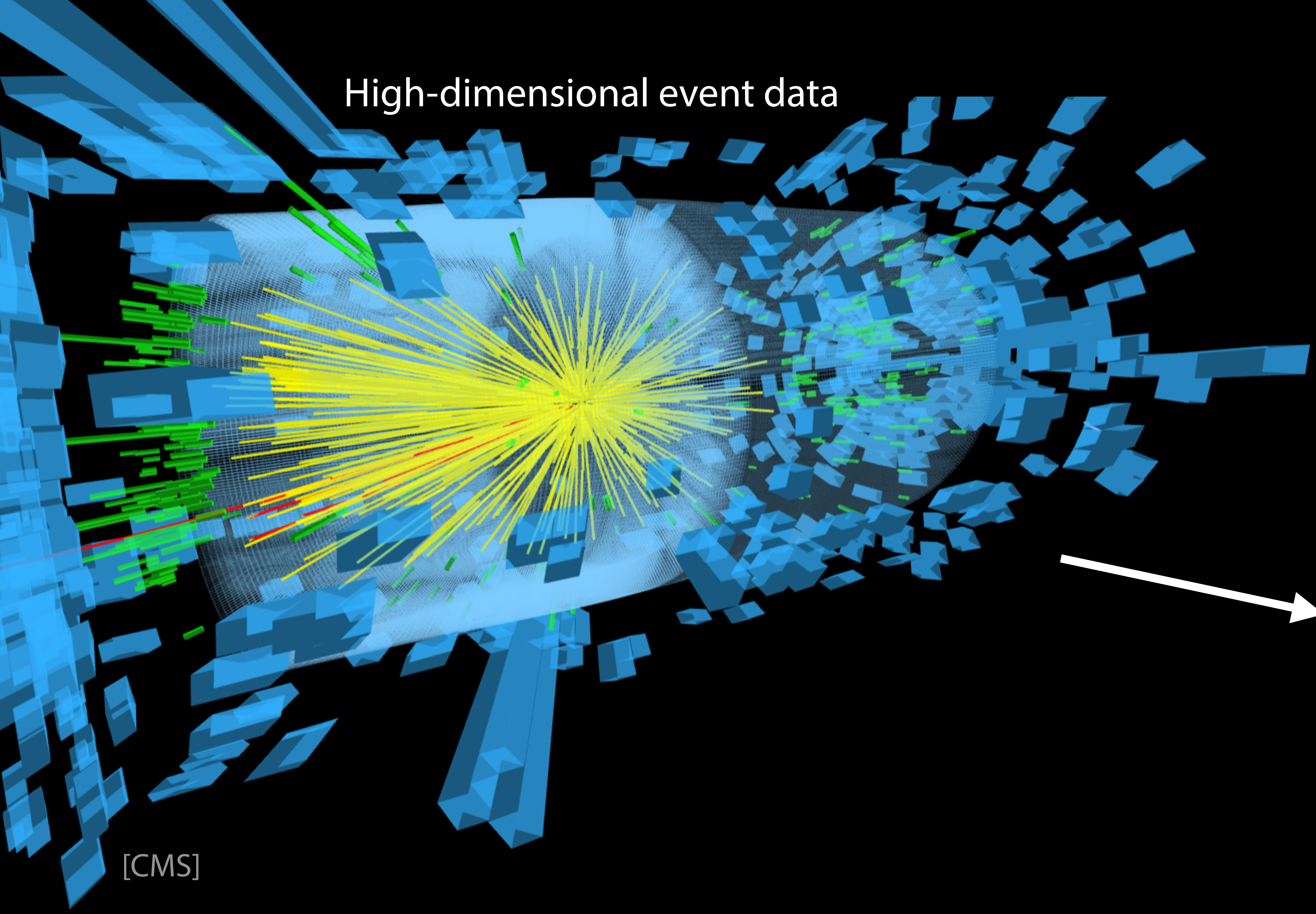
Simulation-based inference in particle physics and beyond

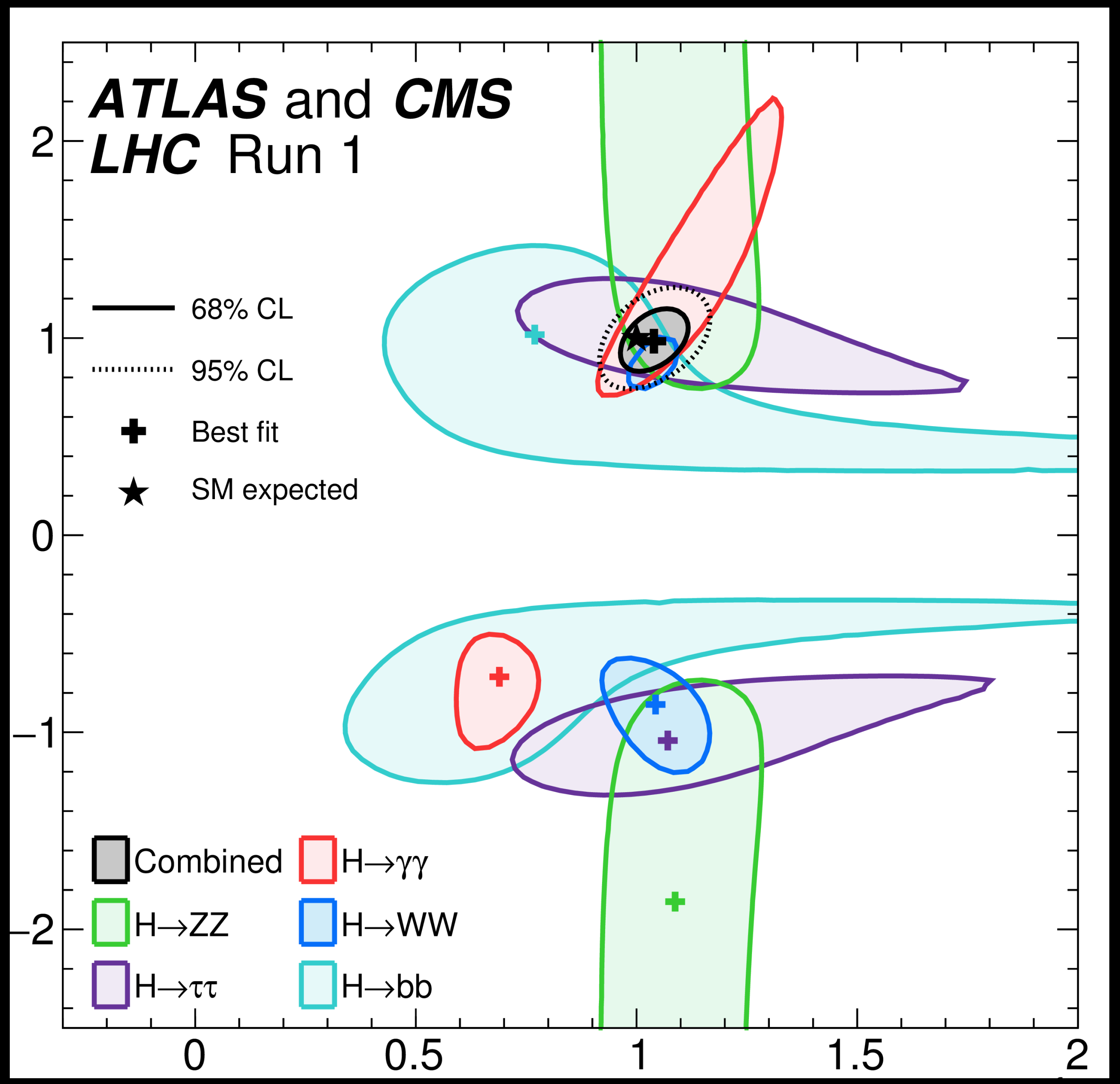
Johann Brehmer
Work done at NYU

CMS ML journal club
November 30, 2021

High-dimensional event data



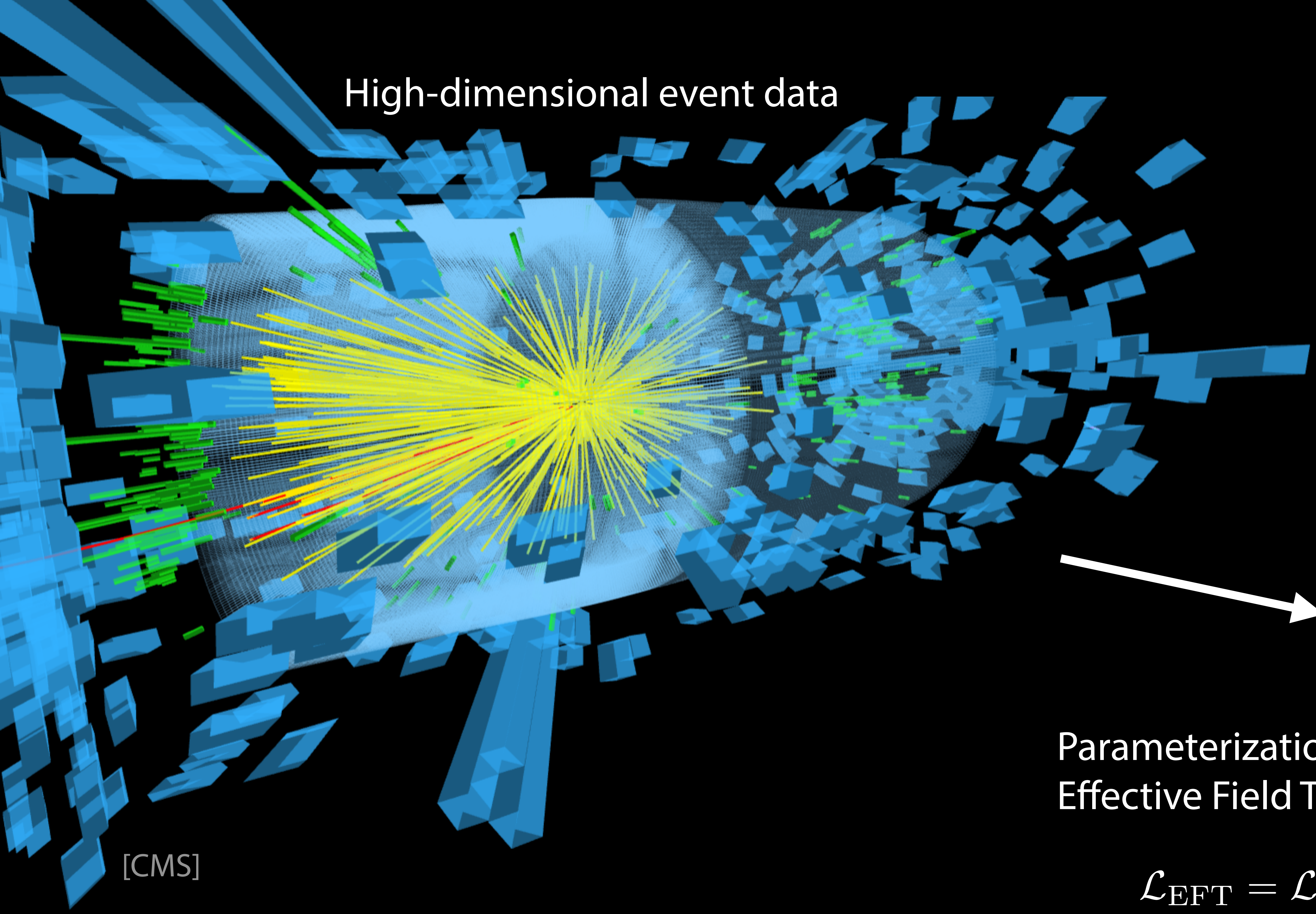
[CMS]



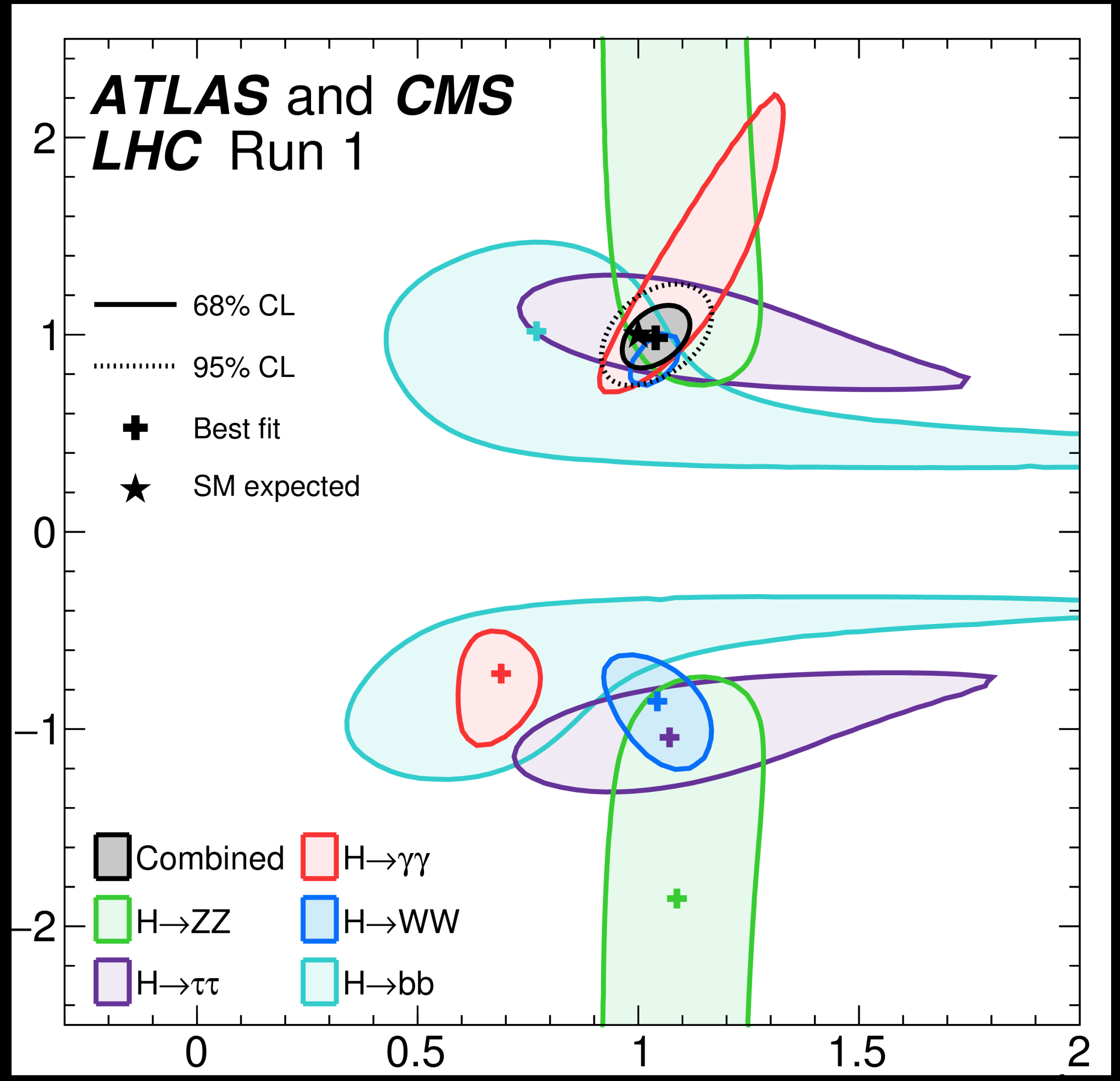
[ATLAS, CMS 1606.02266]

Precision constraints on new physics

High-dimensional event data



[CMS]



[ATLAS, CMS 1606.02266]

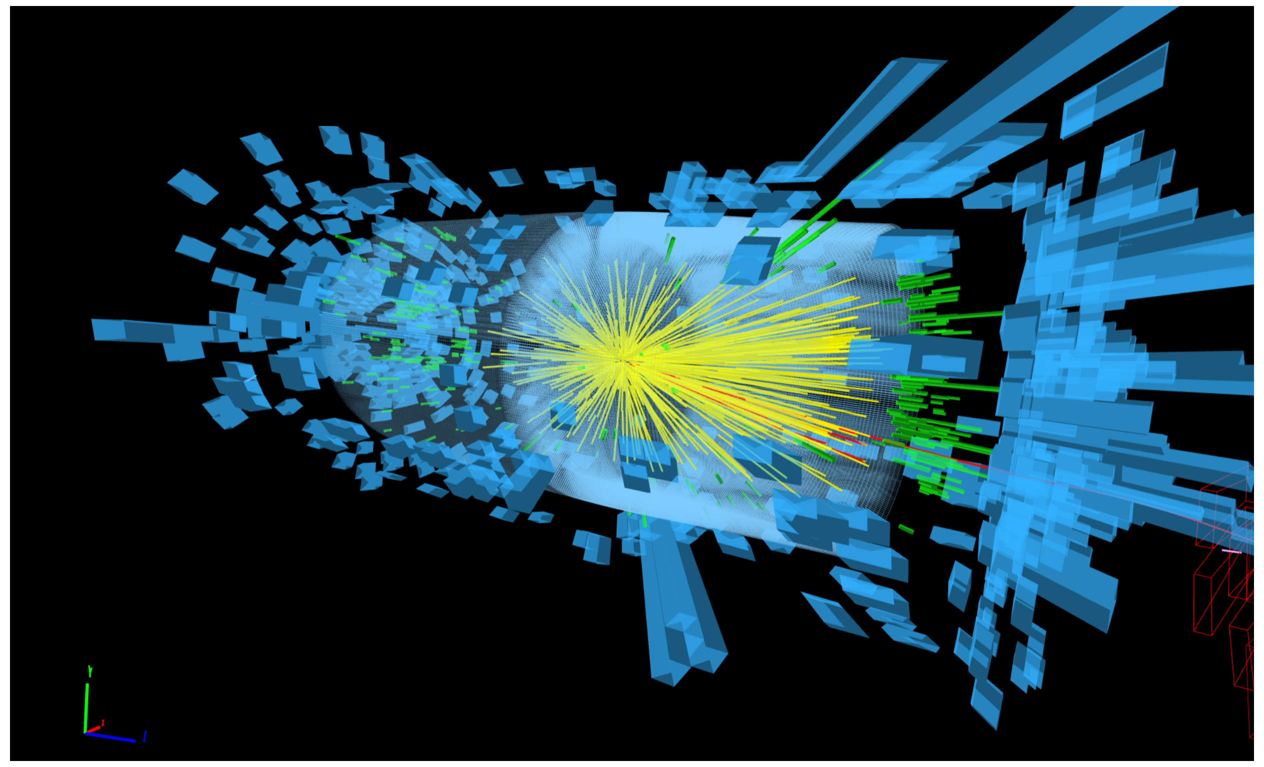
Precision constraints on new physics

Parameterization e.g. in Effective Field Theory:

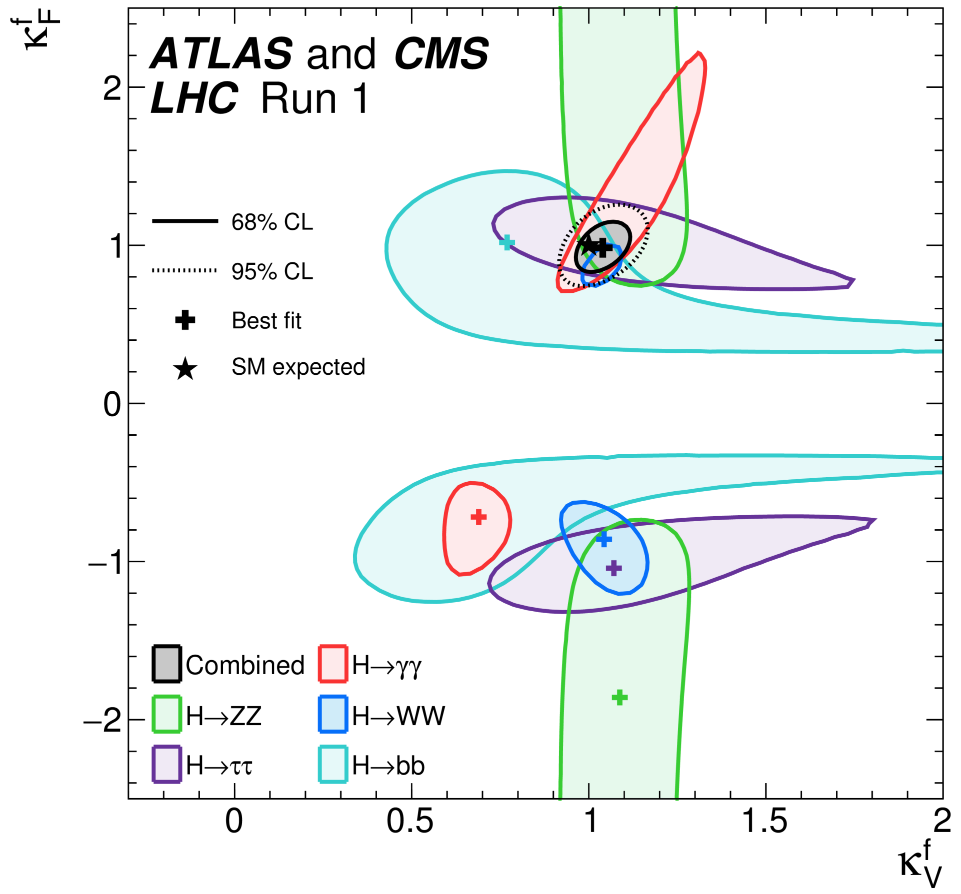
systematic expansion of new physics around Standard Model

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

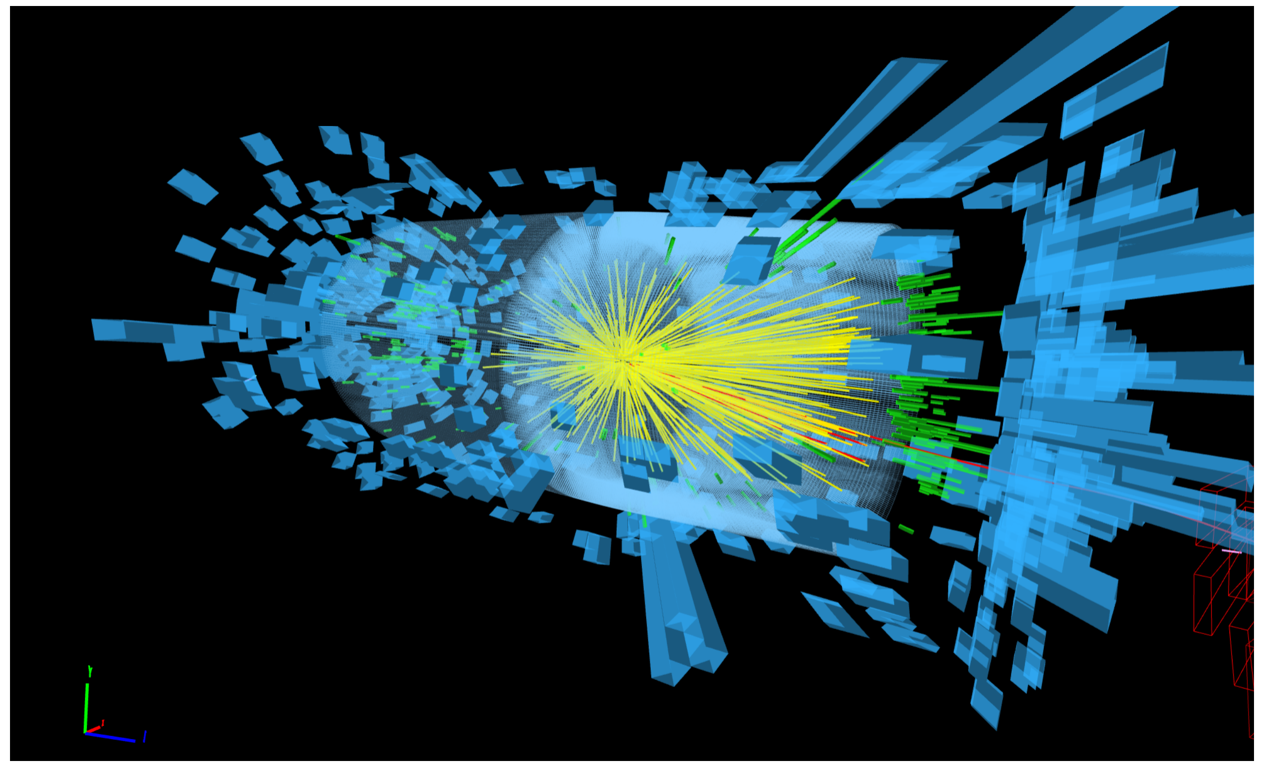
10s to 100s "universal" parameters to measure



High-dimensional
event data x



Constraints on
parameters θ

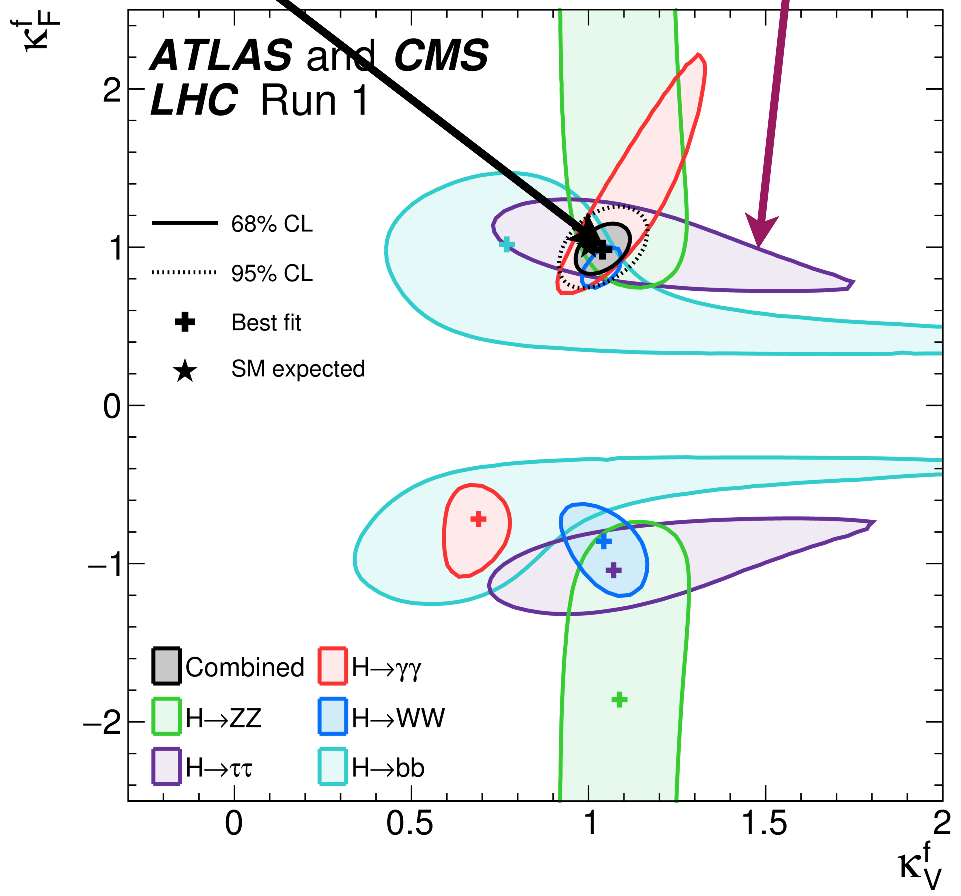


High-dimensional
event data x

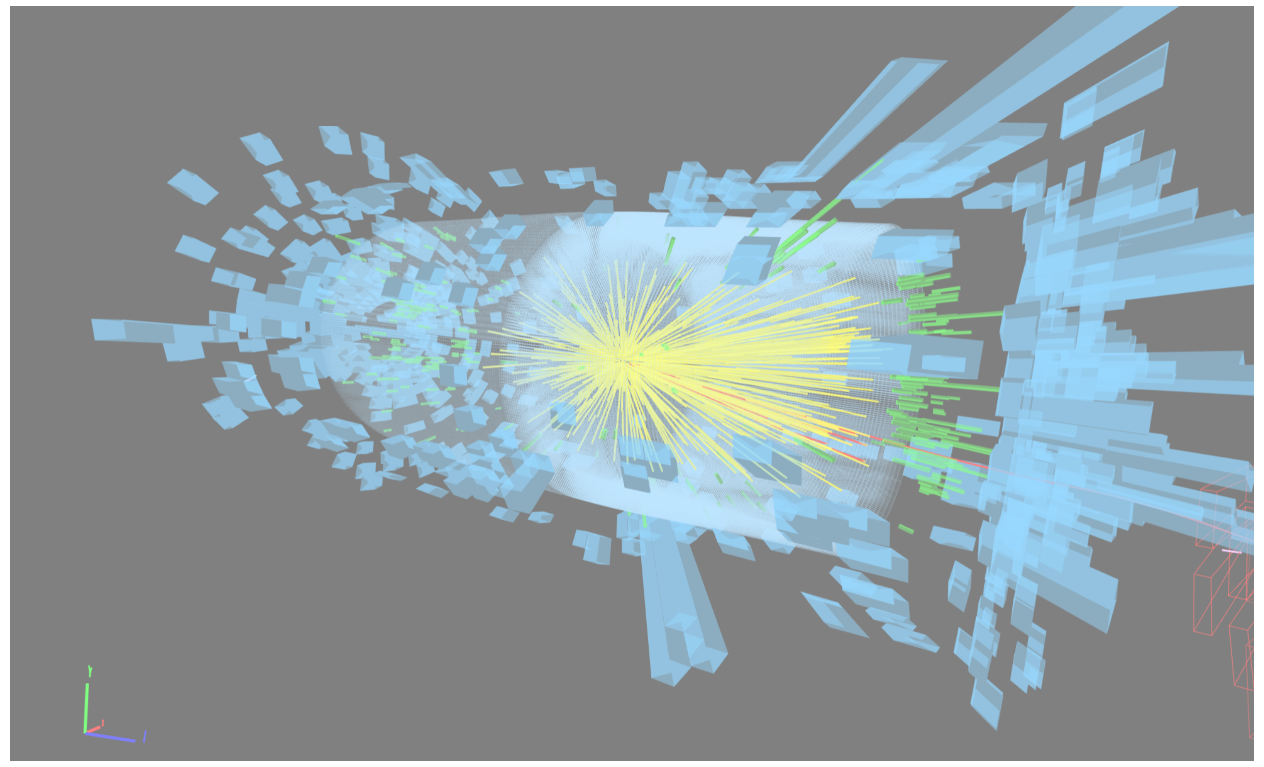
Likelihood function
$p(x|\theta)$

Maximum-likelihood
estimator

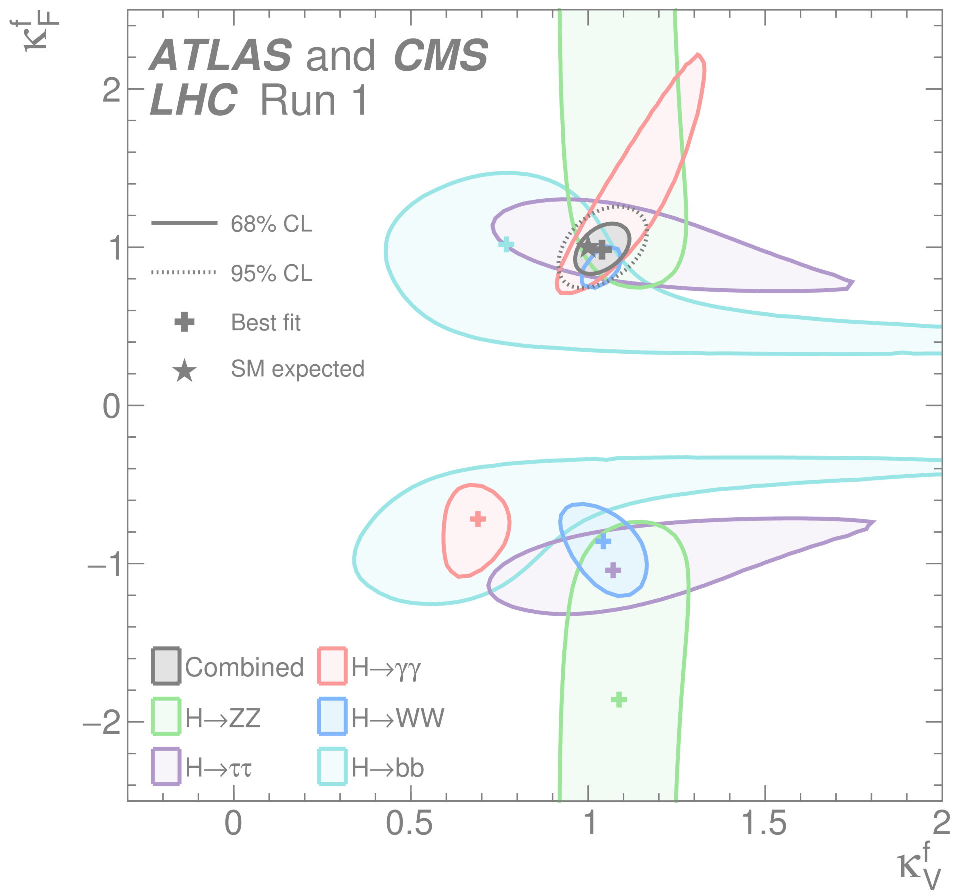
Confidence limits based
on likelihood ratio tests



Constraints on
parameters θ



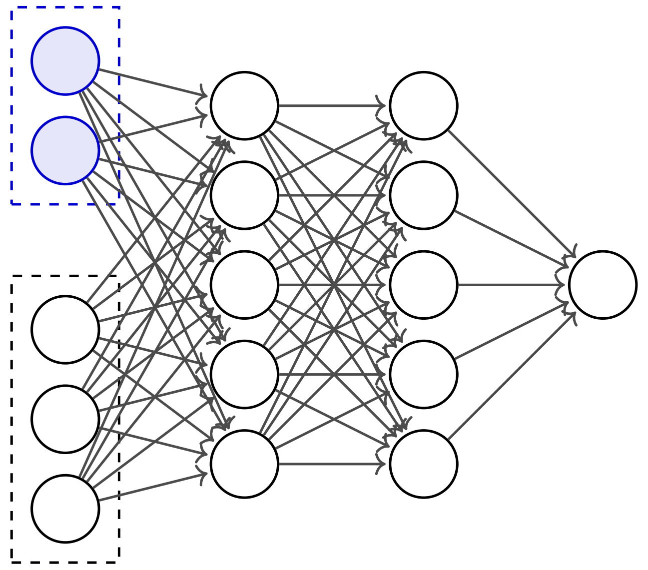
Likelihood function
$p(x|\theta)$



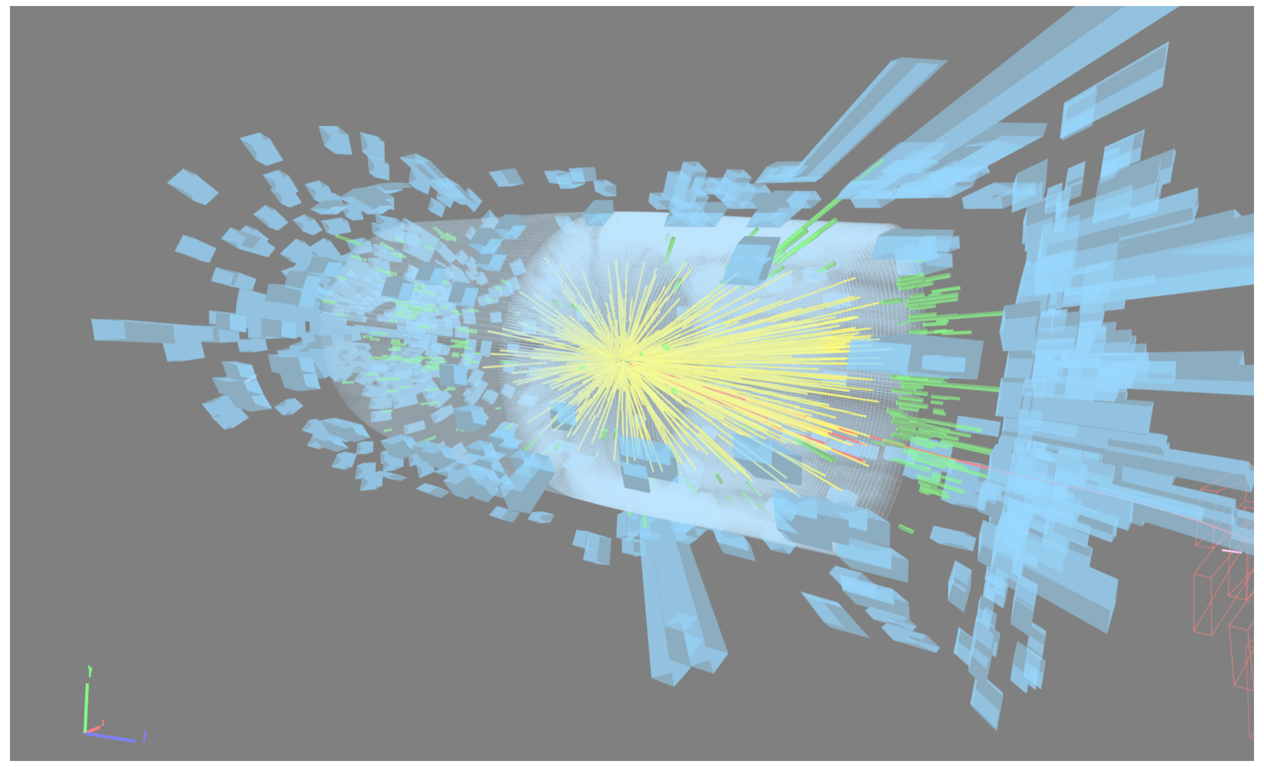
High-dimensional event data x

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

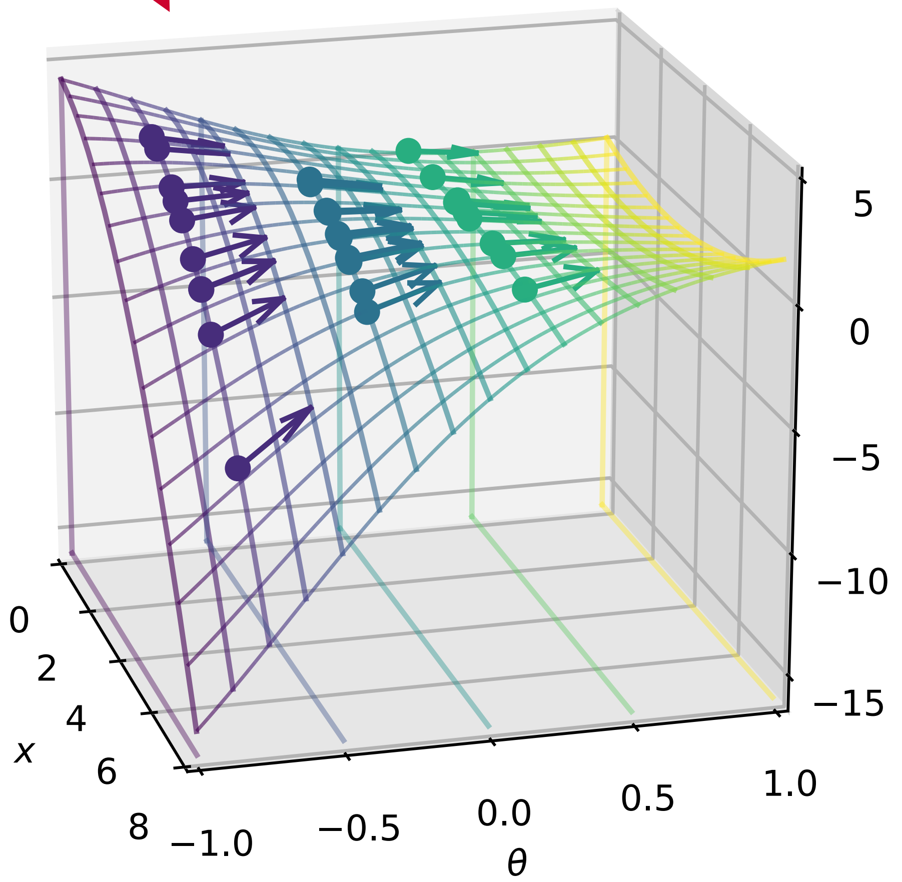
Constraints on parameters θ



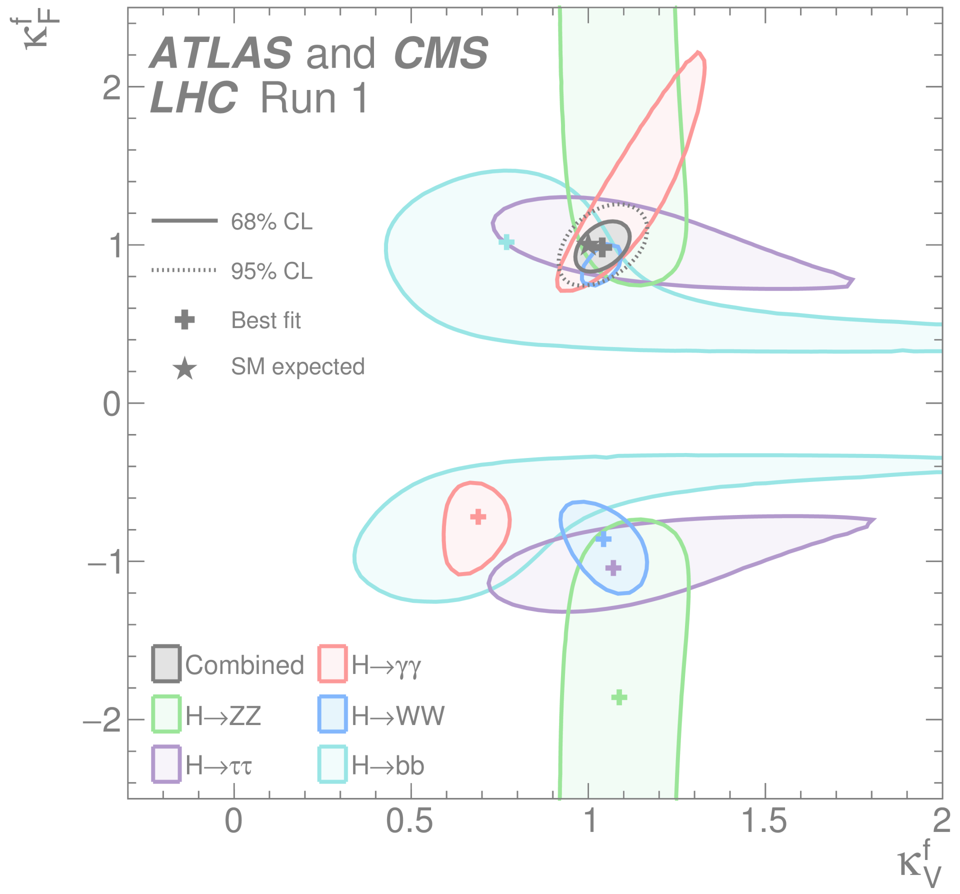
Machine learning



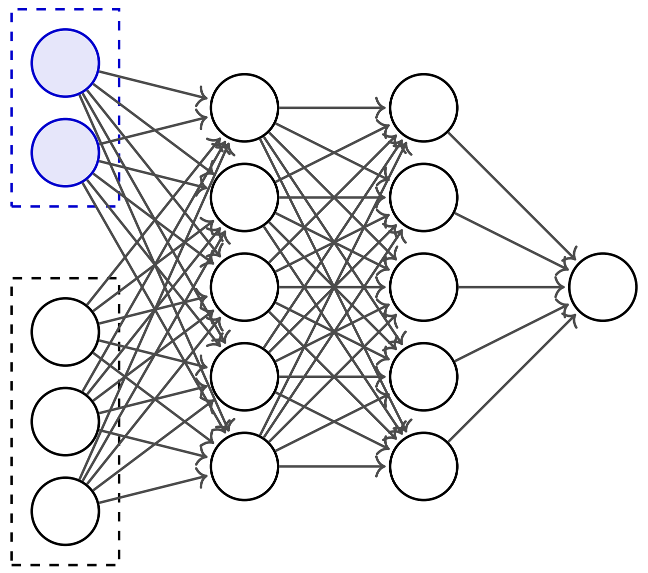
High-dimensional event data x



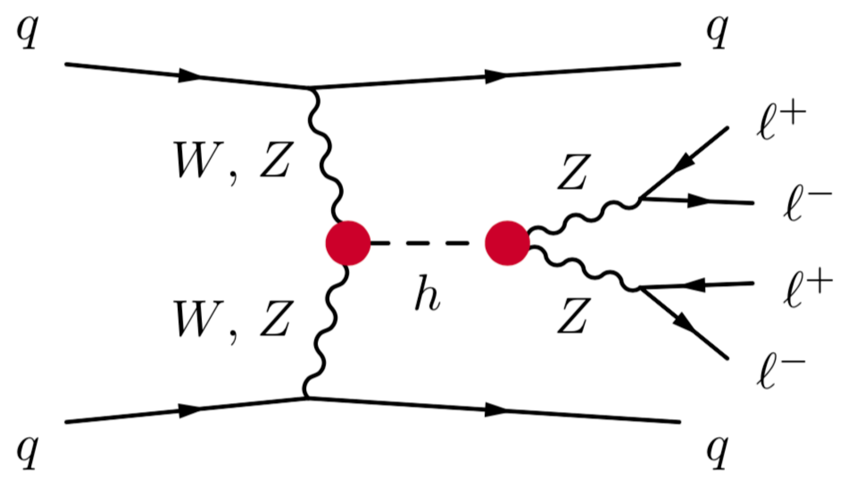
Estimator of the likelihood $p(x|\theta)$



