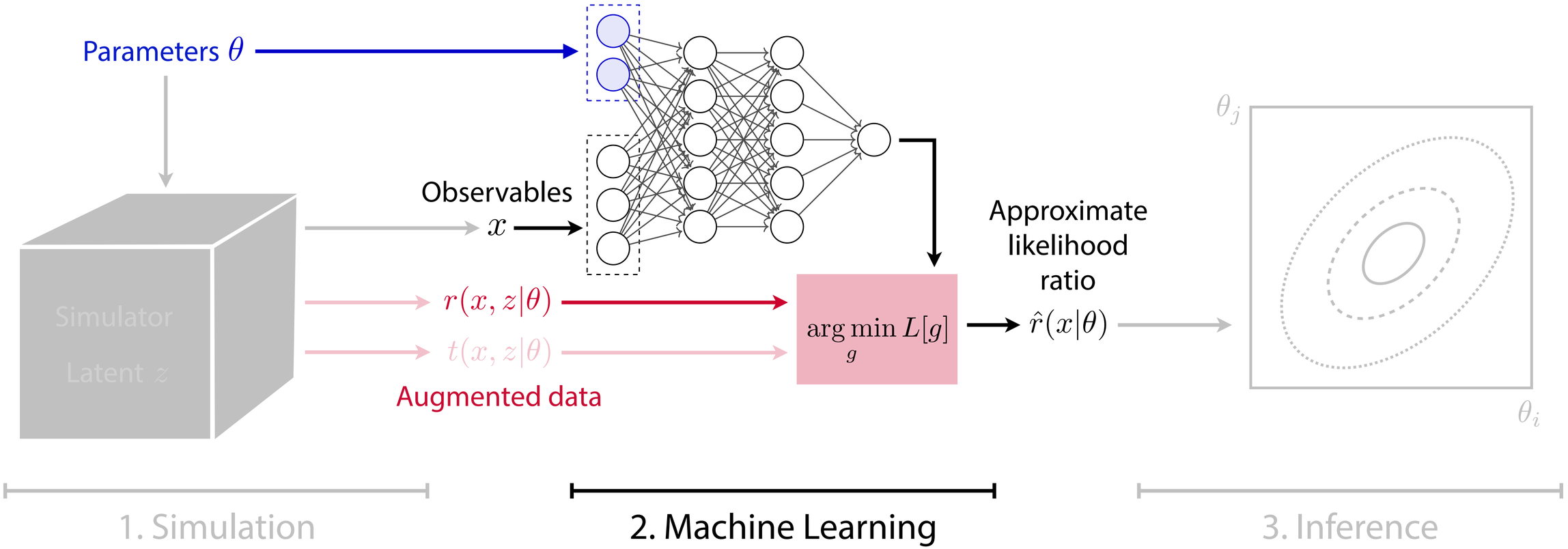
- The probabilities of each branching $p_i(z_i|z_{i-1},\theta)$ are often clearly defined in the code:

```
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":

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




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

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

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

We want the likelihood ratio function

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

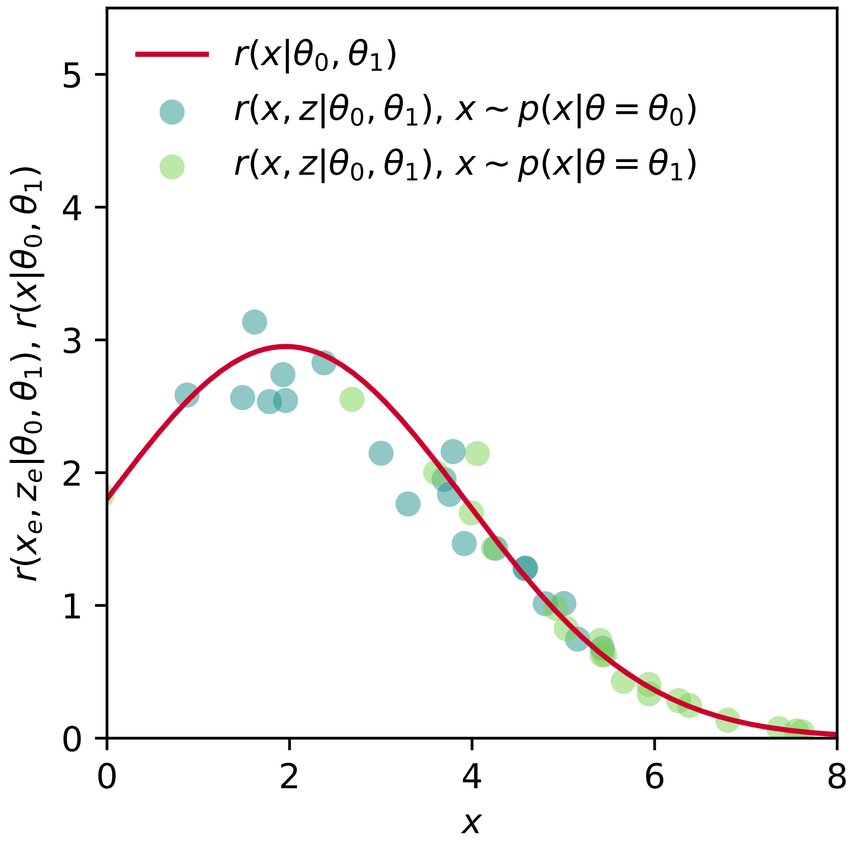
("How much more likely is the observation x for θ_0 compared to θ_1?")

The value of gold

We can calculate the joint likelihood ratio

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

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



We want the likelihood ratio function

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

The value of gold

We can calculate the joint likelihood ratio

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

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

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

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

It is minimized by

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

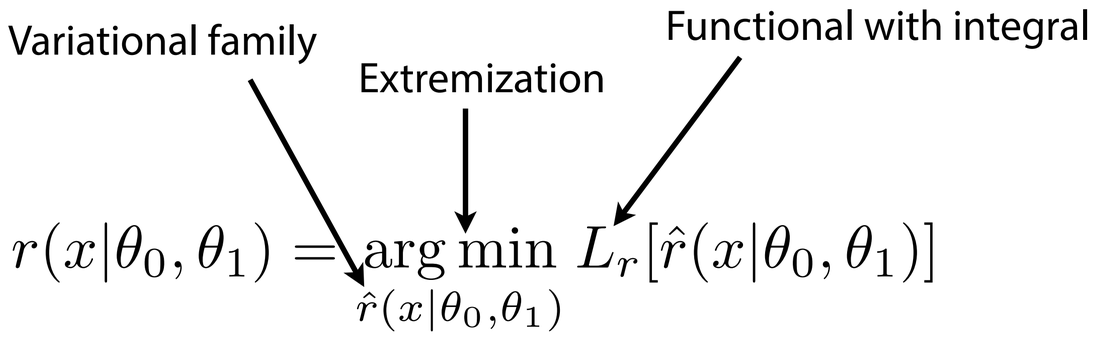
(And we can sample from $p(x, z|\theta)$ by running the simulator.)

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

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



Machine learning = applied calculus of variatios

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

Variational family

Extremization

Functional with integral

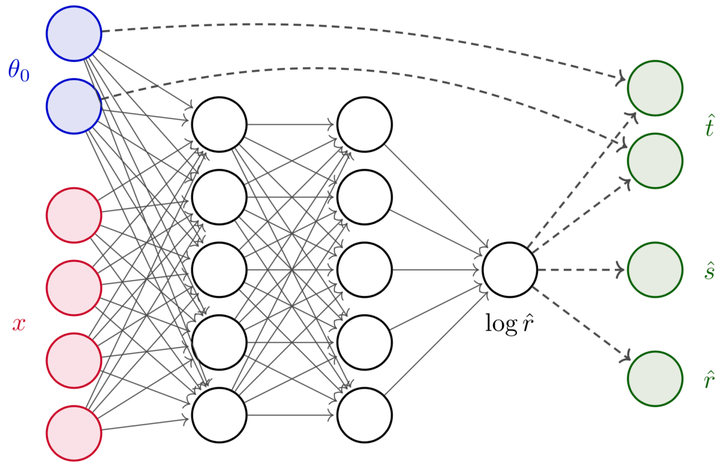
$$r(x|\theta_0,\theta_1) = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\ L_r[\hat{r}(x|\theta_0,\theta_1)]$$

This is where machine learning comes in!

Neural network

Stochastic gradient descent

Loss function with
finite sum over samples



Machine learning = applied calculus of variatios

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

Variational family

Extremization

Functional with integral

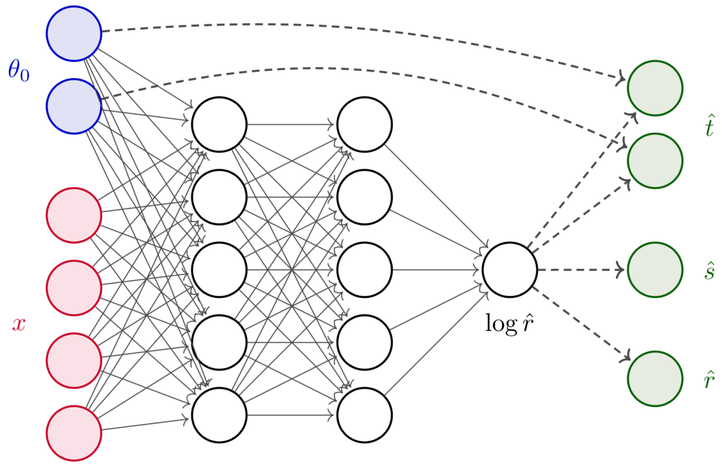
$$r(x|\theta_0,\theta_1) = \underset{\hat{r}(x|\theta_0,\theta_1)}{\arg\min}\ L_r[\hat{r}(x|\theta_0,\theta_1)]$$

This is where machine learning comes in!

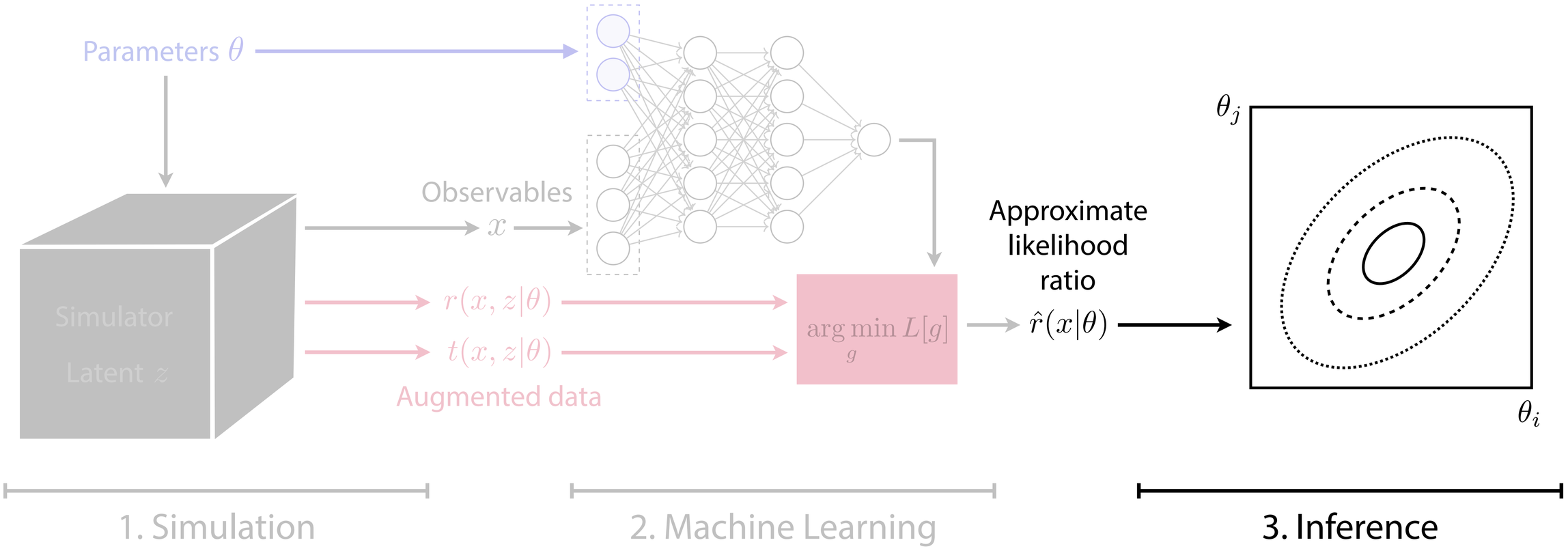
Neural network

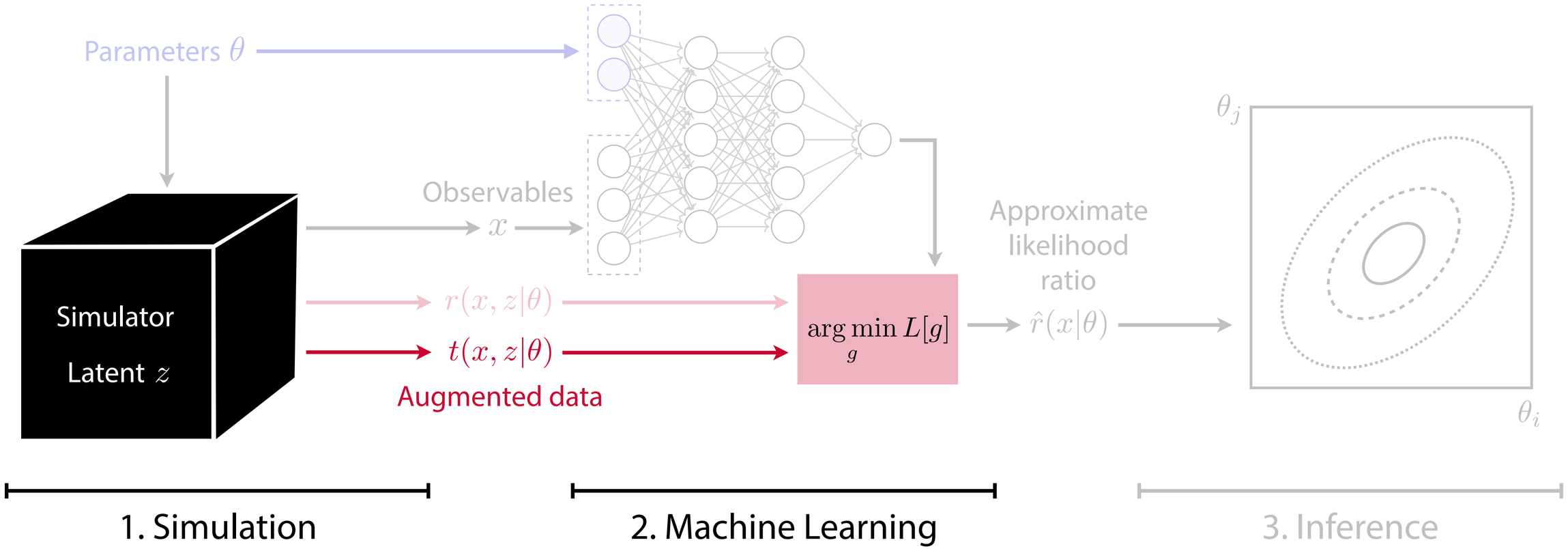
Stochastic gradient descent

Loss function with
finite sum over samples



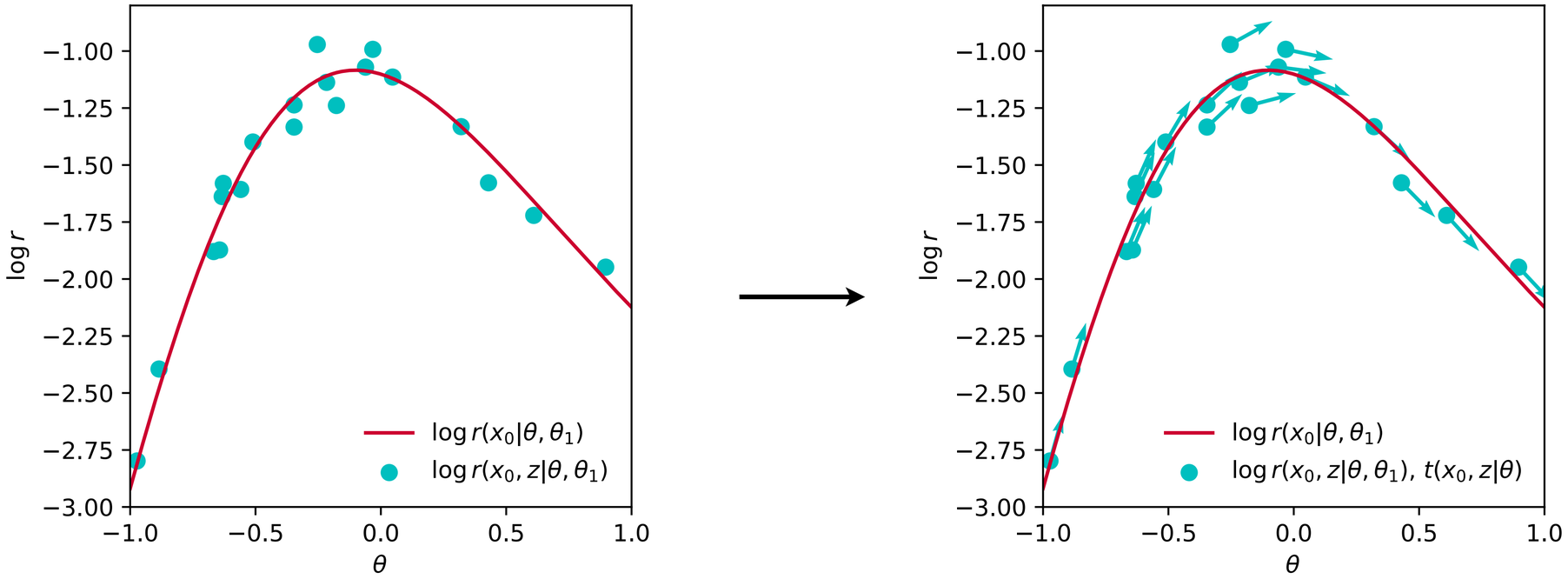
A sufficiently expressive neural network
efficiently trained in this way
with enough data will learn
the likelihood ratio function $r(x|\theta_0,\theta_1)$!


