Bringing together simulations, physics insight, and machine learning to constrain new physics

Johann Brehmer
New York University

Brandeis University, Dark Universe Seminar
February 28, 2019


Kyle Cranmer


Gilles Louppe


Juan Pavez


Felix Kling

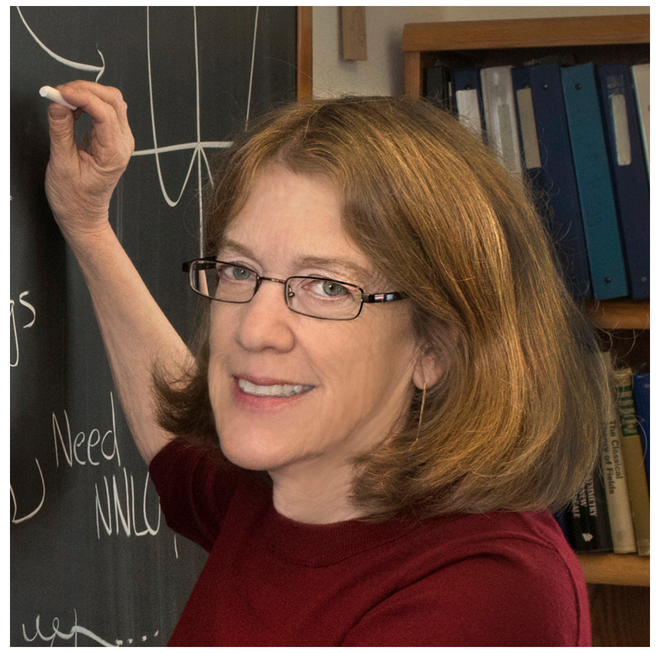

Irina Espejo


Zubair Bhatti


Markus Stoye


Sid Mishra-Sharma


Tilman Plehn


Sally Dawson


Sam Homiller


Josh Rudermann


Duccio Pappadopulo


Marco Farina

NYU

MSDSE
Moore - Sloan
Data Science Environments

dianahep

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

The SCAILFIN Project
scailfin.github.io

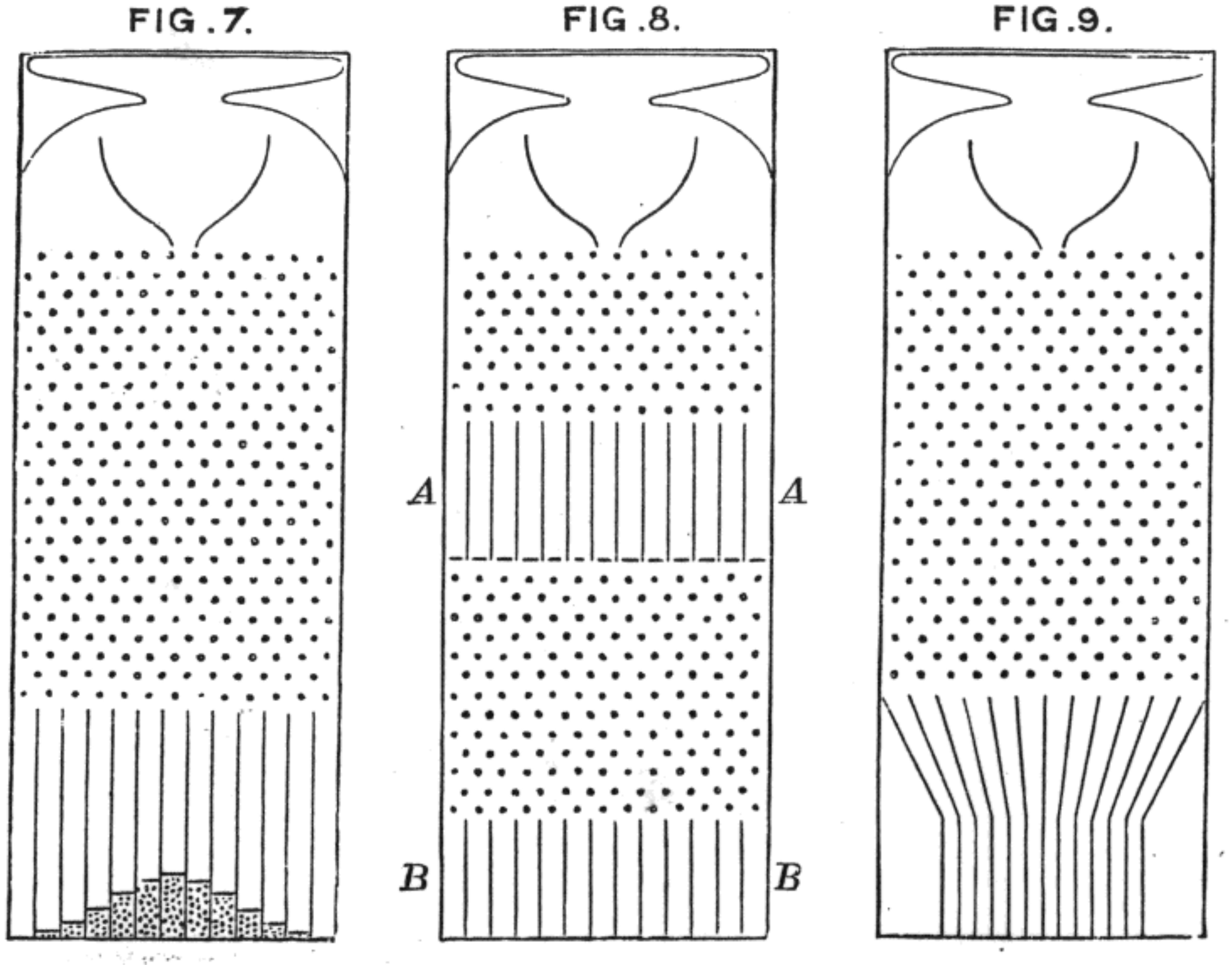

5. Particle physics example
4. Learning optimal summary statistics
3. “Mining gold”: a new approach
2. Existing solutions
1. The problem of
likelihood-free inference

The problem of
“likelihood-free” inference

The Galton board

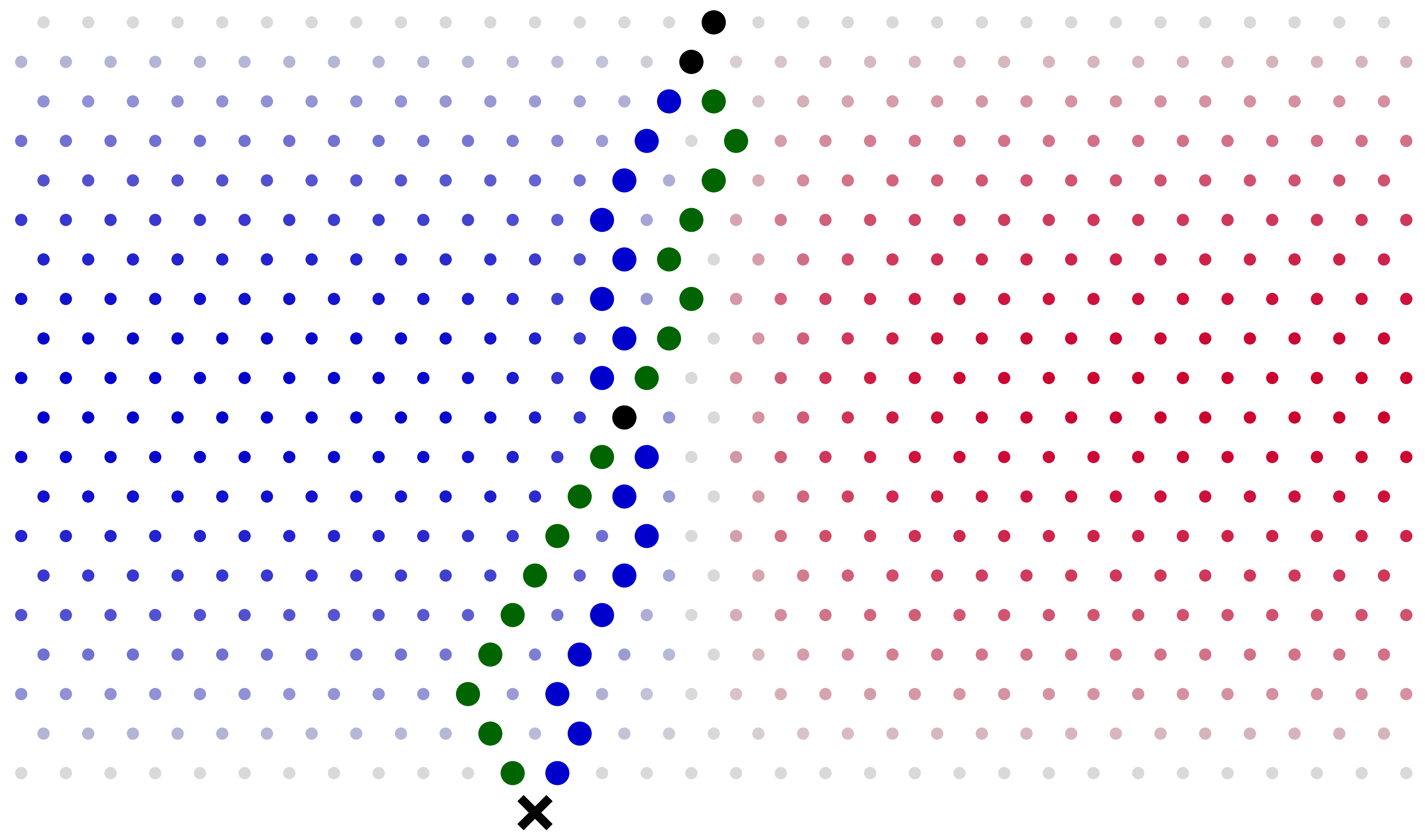
[galtonboard.com]

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!

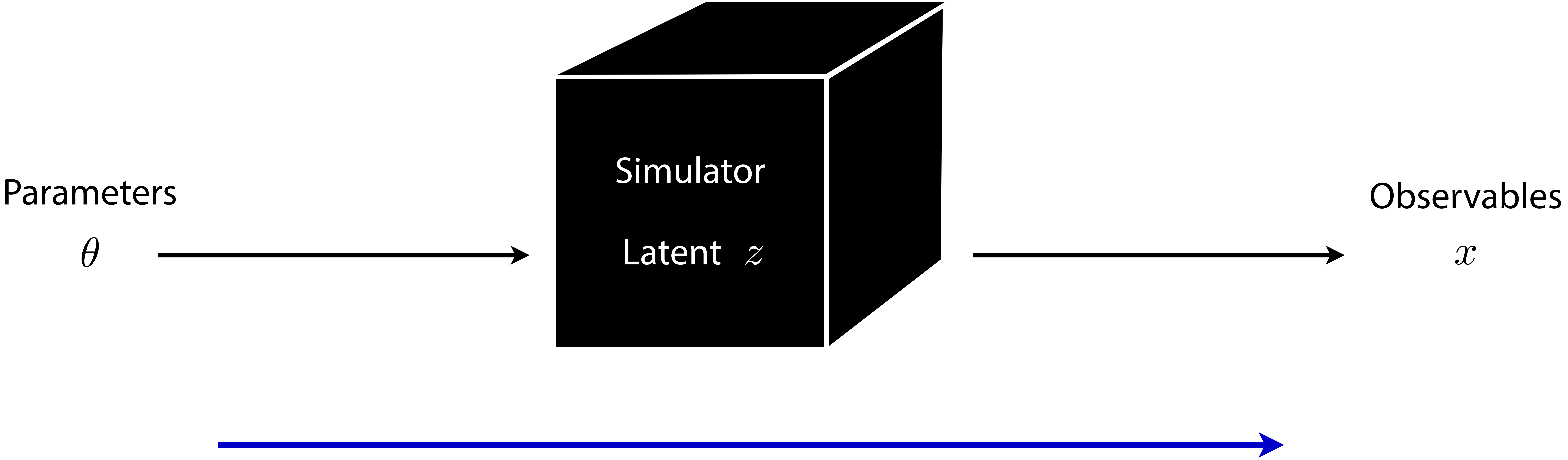
- Inference: given observations $\{x_i\}$, what are the most likely values for θ ?

  Usually we solve this with the 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”

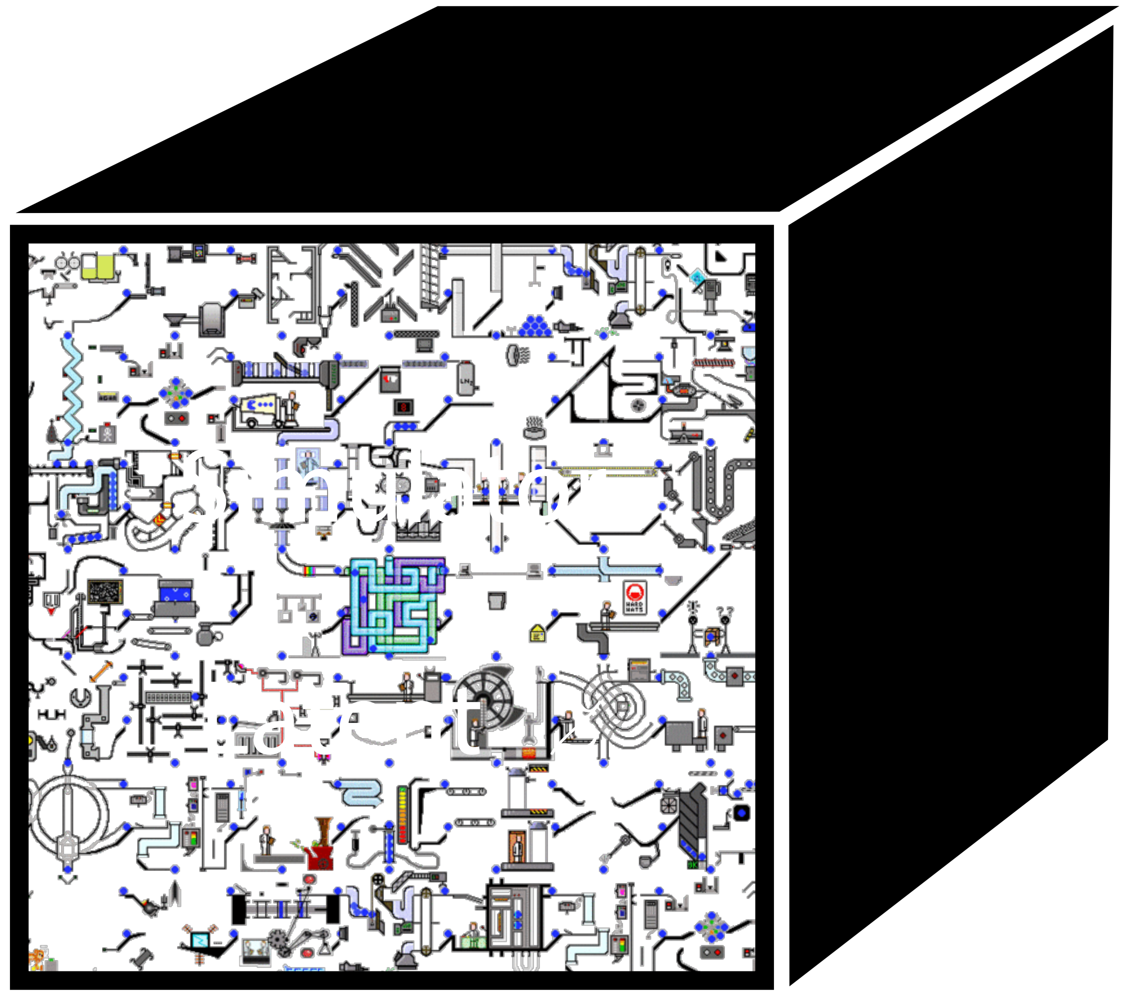


Prediction:

- Well-understood mechanistic model
- Simulator can generate samples

“Likelihood-free inference”

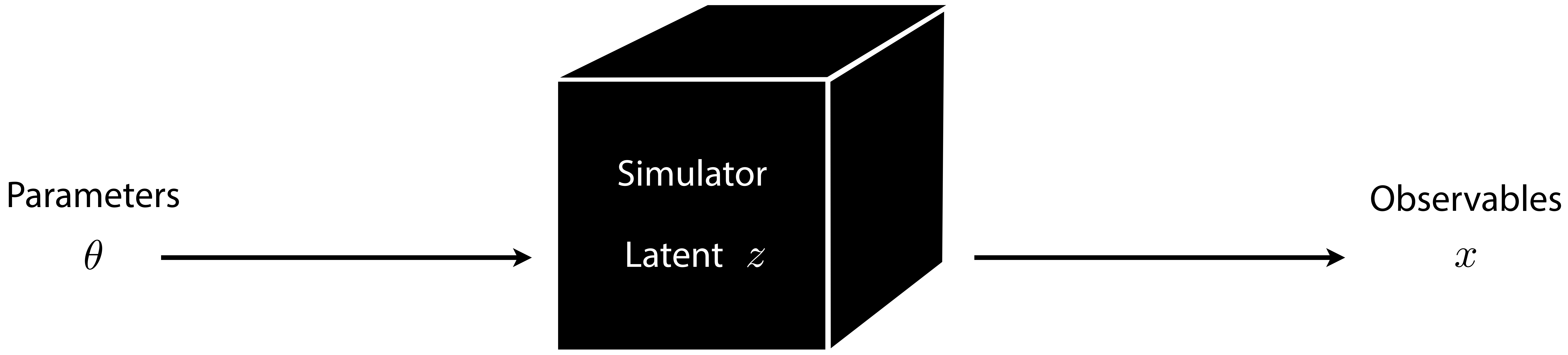
Parameters
θ



Observables
x

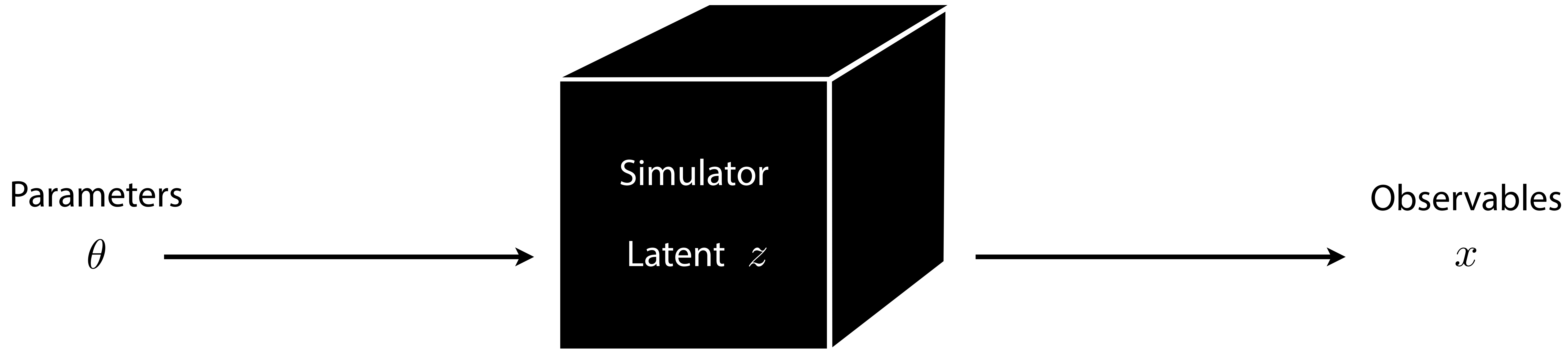
Prediction:
- Well-understood mechanistic model
- Simulator can generate samples

“Likelihood-free inference”



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

“Likelihood-free inference”



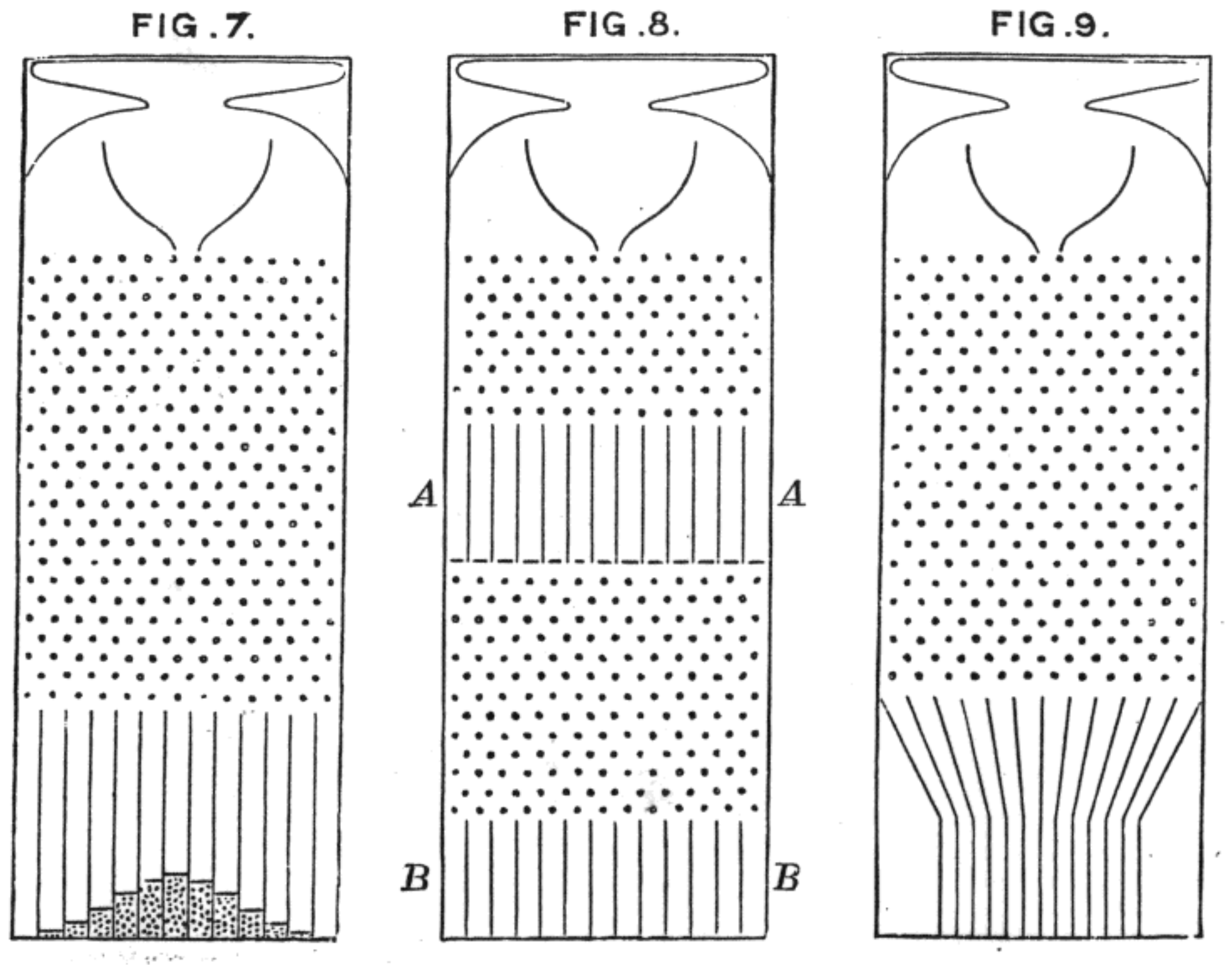
Prediction:

- Well-understood mechanistic model
- Simulator can generate samples

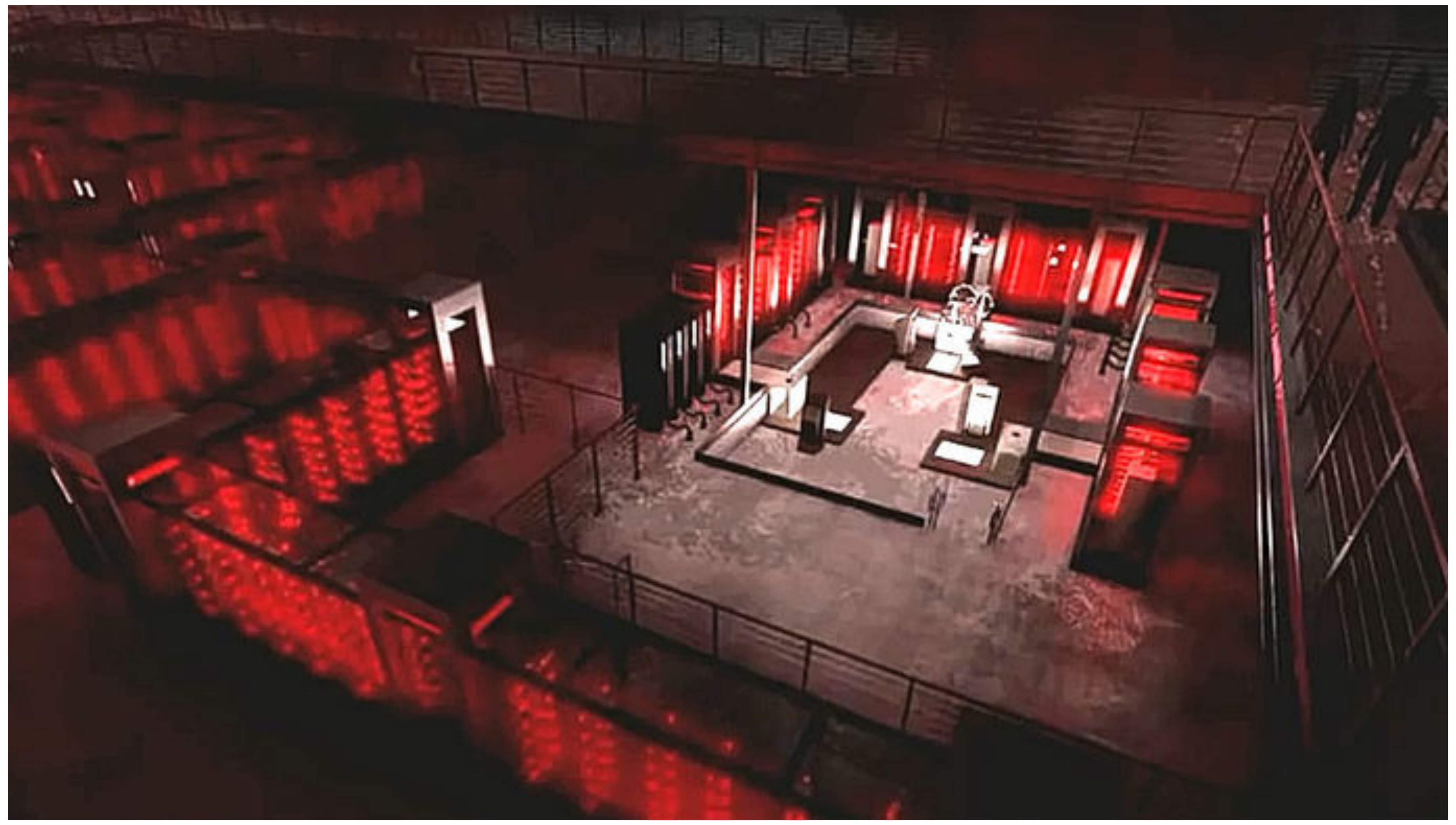
Inference:

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

Galton board: metaphor for simulator-based science



[F. Galton 1889]

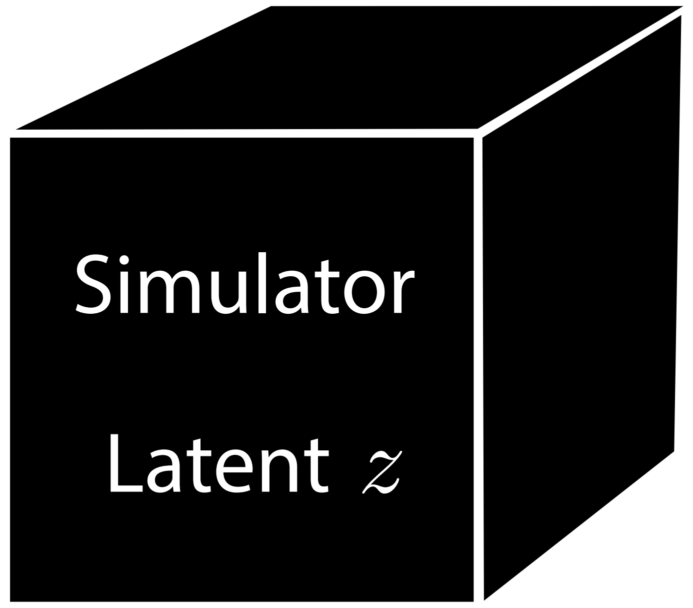


[HBO 2018]

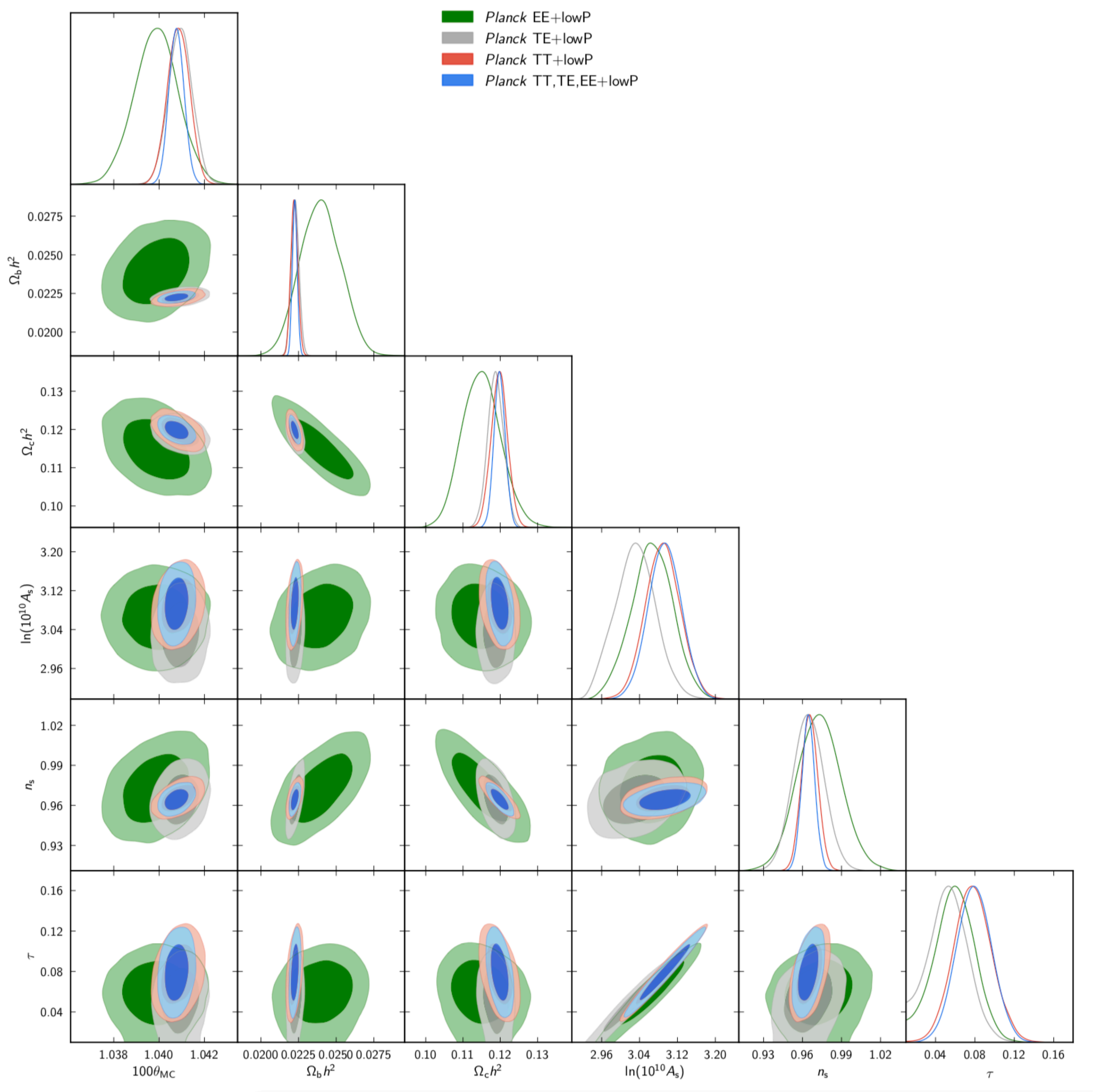
Galton board device	$\longrightarrow$	Computer simulation
Parameters θ	$\longrightarrow$	Model parameters θ
Bins x	$\longrightarrow$	Observables x
Path z	$\longrightarrow$	Latent variables z (stochastic execution trace through simulator)

