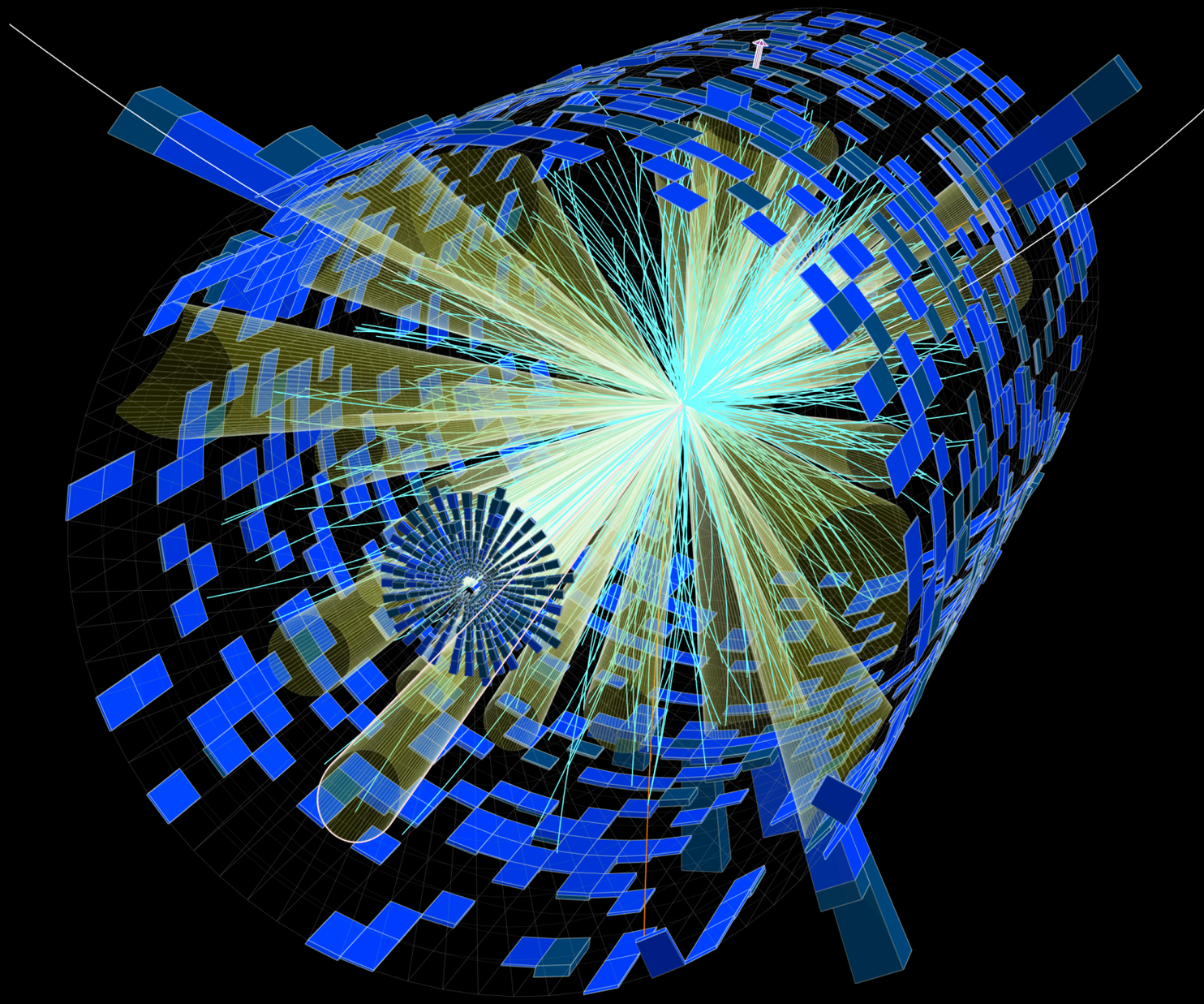
NYU Center for Data Science

Center for Cosmology and Particle Physics

EXPERIENCES WITH
DEEP LEARNING AND PARTICLE PHYSICS