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


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

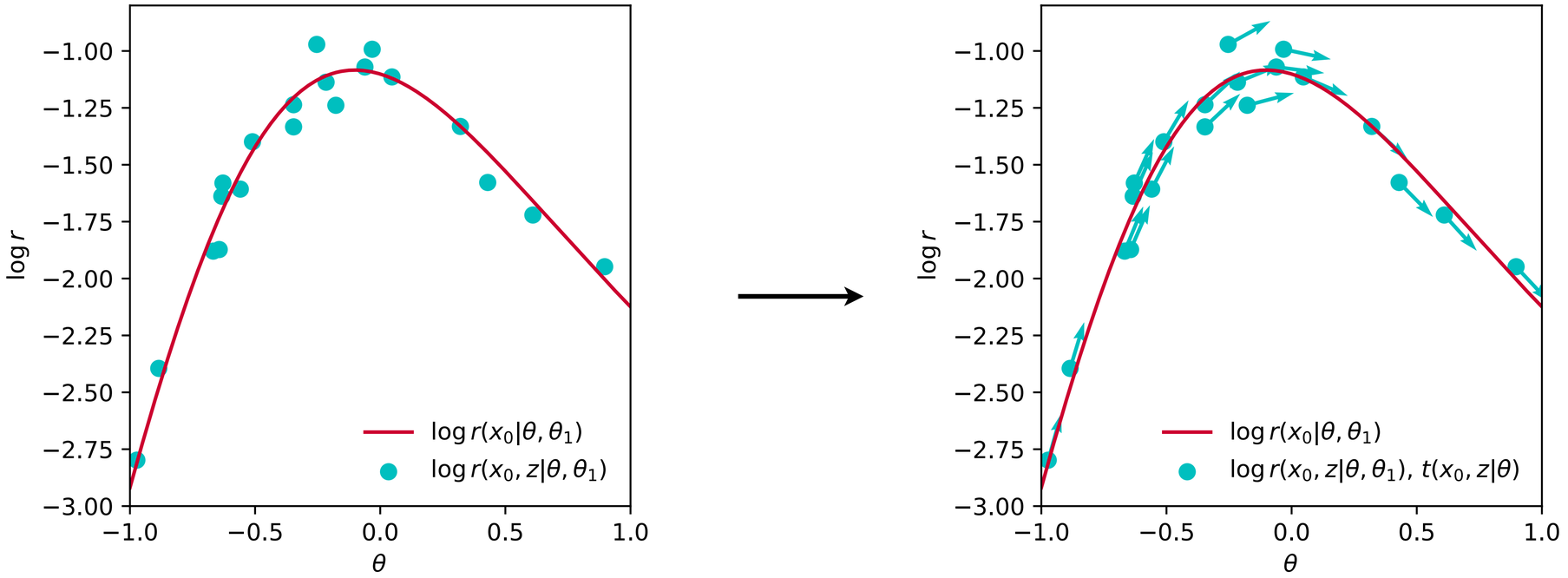
One more piece: the score

- 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

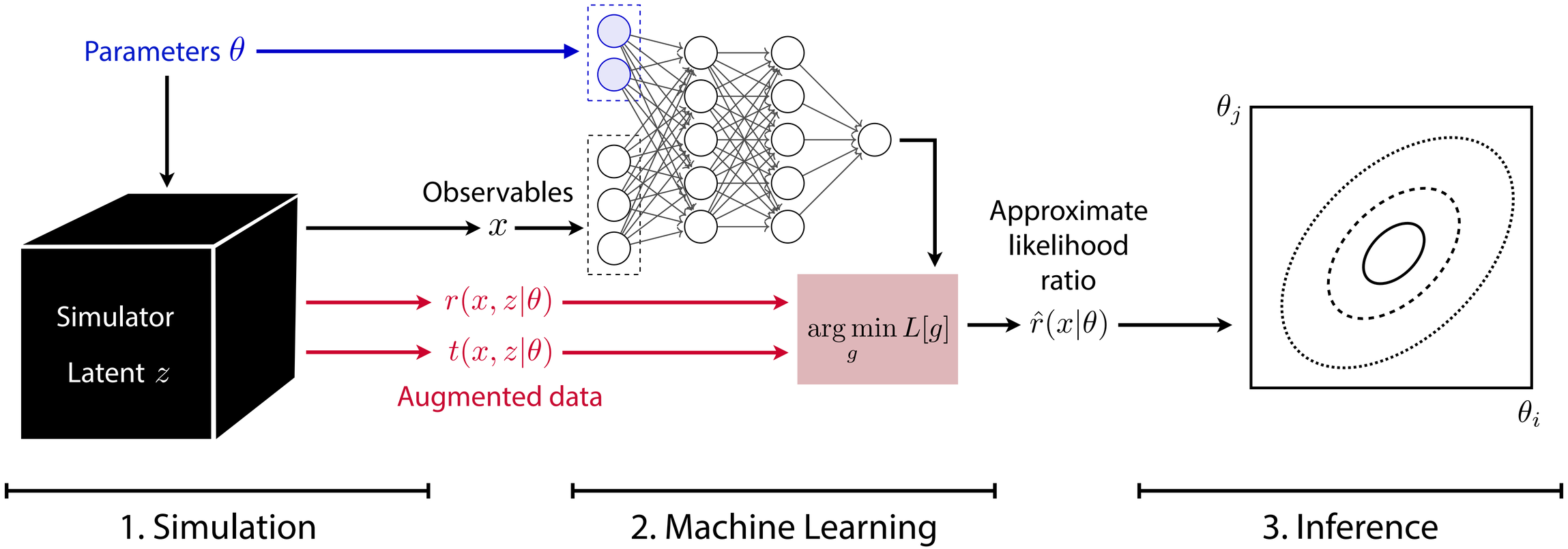
$$L_t[\hat{t}(x|\theta_0)] = \int \mathrm{d}x \int \mathrm{d}z \;\; p(x, z|\theta_0) \;\; \Big[\big(\hat{t}(x|\theta_0) - t(x, z|\theta_0)\big)^2\Big].$$

One can show it is minimized by

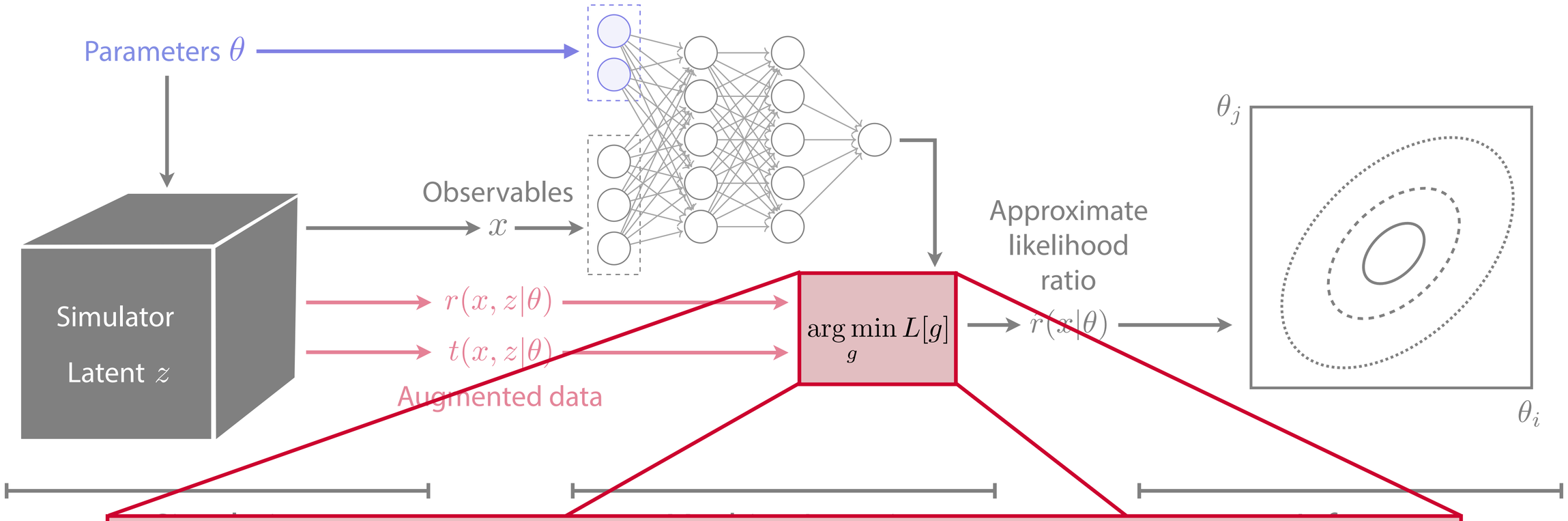
$$t(x|\theta_0) = \arg\min_{\hat{t}(x|\theta_0)} L_t[\hat{t}(x|\theta_0)]\,.$$

Again, we implement this minimization
through machine learning.

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



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

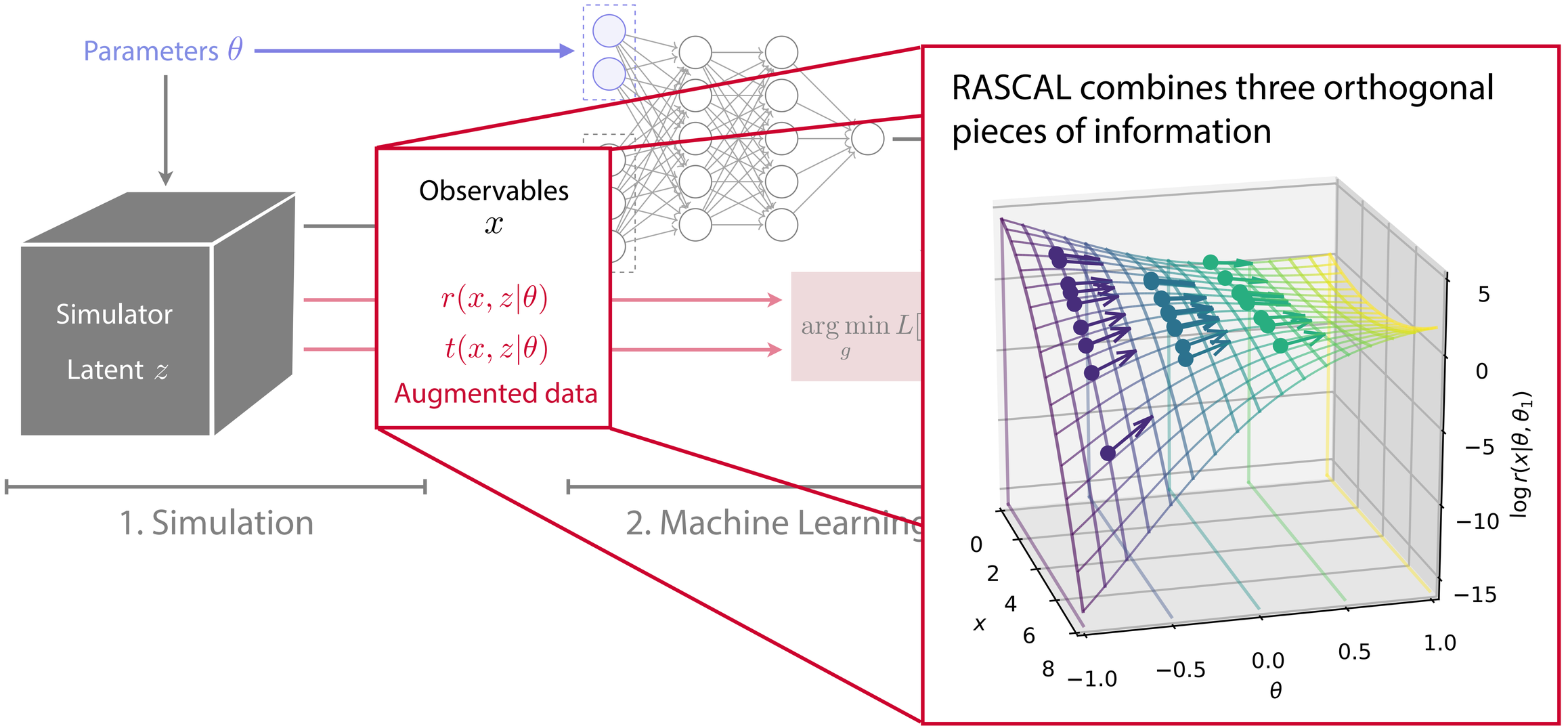


Combining ratio and score information:

- 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:

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

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



Learning locally optimal summary statistics

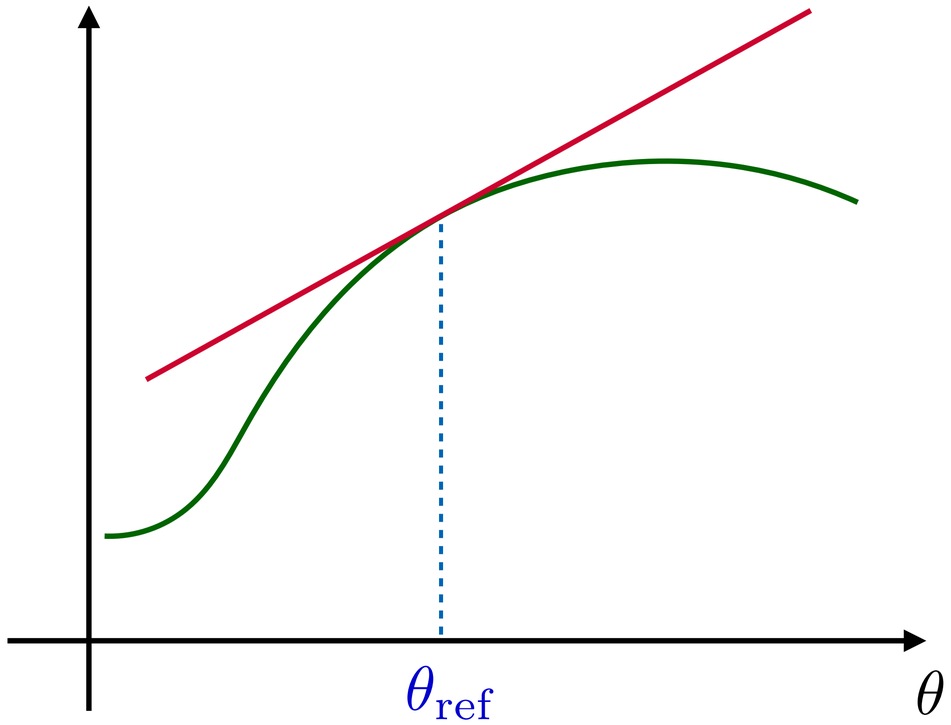
[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244]

The local model

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

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

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

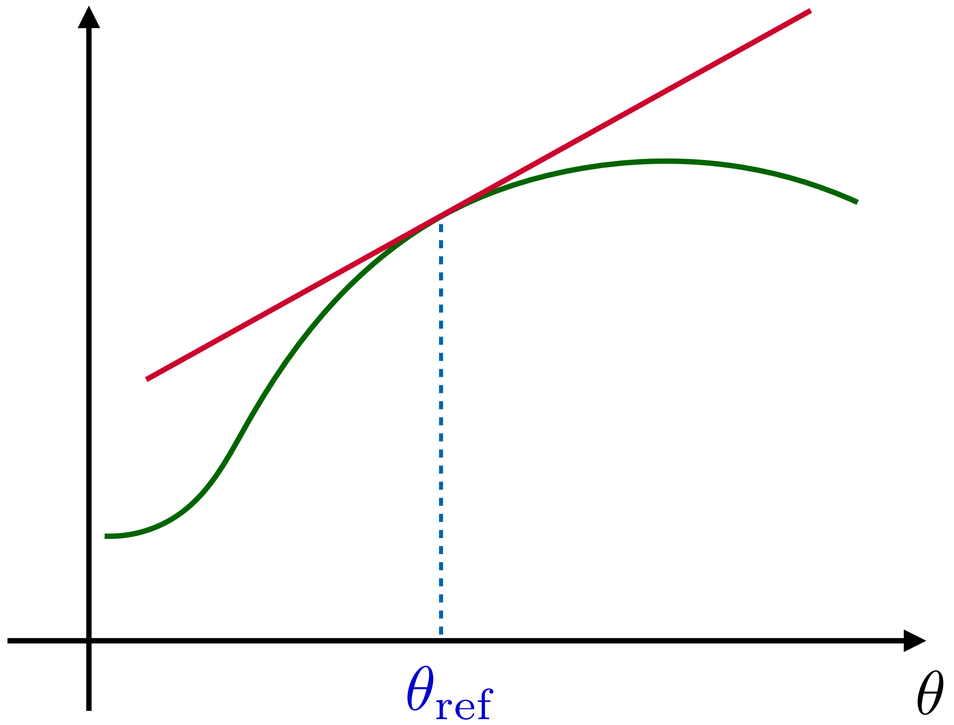


The local model

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

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

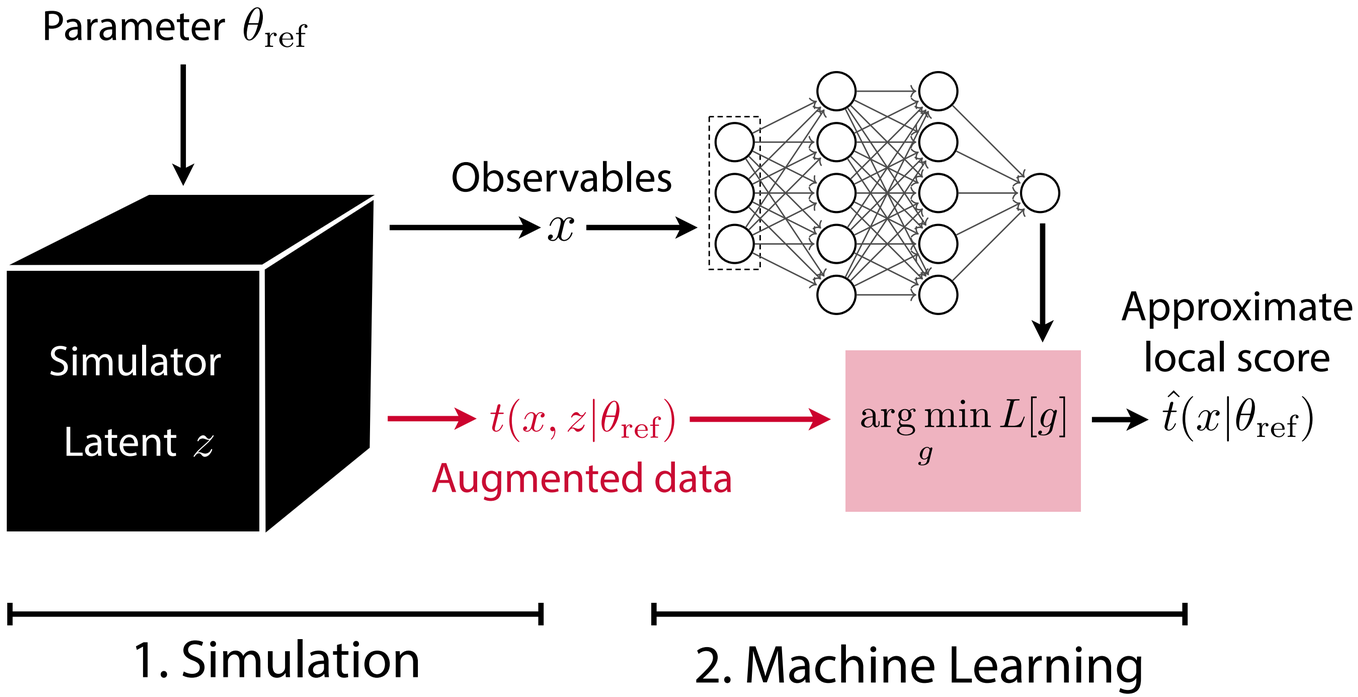
$$\log p(x|\theta) = \log p(x|\theta_{\rm ref}) + \underbrace{\nabla_\theta \log p(x|\theta)\Big|_{\theta_{\rm ref}}}_{\equiv t(x|\theta_{\rm ref})} \cdot(\theta - \theta_{\rm ref}) + \mathcal{O}\left((\theta - \theta_{\rm ref})^2\right)$$



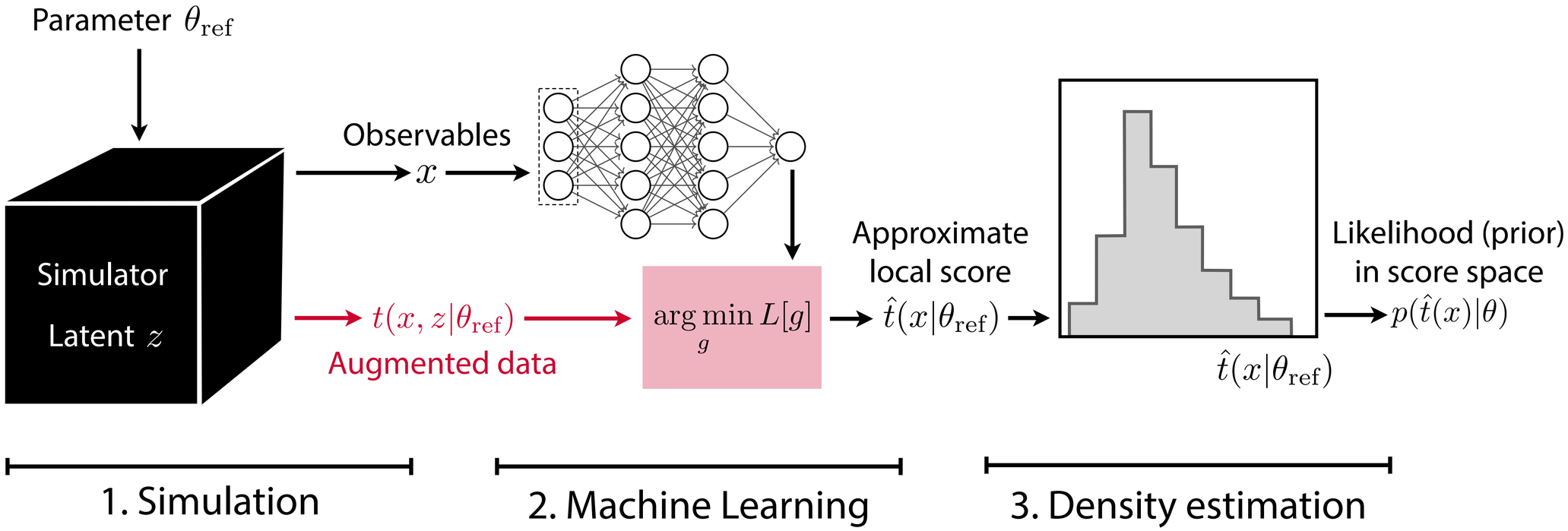
In the neighborhood of $\theta_{\rm ref}$ (e.g. close to the SM):

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

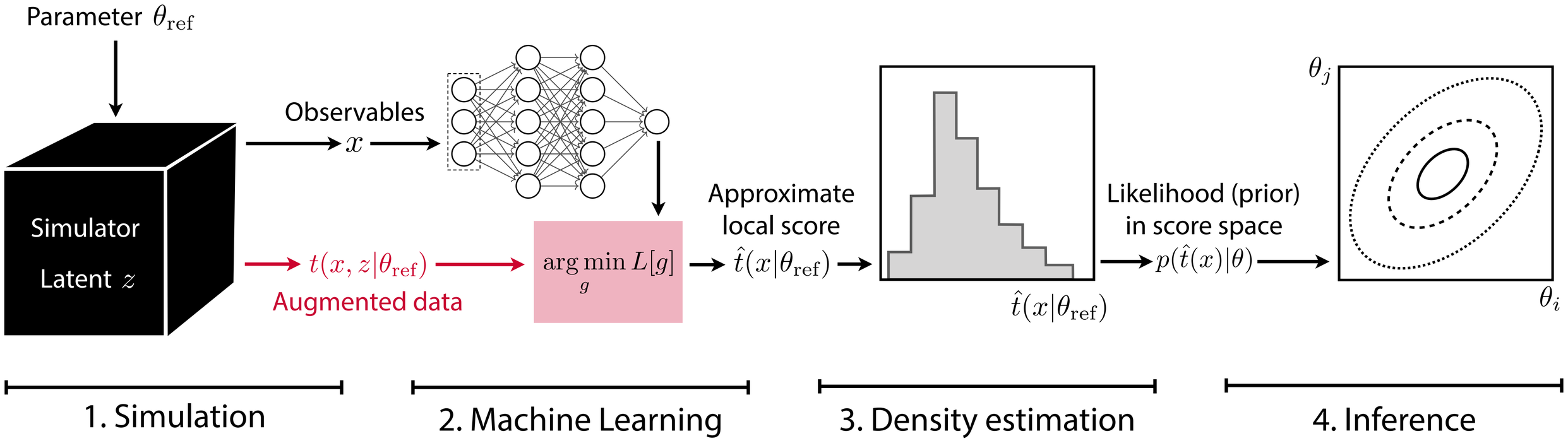
SALLY (Score approximates likelihood locally)



SALLY (Score approximates likelihood locally)



SALLY (Score approximates likelihood locally)



A family of new inference techniques

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

A family of new inference techniques

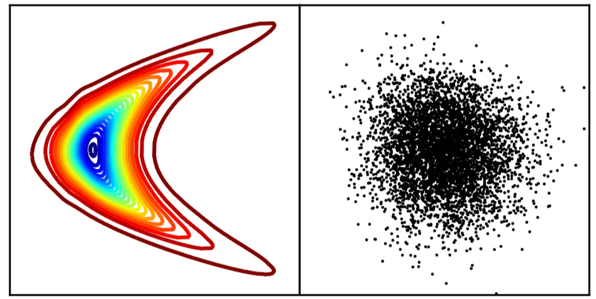
Method	Simulate	Extract $r(x,z)$	Extract $t(x,z)$	NN estimates	Asympt. exact	Generative
ROLR	$\theta_0 \sim \pi(\theta), \theta_1$	✓		$\hat{r}(x\|\theta_0,\theta_1)$	✓	
CASCAL	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0,\theta_1)$	✓	
ALICE	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0,\theta_1)$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

Performance gains with cross-entropy-based loss

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

A family of new inference techniques