Cosmological N-body simulations

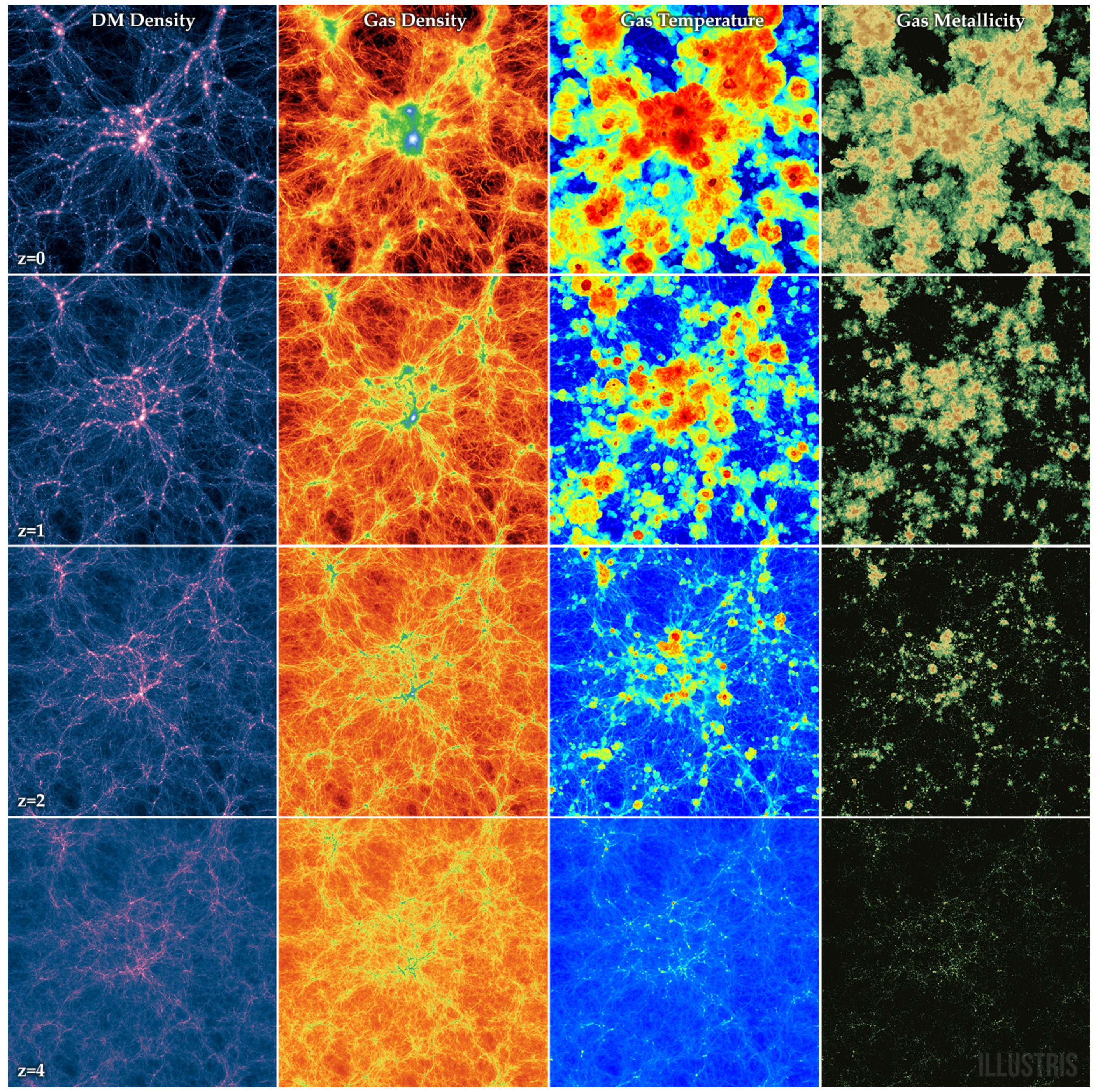
Parameters
θ



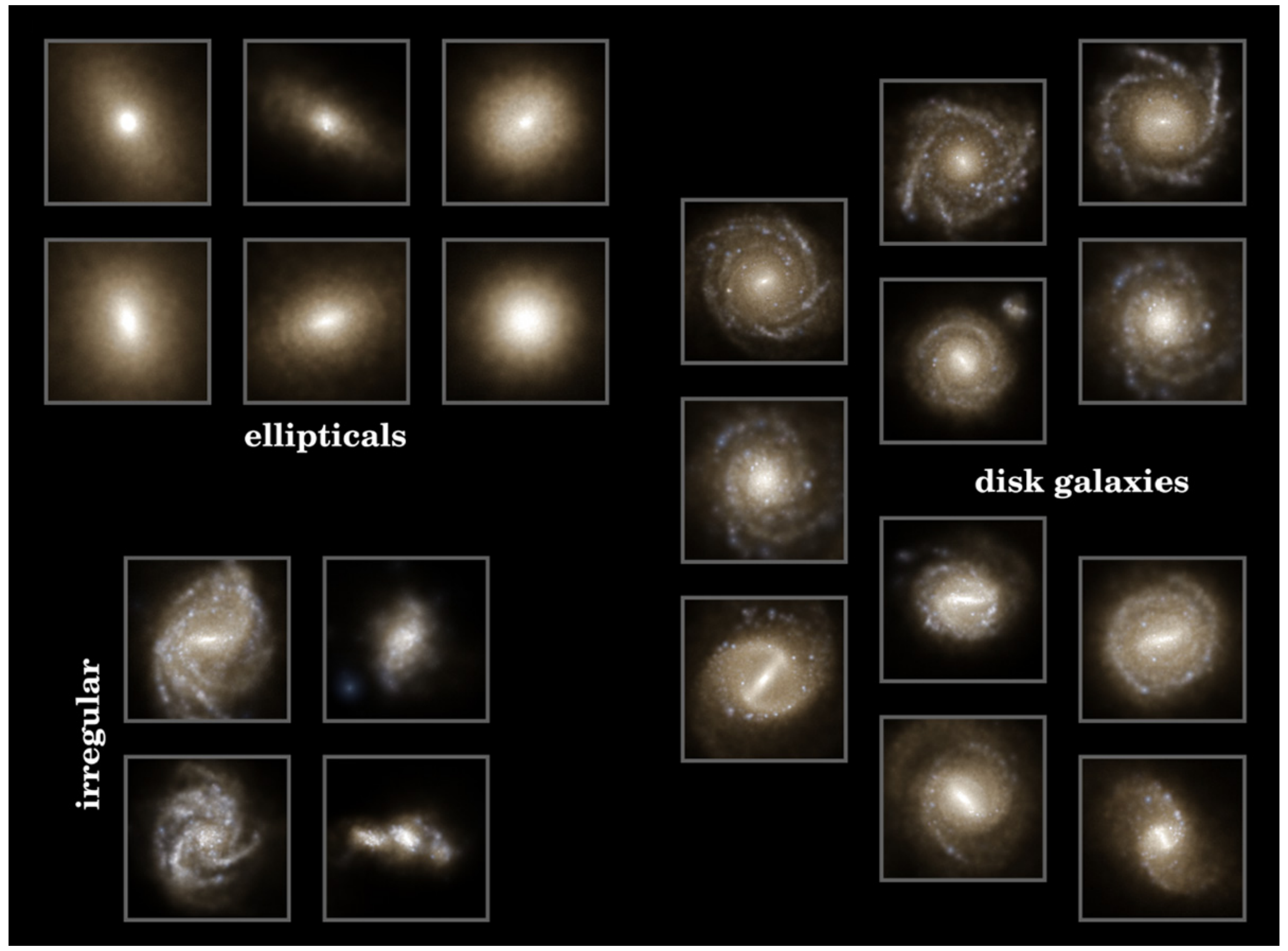
Observables
x



[Planck 1502.01589]



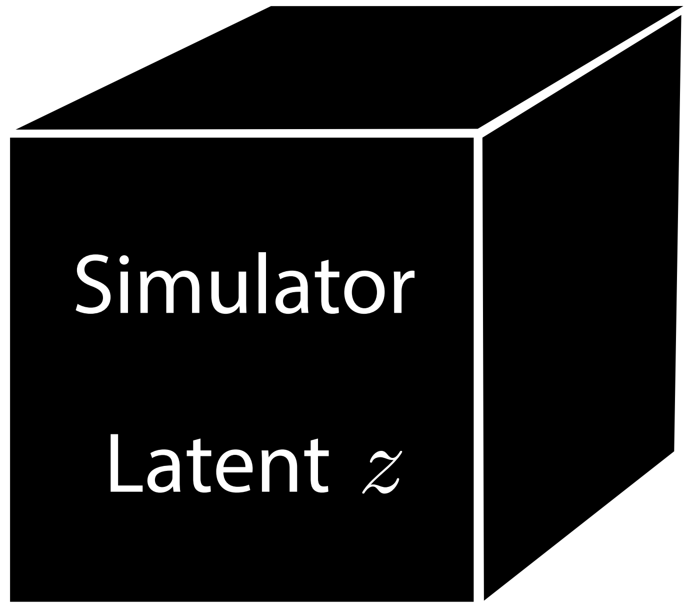
[Illustris 1405.2921]



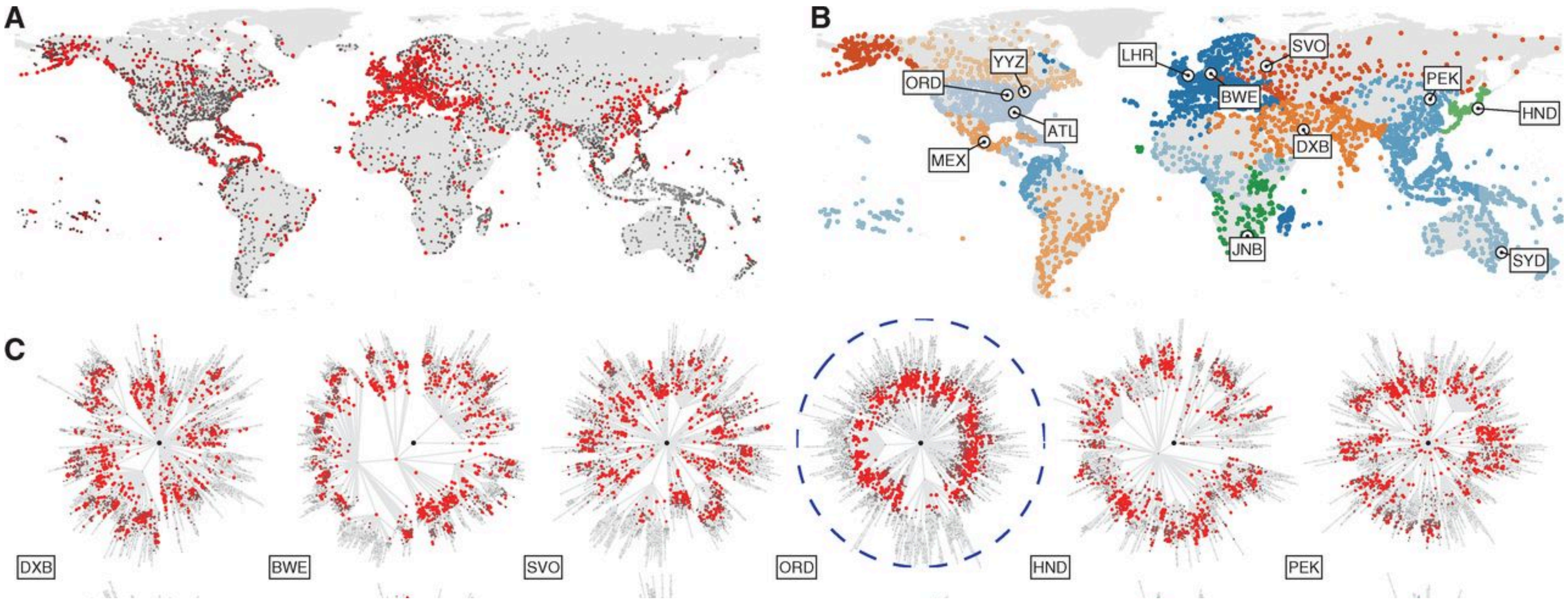
[Illustris 1405.2921]

Epidemiology

Parameters θ $\longrightarrow$

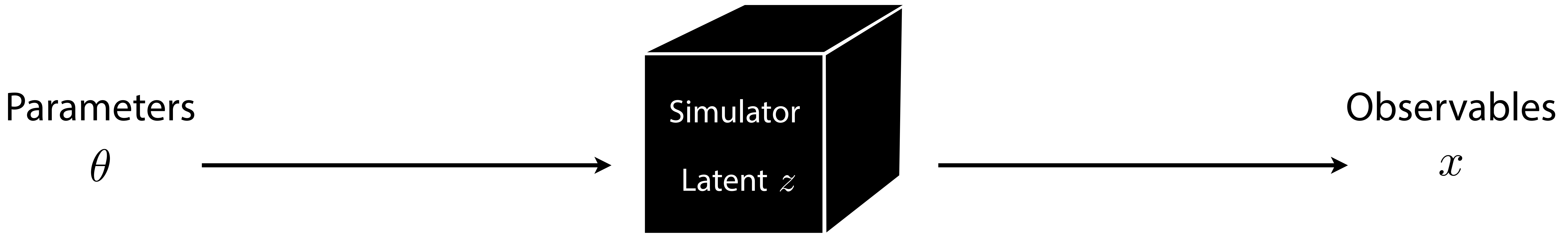


$\longrightarrow$ Observables x

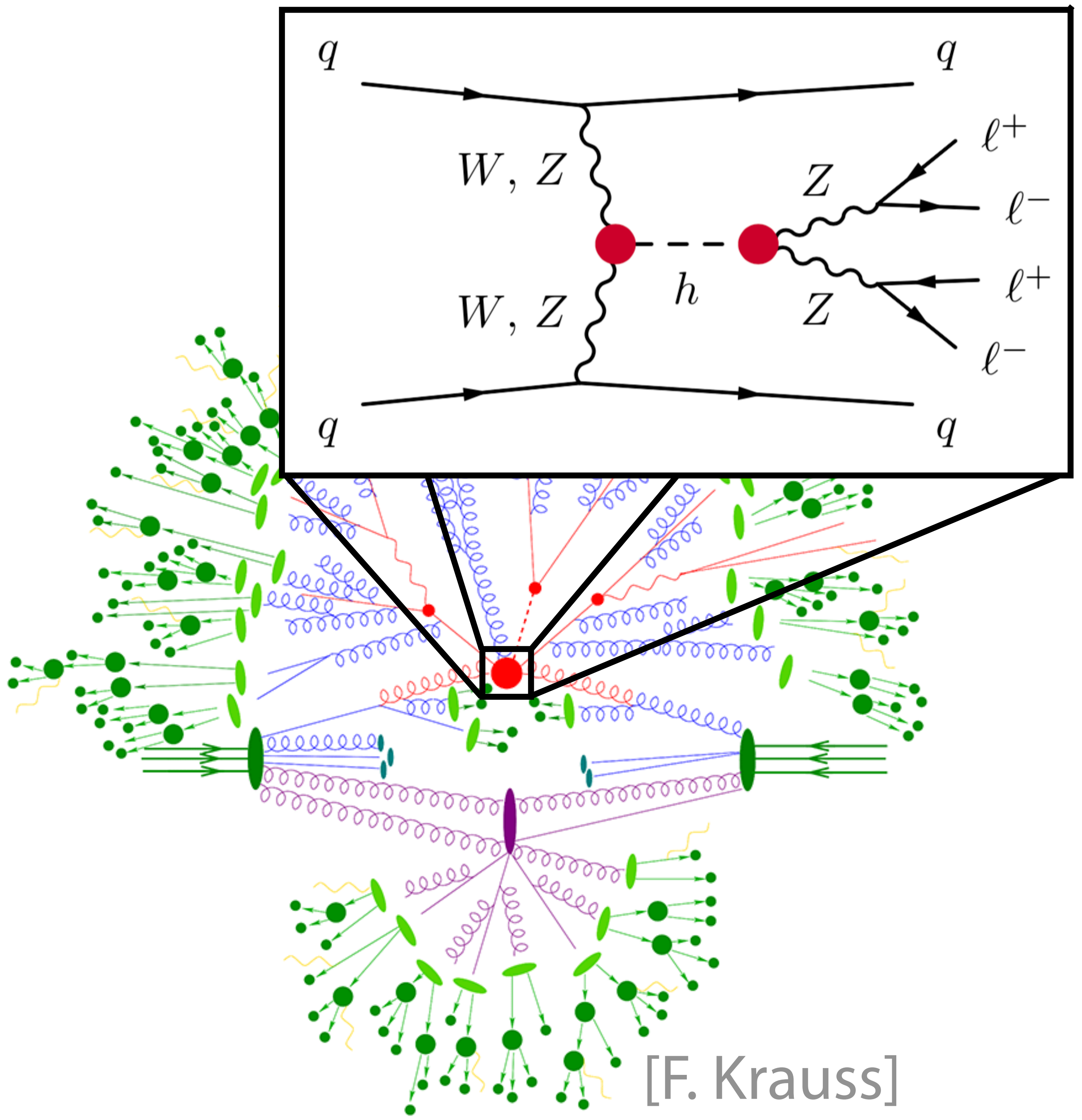


[D. Brockmann, D. Helbing 2013]

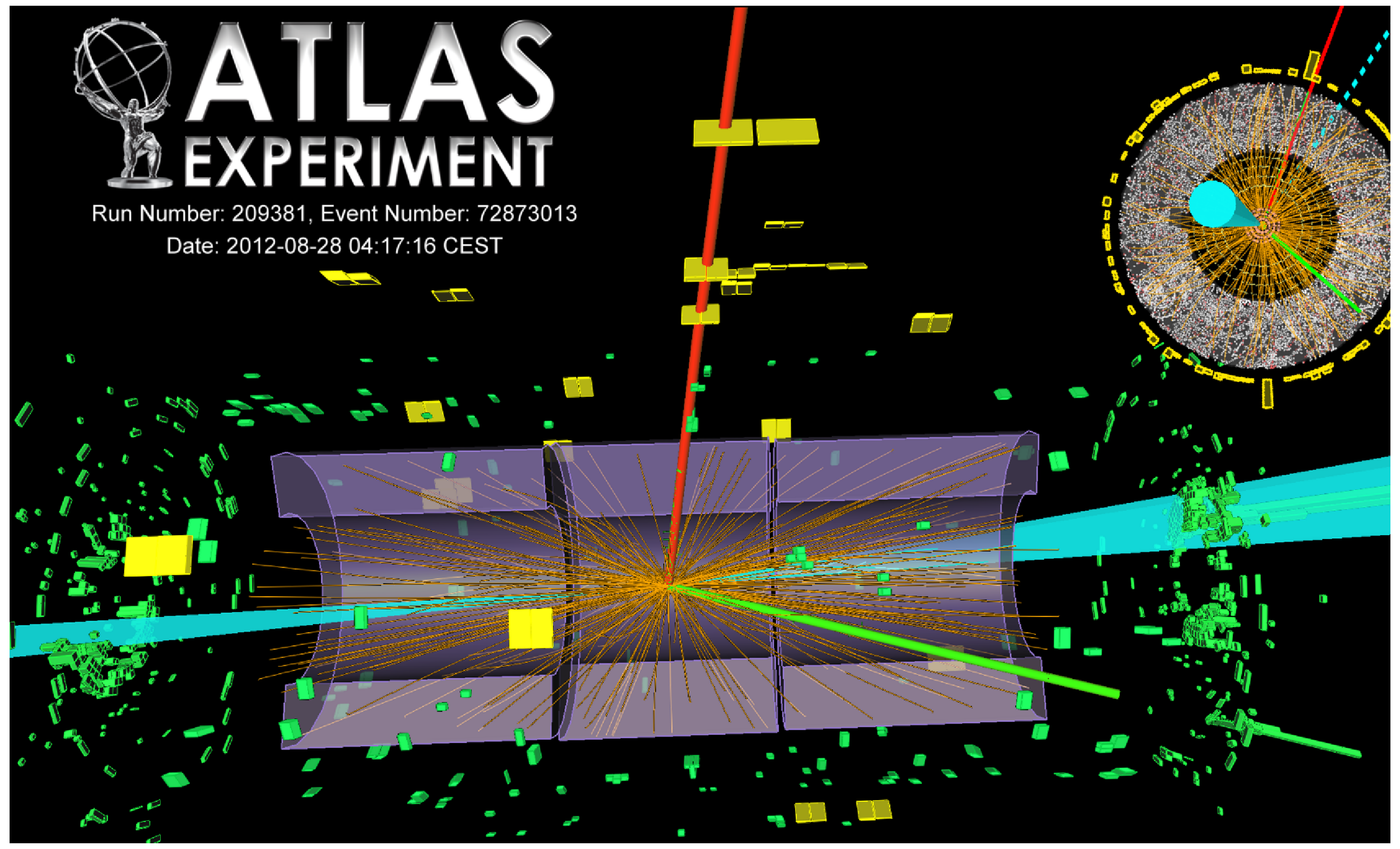
Particle physics



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



[F. Krauss]



[ATLAS 1501.04943]

Why has that not stopped us before?

Solve it by histogramming summary statistics

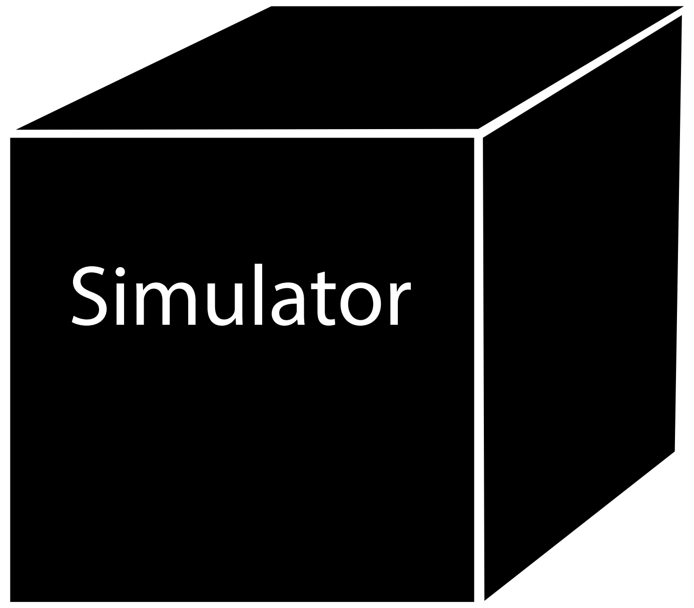
- Typical particle physicists' solution:

Solve it by histogramming summary statistics

- Typical particle physicists' solution:

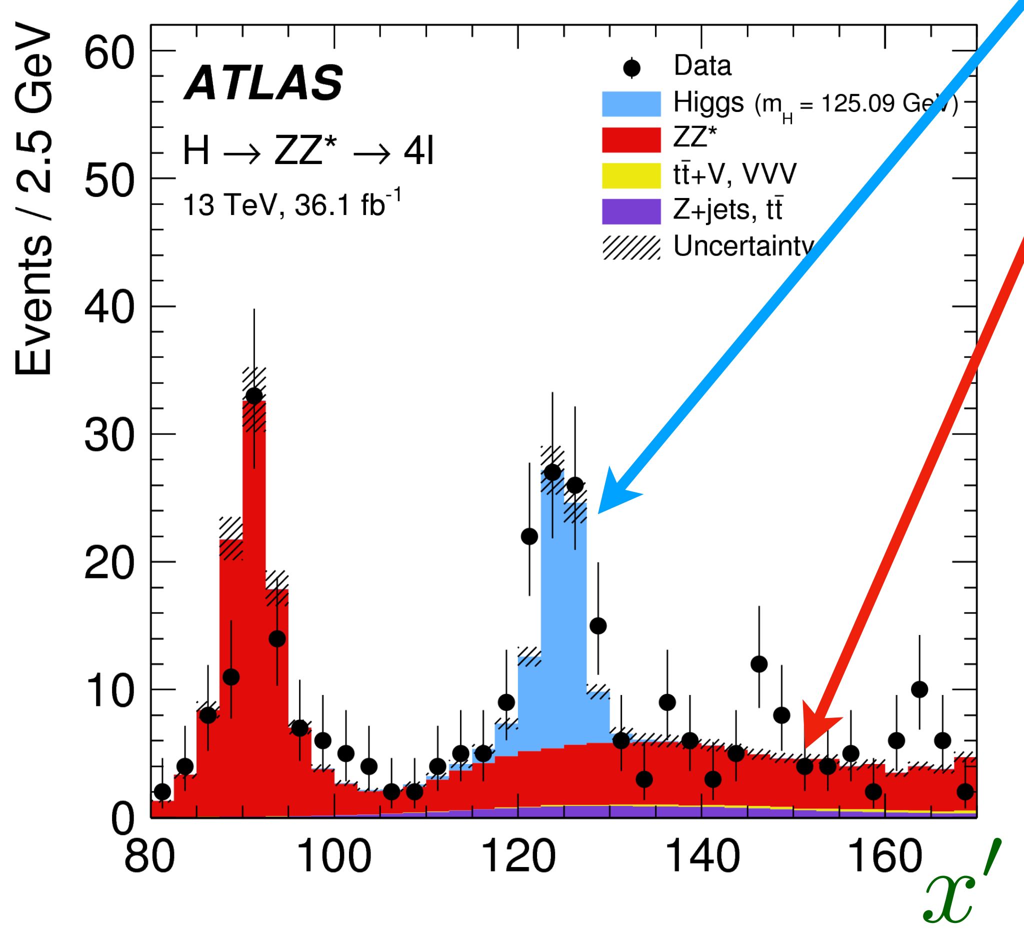
Parameters

θ $\longrightarrow$



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

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



[ATLAS 1712.02304]

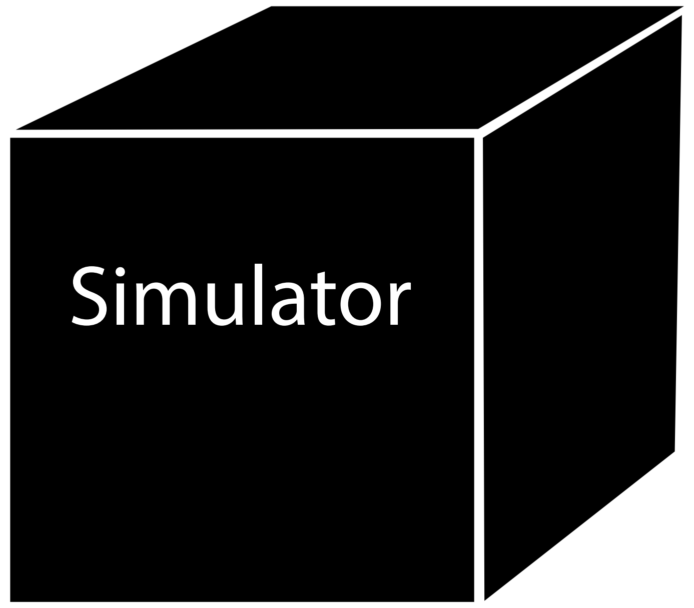
Run simulator for different θ, fill histograms

Solve it by histogramming summary statistics

- Typical particle physicists' solution:

Parameters

θ



Observables

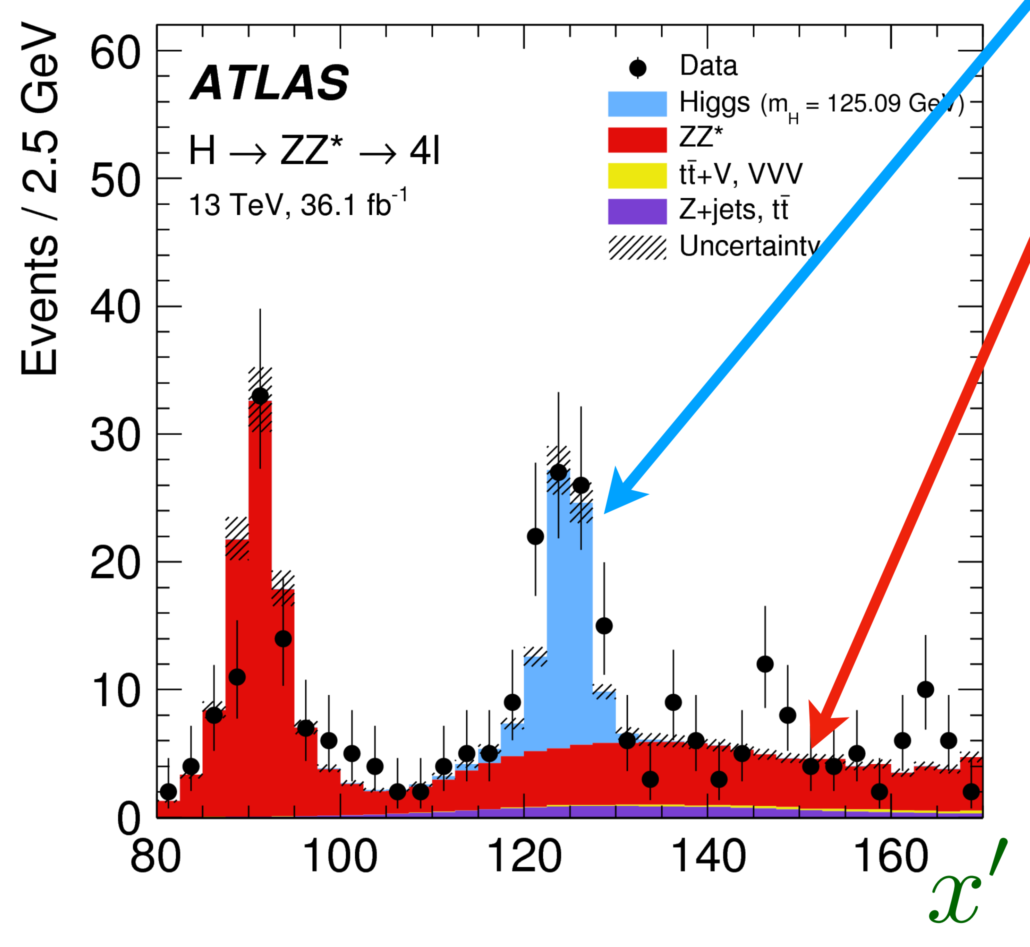
x

Summary statistics

x'

Run simulator for different θ, fill histograms

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



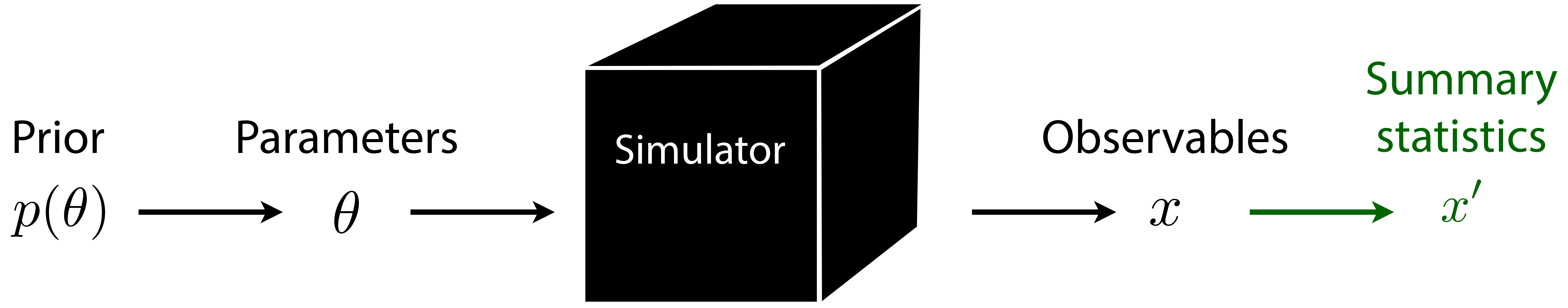
[ATLAS 1712.02304]

- 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

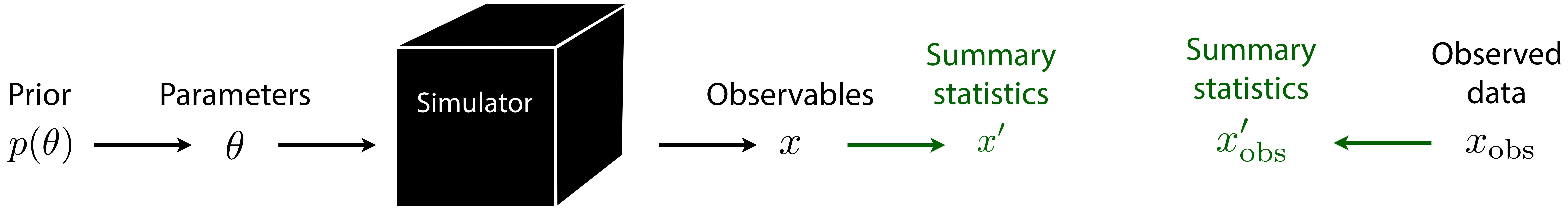
Solve it with Approximate Bayesian Computation (ABC)

[D. Rubin 1984]



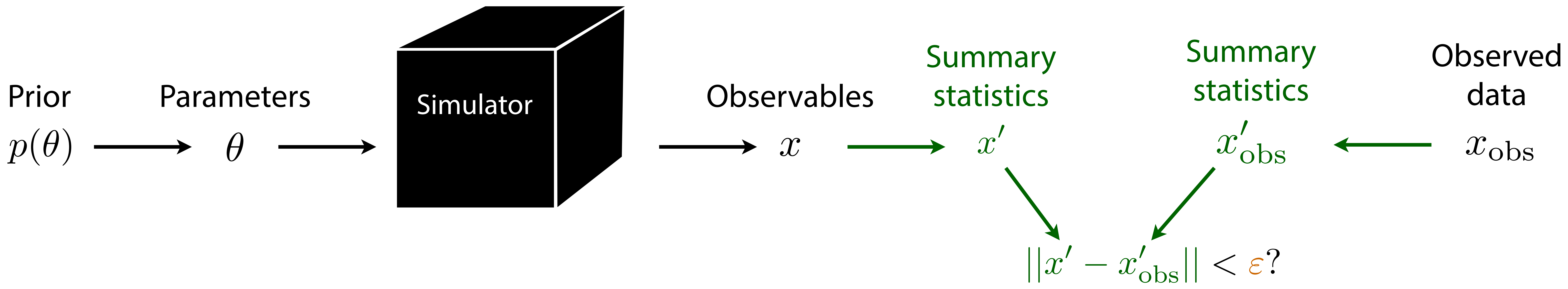
Solve it with Approximate Bayesian Computation (ABC)

[D. Rubin 1984]