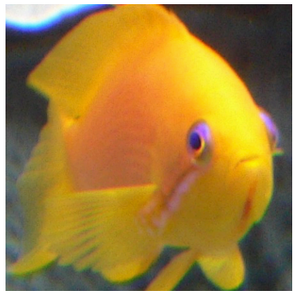


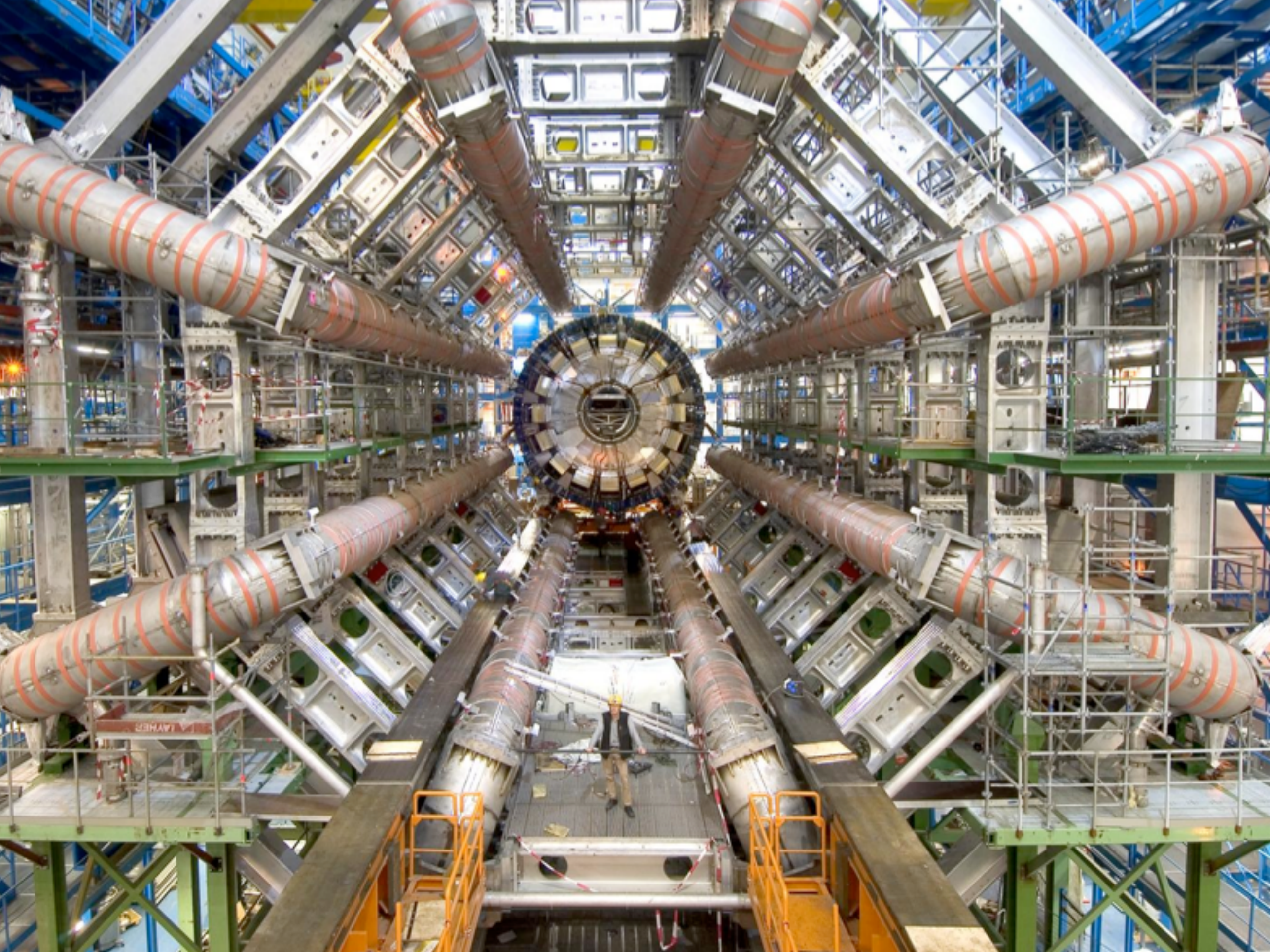
@KyleCranmer
New York University
Department of Physics
Center for Data Science
CILVR Lab