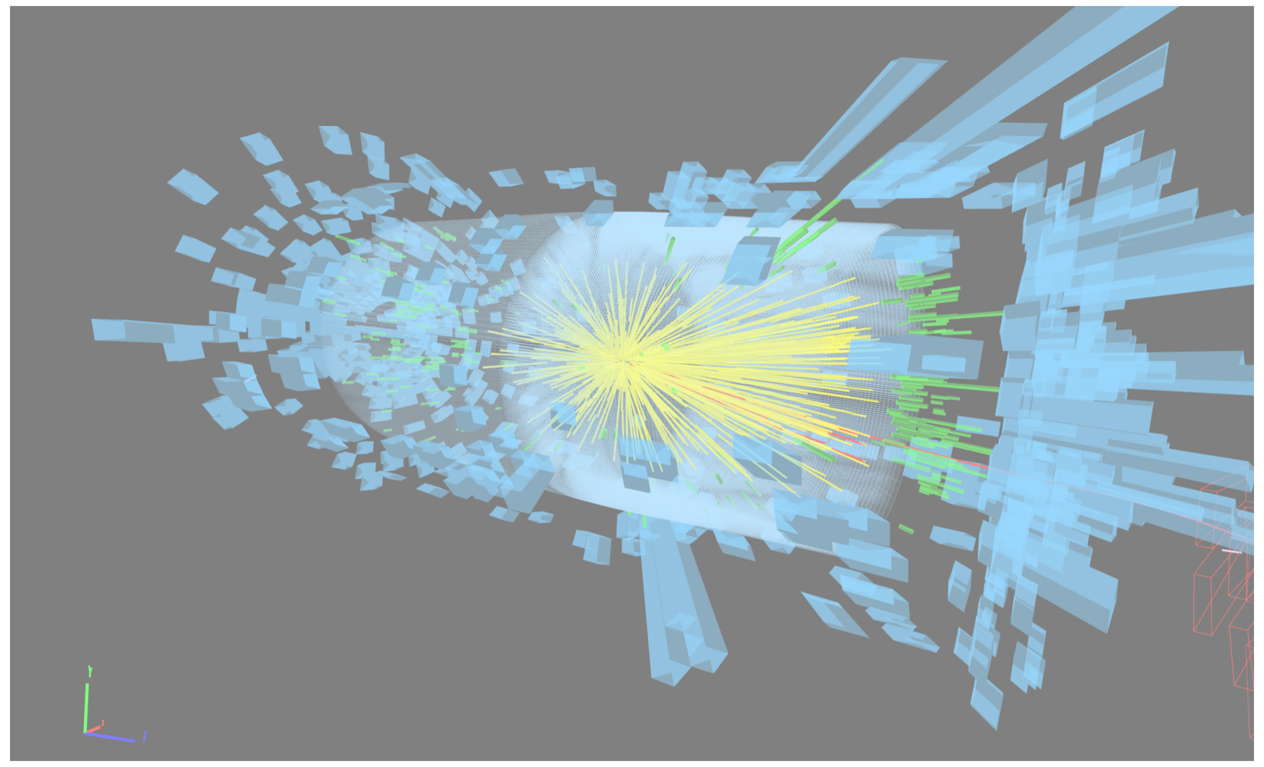
Constraints on parameters θ



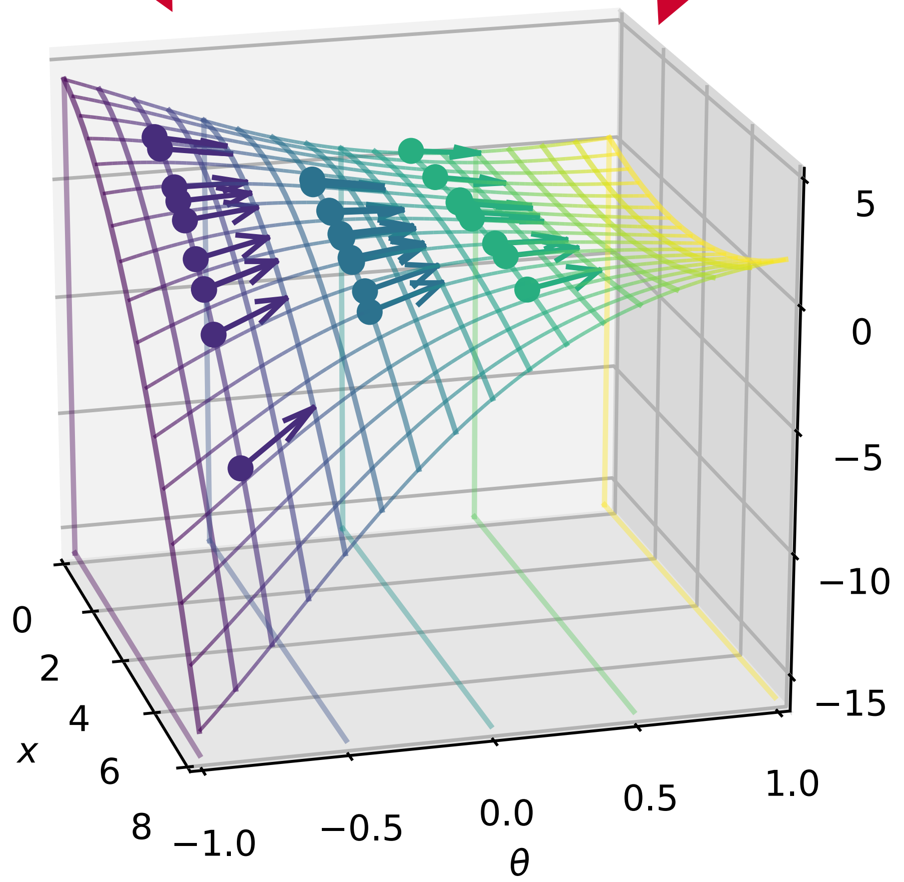
Machine learning



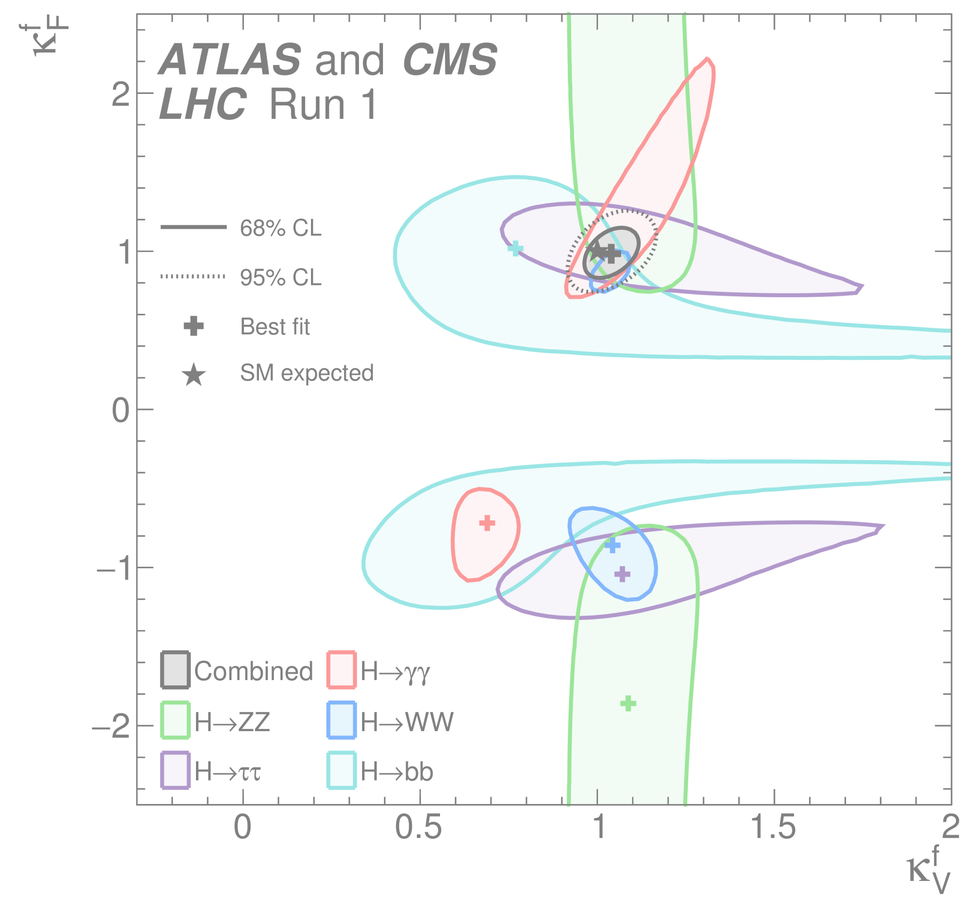
Physics insight:
matrix element information



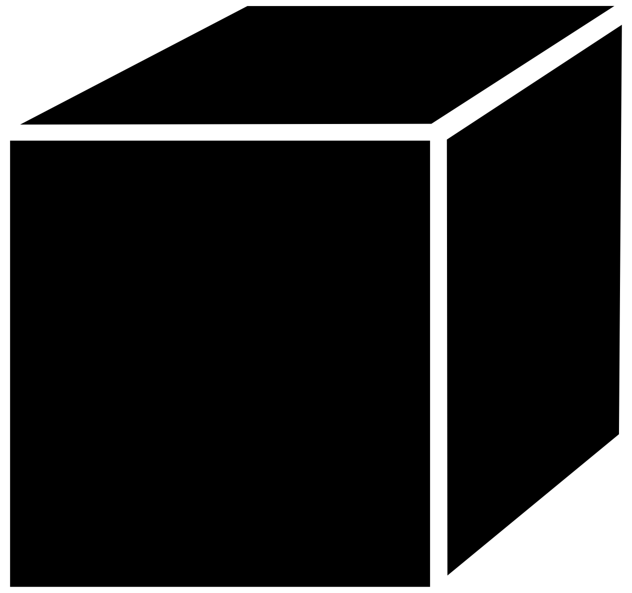
High-dimensional
event data x



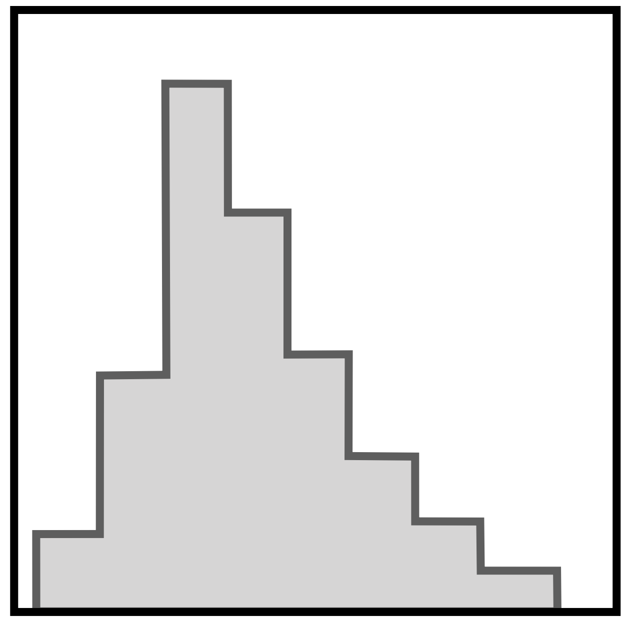
Estimator of the
likelihood $p(x|\theta)$



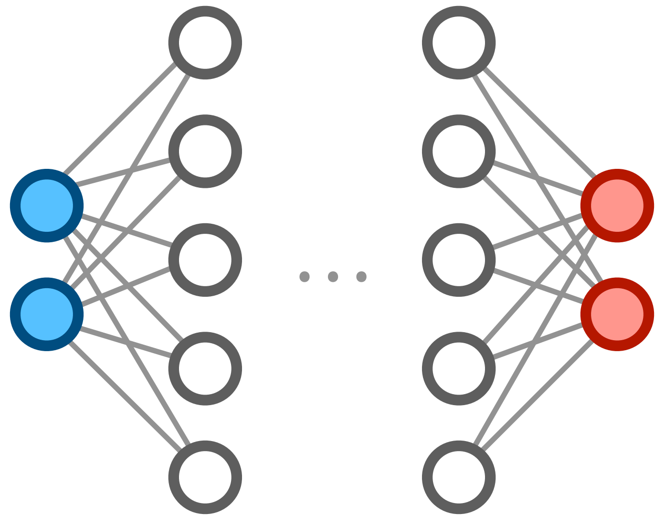
Constraints on
parameters θ



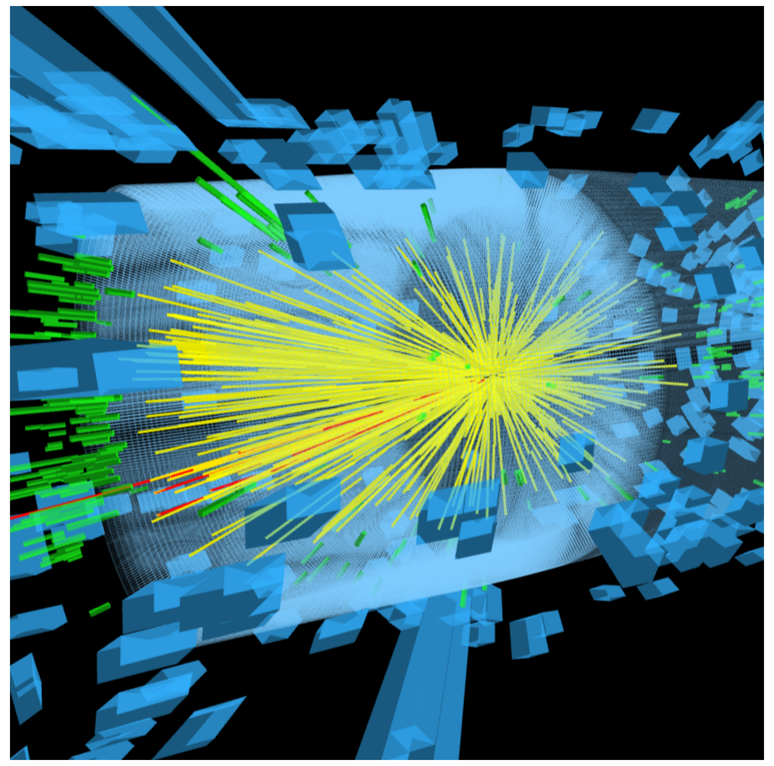
1. The simulation-based inference problem



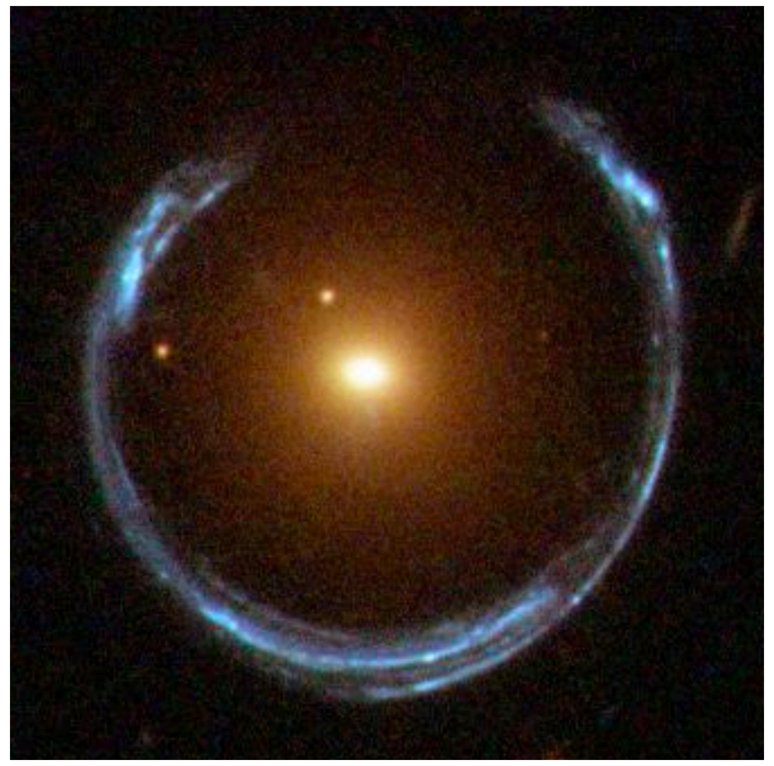
2. Why has that not stopped us before?



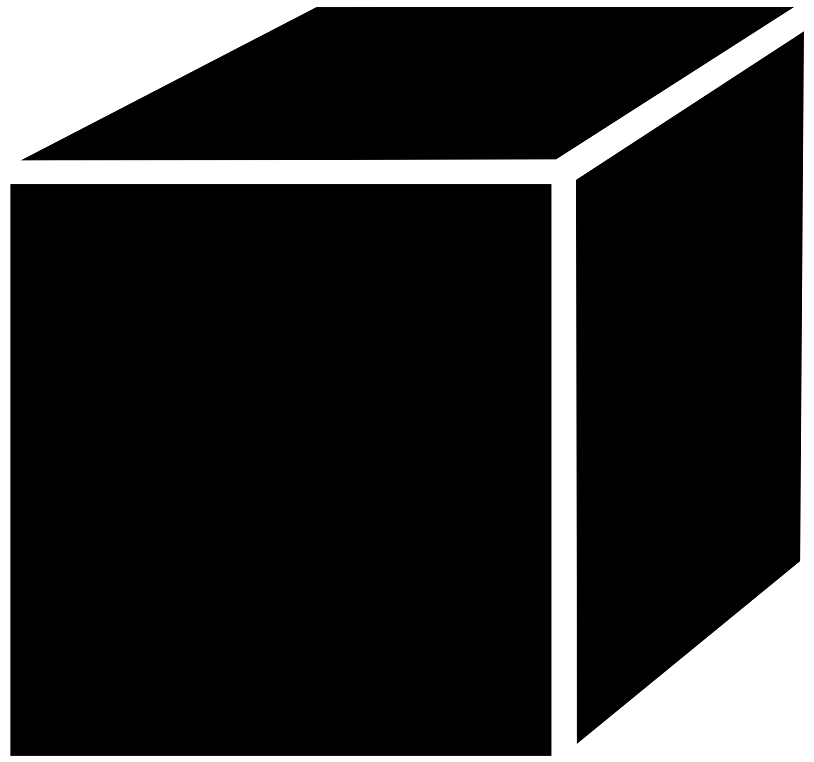
3. Machine learning methods



4. Examples



5. Beyond the LHC



1. LHC measurements as a simulation-based inference problem

Modelling LHC processes

Theory parameters

θ

Evolution

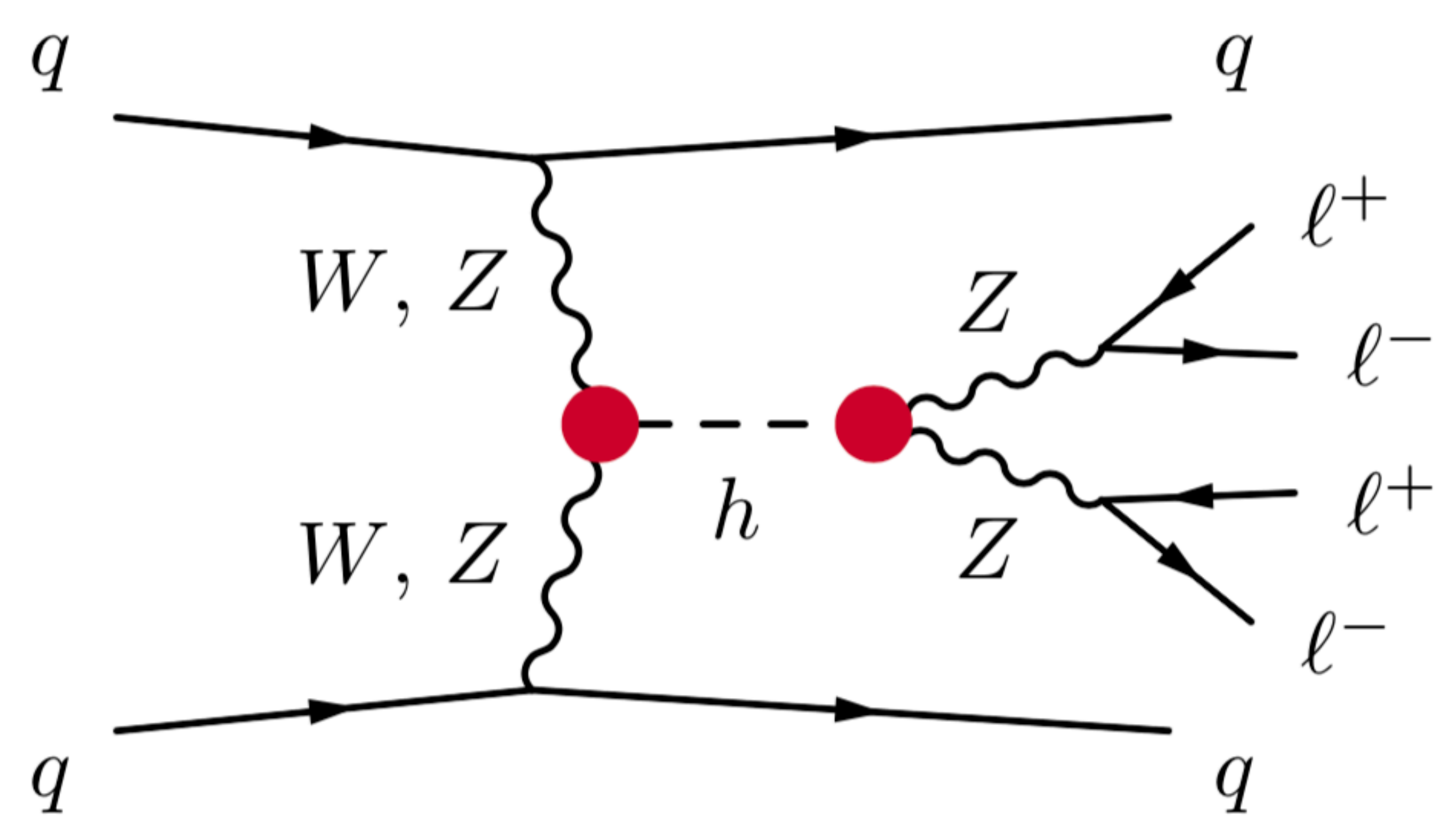
Modelling LHC processes

Latent variables

Parton-level momenta

Theory parameters

$z_p \longleftarrow \theta$



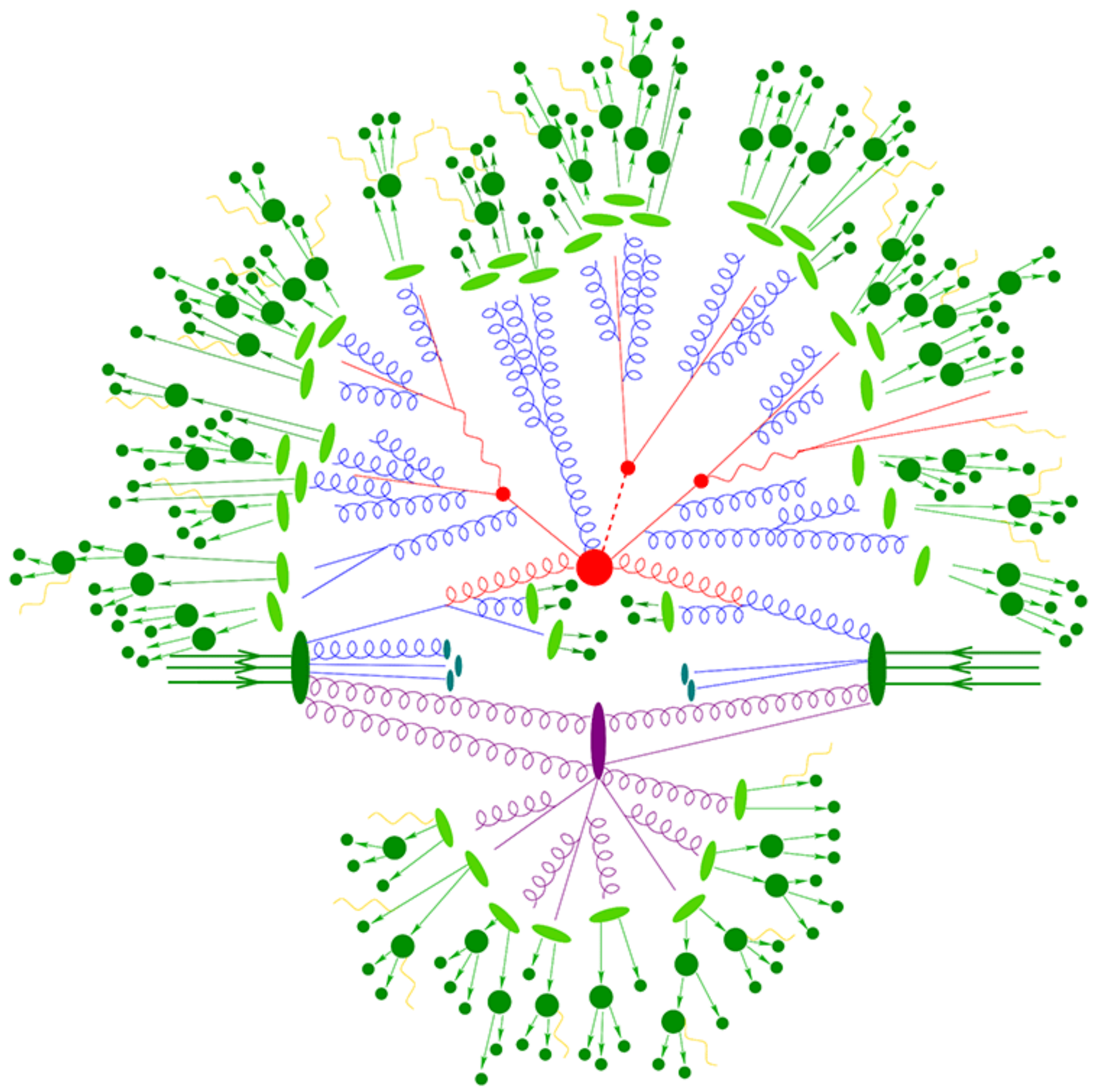
Evolution

Modelling LHC processes

Latent variables

Shower splittings	Parton-level momenta	Theory parameters
z_s	z_p	θ

$z_s \longleftarrow z_p \longleftarrow \theta$



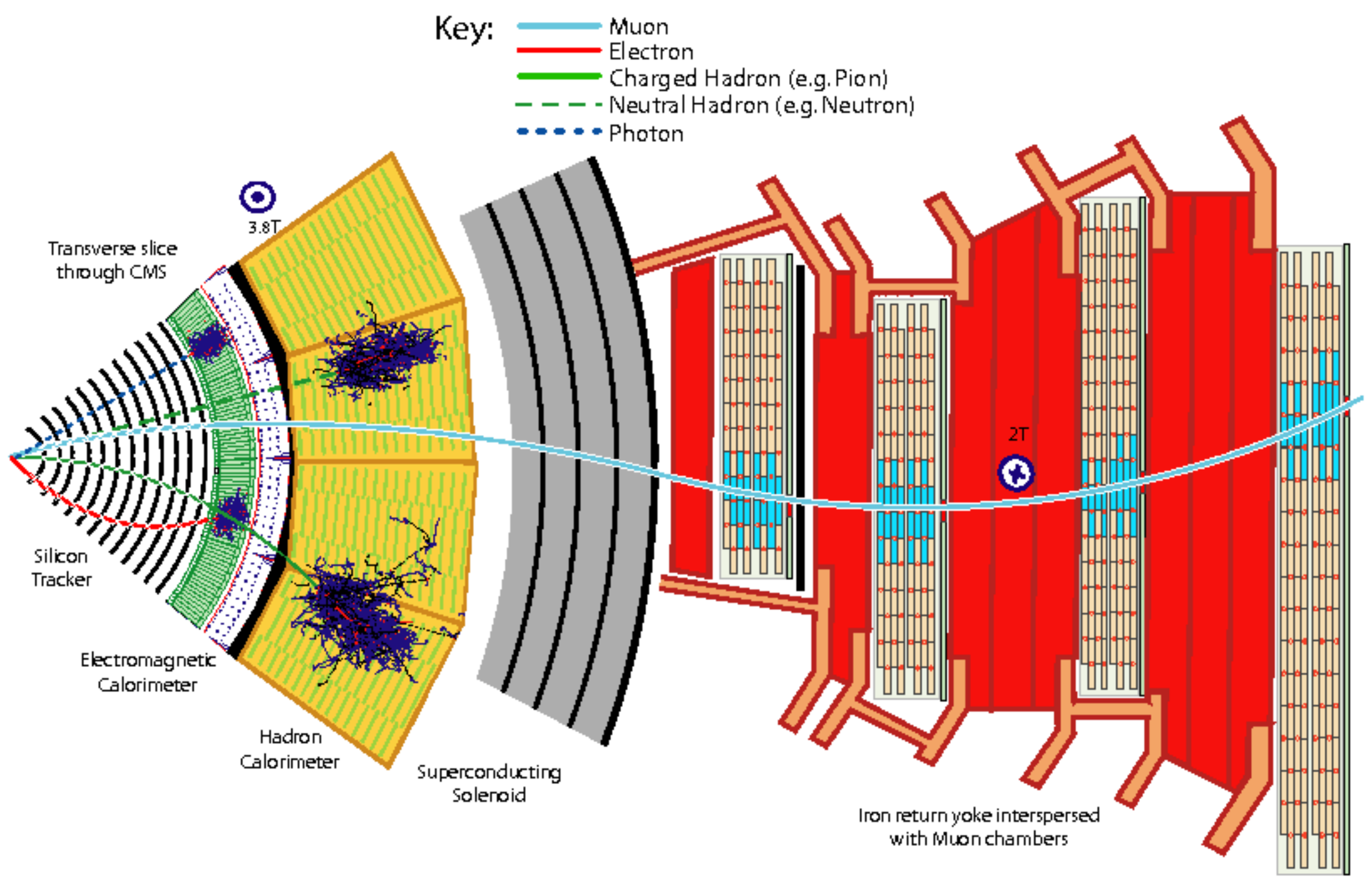
[F. Krauss]

Evolution

Modelling LHC processes

Latent variables

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



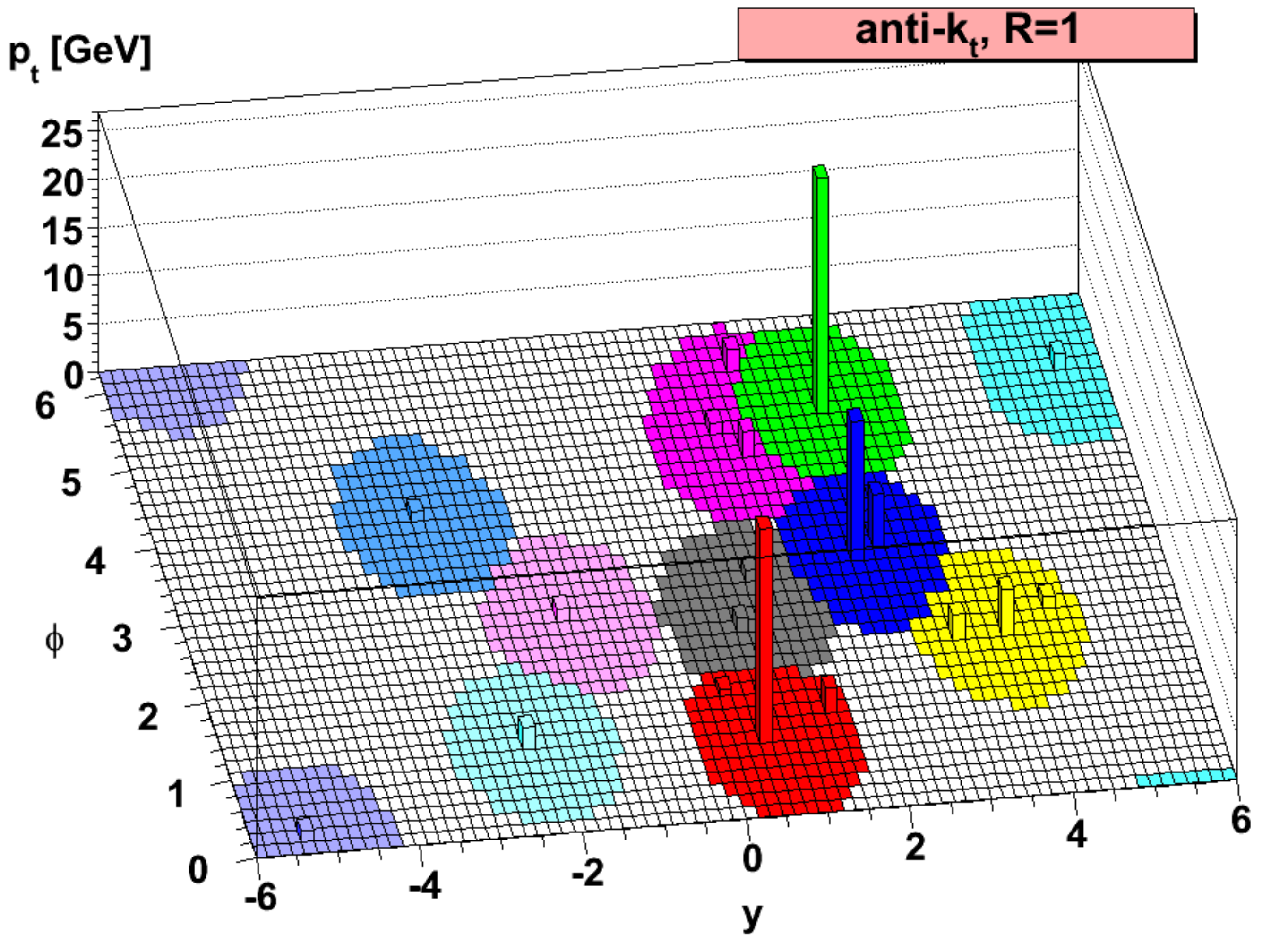
[CMS]

$\longleftarrow$ Evolution

Modelling LHC processes

Latent variables

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



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

$\longleftarrow$

Evolution

Modelling LHC processes

Latent variables

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

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

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

Prediction (simulation)

Modelling LHC processes

Latent variables

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

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

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

$\longrightarrow$

Inference

Modelling LHC processes

Latent variables

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

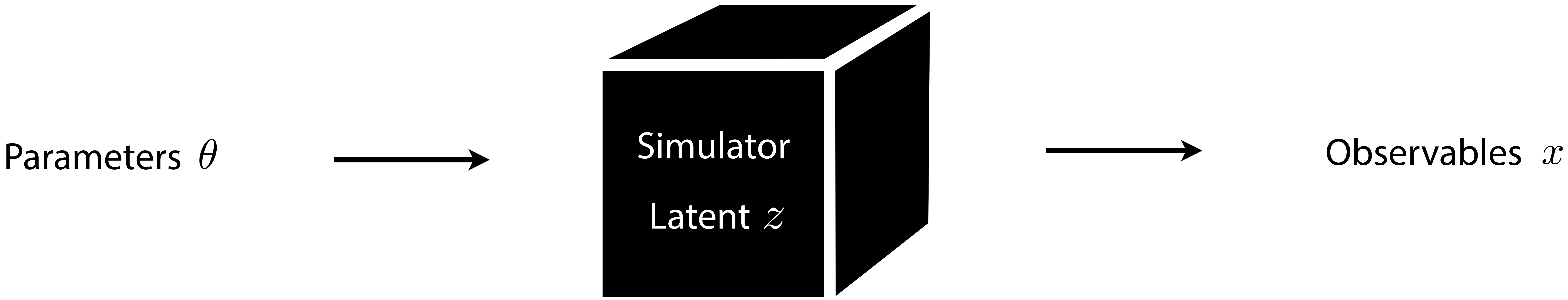
$$p(x|\theta) = \boxed{\int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p} \; p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad p(z_p|\theta)$$

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

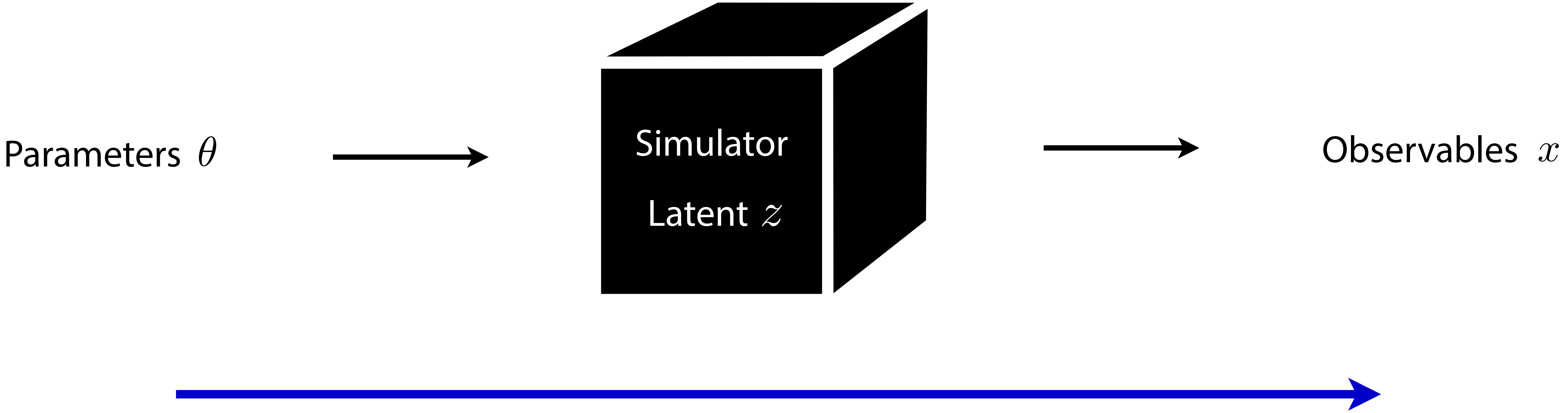
(More subtle: We cannot sample from $p(z|x,\theta)$ efficiently.)

$\longrightarrow$ Inference

The problem of simulation-based ("likelihood-free") inference



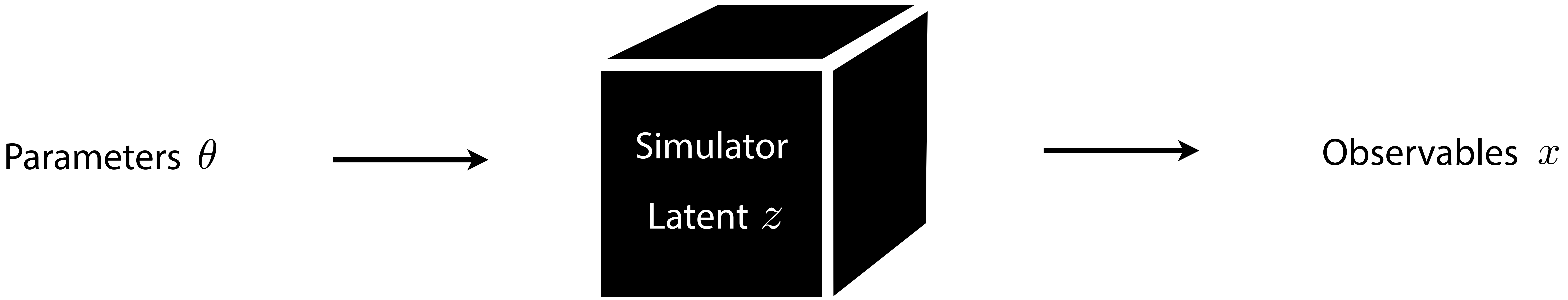
The problem of simulation-based ("likelihood-free") inference



Prediction:

- Well-motivated mechanistic, causal model
- Simulator can generate samples $x \sim p(x|\theta)$

The problem of simulation-based ("likelihood-free") inference



Prediction:
- Well-motivated mechanistic, causal model
- Simulator can generate samples $x \sim p(x|\theta)$

Inference:
- Interactions between low-level components lead to challenging inverse problems
- Likelihood $p(x|\theta) = \int \mathrm{d}z \, p(x, z|\theta)$ is intractable

Three problem statements

Given

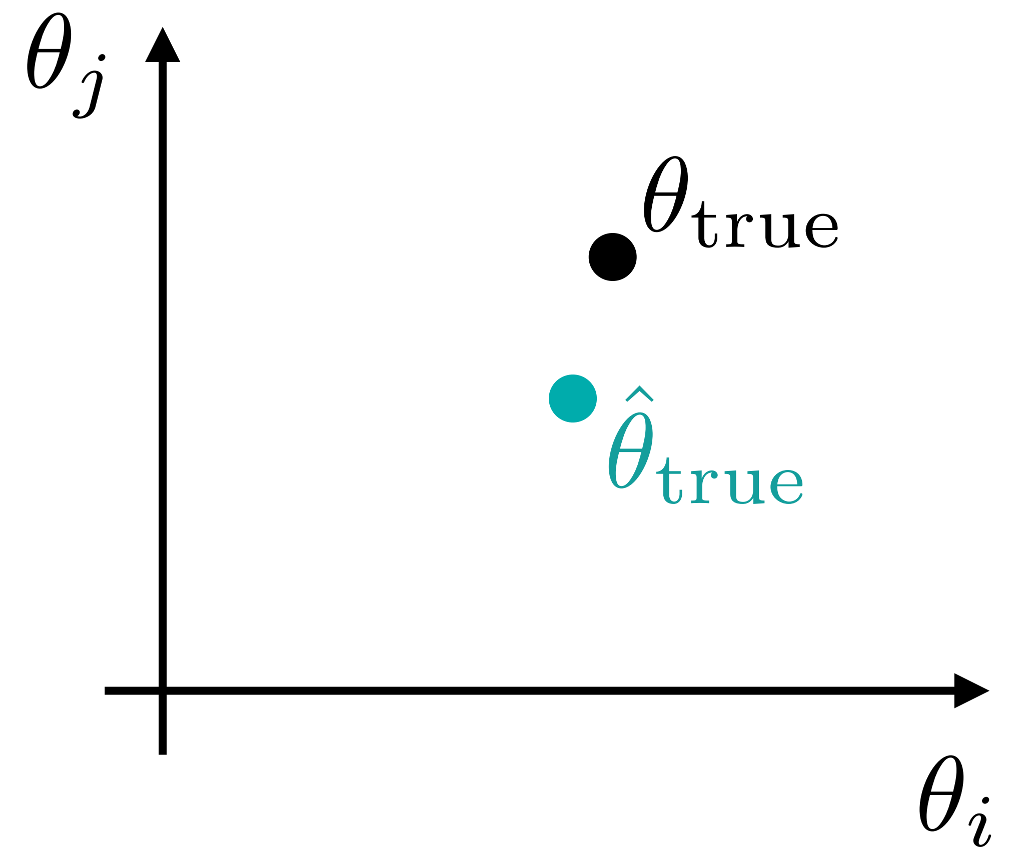
- a simulator that lets you generate N samples $x_i \sim p(x_i|\theta_i)$ (for parameters θ_i of our choice),
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$, and
- a prior $p(\theta)$,

Three problem statements

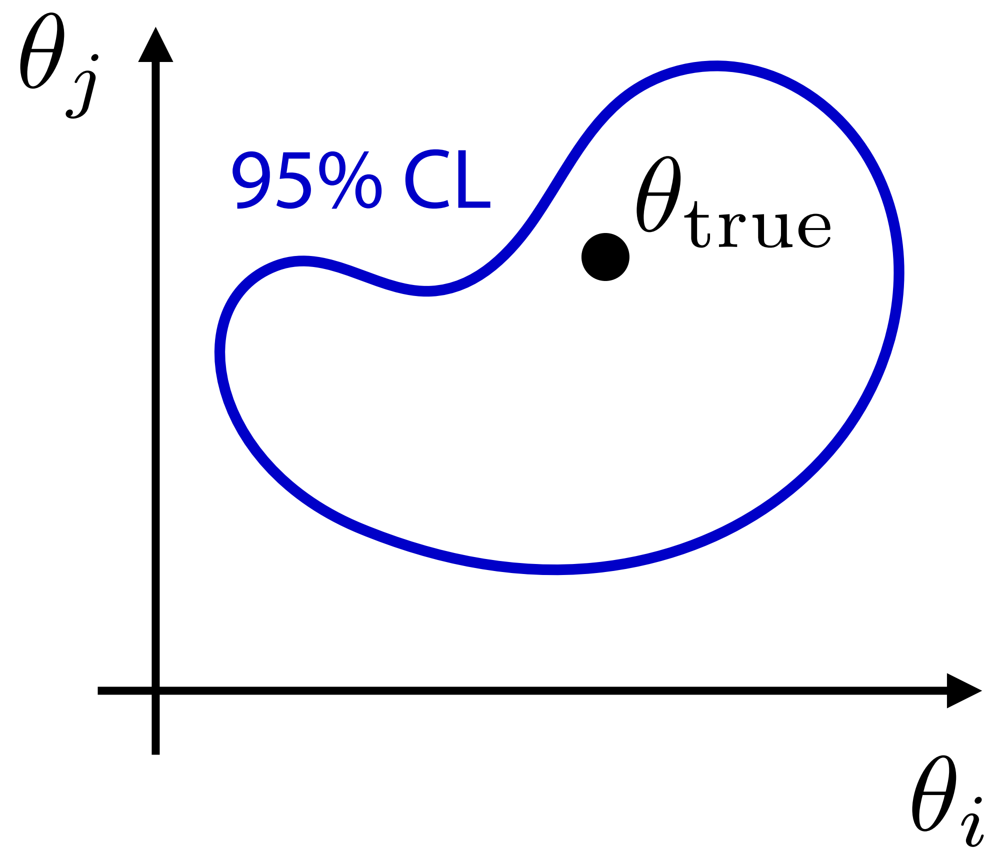
Given

- a simulator that lets you generate N samples $x_i \sim p(x_i|\theta_i)$ (for parameters θ_i of our choice),
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$, and
- a prior $p(\theta)$,

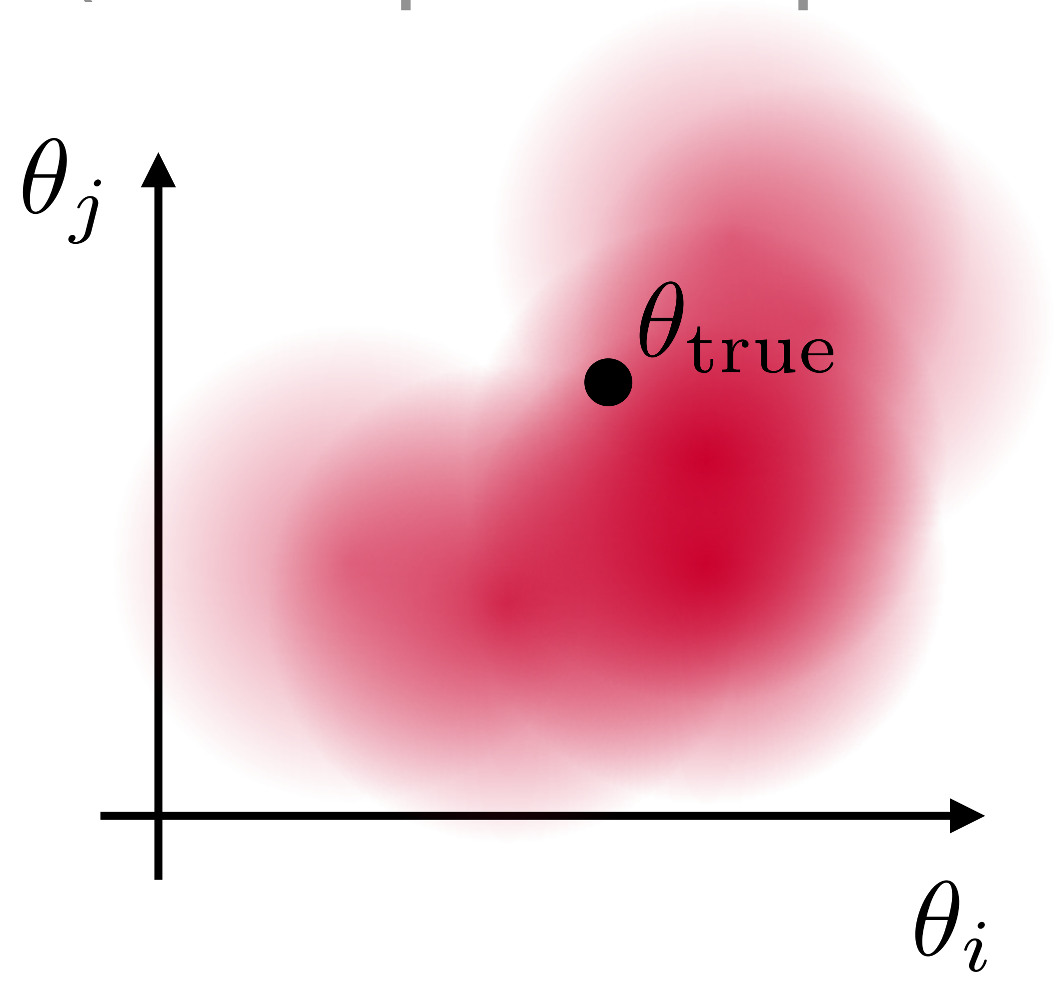
a) estimate $\hat{\theta}_{\text{true}}$
(e.g. MLE)

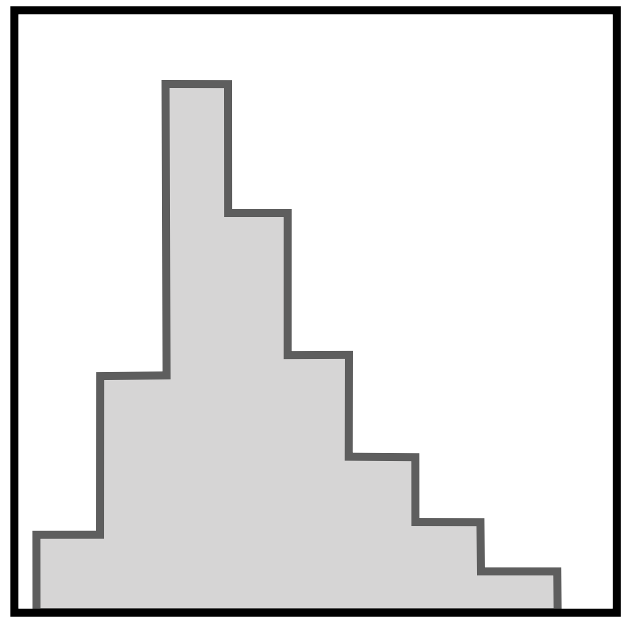


b) construct confidence sets
(e.g. likelihood ratio tests)