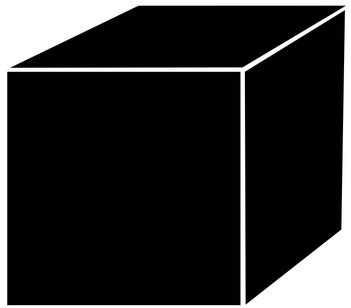
Method	Simulate	Extract $r(x,z)$	Extract $t(x,z)$	NN estimates	Asympt. exact	Generative
ROLR	$\theta_0 \sim \pi(\theta), \theta_1$	✓		$\hat{r}(x\|\theta_0,\theta_1)$	✓	
CASCAL	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0,\theta_1)$	✓	
ALICE	$\theta_0 \sim \pi(\theta), \theta_1$		✓	$\hat{r}(x\|\theta_0,\theta_1)$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



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

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

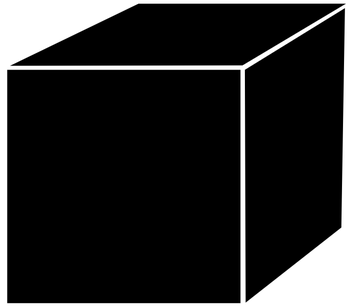
Systematics

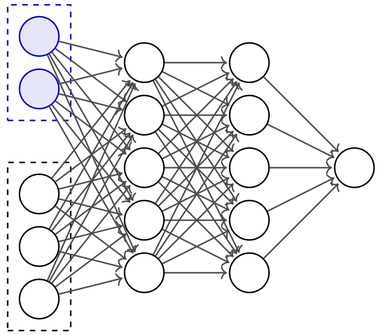


Don’t fully trust the simulator?

- 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
  [P. de Castro, T. Dorigo 1806.04743;
  J. Alsing, B. Wandelt 1903.01473]
- 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
  [P. de Castro, T. Dorigo 1806.04743;
  J. Alsing, B. Wandelt 1903.01473]
- Alternatively: Robustness to nuisance with adversarial training
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046]

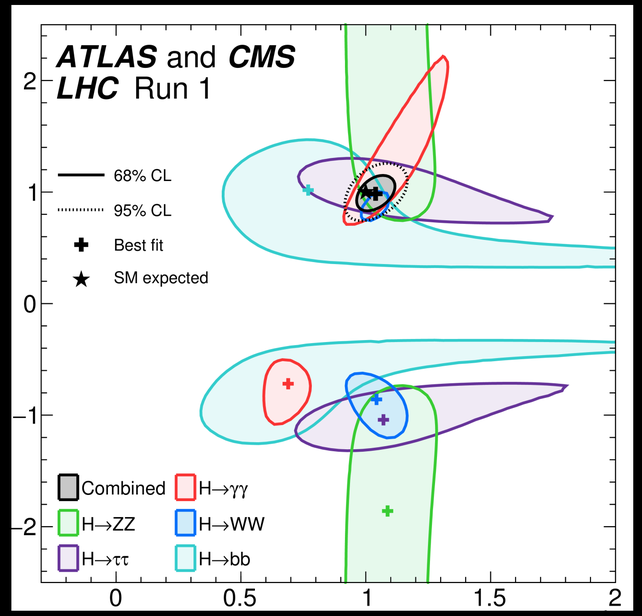
Don’t blindly trust the neural network?

- 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

Particle physics

High-dimensional event data

[CMS]

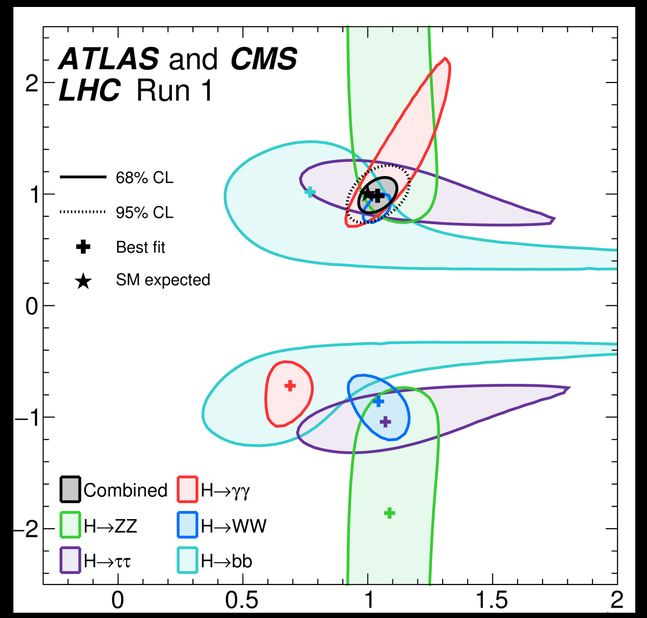


[ATLAS, CMS 1606.02266]

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:

systematic expansion of new physics around Standard Model

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

10s to 1000s “universal” parameters to measure

Modelling particle physics processes

Theory
parameters
θ

Evolution

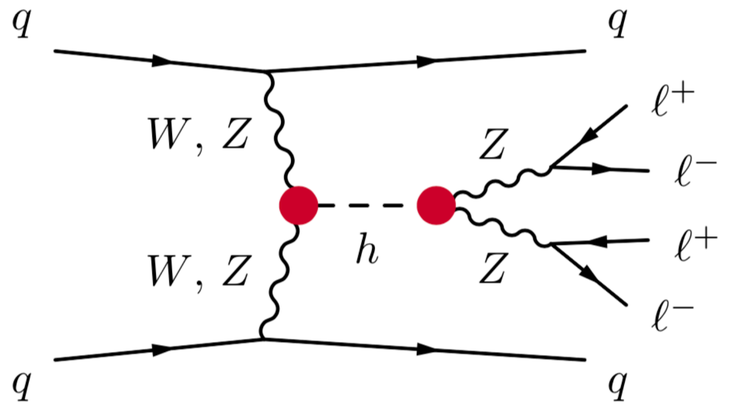
Modelling particle physics processes

Latent variables

Parton-level momenta

Theory parameters

z_p ← θ

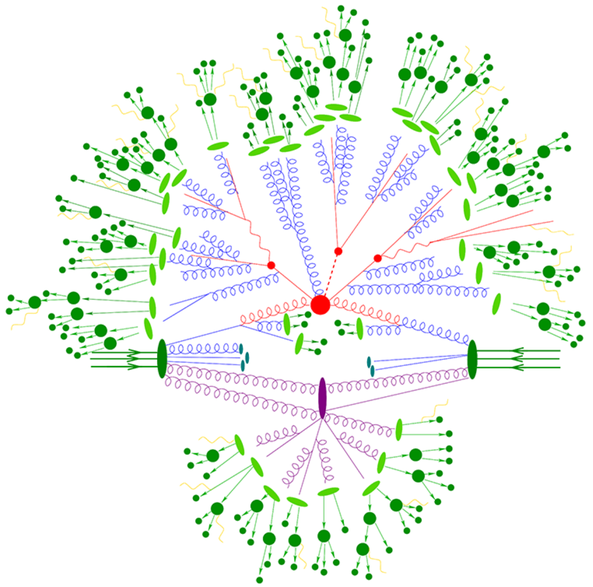


Evolution

Modelling particle physics processes

Latent variables

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



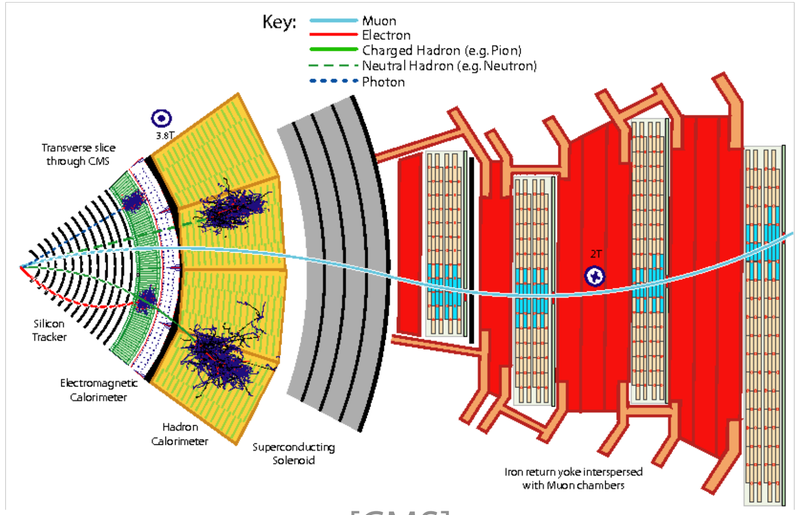
[F. Krauss]

Evolution

Modelling particle physics processes

Latent variables

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



[CMS]

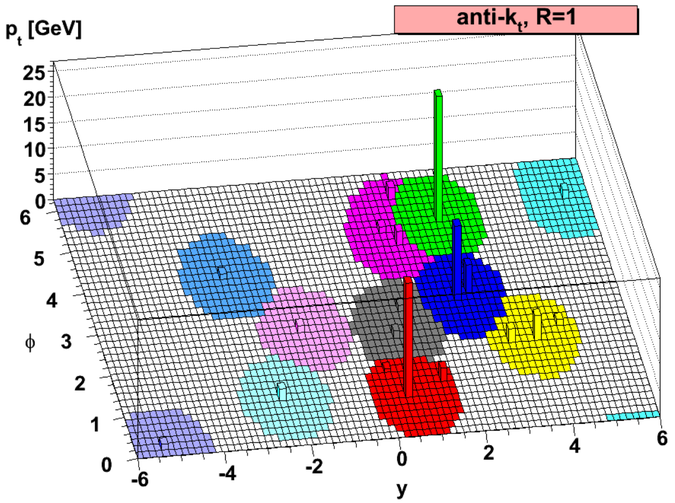
Evolution

Modelling particle physics processes

Latent variables

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

$x \leftarrow z_d \leftarrow z_s \leftarrow z_p \leftarrow \theta$



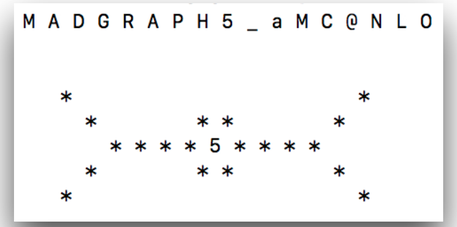
[M. Cacciari, G. Salam, G. Soyez 0802.1189]

Evolution

Modelling particle physics processes

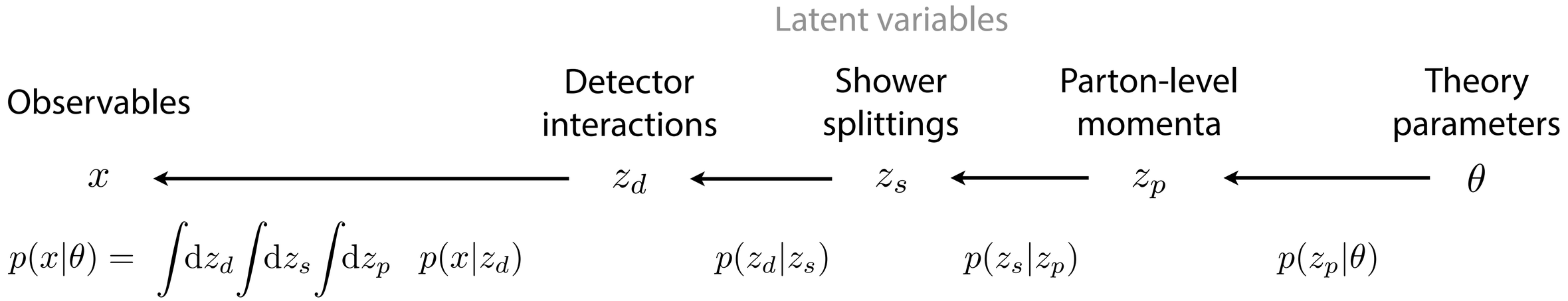
Latent variables

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



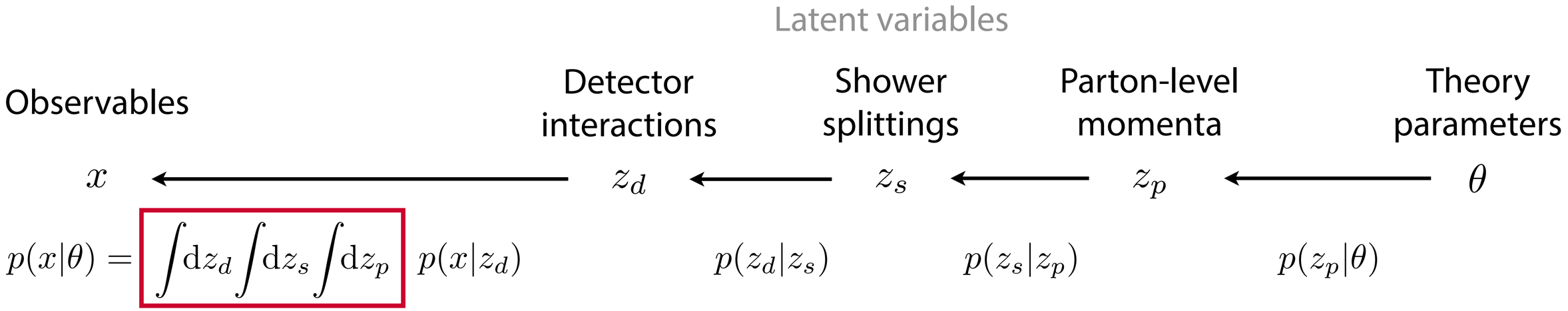
Prediction (simulation)

Modelling particle physics processes



Inference

Modelling particle physics processes

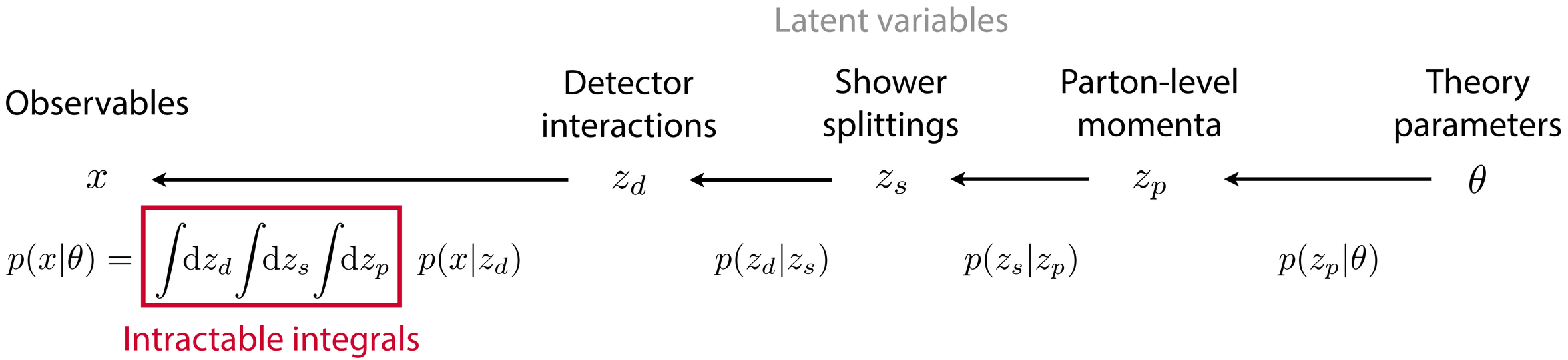


It's infeasible to calculate the integral over this enormous space!