SUPPORT

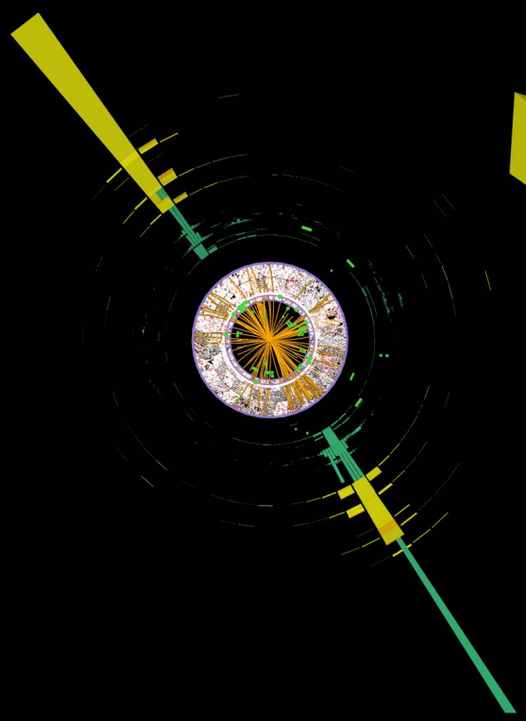
The SCAILFIN Project

scailfin.github.io





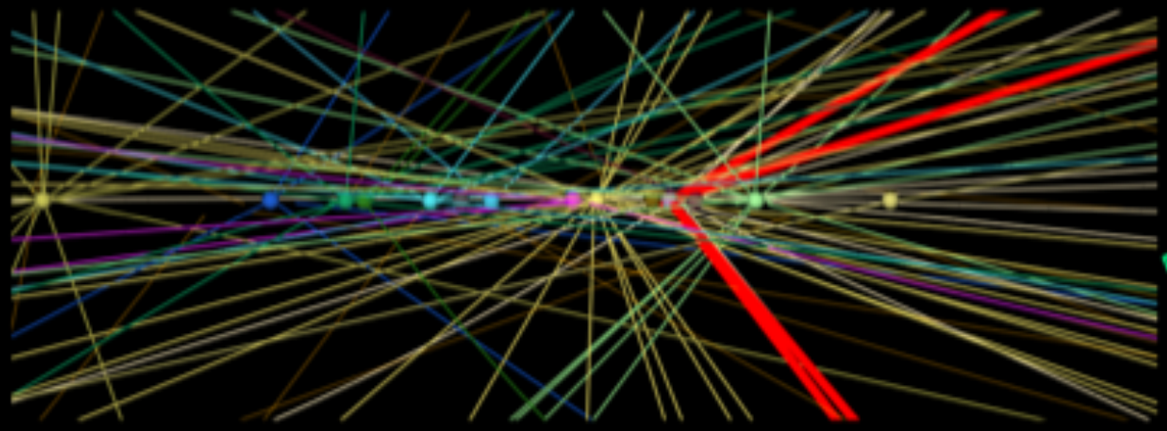
JETS



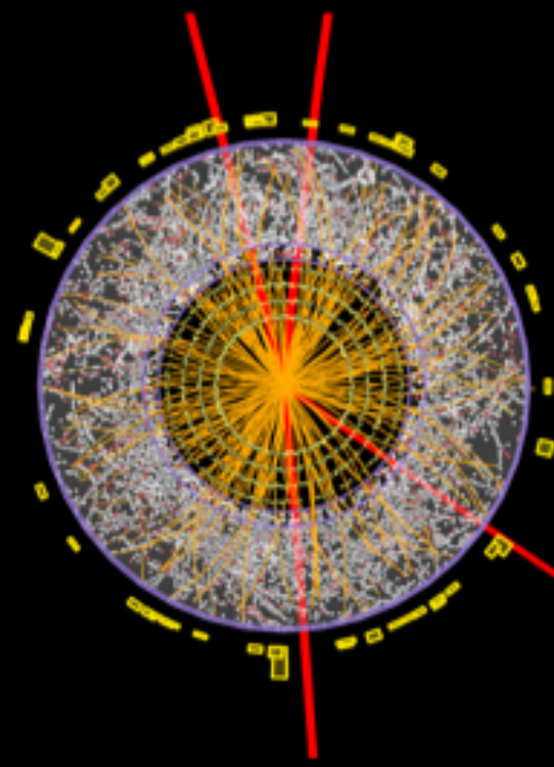
Run: 329716
Event: 857582452
2017-07-14 10:48:51 CEST