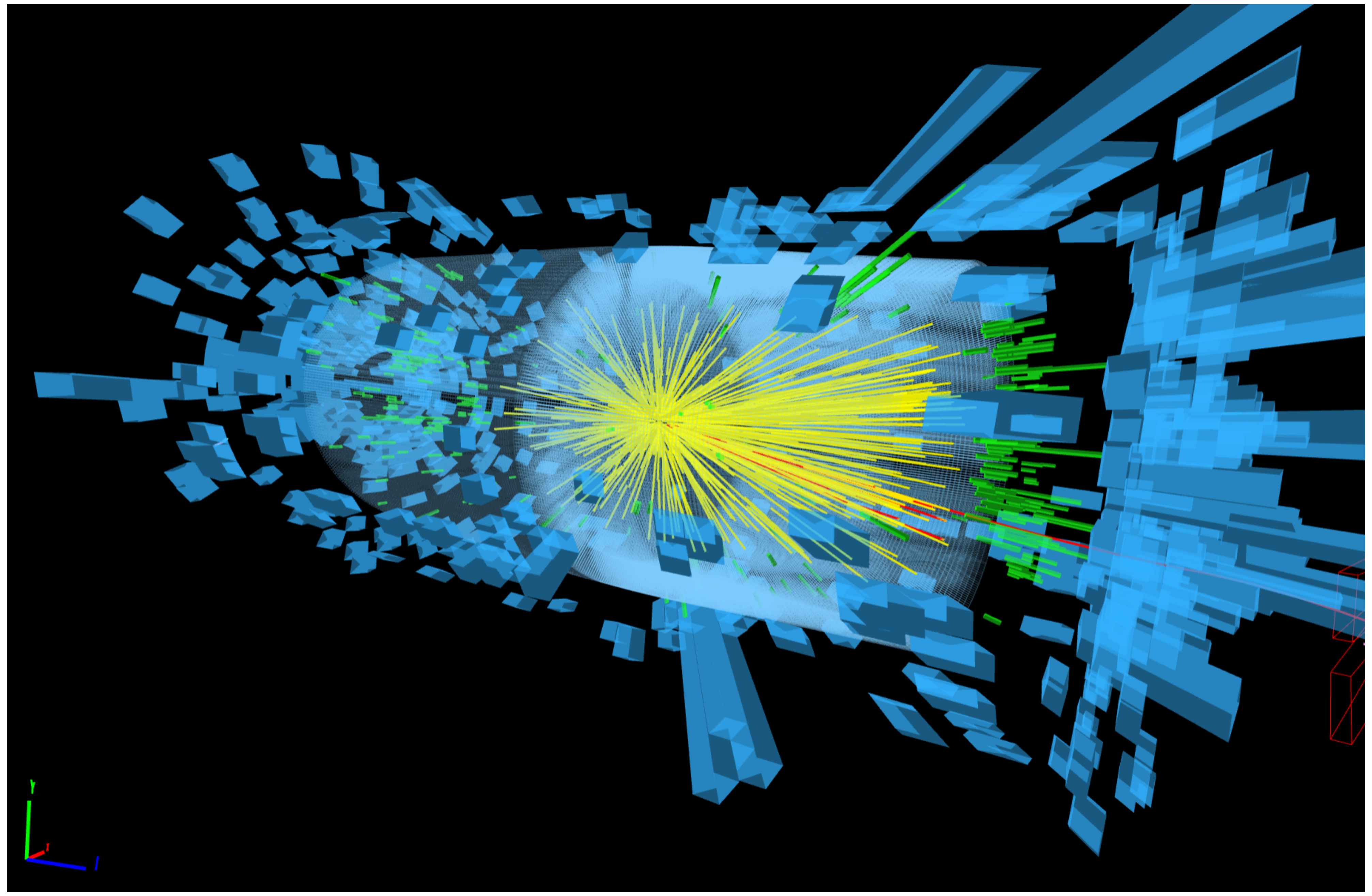
c) estimate the posterior
(or sample from posterior)





2. Why has that not stopped us before?

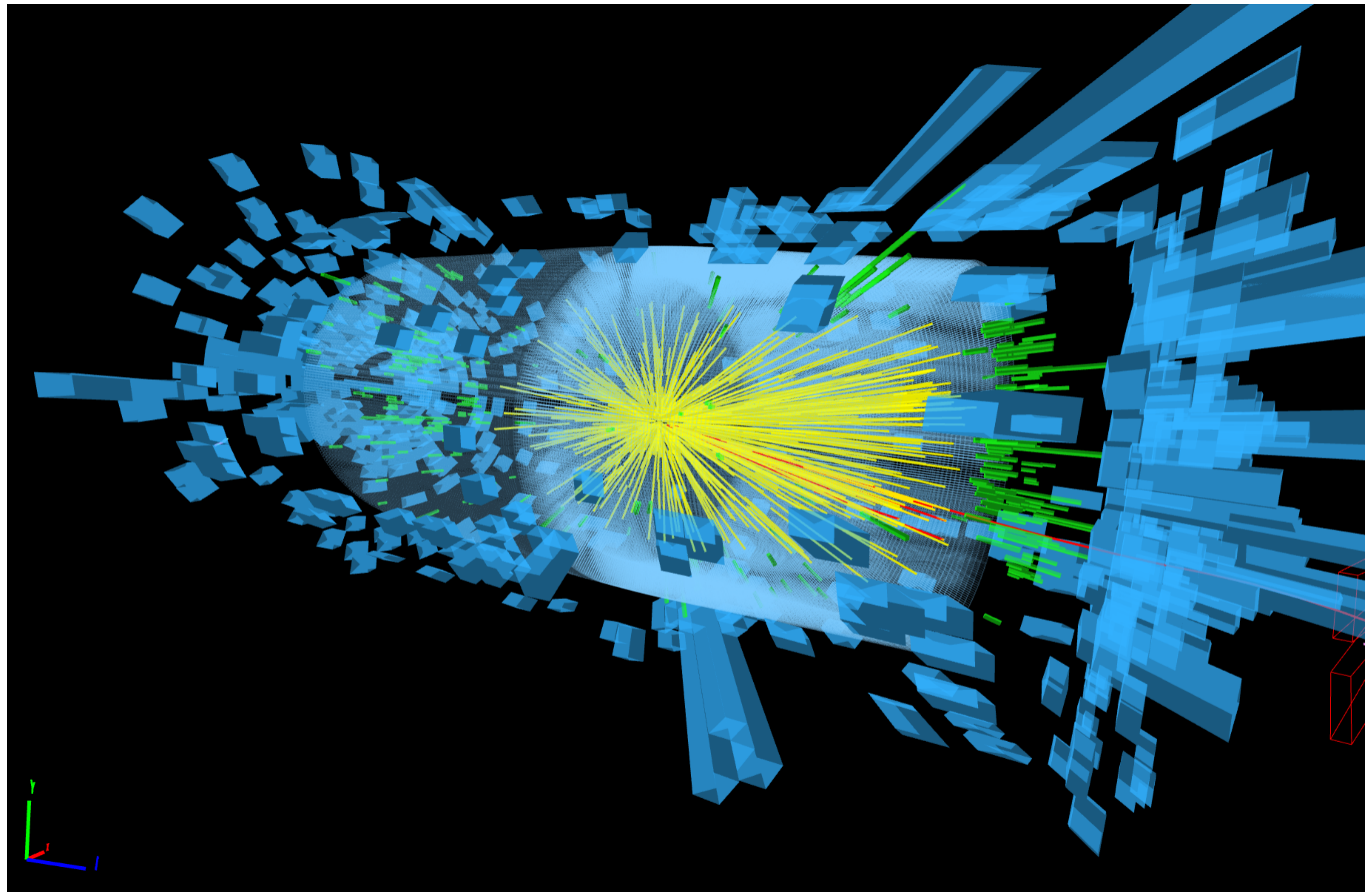
Solve it with summary statistics

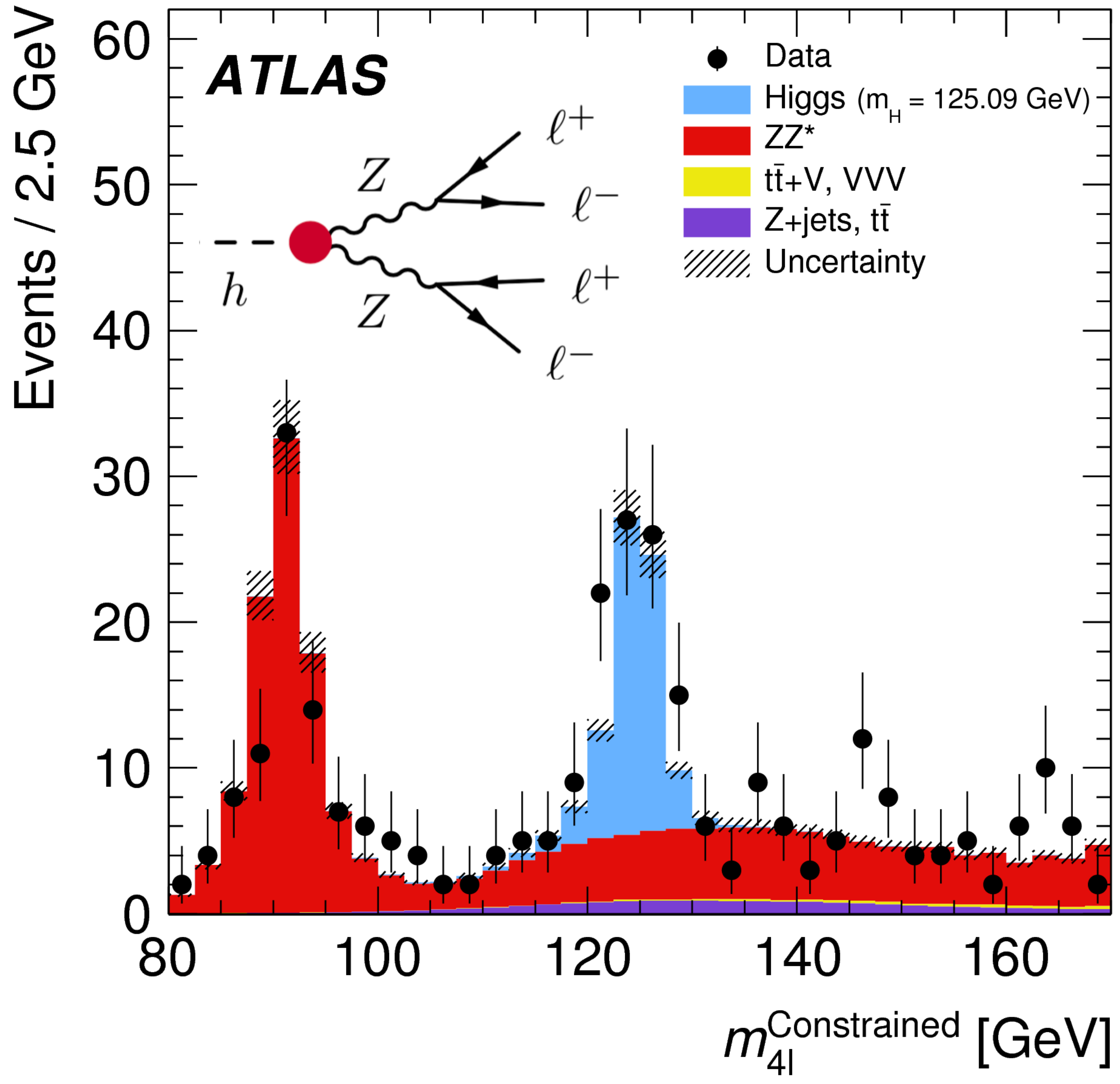


High-dimensional event data x

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

Solve it with summary statistics





[ATLAS 1712.02304]

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, …

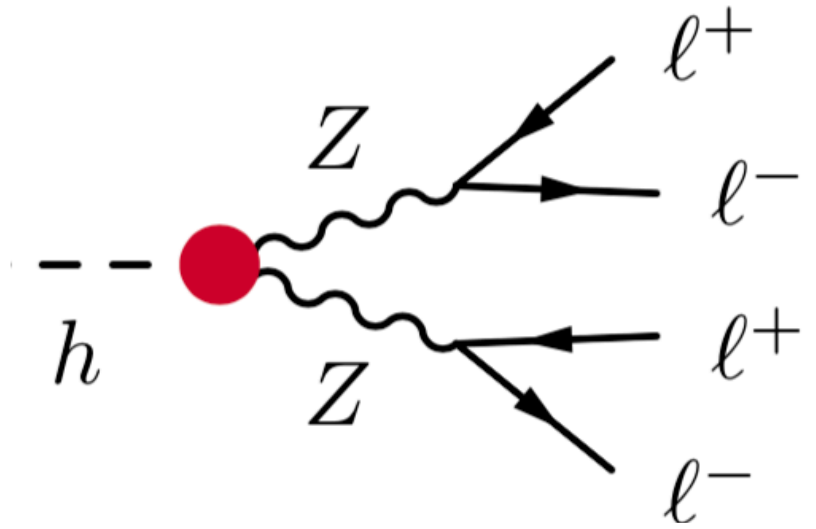
Summary statistics for LHC measurements?

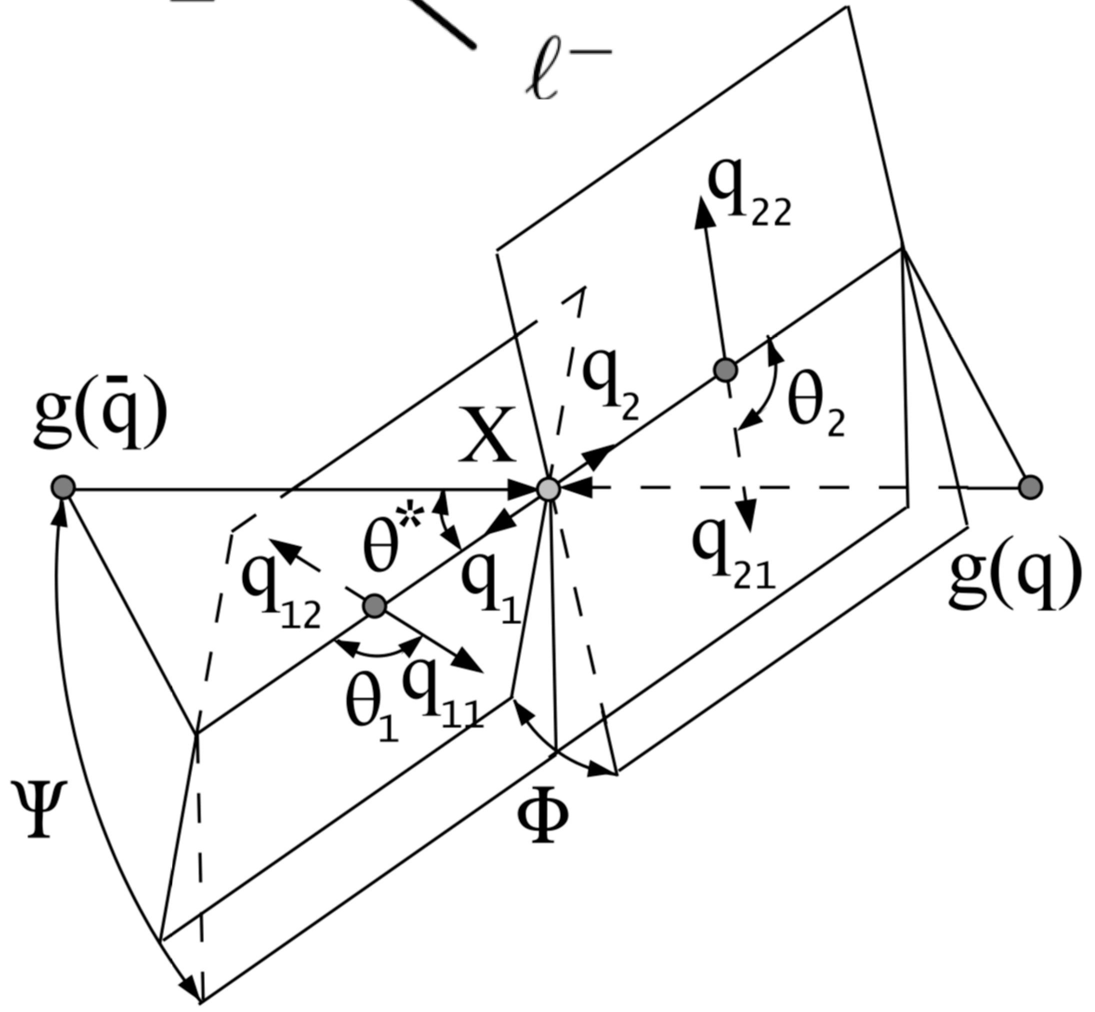
- In many LHC problems there is no single good summary statistics: compressing to any x' loses information!
  [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261; JB, F. Kling, T. Plehn, T. Tait 1712.02350]

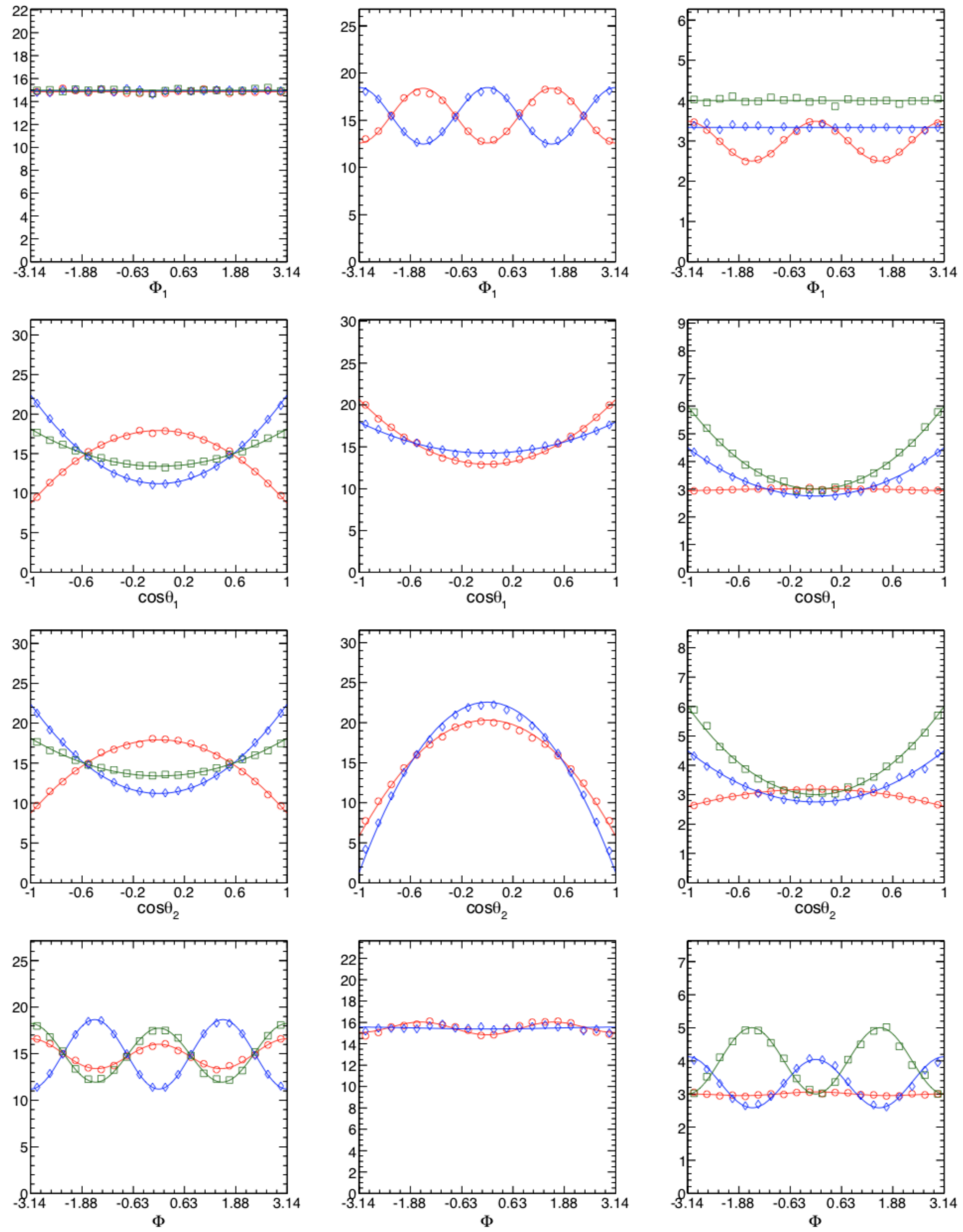
- Ideally: analyze all trustworthy high-level features (reconstructed four-momenta…), or some form of low-level features, including correlations
  (“fully differential cross section”)

Summary statistics for LHC measurements?

- In many LHC problems there is no single good summary statistics: compressing to any x' loses information!
  [JB, K. Cranmer, F. Kling, T. Plehn 1612.05261; JB, F. Kling, T. Plehn, T. Tait 1712.02350]







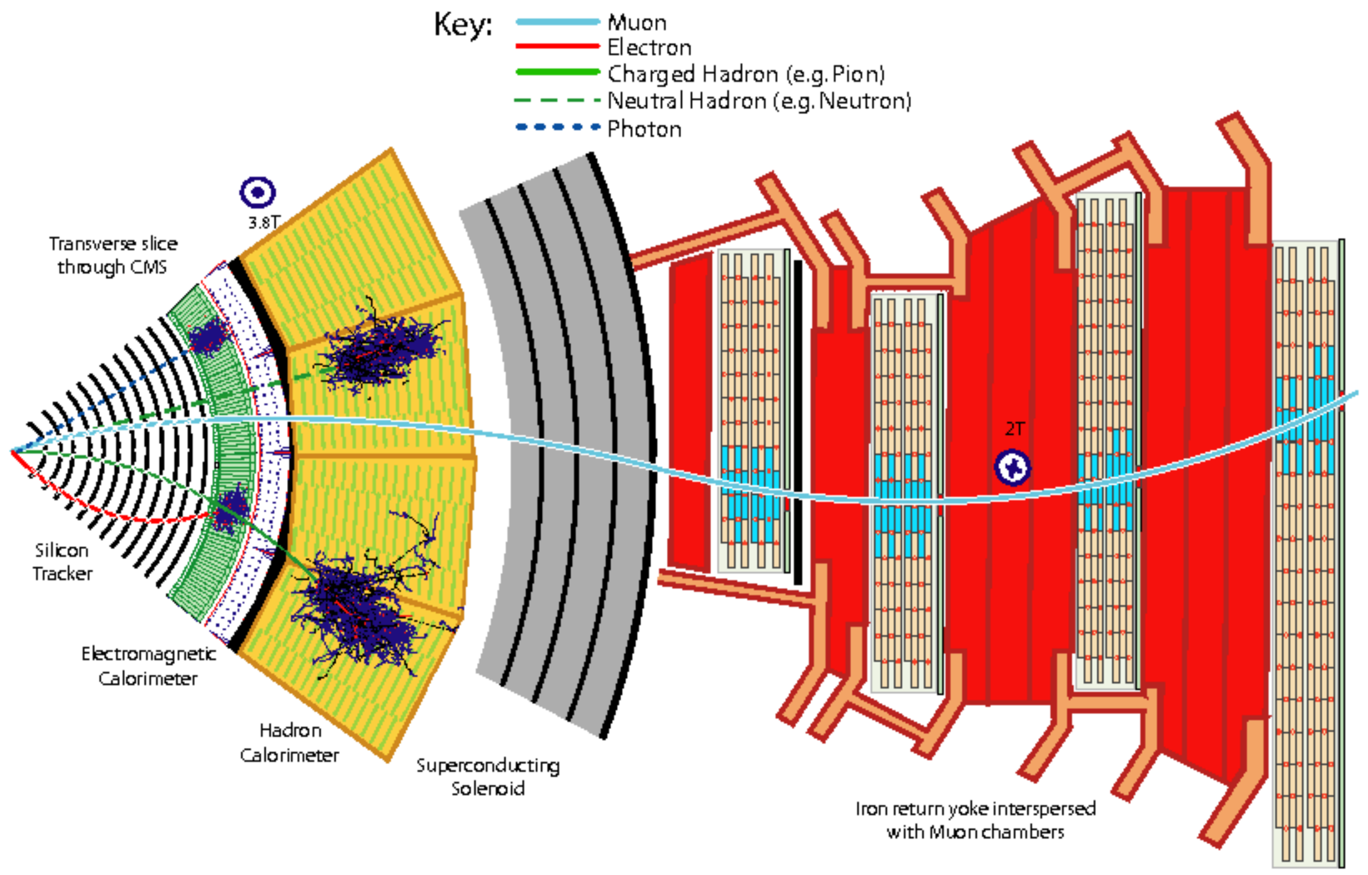
[Bolognesi et al. 1208.4018]

- Ideally: analyze all trustworthy high-level features (reconstructed four-momenta…), or some form of low-level features, including correlations
  (“fully differential cross section”)

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]

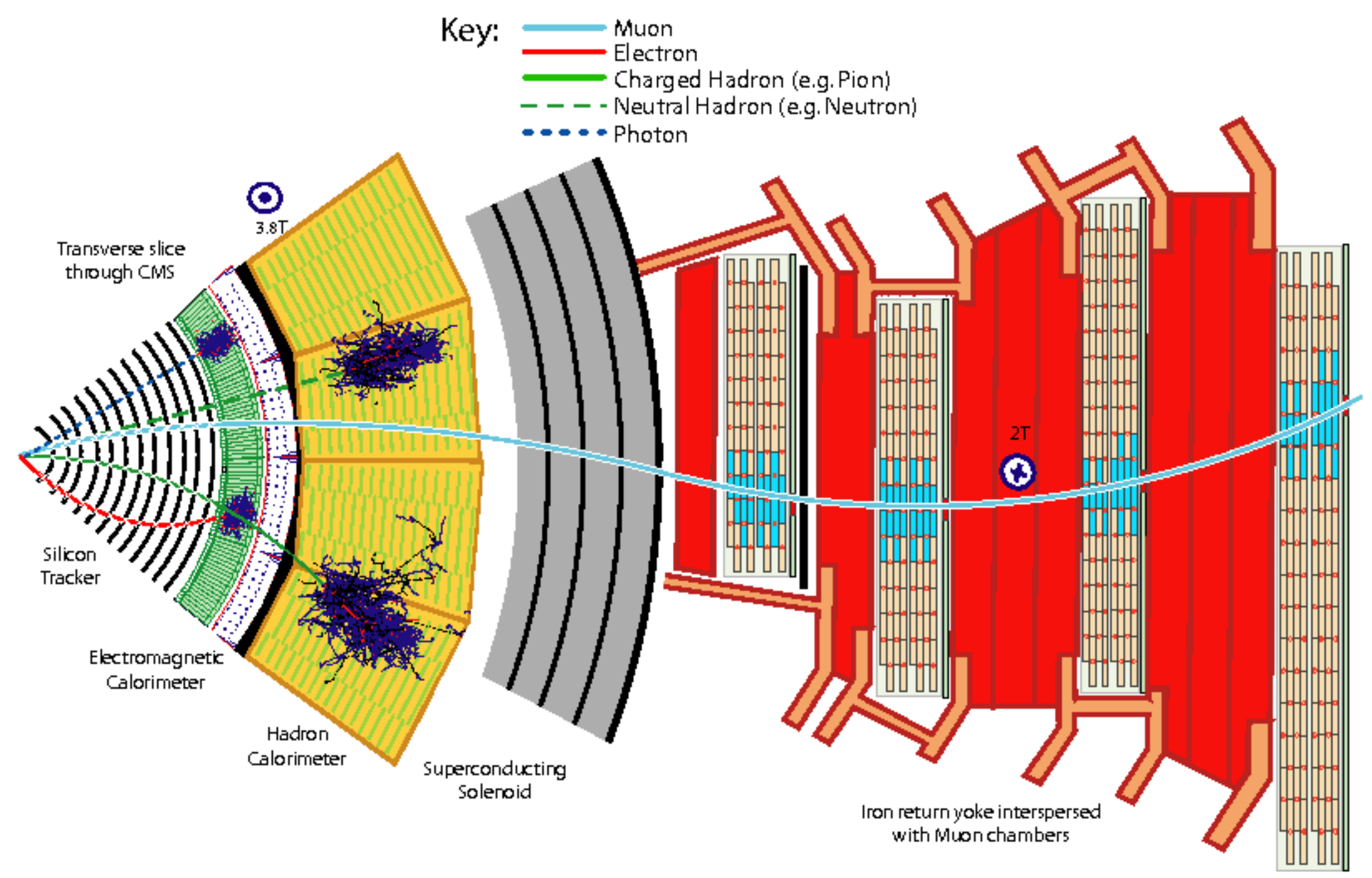
Solve it by approximating the integral

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

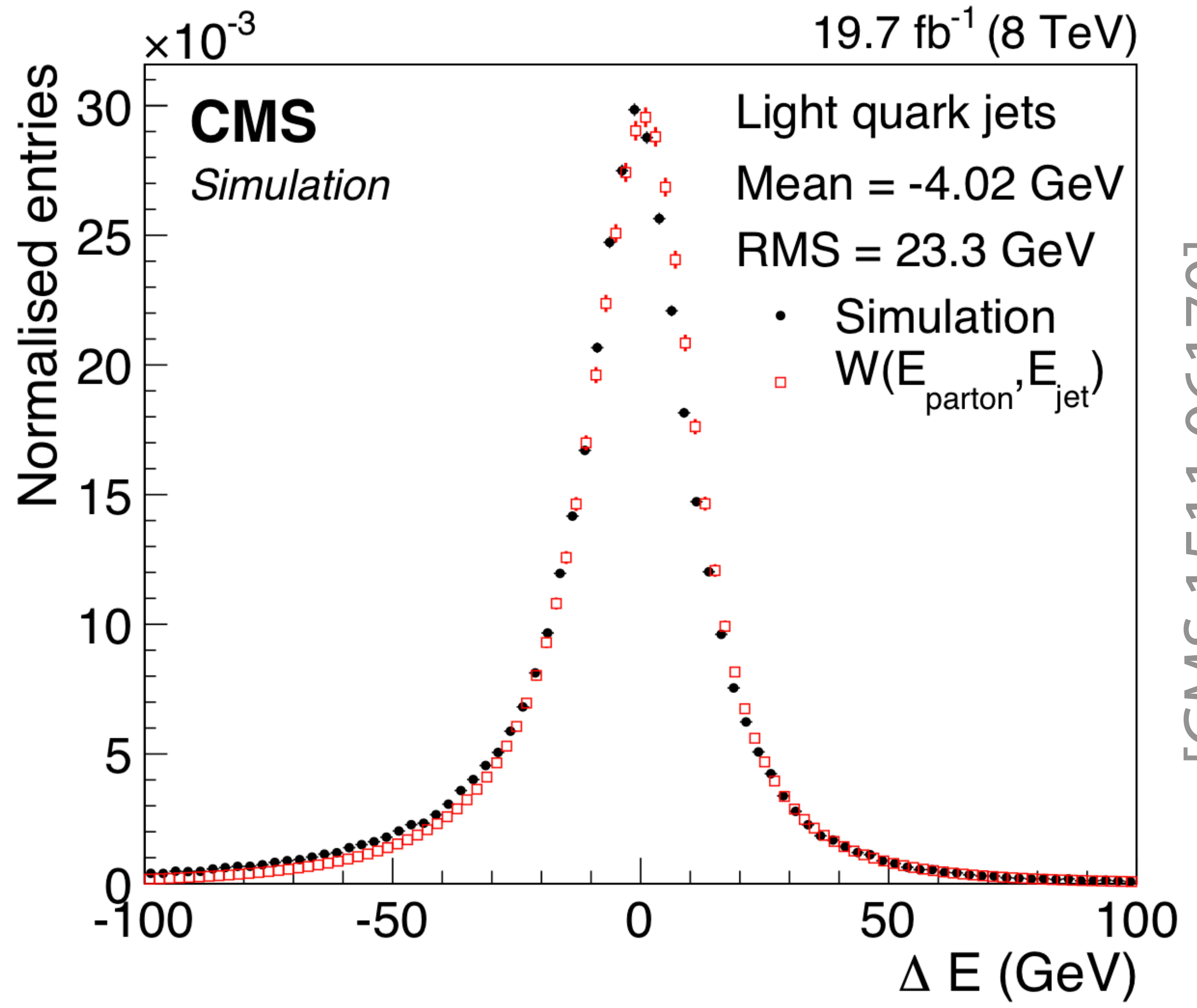
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$

- Matrix Element Method (and similarly Optimal Observables): [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]

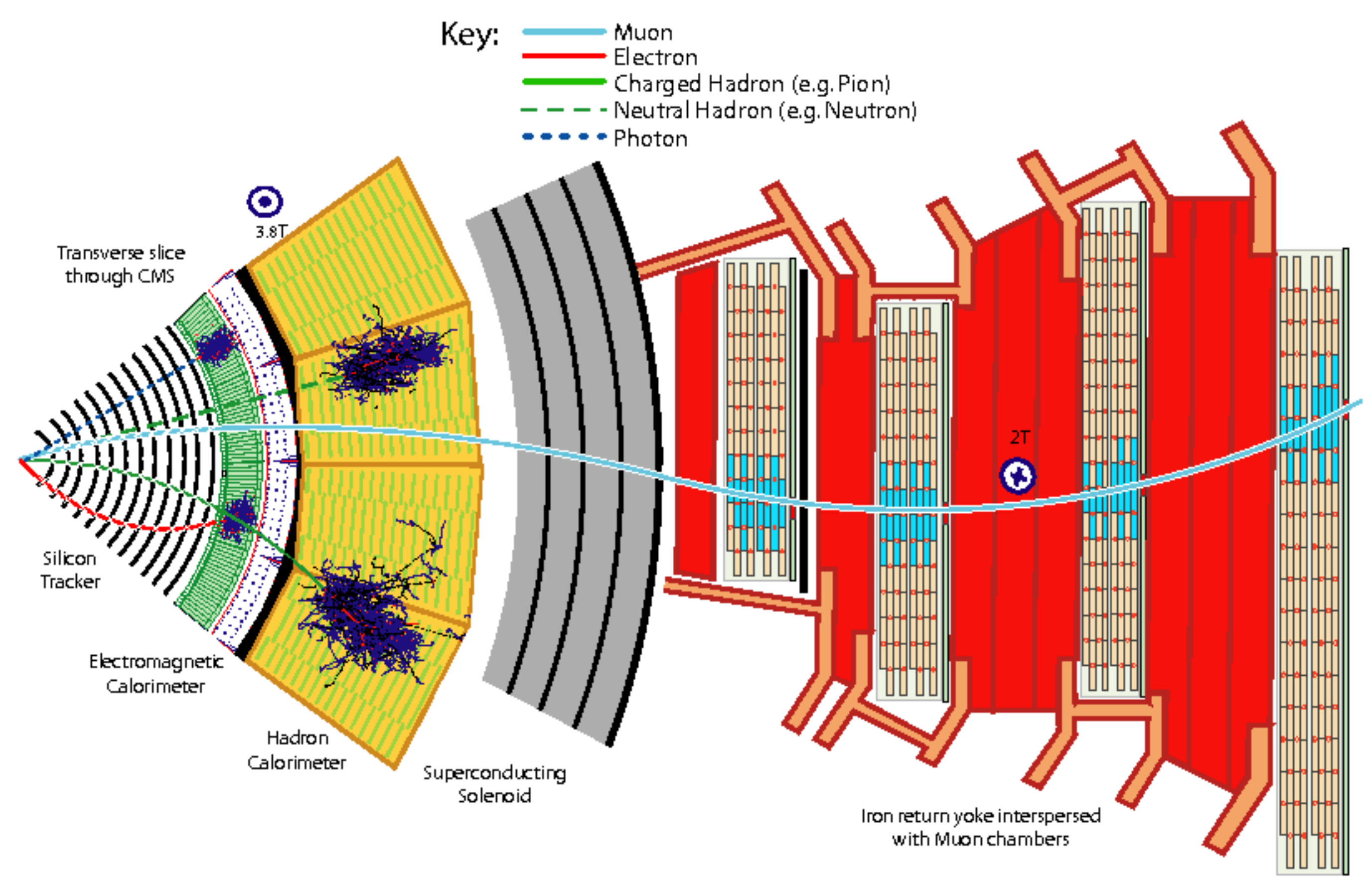


[CMS 1511.06170]

Solve it by approximating the integral

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

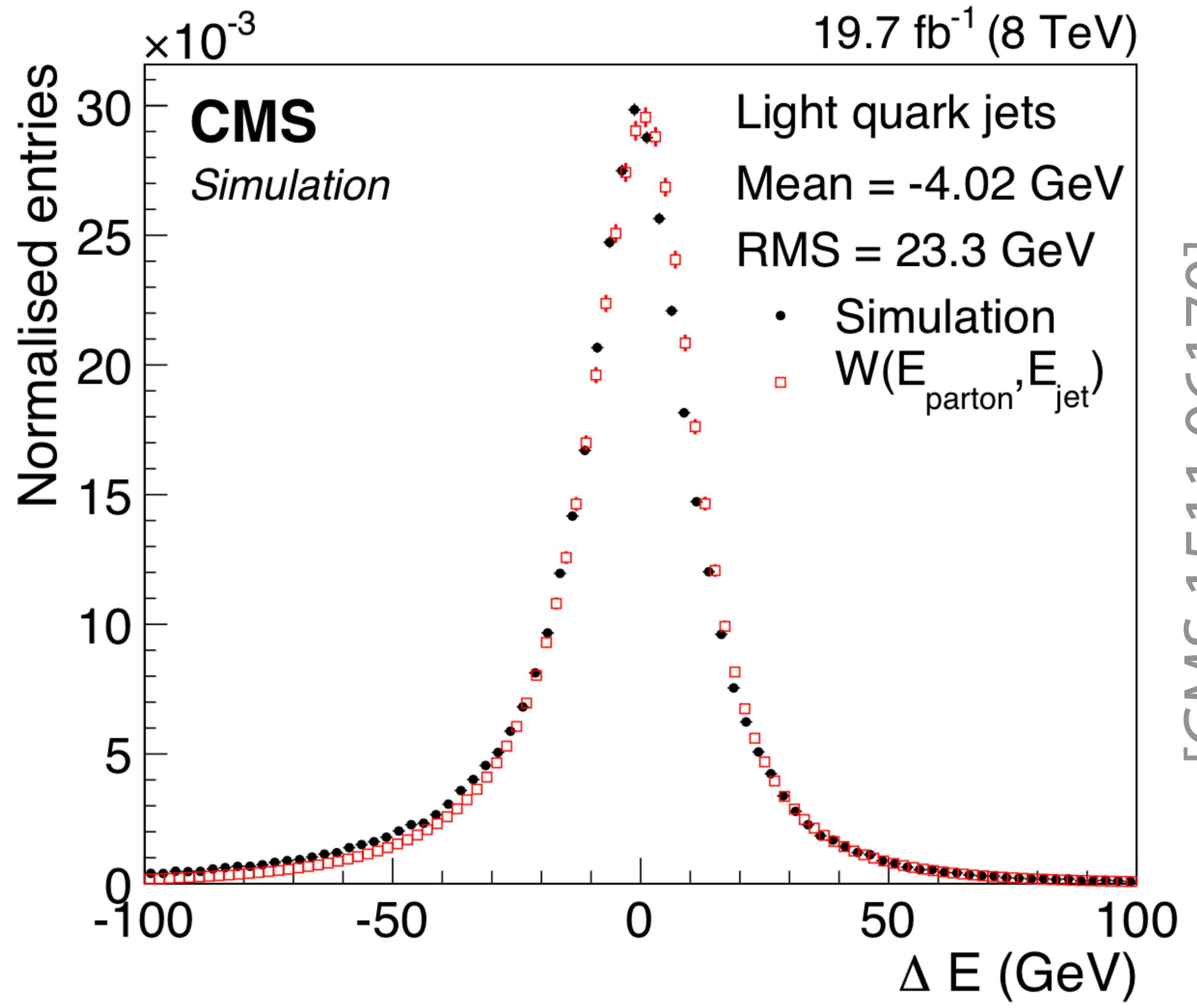
$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \; p(x|z_d)\, p(z_d|z_s)\, p(z_s|z_p)\, p(z_p|\theta)$$



[CMS]

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

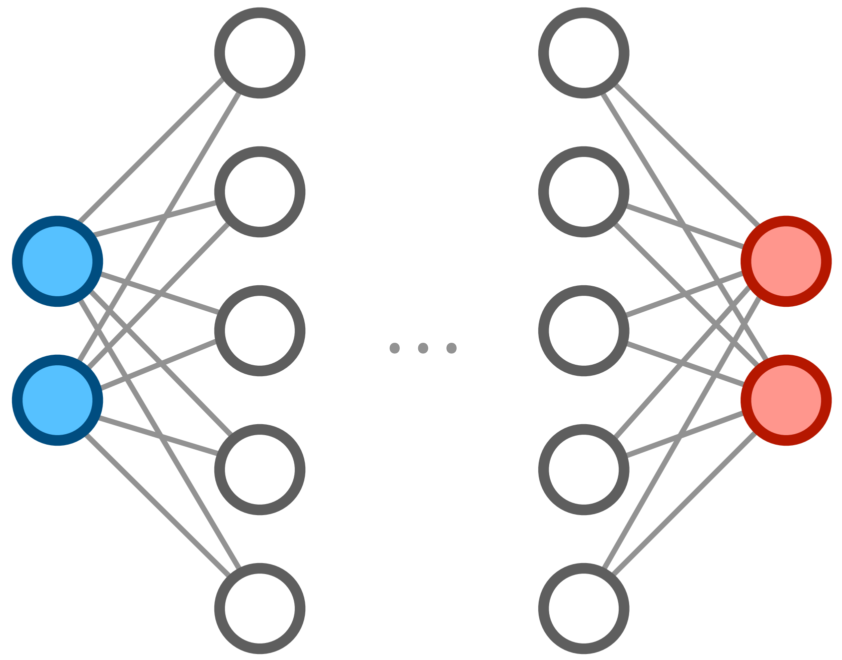
$$\hat{p}(x|\theta) = \int \mathrm{d}z_p \; \hat{p}(x|z_p)\, p(z_p|\theta)$$



[CMS 1511.06170]

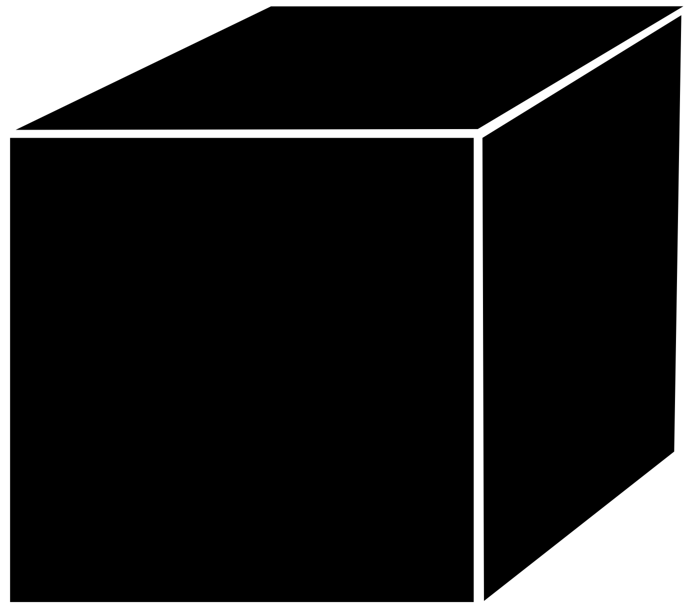
⇒ Uses matrix-element information, no summary statistics necessary, but:

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



3. Machine learning solutions

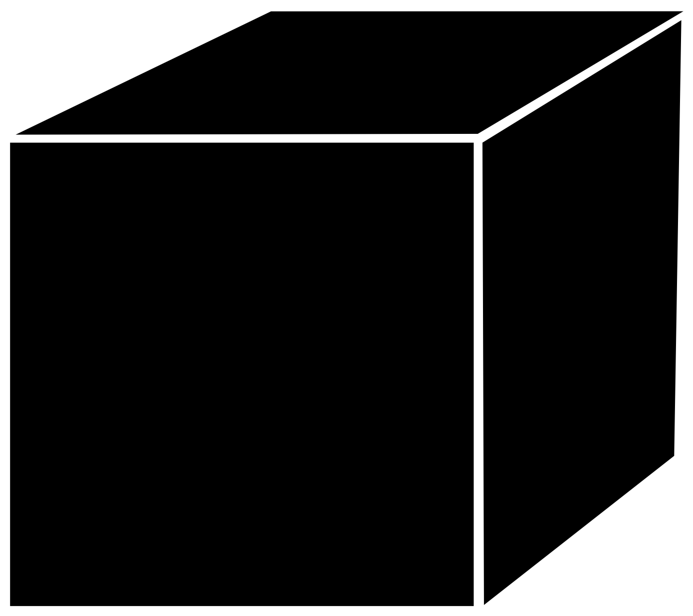
Get the best of two worlds



Simulators: focus on understanding

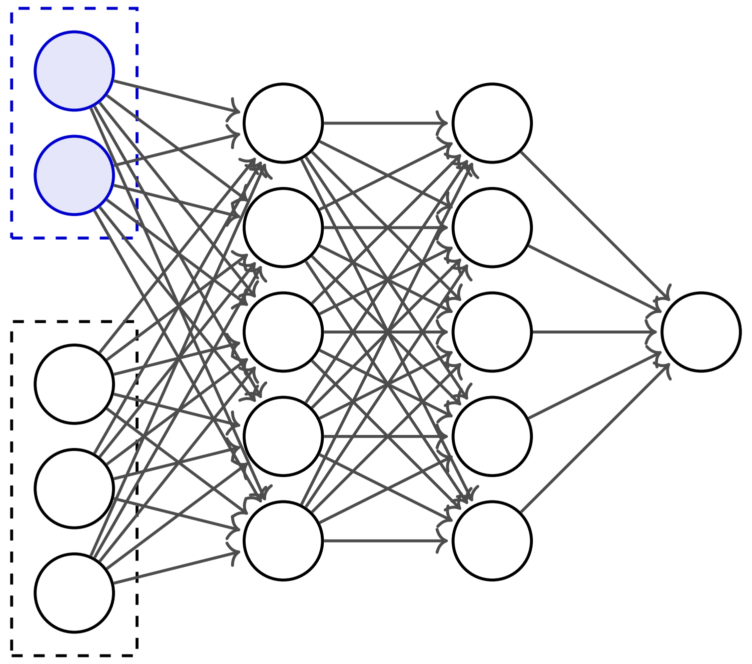
- based on mechanistic, causal model
- interpretable parameters

Get the best of two worlds



Simulators: focus on understanding

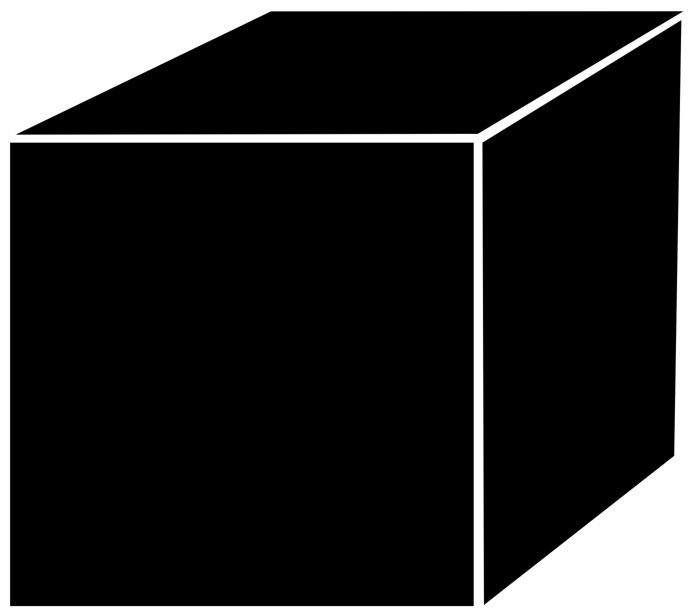
- based on mechanistic, causal model
- interpretable parameters

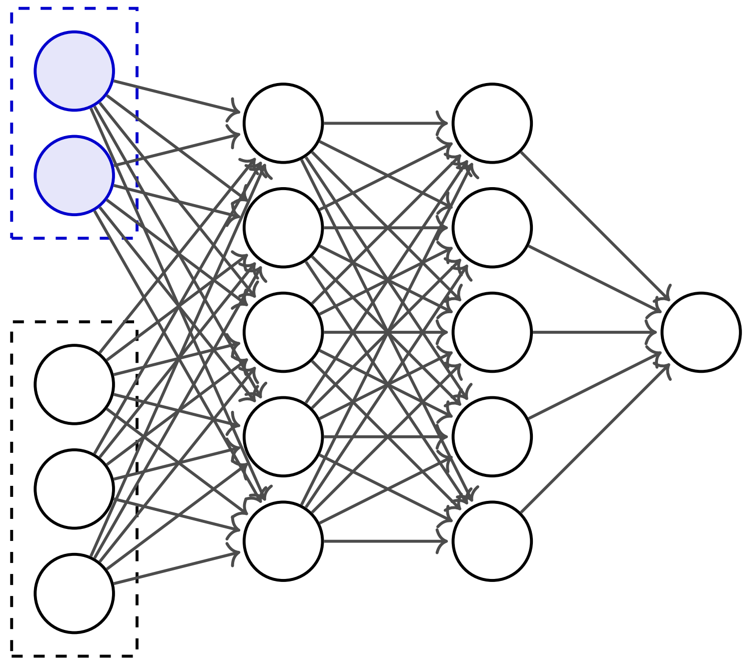


Machine learning models: focus on performance

- good at learning representations from data
- good inductive biases (images, sequences, graphs, symmetries, hierarchical structures…)
- differentiable, often invertible, probabilistic: well-suited for inference / fitting

Get the best of two worlds





Can we use ML models to fit simulators to data?

Simulators: focus on understanding

- based on mechanistic, causal model
- interpretable parameters

Can we inject domain knowledge into ML models?