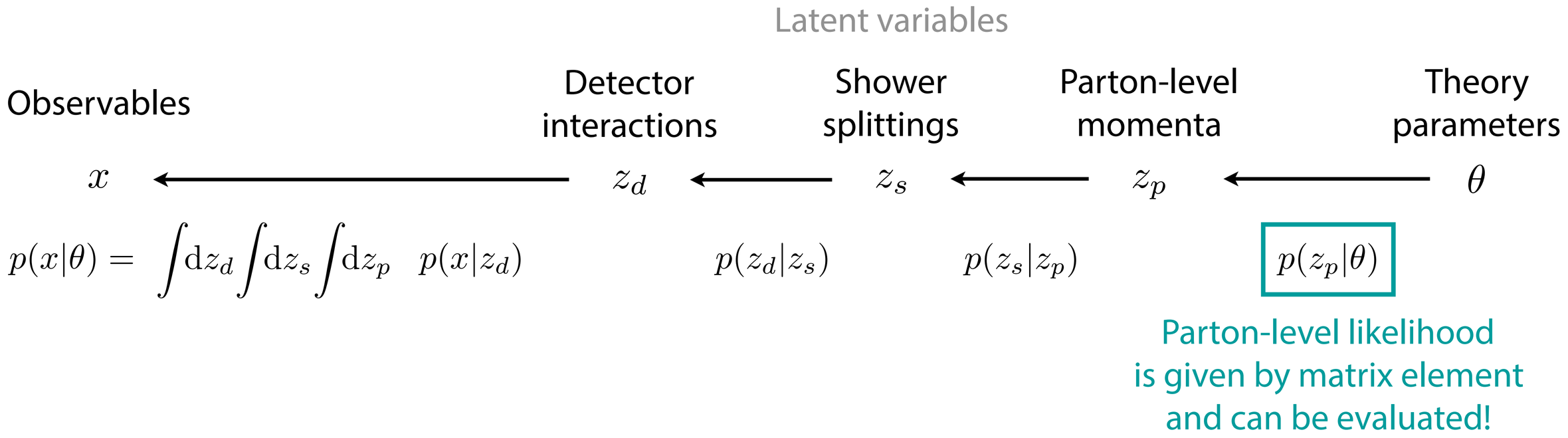
(More subtle: We cannot sample from $p(z|x,\theta)$ efficiently.)

Inference

Mining gold from particle physics simulators



Mining gold from particle physics simulators



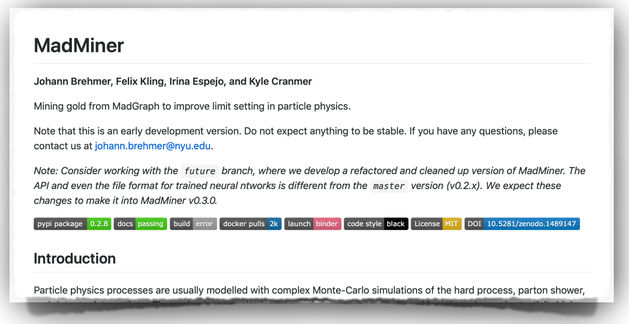
⇒ For each simulated event, we can calculate the joint likelihood ratio which depends on the specific evolution of the simulation:

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

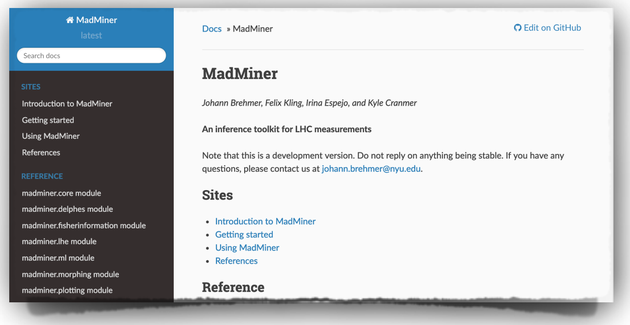
MadMiner

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

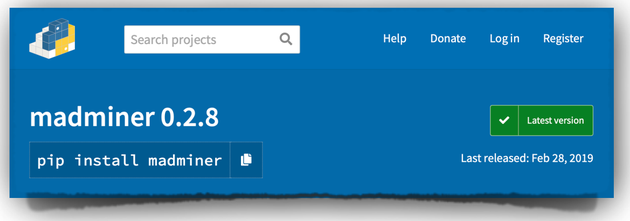
We are developing the Python module MadMiner, which makes it straightforward to "mine the gold" from LHC simulations and to apply the new inference techniques



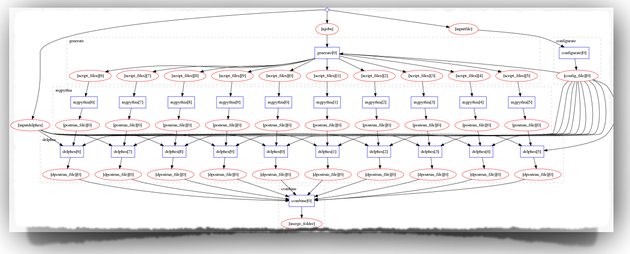
Repository and tutorials:
github.com/johannbrehmer/madminer



Documentation:
madminer.readthedocs.io



Installation:
`pip install madminer`

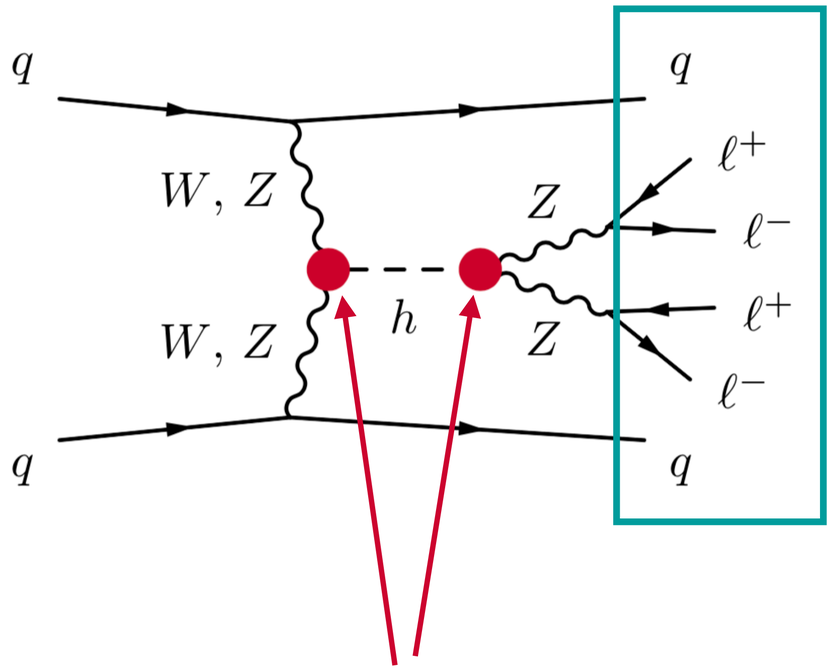


Deployment with Docker, yadage, REANA:
github.com/irinaespejo/workflow-madminer

Proof of concept: Higgs production in weak boson fusion

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

at least 16-dimensional
observable space



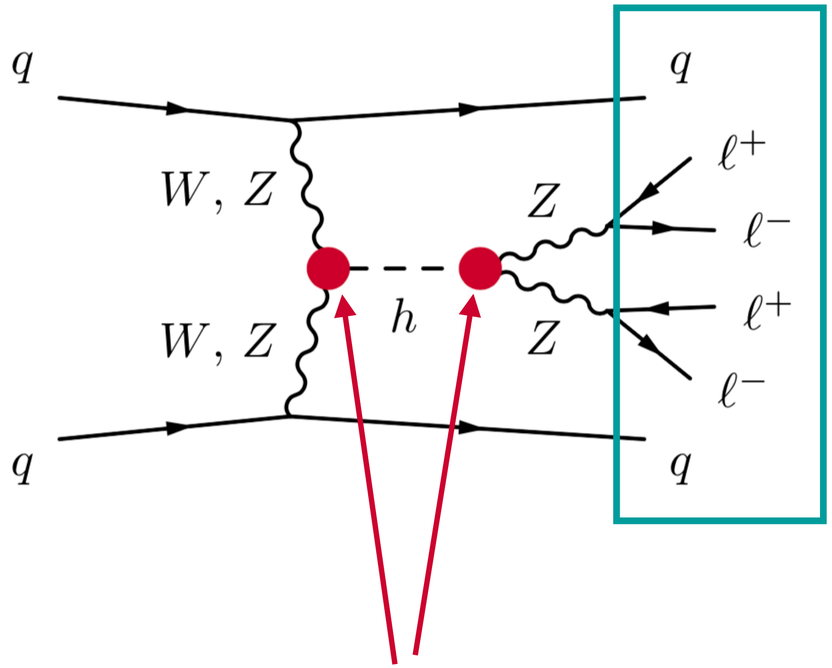
Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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

Proof of concept: Higgs production in weak boson fusion

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

at least 16-dimensional observable space



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

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

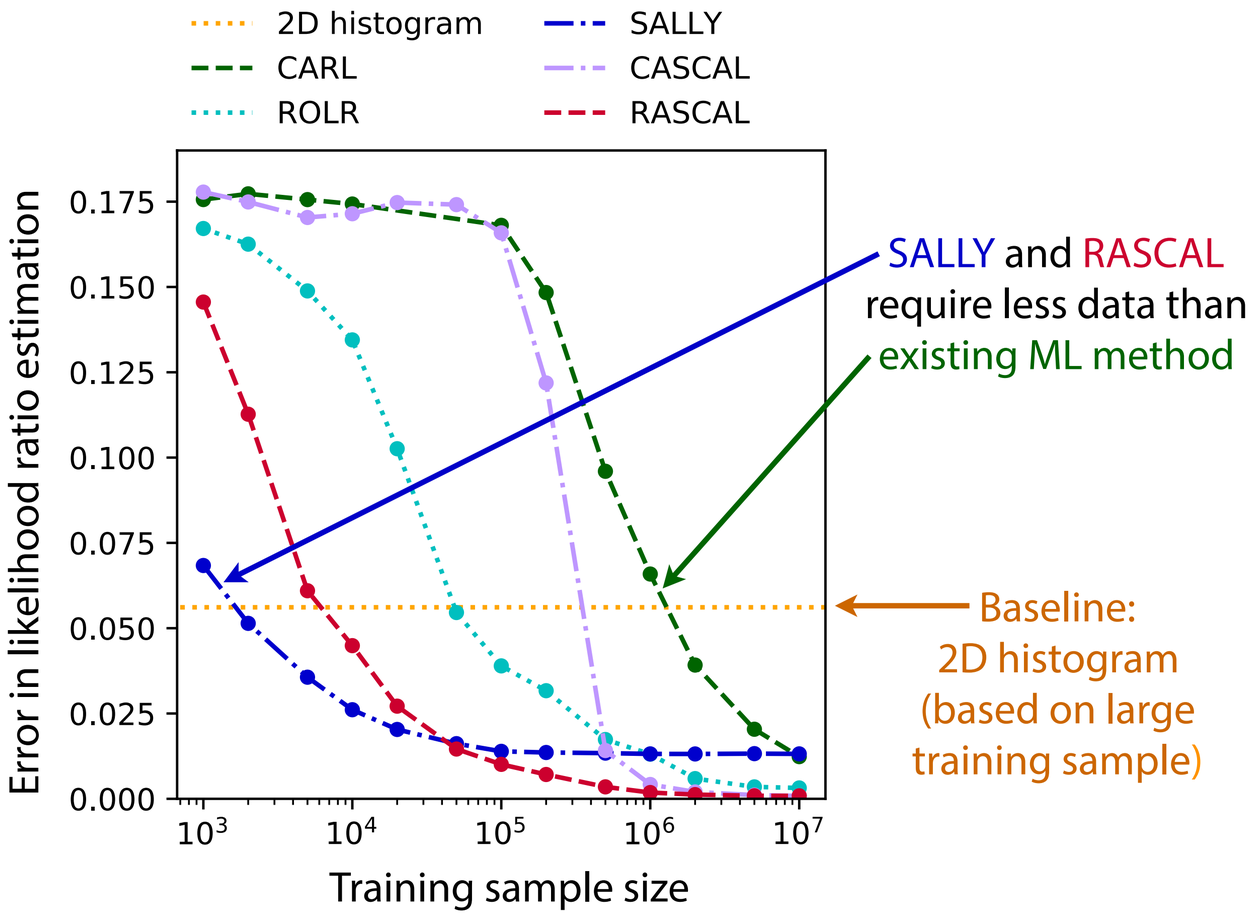
Goal: constrain the two EFT parameters

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

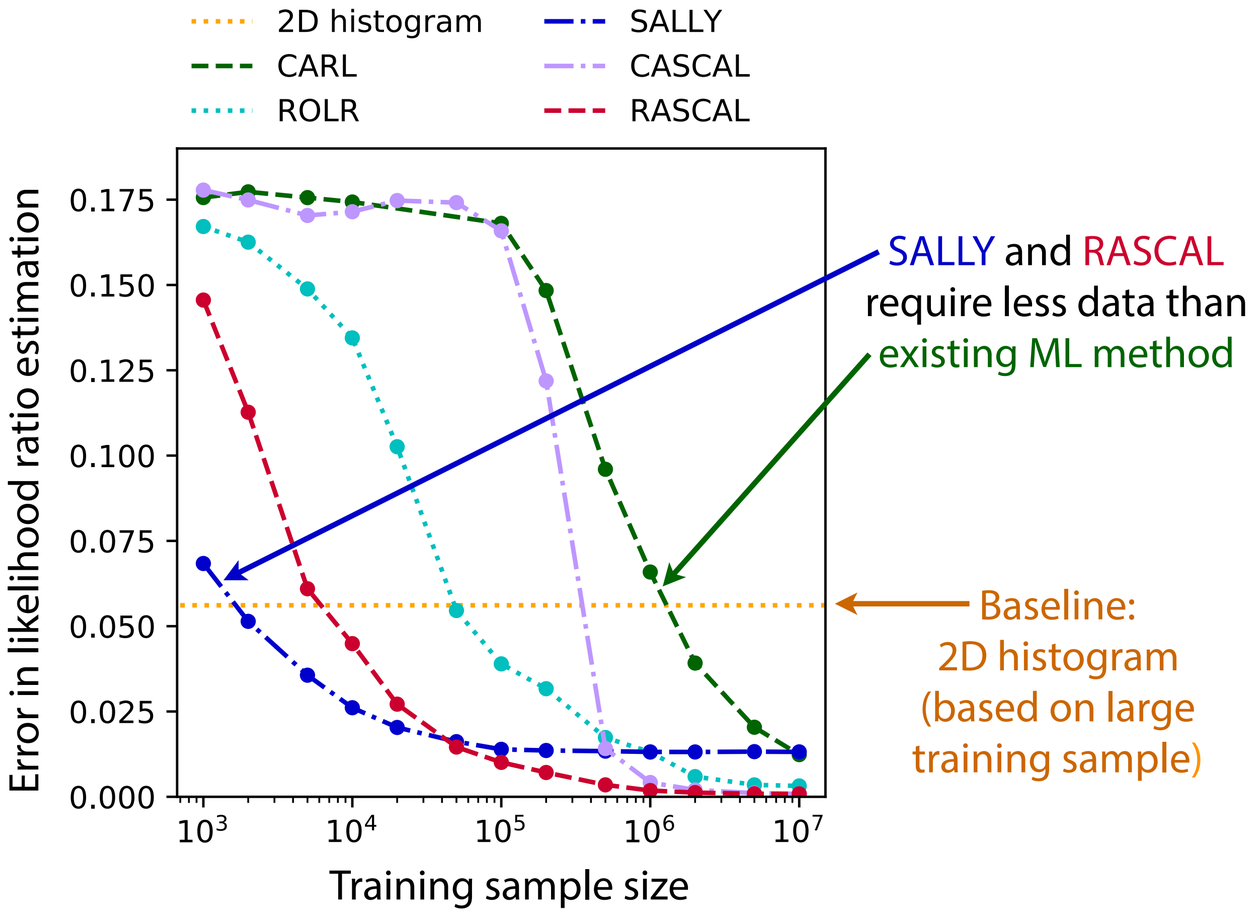
Two scenarios:

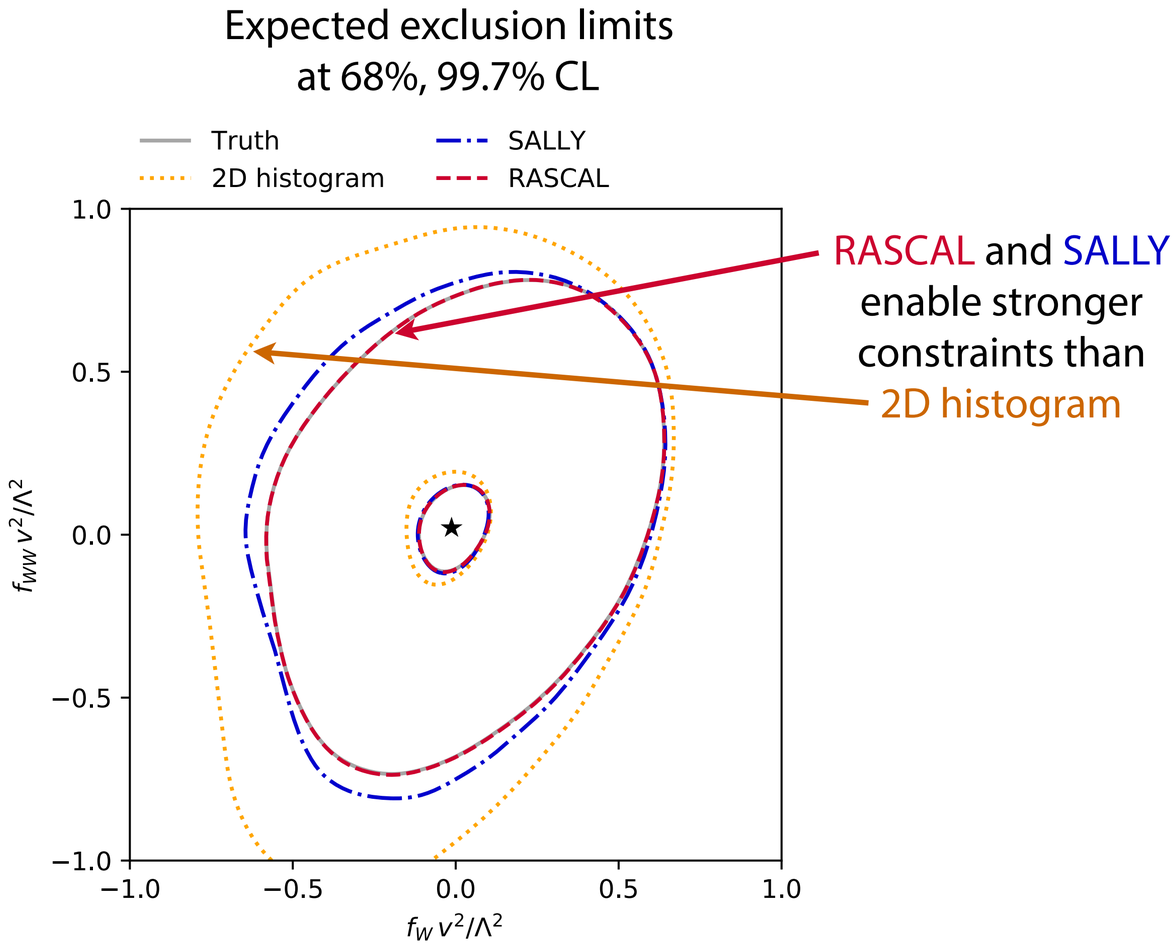
- Simplified setup in which we can compare to true likelihood