ATLAS
EXPERIMENT

THE HIGGS BOSON



ATLAS EXPERIMENT
http://atlas.ch



Run: 204769
Event: 71902630
Date: 2012-06-10
Time: 13:24:31 CEST

STATISTICAL FRAMING

forward modeling
generation
simulation

PREDICTION

θ
parameters of interest

p(x, z | θ, ν)

x
observed data
simulated data

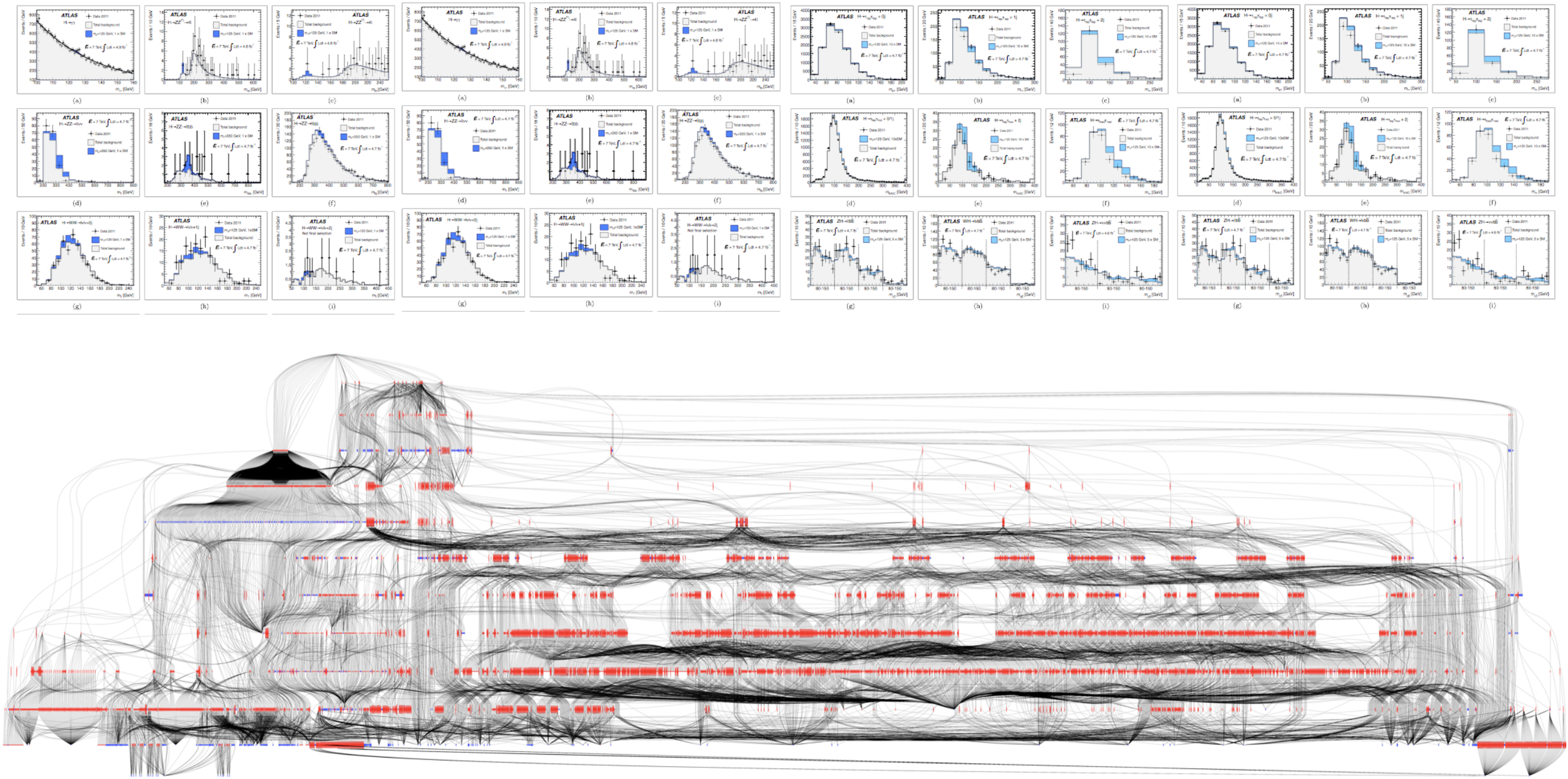
z
latent variables
Monte Carlo truth

ν
nuisance parameters

INFERENCE

inverse problem
measurement
parameter estimation

THE DISCOVERY OF THE HIGGS BOSON



$$\mathbf{f}_{\text{tot}}(\mathcal{D}_{\text{sim}}, \mathcal{G}|\boldsymbol{\alpha}) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right] \cdot \prod_{p \in \mathbb{S}} f_p(a_p|\alpha_p)$$