Machine learning models: focus on performance

- good at learning representations from data
- good inductive biases (images, sequences, graphs, symmetries, hierarchical structures…)
- differentiable, often invertible, probabilistic: well-suited for inference / fitting

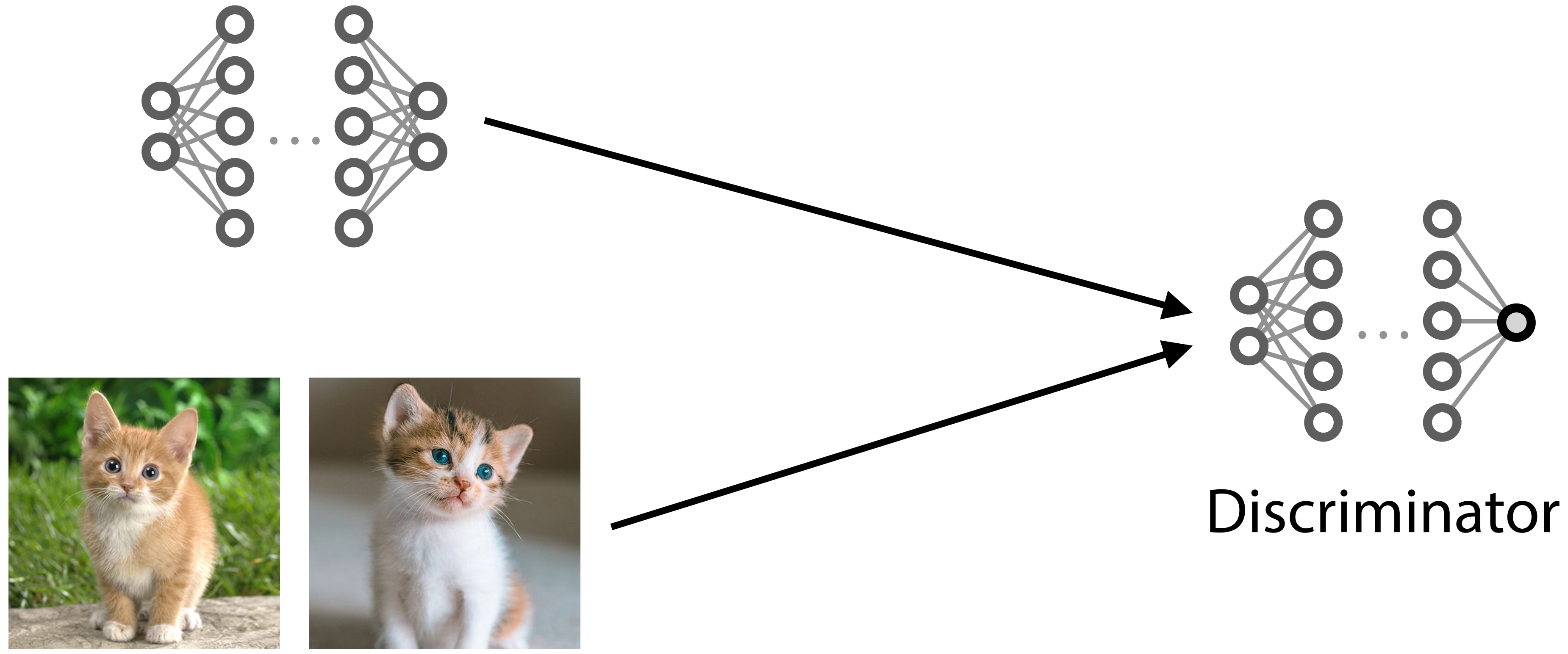
Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\rm gen}(x)$)

Examples
(intractable likelihood $p_{\rm true}(x)$)



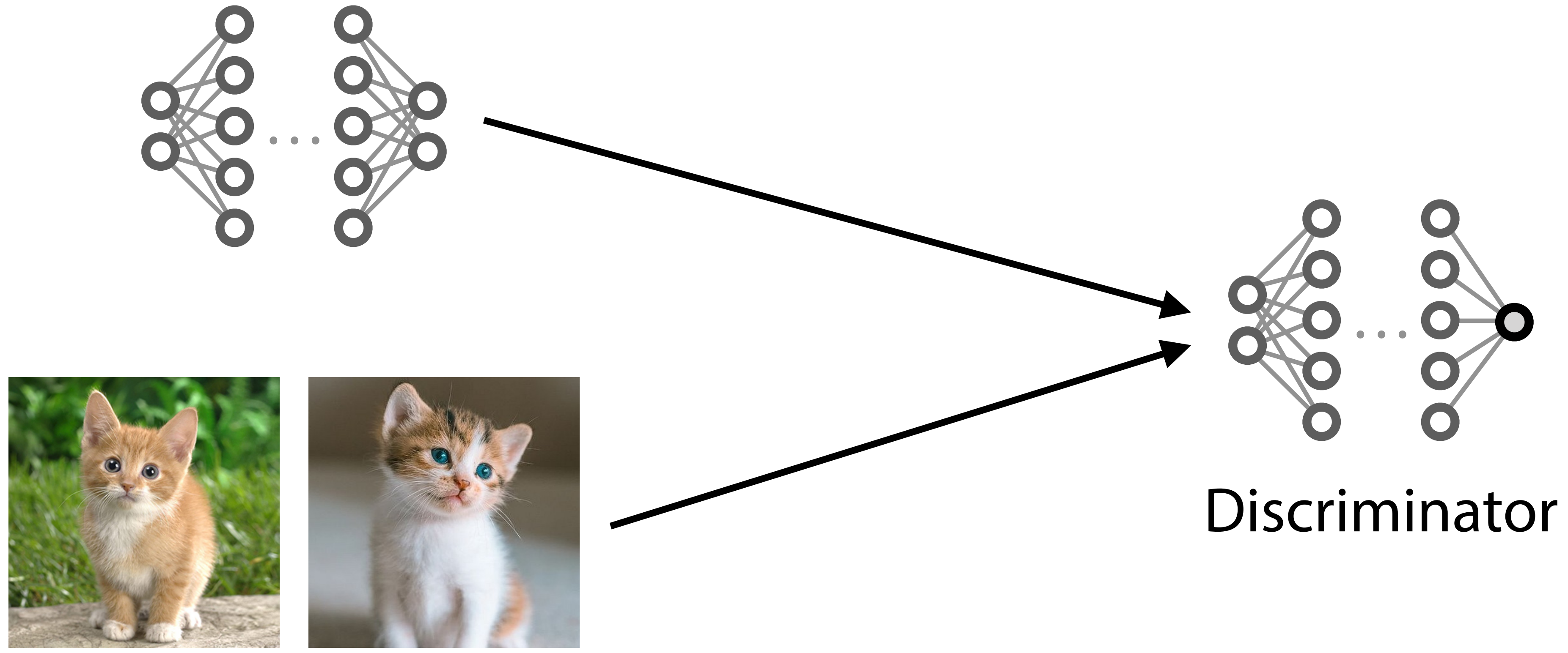
Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs):

[I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\mathrm{gen}}(x)$)

Examples
(intractable likelihood $p_{\mathrm{true}}(x)$)



Discriminator learns decision function

$$s(x) \to \frac{p_{\mathrm{true}}(x)}{p_{\mathrm{gen}}(x) + p_{\mathrm{true}}(x)}$$

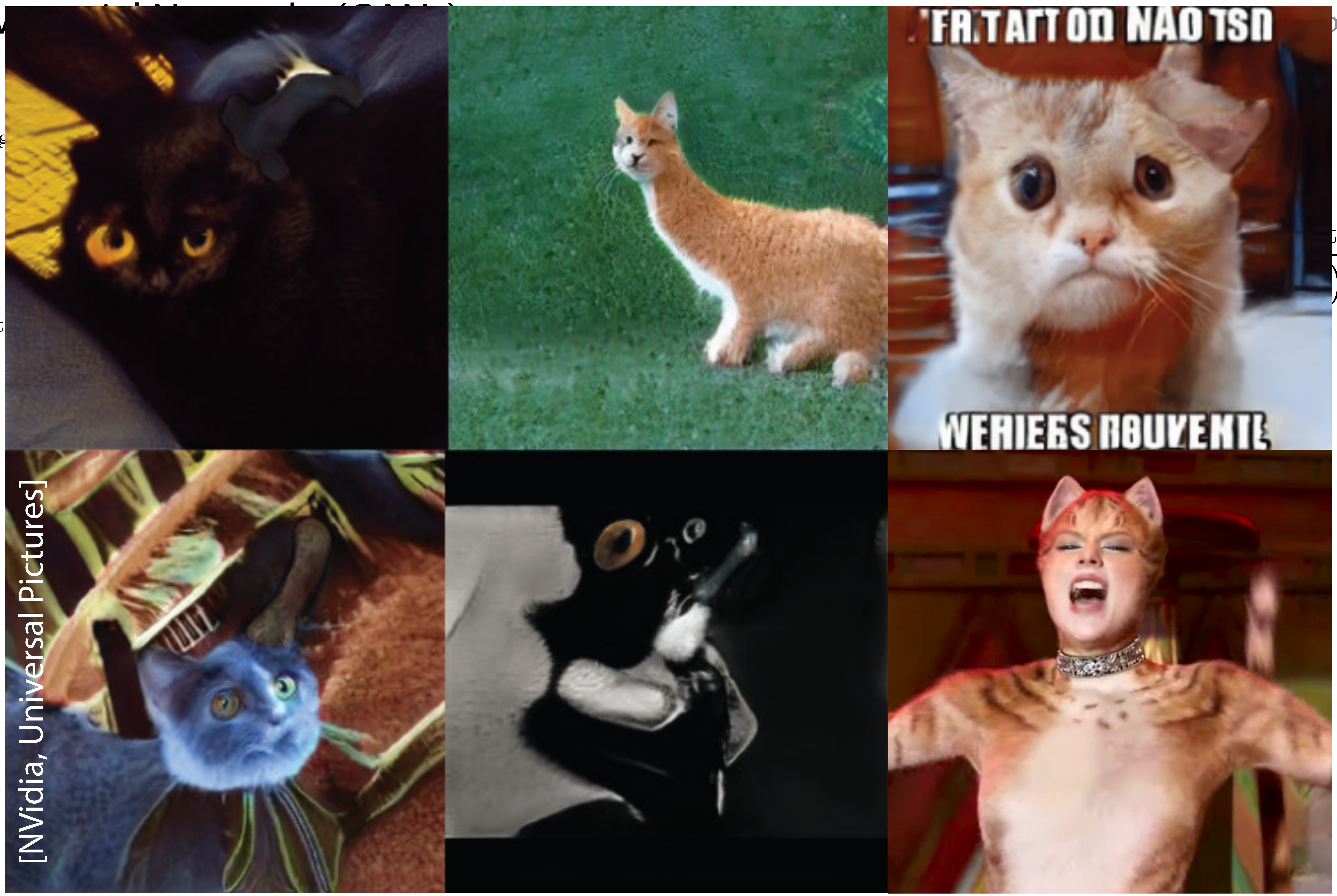
Idea 1: the likelihood ratio trick

- Generative Adv

[…oodfellow et al. 1406.2661]

Generator
(intractable likelihood p_{g}

Examples
(intractable likelihood p_{t}


[NVidia, Universal Pictures]

decision function

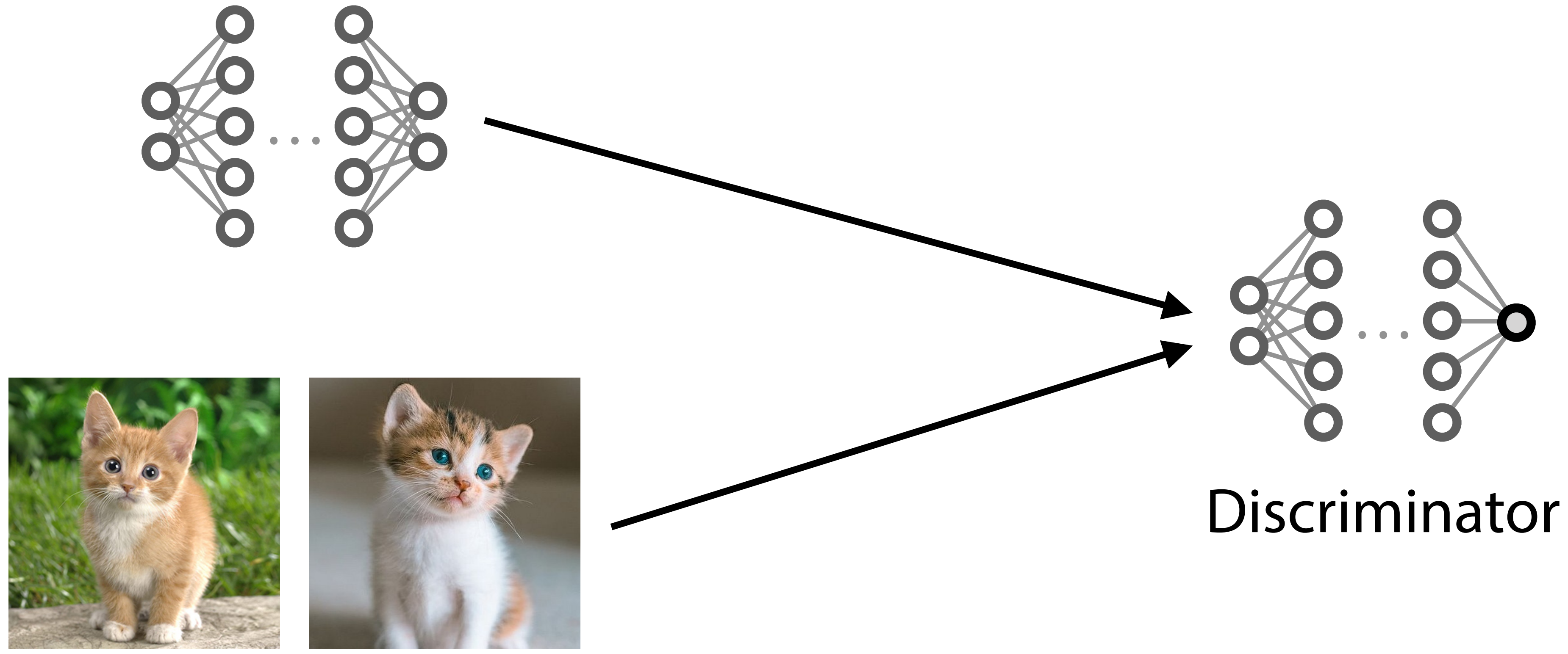
$$\frac{\dots_{\mathrm{true}}(x)}{\dots) + p_{\mathrm{true}}(x)}$$

Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs): [I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\text{gen}}(x)$)

Examples
(intractable likelihood $p_{\text{true}}(x)$)



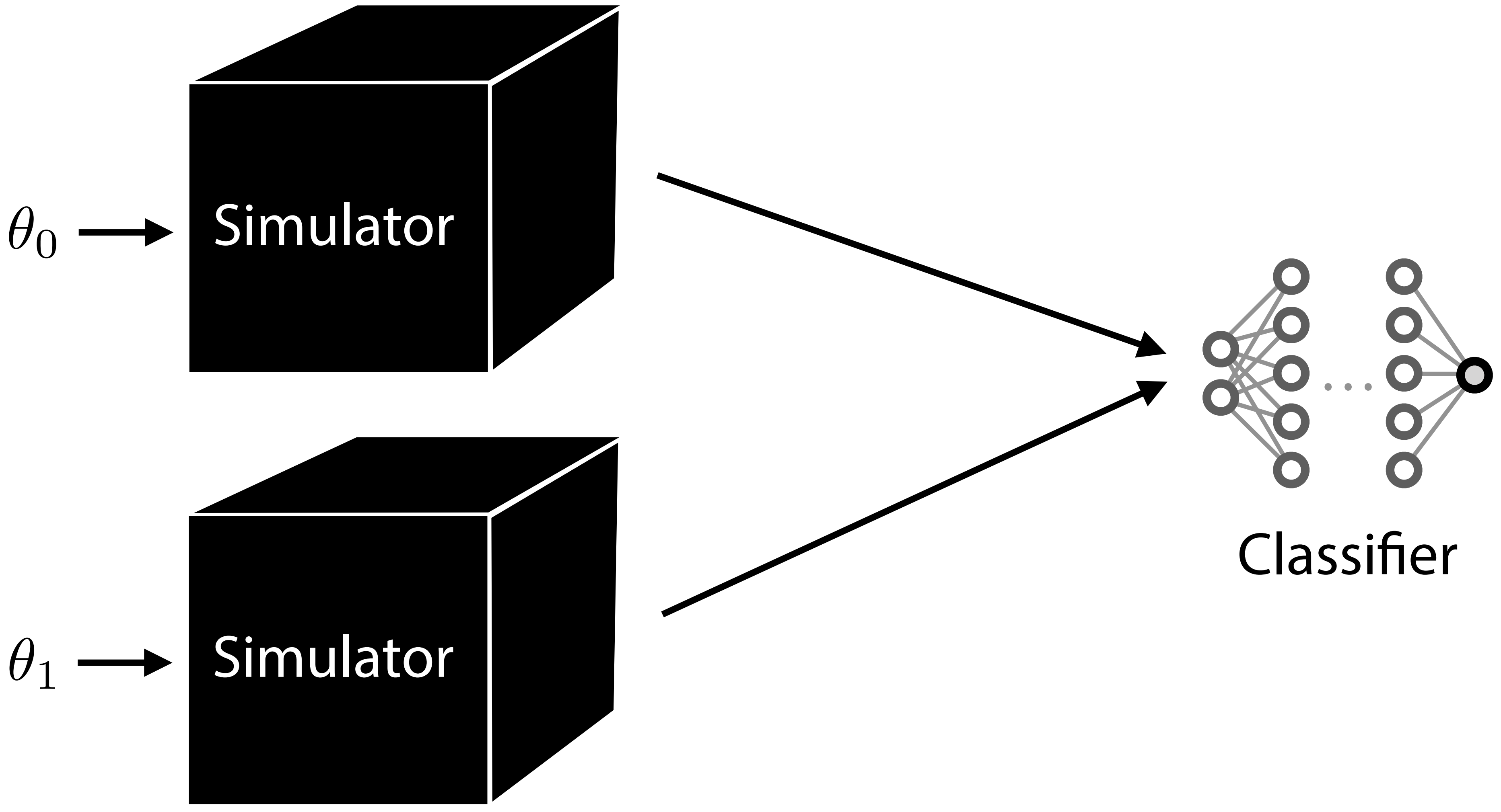
Discriminator learns decision function

$$s(x) \to \frac{p_{\text{true}}(x)}{p_{\text{gen}}(x) + p_{\text{true}}(x)}$$

- Similarly, we can train a classifier between two sets of simulated samples [K. Cranmer, J. Pavez, G. Louppe 1506.02169]

Simulator
(intractable likelihood $p(x|\theta_0)$)

Simulator
(intractable likelihood $p(x|\theta_1)$)

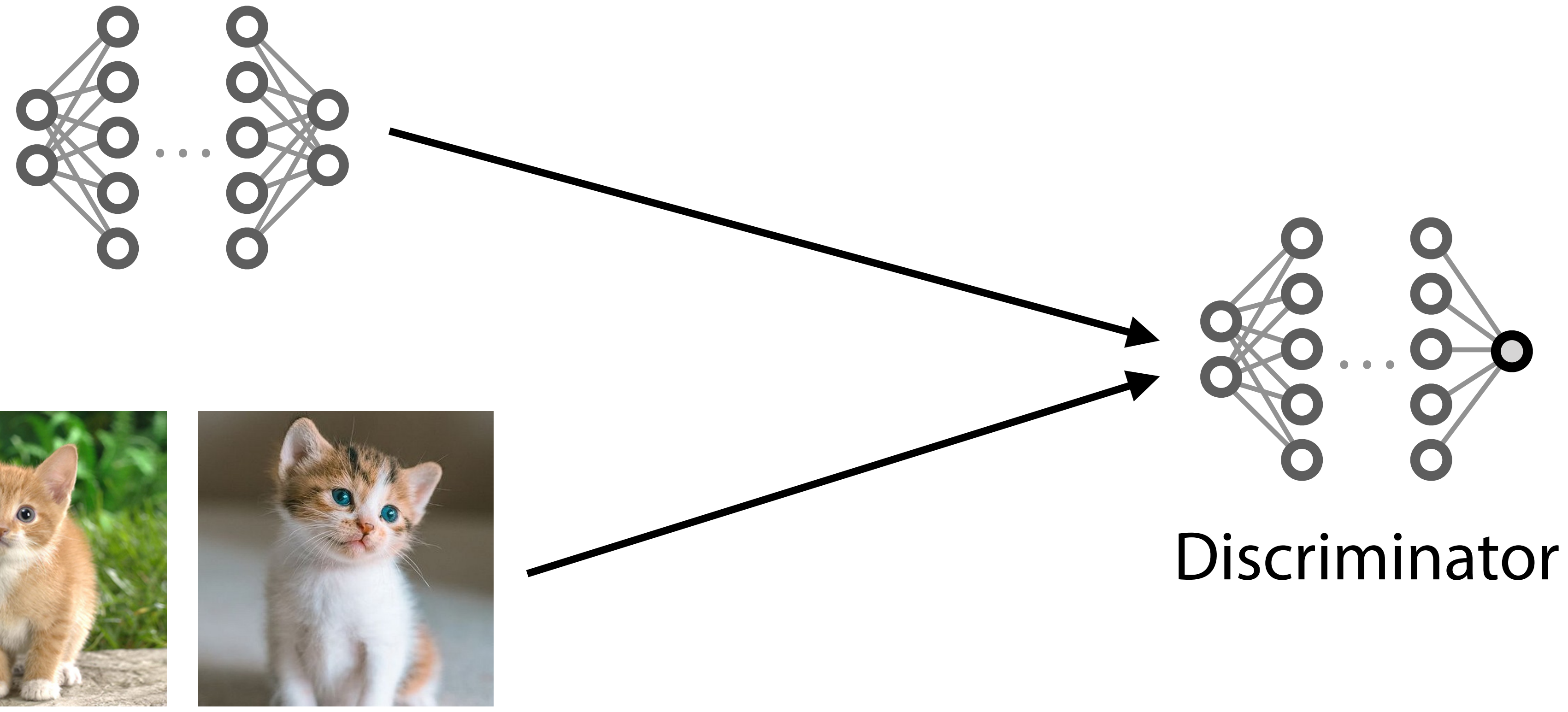


Idea 1: the likelihood ratio trick

- Generative Adversarial Networks (GANs): [I. Goodfellow et al. 1406.2661]

Generator
(intractable likelihood $p_{\mathrm{gen}}(x)$)

Examples
(intractable likelihood $p_{\mathrm{true}}(x)$)



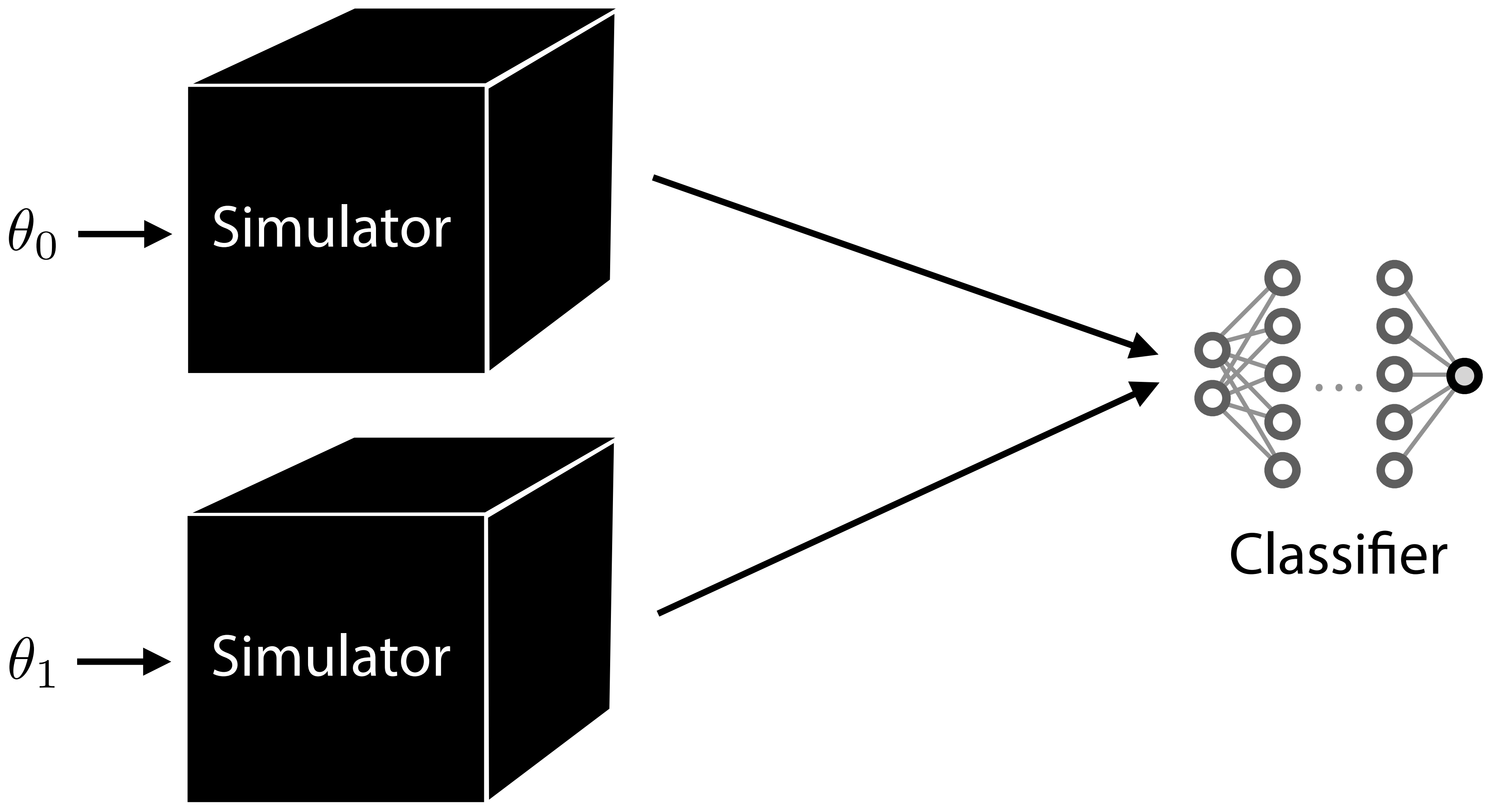
Discriminator learns decision function

$$s(x) \to \frac{p_{\mathrm{true}}(x)}{p_{\mathrm{gen}}(x) + p_{\mathrm{true}}(x)}$$

- Similarly, we can train a classifier between two sets of simulated samples [K. Cranmer, J. Pavez, G. Louppe 1506.02169]

Simulator
(intractable likelihood $p(x|\theta_0)$)

Simulator
(intractable likelihood $p(x|\theta_1)$)



Classifier learns decision function

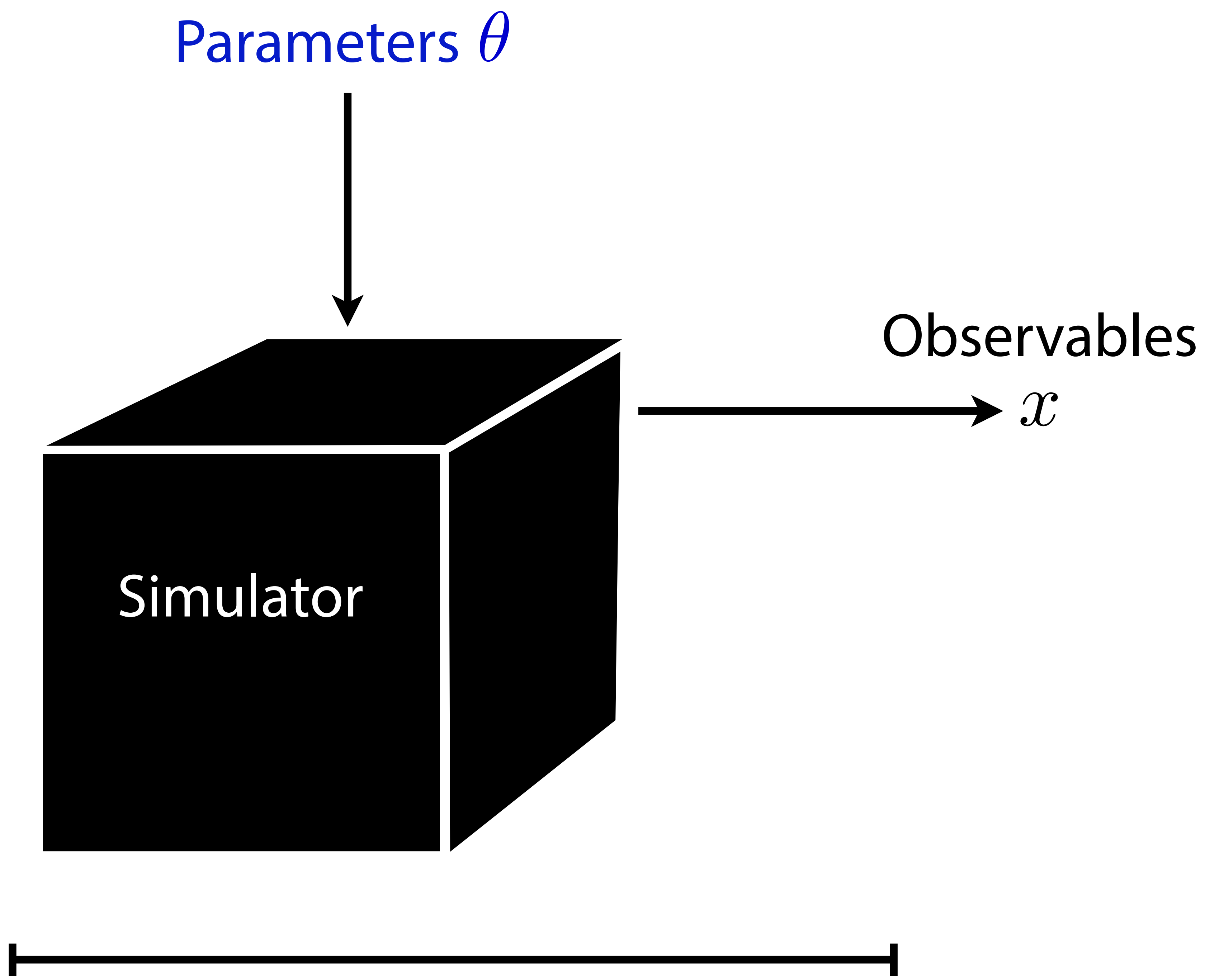
$$s(x) \to \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

$\Rightarrow$ Estimator for likelihood ratio

$$\hat{r}(x) = \frac{1 - s(x)}{s(x)} \to \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

Inference by likelihood ratio trick

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

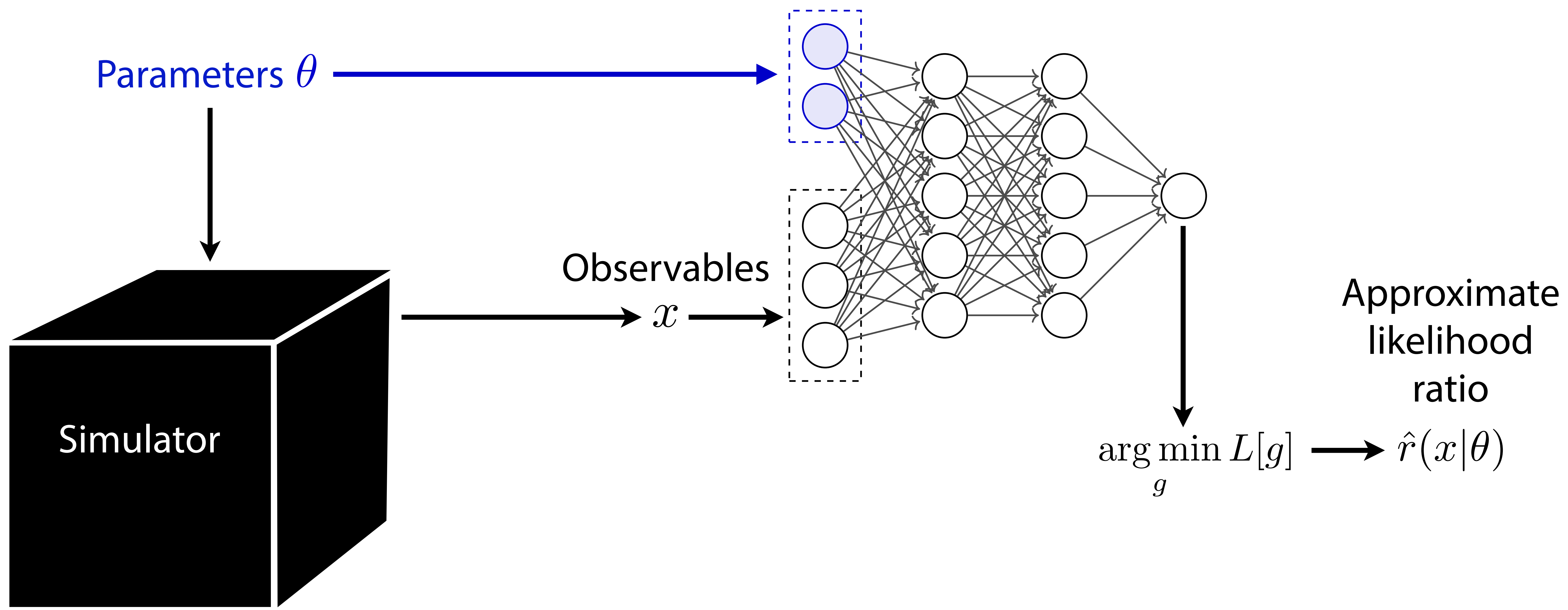


1. Simulation

Run simulator and save data

Inference by likelihood ratio trick

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



1. Simulation

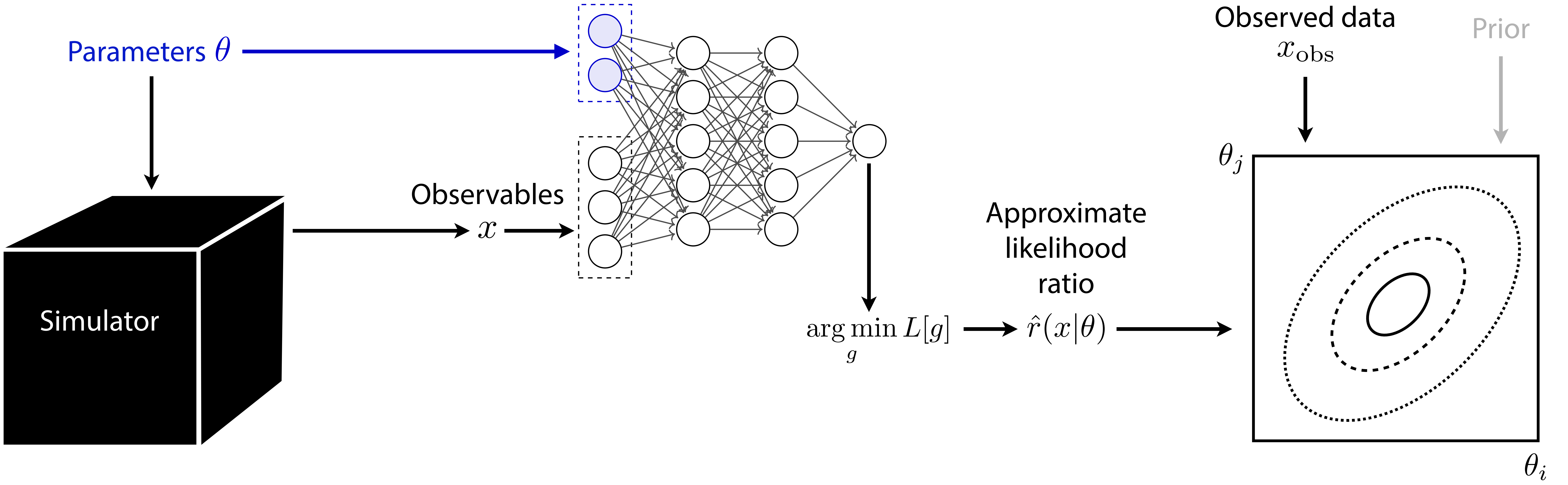
Run simulator and save data

2. Machine Learning

Train NN classifier, interpret as likelihood ratio estimator

Inference by likelihood ratio trick

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



1. Simulation

2. Machine Learning

3. Inference

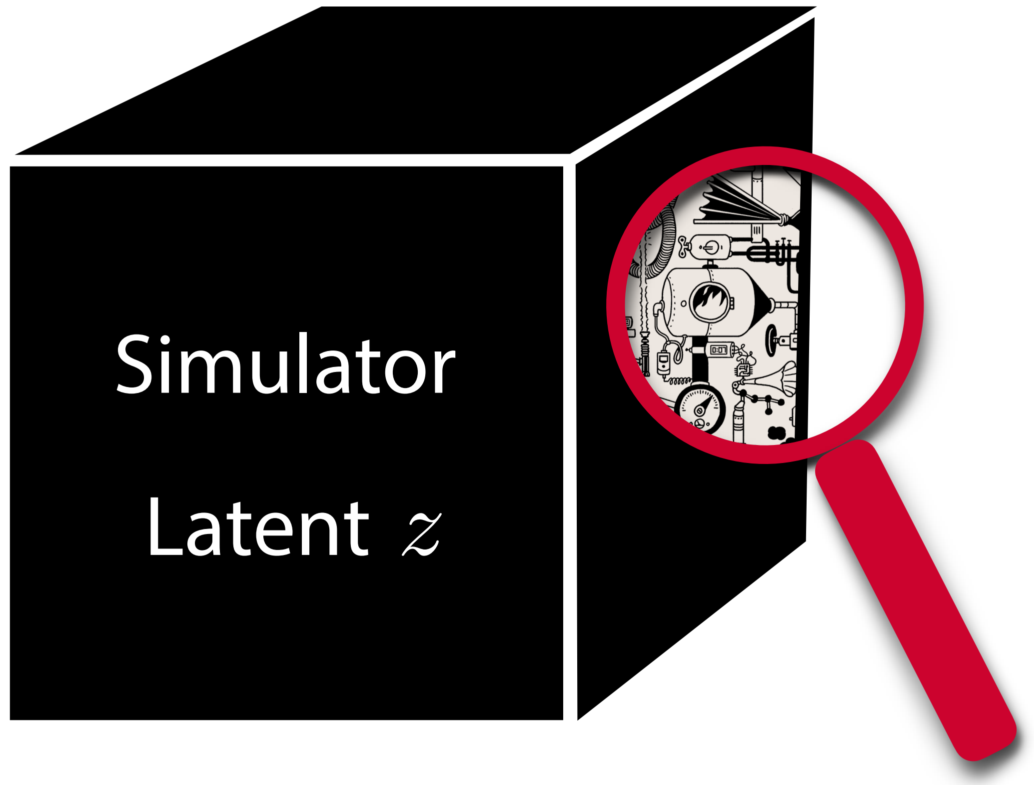
Run simulator and save data

Train NN classifier, interpret as likelihood ratio estimator

Amortized: cheap to repeat for new data

Idea 2: “mining gold”

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



We cannot compute $p(x|\theta) = \int \mathrm{d}z \, p(x, z|\theta)$,

but for each simulated event we can compute

- the joint likelihood ratio

$$r(x, z|\theta) = \frac{p(x, z|\theta)}{p_{\mathrm{ref}}(x, z)} \sim \frac{|\mathcal{M}|^2(z|\theta)}{|\mathcal{M}|^2_{\mathrm{ref}}(z)}$$

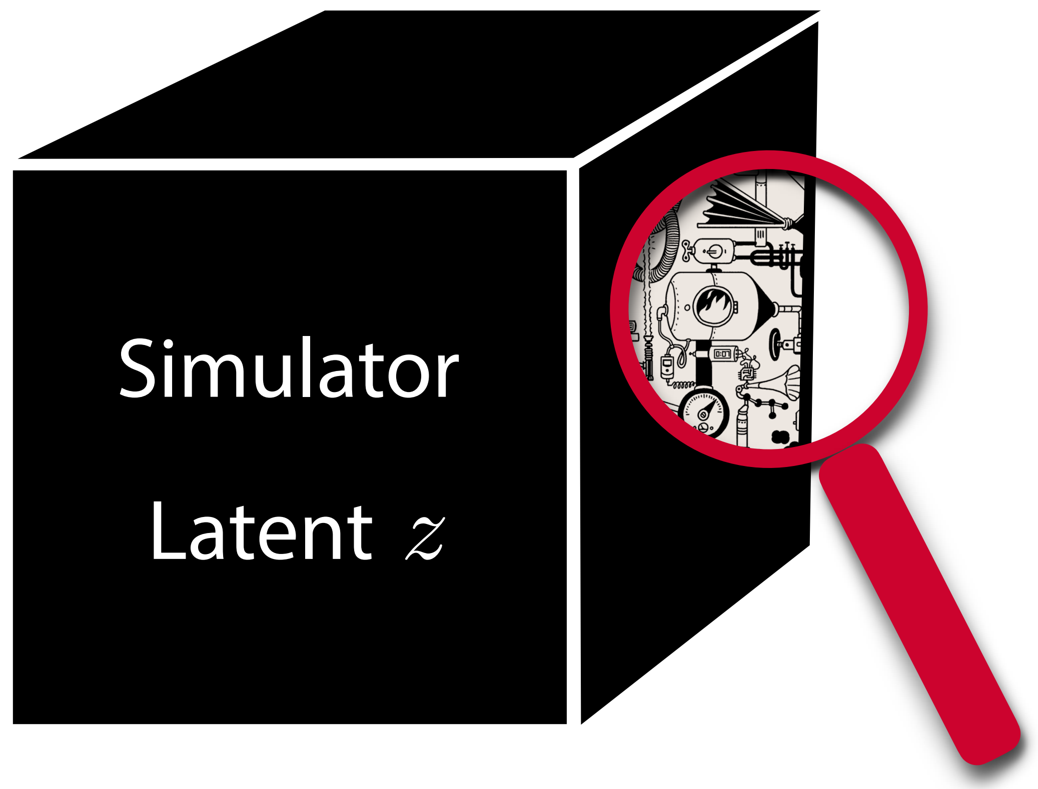
- the joint score

$$t(x, z|\theta) = \nabla_\theta \log p(x, z|\theta) \sim \frac{\nabla_\theta |\mathcal{M}|^2(z|\theta)}{|\mathcal{M}|^2(z|\theta)}$$

(Both depend on the truth-level four-momenta z)

Idea 2: “mining gold”

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



We cannot compute $p(x|\theta) = \int \mathrm{d}z\, p(x,z|\theta)$,

but for each simulated event we can compute

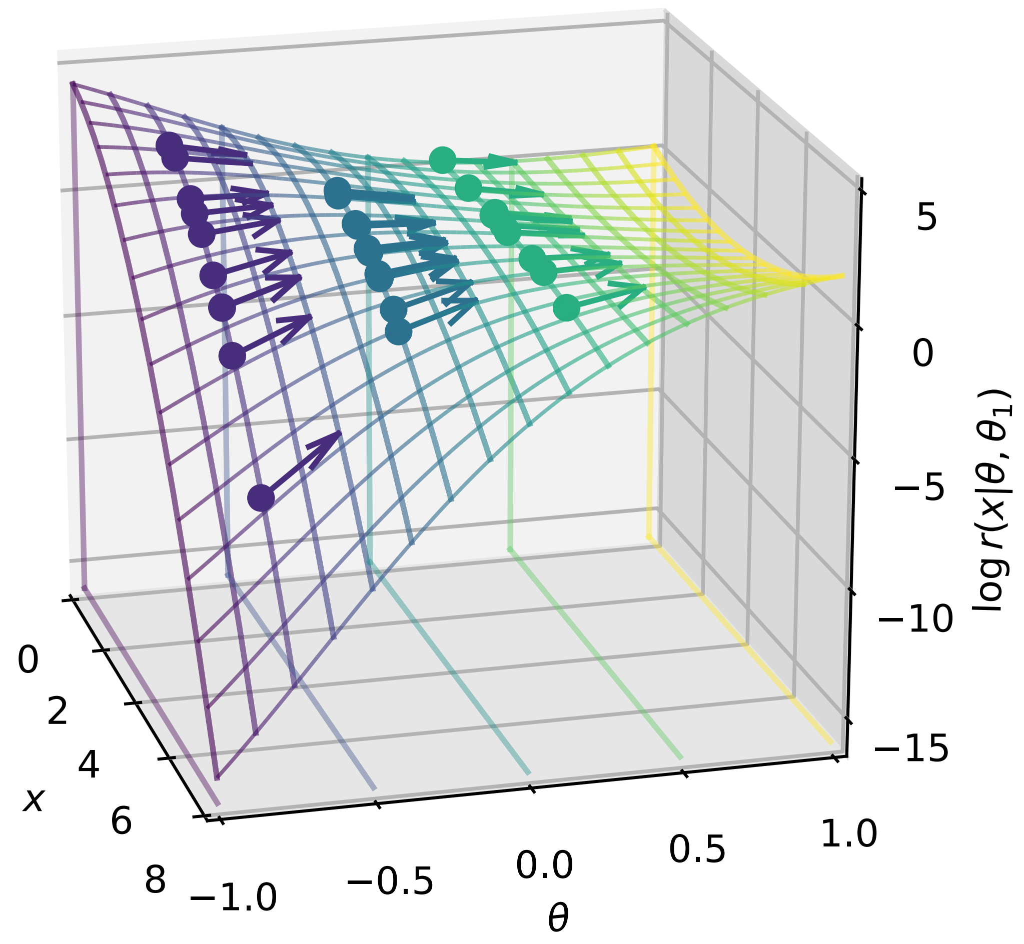
- the joint likelihood ratio

$$r(x,z|\theta) = \frac{p(x,z|\theta)}{p_{\text{ref}}(x,z)} \sim \frac{|\mathcal{M}|^2(z|\theta)}{|\mathcal{M}|^2_{\text{ref}}(z)}$$

- the joint score

$$t(x,z|\theta) = \nabla_\theta \log p(x,z|\theta) \sim \frac{\nabla_\theta |\mathcal{M}|^2(z|\theta)}{|\mathcal{M}|^2(z|\theta)}$$

(Both depend on the truth-level four-momenta z)



Why are they useful? One can show that

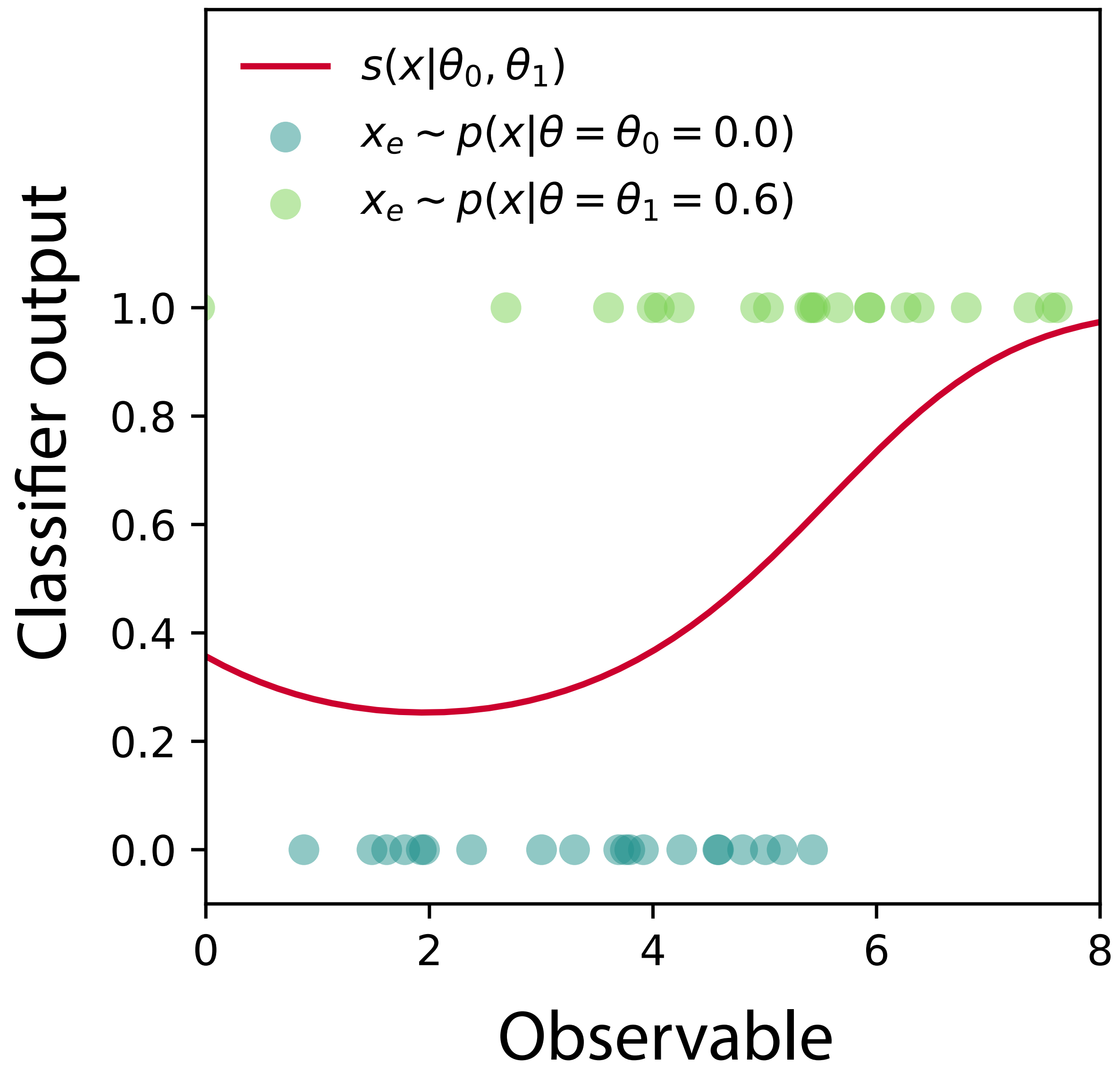
- the joint likelihood ratio is an unbiased estimator of the likelihood ratio
- the joint score provides unbiased gradient information

⇒ use them as labels in supervised NN training!