- "Realistic" simulation with approximate detector effects

Stronger constraints with less training data

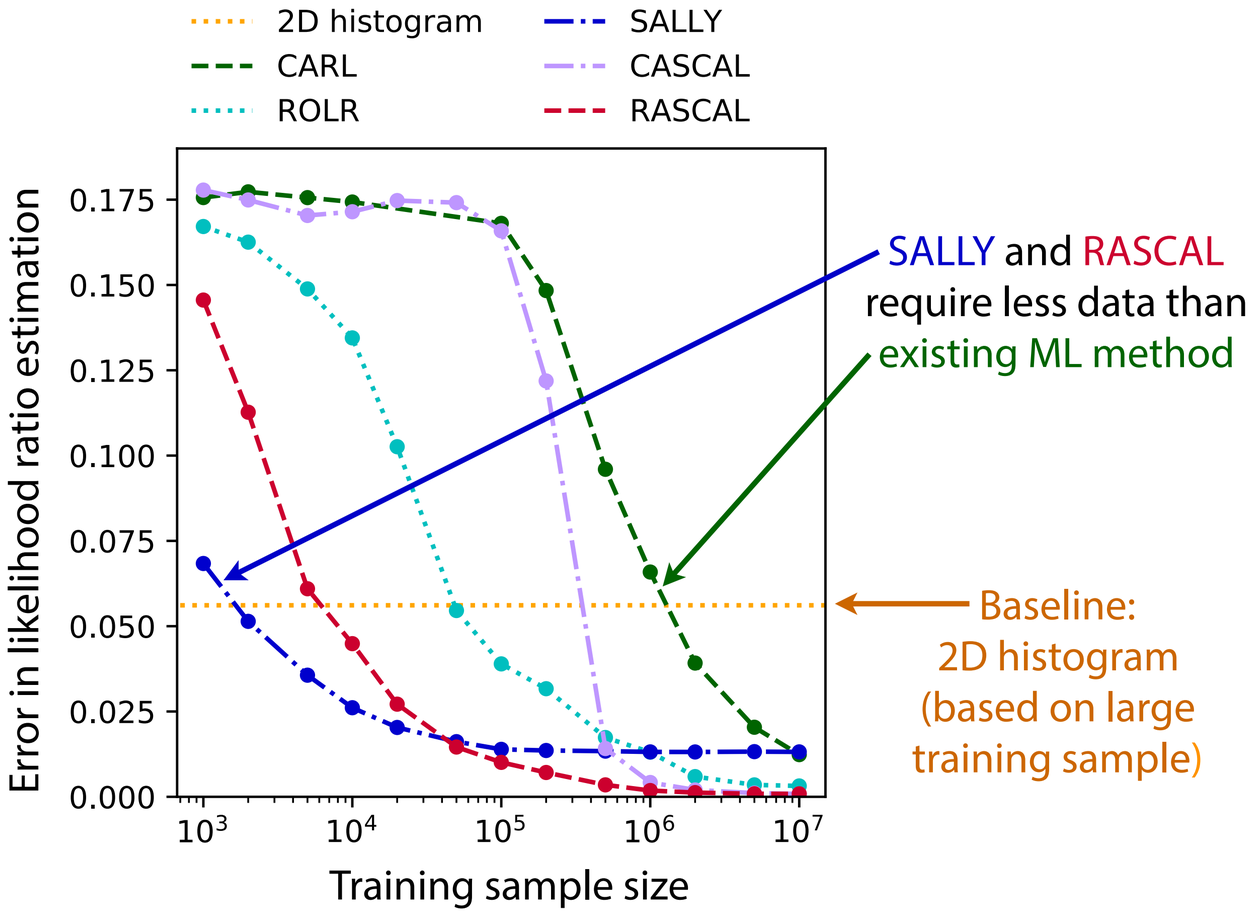


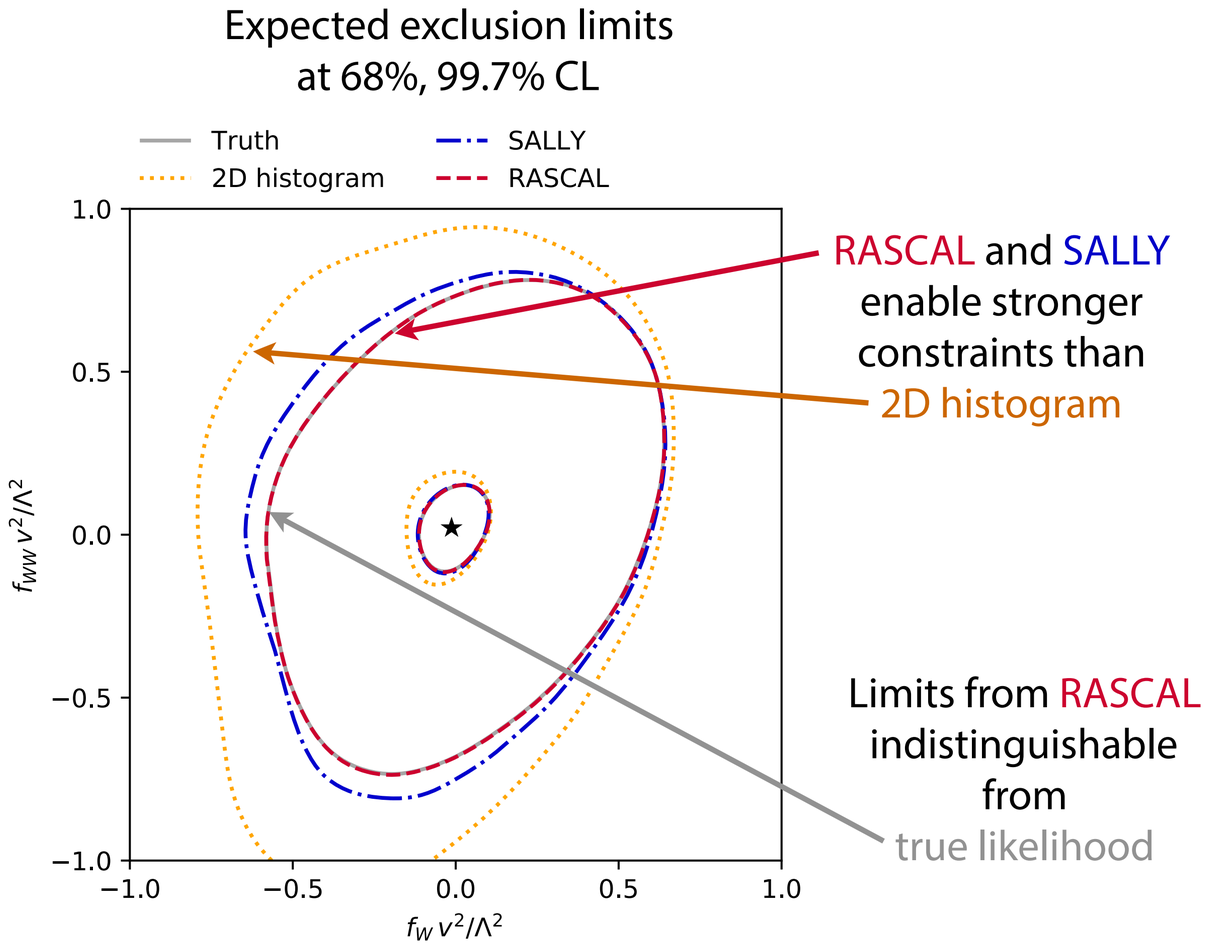
Stronger constraints with less training data





Stronger constraints with less training data

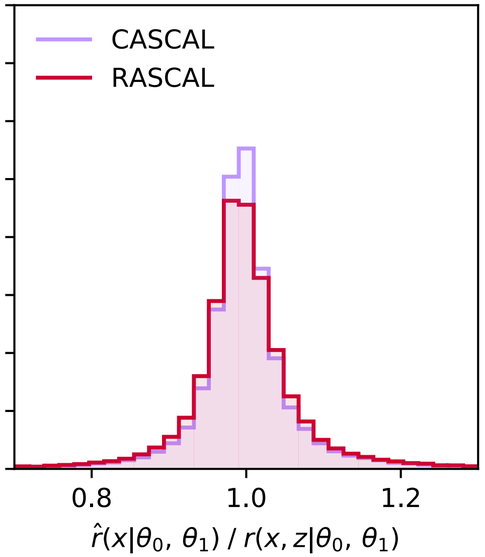




Why / when does it work?

Particle physics:

- it is ~easy to calculate joint likelihood ratio / score (many parts cancel)
- Joint probability distributions $p(x, z|\theta)$ overlap for different θ
- Variance of $r(x, z|\theta_0, \theta_1)$ around $r(x|\theta_0, \theta_1)$ (and of $t(x, z|\theta_0, \theta_1)$ around $t(x|\theta_0, \theta_1)$) turns out to be small:

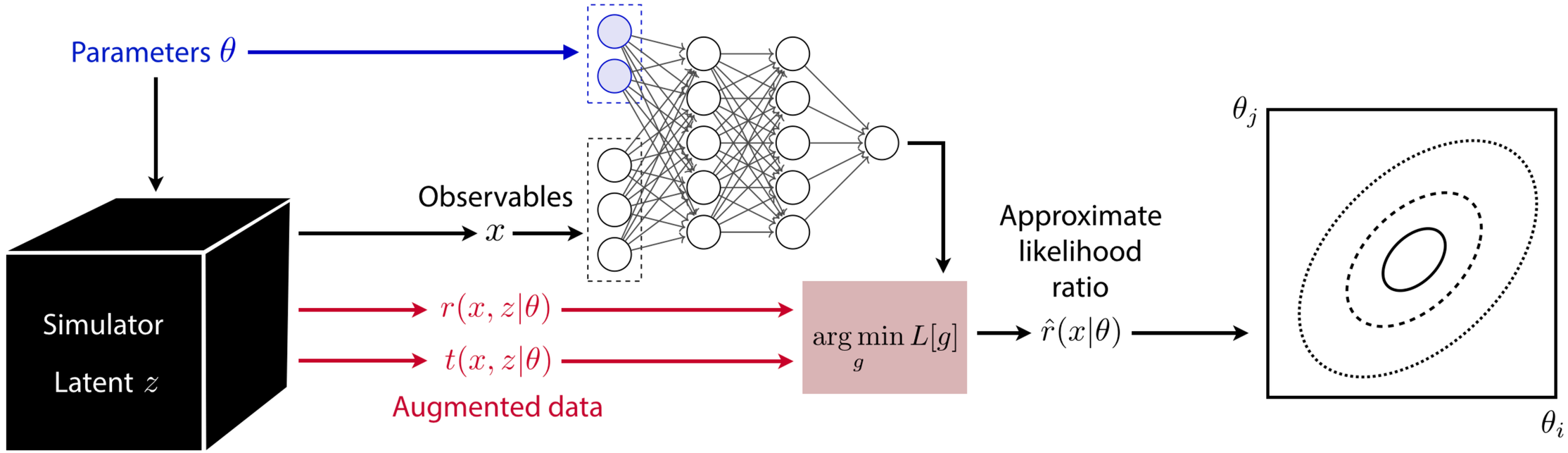


General simulators:

- Calculating joint likelihood ratio and score may require small changes to simulator code (somewhat similar to prob prog)
- Joint probability distributions might not overlap as much
- Variance of joint quantities might be large

⇒ Lots of questions (and some ideas) to investigate!

A new approach to simulator-based inference



When we run simulators, we can extract additional information that characterizes the latent process.

We can train surrogate models based on neural networks on this data to

- estimate the likelihood (ratio)
- learn optimal summary statistics

In particle physics problems, this lets us substantially improve sample efficiency and inference quality.

What about other fields?

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, FK, IE, KC: MadMiner: An inference toolkit for particle physics [doi: 10.5281/zenodo.1489147]

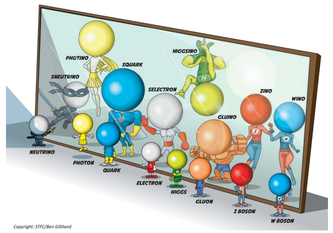
Bonus material

[STFC / Ben Gililand, Sean Carroll, Friedrich Justin Bertuch 1806, symmetry]

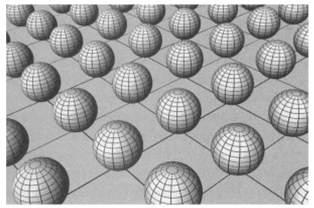
Effective field theory

Energy

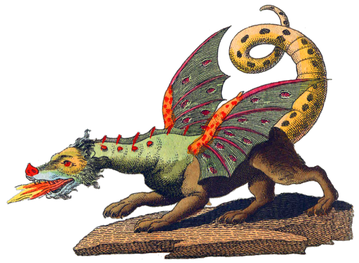
Λ



Supersymmetry?



Extra dimensions?



Dragons?

Composite Higgs?