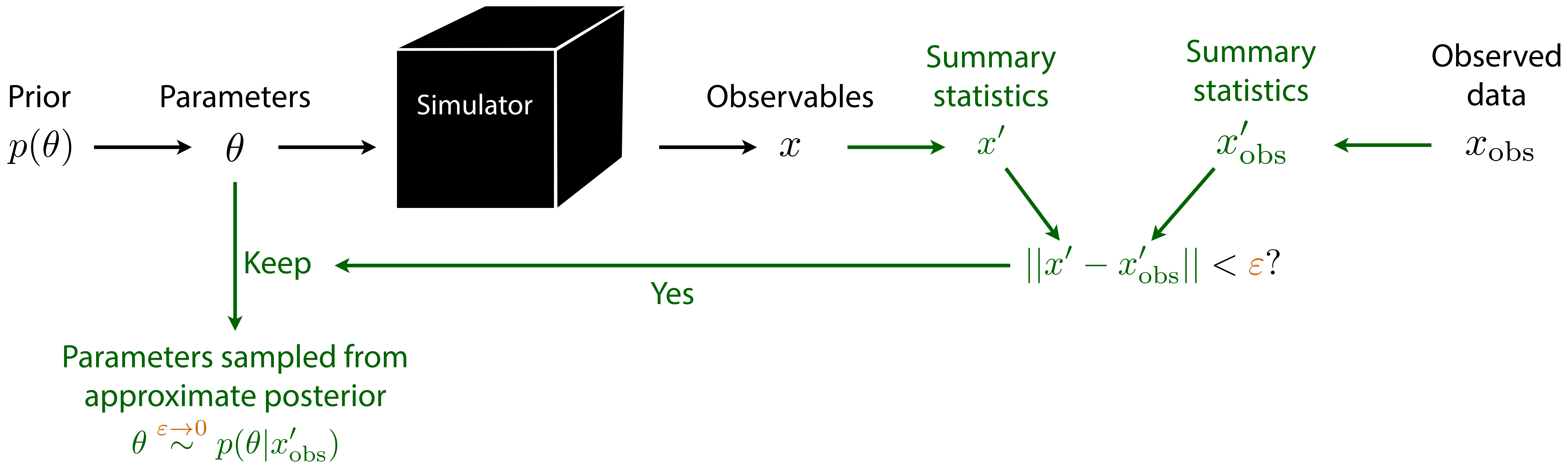
Solve it with Approximate Bayesian Computation (ABC)

[D. Rubin 1984]



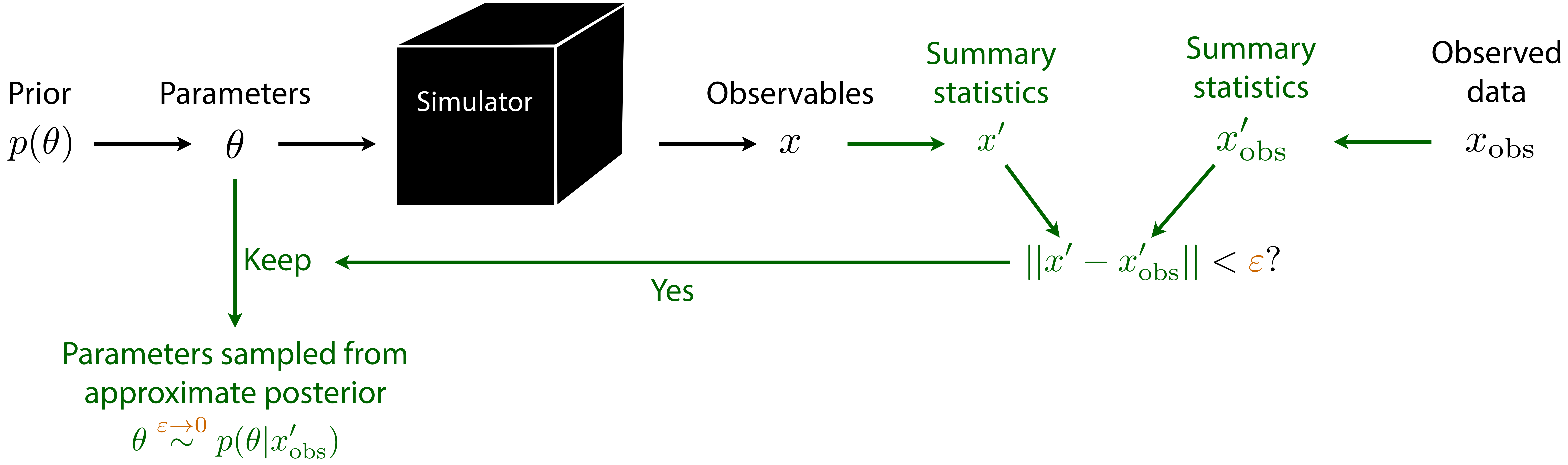
Solve it with Approximate Bayesian Computation (ABC)

[D. Rubin 1984]



Solve it with 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 with machine learning classifiers

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

Solve it with machine learning classifiers

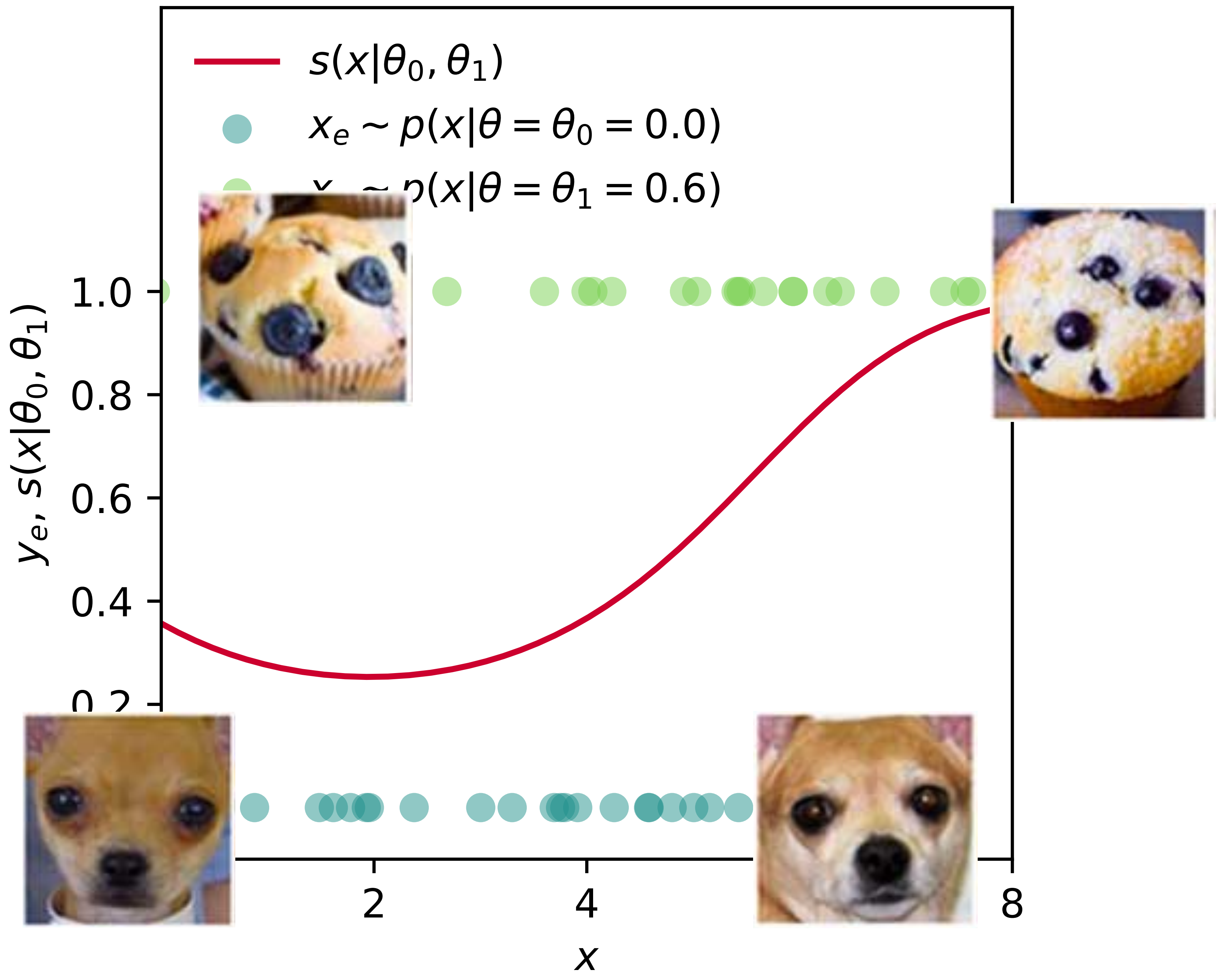
[K. Cranmer, J. Pavez, G. Louppe 1506.02169]

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

Solve it with machine learning classifiers

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

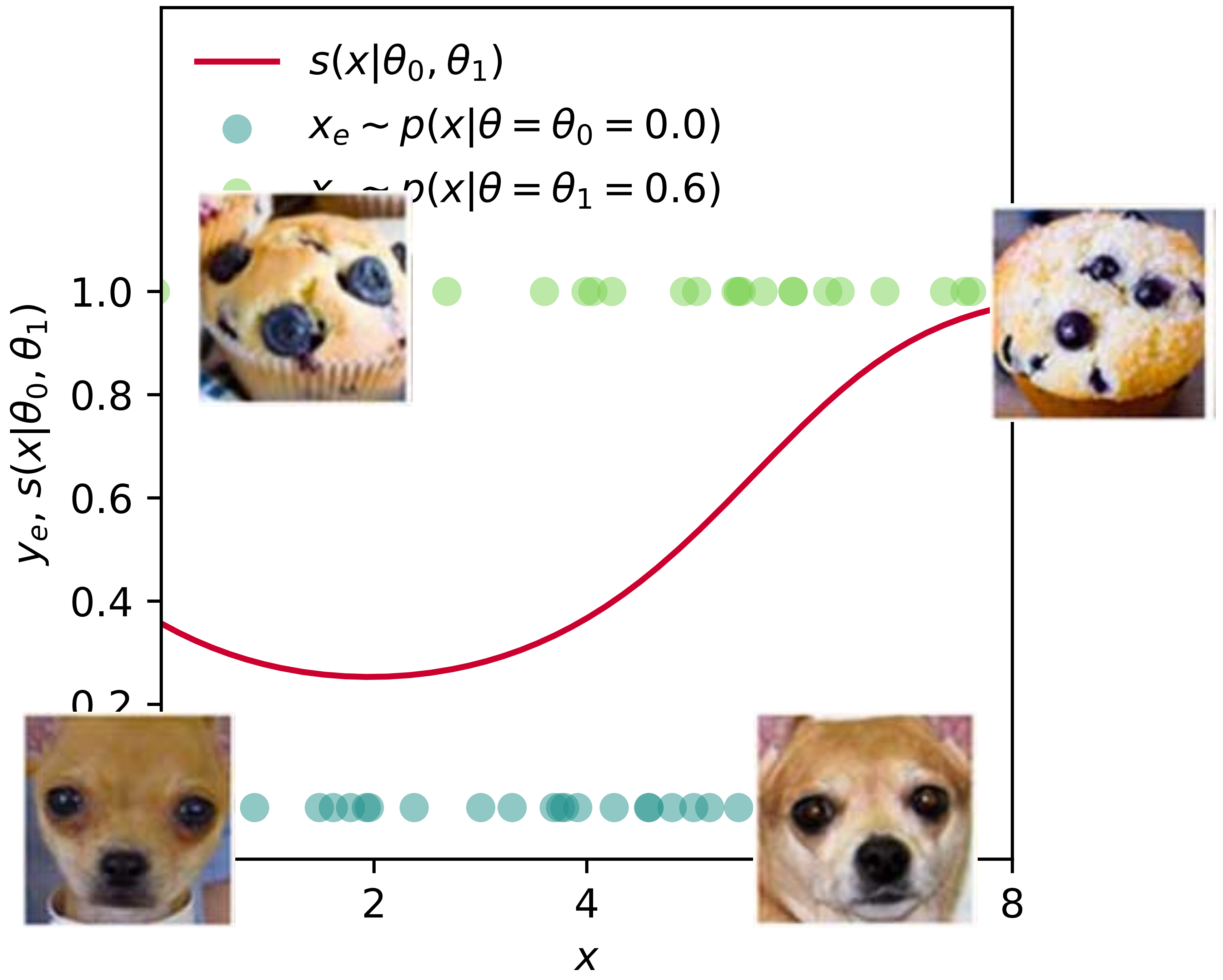
- Train neural network (BDT, ...) to tell $x \sim p(x|\theta_0)$ from $x \sim p(x|\theta_1)$
  - Classifier output $\hat{s}(x)$ is closer to 0 for θ_0-like events (closer to 1 for θ_1-like events)



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)



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

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

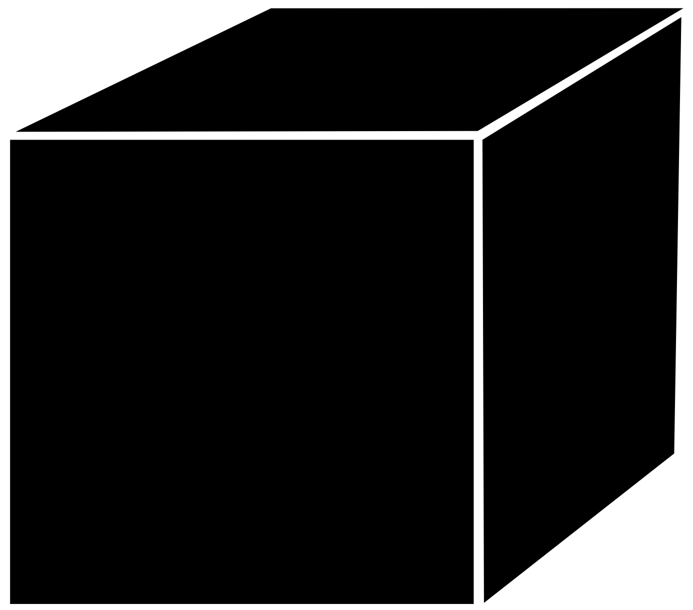
$\Rightarrow$ No summary statistics necessary, very fast evaluation… but may require large training samples

An incomplete list of likelihood-free inference methods

Treat simulator as black box:

- Histograms of observables,
  Approximate Bayesian Computation
  Rely on summary statistics

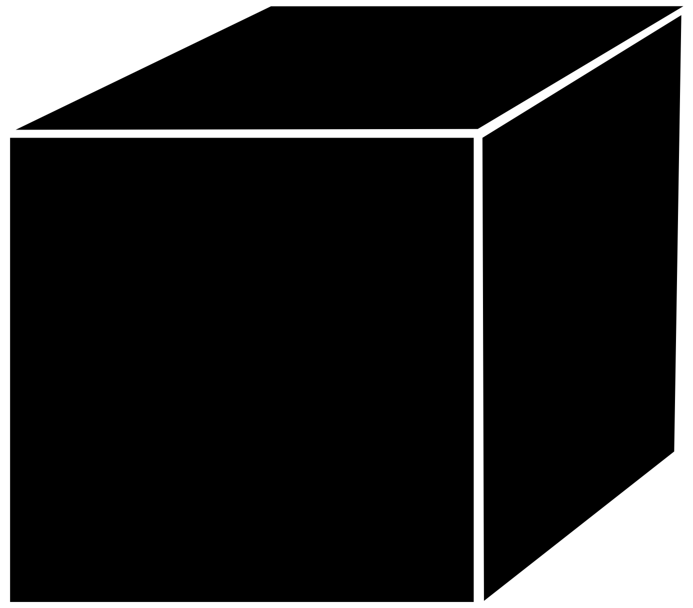
- Machine learning techniques
  Density networks, CARL, autoregressive models,
  normalizing flows, …



An incomplete list of likelihood-free inference methods

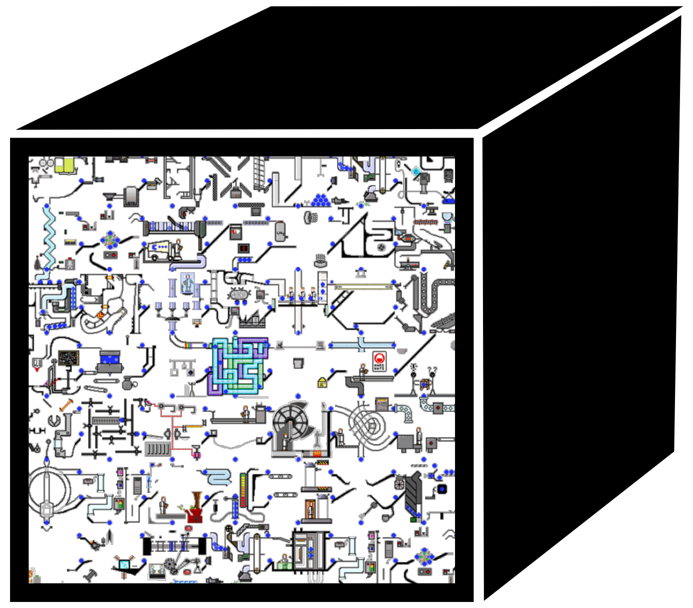
Treat simulator as black box:

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



Use latent structure:

- Matrix Element Method, Optimal Observables, Shower Deconstruction, Event Deconstruction
  Specific to particle physics: approximate integrand, explicitly calculate z integral

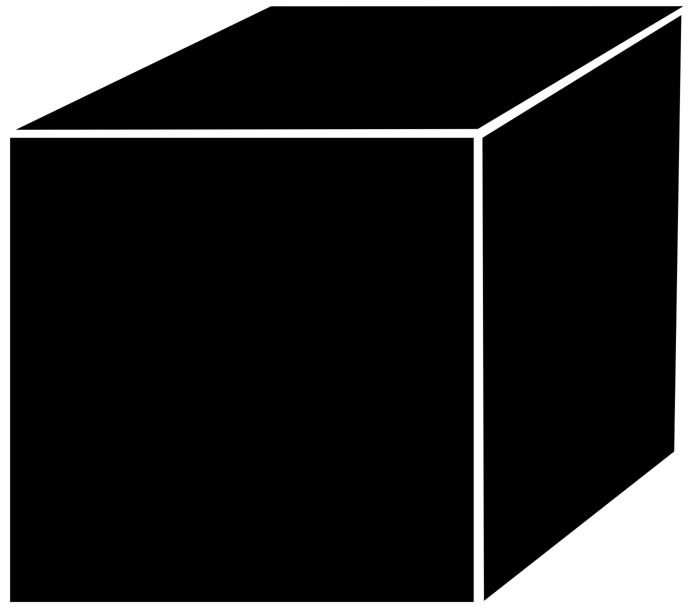


An incomplete list of likelihood-free inference methods

Treat simulator as black box:

- Histograms of observables, Approximate Bayesian Computation
  Rely on summary statistics

- Machine learning techniques
  Density networks, CARL, autoregressive models, normalizing flows, …

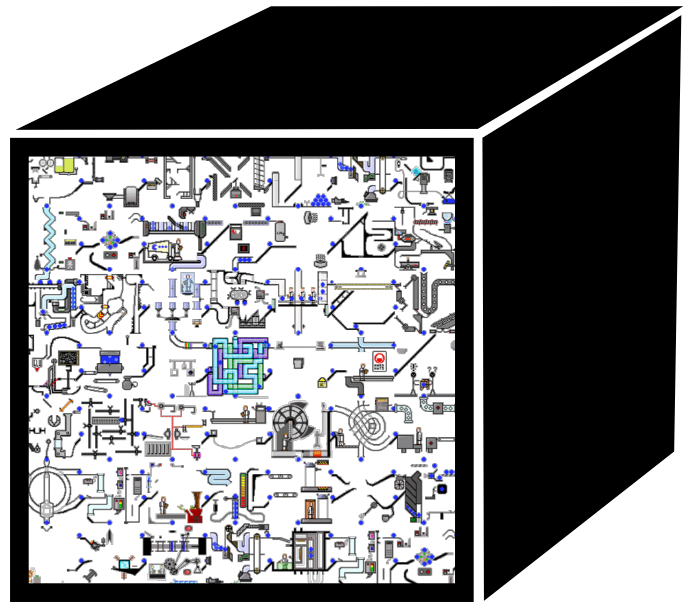


Use latent structure:

- Matrix Element Method, Optimal Observables, Shower Deconstruction, Event Deconstruction
  Specific to particle physics: approximate integrand, explicitly calculate z integral

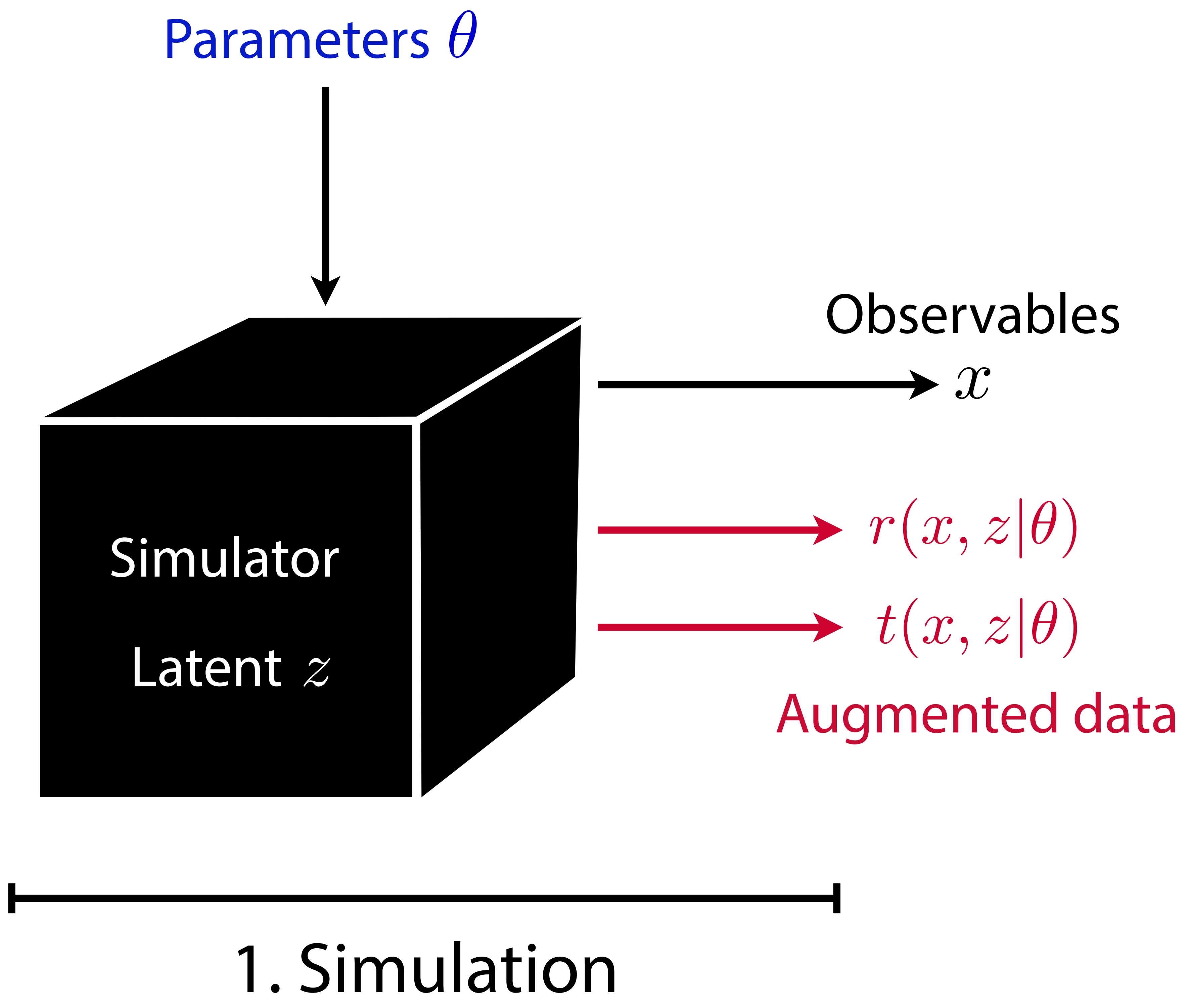
New!

- Mining gold from the simulator
  Leverage physics insight + machine learning



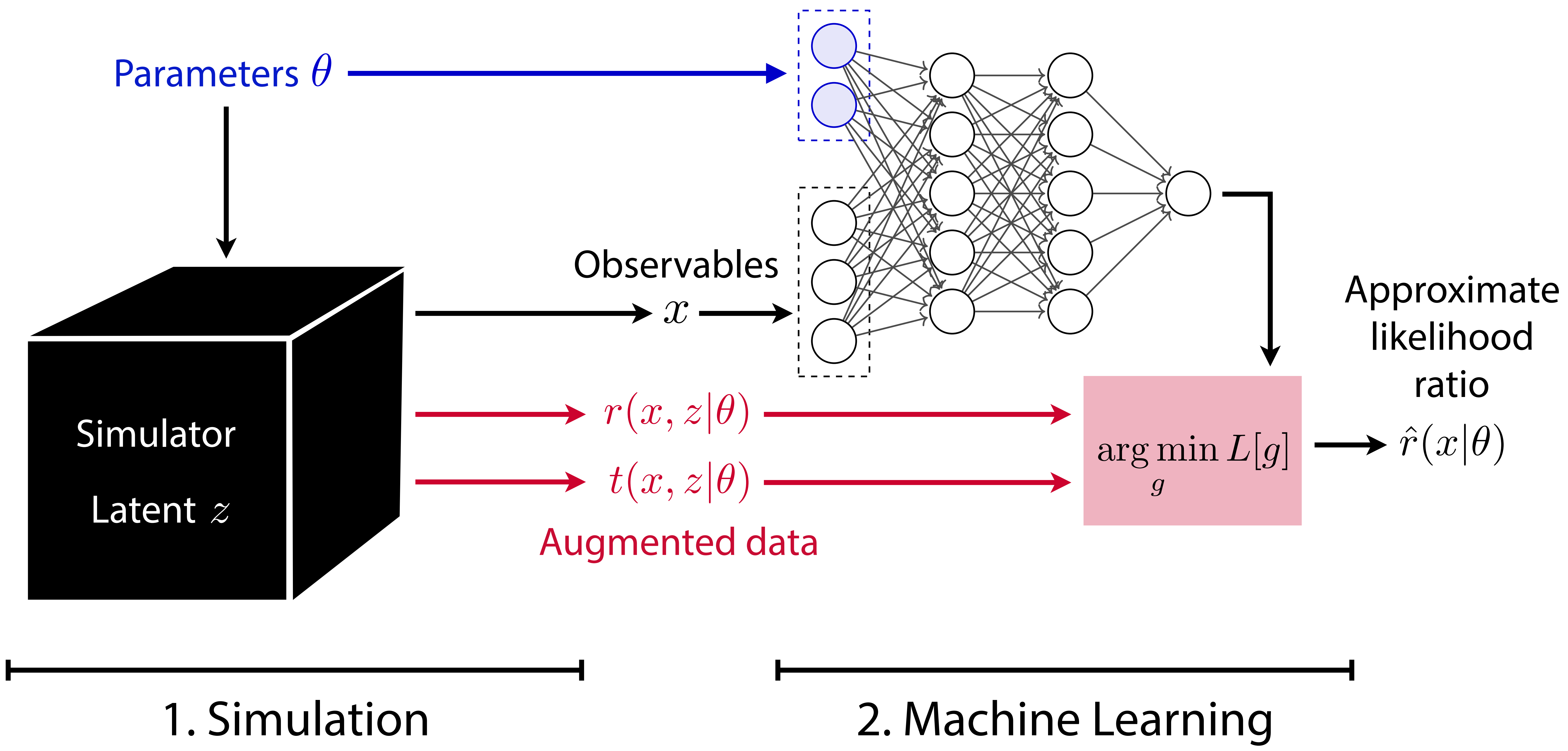
A new approach:
Mining gold from the simulator

Bird's-eye view



"Mining gold": Extract additional information from simulator

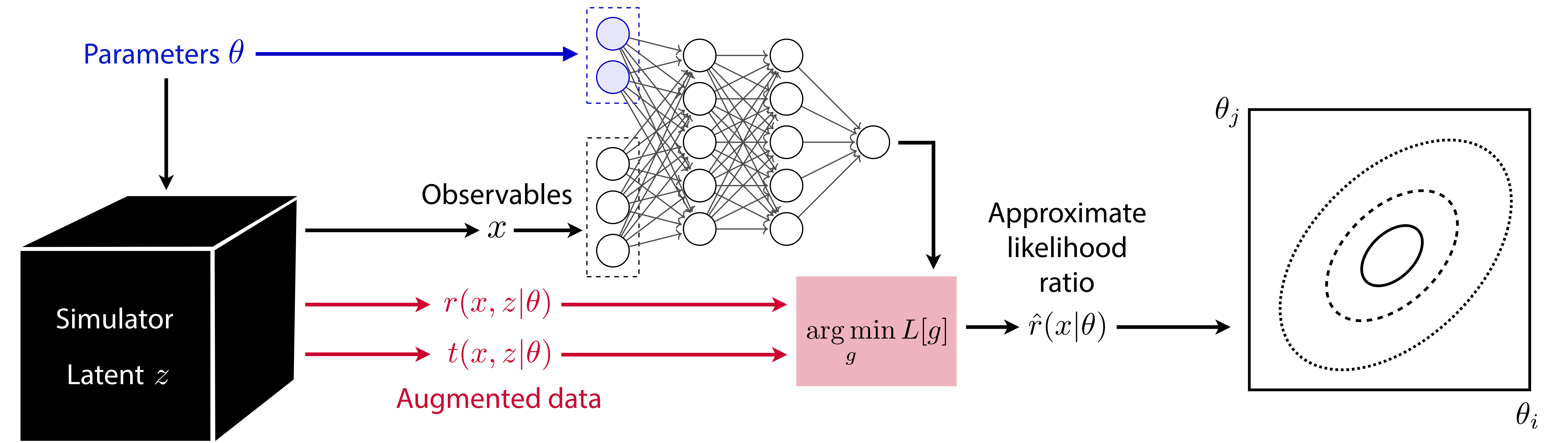
Bird's-eye view



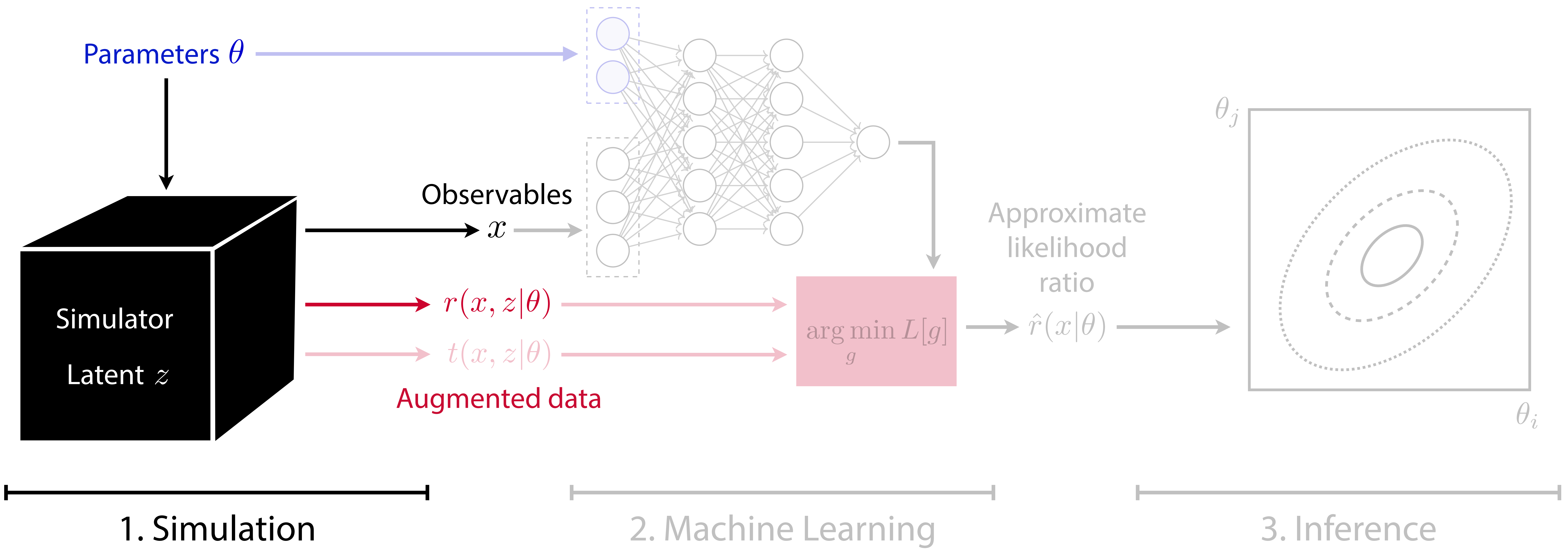
"Mining gold": Extract additional information from simulator

Use this information to train estimator for likelihood ratio

Bird's-eye view

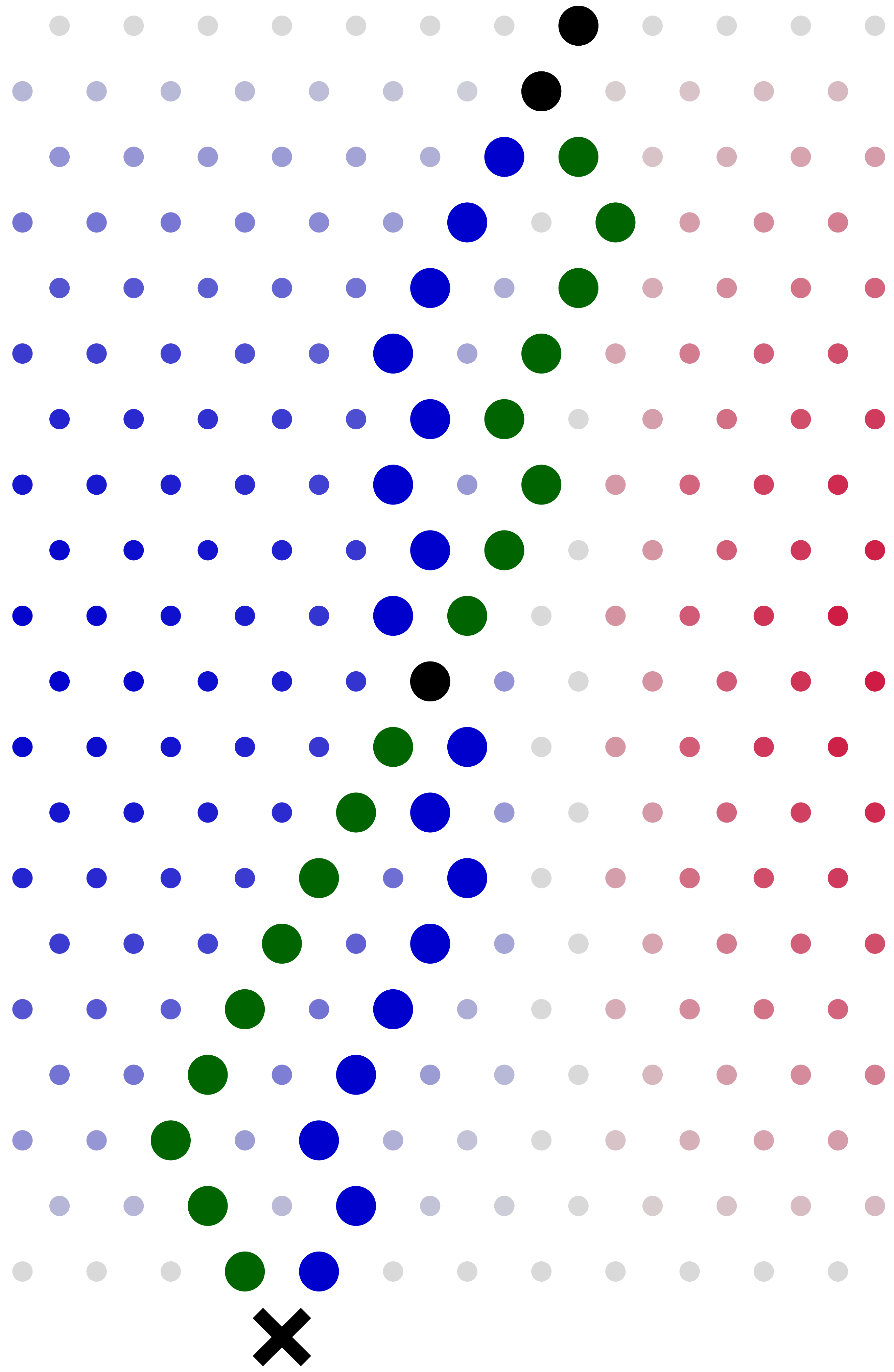


1. Simulation	2. Machine Learning	3. Inference
"Mining gold": Extract additional information from simulator	Use this information to train estimator for likelihood ratio	Limit setting with standard hypothesis tests