Mining gold adds information

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

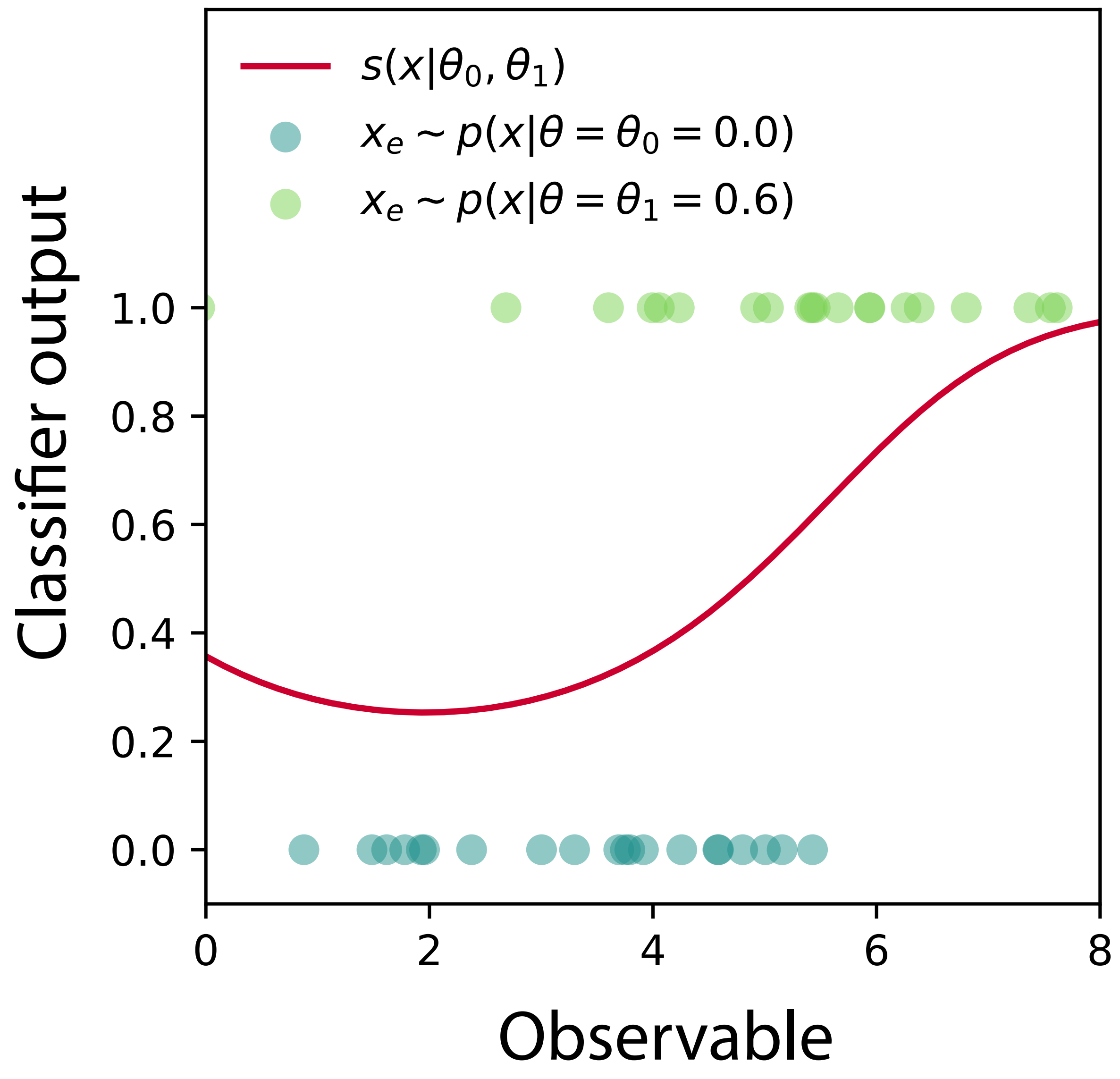
Likelihood ratio trick



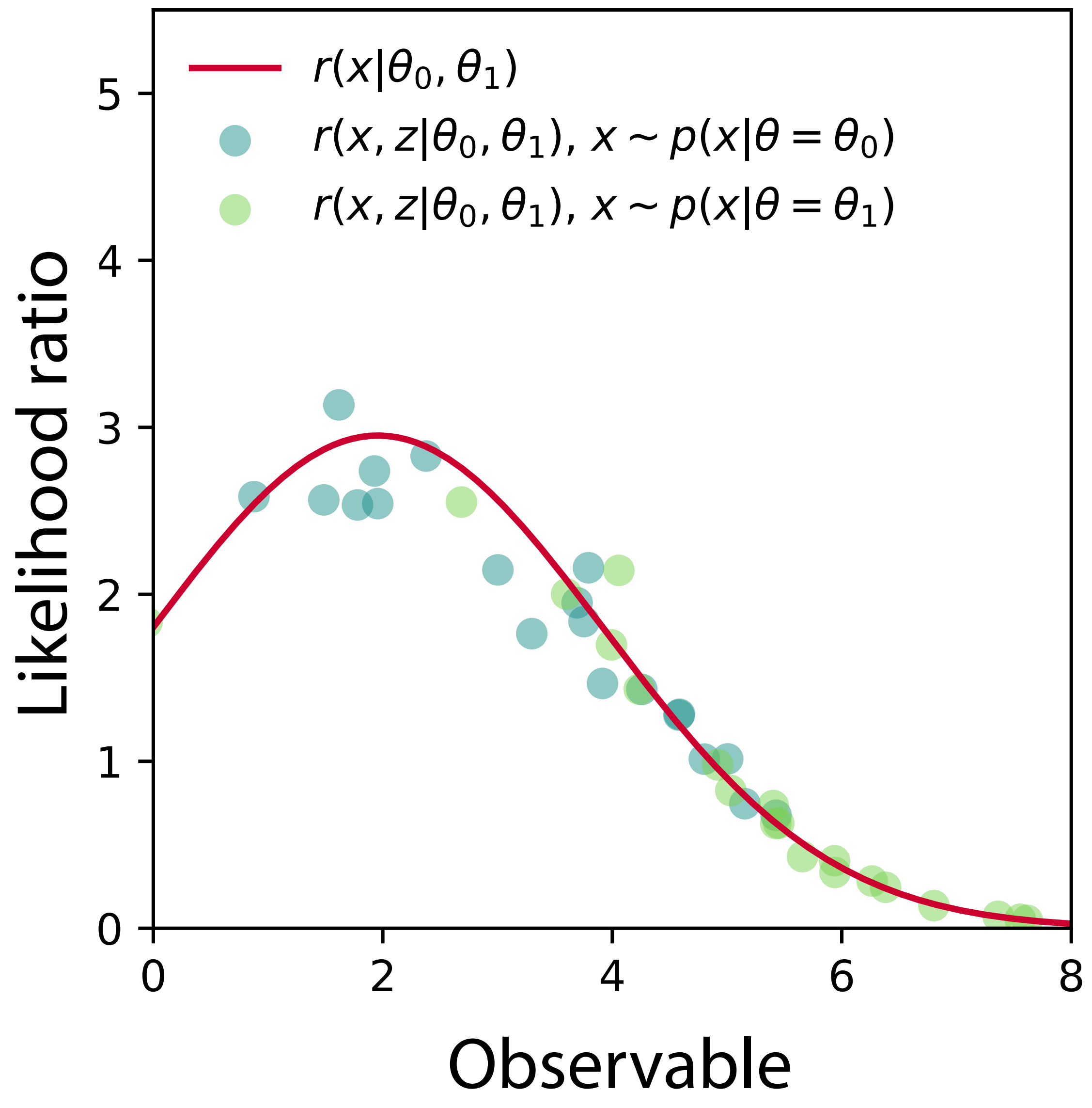
Mining gold adds information

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

Likelihood ratio trick



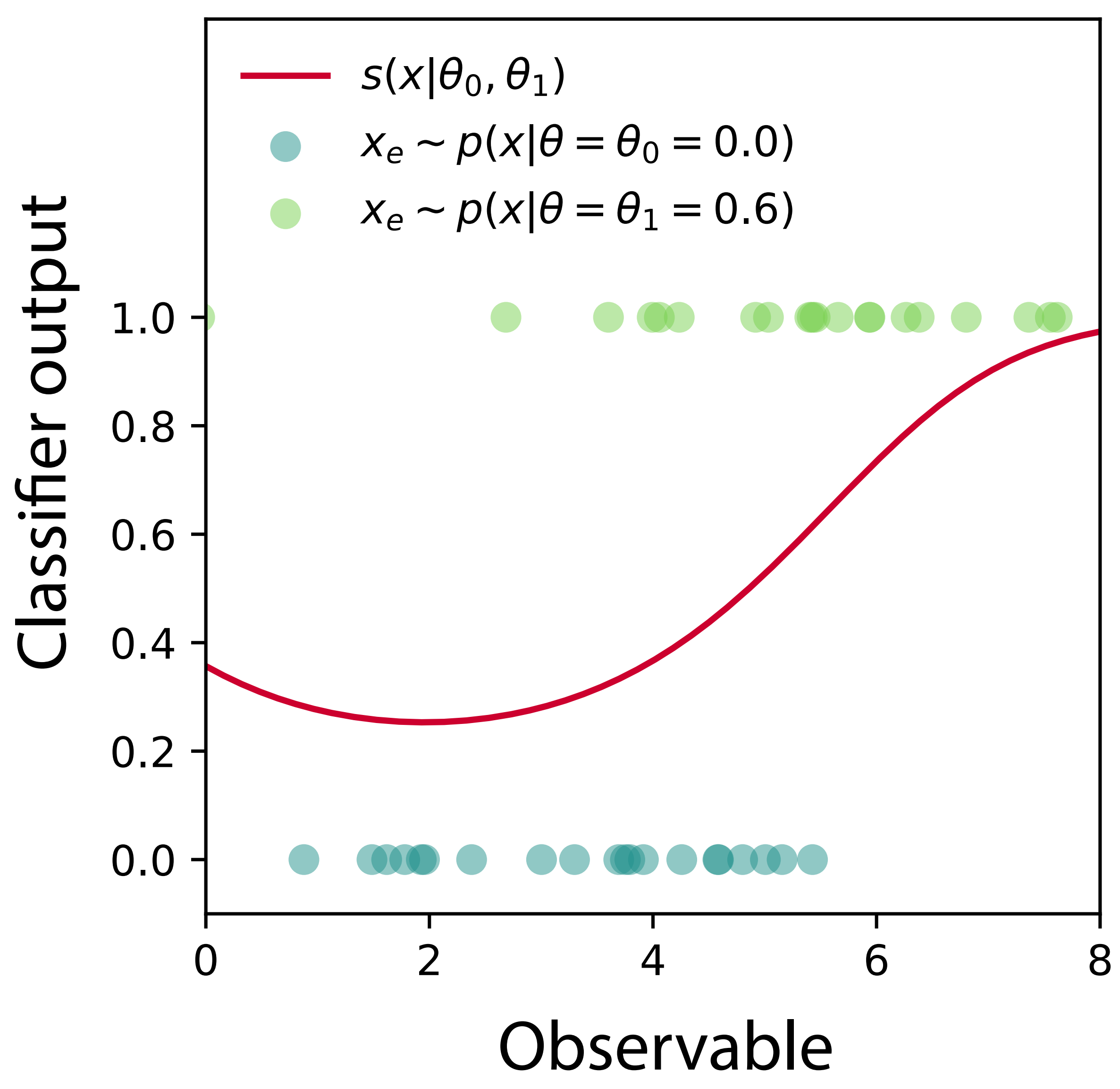
+ joint likelihood ratio



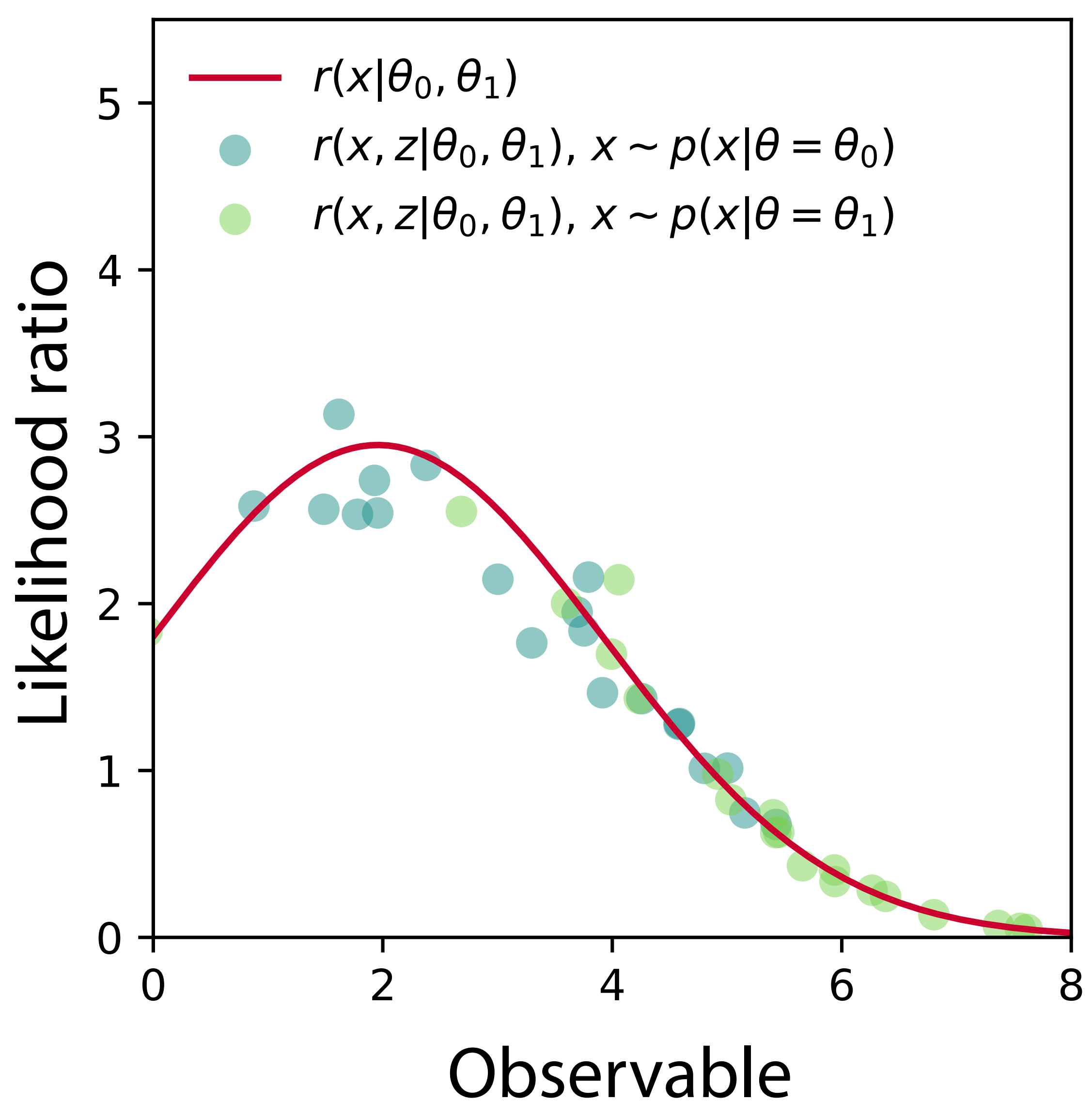
Mining gold adds information

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

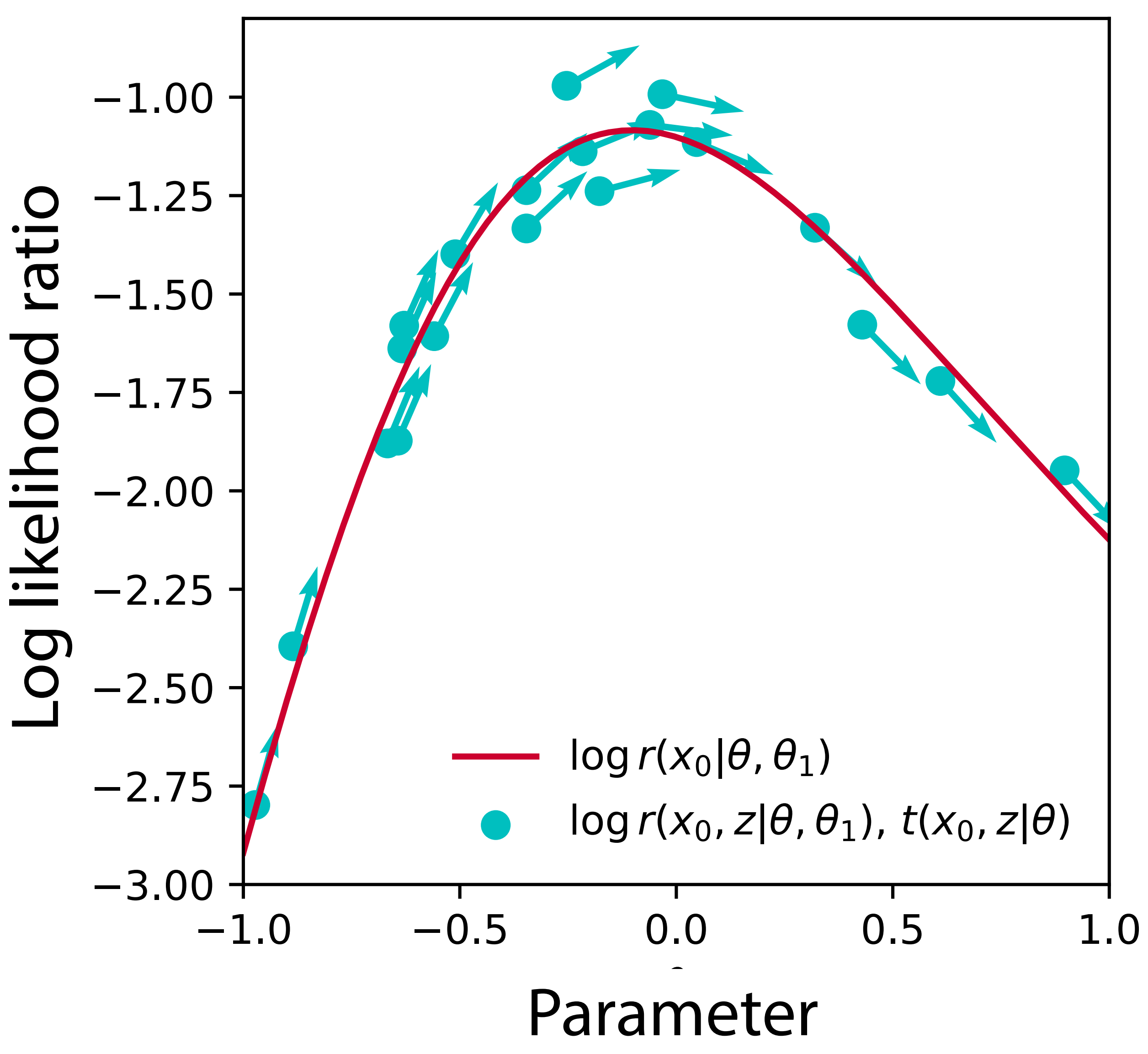
Likelihood ratio trick



+ joint likelihood ratio



+ joint score



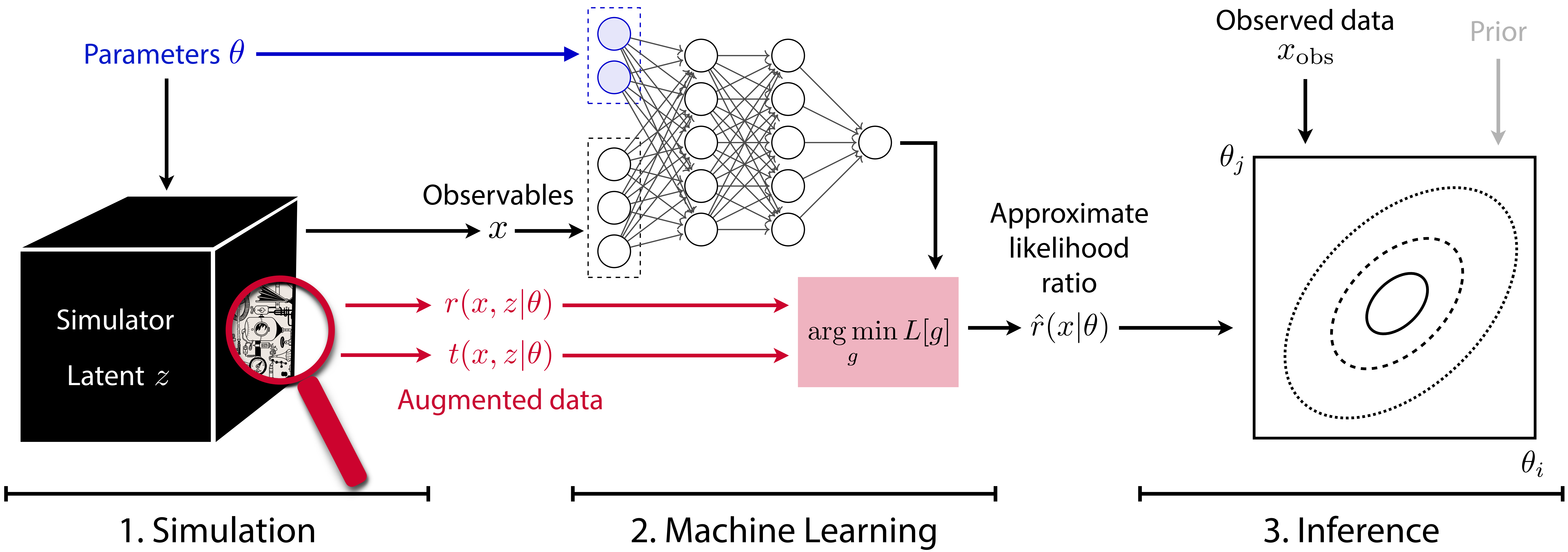
Mining gold adds information

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

Using more information = more sample-efficient inference

RASCAL

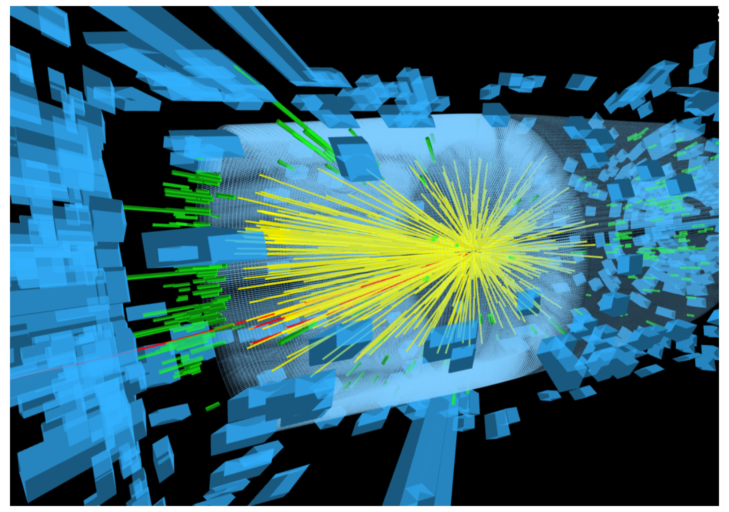
[JB, G. Louppe, J. Pavez, K. Cranmer
1805.12244, 1805.00013, 1805.00020]



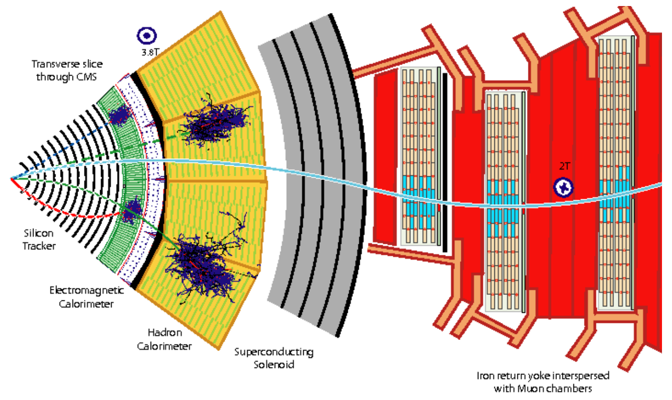
Extract joint likelihood ratio
and joint score from simulator

Augment training data &
use as labels in new loss functions
⇒ improve training efficiency

Sales pitch



Get all the information in high-dimensional data (no need for summary statistics)

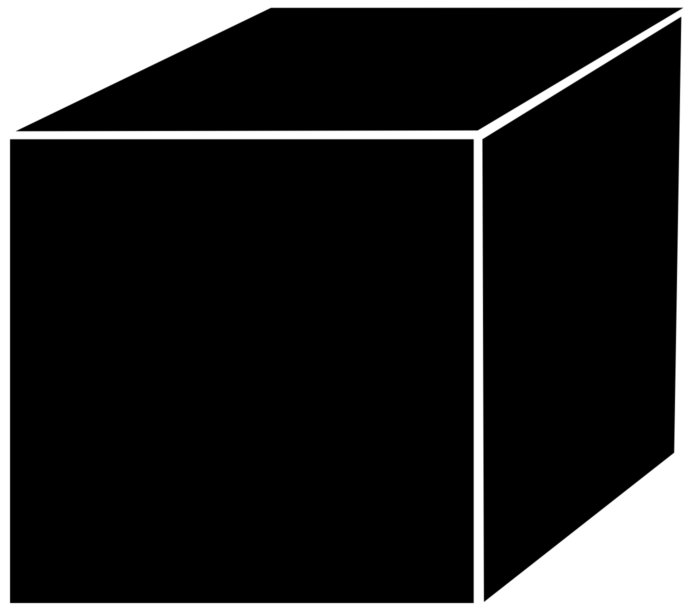


Use state-of-the-art shower and detector models (no transfer fns)

Evaluate events in microseconds (amortized inference)

Need less training data than black-box ML methods (using matrix-element information)

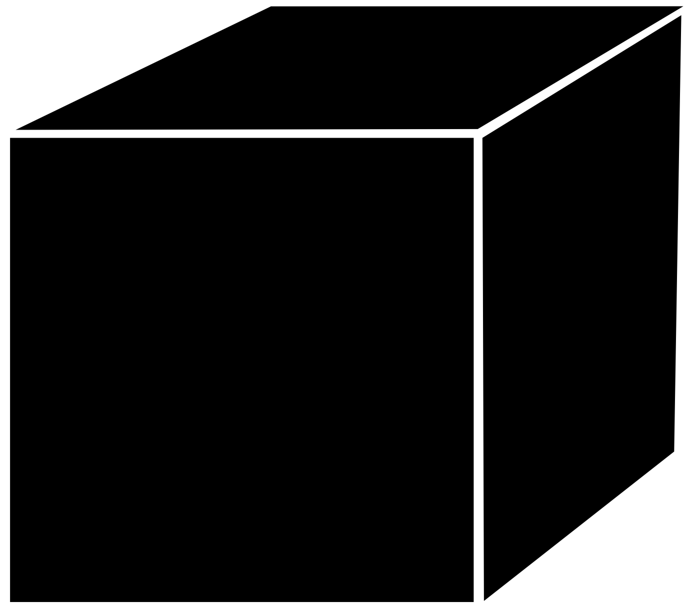
Systematics



Can you trust the simulator?

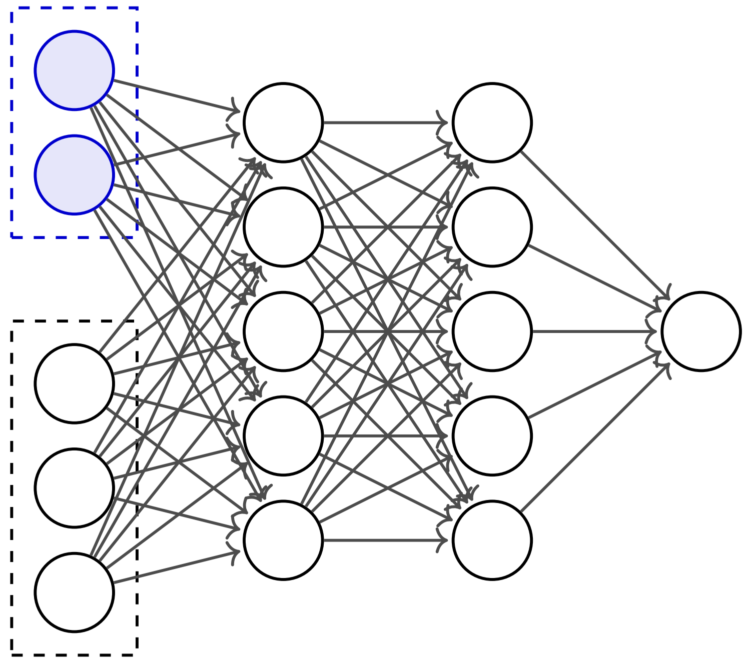
- Model uncertainties explicitly:
  nuisance parameters + profiling / marginalization
- Make analysis robust:
  ideas from domain adaptation, algorithmic fairness
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046; J. Alsing, B. Wandelt 1903.01473; P. de Castro, T. Dorigo 1806.04743]

Systematics



Can you trust the simulator?

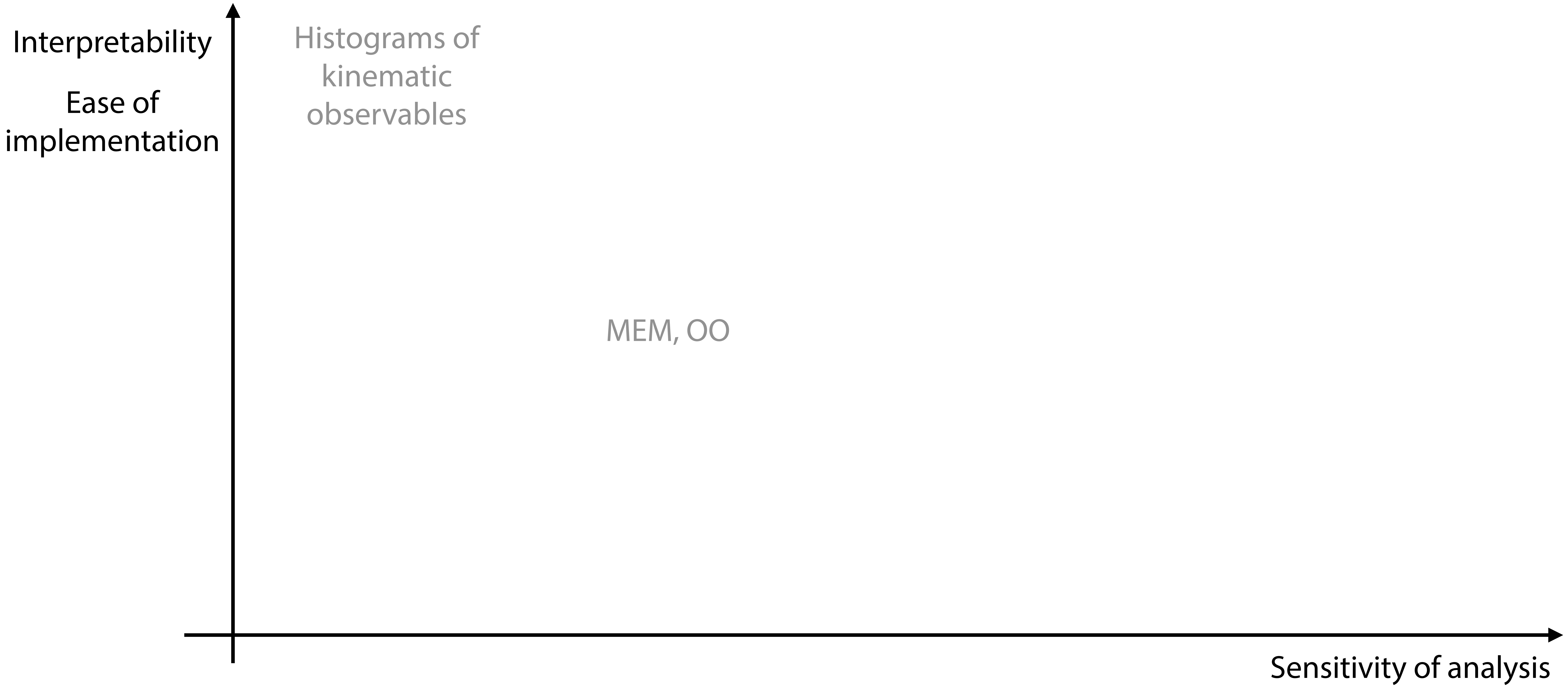
- Model uncertainties explicitly:
  nuisance parameters + profiling / marginalization
- Make analysis robust:
  ideas from domain adaptation, algorithmic fairness
  [G. Louppe, M. Kagan, K. Cranmer 1611.01046; J. Alsing, B. Wandelt 1903.01473; P. de Castro, T. Dorigo 1806.04743]



Can you trust the neural network?

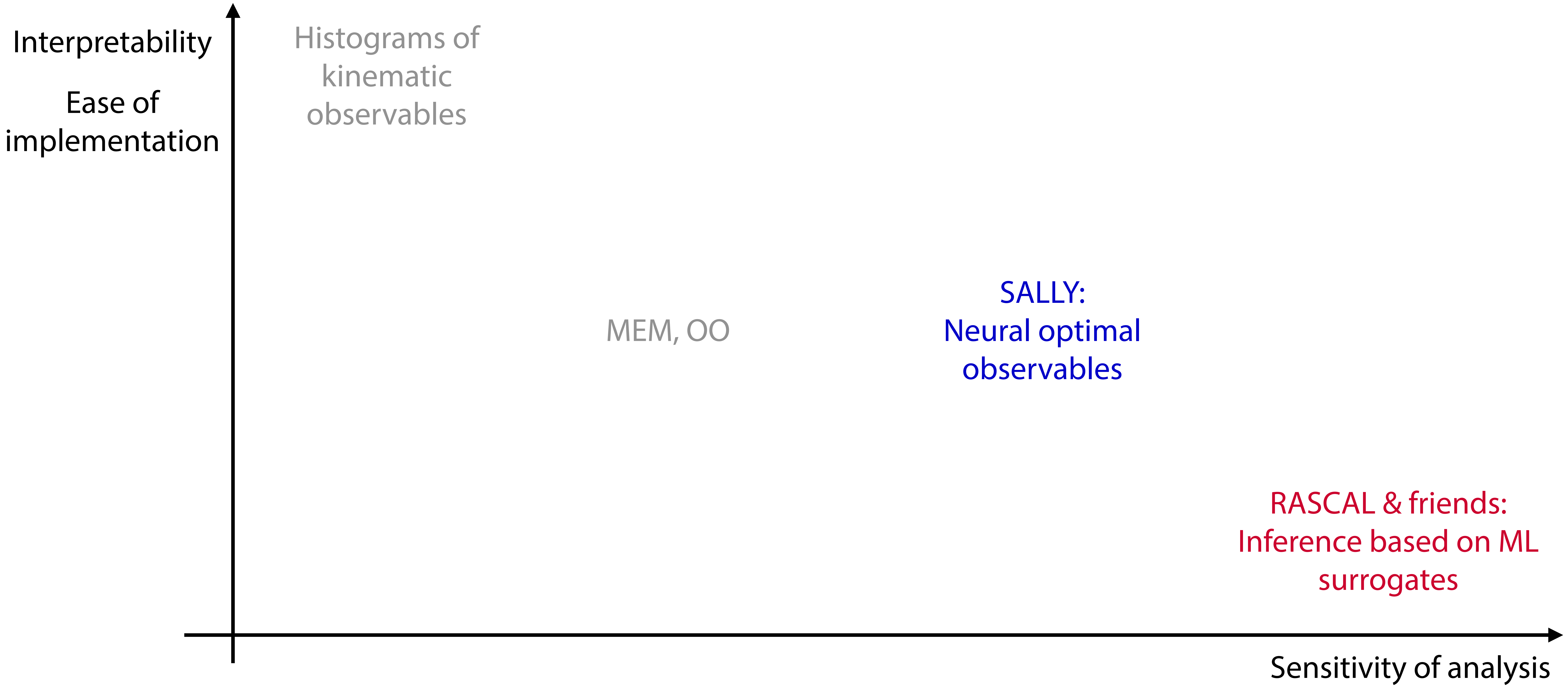
- Sanity checks: expectation values, "critic" tests
- Calibrate NN output
- Neyman construction with toys
  (badly trained network can lead to suboptimal limits, but not to wrong limits)
  [JB, G. Louppe, J. Pavez, K. Cranmer 1805.00020]

Simpler alternatives: ML-based optimal observables

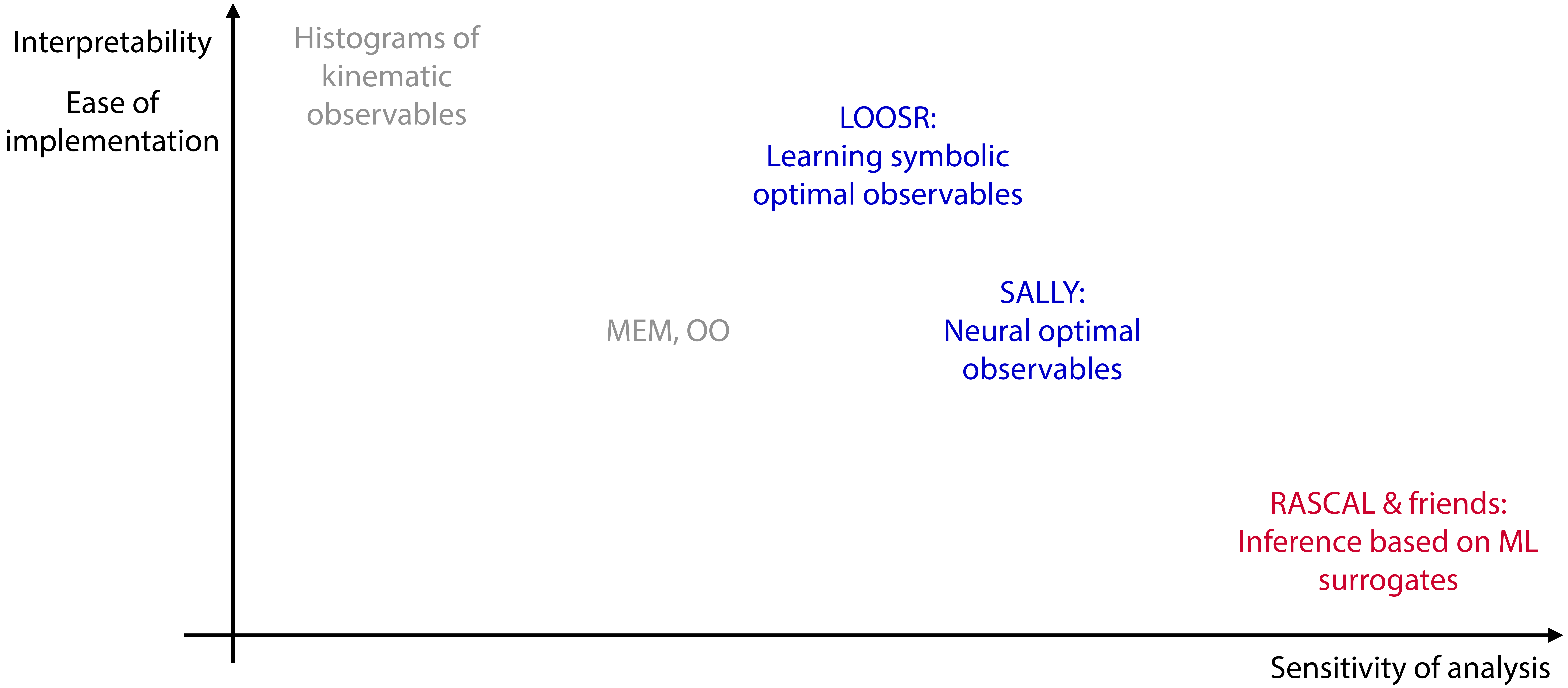


Simpler alternatives: ML-based optimal observables

Simpler alternatives: ML-based optimal observables



Simpler alternatives: ML-based optimal observables



Learning optimal observables

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244; N. Soybelmann, A. Butter, T. Plehn, JB 2109.10414]

- Rather than learning the likelihood ratio, can we just use ML to find better observables?
  - Holy grail: sufficient statistics (contain all information in data on the theory parameters)
  - Histogram analysis of sufficient statistics will give best possible sensitivity

Learning optimal observables

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244; N. Soybelmann, A. Butter, T. Plehn, JB 2109.10414]

- Rather than learning the likelihood ratio, can we just use ML to find better observables?
  - Holy grail: sufficient statistics (contain all information in data on the theory parameters)
  - Histogram analysis of sufficient statistics will give best possible sensitivity

- In the neighborhood of a reference point θ_{ref} (e.g. the SM), the sufficient statistics are just the score

$$t(x) = \nabla_\theta \log p(x|\theta)\Big|_{\theta_{\mathrm{ref}}}$$

Learning optimal observables

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013, 1805.00020, 1805.12244; N. Soybelmann, A. Butter, T. Plehn, JB 2109.10414]

- Rather than learning the likelihood ratio, can we just use ML to find better observables?
  - Holy grail: sufficient statistics (contain all information in data on the theory parameters)
  - Histogram analysis of sufficient statistics will give best possible sensitivity

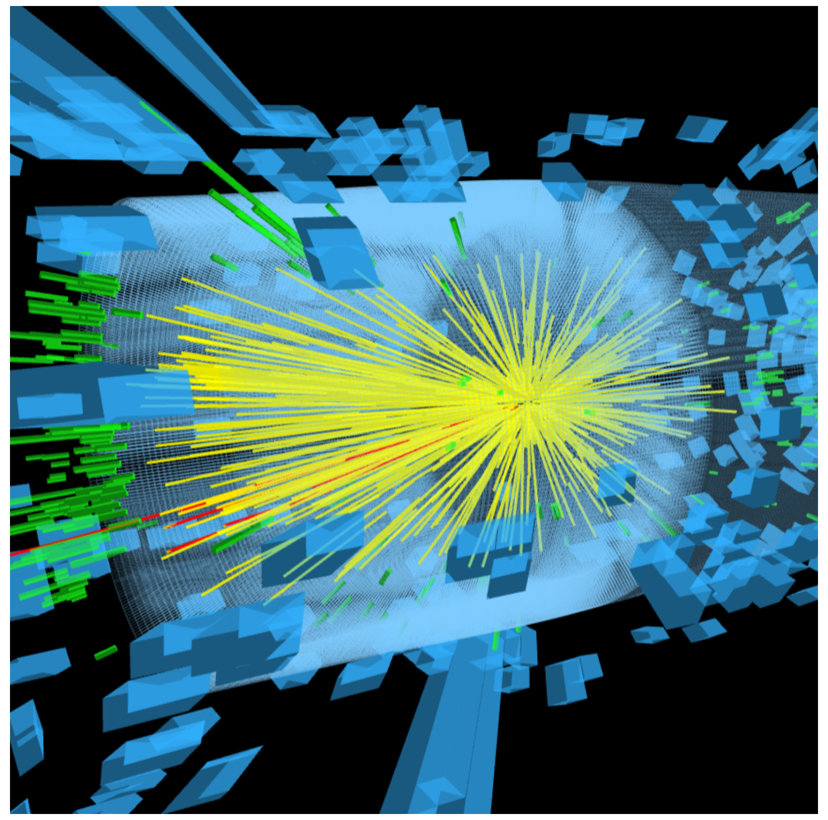
- In the neighborhood of a reference point θ_{ref} (e.g. the SM), the sufficient statistics are just the score

$$t(x) = \nabla_\theta \log p(x|\theta)\Big|_{\theta_{\text{ref}}}$$

- We can machine-learn the score from simulated data!

$$\hat{t}(x) = \underset{g(x)}{\arg\min}\ \mathbb{E}_{x,z} \Big| g(x) - \underbrace{t(x,z)}_{\sim \frac{\nabla_\theta \mathcal{M}^2(z|\theta_{\text{ref}})}{\mathcal{M}^2(z|\theta_{\text{ref}})}} \Big|^2$$

- SALLY: neural networks
  (straightforward to train, highly performant)
- LOOSR: symbolic expressions like "$p_{T1}\, p_{T2}\, \sin\Delta\phi_{jj}$"
  (physicist-interpretable, easy to implement)