PREDICTIONS IN PARTICLE PHYSICS

$$
\begin{aligned}
\mathcal{L}_{SM} = & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
& + \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L \; + \; \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}} \\
& + \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}} \\
& + \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} \quad + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}
\end{aligned}
$$

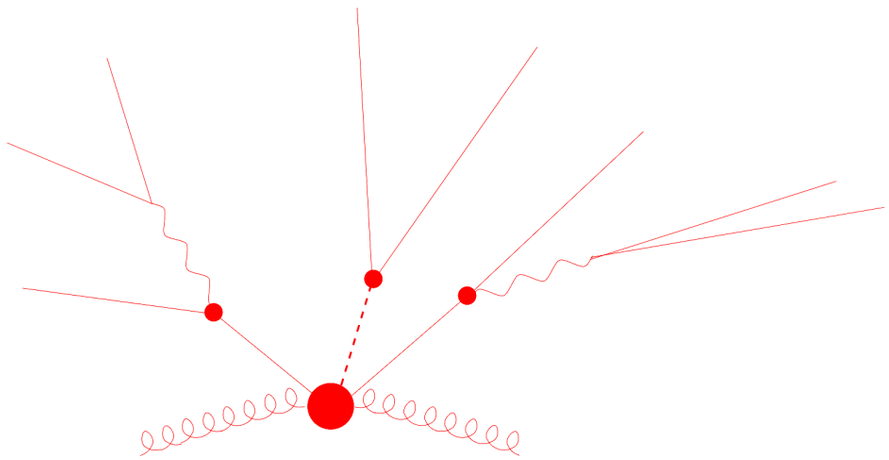