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

Back to the Galton board



- Remember: the likelihood

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

is intractable because of the integral over all possible paths z

- But: we can calculate the probability of each individual path

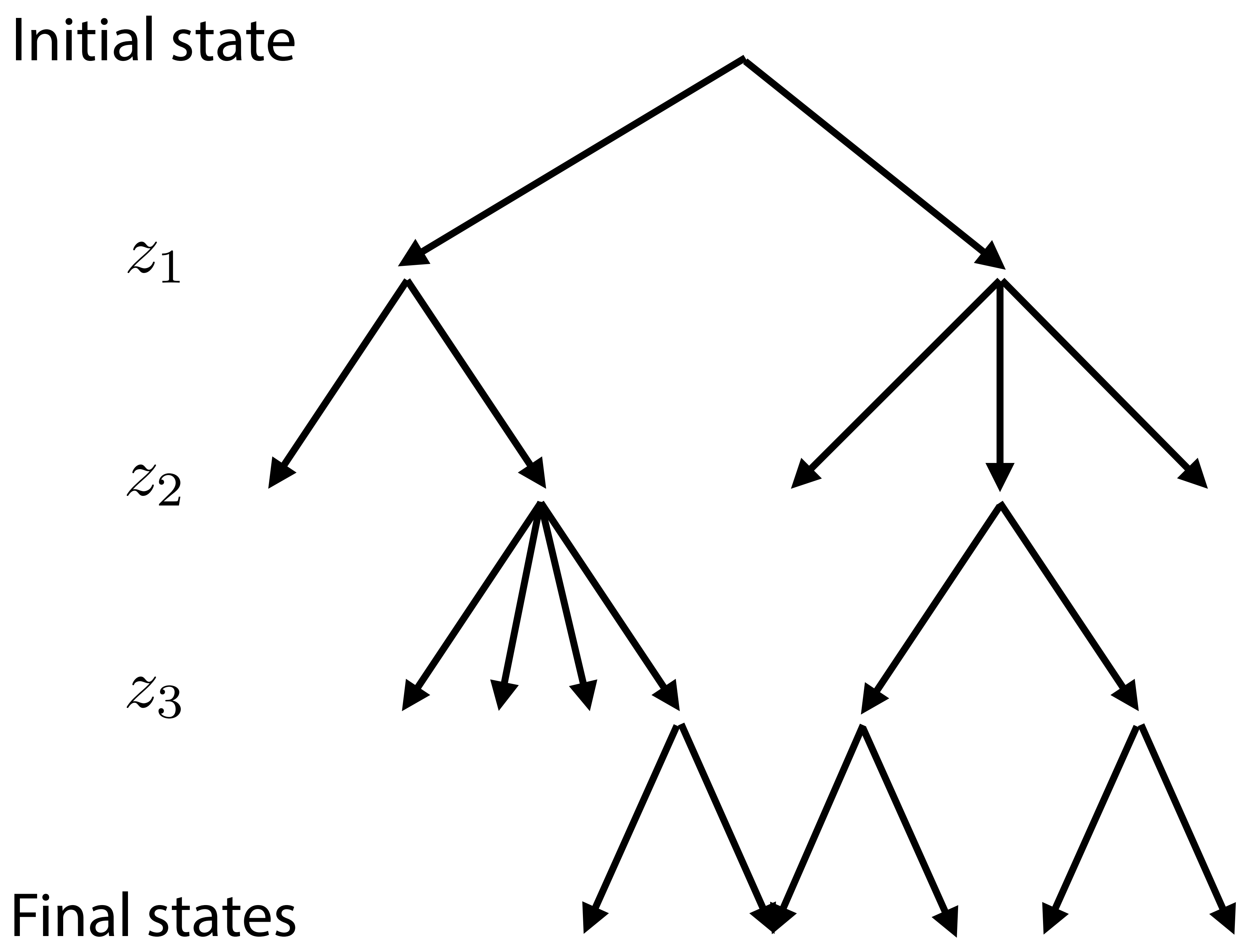
$$p(x, z|\theta) = \prod_{\text{nails } i} p_i(x, z|\theta)$$

and the "joint likelihood ratio" conditional on a particular path

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

Generalizing this idea

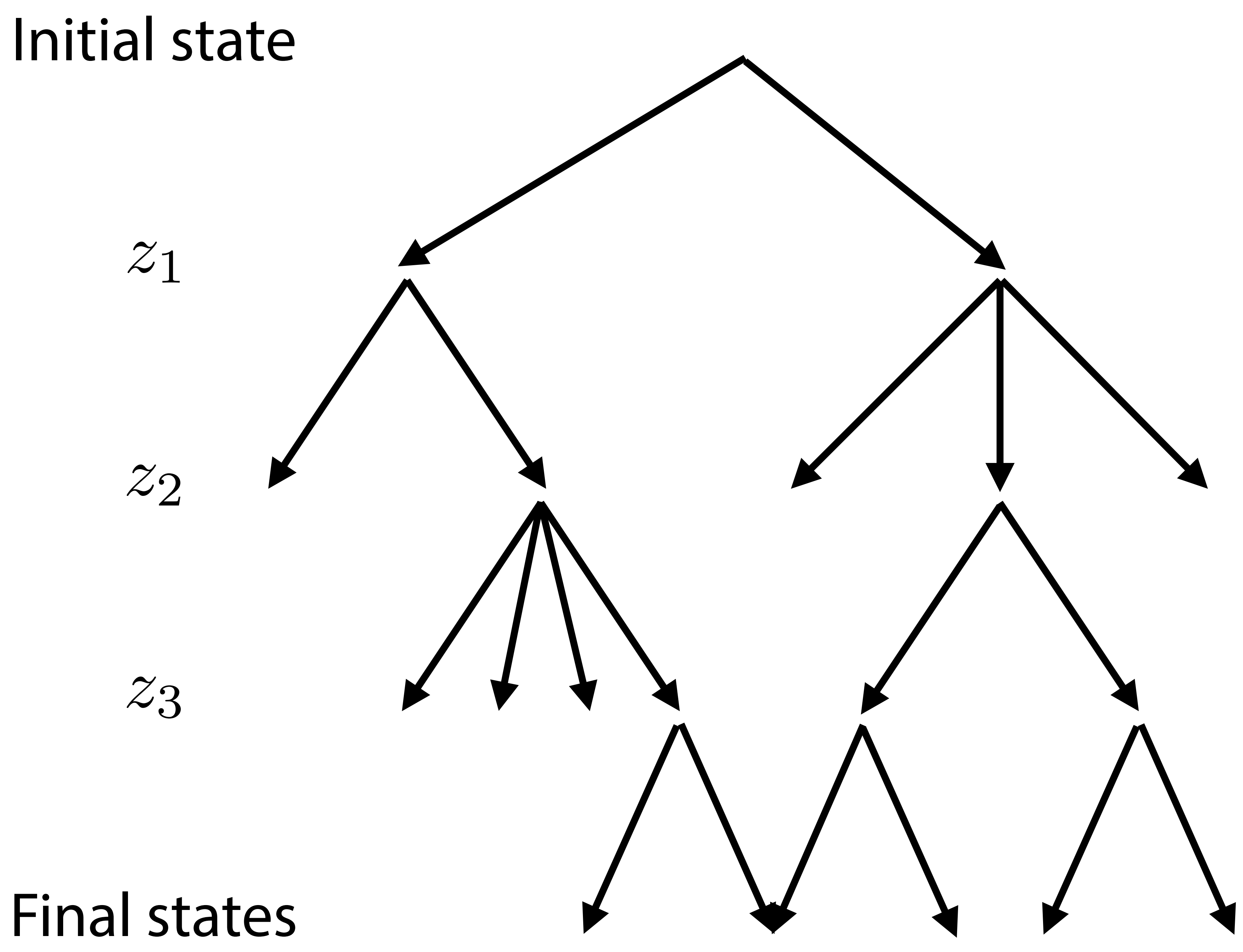
- Computer simulation typically evolve along a tree-like structure of successive random branchings



Generalizing this idea

- 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()
```



Generalizing this idea

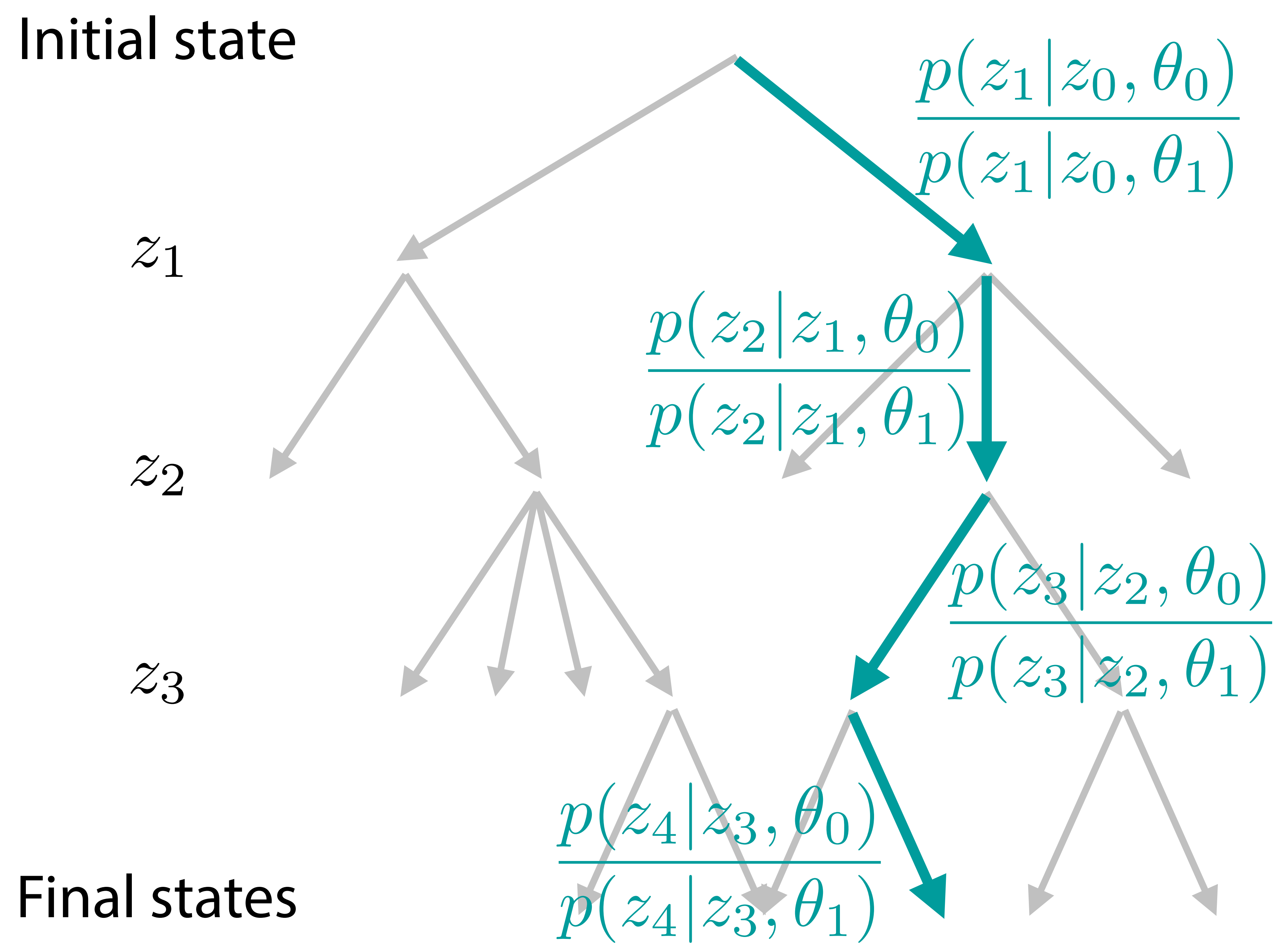
- Computer simulation typically evolve along a tree-like structure of successive random branchings

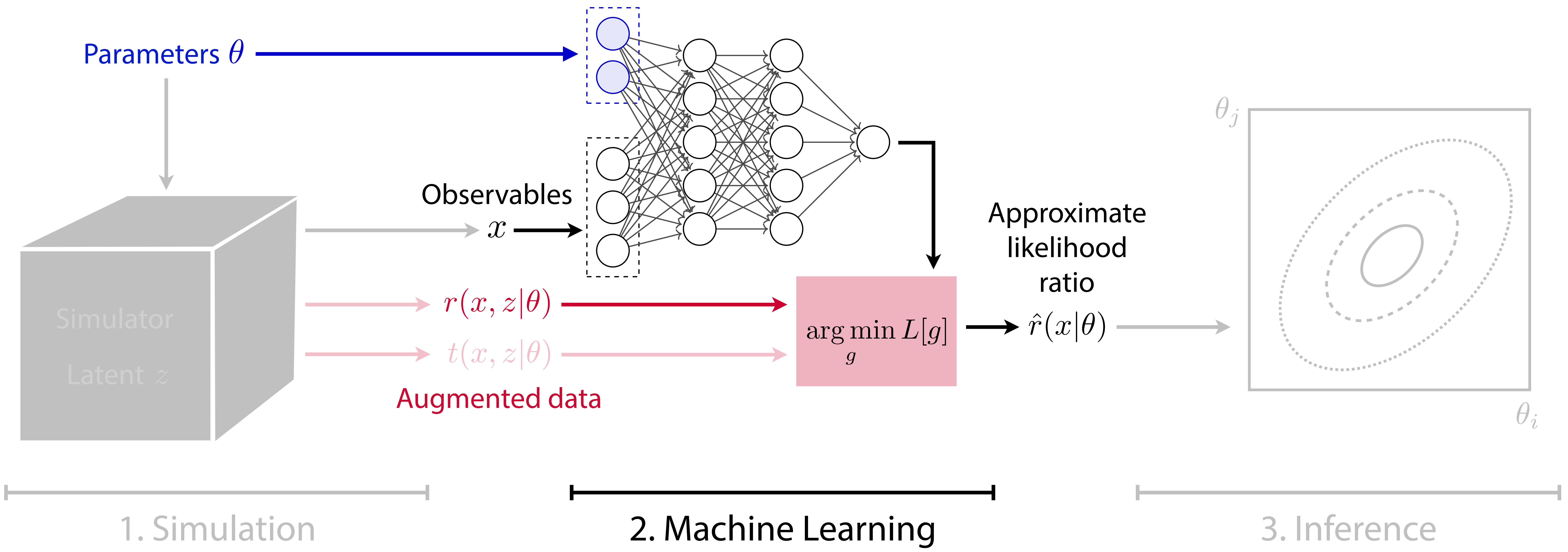
- The probabilities of each branching $p_i(z_i|z_{i-1}, \theta)$ are often clearly defined in the code:

```
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

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

We want the likelihood ratio function

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

The value of gold

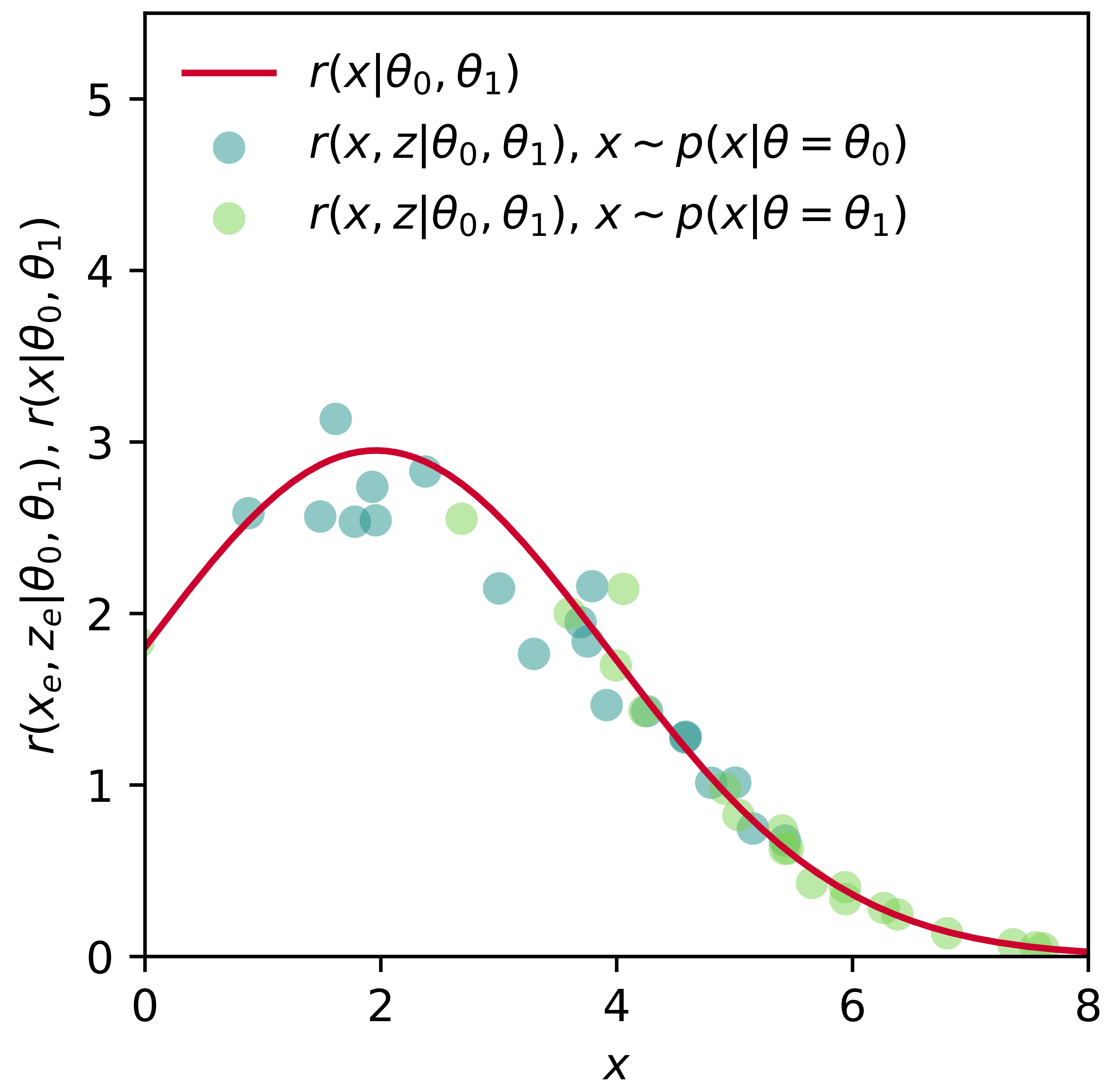
We can calculate the joint likelihood ratio

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

We want the likelihood ratio function

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

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



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

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

It is minimized by

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

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

We want the likelihood ratio function

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

Machine learning = applied calculus of variatios

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

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

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:

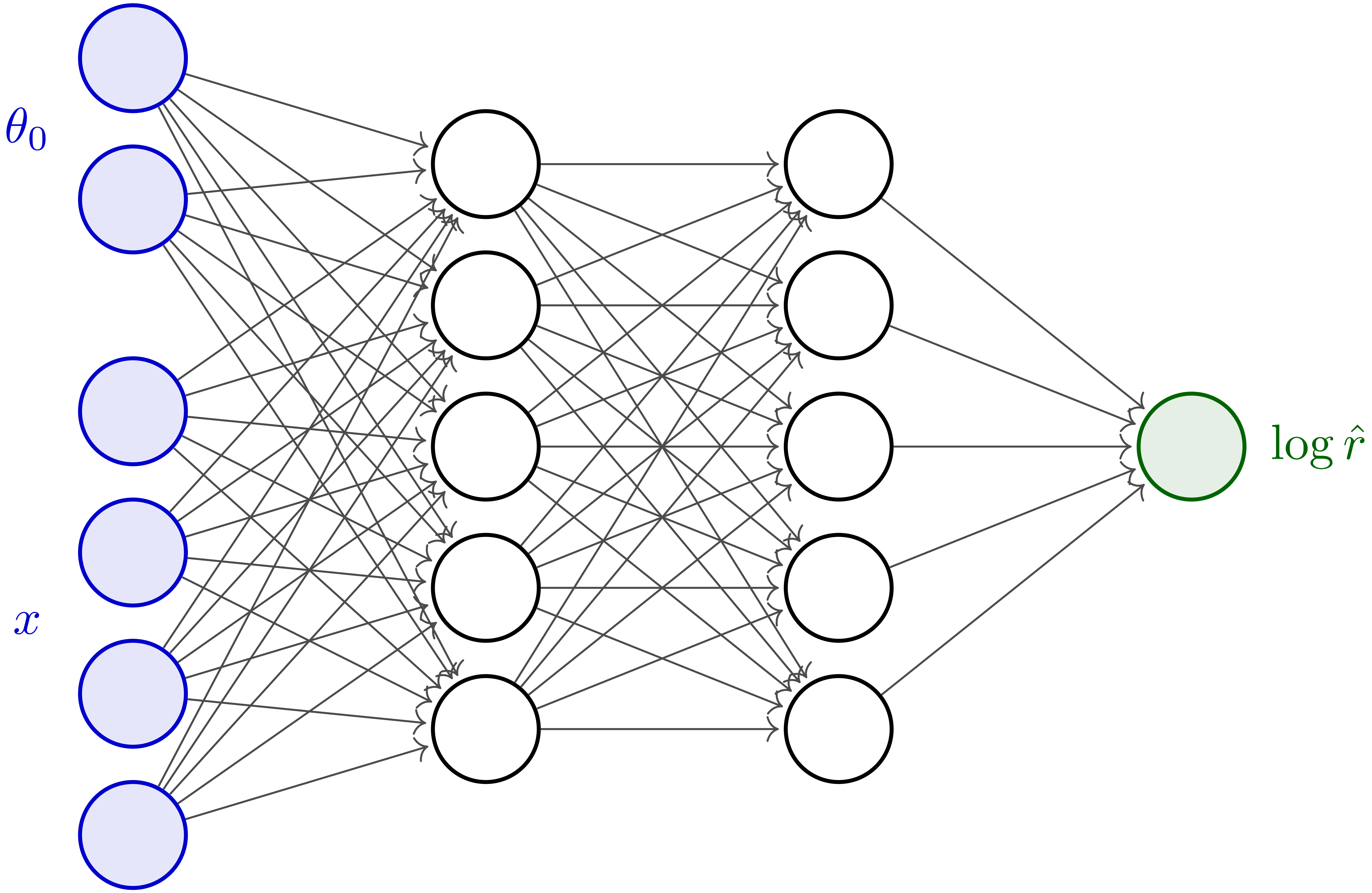
- Functional L_r $\longrightarrow$ Loss function

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

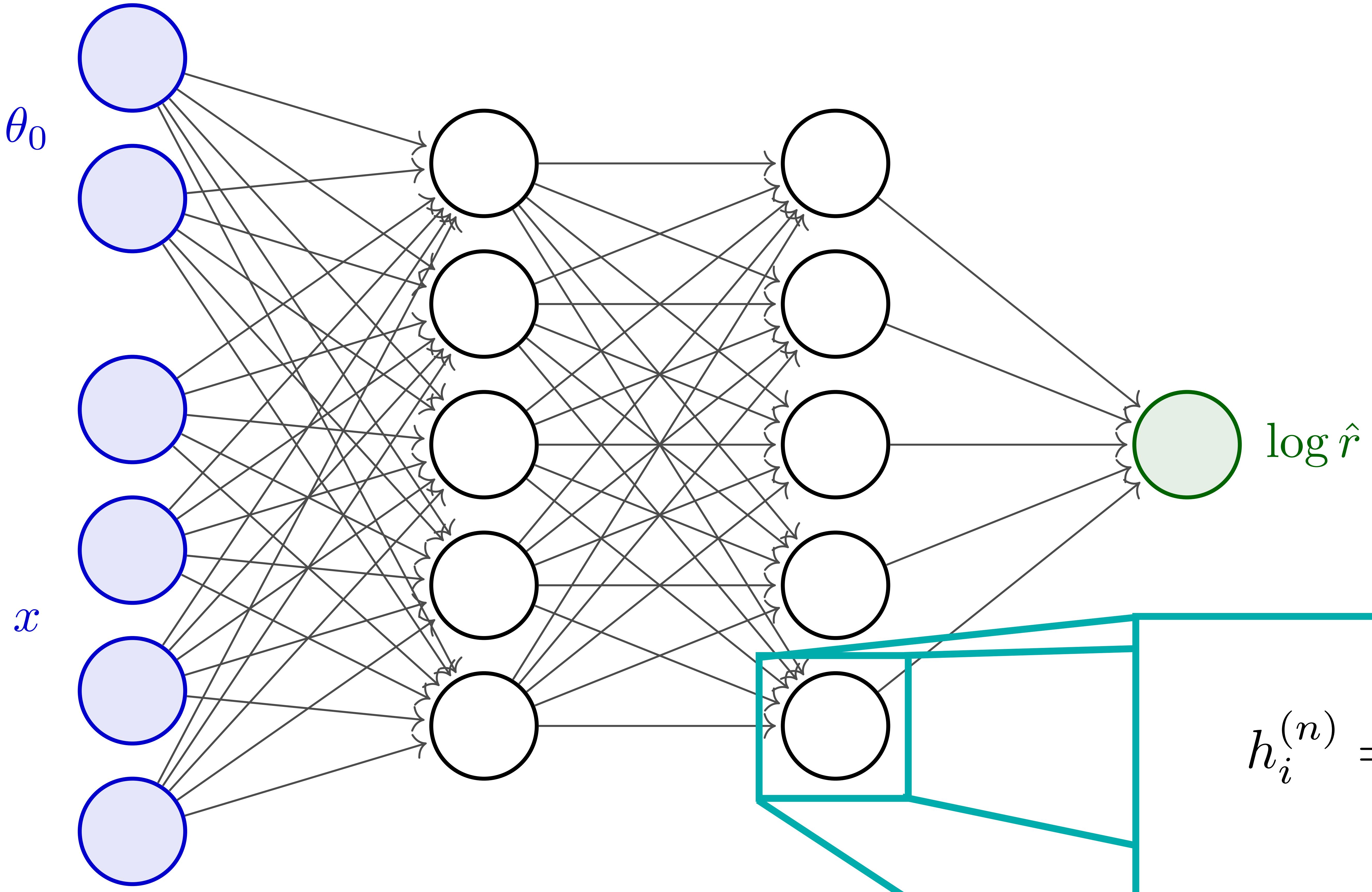
- Variational family $\hat{r}(x|\theta_0,\theta_1)$ $\longrightarrow$ Flexible parametric function (e.g. neural network)
- Exact minimization $\longrightarrow$ Numerical optimization algorithm (e.g. stochastic gradient descent)

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

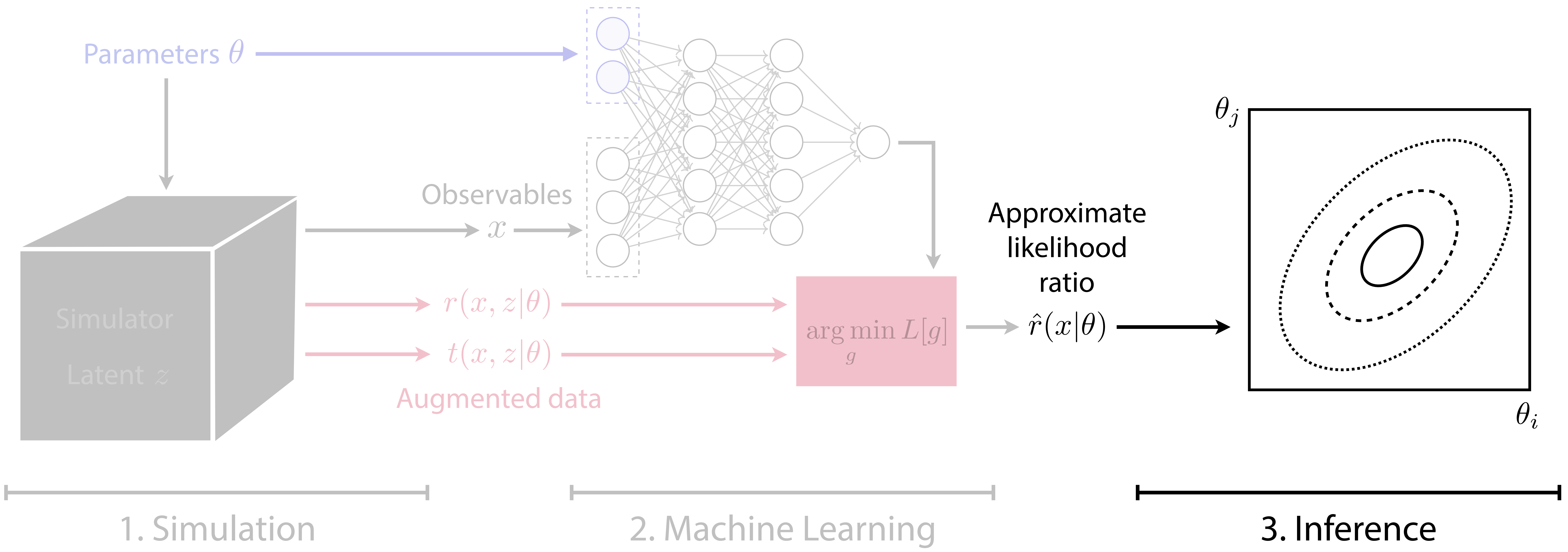


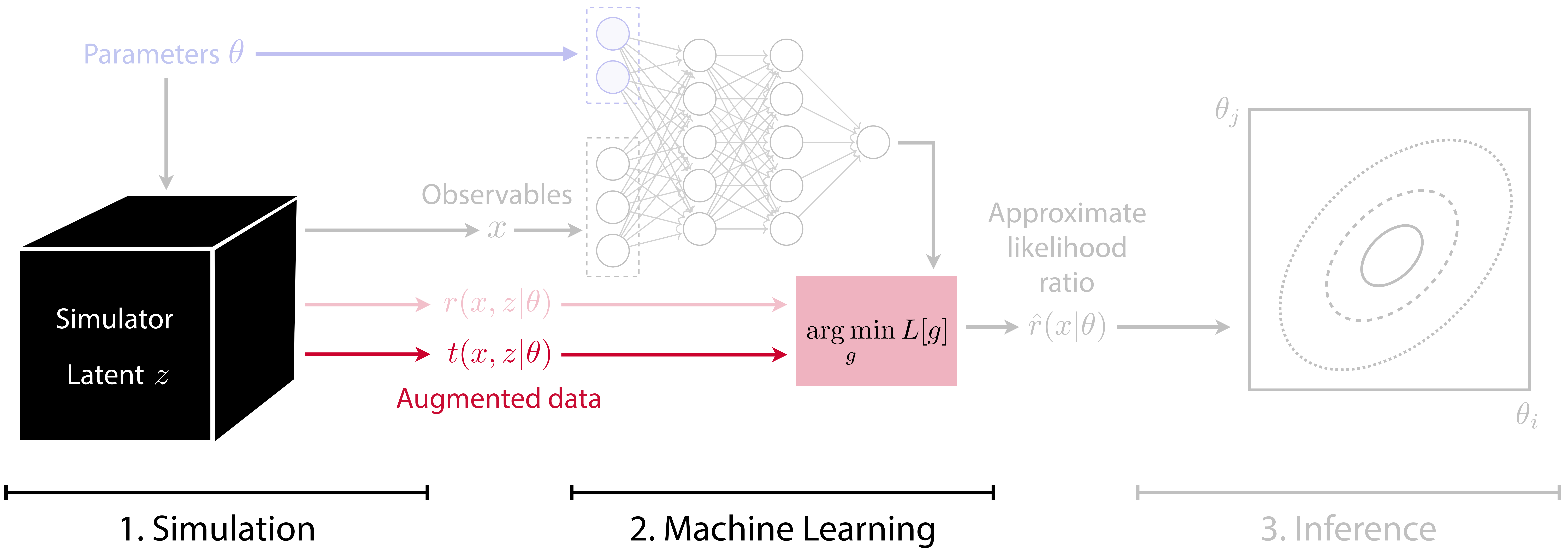
Neural networks = universal function approximators