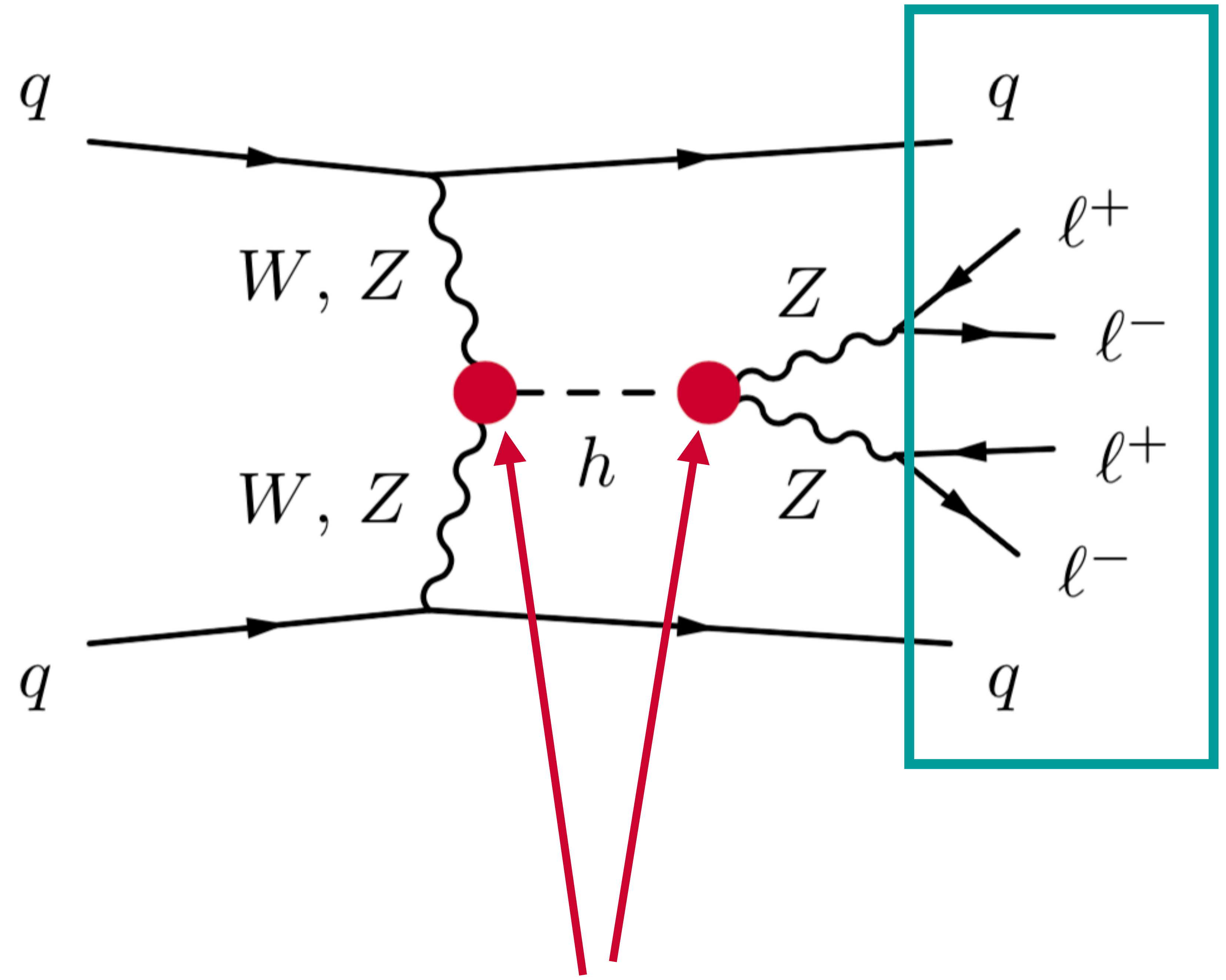


4. Examples

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

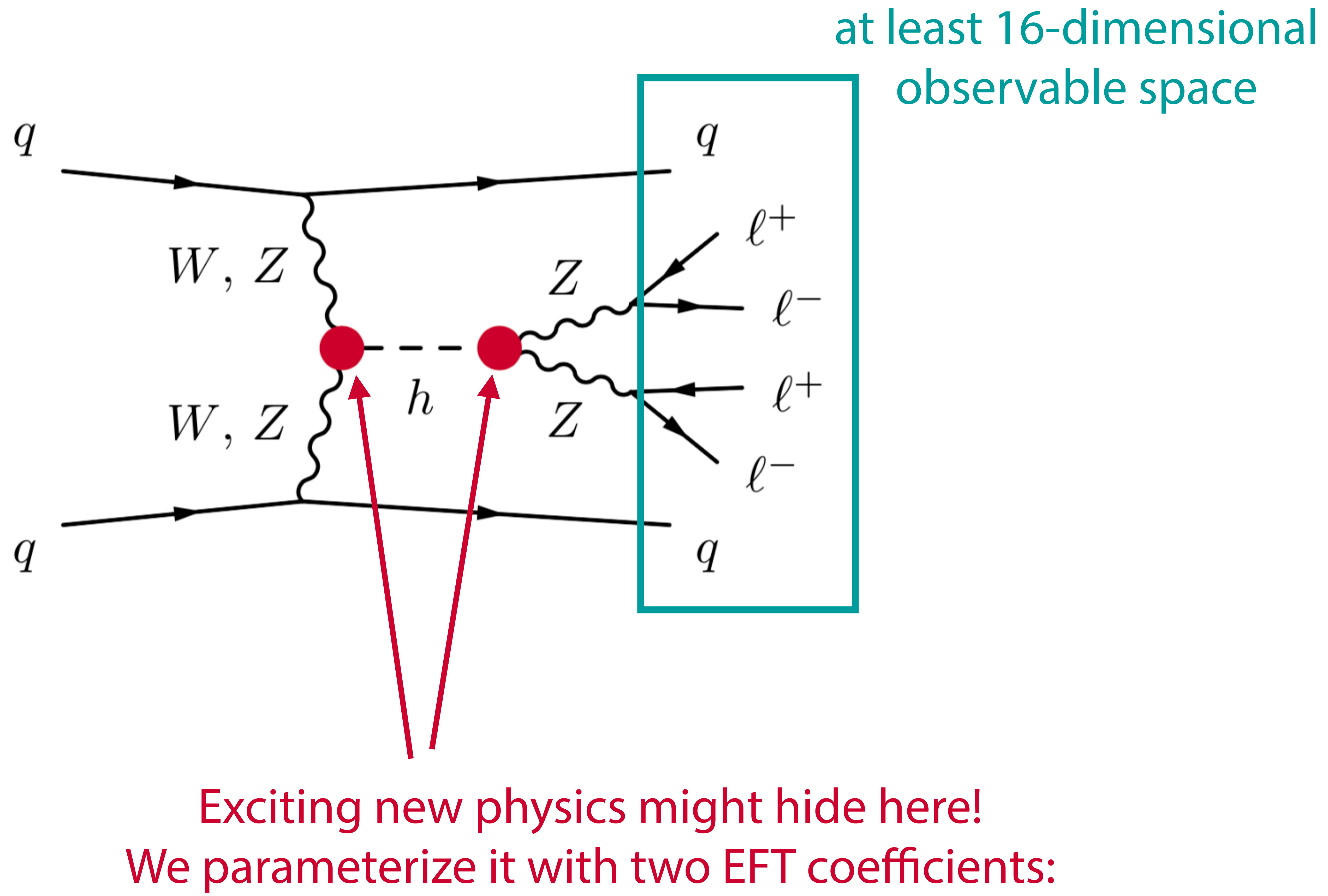


Exciting new physics might hide here!
We parameterize it with two EFT coefficients:

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

Proof of concept: Higgs production in weak boson fusion

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



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

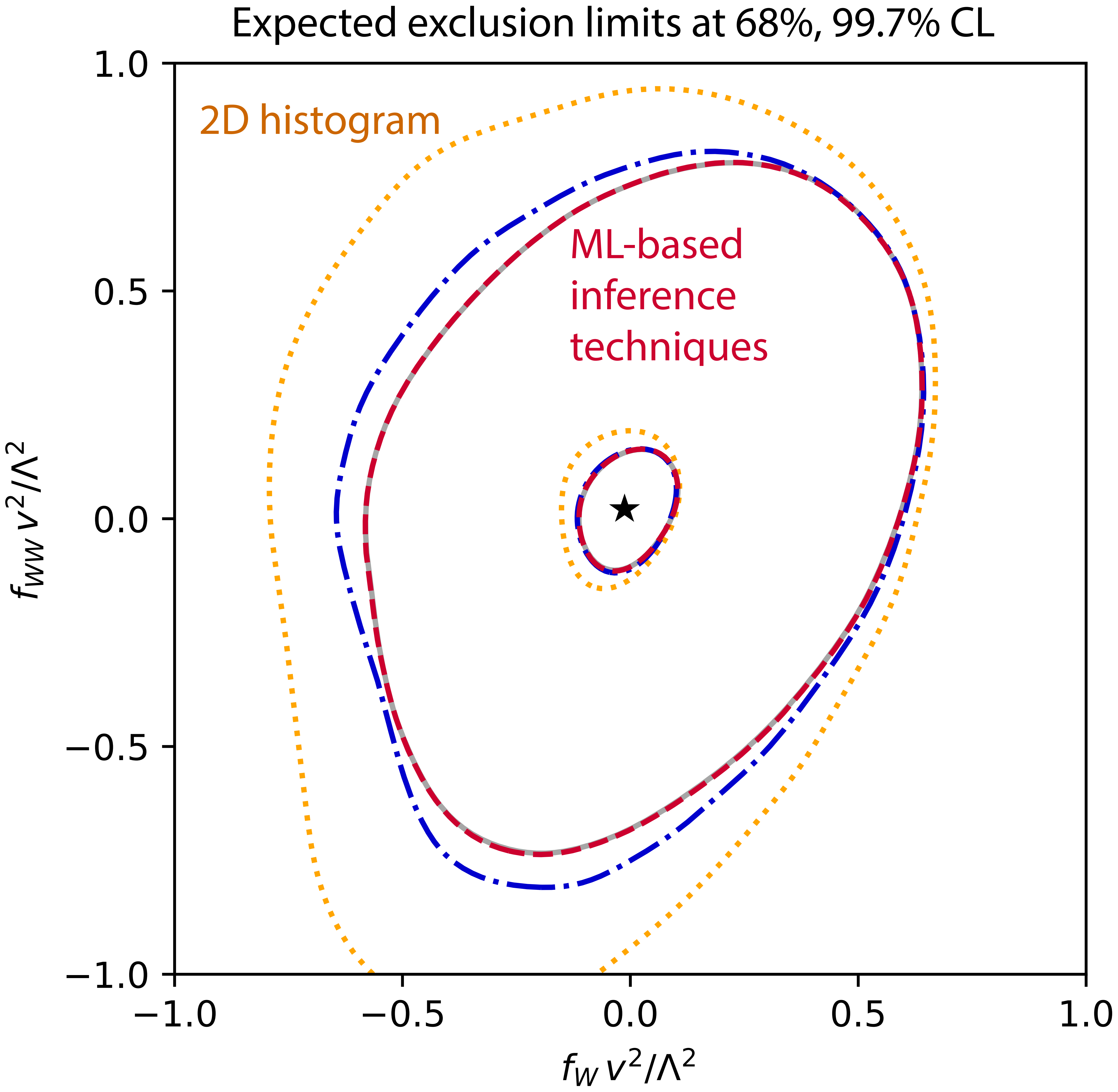
Goal: constrain the two EFT parameters

- 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

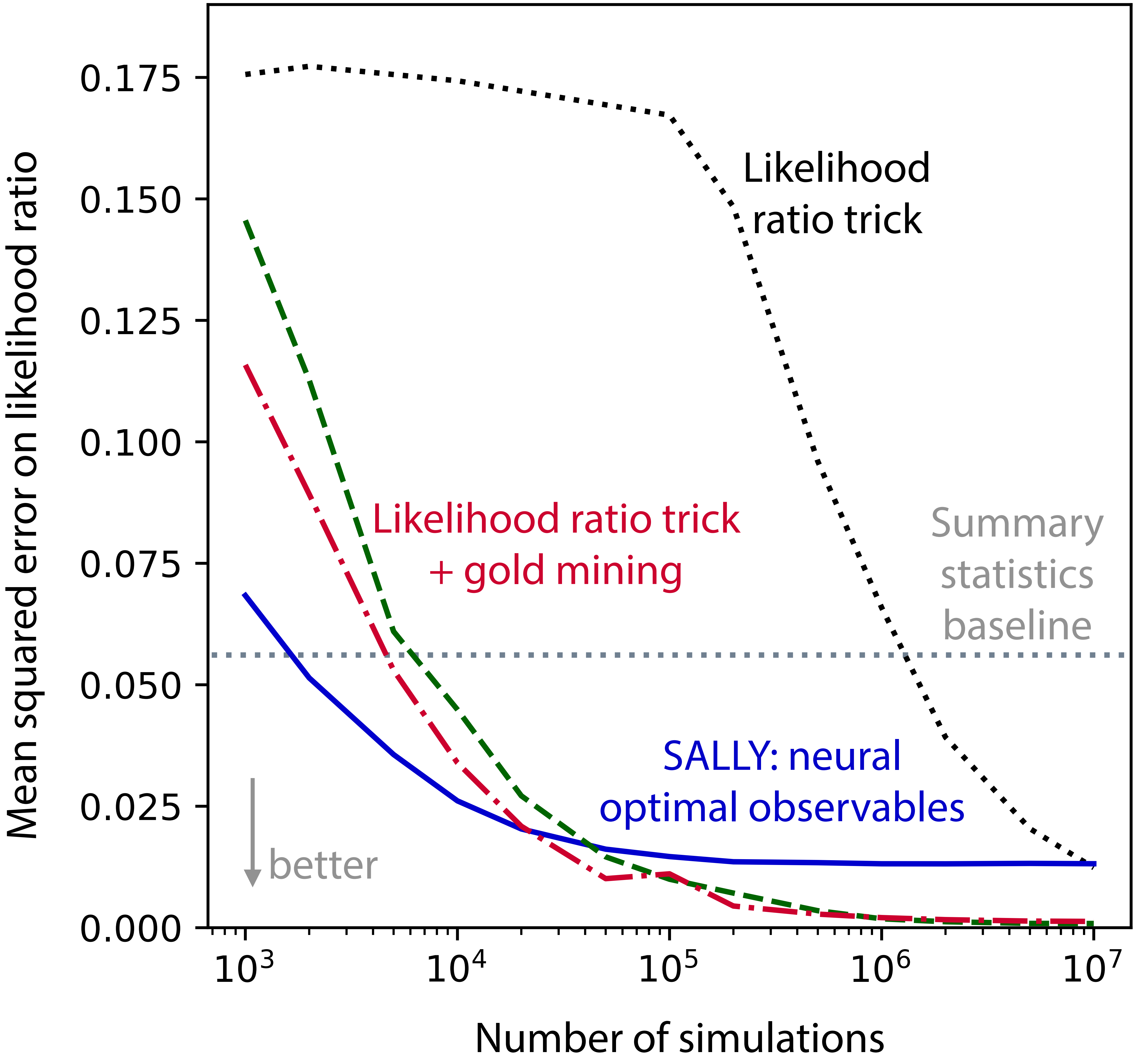
Stronger limits…



In some regions of parameter space, the ML-based inference techniques improve the sensitivity as much as taking 90% more data would!

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013; 1805.00020; M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez 1808.00973]

…with less training data



With enough training data, the ML algorithms get the likelihood function right.

Using more information from the simulator improves sample efficiency substantially.

[JB, K. Cranmer, G. Louppe, J. Pavez 1805.00013; 1805.00020; M. Stoye, JB, K. Cranmer, G. Louppe, J. Pavez 1808.00973]

Constraining operators in ttH effectively

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

- Pheno-level analysis of

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

  with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

$$\mathcal{O}_u = -\frac{1}{v^2}(H^\dagger H)(H^\dagger \bar{Q}_L)u_R\,,\ \mathcal{O}_G = \frac{g_s^2}{m_W^2}(H^\dagger H)G^a_{\mu\nu}G_a^{\mu\nu}\,,$$
$$\mathcal{O}_{uG} = -\frac{4g_s}{m_W^2}y_u(H^\dagger \bar{Q}_L)\gamma^{\mu\nu}T_a u_R G^a_{\mu\nu}$$

Constraining operators in ttH effectively

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

- Pheno-level analysis of

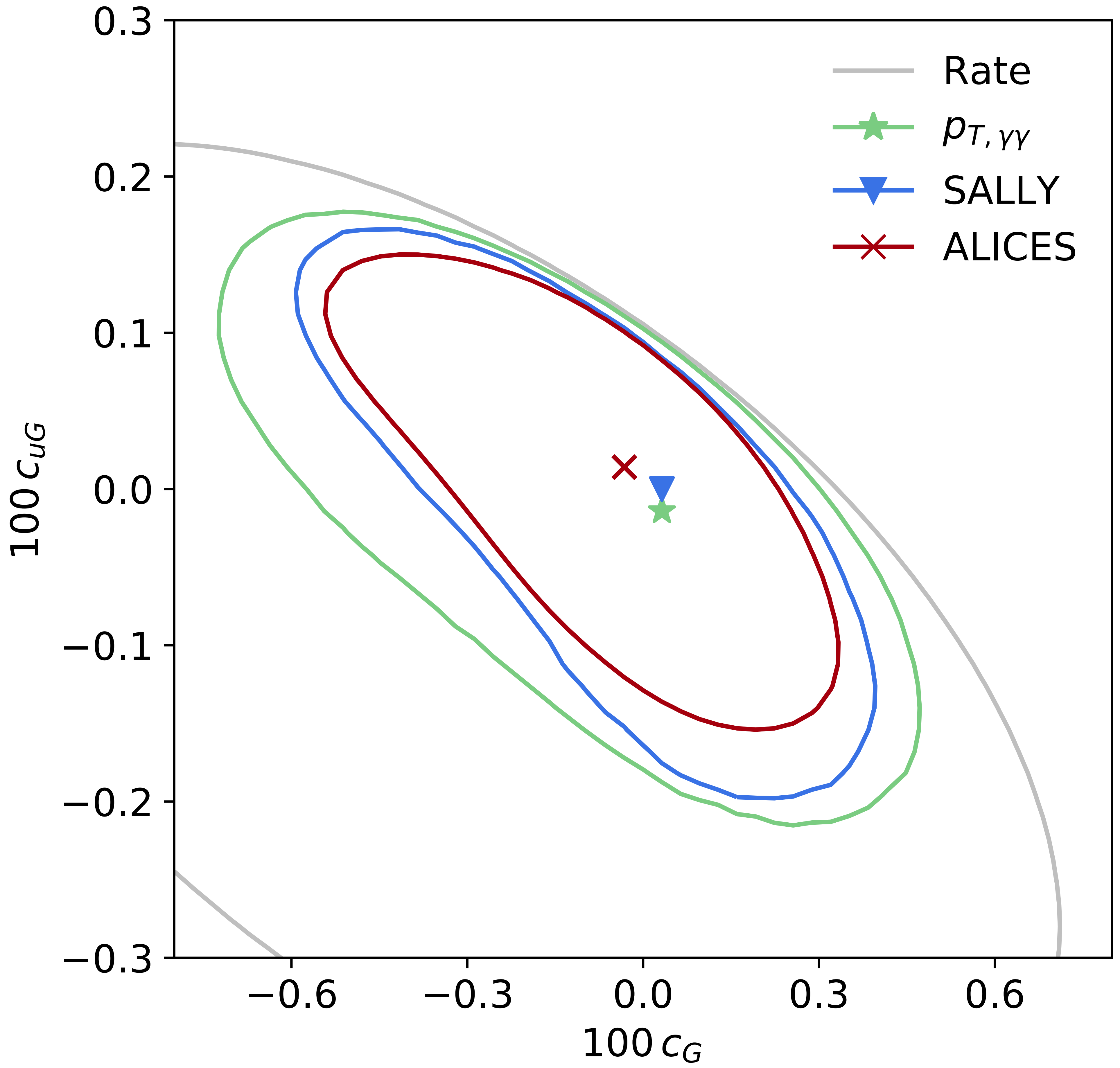
$$pp \to t\bar{t}\,h \to (b\ell^+)\,(\bar{b}\ell^-)\,(\gamma\gamma)\,E_T^{\text{miss}}$$

with MadGraph + Pythia + Delphes

- Inference on three EFT operators:

$$\mathcal{O}_u = -\frac{1}{v^2}(H^\dagger H)(H^\dagger \bar{Q}_L)u_R\,,\;\; \mathcal{O}_G = \frac{g_s^2}{m_W^2}(H^\dagger H)G^a_{\mu\nu}G_a^{\mu\nu}\,,$$
$$\mathcal{O}_{uG} = -\frac{4g_s}{m_W^2}y_u(H^\dagger \bar{Q}_L)\gamma^{\mu\nu}T_a u_R G^a_{\mu\nu}$$

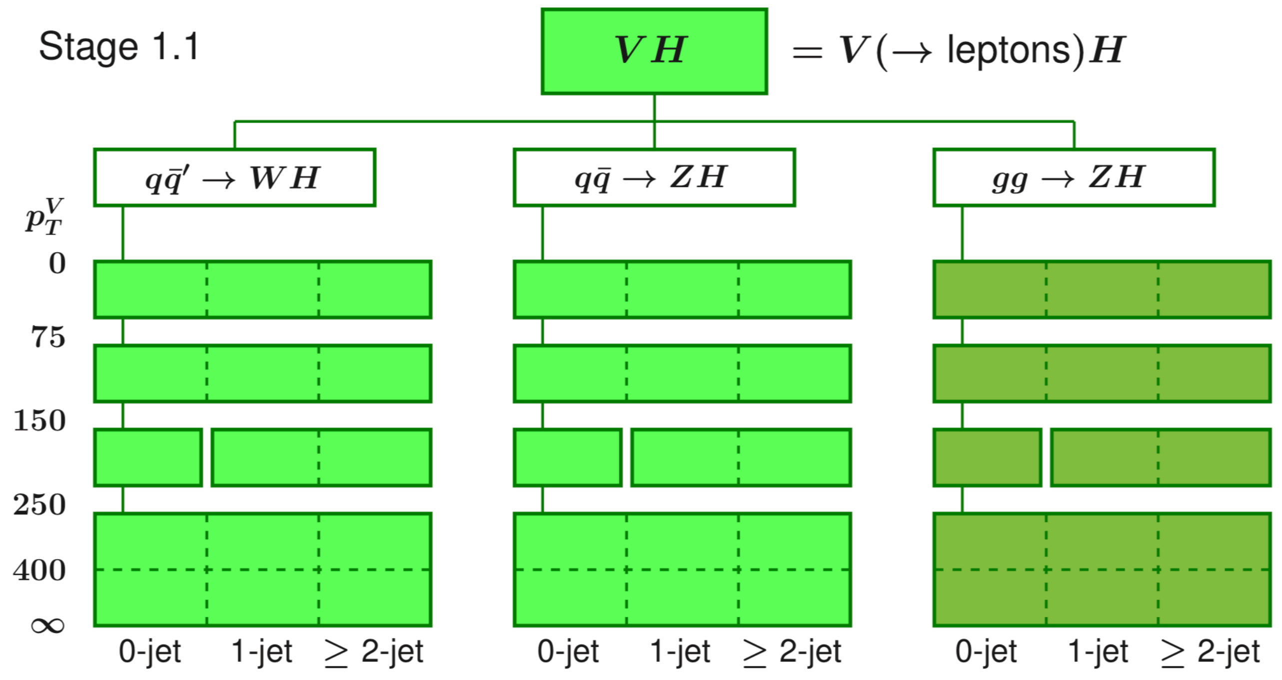
- New inference techniques improve expected HL-LHC limits compared to histogram baseline:



Benchmarking STXS in WH

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

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



- Let's check! How much information on

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

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

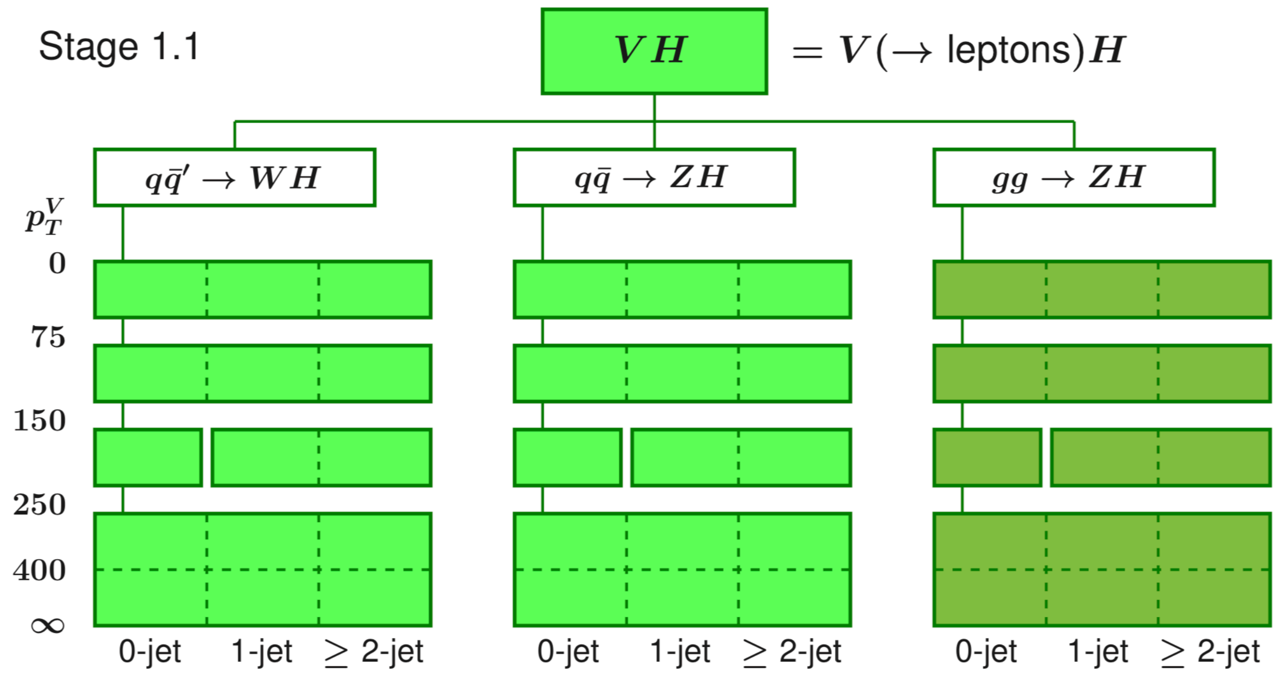
$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu\phi)(\overline{Q}_L\sigma^a\gamma^\mu Q_L)\ ,$$

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

Benchmarking STXS in WH

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

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

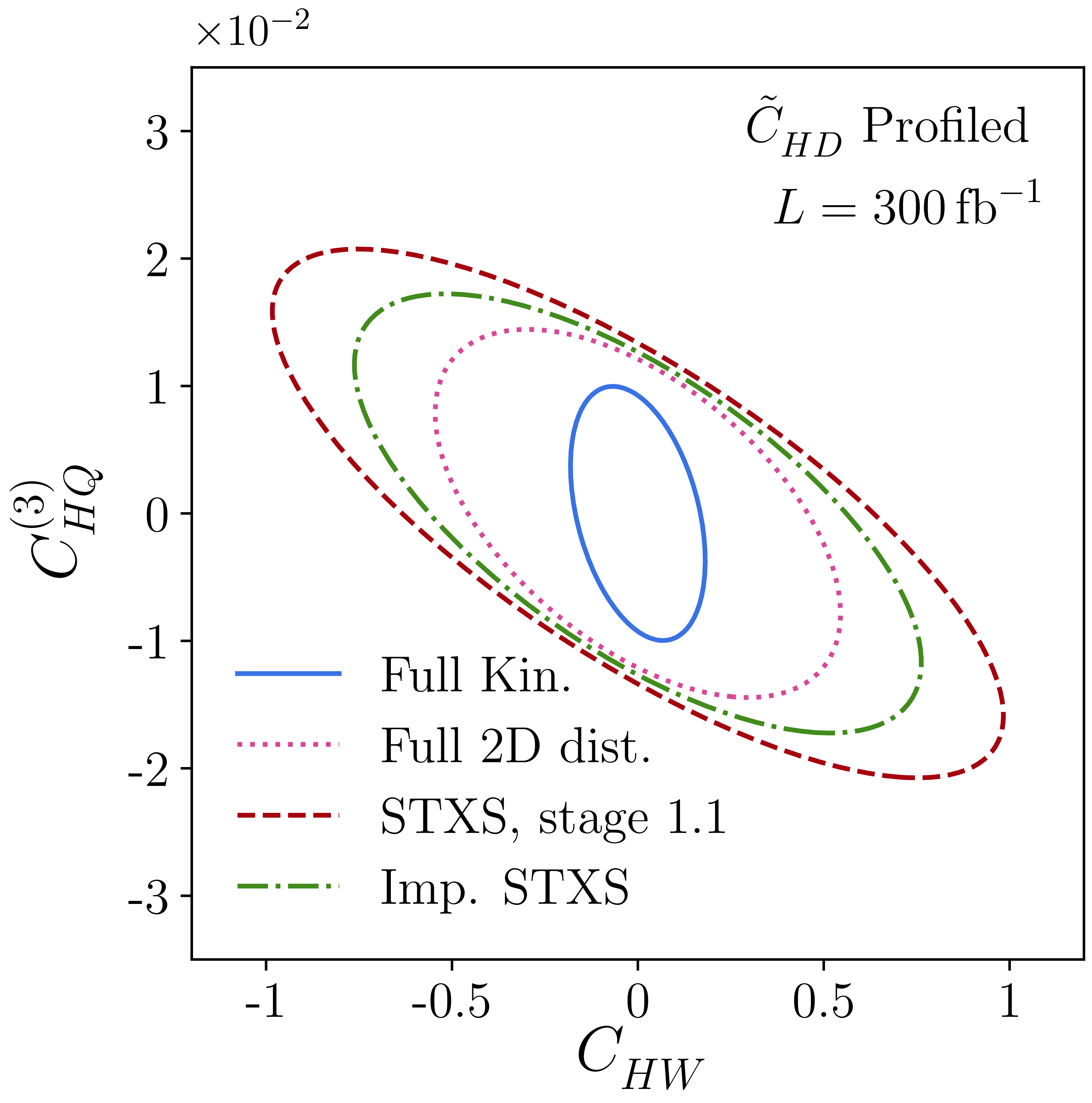


- Let's check! How much information on

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

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

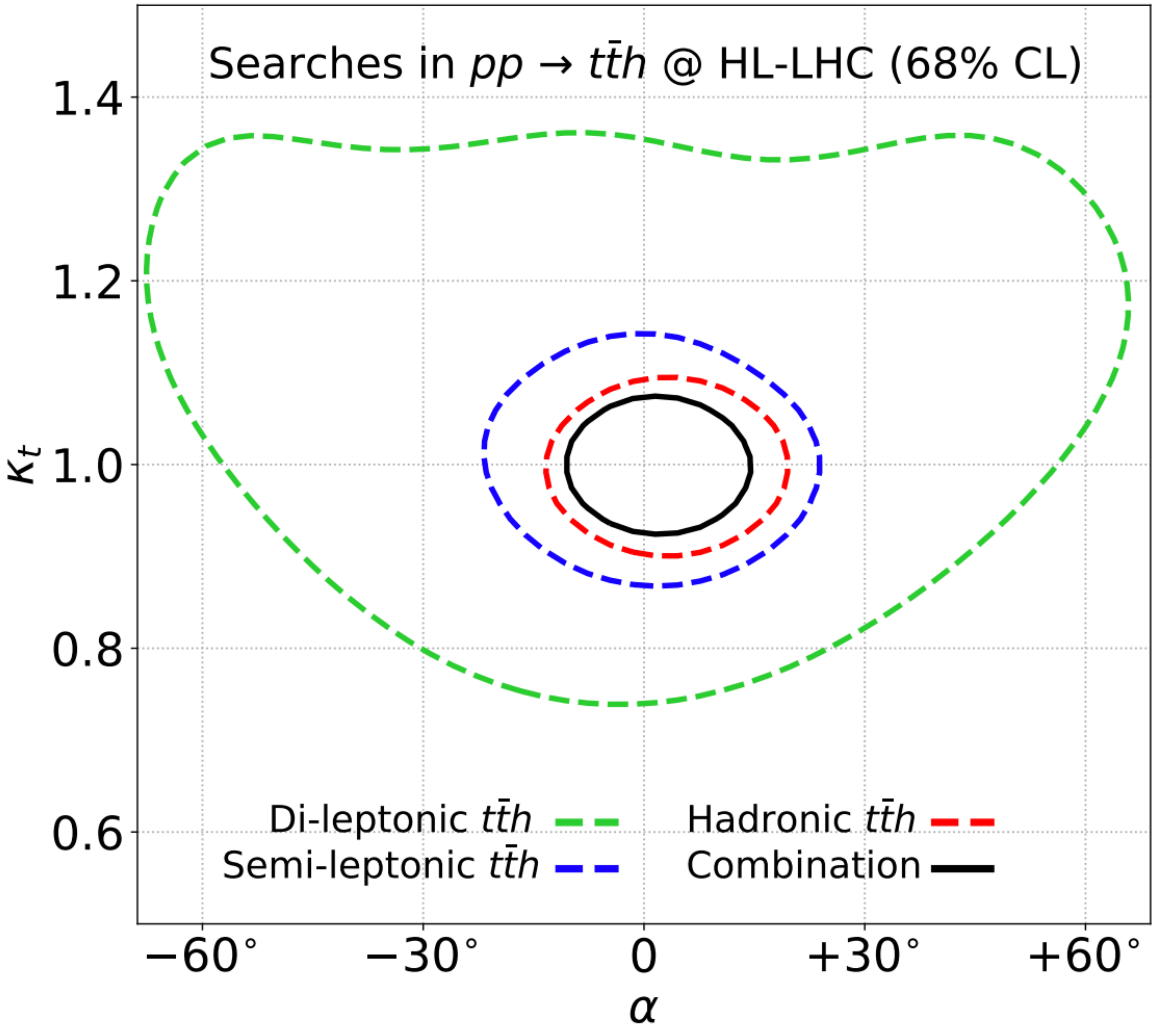
- Results: STXS are indeed sensitive to operators, adding a few more bins improve them, but a multivariate analysis is still stronger



Higgs-top CP phases

[R. Barman, D. Gonçalves, F. Kling 2110.07635;
H. Bahl, S. Brass 2110.10177]

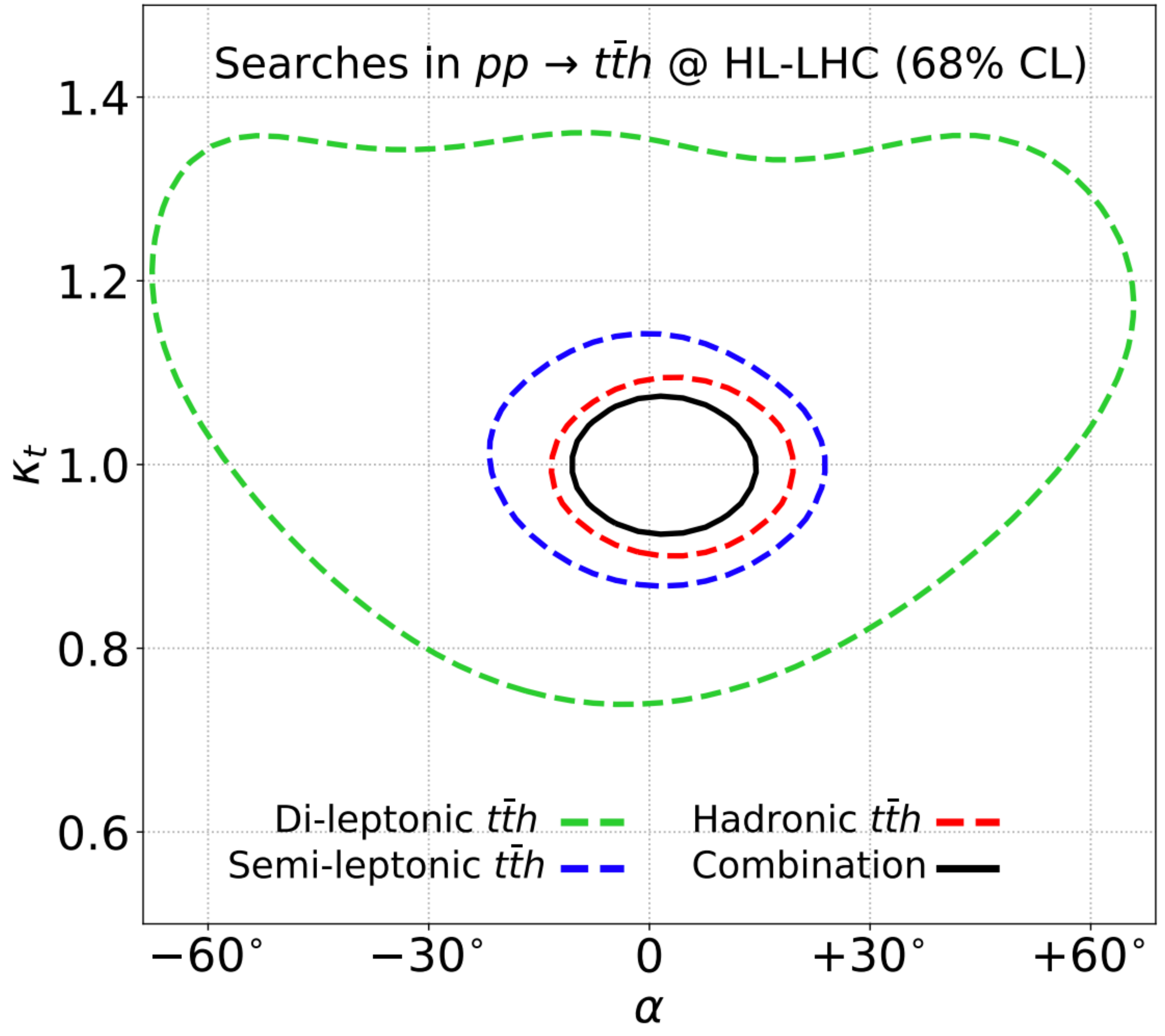
- These methods also allow us to measure the CP structure of Higgs-top coupling in $t\bar{t}H$ production ($H \to \gamma\gamma$)



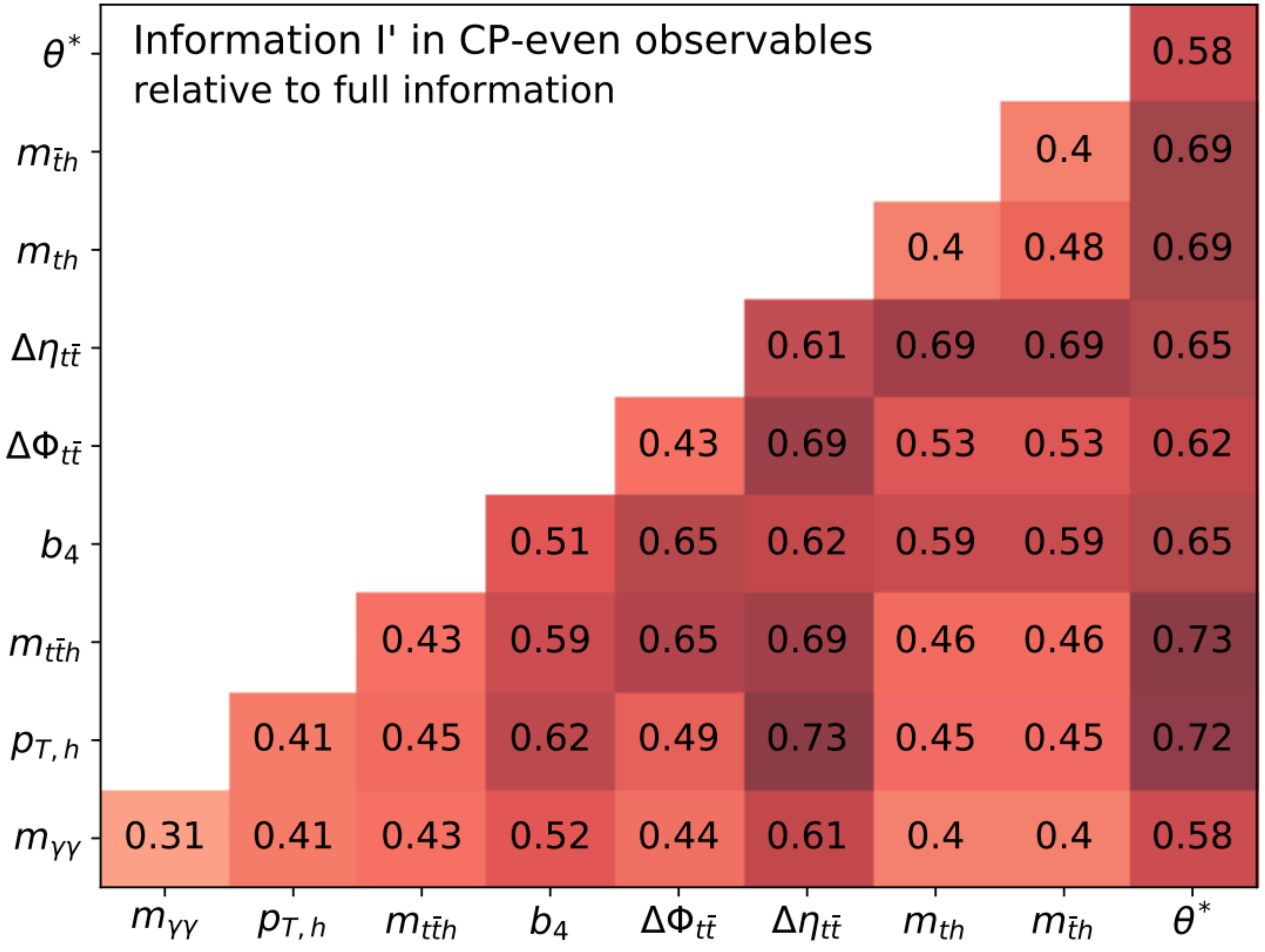
Higgs-top CP phases

[R. Barman, D. Gonçalves, F. Kling 2110.07635;
H. Bahl, S. Brass 2110.10177]

- These methods also allow us to measure the CP structure of Higgs-top coupling in $t\bar{t}H$ production ($H \to \gamma\gamma$)



- ML-based analysis 50% more sensitive than any considered 2D histogram analysis



	$m_{\gamma\gamma}$	$p_{T,h}$	$m_{t\bar{t}h}$	b_4	$\Delta\Phi_{t\bar{t}}$	$\Delta\eta_{t\bar{t}}$	m_{th}	$m_{\bar{t}h}$	θ^*
θ^*									0.58
$m_{\bar{t}h}$								0.4	0.69
m_{th}							0.4	0.48	0.69
$\Delta\eta_{t\bar{t}}$						0.61	0.69	0.69	0.65
$\Delta\Phi_{t\bar{t}}$					0.43	0.69	0.53	0.53	0.62
b_4				0.51	0.65	0.62	0.59	0.59	0.65
$m_{t\bar{t}h}$			0.43	0.59	0.65	0.69	0.46	0.46	0.73
$p_{T,h}$		0.41	0.45	0.62	0.49	0.73	0.45	0.45	0.72
$m_{\gamma\gamma}$	0.31	0.41	0.43	0.52	0.44	0.61	0.4	0.4	0.58

Automation

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

Our open-source Python package MadMiner makes it straightforward to apply these ML-based inference techniques

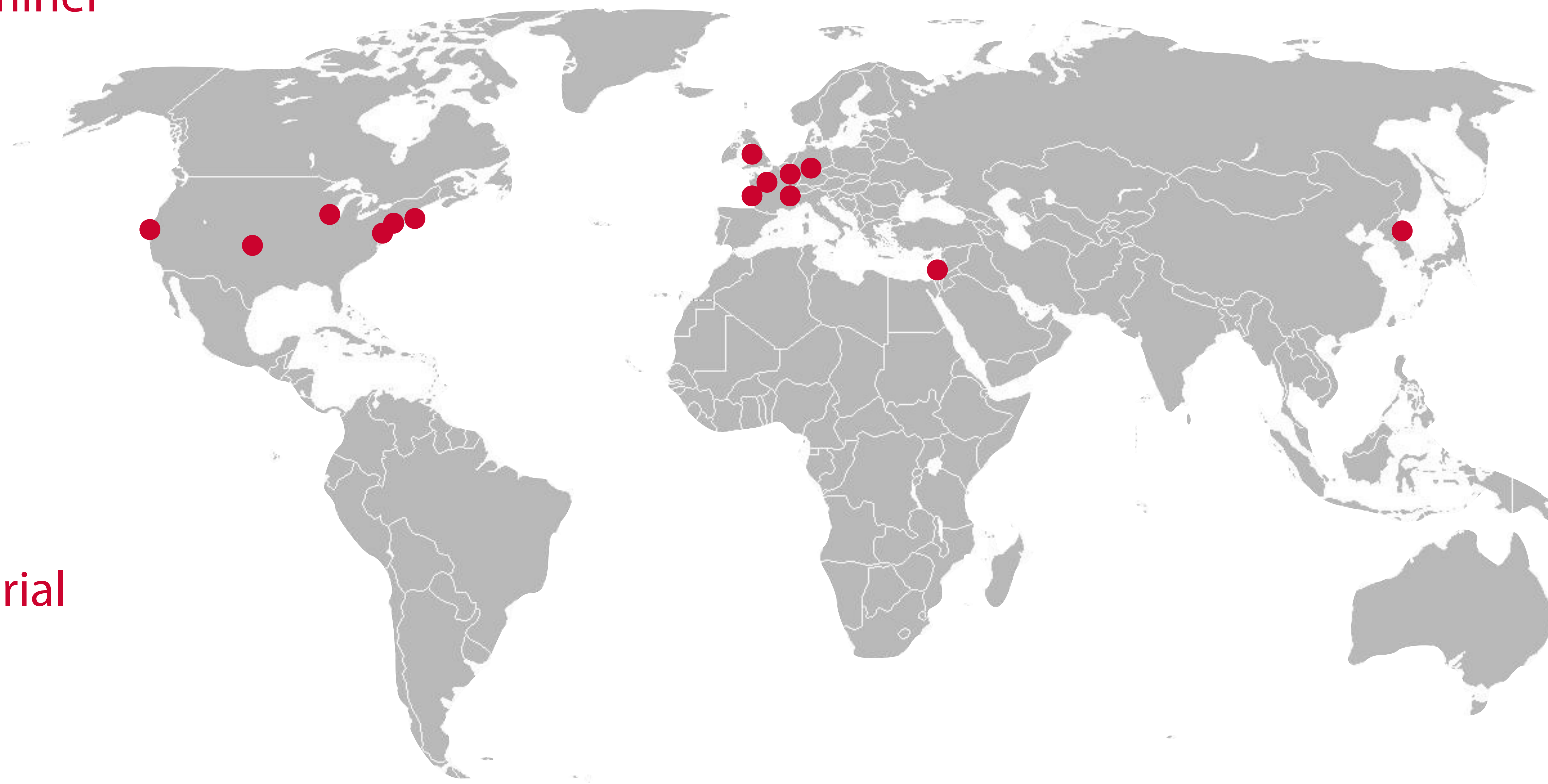
github.com/madminer-tool/madminer

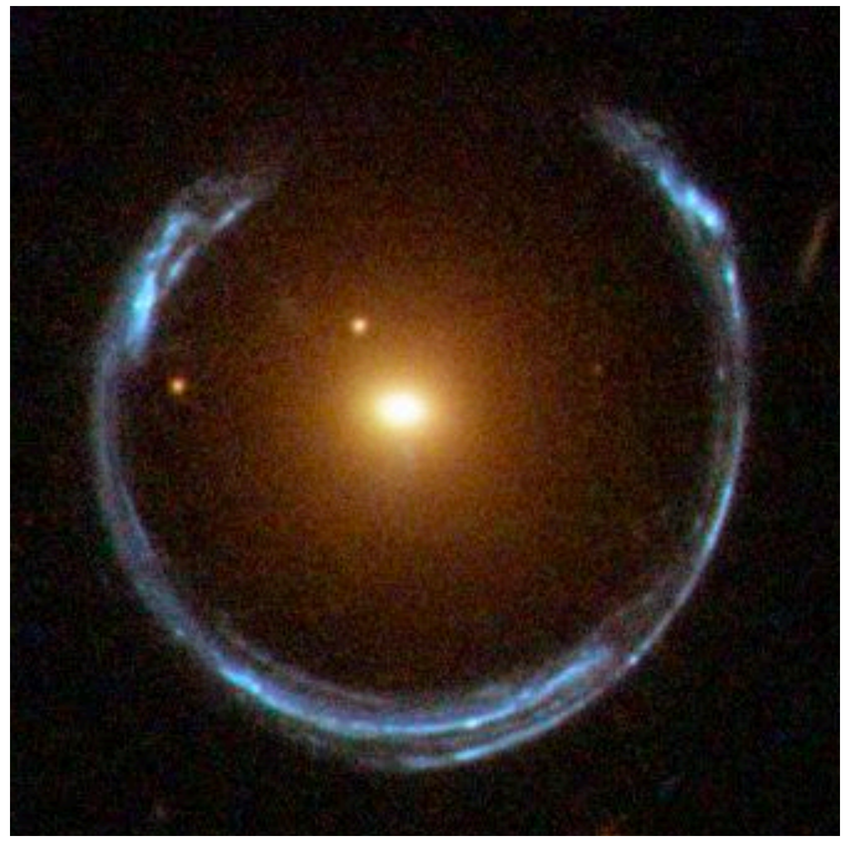
`pip install madminer`

hub.docker.com/u/madminertool

cranmer.github.io/madminer-tutorial

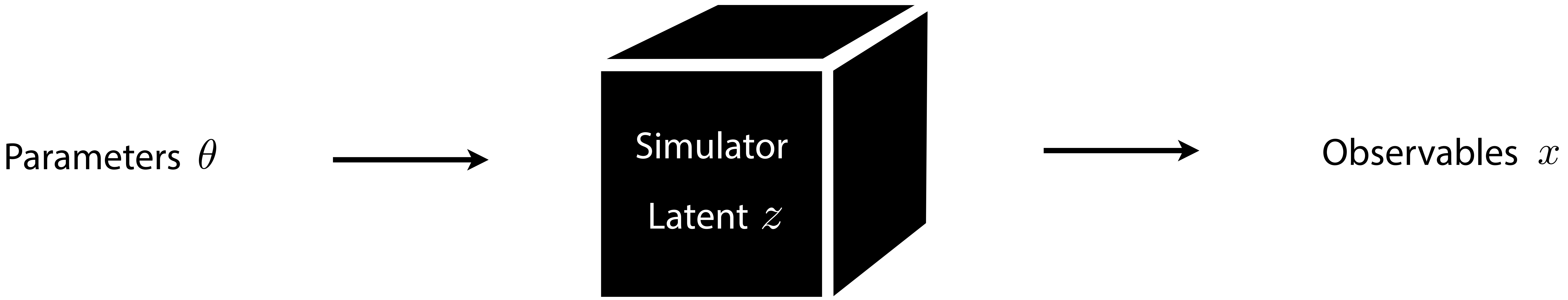
madminer.readthedocs.io





5. Beyond the LHC

Simulation-based ("likelihood-free") inference problems…



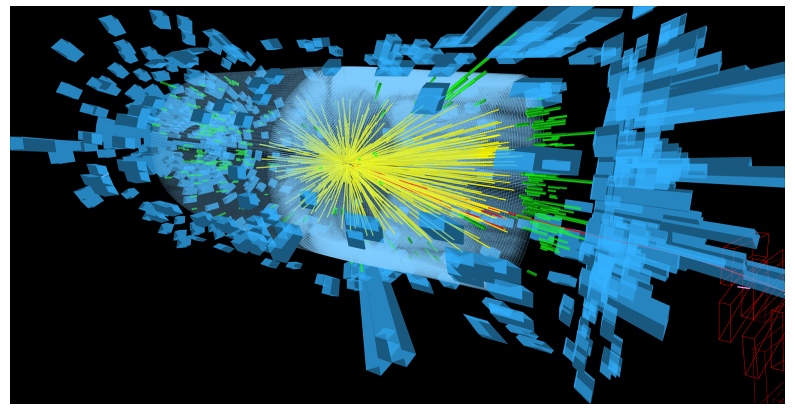
Prediction:

- Well-motivated mechanistic, causal model
- Simulator can generate samples $x \sim p(x|\theta)$

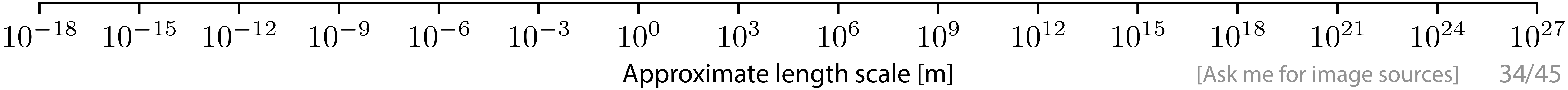
Inference:

- Interactions between low-level components lead to challenging inverse problems
- Likelihood $p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta)$ is intractable

… appear in many fields of science

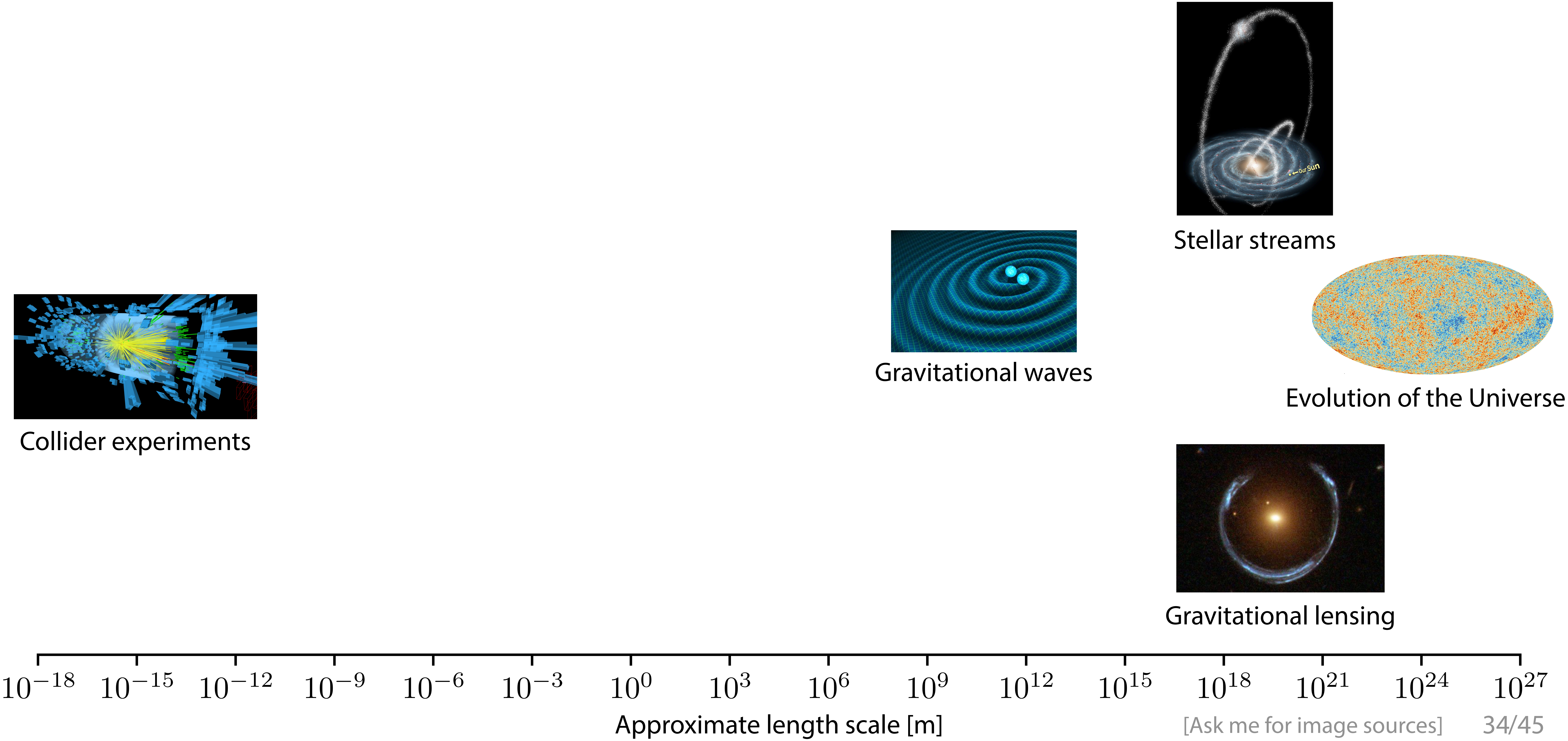


Collider experiments

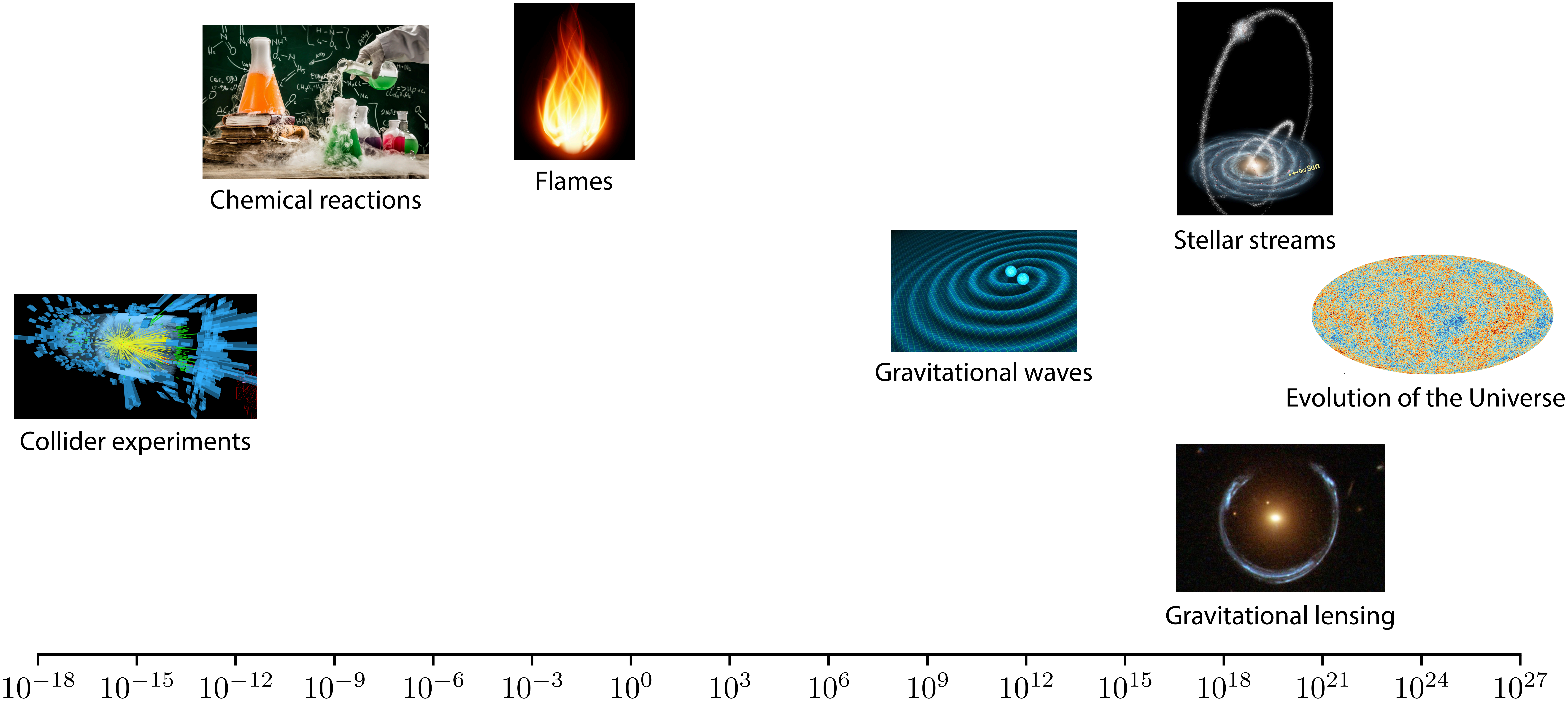


[Ask me for image sources]

… appear in many fields of science



… appear in many fields of science



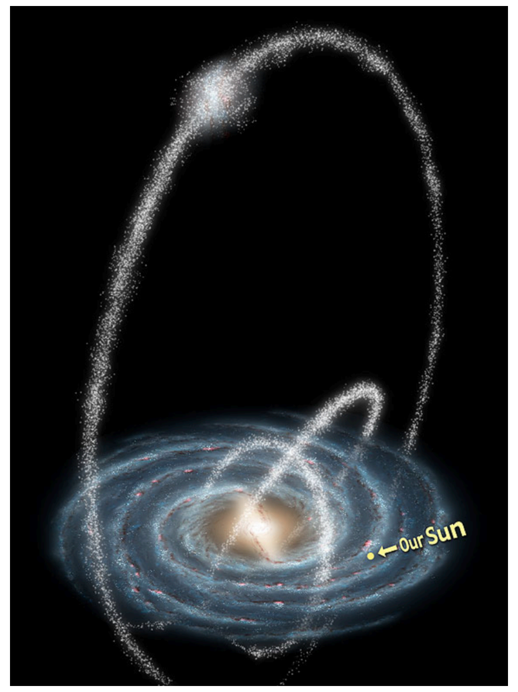
Approximate length scale [m]

[Ask me for image sources]

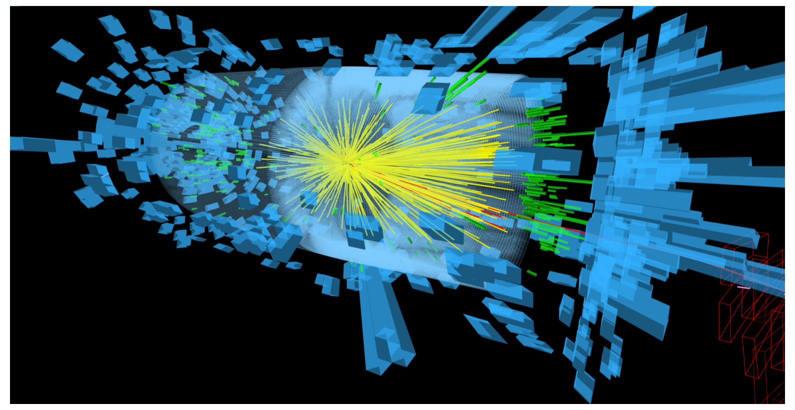
… appear in many fields of science

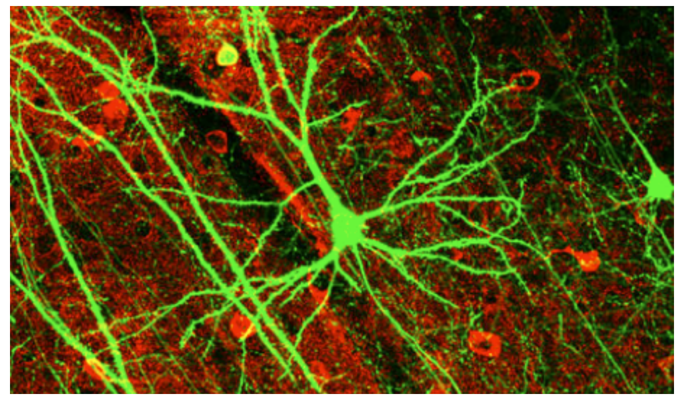

Chemical reactions

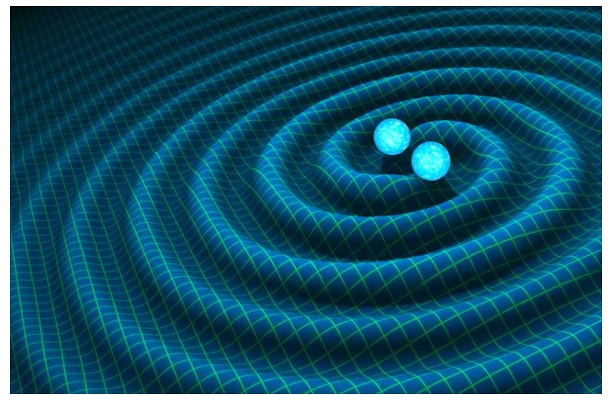

Flames

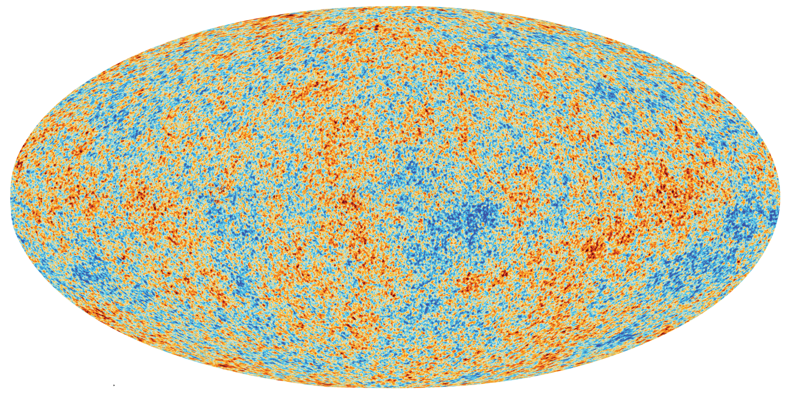


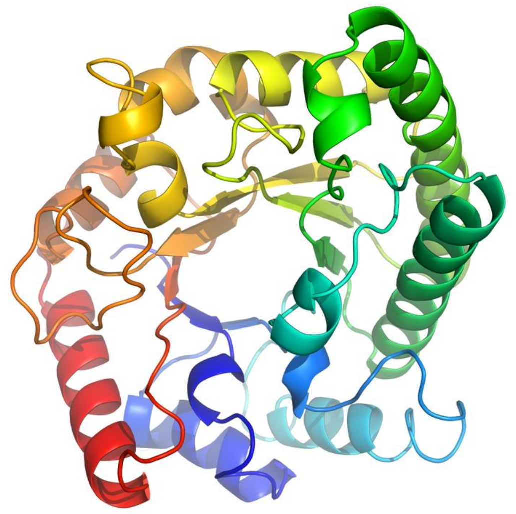
Stellar streams

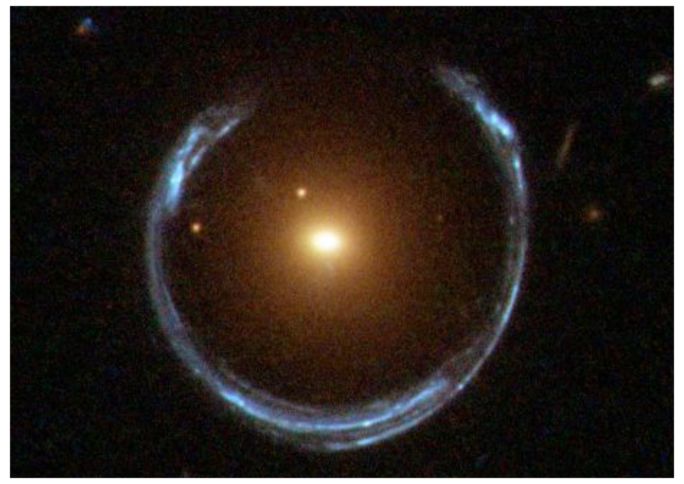

Collider experiments


Neurons


Gravitational waves


Evolution of the Universe

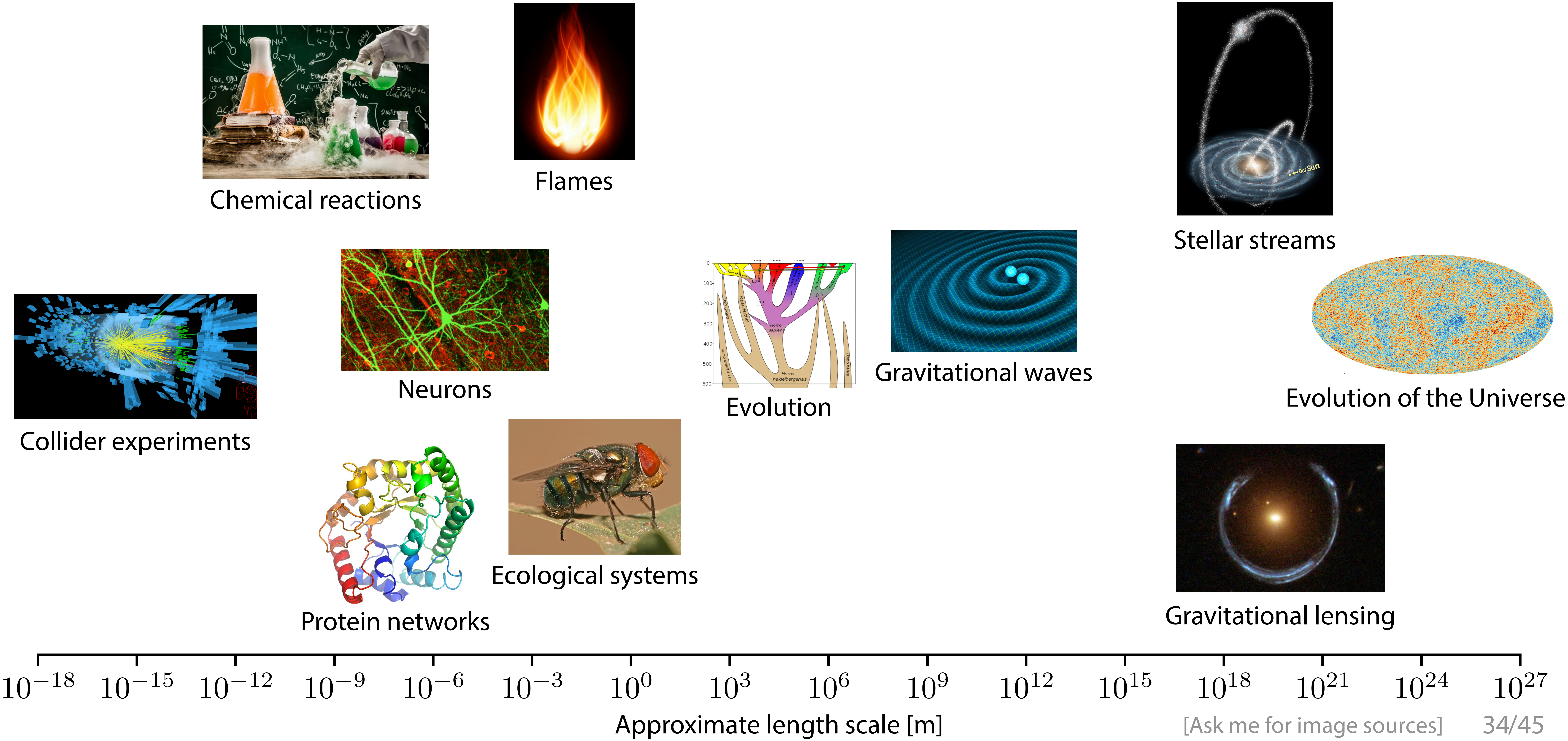

Protein networks


Gravitational lensing

Approximate length scale [m]

[Ask me for image sources]

… appear in many fields of science

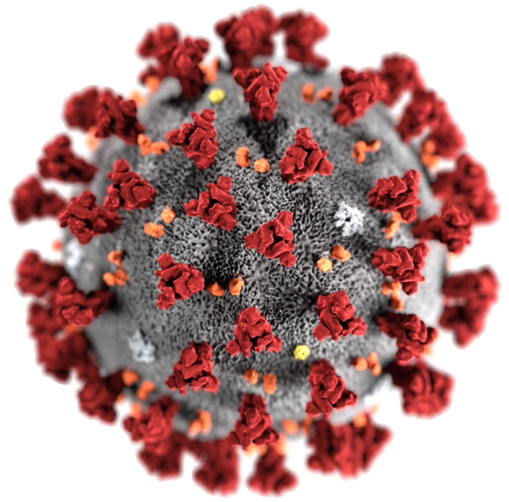


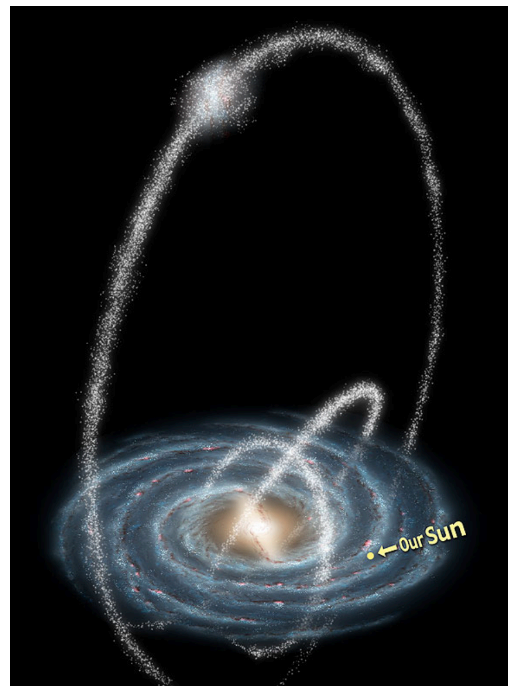
[Ask me for image sources]

… appear in many fields of science

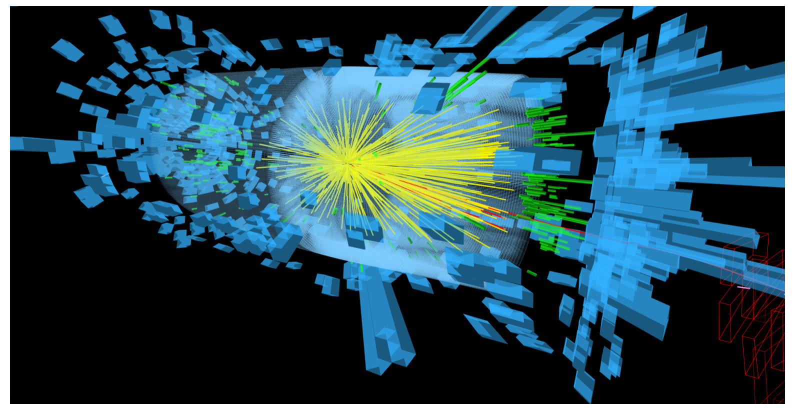

Chemical reactions

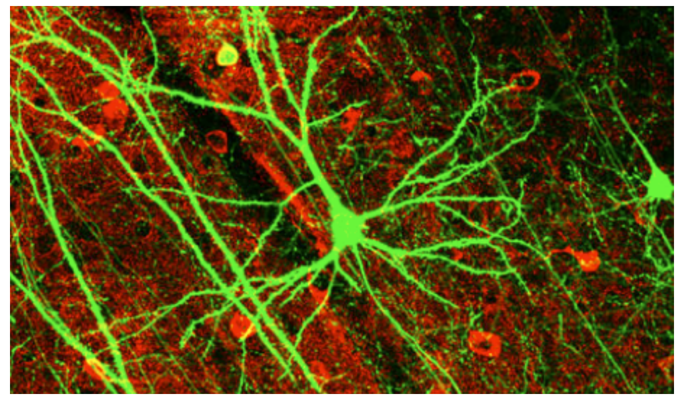

Flames

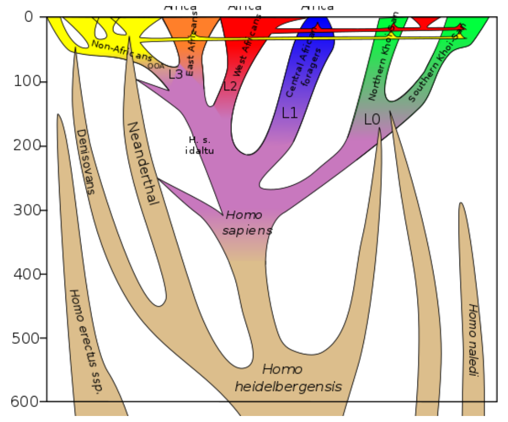

Epidemics

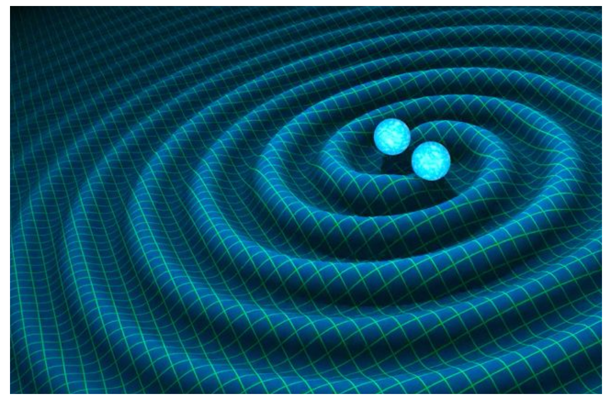


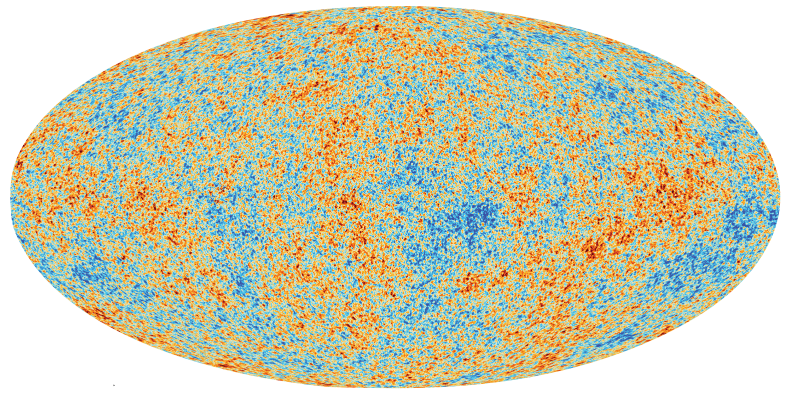
Stellar streams

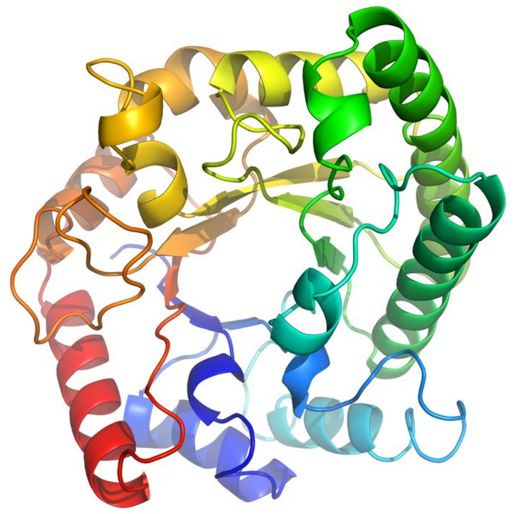

Collider experiments


Neurons

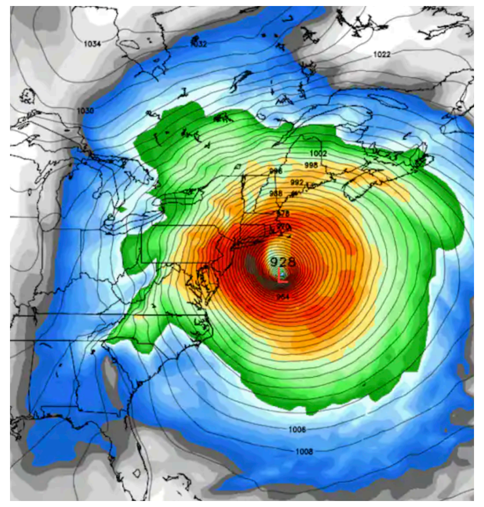

Evolution

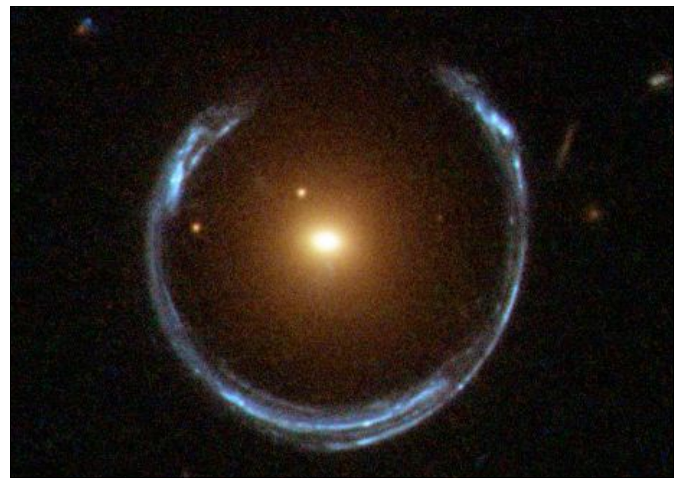

Gravitational waves


Evolution of the Universe


Protein networks


Ecological systems

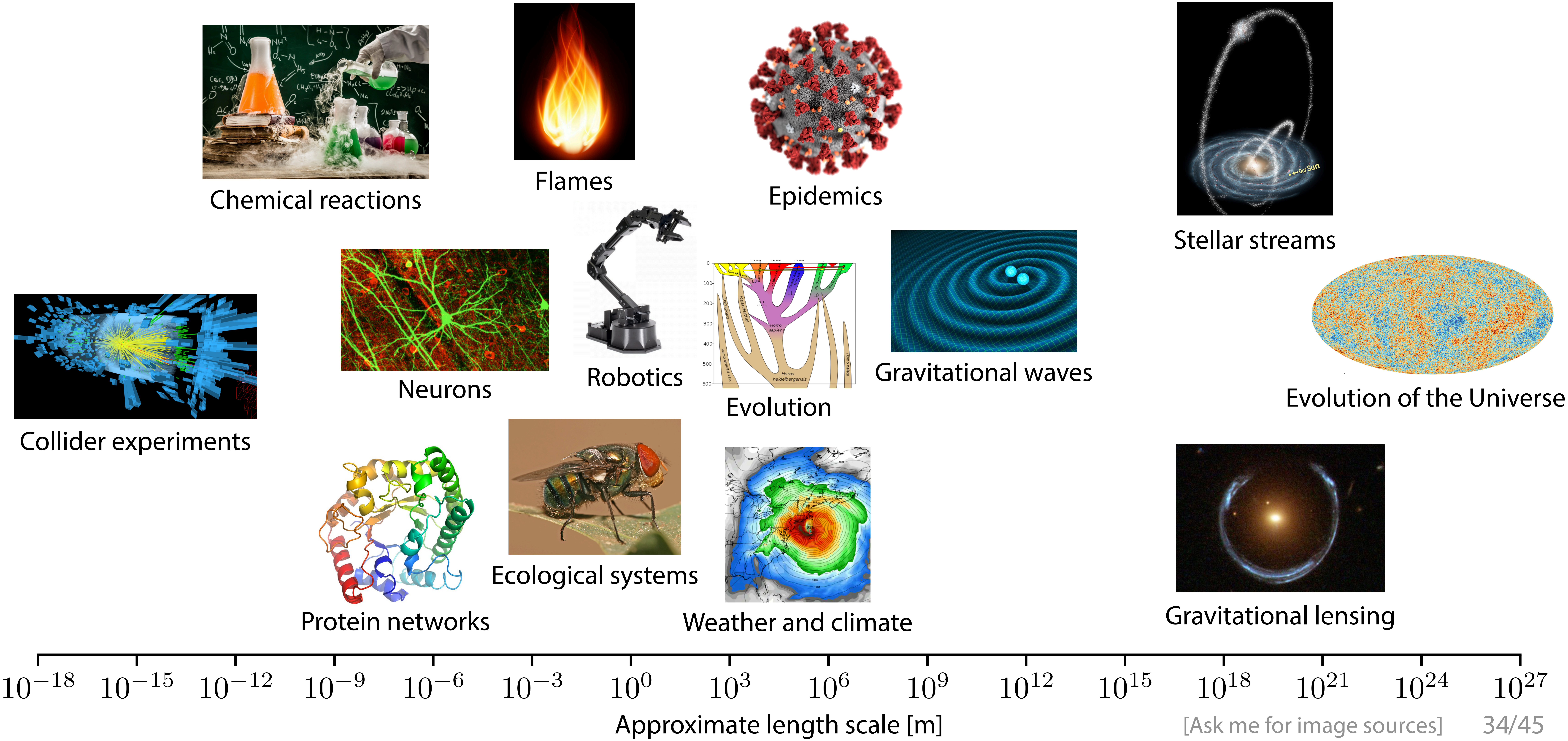

Weather and climate


Gravitational lensing

Approximate length scale [m]

[Ask me for image sources]

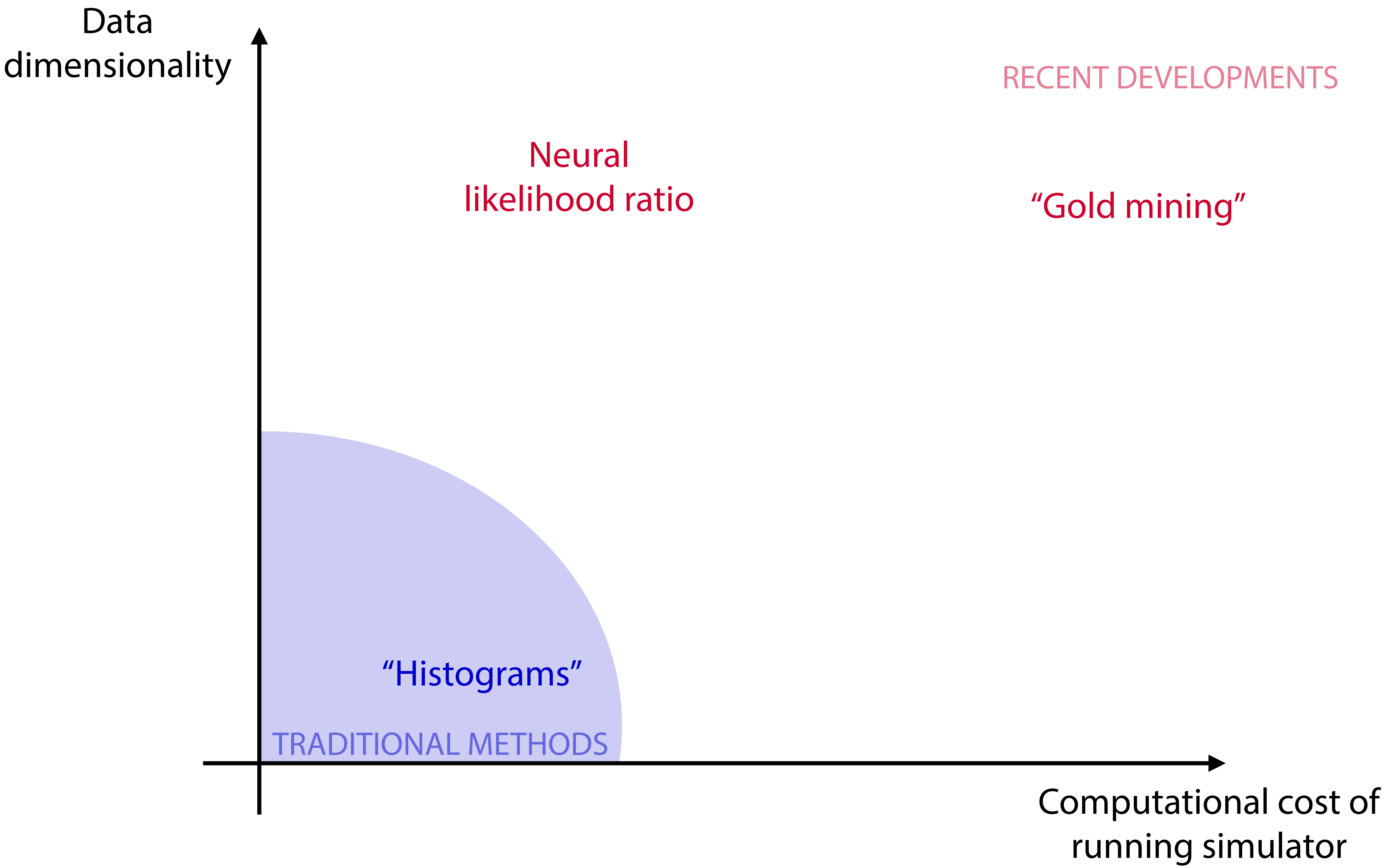
… appear in many fields of science



[Ask me for image sources]

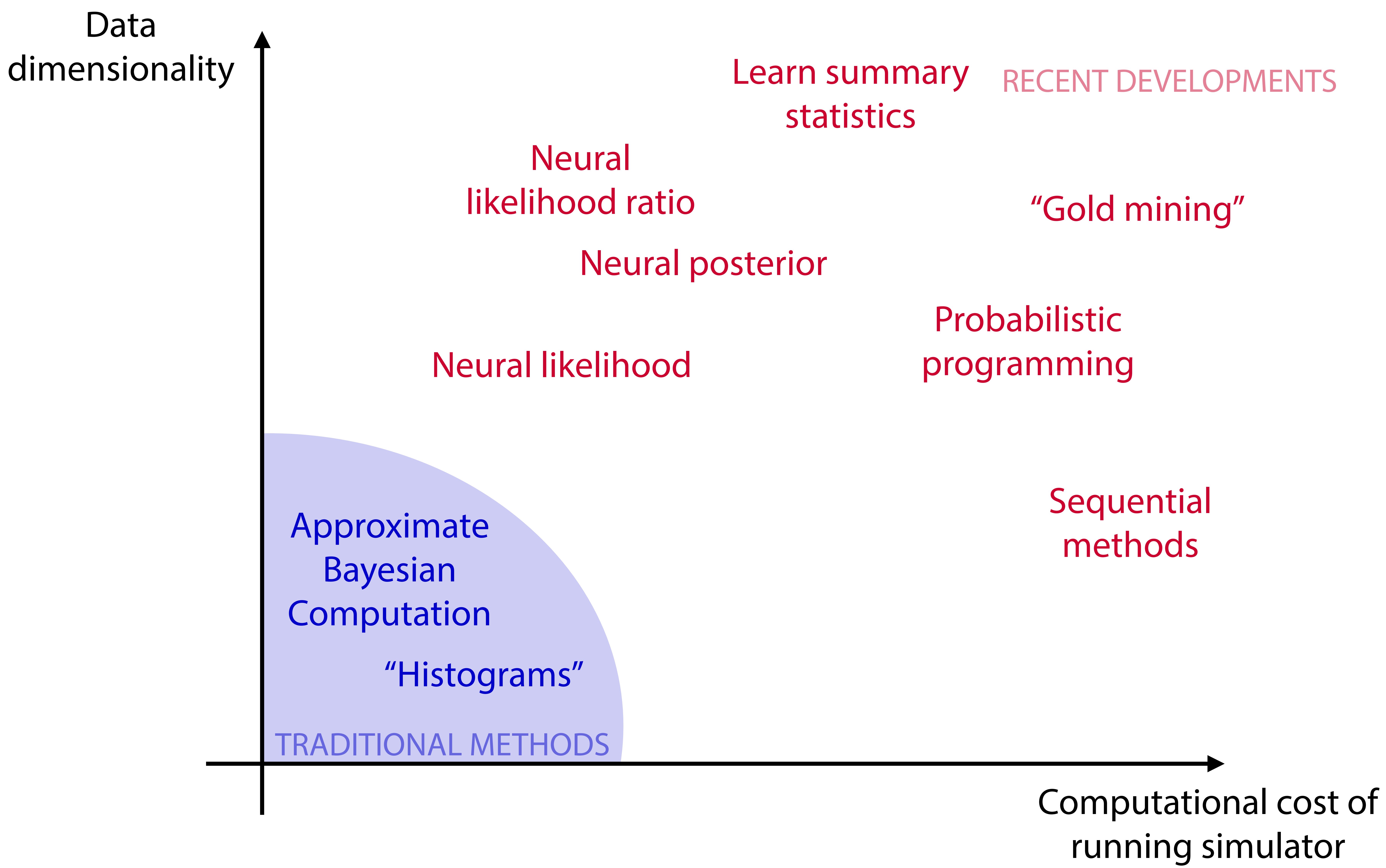
The frontier of simulation-based inference

[K. Cranmer, JB, G. Louppe 1911.01429]



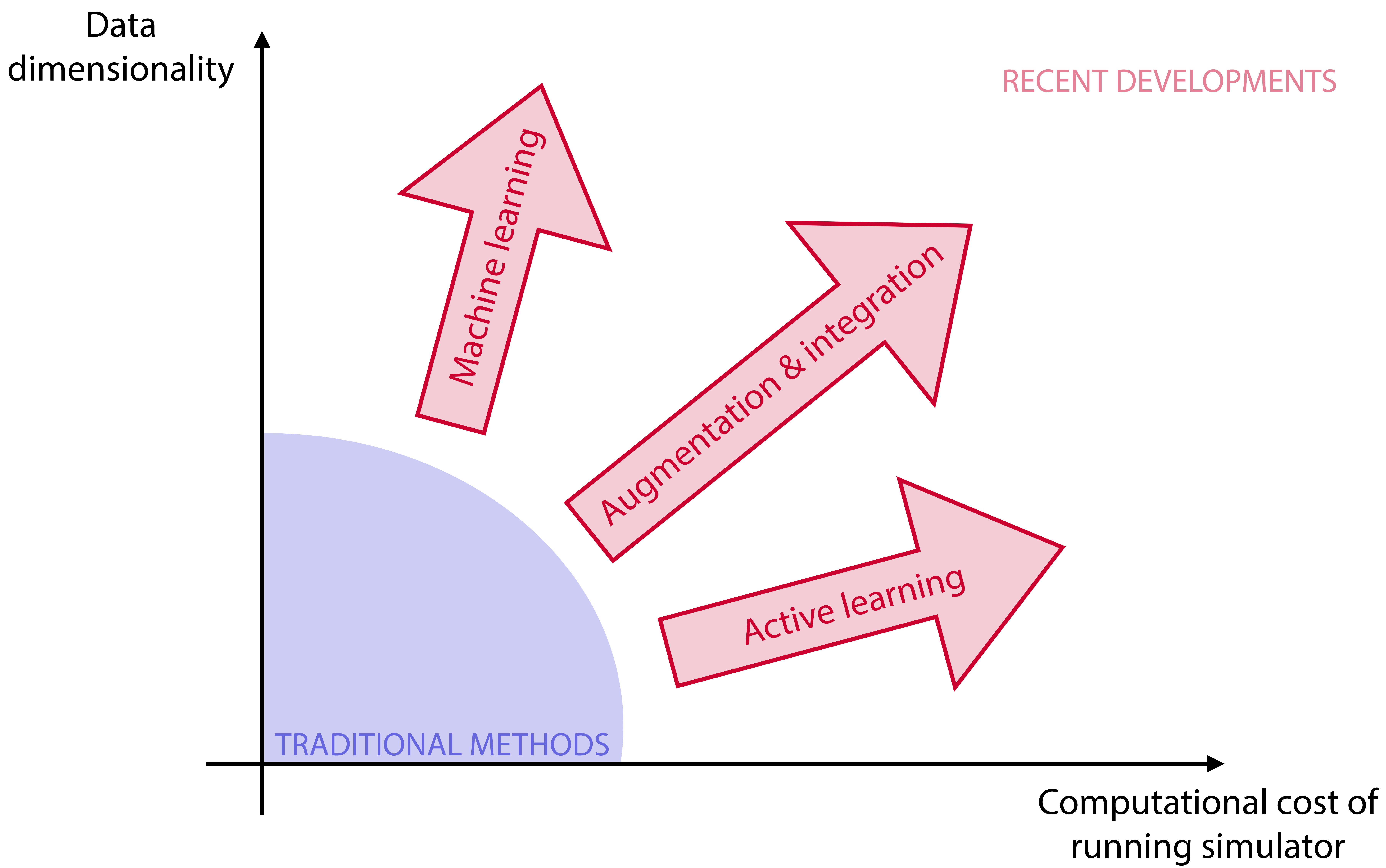
The frontier of simulation-based inference

[K. Cranmer, JB, G. Louppe 1911.01429]



The frontier of simulation-based inference

[K. Cranmer, JB, G. Louppe 1911.01429]



Th… …e

[K. Cranmer, JB, G. Louppe 1911.01429]

The frontier of simulation-based inference

Kyle Cranmer[a,b,1], Johann Brehmer[a,b], and Gilles Louppe[c]

[a]Center for Cosmology and Particle Physics, New York University, New York, NY 10003; [b]Center for Data Science, New York University, New York, NY 10011; and [c]Montefiore Institute, University of Liège, B-4000 Liège, Belgium

Edited by Jitendra Malik, University of California, Berkeley, CA, and approved April 10, 2020 (received for review November 4, 2019)

COLLOQUIUM PAPER

STATISTICS

Many domains of science have developed complex simulations to describe phenomena of interest. While these simulations provide high-fidelity models, they are poorly suited for inference and lead to challenging inverse problems. We review the rapidly developing field of simulation-based inference and identify the forces giving additional momentum to the field. Finally, we describe how the frontier is expanding so that a broad audience can appreciate the profound influence these developments may have on science.

statistical inference | implicit models | likelihood-free inference | approximate Bayesian computation | neural density estimation

Mechanistic models can be used to predict how systems will behave in a variety of circumstances. These run the gamut of distance scales, with notable examples including particle physics, molecular dynamics, protein folding, population genetics, neuroscience, epidemiology, economics, ecology, climate science, astrophysics, and cosmology. The expressiveness of programming languages facilitates the development of complex, high-fidelity simulations and the power of modern computing provides the ability to generate synthetic data from them. Unfortunately, these simulators are poorly suited for statistical inference. The source of the challenge is that the probability density (or likelihood) for a given observation—an essential ingredient for both frequentist and Bayesian inference methods—is typically intractable. Such models are often referred to as implicit models and contrasted against prescribed models where the likelihood for an observation can be explicitly calculated (1). The problem setting of statistical inference under intractable likelihoods has been dubbed likelihood-free inference—although it is a bit of a misnomer as typically one attempts to estimate the intractable likelihood, so we feel the term simulation-based inference is more apt.

The intractability of the likelihood is an obstruction for scientific progress as statistical inference is a key component of the scientific method. In areas where this obstruction has appeared, scientists have developed various ad hoc or field-specific methods to overcome it. In particular, two common traditional approaches rely on scientists to use their insight into the system to construct powerful summary statistics and then compare the observed data to the simulated data. In the first one, density estimation methods are used to approximate the distribution of the simulator—is being recognized as a key idea to improve the sample efficiency of various inference methods. A third direction of research has stopped treating the simulator as a black box and focused on integrations that allow the inference engine to tap into the internal details of the simulator directly.

Amidst this ongoing revolution, the landscape of simulation-based inference is changing rapidly. In this review we aim to provide the reader with a high-level overview of the basic ideas behind both old and new inference techniques. Rather than discussing the algorithms in technical detail, we focus on the current frontiers of research and comment on some ongoing developments that we deem particularly exciting.

Simulation-Based Inference

Simulators. Statistical inference is performed within the context of a statistical model, and in simulation-based inference the simulator itself defines the statistical model. For the purpose of this paper, a simulator is a computer program that takes as input a vector of parameters θ, samples a series of internal states or latent variables $z_i \sim p_i(z_i|\theta, z_{<i})$, and finally produces a data vector $x \sim p(x|\theta, z)$ as output. Programs that involve random samplings and are interpreted as statistical models are known as probabilistic programs, and simulators are an example. Within this general formulation, real-life simulators can vary substantially:

- The parameters θ describe the underlying mechanistic model and thus affect the transition probabilities $p_i(z_i|\theta, z_{<i})$. Typically the mechanistic model is interpretable by a domain scientist and θ has relatively few components and a fixed dimensionality. Examples include coefficients found in the Hamiltonian of a physical system, the virulence and incubation rate of a pathogen, or fundamental constants of Nature.
- The latent variables z that appear in the data-generating process may directly or indirectly correspond to a physically meaningful state of a system, but typically this state is unobservable in practice. The structure of the latent space varies substantially between simulators. The latent variables may be continuous or discrete and the dimensionality of the latent space may be fixed or may vary, depending on the control flow of the simulator. The simulation can freely combine deterministic and stochastic steps. The deterministic components of the simulator may be differentiable or may involve discontinuous control

…CENT DEVELOPMENTS

…omputational cost of running simulator

Th e

[K. Cranmer, JB, G. Louppe 1911.01429]

Check for

The frontier of simulation-based inference

Kyle Cranmer[a,b,1], Johann Brehmer[a,b], and Gilles Louppe[c]

[a]Center for Cosmology and Particle Physics, New York University, New York, NY 10003; [b]Center for Data Science, New Yor
and [c]Montefiore Institute, University of Liège, B-4000 Liège, Belgium

Edited by Jitendra Malik, University of California, Berkeley, CA, and approved April 10, 2020 (received for review Novemb

Many domains of science have developed complex simulations to describe phenomena of interest. While these simulations provide high-fidelity models, they are poorly suited for inference and lead to challenging inverse problems. We review the rapidly developing field of simulation-based inference and identify the forces giving additional momentum to the field. Finally, we describe how the frontier is expanding so that a broad audience can appreciate the profound influence these developments may have on science.

statistical inference | implicit models | likelihood-free inference | approximate Bayesian computation | neural density estimation

Mechanistic models can be used to predict how systems will behave in a variety of circumstances. These run the gamut of distance scales, with notable examples including particle physics, molecular dynamics, protein folding, population genetics, neuroscience, epidemiology, economics, ecology, climate science, astrophysics, and cosmology. The expressiveness of programming languages facilitates the development of complex, high-fidelity simulations and the power of modern computing provides the ability to generate synthetic data from them. Unfortunately, these simulators are poorly suited for statistical inference. The source of the challenge is that the probability density (or likelihood) for a given observation—an essential ingredient for both frequentist and Bayesian inference methods—is typically intractable. Such models are often referred to as implicit models and contrasted against prescribed models where the likelihood for an observation can be explicitly calculated (1). The problem setting of statistical inference under intractable likelihoods has been dubbed likelihood-free inference—although it is a bit of a misnomer as typically one attempts to estimate the intractable likelihood, so we feel the term simulation-based inference is more apt.