…

Matching

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

Effective field theory

Measurements

LHC Higgs

Other LHC measurements

LEP

Flavour

…

$E \ll \Lambda$

SMEFT (Standard Model Effective Field Theory)

[W. Buchmuller, D. Wyler 85;
B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek 1008.4884; …]

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

Operators

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

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

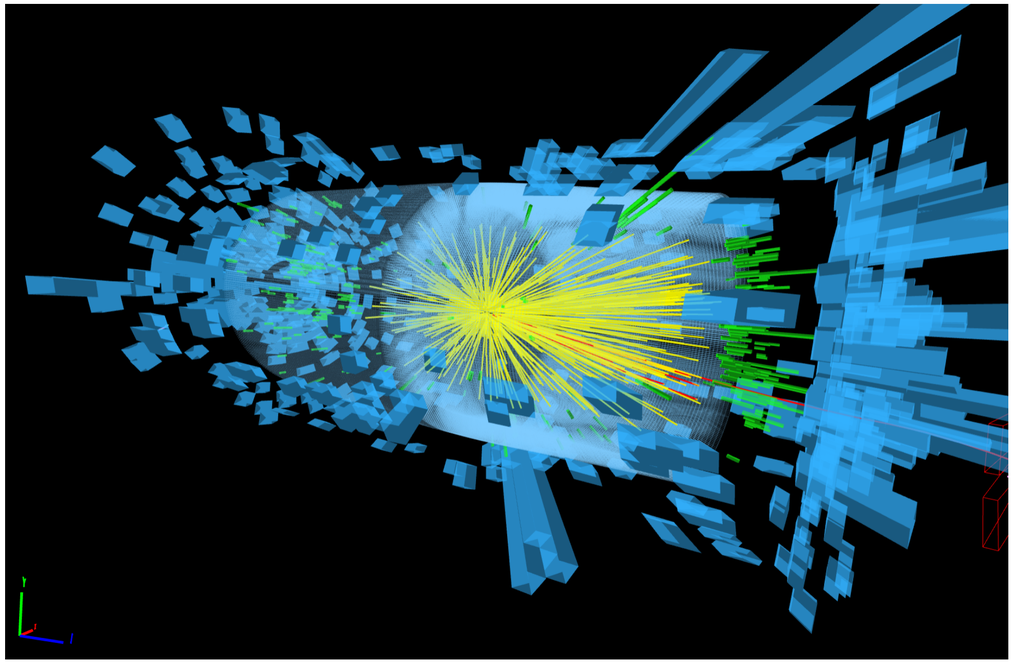
Wilson coefficients

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

Higher-order terms

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

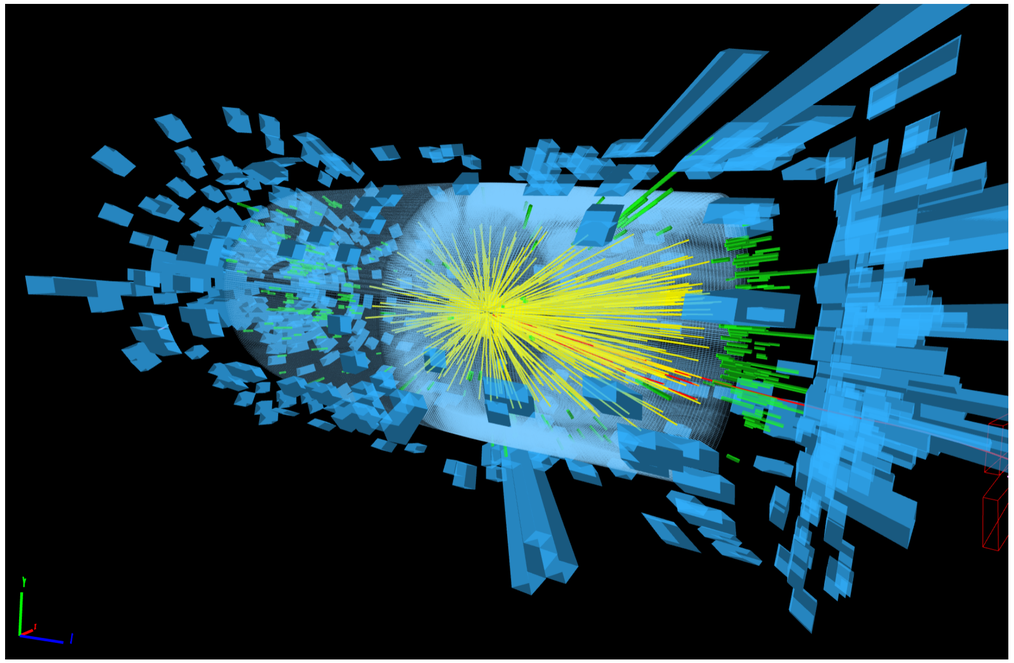
Solve it with summary statistics

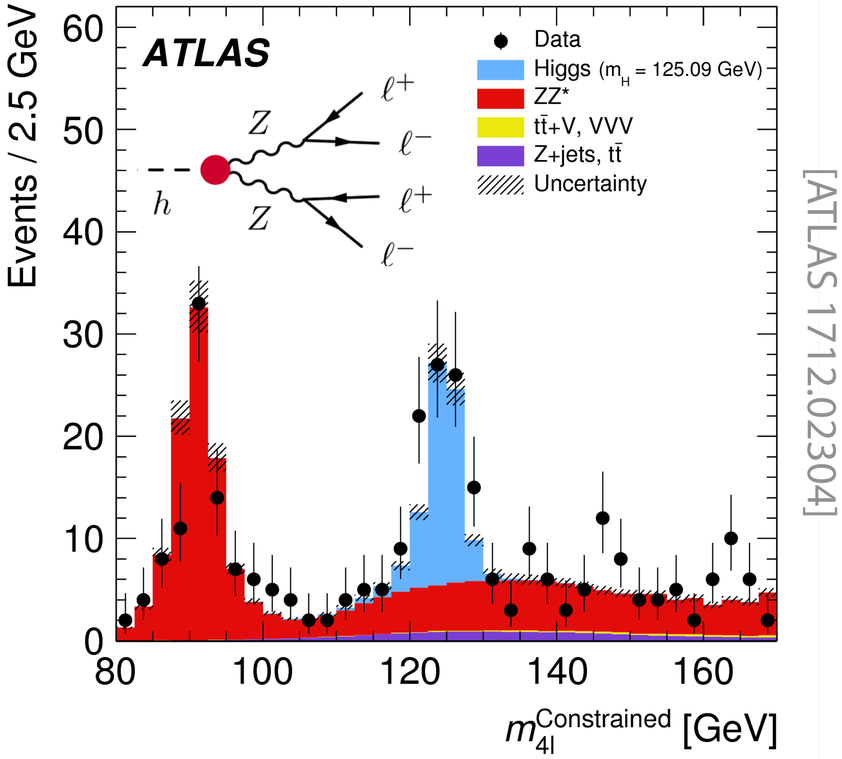


High-dimensional event data x

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

Solve it with summary statistics





High-dimensional event data x

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

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]

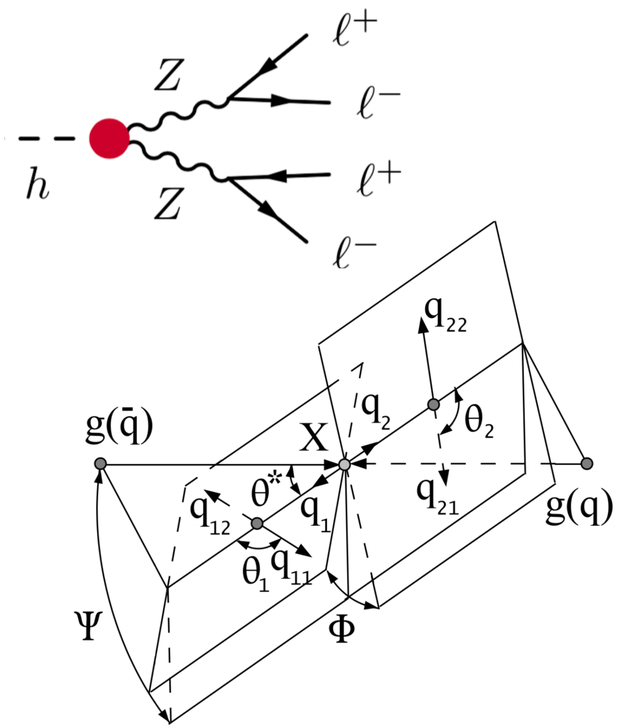
- Ideally: analyze high-dimensional x including all correlations
  ("fully differential cross section")

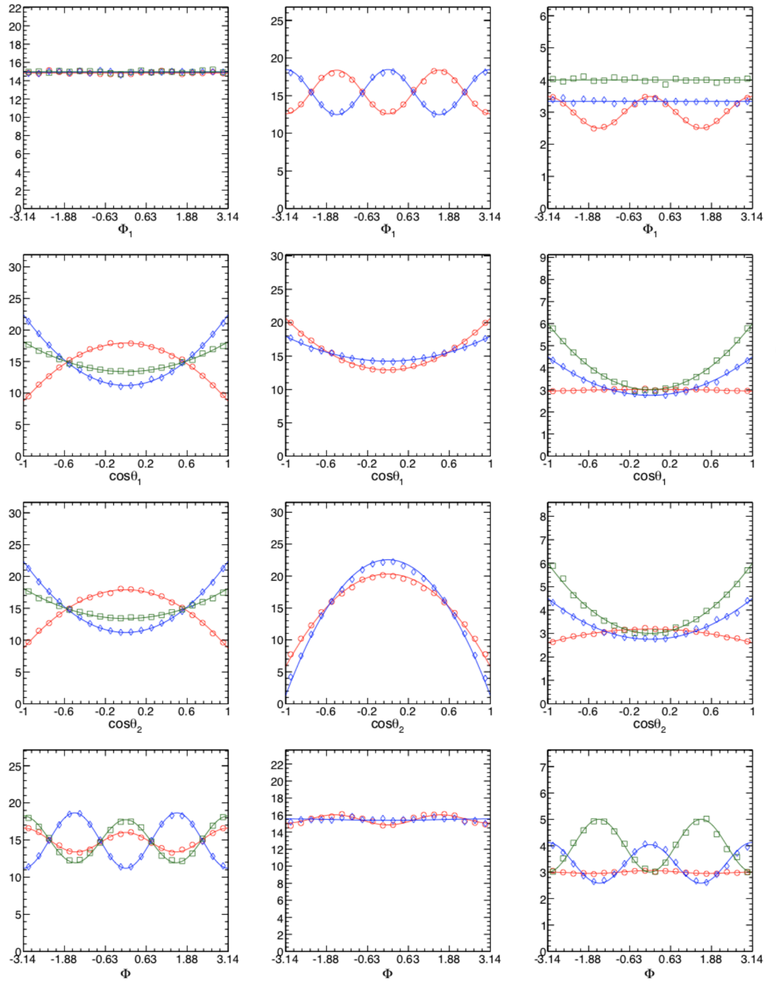
Summary statistics for EFT measurements?

- 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")

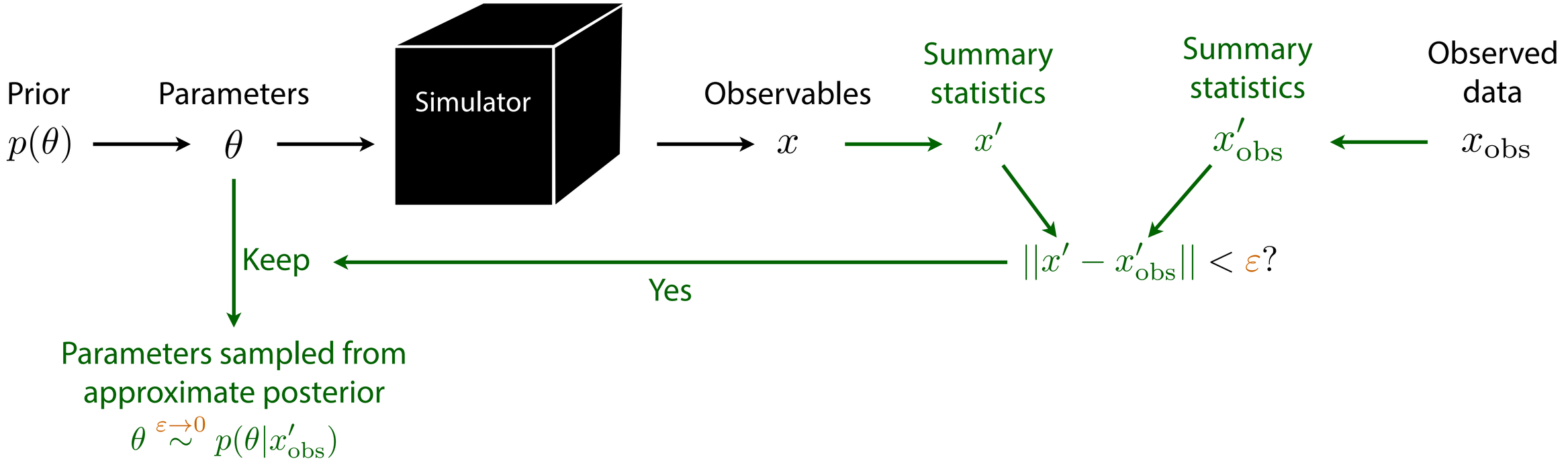




[Bolognesi et al. 1208.4018]

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

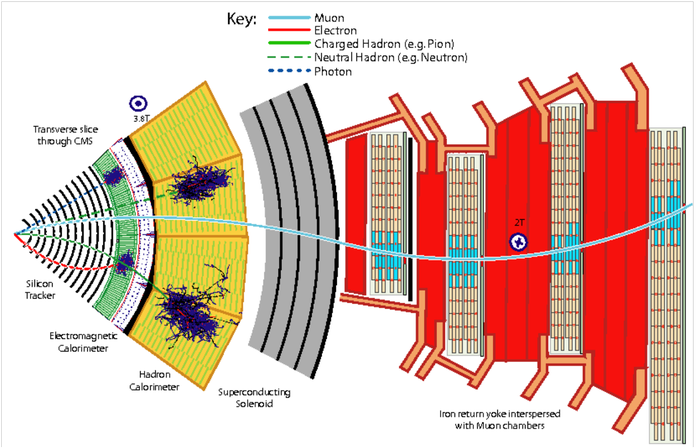
"Curse of dimensionality"

Precision vs efficiency tradeoff

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]

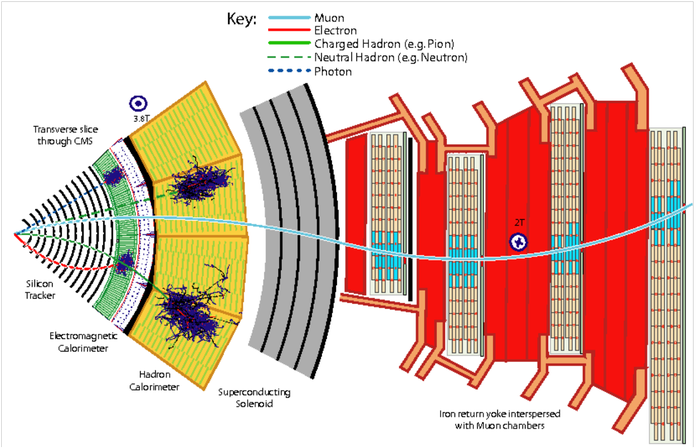
Solve it by approximating the integral

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

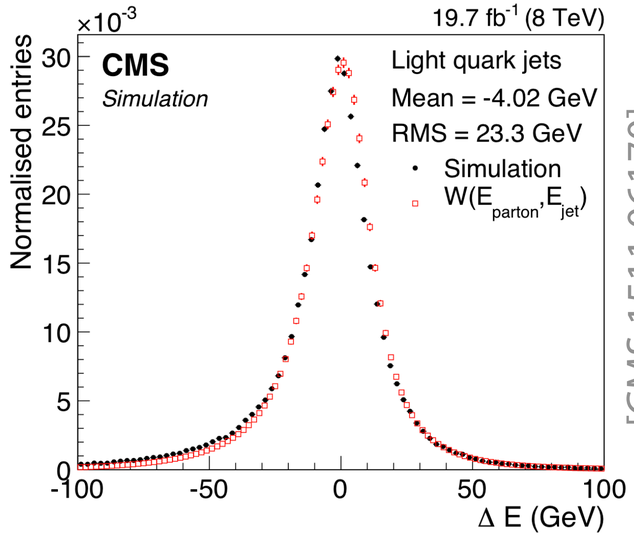
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$