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

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

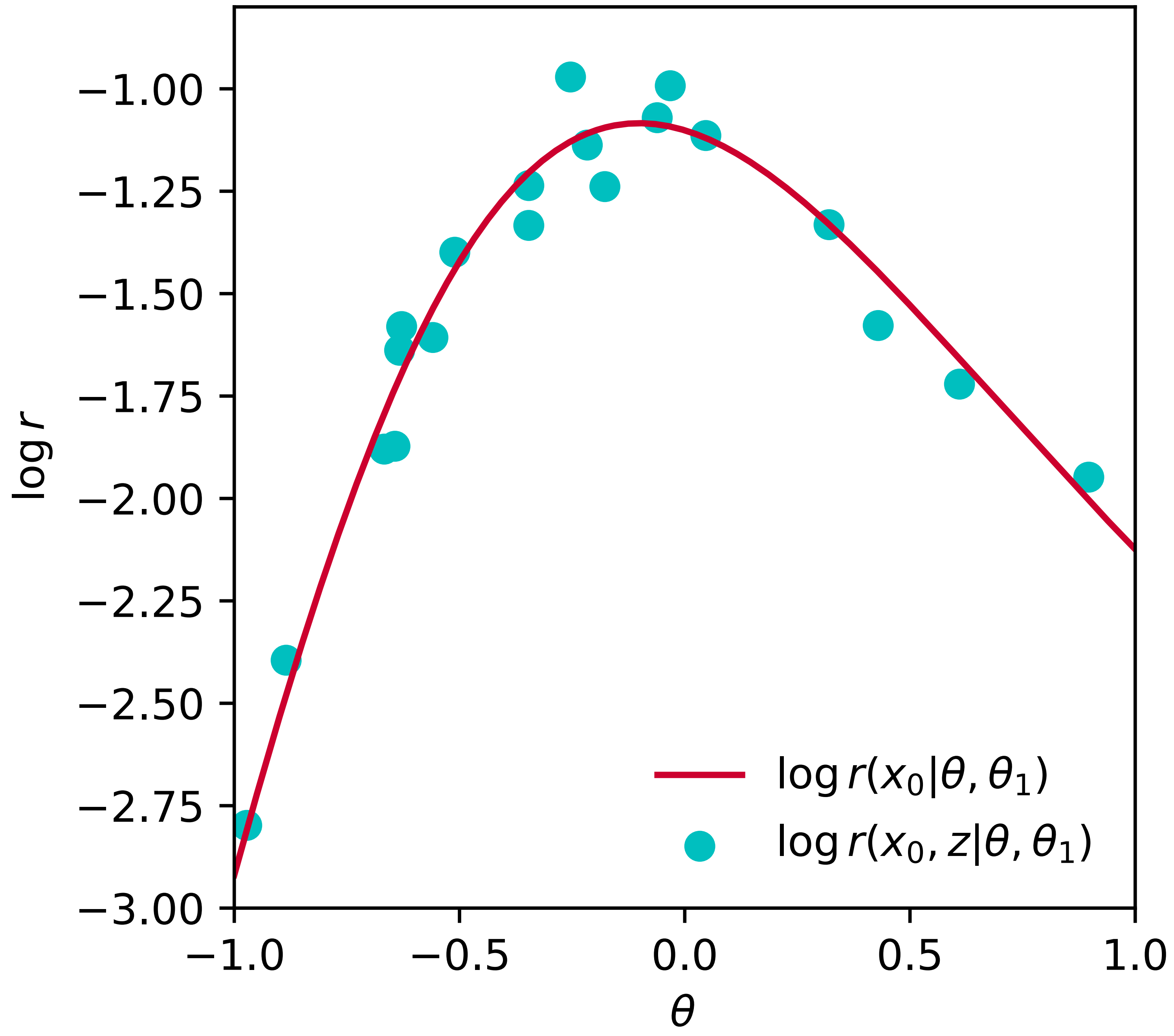

Parameters θ
Observables
x
Simulator
Latent z
$r(x, z|\theta)$
$t(x, z|\theta)$
Augmented data
$\arg\min_g L[g]$
Approximate likelihood ratio
$\hat{r}(x|\theta)$
θ_j
θ_i
1. Simulation
2. Machine Learning
3. Inference

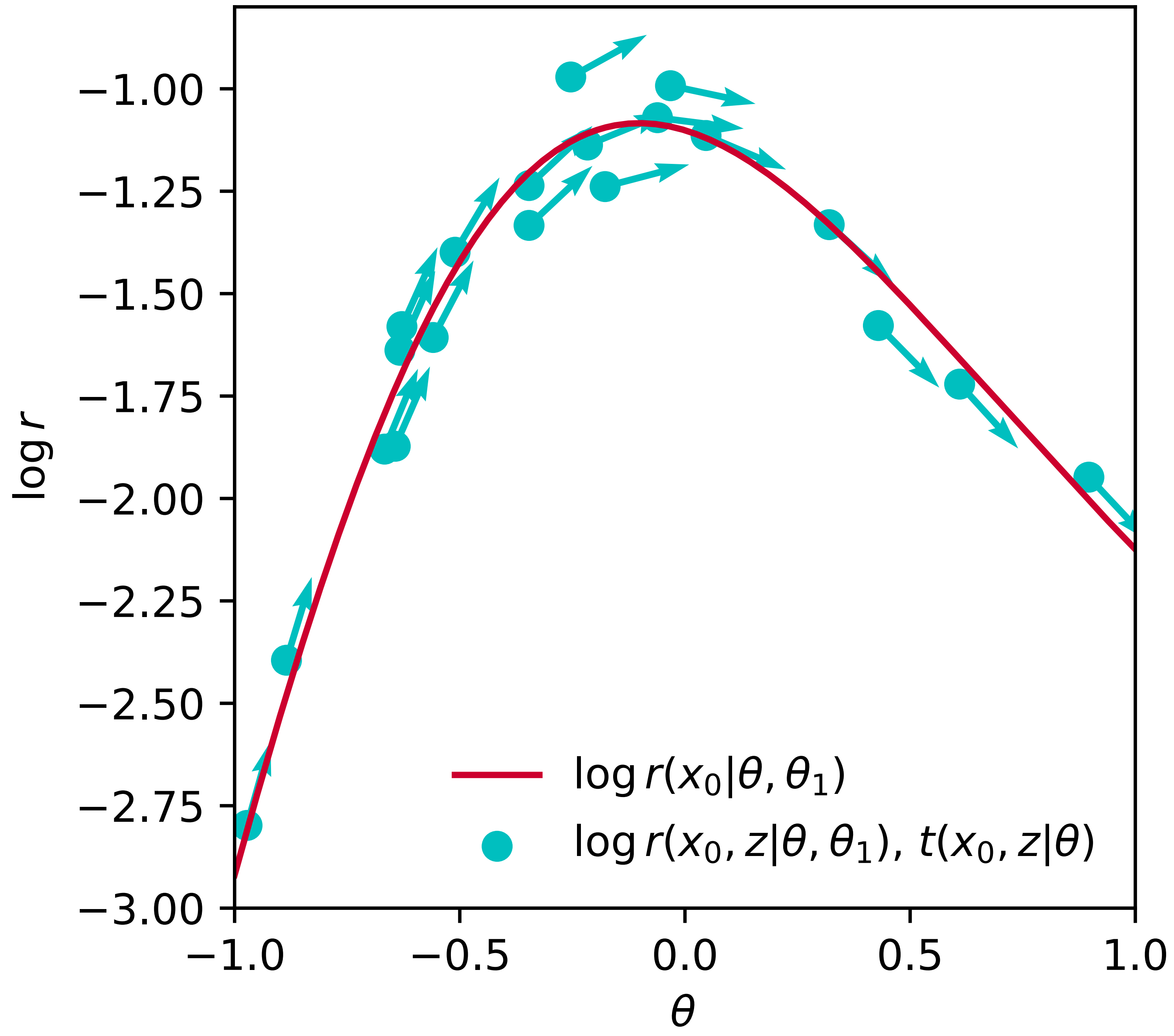


Parameters θ
Observables x
Simulator
Latent z
$r(x, z|\theta)$
$t(x, z|\theta)$
Augmented data
$\arg\min_g L[g]$
Approximate likelihood ratio
$\hat{r}(x|\theta)$
θ_j
θ_i
1. Simulation
2. Machine Learning
3. Inference

One more piece: the score

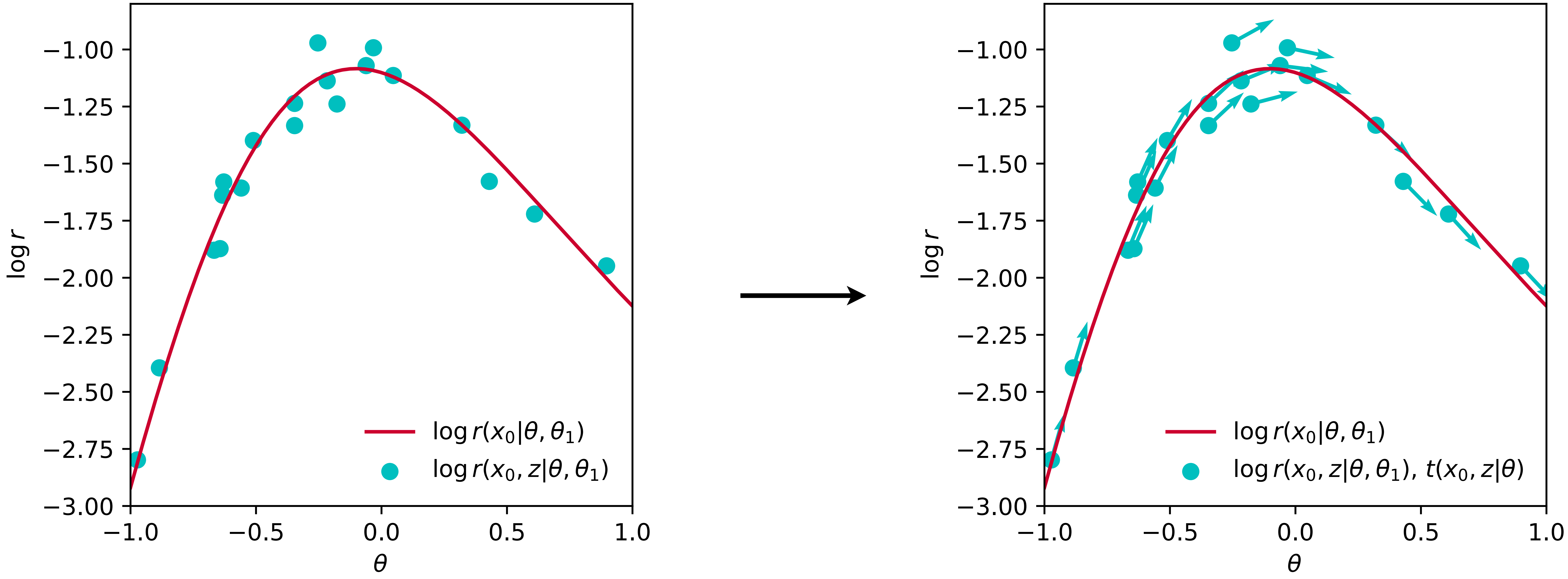
- Knowing derivative often helps fitting:





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

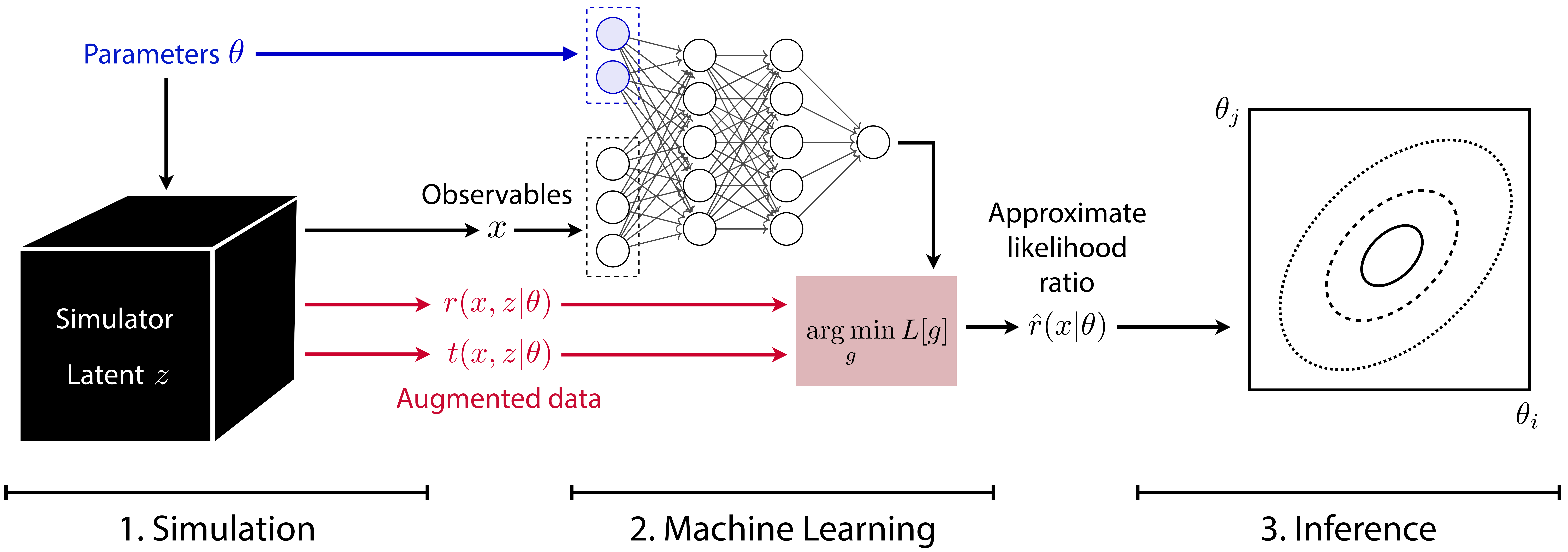
$$L_t[\hat{t}(x|\theta_0)] = \int \mathrm{d}x \int \mathrm{d}z \;\; p(x, z|\theta_0) \, \Big[\Big(\hat{t}(x|\theta_0) - t(x, z|\theta_0)\Big)^2\Big].$$

One can show it is minimized by

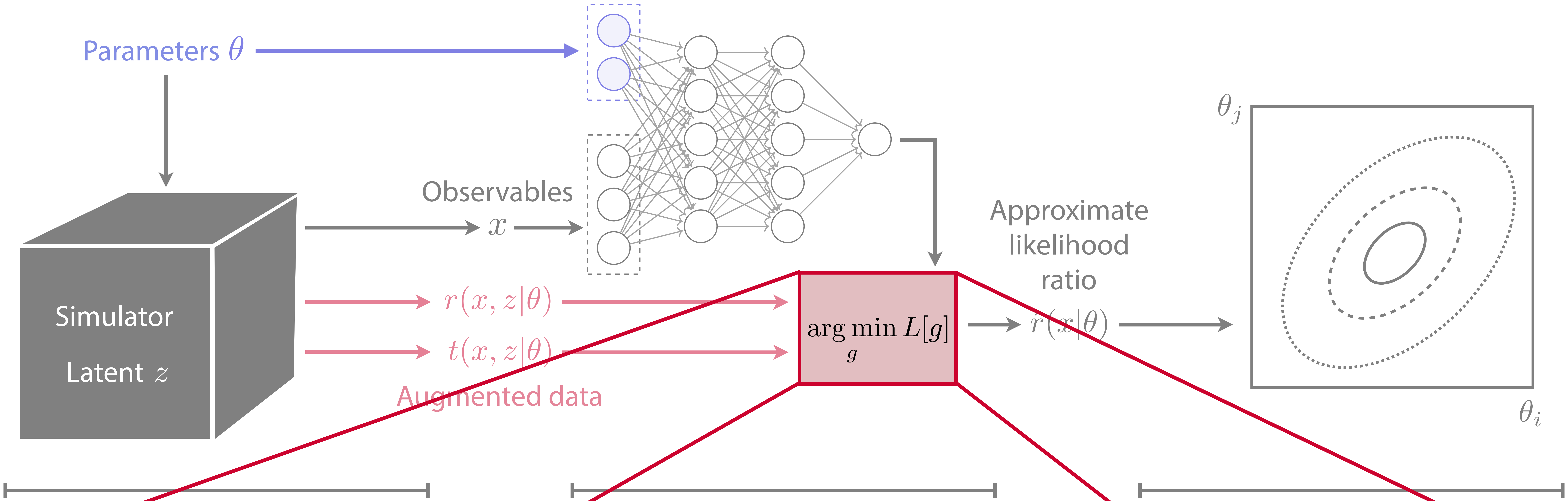
$$t(x|\theta_0) = \underset{\hat{t}(x|\theta_0)}{\arg\min}\, L_t[\hat{t}(x|\theta_0)].$$

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)

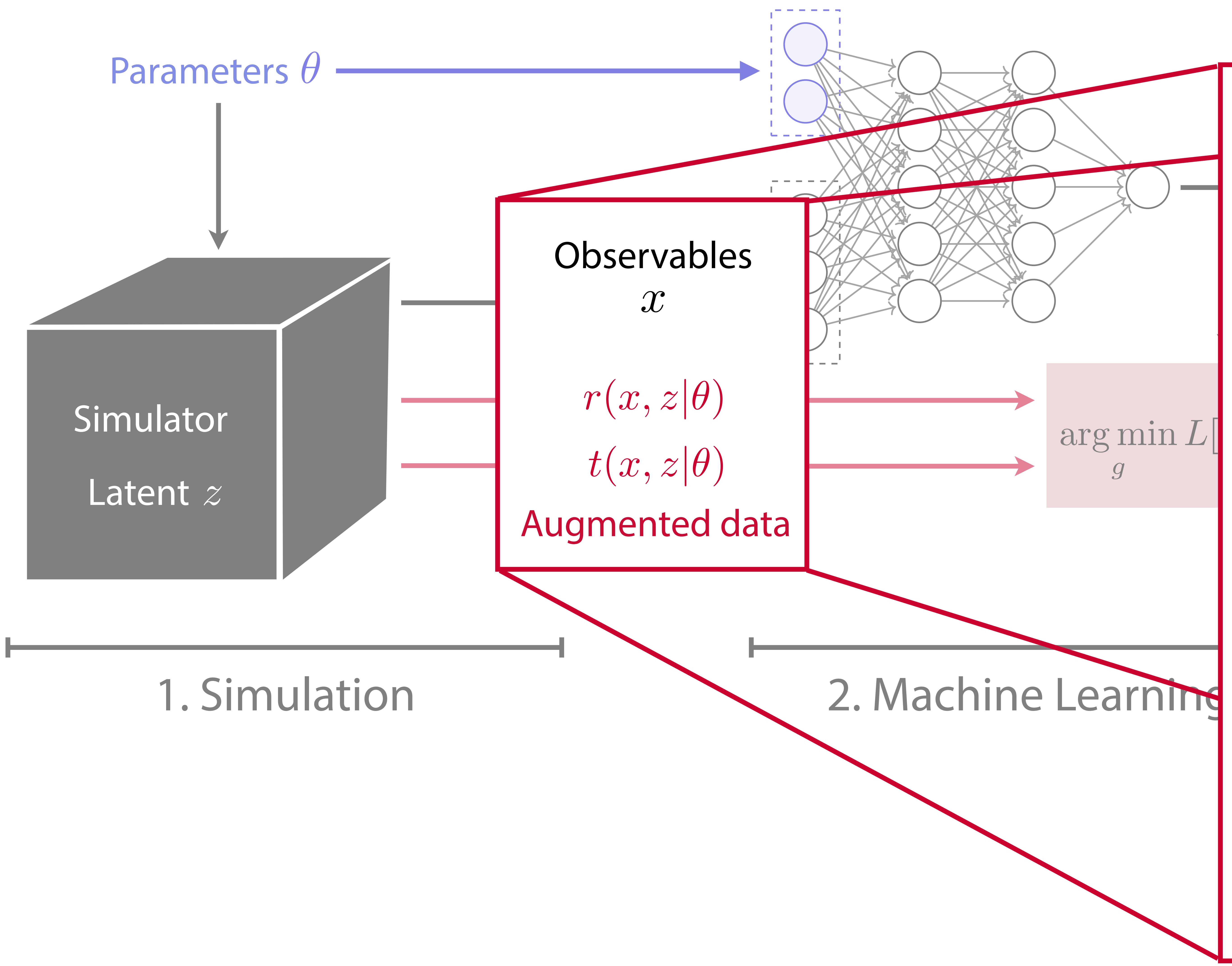


Combining ratio and score information:

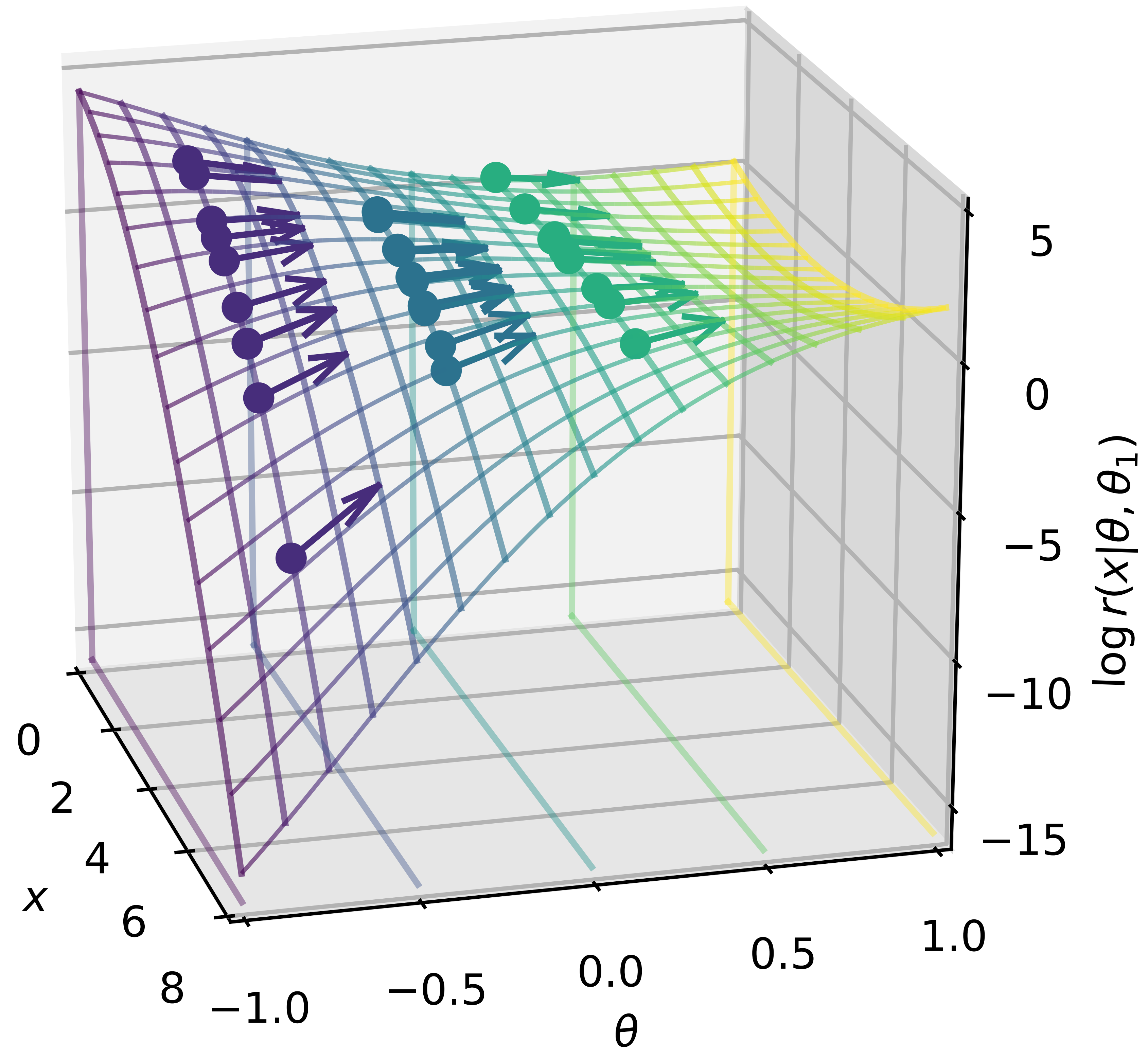
- If ratio estimator is differentiable, we can calculate the estimated score $\hat{t}(x|\theta_0) = \nabla_{\theta_0} \log \hat{r}(x|\theta_0, \theta_1)$
- Minimize combined loss function:

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

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



RASCAL combines three orthogonal pieces of information



A simpler variation:

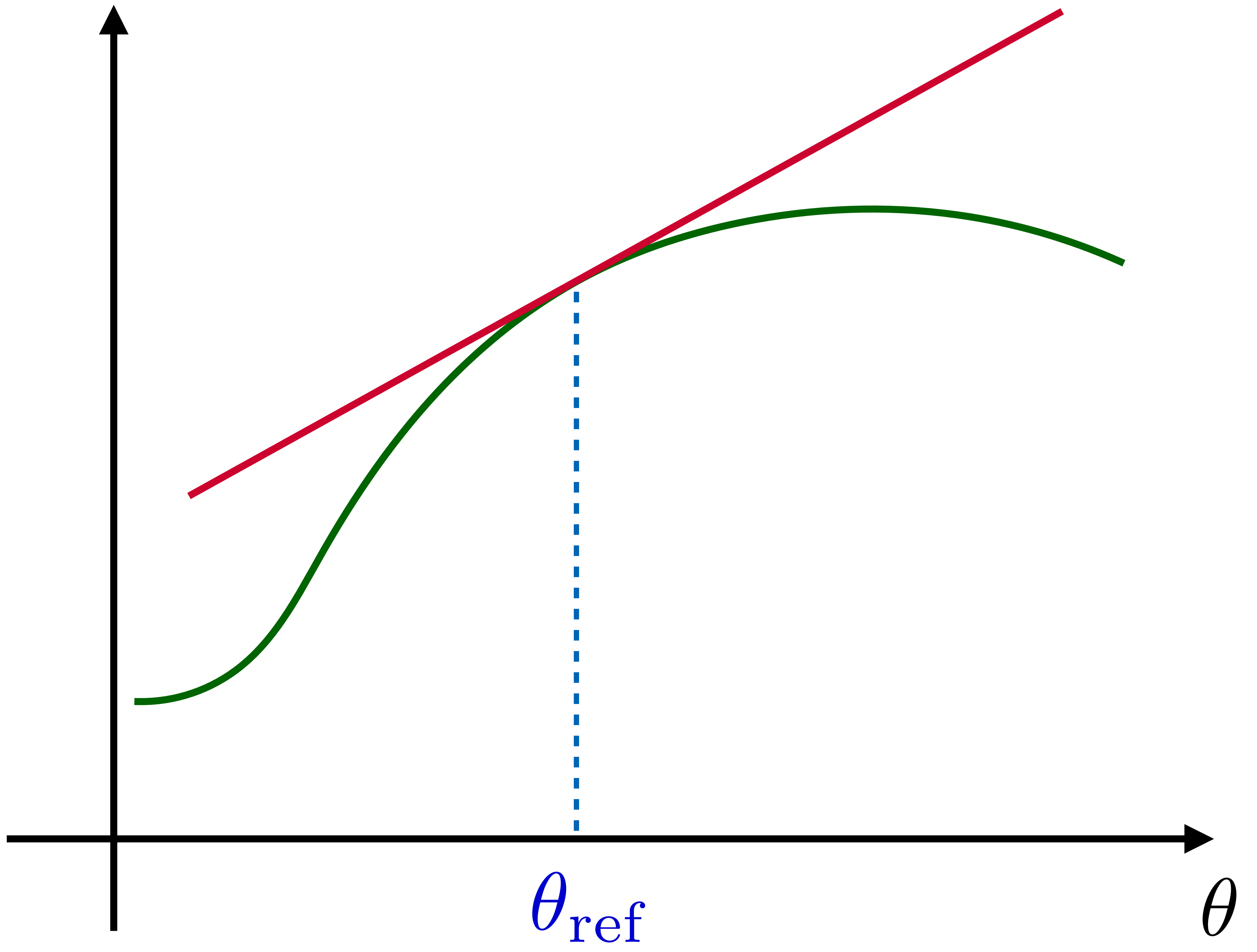
Learning optimal summary statistics

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:

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

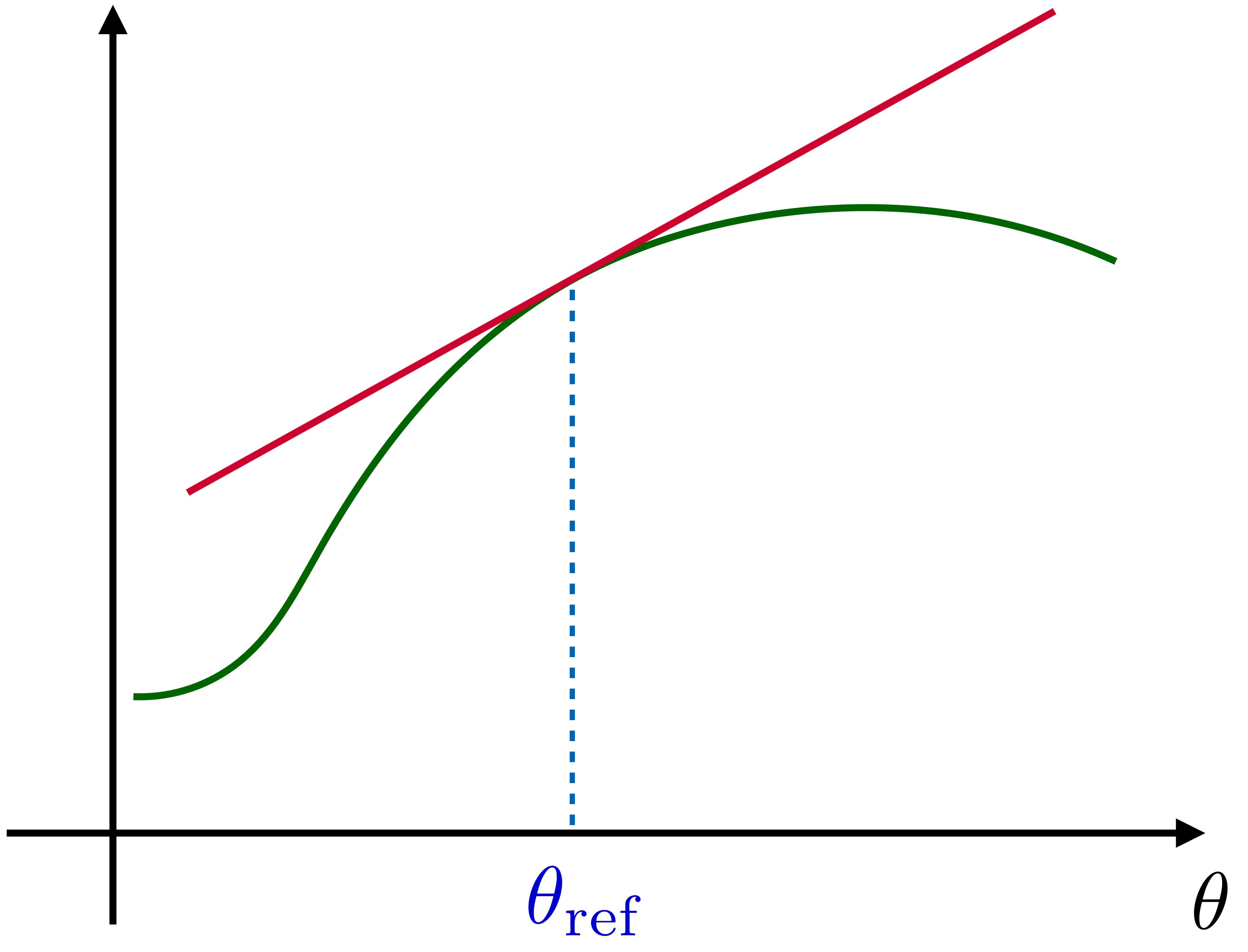


The local model

[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}:

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

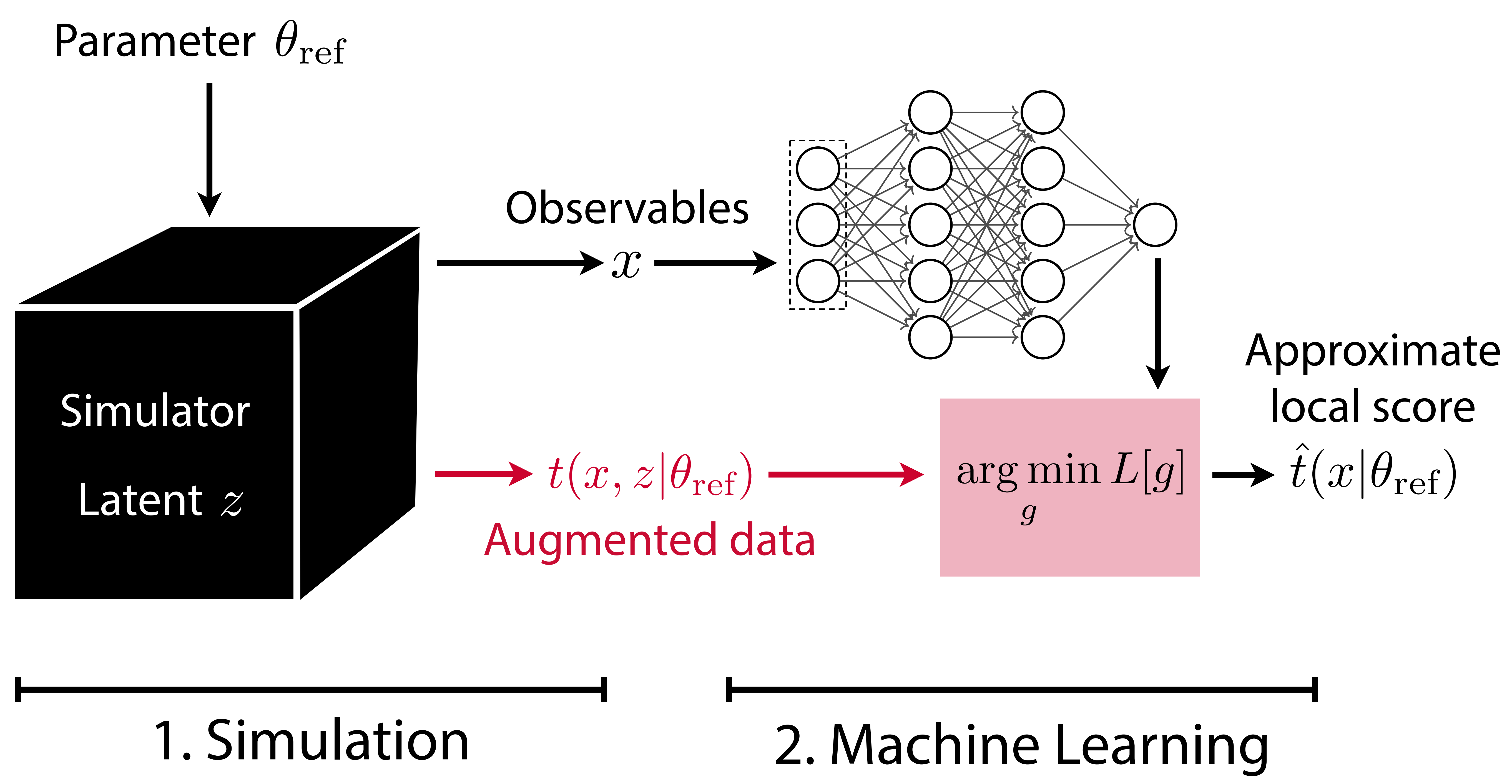


⇒ In the neighborhood of θ_{ref} (e.g. close to the SM), the score vector $t(x|\theta_{\text{ref}})$ is the sufficient statistics: it contains all information on θ