The intractability of the likelihood is an obstruction for scientific progress as statistical inference is a key component of the scientific method. In areas where this obstruction has appeared, scientists have developed various ad hoc or field-specific methods to overcome it. In particular, two common traditional approaches rely on scientists to use their insight into the system to construct powerful summary statistics and then compare the observed data to the simulated data. In the first one, density estimation methods are used to approximate the distribution of

the simulator—is being recognized
sample efficiency of various inferen
of research has stopped treating the
focused on integrations that allow
into the internal details of the simu

Amidst this ongoing revolution,
based inference is changing rapid
provide the reader with a high-leve
behind both old and new inferen
discussing the algorithms in techn
current frontiers of research and
developments that we deem particu

Simulation-Based Inference

Simulators. Statistical inference is p
of a statistical model, and in sim
simulator itself defines the statisti
of this paper, a simulator is a co
as input a vector of parameters θ,
states or latent variables $z_i \sim p_i(z_i$
a data vector $x \sim p(x|\theta, z)$ as ou
random samplings and are interpre
known as probabilistic programs, a
ple. Within this general formulation
substantially:

- The parameters θ describe the u
and thus affect the transition pro
ically the mechanistic model is
scientist and θ has relatively fe
dimensionality. Examples includ
Hamiltonian of a physical system,
rate of a pathogen, or fundament
- The latent variables z that appea
cess may directly or indirectly corr
ingful state of a system, but typica
in practice. The structure of the la
between simulators. The latent v
or discrete and the dimensionalit
fixed or may vary, depending on
ulator. The simulation can freely
stochastic steps. The deterministi
tor may be differentiable or may i

Likelihood-Free Inference Workshop

18-22 March 2019 @ Flatiron Institute, NYC

Home Schedule Hackathon Logistics Participants Registration

Rationale

The goal of this interdisciplinary meeting is to gather developers and users of Likelihood-Free Inference methods to share latest techniques, use cases and applications across different fields, and discuss open challenges.

The first two days of the workshop will be focused on talks and discussions, while the remaining of the week will be dedicated to a hackathon with the goal of seeding the development of a common probabilistic programming framework for Likelihood-Free Inference as well as collaboratively working on LFI-related hack projects.

News

- *March 4th, 2019* : Preliminary schedule available, new Gitter channel chat on gitter, new Hackathon page
- *February 19th, 2019* : Main registration is closed, contact organizers for late registration
- *February 19th, 2019* : Travel funding application deadline
- *February 6th, 2019* : Opening registration

Organizing Committee

- Justin Alsing, Oskar Klein Center, Stockholm University
- Johann Brehmer, Center for Data Science, New York University
- Stephen Feeney, Center for Computational Astrophysics, Flatiron Institute

Gravitational lensing example

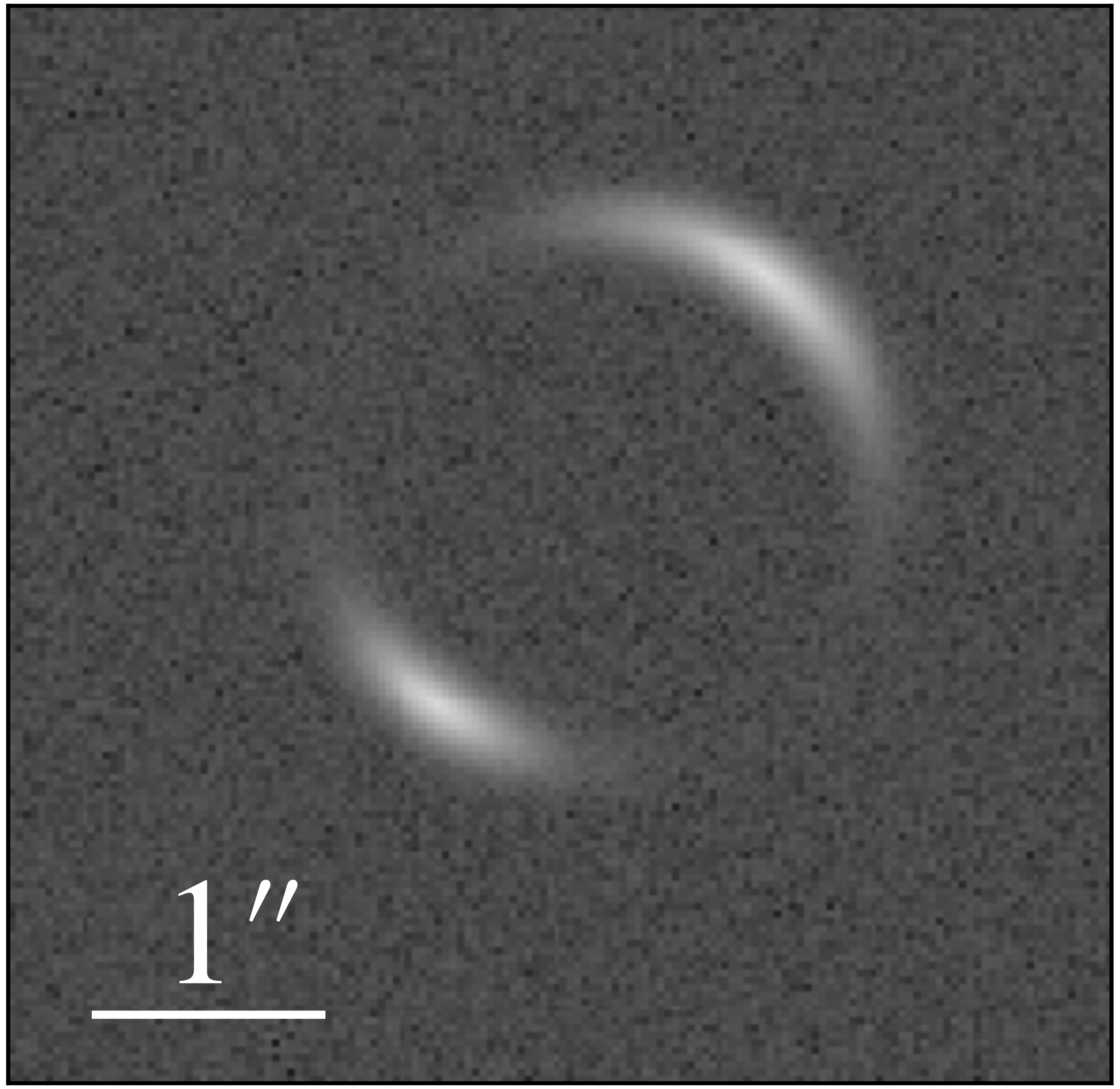
[T. Brown, J.Tumlinson]

[NASA/ESA; Abe Books]

[Hubble Space Telescope]

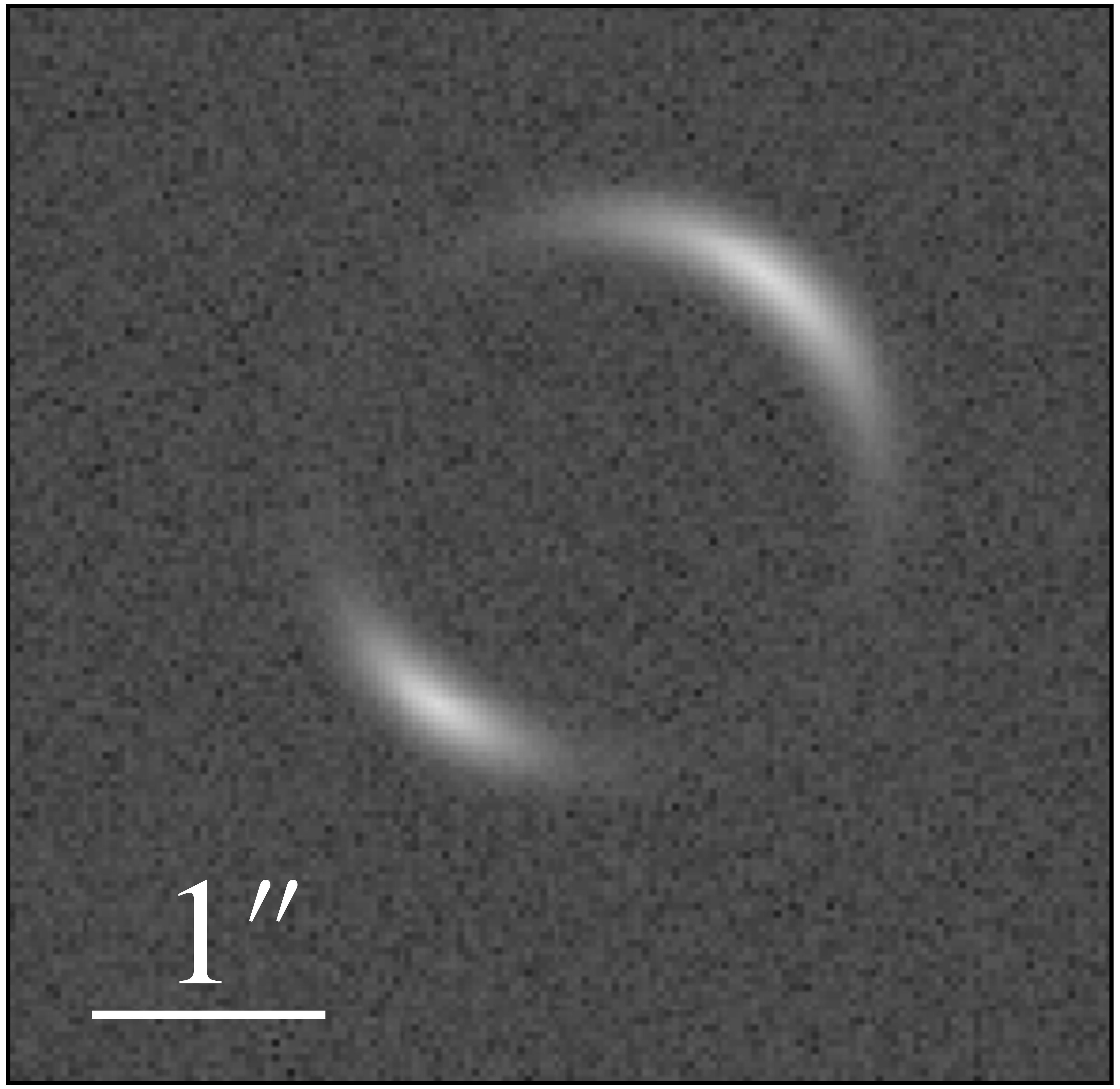
Subhalos affect strong lensing

Smooth halo only

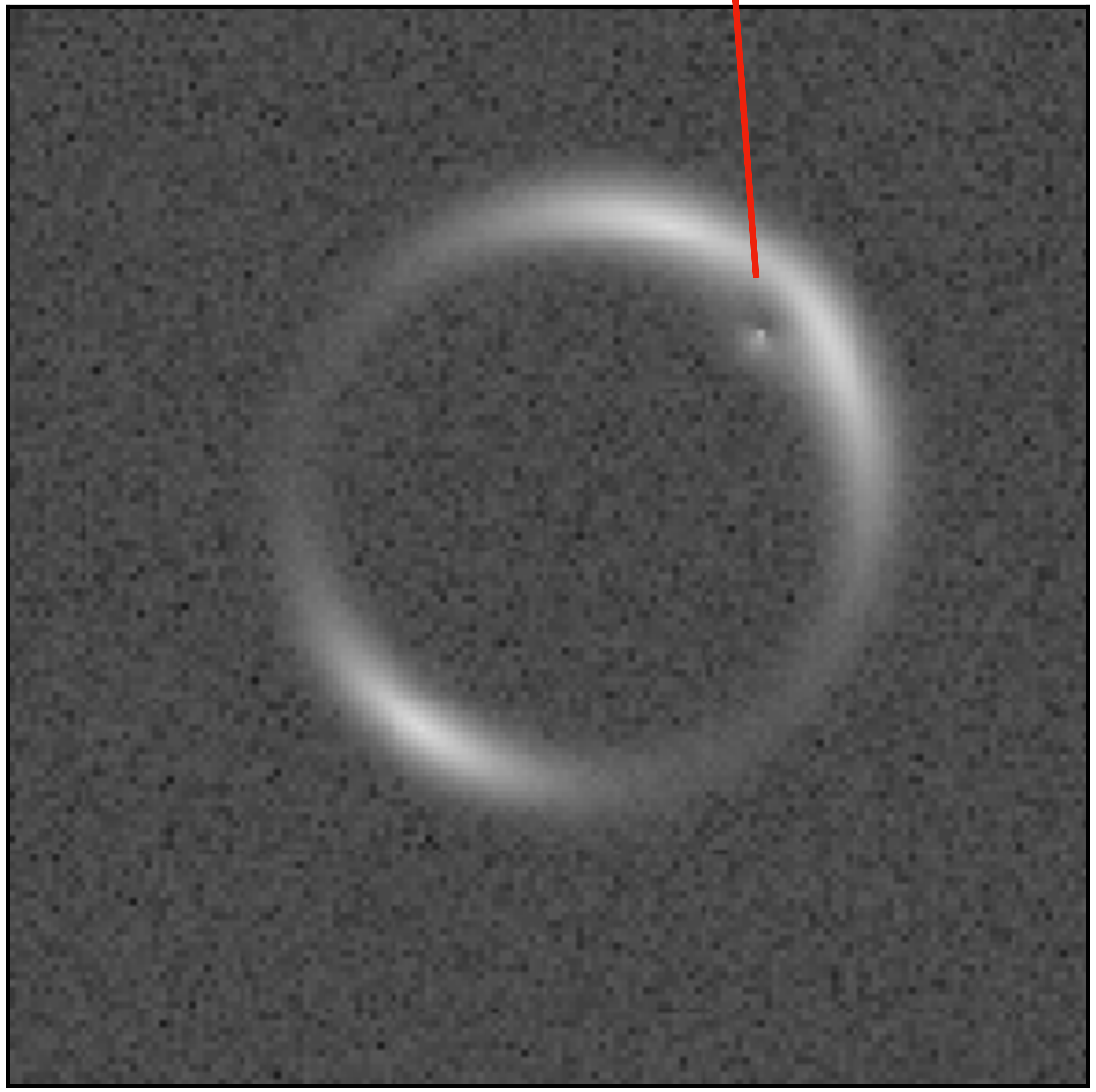


Subhalos affect strong lensing

Smooth halo only

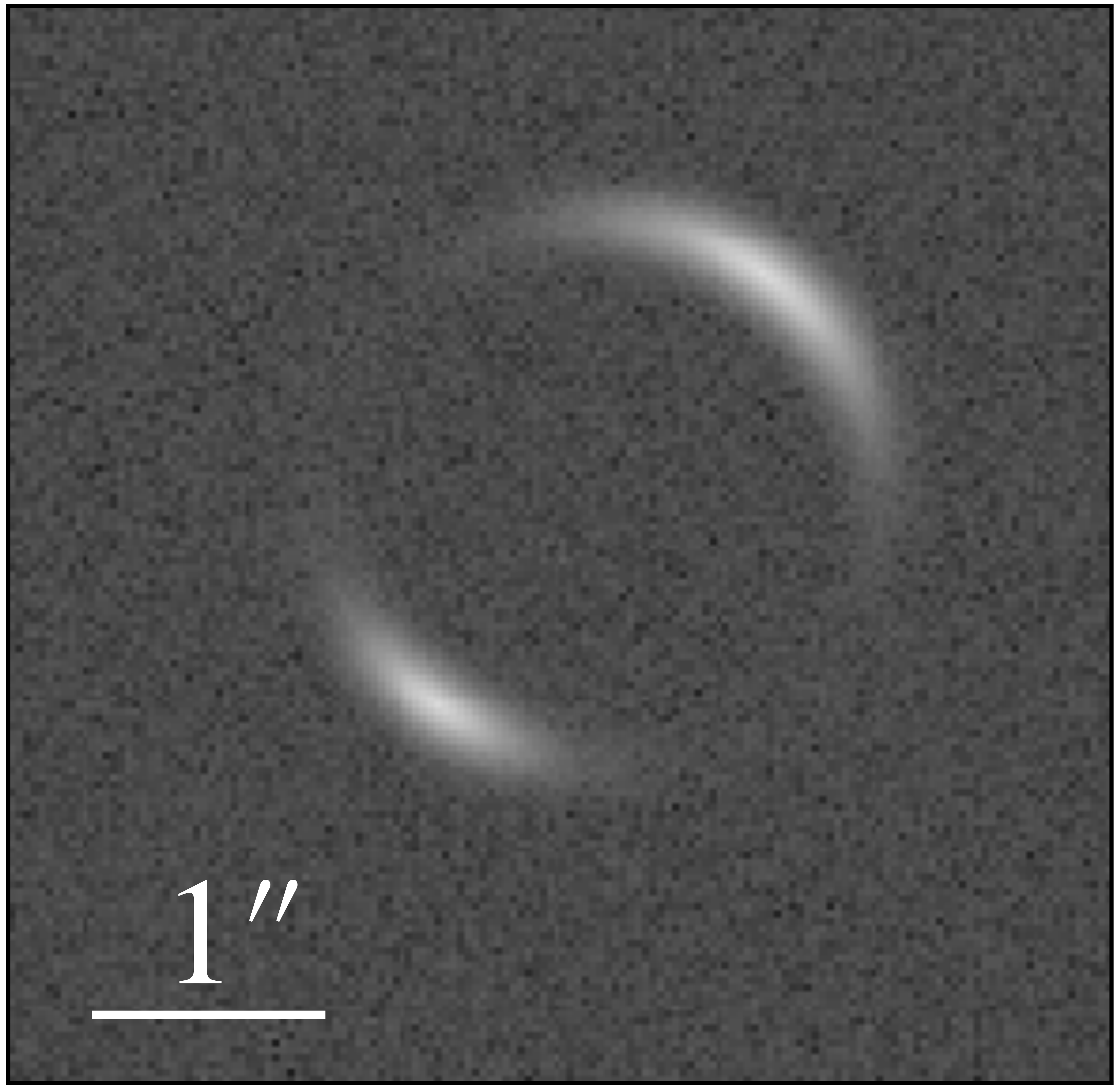


Smooth halo + subhalo

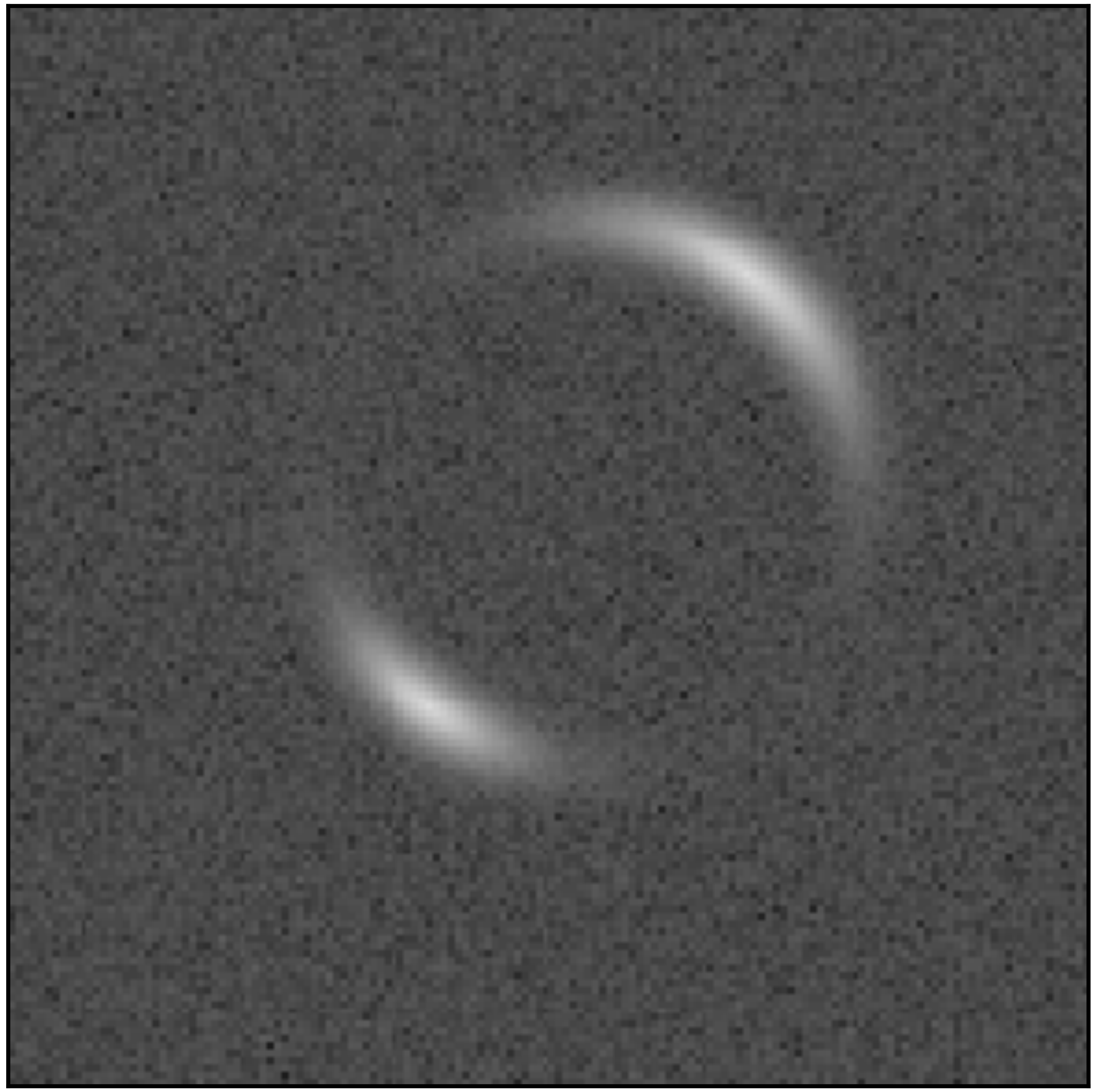


Subhalos affect strong lensing… realistically

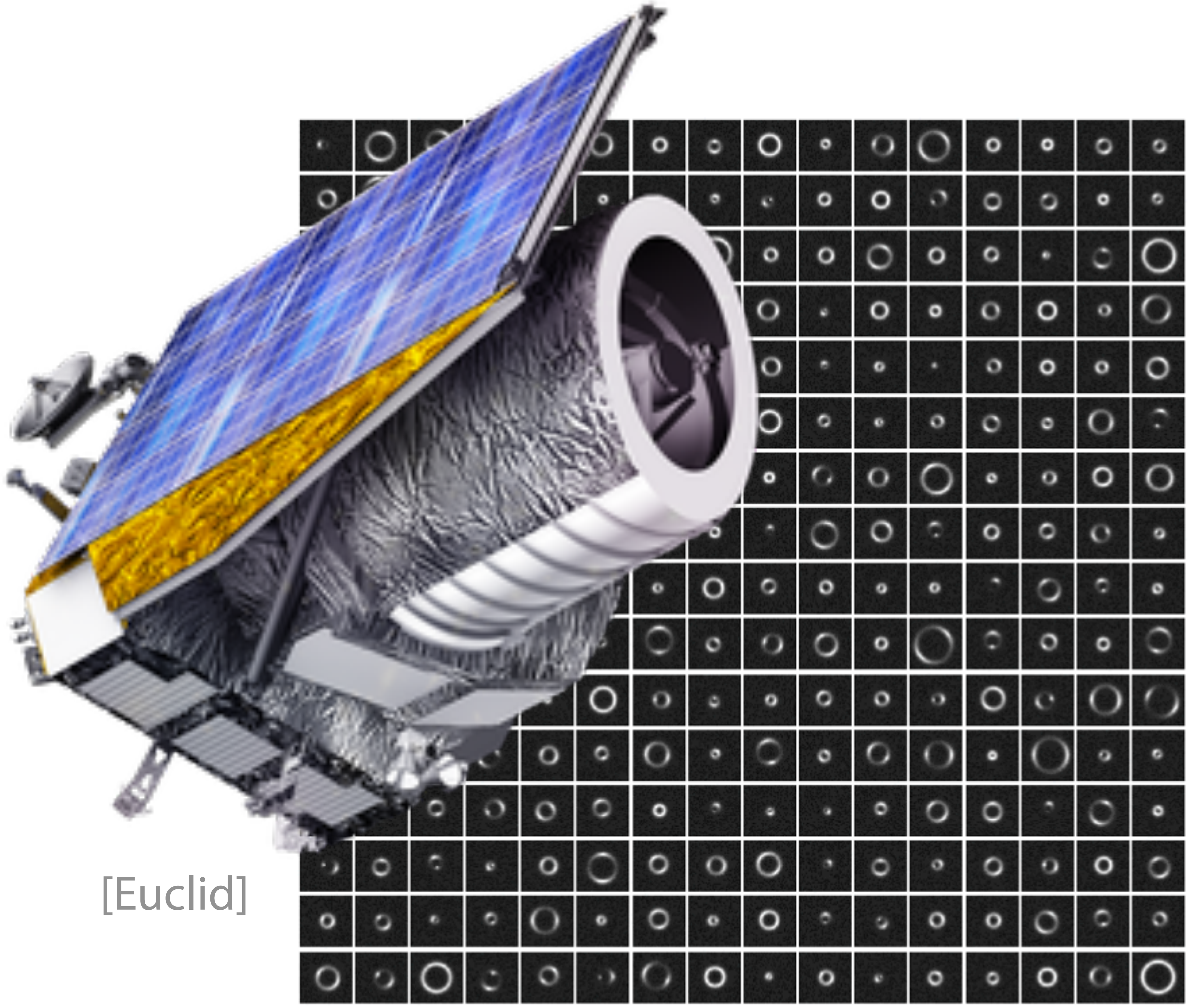
Smooth halo only

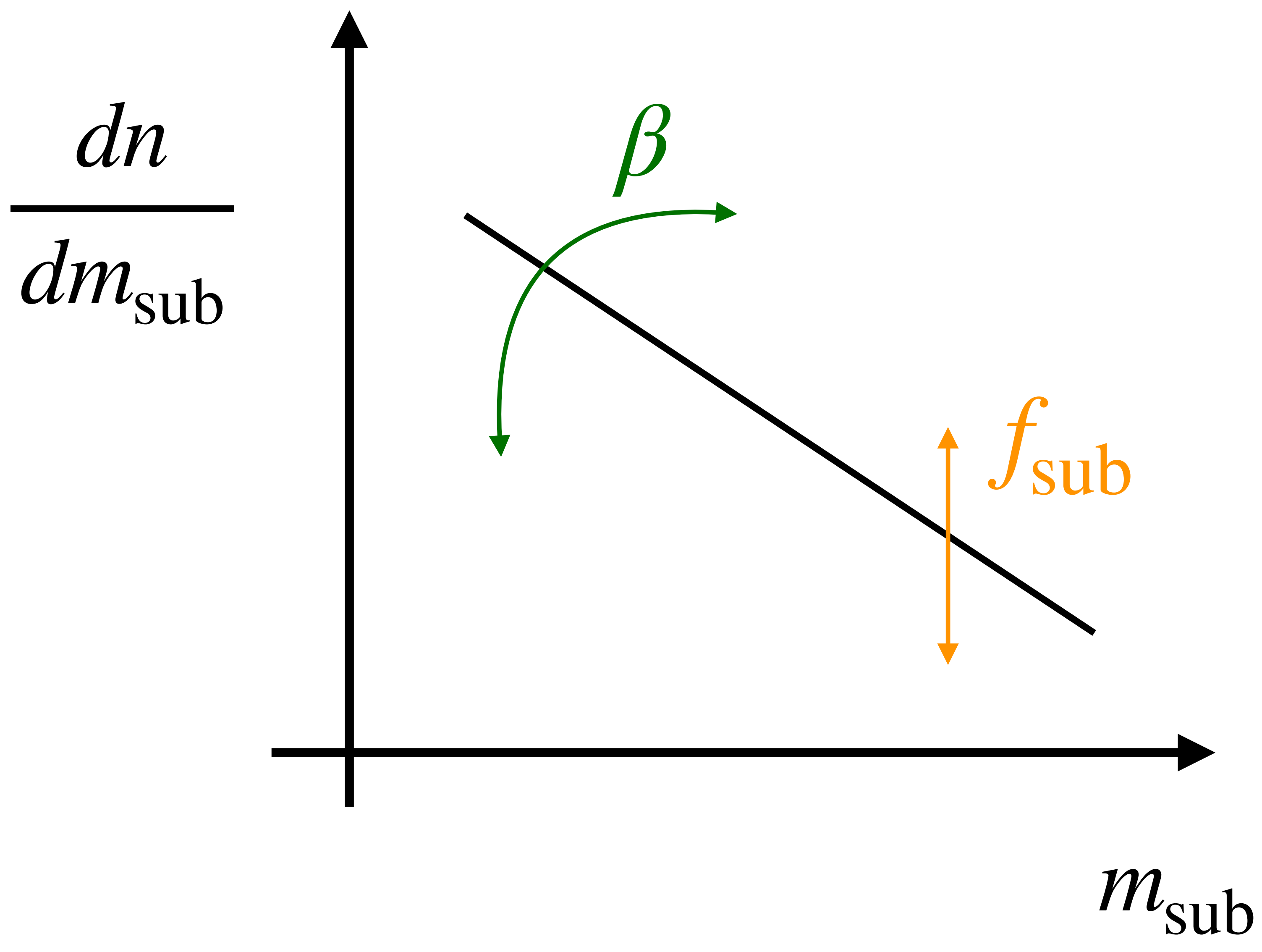


Smooth halo + subhalos



Scalable inference for small subhalos



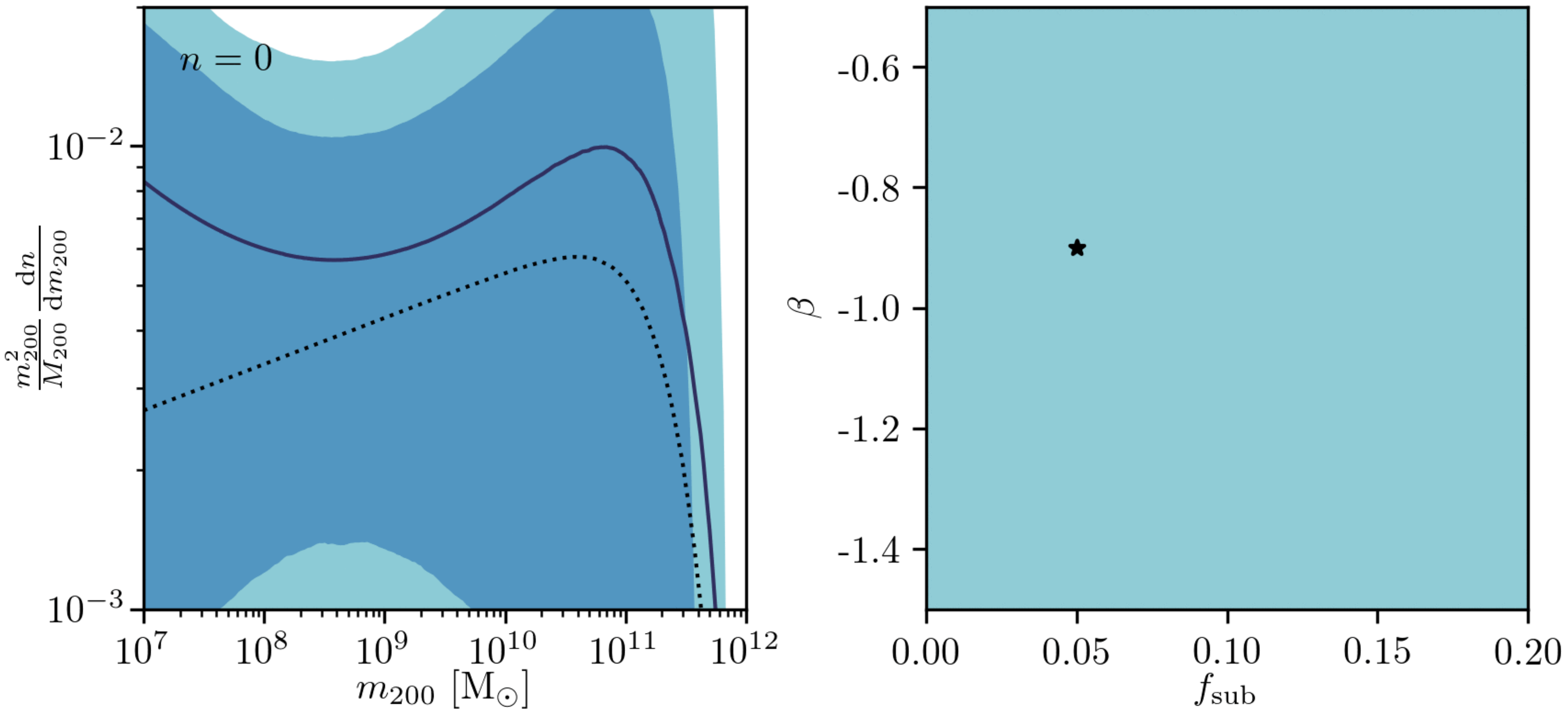


Near-future telescopes and satellites will collect hundreds of lensing images [Collett et al 1507.02657]

Goal: infer DM properties from all images and all clumps at once

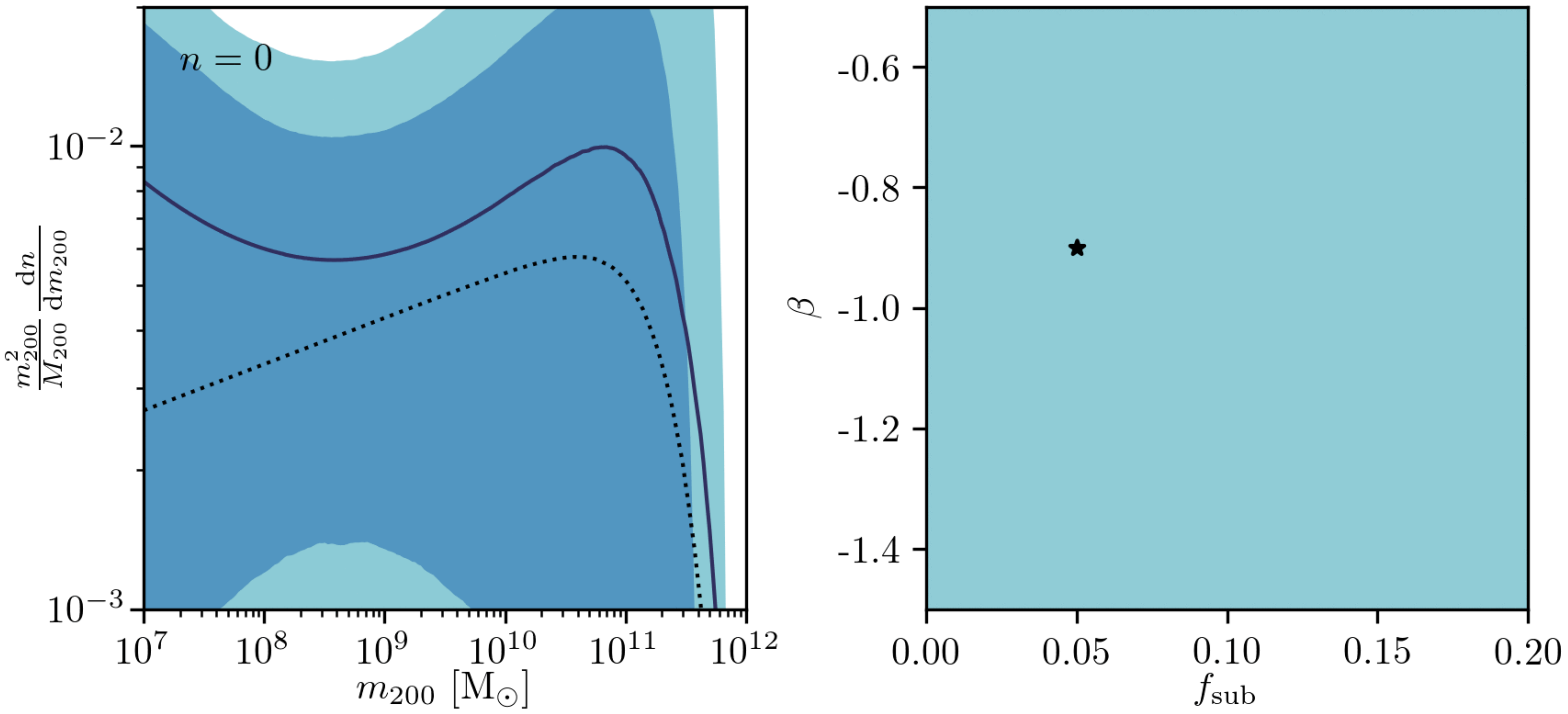
ML-based Bayesian inference

[JB, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer 1909.02005]



ML-based Bayesian inference

[JB, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer 1909.02005]




Kyle Cranmer


Gilles Louppe


Juan Pavez


Markus Stoye


Felix Kling


Irina Espejo


Sinclert Perez

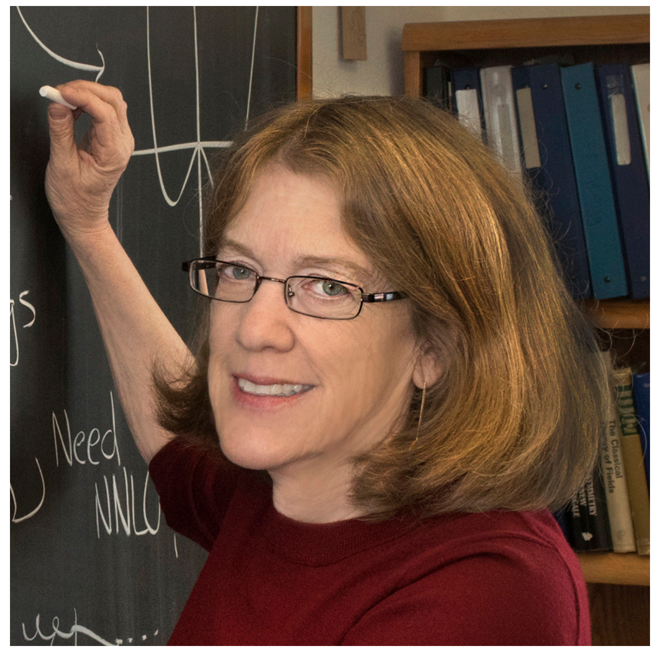

Sid Mishra-Sharma


Zubair Bhatti


Tilman Plehn


Anja Butter


Nathalie Soybelman


Sally Dawson


Sam Homiller

Parts of this talk were inspired by great presentations by Kyle Cranmer, Gilles Louppe, Sid Mishra-Sharma, and Jakob Macke

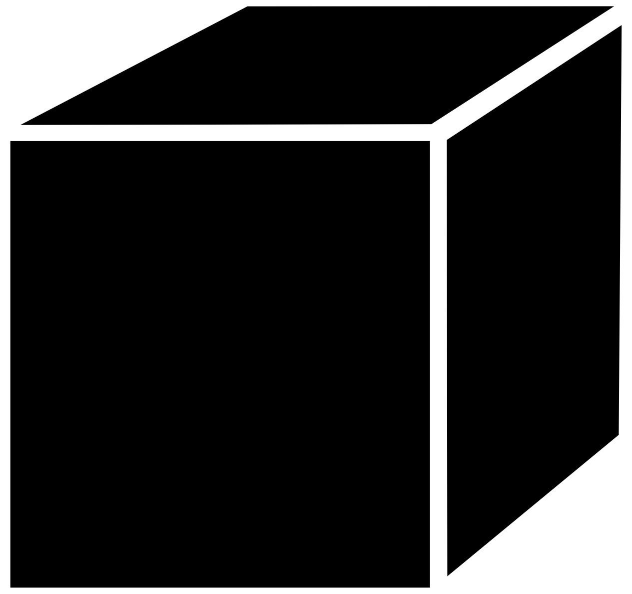
NYU

MSDSE
Moore - Sloan
Data Science Environments

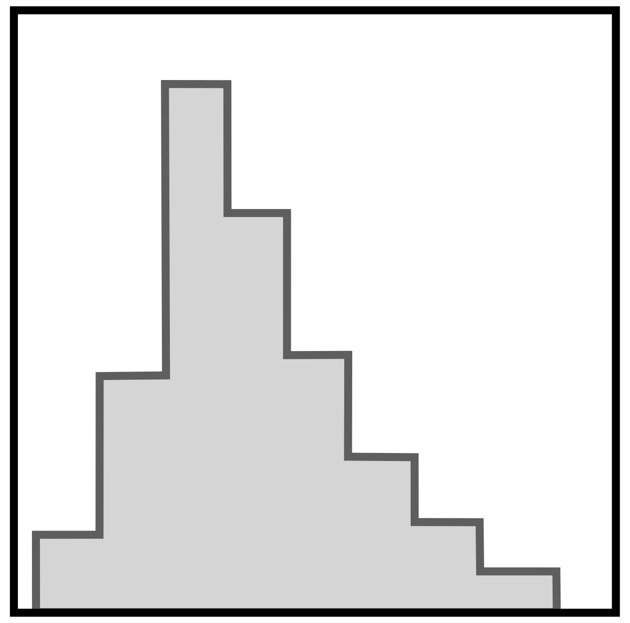
dianahep

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

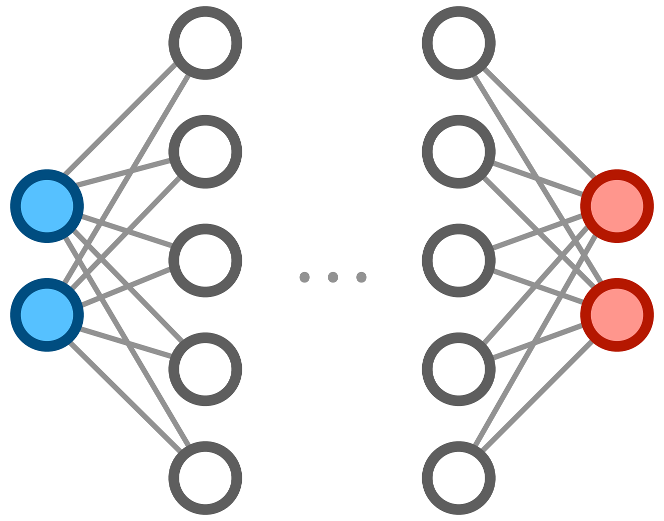
The SCAILFIN Project
scailfin.github.io



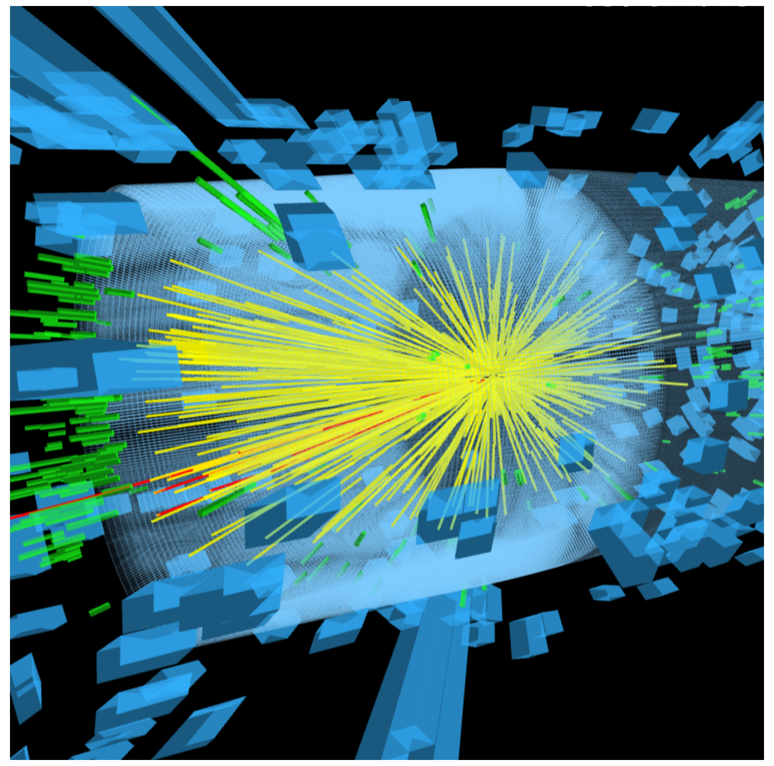
Simulators make precise predictions, but inference is challenging.



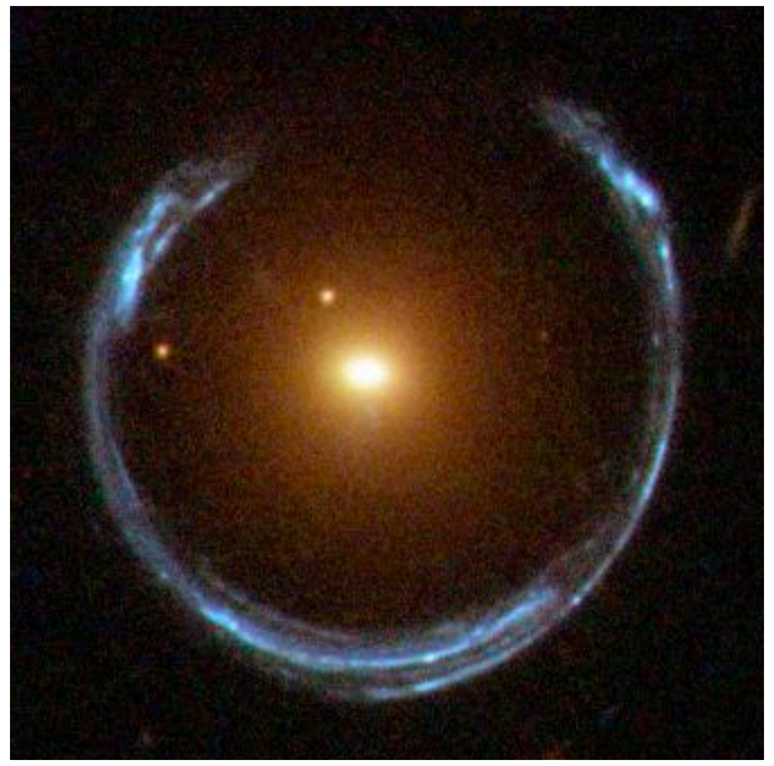
Scientists have side-stepped this problem with summary statistics.



Machine learning enables powerful inference methods, especially when we inject domain information.



They work in problems from the smallest…



… to the largest scales.

Selected references

Reviews

K. Cranmer, **J. Brehmer**, G. Louppe:
"The frontier of simulation-based inference"
PNAS, 1911.01429

J. Brehmer and K. Cranmer:
"Simulation-based inference methods for particle physics"
Chapter in "AI for HEP", 2010.06439

Simulation–based inference methods

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

M. Stoye, **J. Brehmer**, K. Cranmer, G. Louppe, J. Pavez:
"Likelihood-free inference with an improved cross-entropy estimator"
NeurIPS workshop, 1808.00973

Particle physics

J. Brehmer, K. Cranmer, G. Louppe, J. Pavez:
"Constraining Effective Field Theories with machine learning"
PRL, 1805.00013

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

J. Brehmer, F. Kling, I. Espejo, K. Cranmer:
"MadMiner: Machine learning–based inference for particle physics"
CSBS, 1907.10621, https://github.com/diana-hep/madminer

N. Soybelman, A. Butter. T. Plehn, **J. Brehmer**:
"Stronger symbolic summary statistics for the LHC"
NeurIPS workshop

J. Brehmer, K. Cranmer, F. Kling, and T. Plehn:
"Better Higgs Measurements Through Information Geometry"
PRD, 1612.05261

Astrophysics

J. Brehmer, S. Mishra-Sharma, J. Hermans, G. Louppe, K. Cranmer
"Mining for Dark Matter Substructure: Inferring subhalo population properties from strong lenses with machine learning"
ApJ, 1909.02005