PREDICTIONS IN PARTICLE PHYSICS

$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$

$$+ \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}}$$

$$+ \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}}$$

$$+ \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$



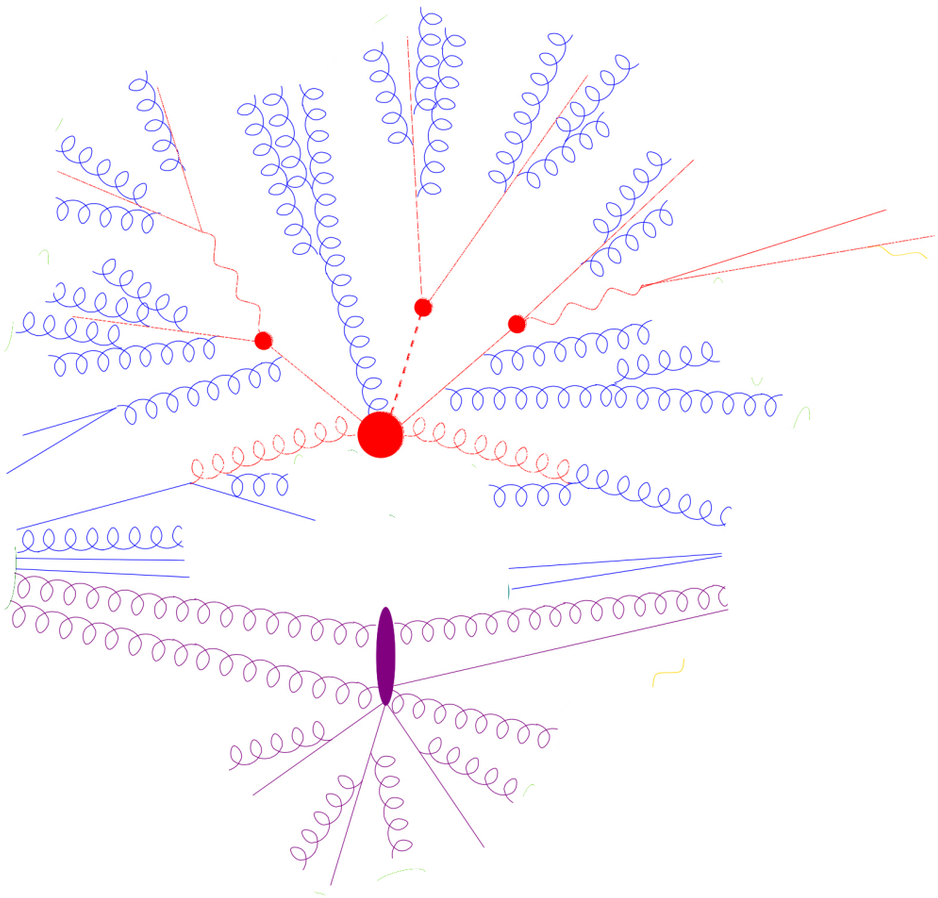
PREDICTIONS IN PARTICLE PHYSICS

$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$

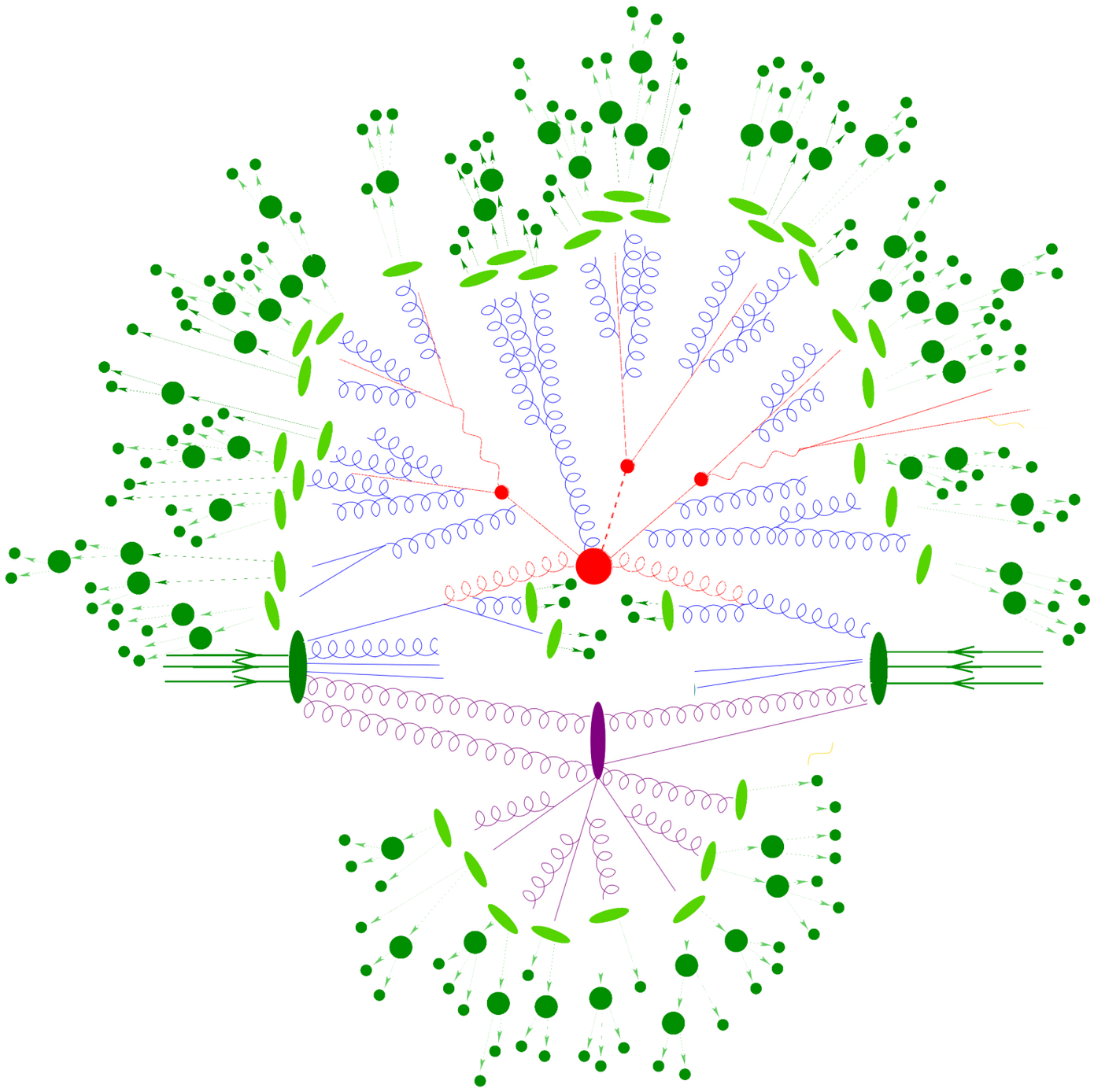
$$+ \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}}$$