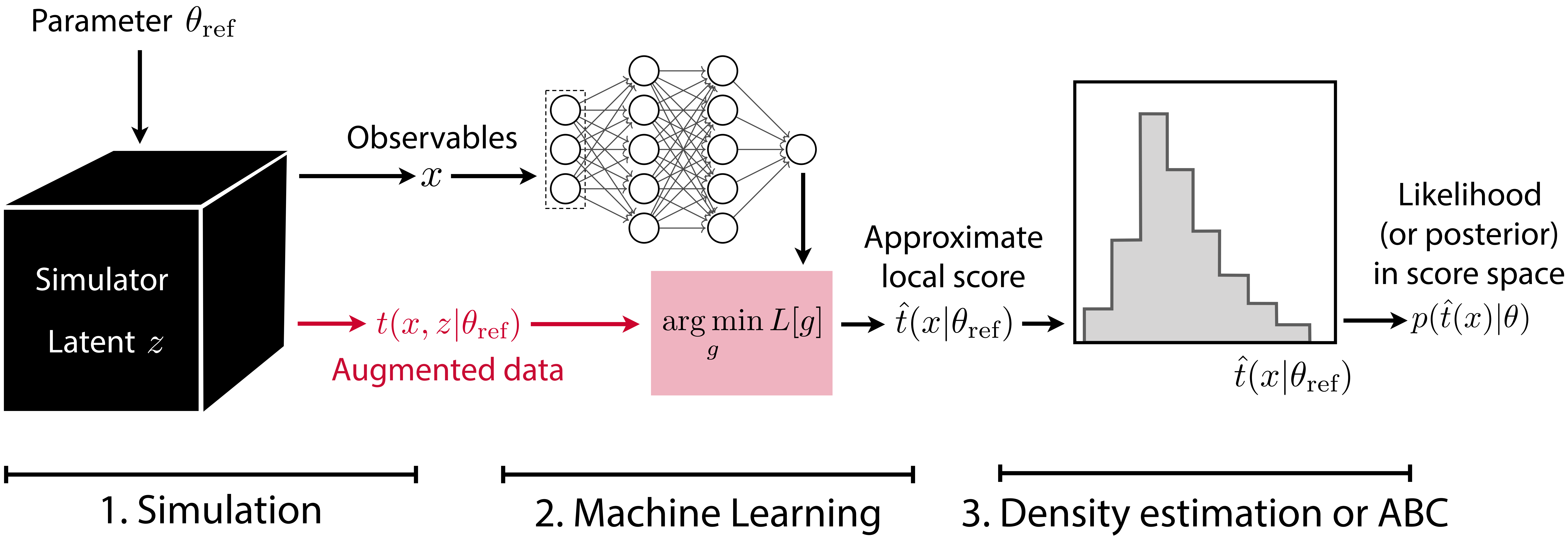
In this part of parameter space, knowing $t(x|\theta_{\text{ref}})$ is just as powerful as knowing the full function $\log p(x|\theta)$

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

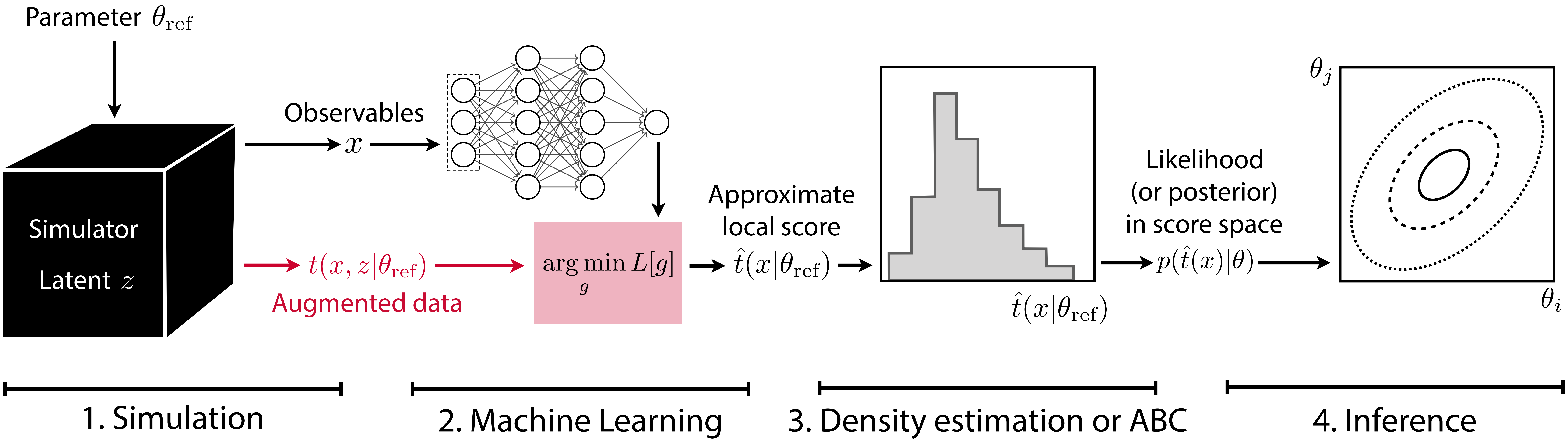
SALLY (Score approximates likelihood locally)



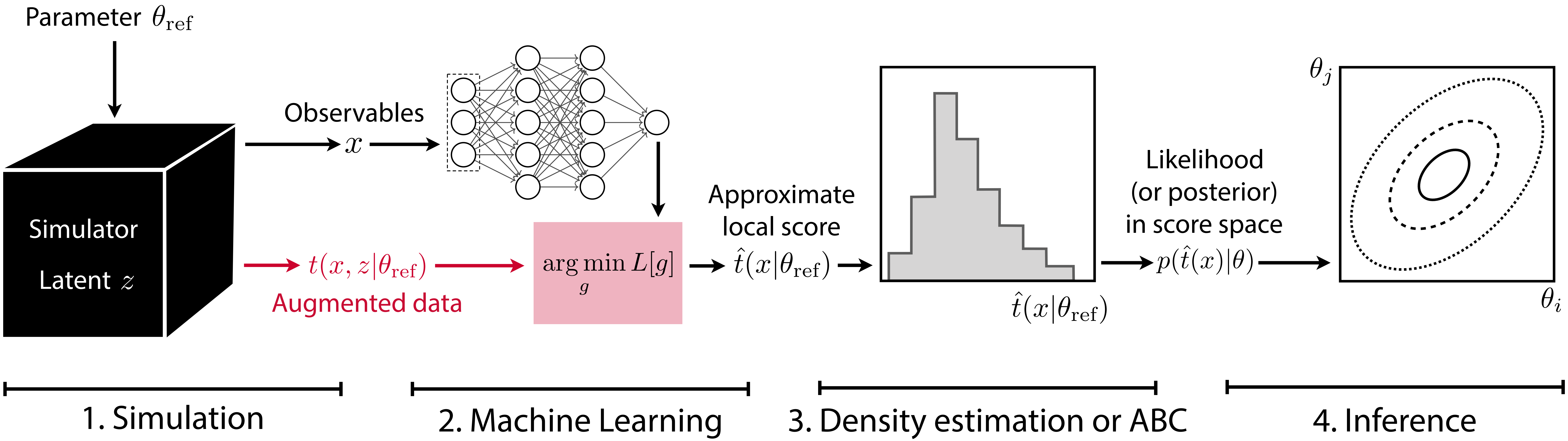
SALLY (Score approximates likelihood locally)



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

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.	

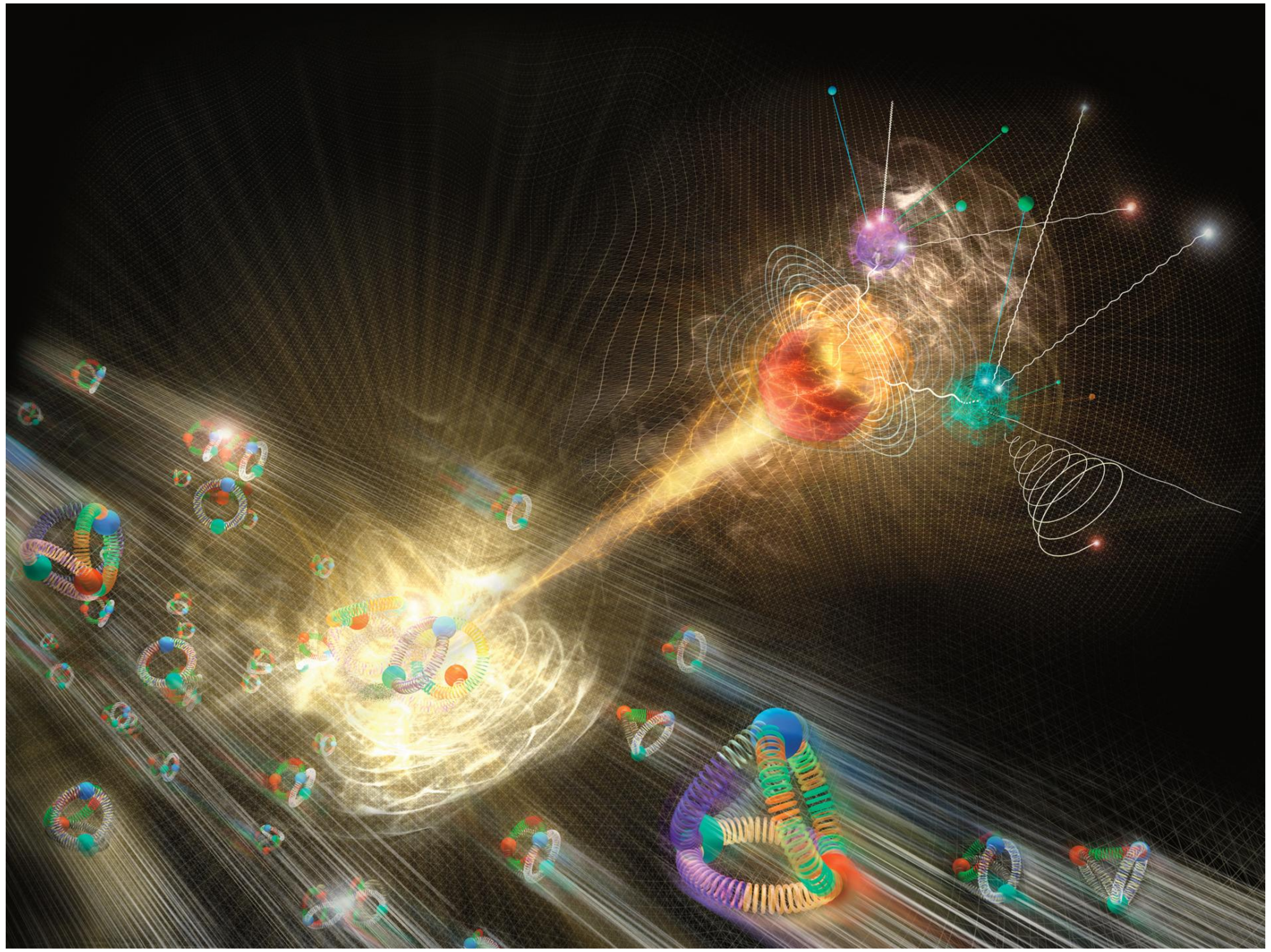
Particle physics example

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!

- Precision 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?)

The legacy of the LHC

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

- Precision 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?)

- These measurements are difficult!
  - Many parameters
  - Many observables
  - Subtle kinematic effects
  - Intractable likelihood

Particle physics processes

Theory parameters
θ

Evolution

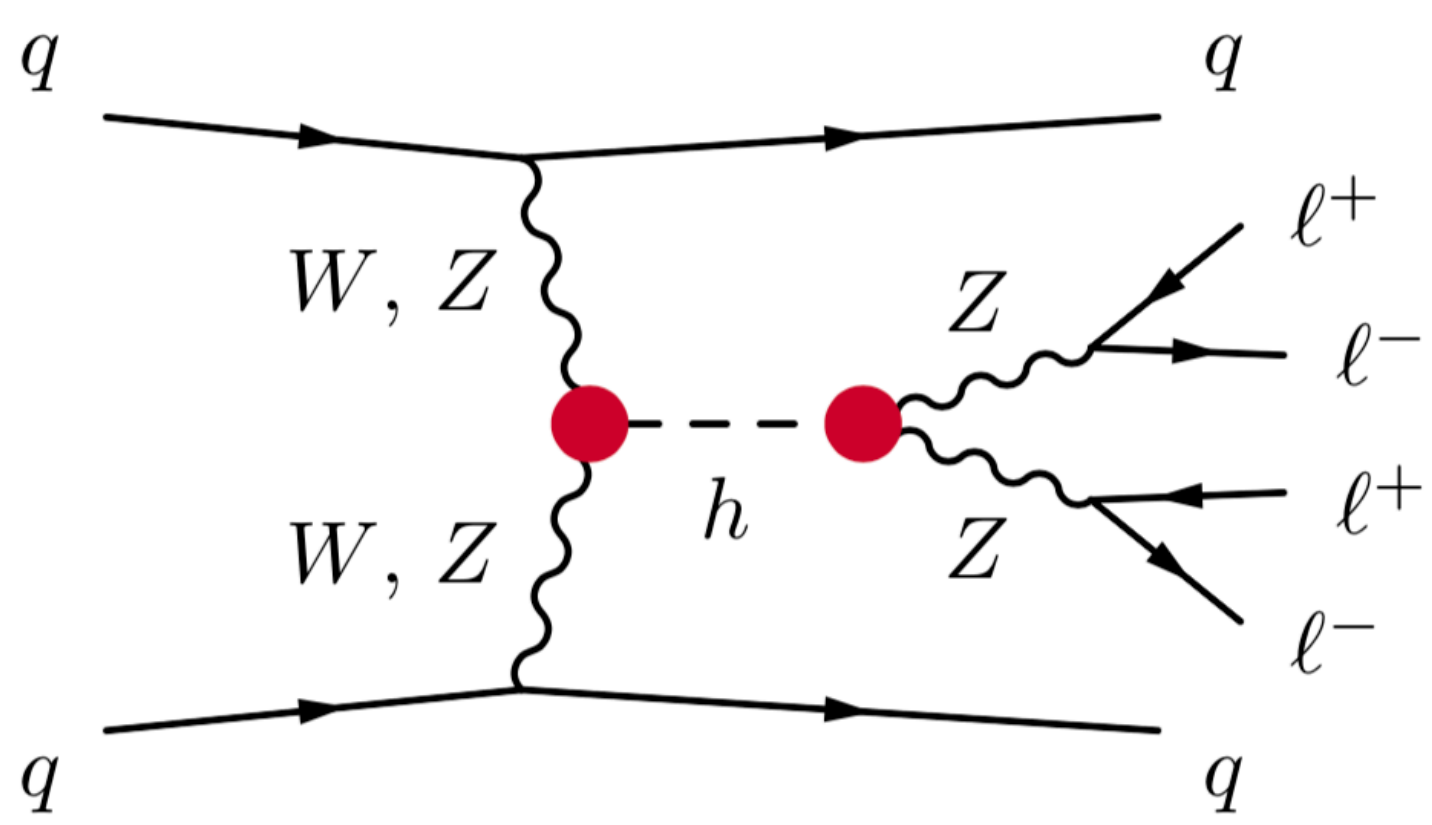
Particle physics processes

Latent variables

Parton-level momenta

Theory parameters

$$z_p \longleftarrow \theta$$



Evolution

Particle physics processes

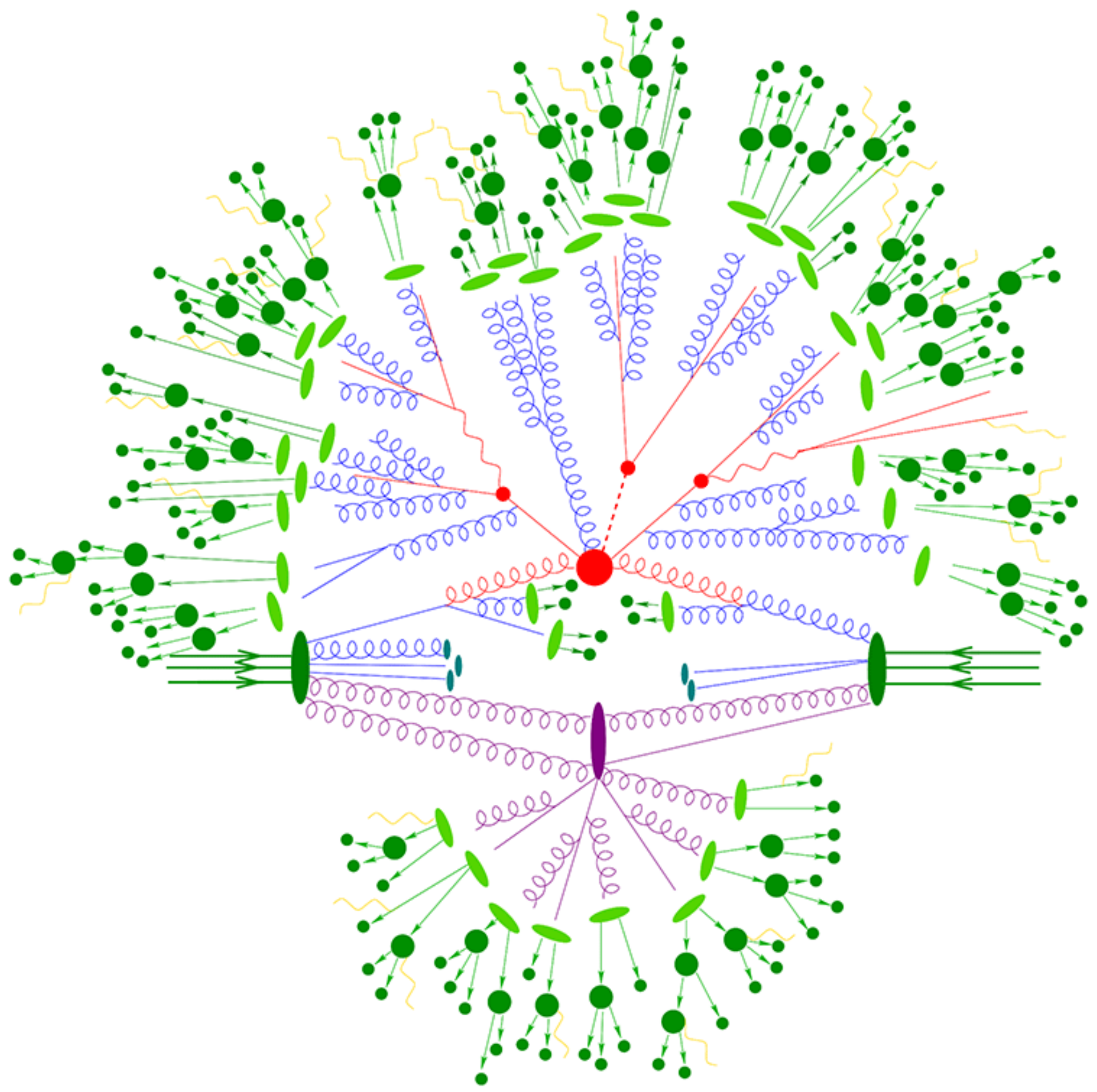
Latent variables

Shower splittings

Parton-level momenta

Theory parameters

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



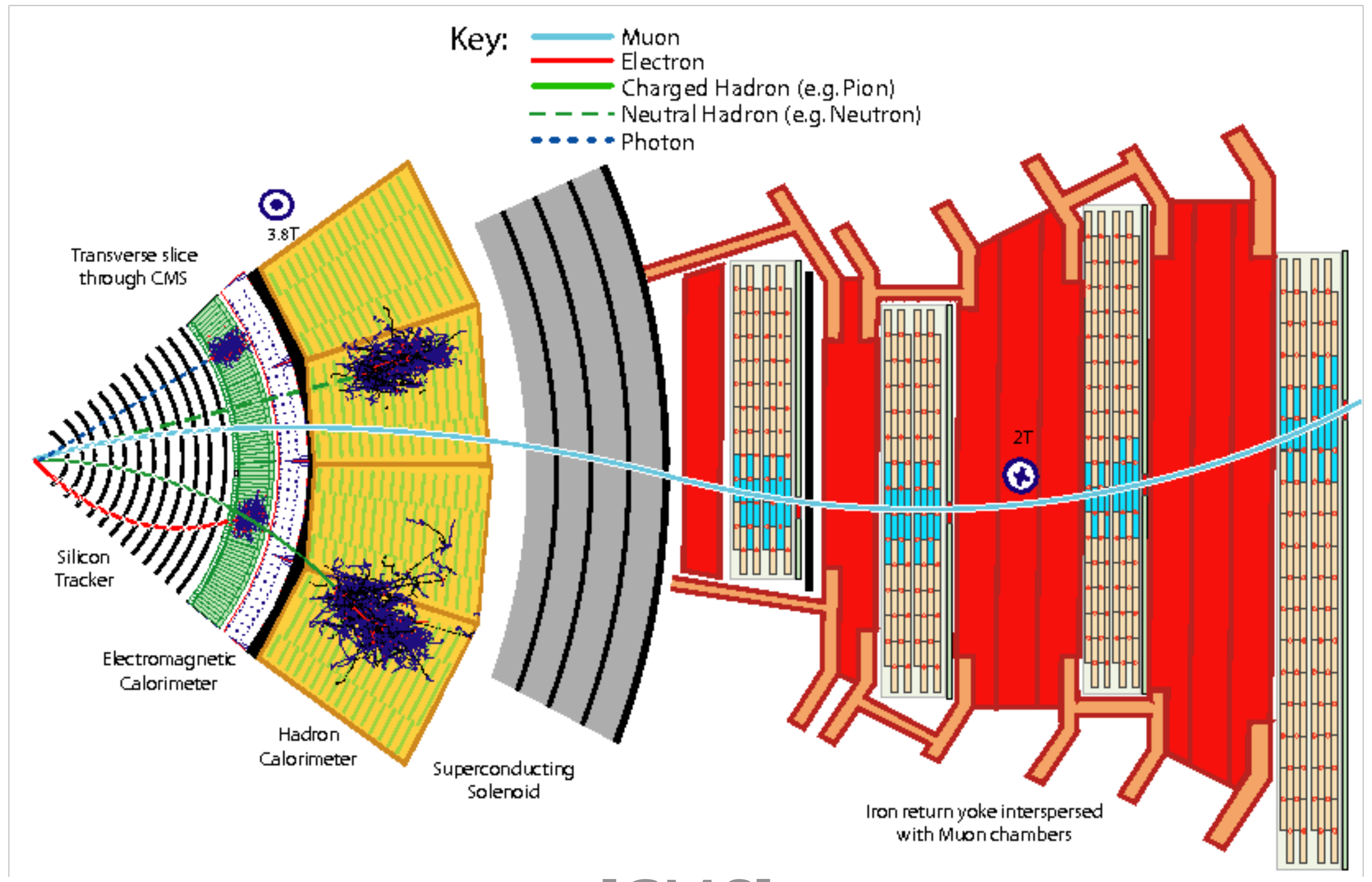
[F. Krauss]

Evolution

Particle physics processes

Latent variables

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



[CMS]

$\longleftarrow$

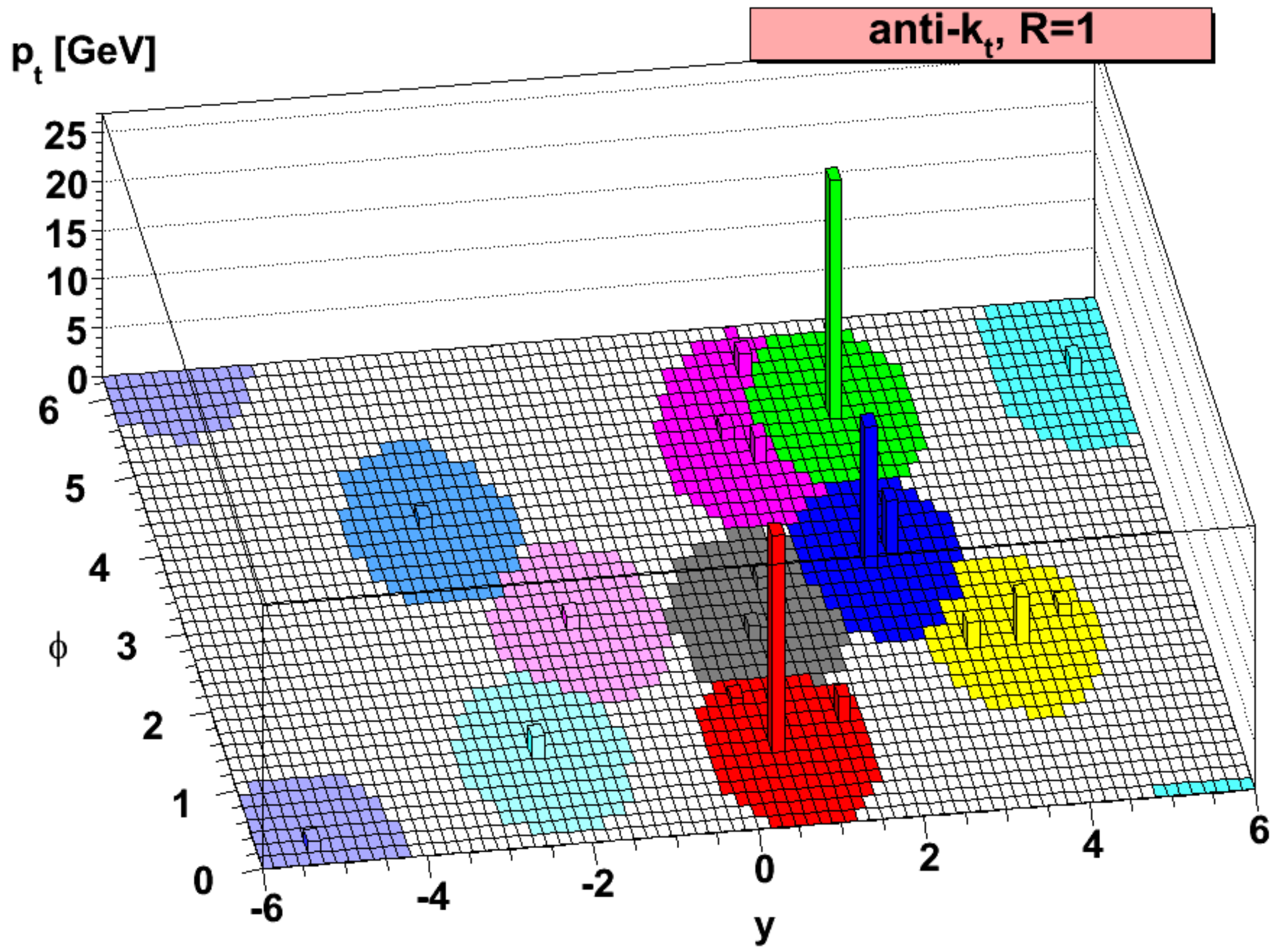
Evolution

Particle physics processes

Latent variables

Observables		Detector interactions	Shower splittings	Parton-level momenta	Theory parameters

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



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

Evolution

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

GEANT4
A SIMULATION TOOLKIT

MADGRAPH5_aMC@NLO

Prediction (simulation)

Particle physics processes

Latent variables

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

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

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

Inference

Particle physics processes

Latent variables

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

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

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

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

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

Inference

Mining gold from LHC simulations

Latent variables

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

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

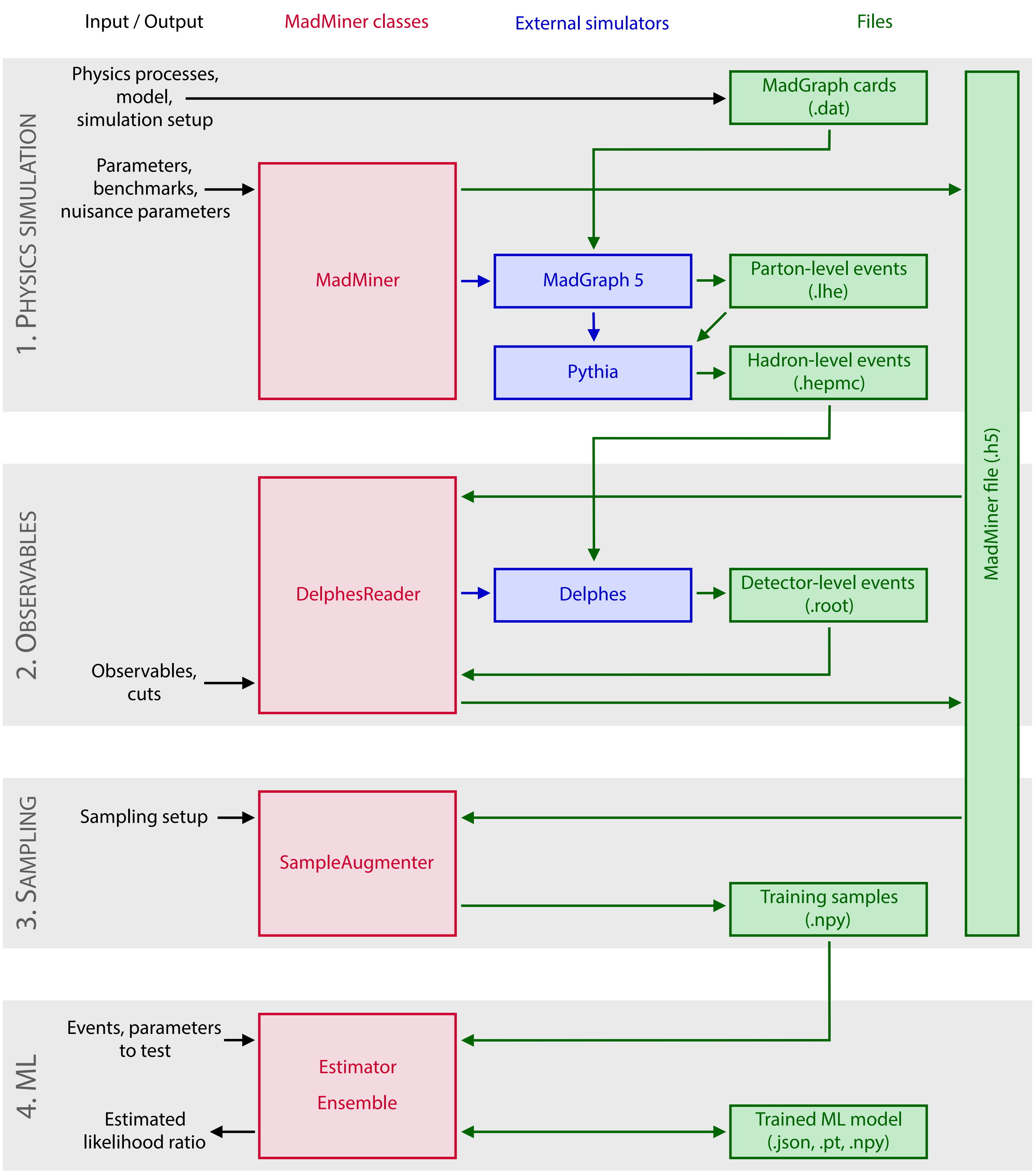
Parton-level likelihood is given by matrix element and can be evaluated!