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

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



[CMS]

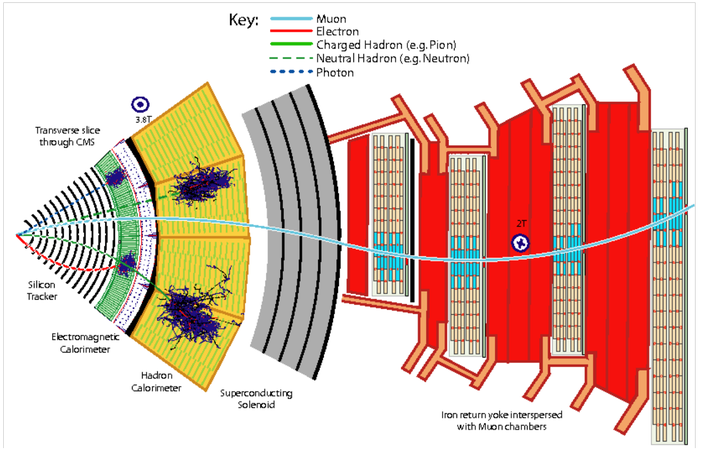


[CMS 1511.06170]

Solve it by approximating the integral

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

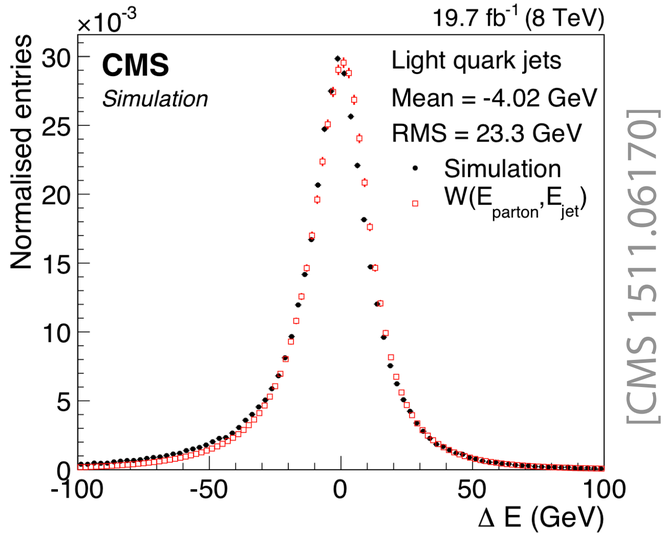
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$



[CMS]

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

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



[CMS 1511.06170]

⇒ 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]

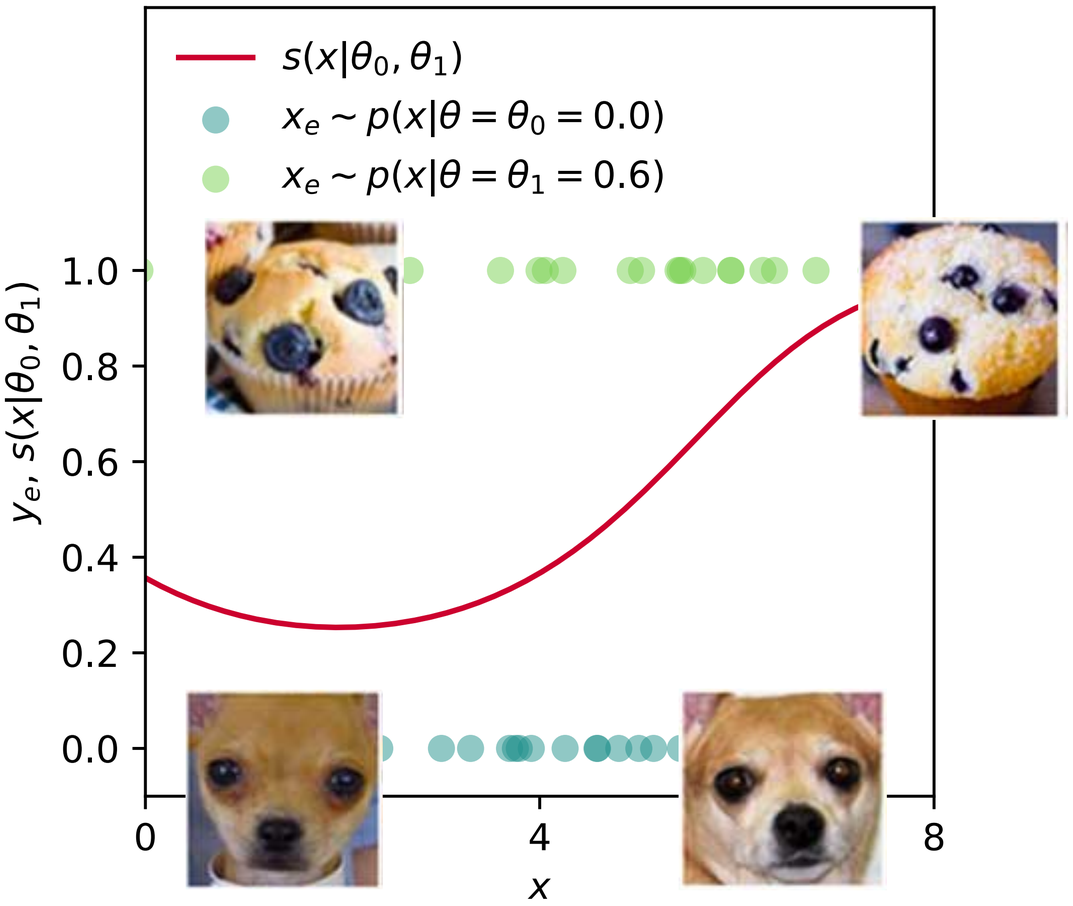
[M. Yao, idea for analogy: K. Cranmer]

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]

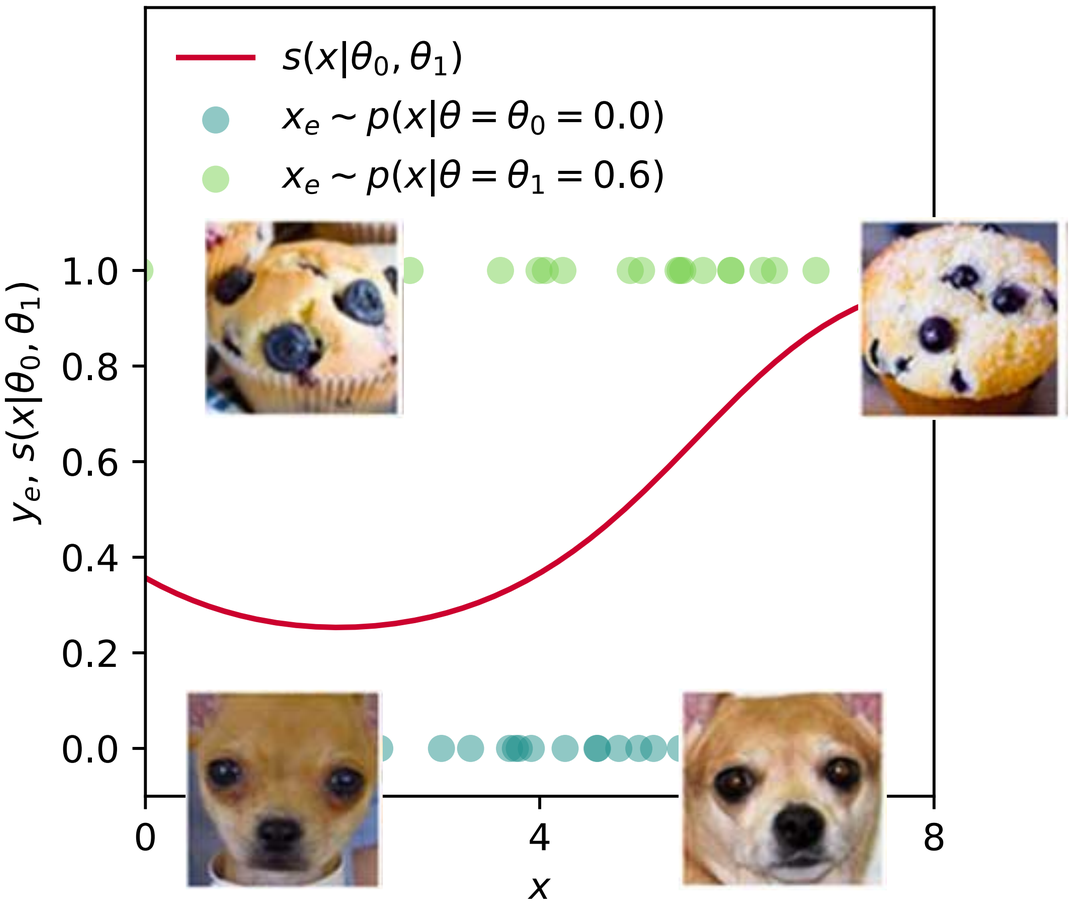
- Train neural network (BDT, …) to tell $x \sim p(x|\theta_0)$ from $x \sim p(x|\theta_1)$

- 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)



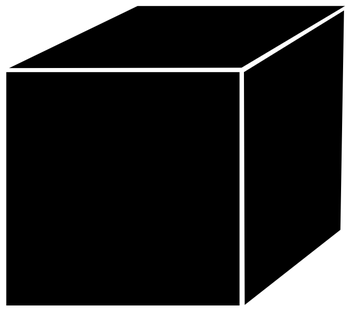
⇒ No summary statistics necessary, very fast evaluation… but may require large training samples

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, …

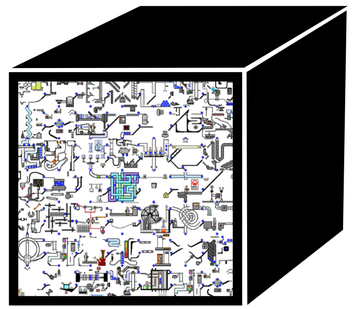


Use latent structure:

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

New!

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



Variational calculus

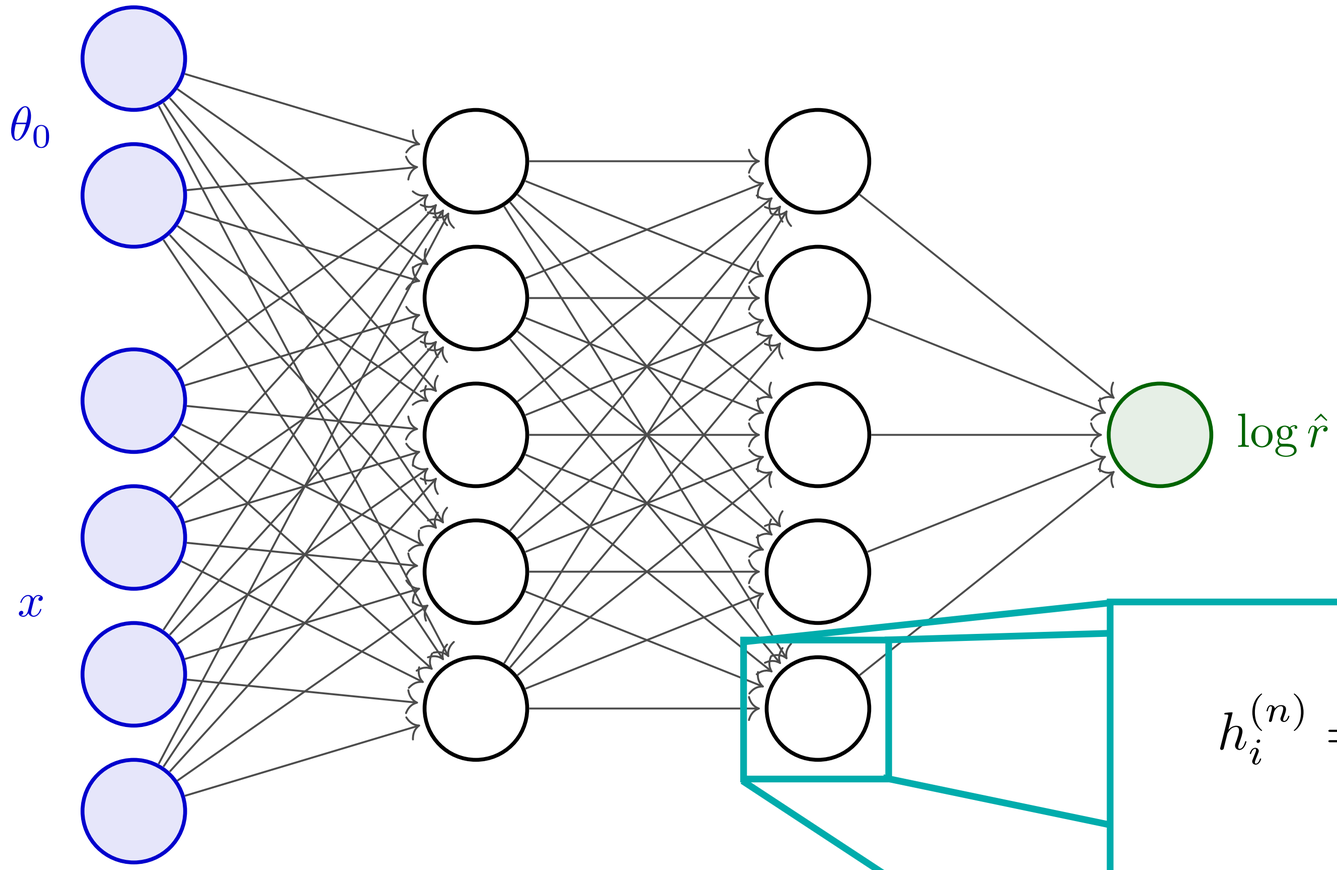
$$L[\hat{g}(x)] = \int \mathrm{d}x\, \mathrm{d}z\; p(x,z|\theta)\; |g(x,z) - \hat{g}(x)|^2$$

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

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

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

Neural networks = universal function approximators



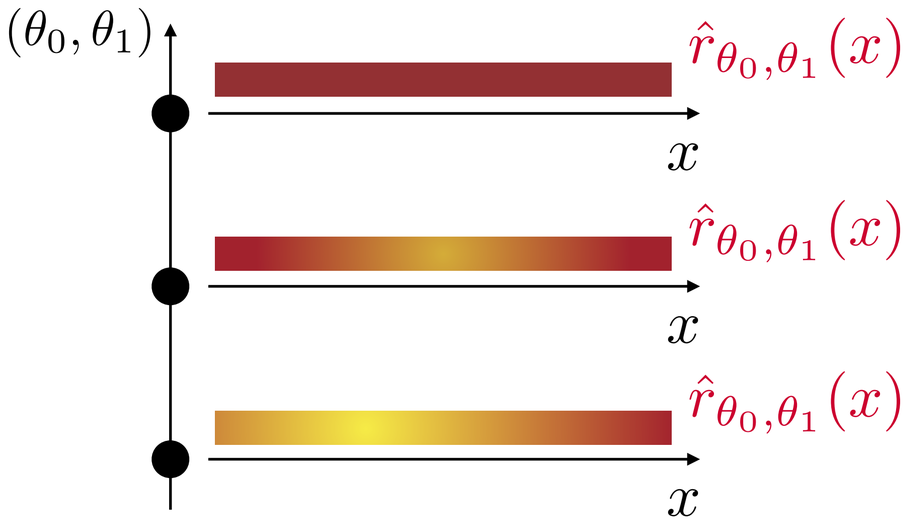
$$h_i^{(n)} = f\left(\sum_k w_{ik}^{(n)}\, h_k^{(n-1)} + b_i^{(n)}\right)$$

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

Two types of likelihood ratio estimators