$$+ \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}}$$

$$+ \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$



PREDICTIONS IN PARTICLE PHYSICS



$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$

$$+ \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}}$$

$$+ \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}}$$

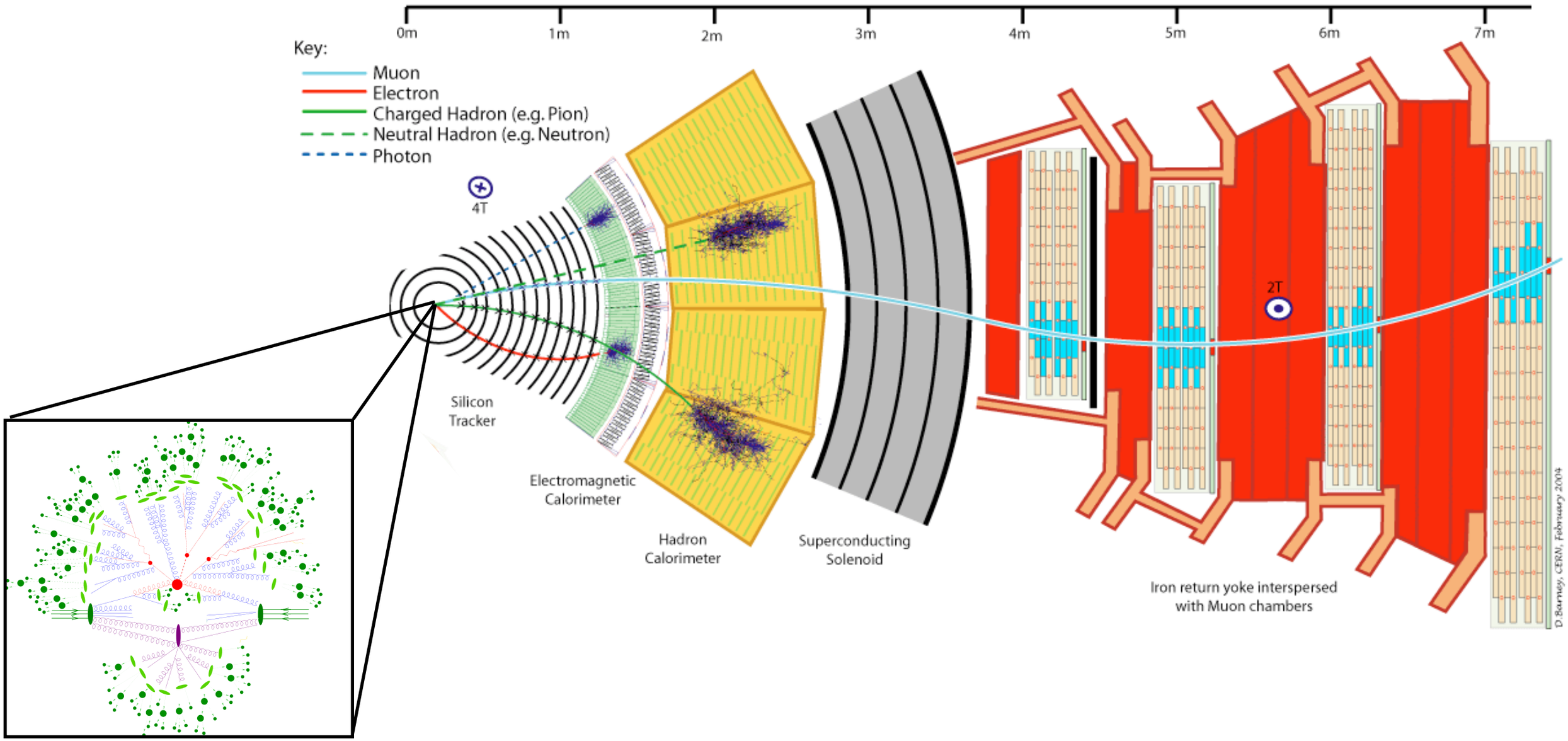
$$+ \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$

DETECTOR SIMULATION

Conceptually: Prob(detector response | particles)

Implementation: Monte Carlo integration over micro-physics

Consequence: evaluation of the likelihood is intractable



Conceptually: Prob(detector response | particles)

Implementation: Monte Carlo integration over micro-physics

Consequence: evaluation of the likelihood is intractable

This motivates a new class of algorithms for what is called **likelihood-free inference**, which only require ability to generate samples from the simulation in the "forward mode"

A COMMON THEME, A COMMON LANGUAGE

ABC

resources on approximate Bayesian computational methods

Search

Home

Home

This website keeps track of developments in approximate Bayesian computation (ABC) (a.k.a. likelihood-free), a class of computational statistical methods for Bayesian inference under intractable likelihoods. The site is meant to be a resource both for biologists and statisticians who want to learn more about ABC and related methods. Recent publications are under Publications 2012. A comprehensive list of publications can be found under Literature. If you are unfamiliar with ABC methods see the Introduction. Navigate using the menu to learn more.