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

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

Automization

We have developed MadMiner, which makes it straightforward to apply the new techniques to LHC problems

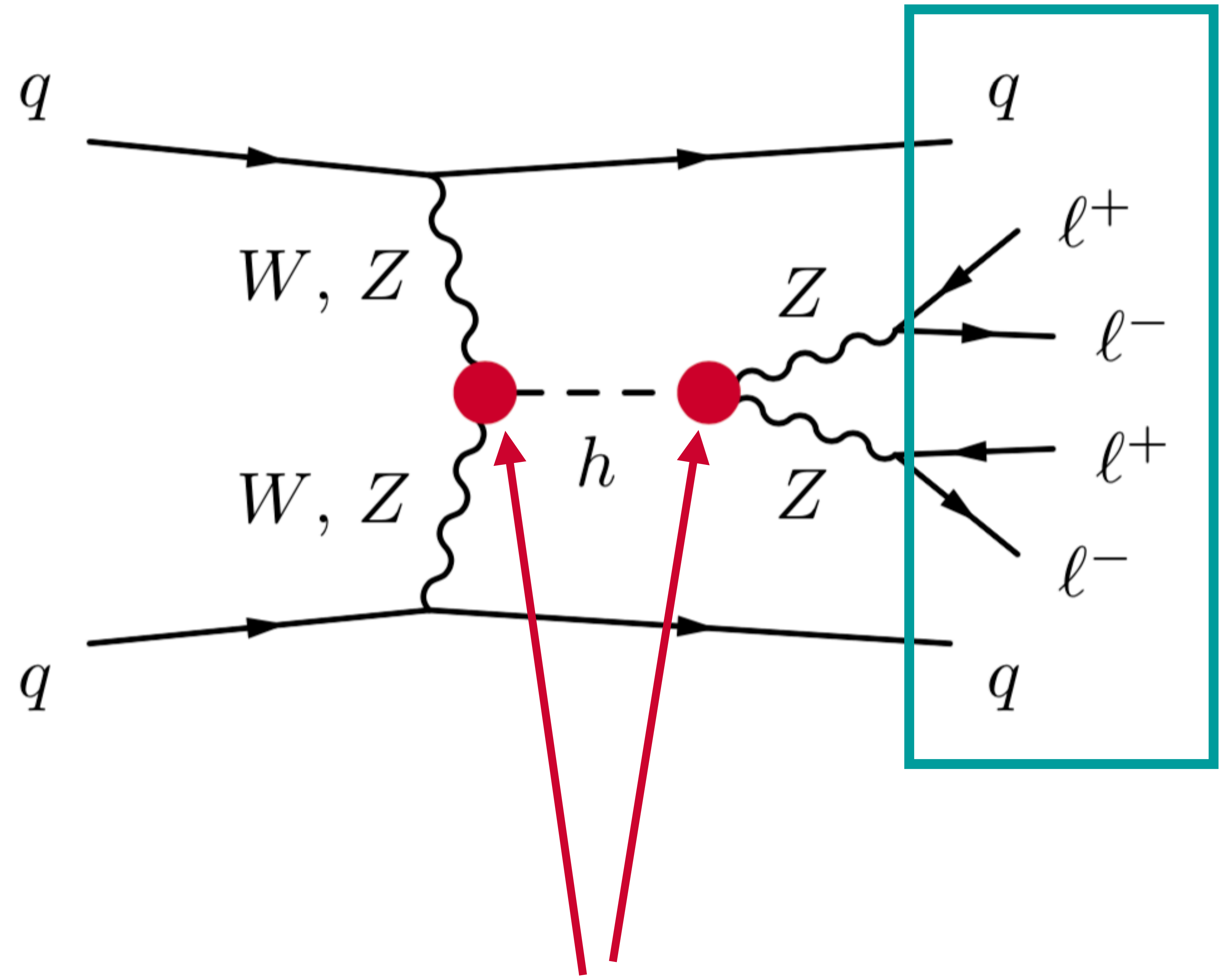


- "Mining gold" from LHC simulations — MadGraph, Pythia, Delphes…
- Likelihood / ratio / score / Fisher info estimation — RASCAL, SALLY & co
- Supports most aspects of real-life analyses:
  - Almost any theory model — UFO standard
  - Parameter morphing — for EFT-like situations
  - Hard process, shower, detector simulation — Currently only LO
  - Reducible + irreducible backgrounds
  - Systematic uncertainties — from PDF variation or scale variation
- Well-documented Python module

github.com/johannbrehmer/madminer   madminer.readthedocs.io   `pip install madminer`

Proof of concept: Higgs production in weak boson fusion

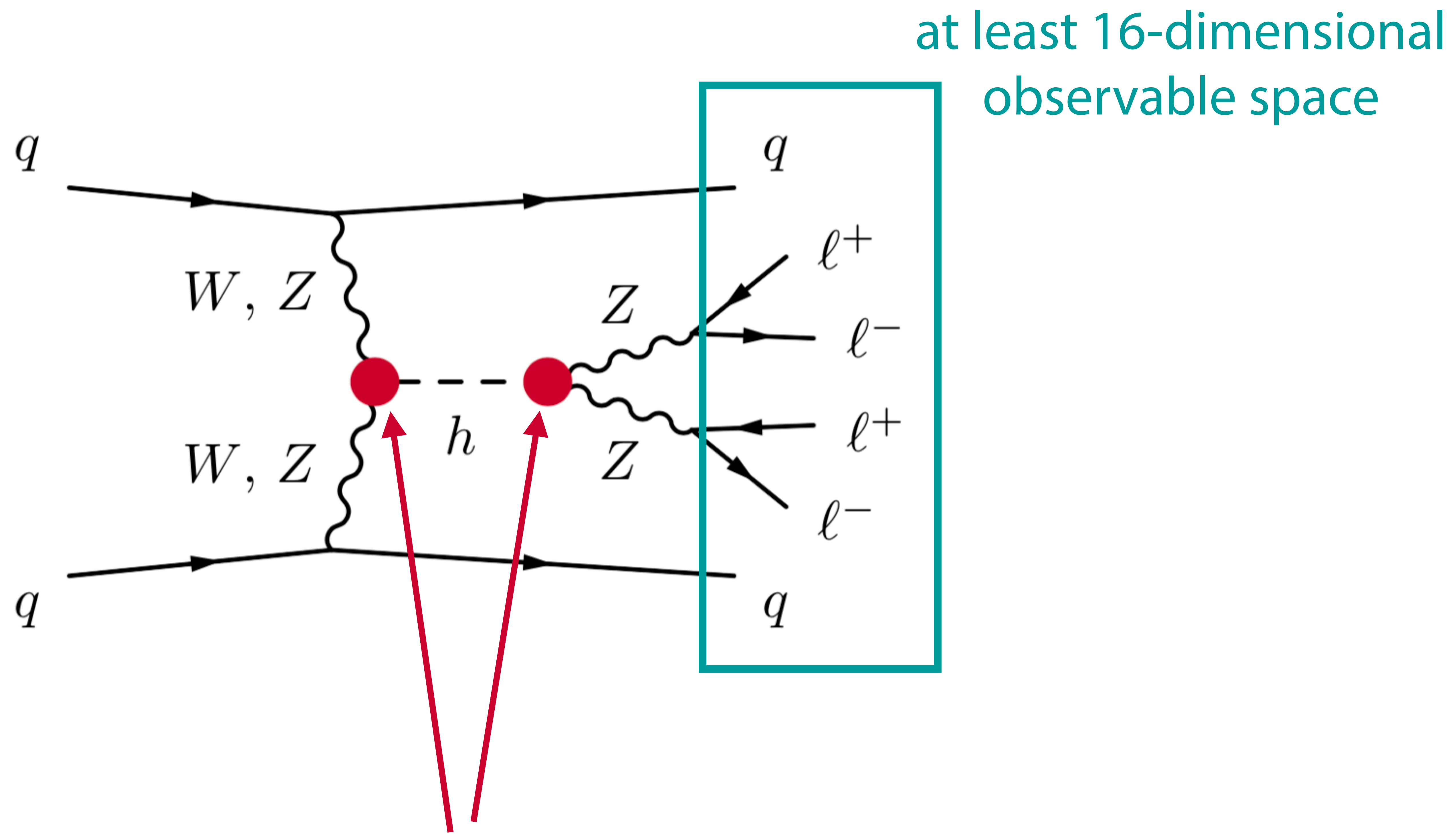
at least 16-dimensional
observable space



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

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

Proof of concept: Higgs production in weak boson fusion



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

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

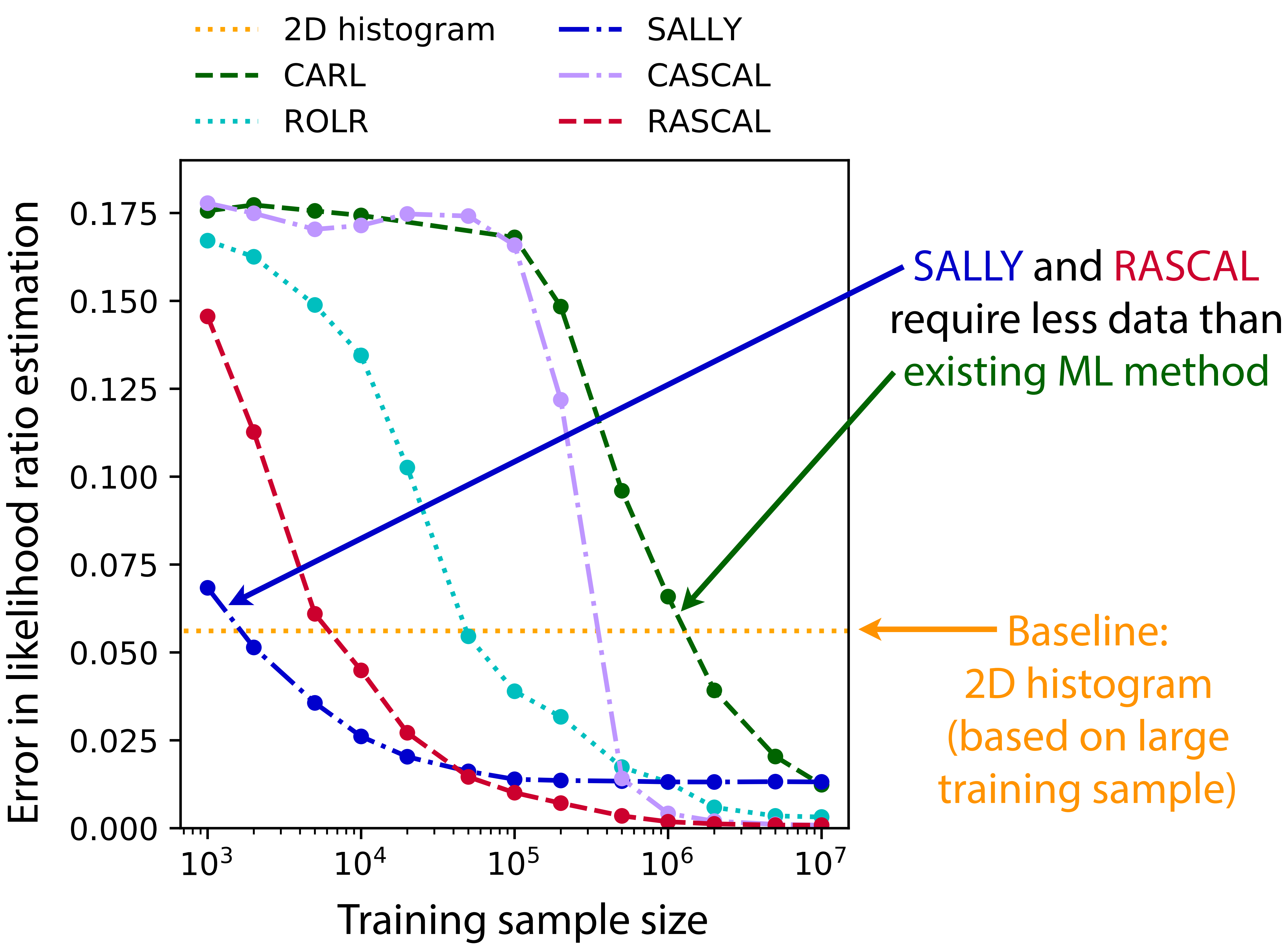
Goal: constrain the two EFT parameters

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

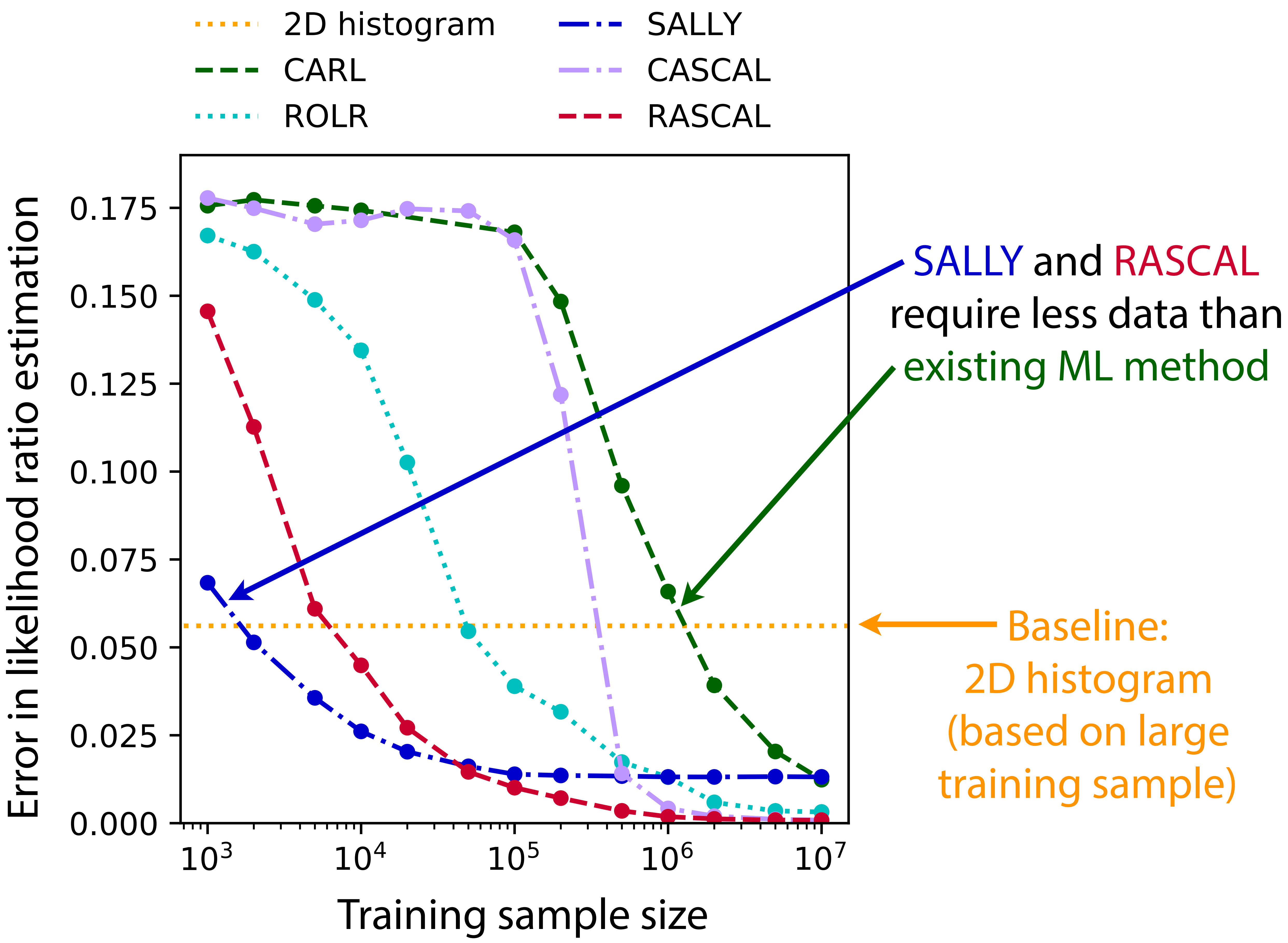
Two scenarios:

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

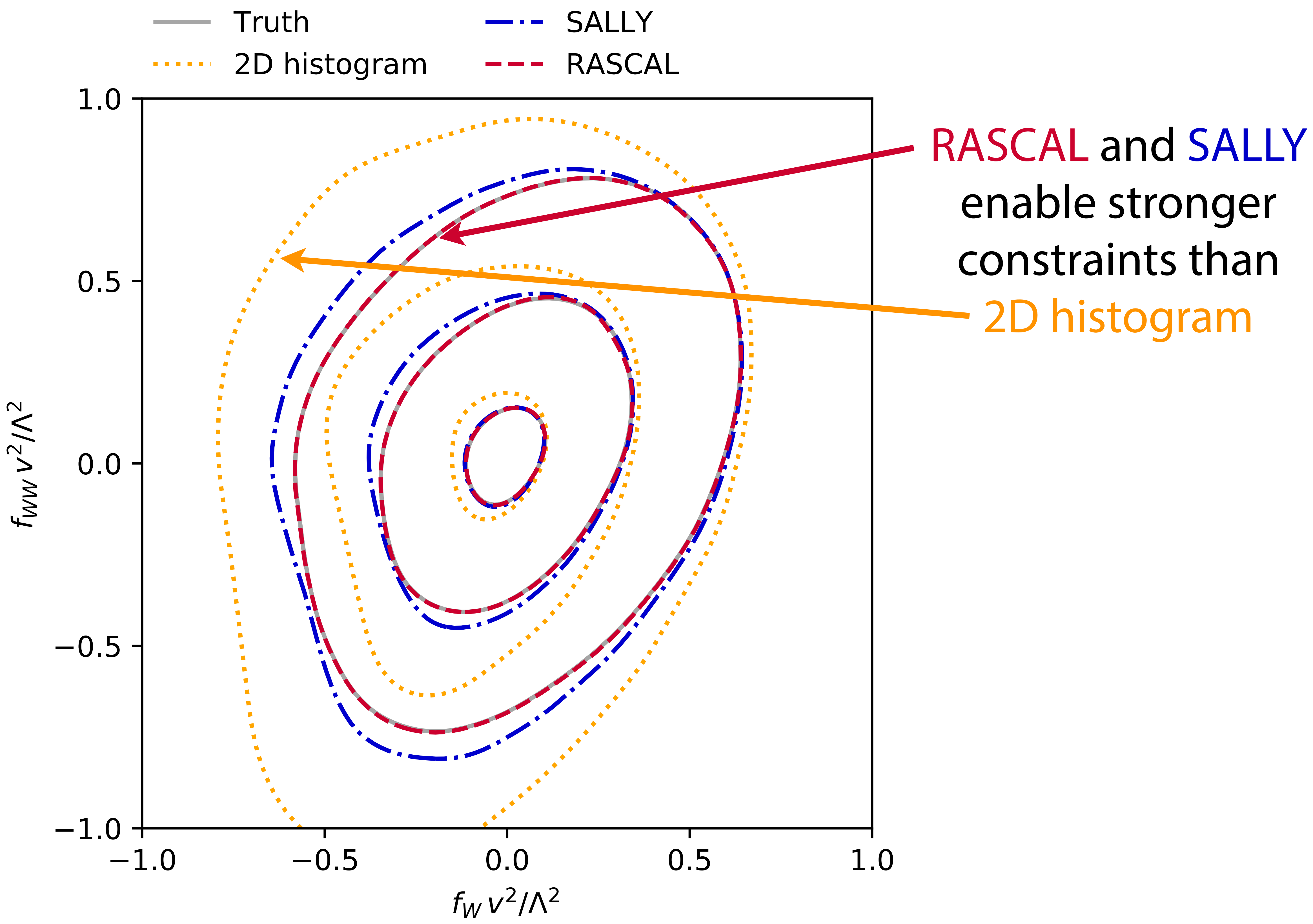
Stronger constraints with less training data



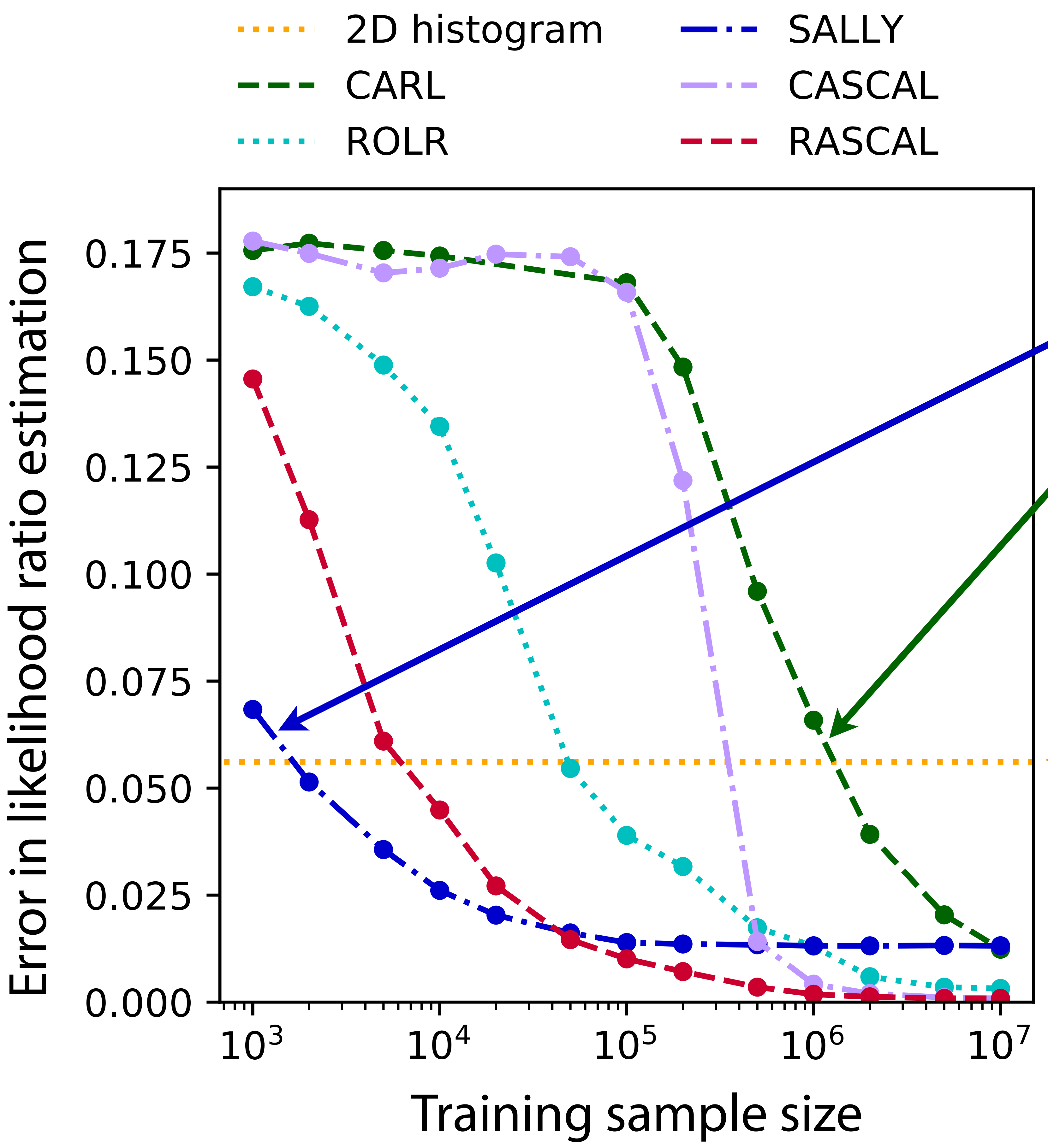
Stronger constraints with less training data



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



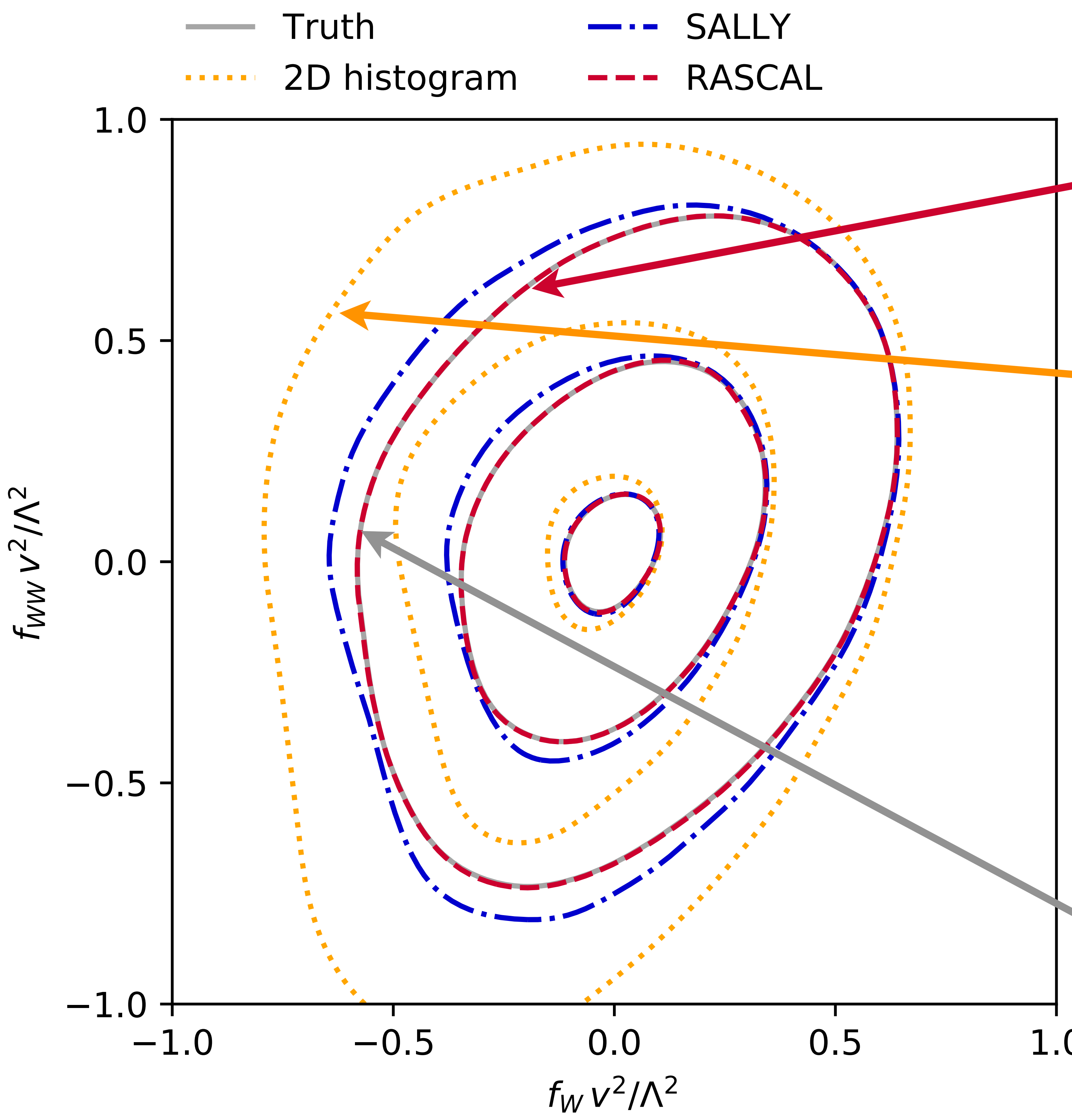
Stronger constraints with less training data



SALLY and RASCAL require less data than existing ML method

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

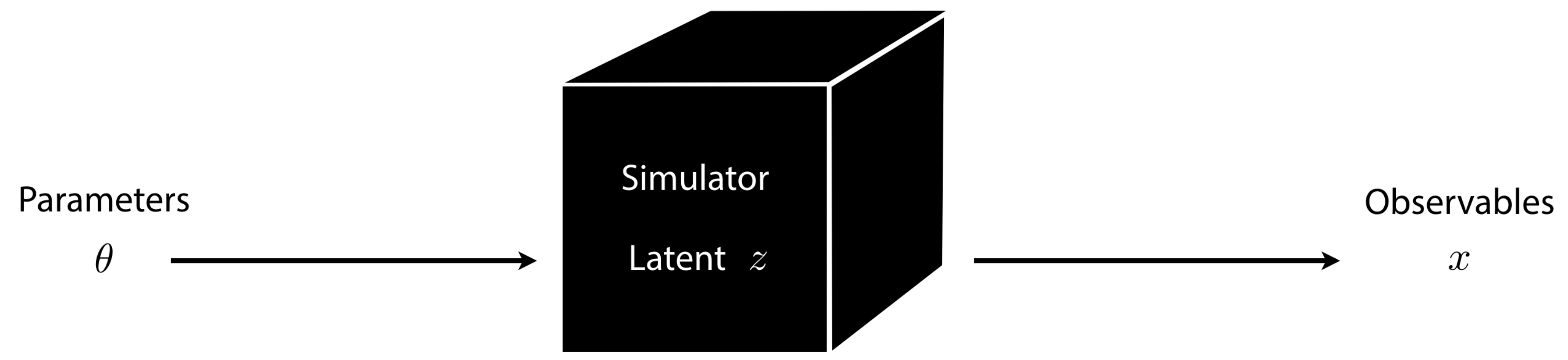
Expected exclusion limits at 68%, 95%, 99.7% CL



RASCAL and SALLY enable stronger constraints than 2D histogram

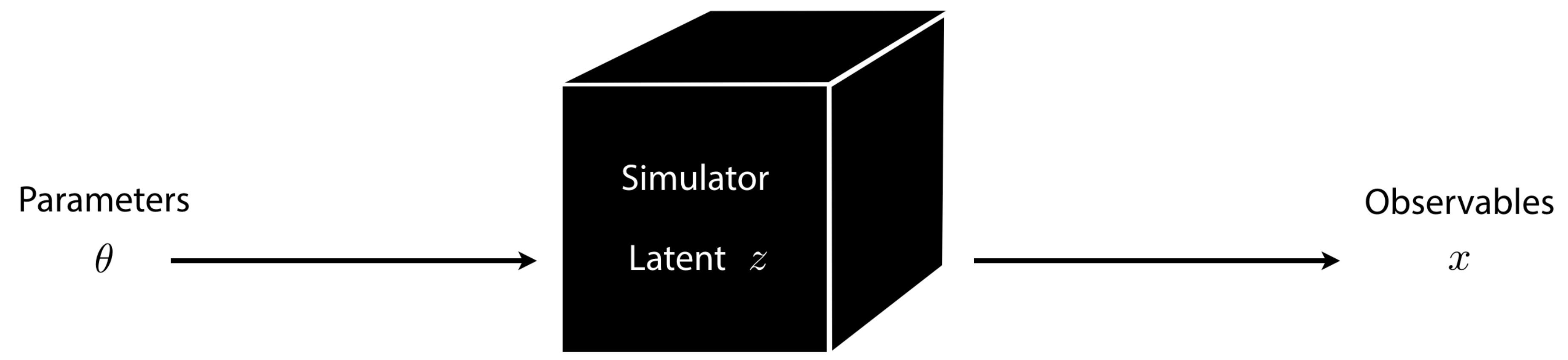
Limits from RASCAL indistinguishable from true likelihood

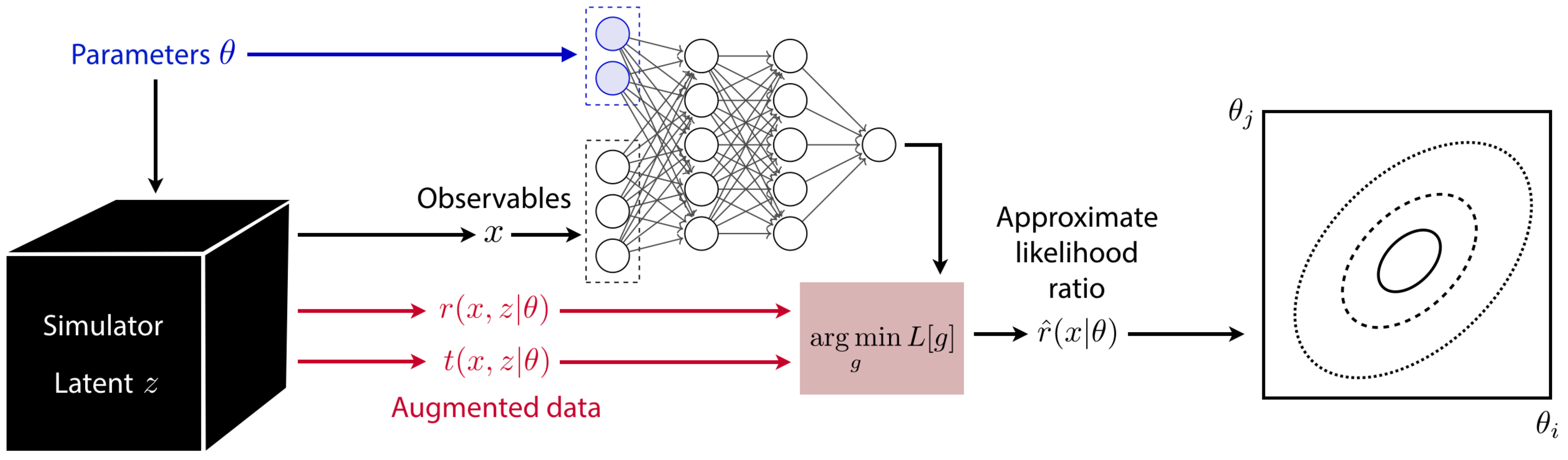
A new approach to simulator-based inference



- Many problems in modern science are based on simulations, “likelihood-free”

A new approach to simulator-based inference

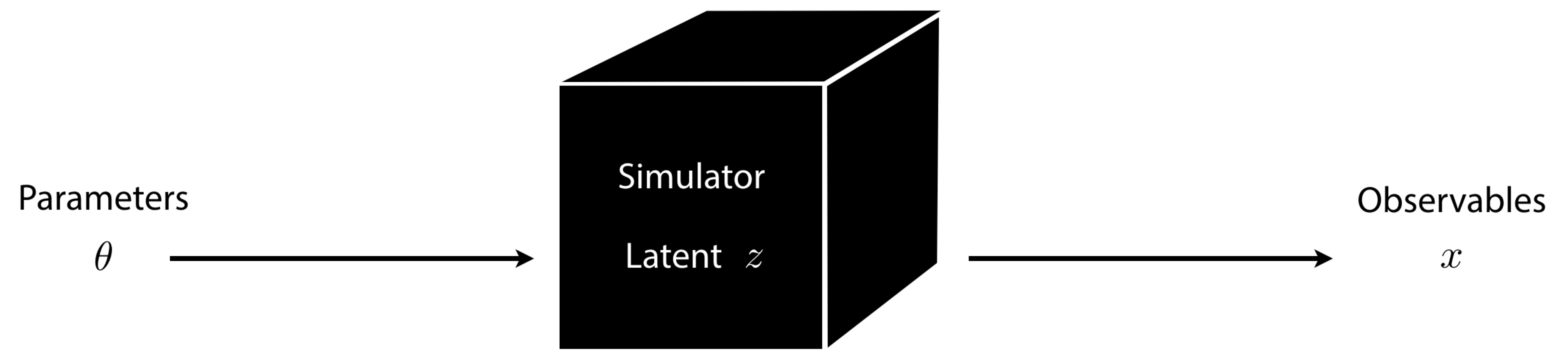


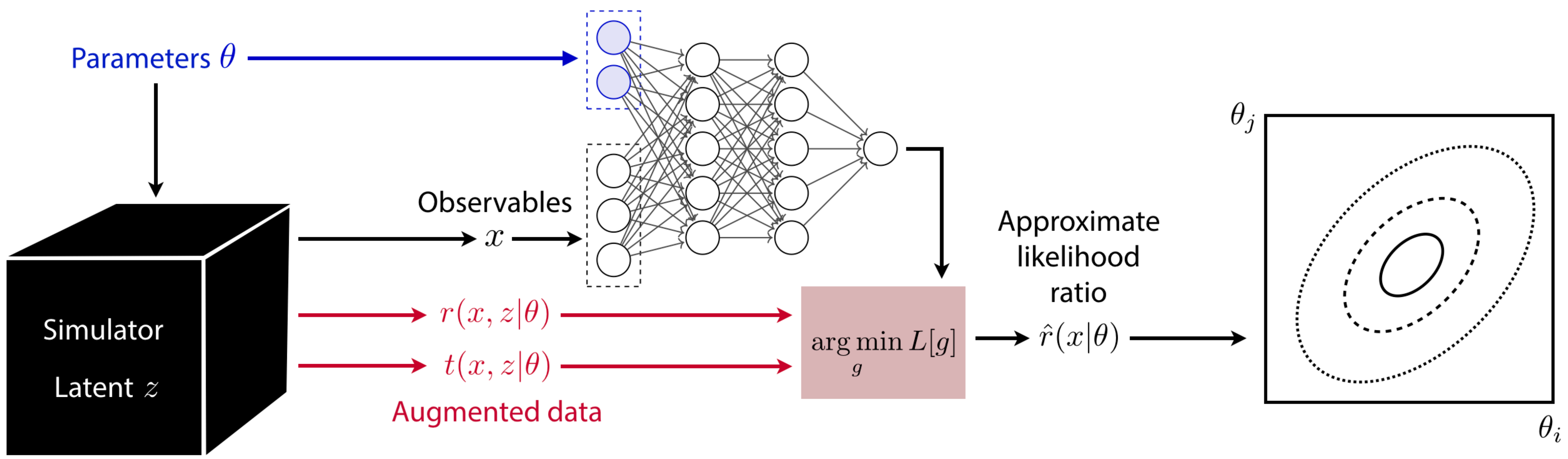


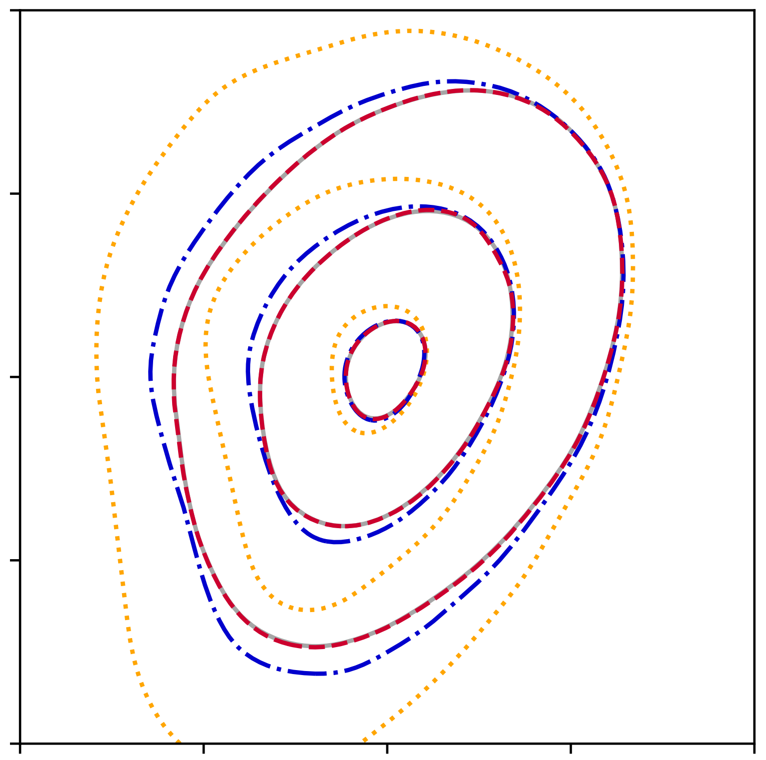
- Many problems in modern science are based on simulations, "likelihood-free"

- New inference techniques:
  Leverage insight into simulated process
  + power of machine learning

A new approach to simulator-based inference







- Many problems in modern science are based on simulations, "likelihood-free"
- New inference techniques:
  Leverage insight into simulated process
  + power of machine learning
- First application to LHC measurements:
  Stronger constraints with less simulations, automization with new tool MadMiner

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 [doi: 10.5281/zenodo.1489147]

Thanks to Kyle and Gilles for inspiring many slides!