A) Point by point:

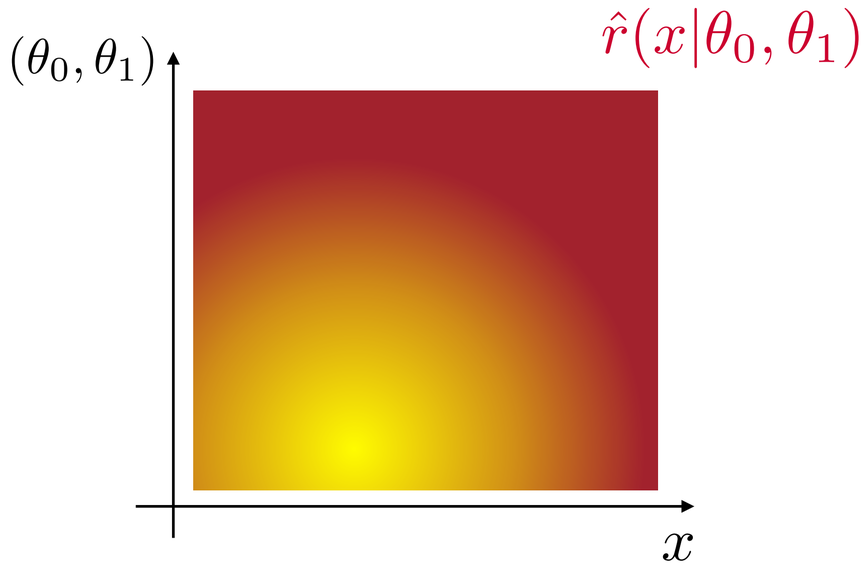
- first, define grid of parameter points $\{(\theta_0, \theta_1)\}$
- for each combination (θ_0, θ_1) ,
  create separate estimator $\hat{r}_{\theta_0,\theta_1}(x)$
- final results can be interpolated between grid points



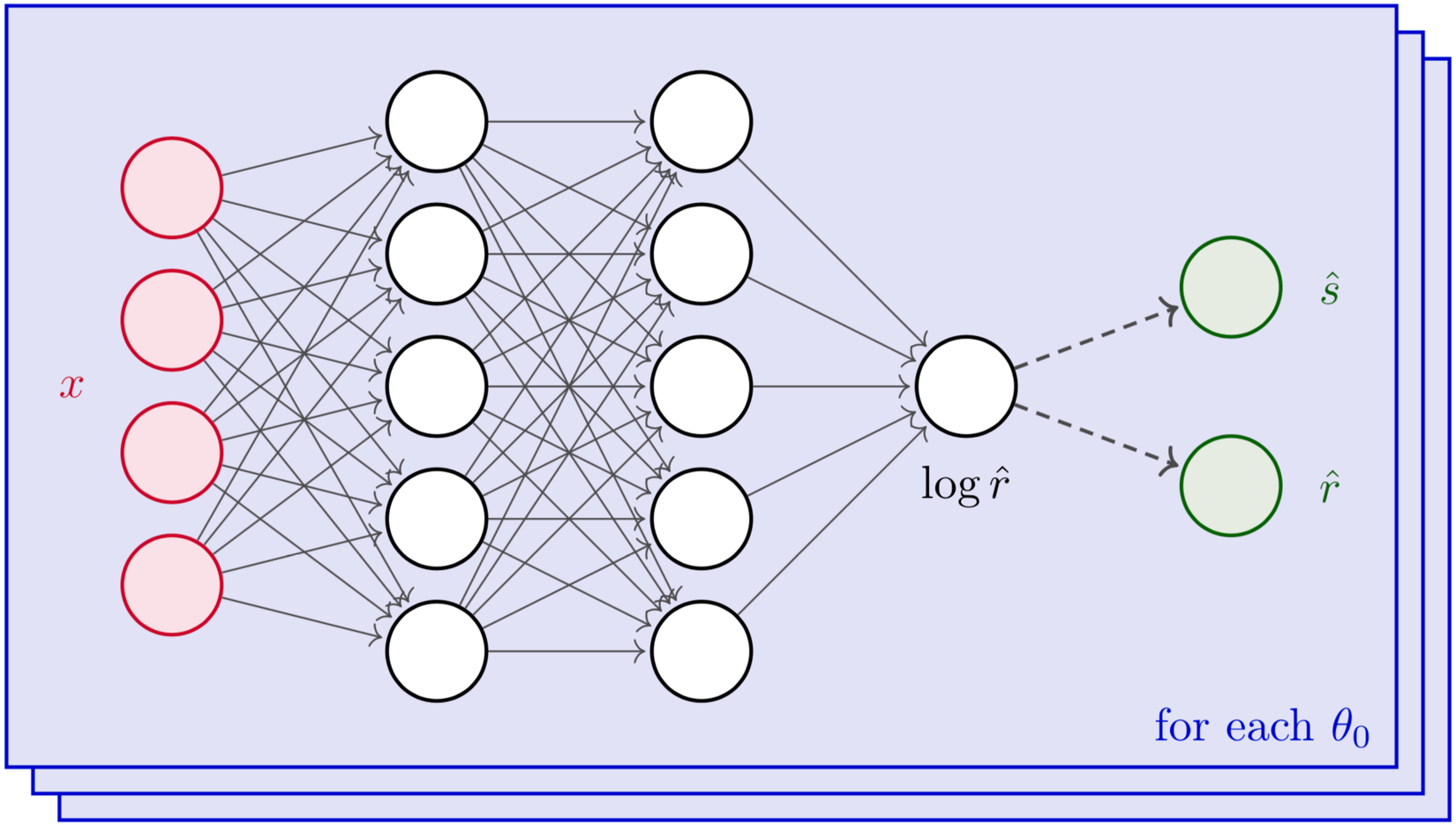
B) Parameterized:

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

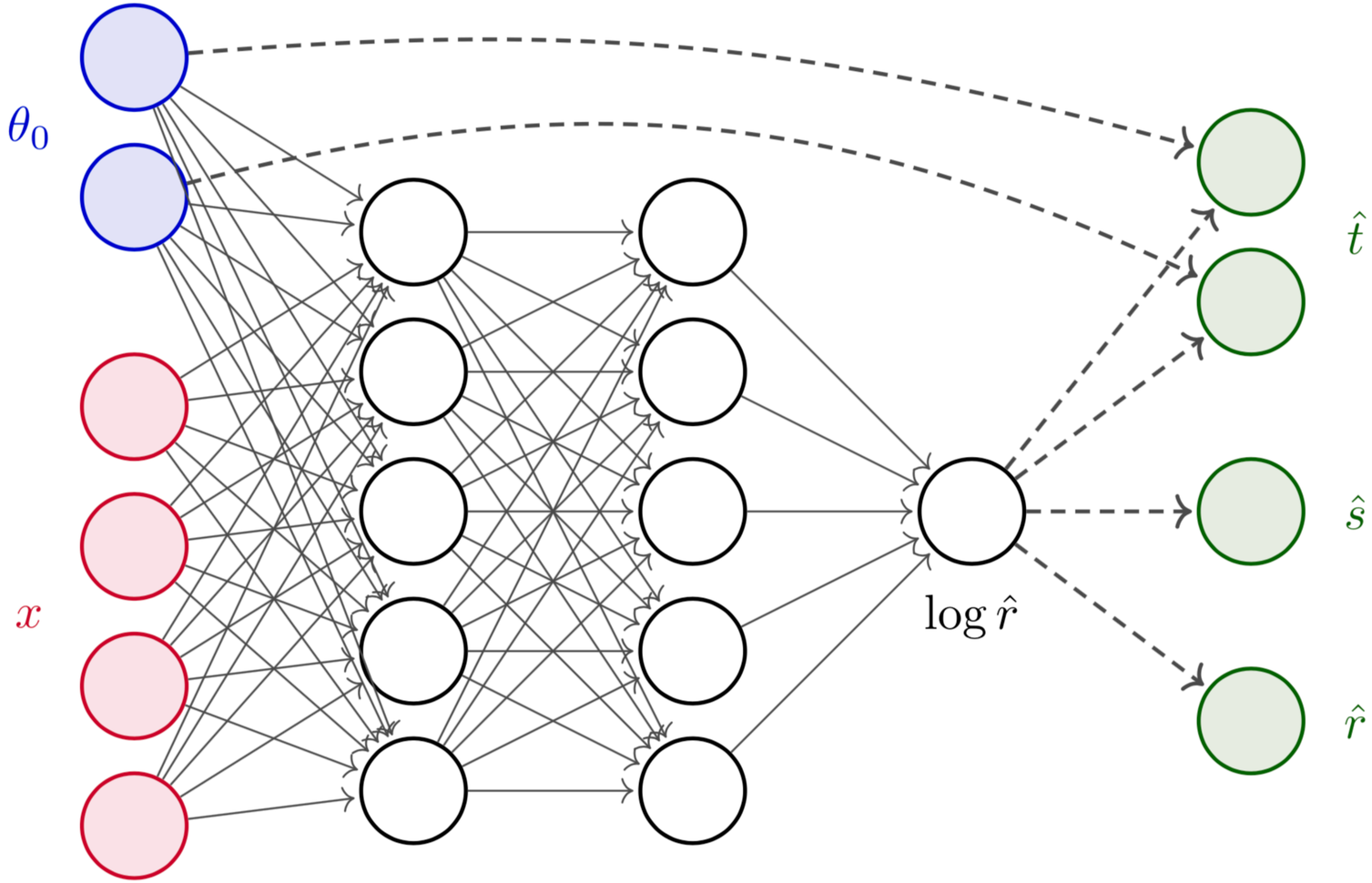
- create one estimator $\hat{r}(x|\theta_0, \theta_1)$
  that is a function of θ_0 and θ_1
- no further interpolation necessary
- "borrows information" from close points



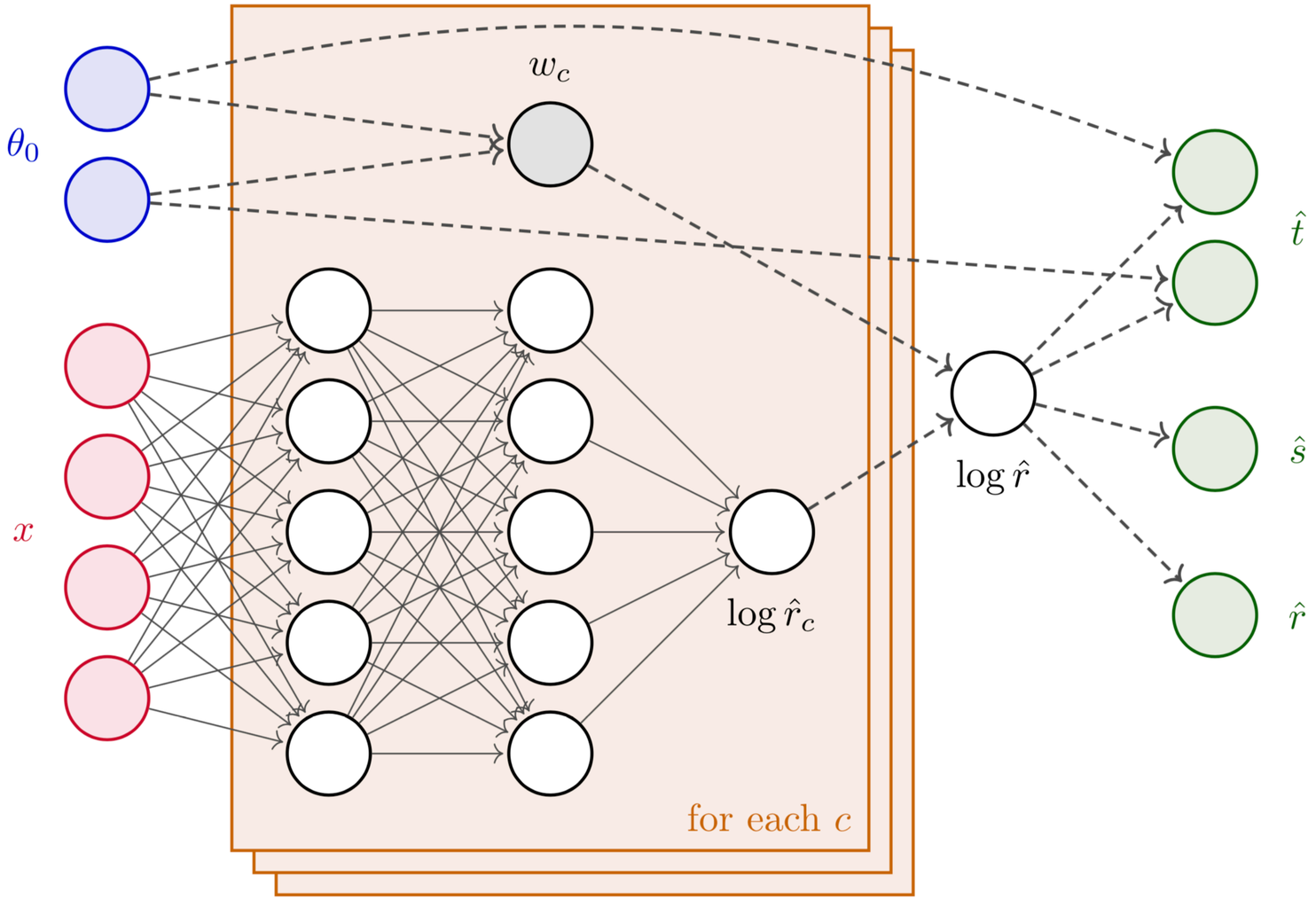
Point by point



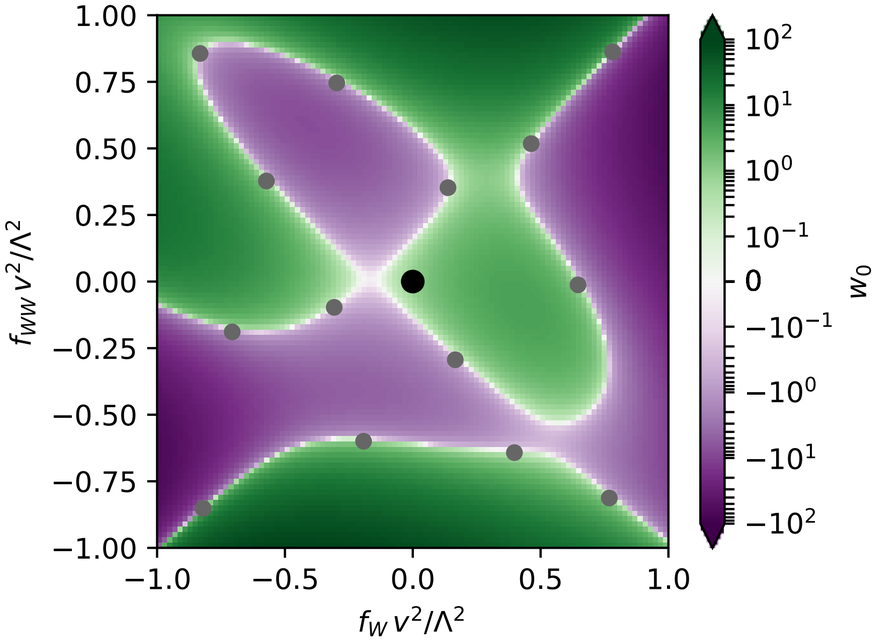
(Agnostic) parameterized estimators



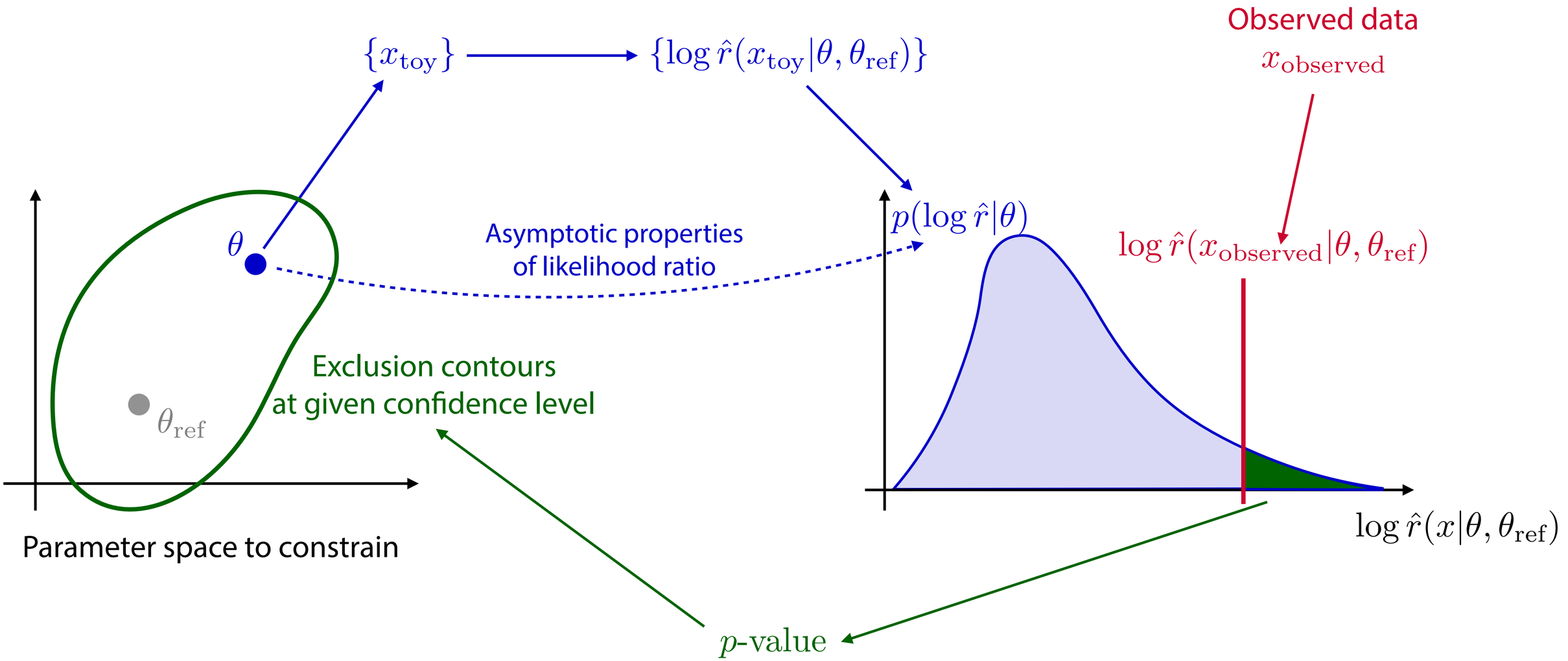
Morphing-aware parameterized estimators



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



Limit setting (frequentist)



The likelihood ratio is the most powerful test statistic

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

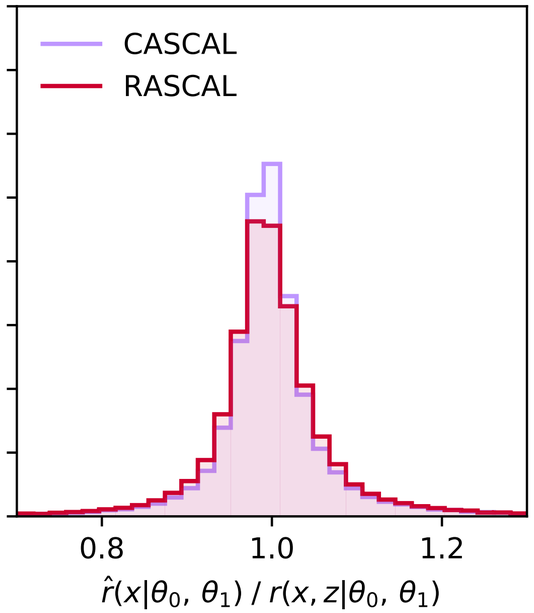
By J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

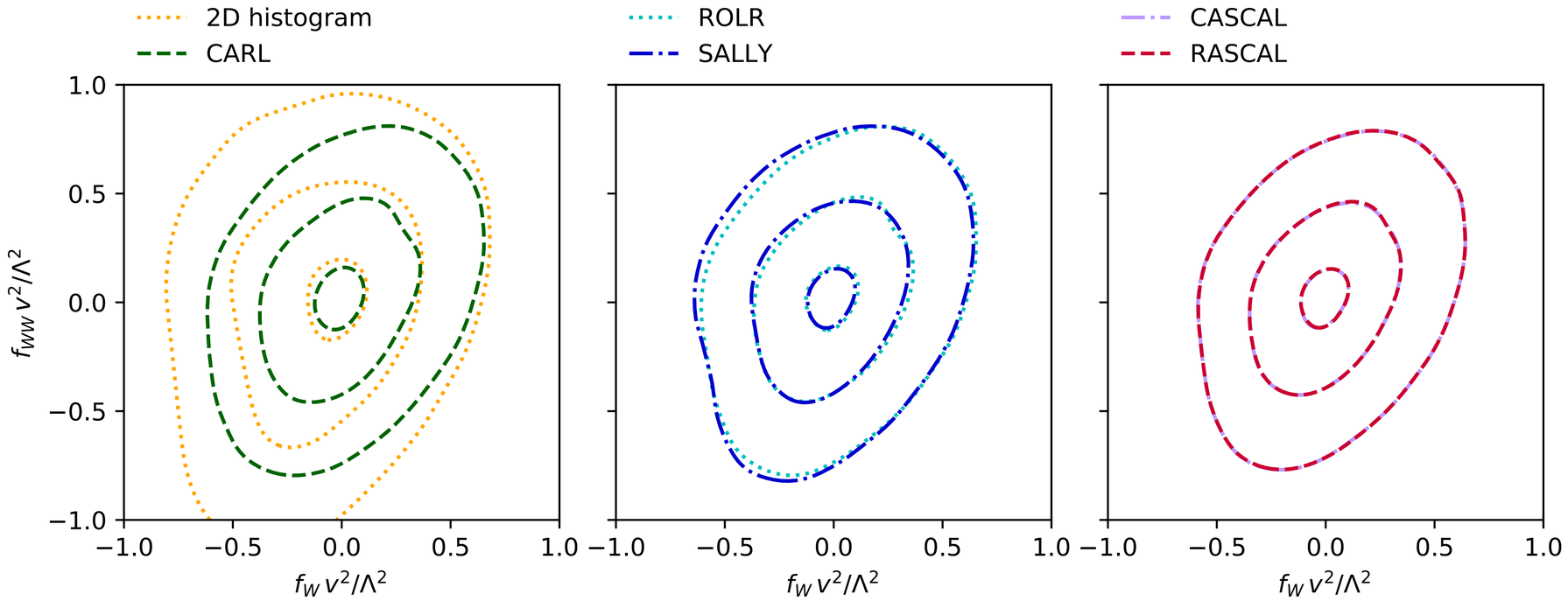
(*Communicated by* K. PEARSON, *F.R.S.*)

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

CONTENTS.

WBF Higgs to four leptons with detector effects



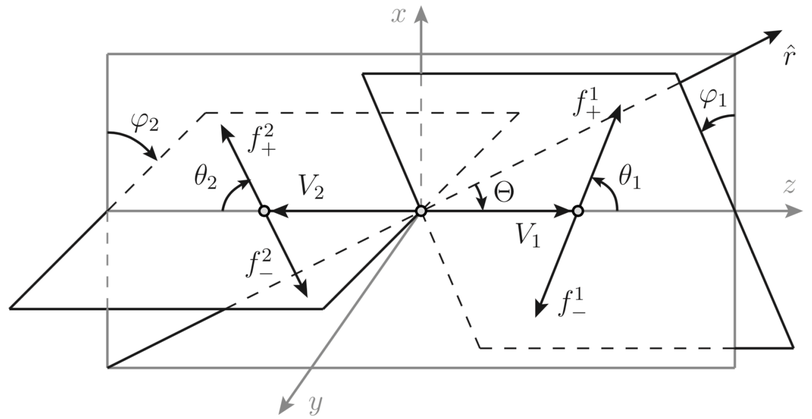


Diboson production

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

  ⇒ Reduced sensitivity to new physics

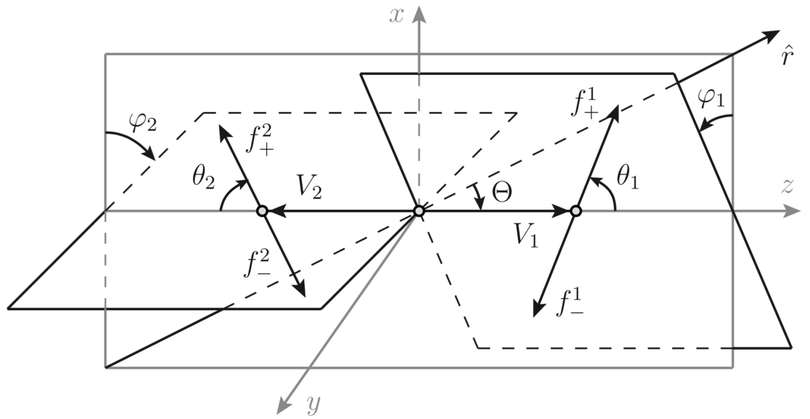
- “Diboson interference resurrection”:
  an angular variable φ can be constructed to be sensitive to this interference
  [G. Panico, F. Riva, A. Wulzer 1708.07823]



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]

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

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