Bonus material

Variational calculus

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

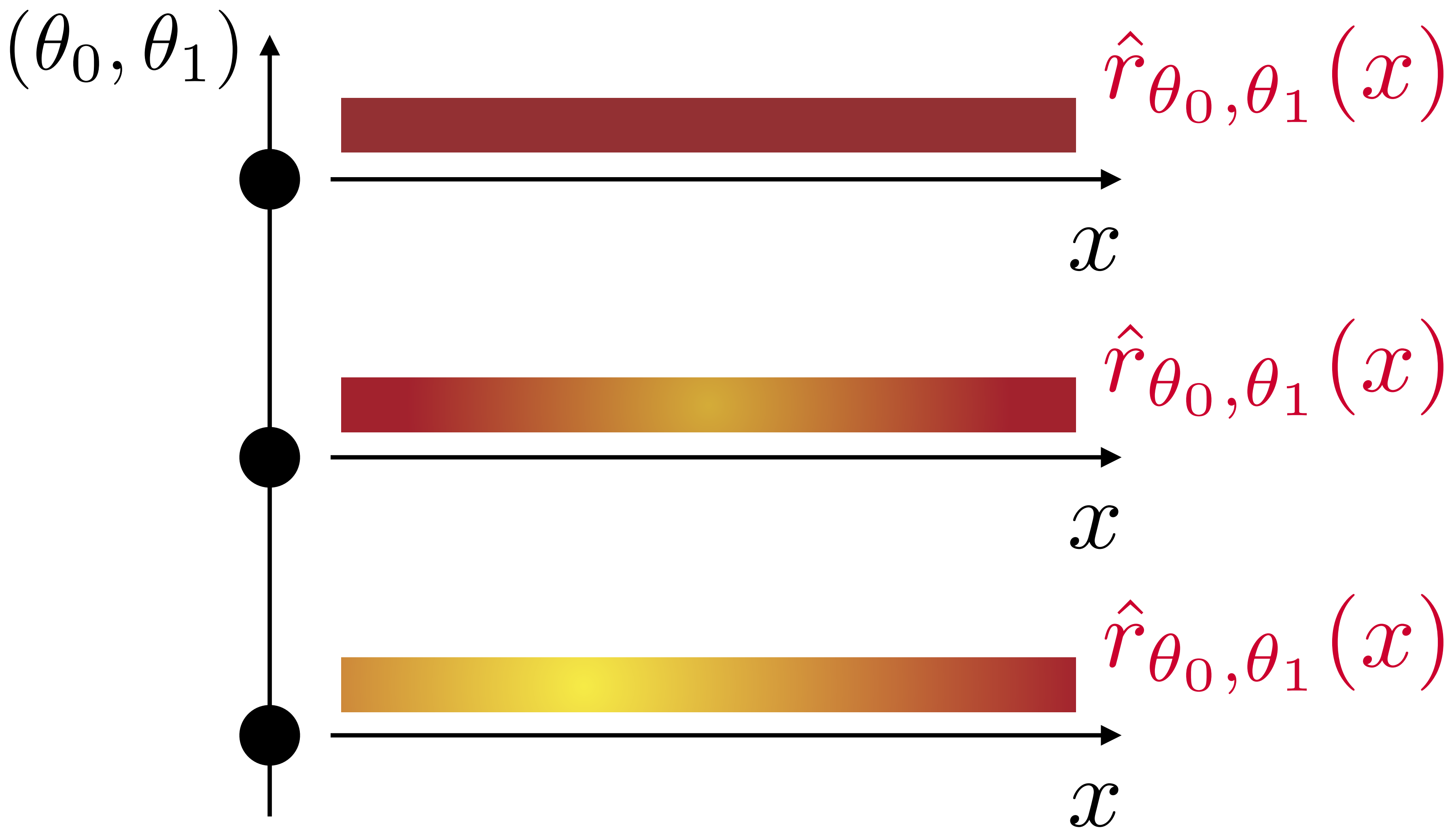
$$
0 = \left.\frac{\delta F}{\delta \hat{g}}\right|_{g^*} = 2\hat{g} \underbrace{\int \mathrm{d}z\; p(x, z|\theta)}_{= p(x|\theta)} - 2 \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z)
$$

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

Two types of likelihood ratio estimators

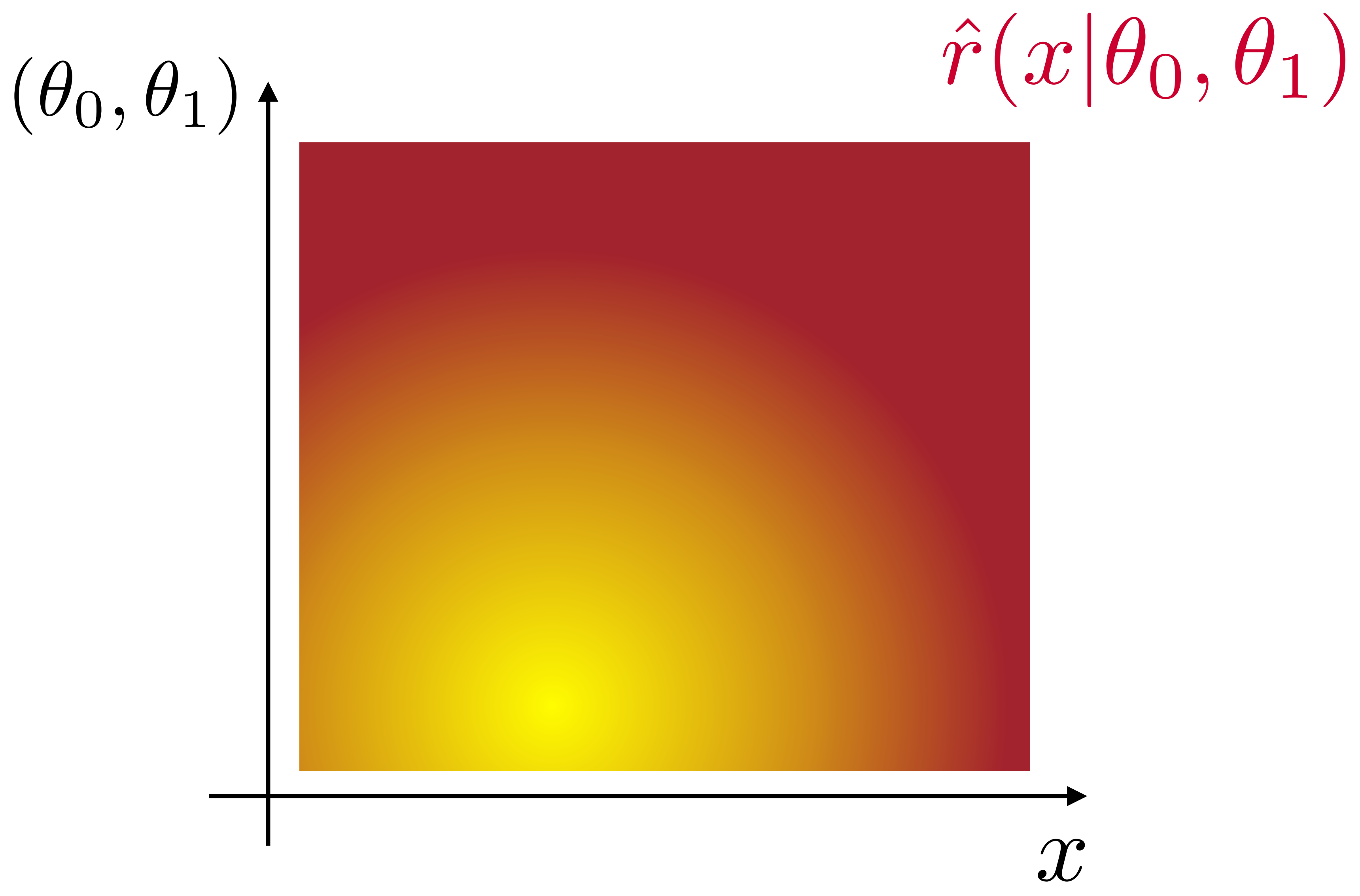
A) Point by point:

- 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

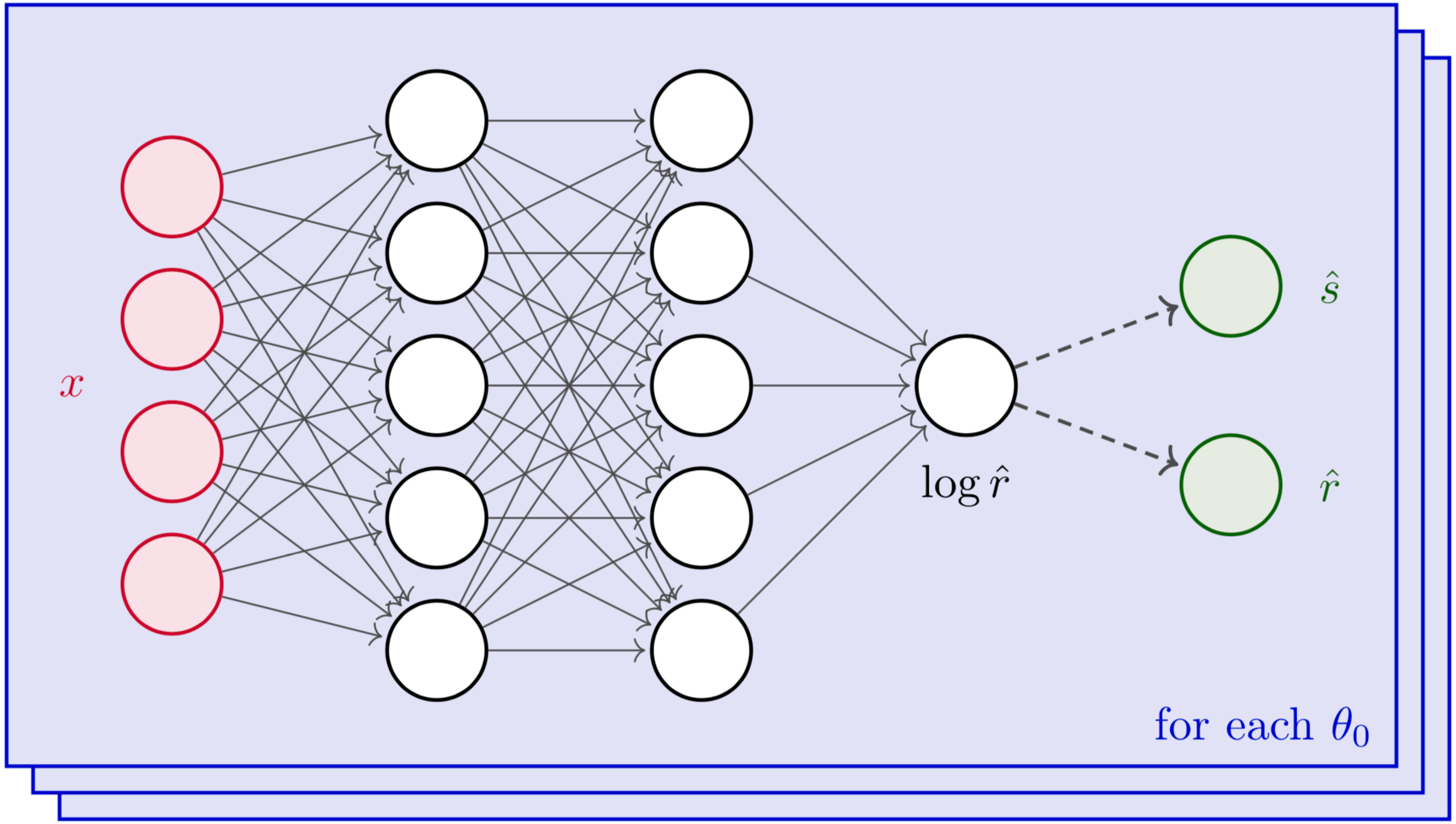


B) Parameterized: [P. Baldi et al. 1506.02169]

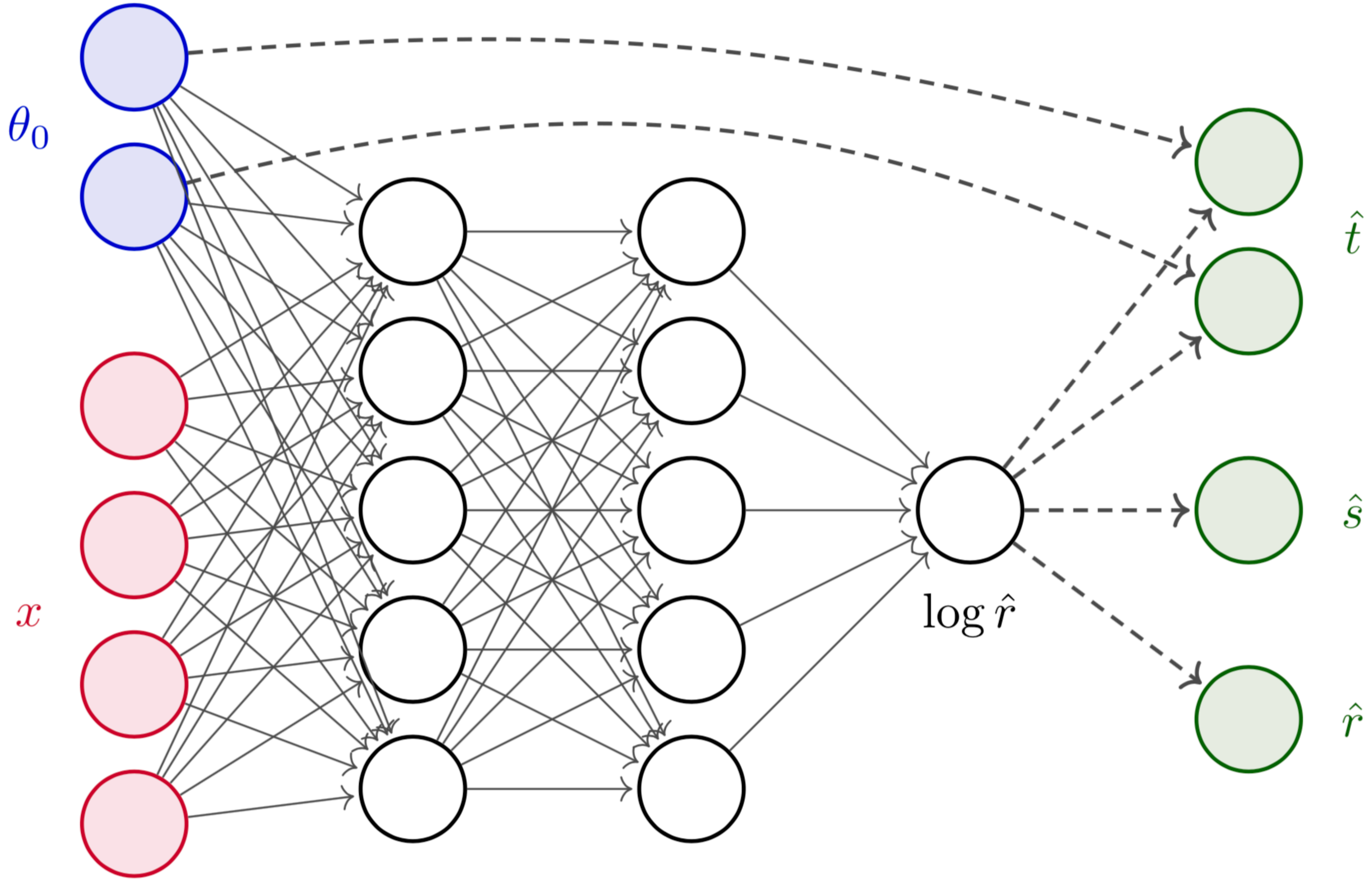
- create one estimator $\hat{r}(x|\theta_0, \theta_1)$ that is a function of θ_0 and θ_1
- no further interpolation necessary
- "borrows information" from close points



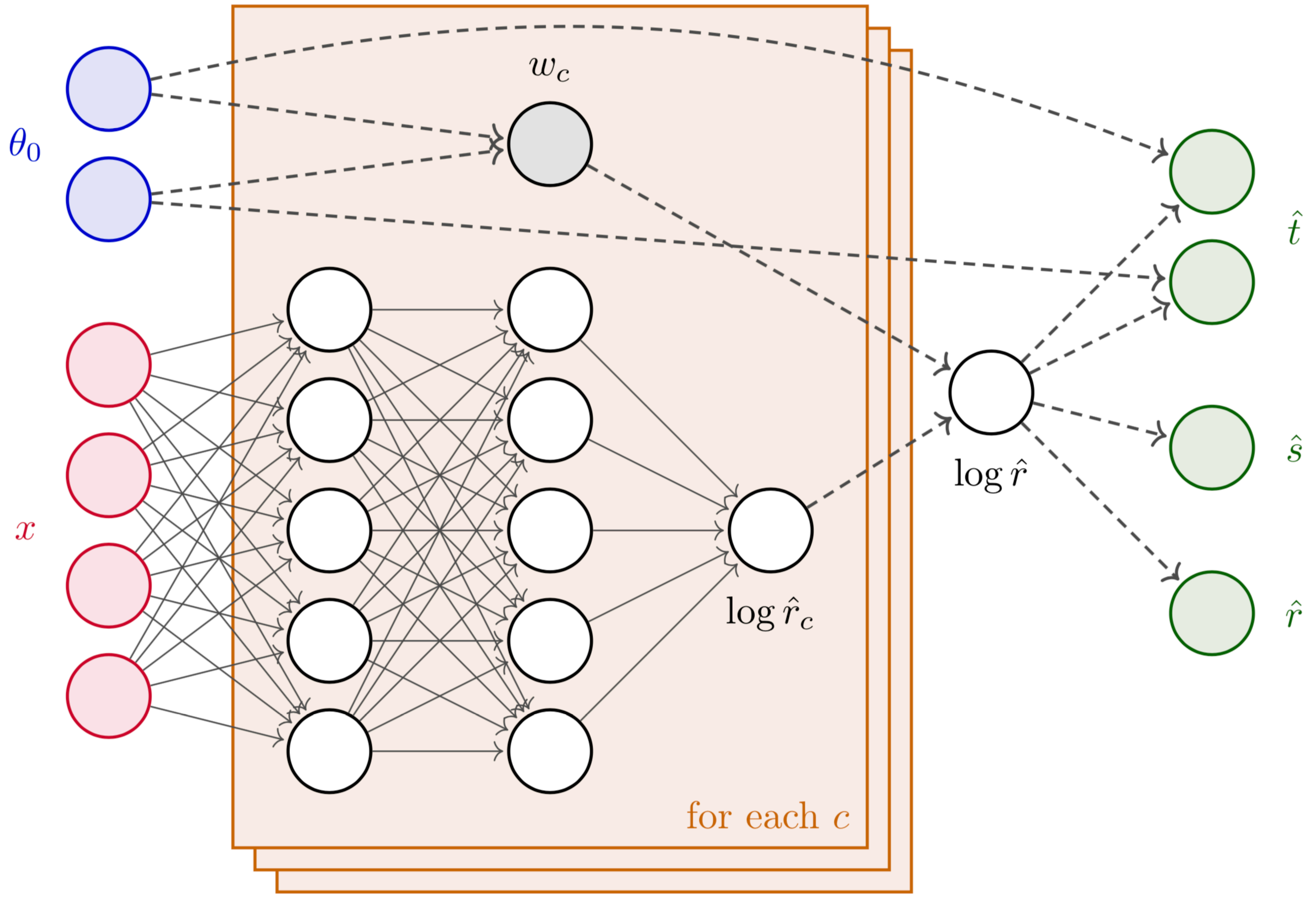
Point by point



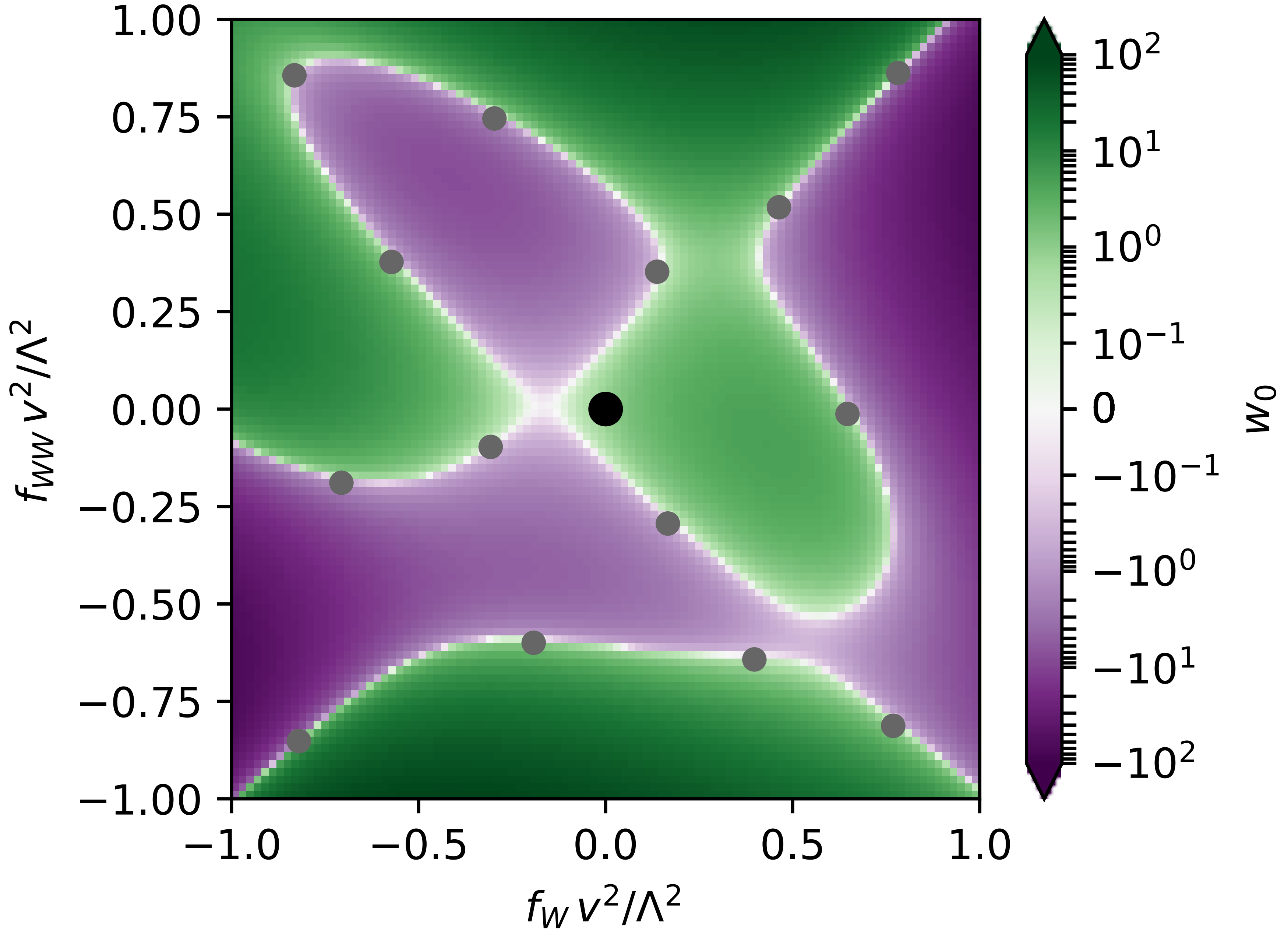
(Agnostic) parameterized estimators



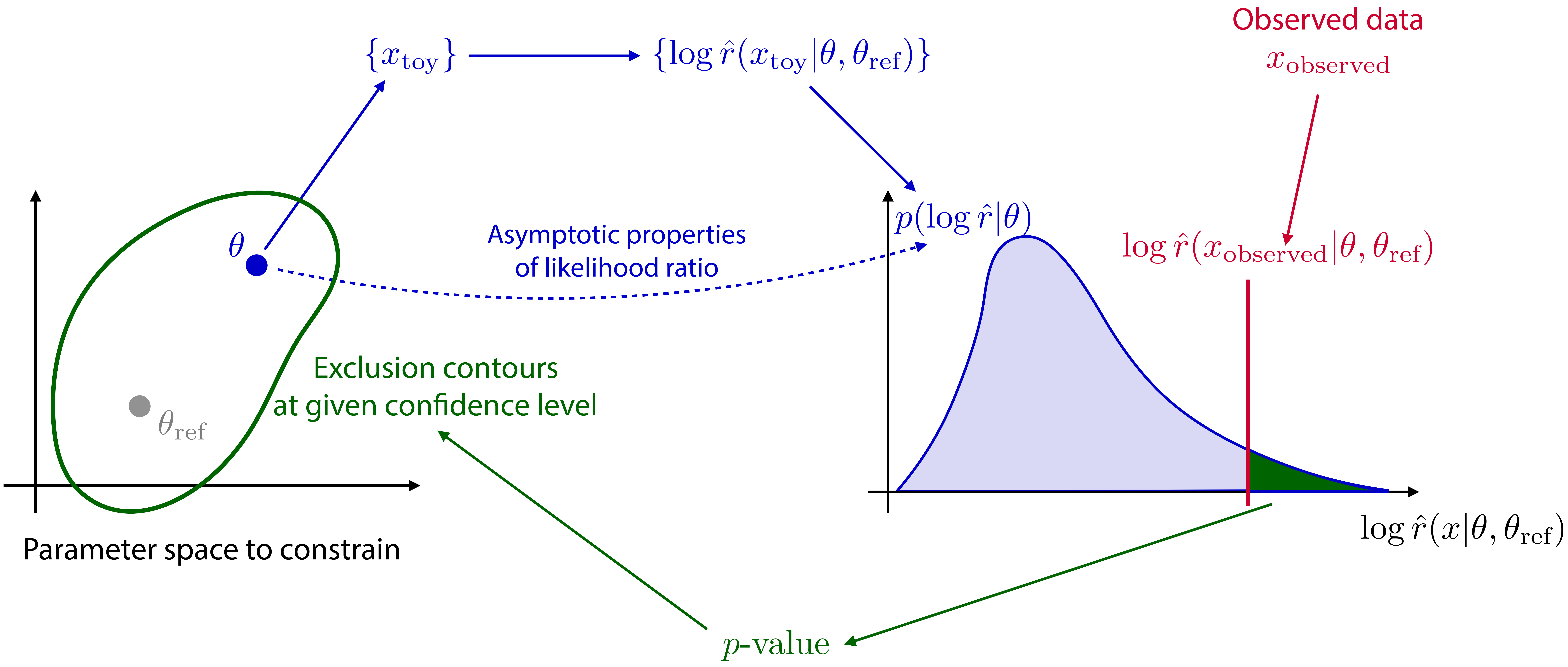
Morphing-aware parameterized estimators



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



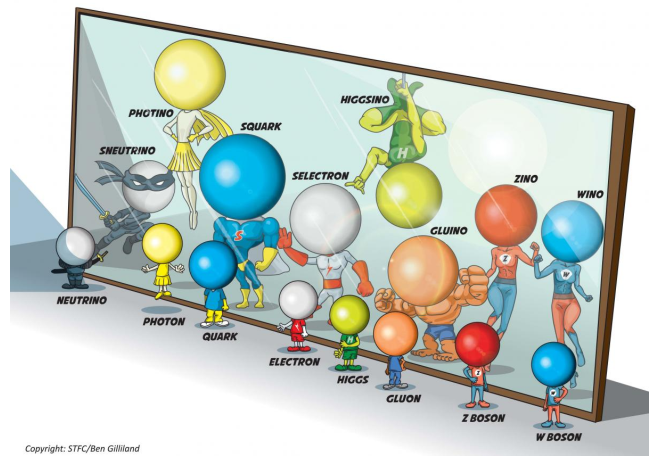
Limit setting (frequentist)



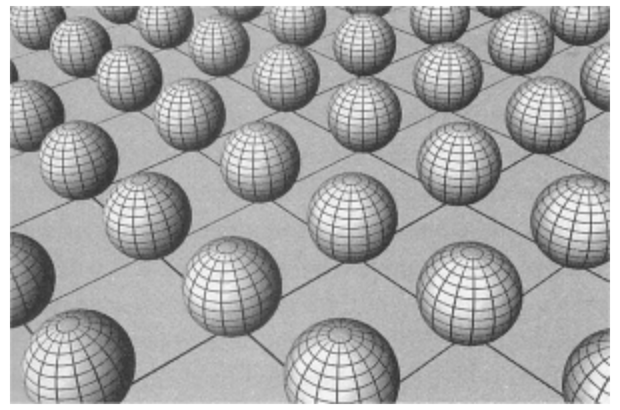
Effective field theory

Energy

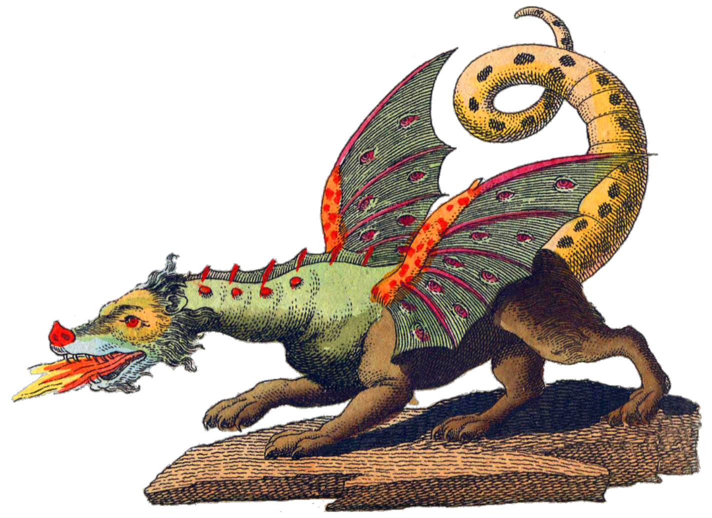
Λ



Supersymmetry?



Extra dimensions?



Dragons?

Composite Higgs?

…

Matching

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

Effective field theory

Measurements

$E \ll \Lambda$

LHC Higgs

Other LHC measurements

LEP

Flavour

…

Detector effects