ABC in Montreal | ABC in Montreal (2014)

ABC in Montreal

Approximate Bayesian computation (ABC) or likelihood-free (LF) methods have developed mostly beyond the radar of the machine learning community, but are important tools for a large and diverse segment of the scientific community. This is particularly true for systems and population biology, computational neuroscience, computer vision, healthcare sciences, but also many others.

Interaction between the ABC and machine learning community has recently started and contributed to important advances. In general, however, there is still significant room for more intense interaction and collaboration. Our workshop aims at being a place for this to happen.

Markov chain Monte Carlo without likelihoods

Paul Marjoram*, John Molitor*, Vincent Plagnol†, and Simon Tavaré†‡

*Biostatistics Division, Department of Preventive Medicine, Keck School of Medicine, and †Molecular and Computational Biology, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089

Communicated by Michael S. Waterman, University of Southern California, Los Angeles, CA, October 24, 2003 (received for review June 20, 2003)

Many stochastic simulation approaches for generating observations from a posterior distribution depend on knowing a likelihood function. However, for many complex probability models, such likelihoods are either impossible or computationally prohibitive to obtain. Here we present a Markov chain Monte Carlo method for generating observations from a posterior distribution without the use of likelihoods. It can also be used in frequentist applications, in particular for maximum-likelihood estimation. The approach is illustrated by an example of ancestral inference in population genetics. A number of open problems are highlighted in the discussion.

One of the basic problems in Bayesian statistics is the computation of posterior distributions. We imagine data $\mathcal{D}$ generated from a model $\mathcal{M}$ determined by parameters θ, the prior density of which is denoted by $\pi(\theta)$. We assume unless otherwise stated that the data are discrete. The posterior distribution of interest is $f(\theta|\mathcal{D})$, which is given by

$$f(\theta|\mathcal{D}) = \mathbb{P}(\mathcal{D}|\theta)\pi(\theta)/\mathbb{P}(\mathcal{D}), \qquad [1]$$

where $\mathbb{P}(\mathcal{D}) = \int \mathbb{P}(\mathcal{D}|\theta)\pi(\theta)d\theta$ is the normalizing constant.

In most scientific contexts, explicit formulae for such posterior densities are few and far between, and we usually resort to stochastic simulation to generate observations from f. Perhaps the simplest approach for this is the rejection method:

A1. Generate θ from $\pi(\cdot)$.
A2. Accept θ with probability $h = \mathbb{P}(\mathcal{D}|\theta)$; return to *A1*.

of ε therefore reflects a tension between computability and accuracy. The method is still honest in that, for a given ρ and ε, we are generating independent and identically distributed observations from $f(\theta|\rho(\mathcal{D}, \mathcal{D}') \leq \varepsilon)$.

When $\mathcal{D}$ is high-dimensional or continuous, this approach can be impractical as well, and then the comparison of $\mathcal{D}'$ with $\mathcal{D}$ can be made by using lower-dimensional summaries of the data. The motivation for this approach is that if the set of statistics $S = (S_1, \ldots, S_p)$ is sufficient for θ, in that $\mathbb{P}(\mathcal{D}|S, \theta)$ is independent of θ, then $f(\theta|\mathcal{D}) = f(\theta|S)$. The normalizing constant $\mathbb{P}(S)$ is typically larger than $\mathbb{P}(\mathcal{D})$, resulting in more acceptances. In practice it will be hard, if not impossible, to identity a suitable set of sufficient statistics, and we then might resort to a more heuristic approach. Thus we seek to use knowledge of the particular problem at hand to suggest summary statistics that capture information about θ. With these statistics in hand, we have the following approximate Bayesian computation scheme for data $\mathcal{D}$ summarized by S:

D1. Generate θ from $\pi(\cdot)$.
D2. Simulate $\mathcal{D}'$ from stochastic model $\mathcal{M}$ with parameter θ, and compute the corresponding statistics S'.
D3. Calculate the distance $\rho(S, S')$ between S and S'.
D4. Accept θ if $\rho \leq \varepsilon$, and return to *D1*.

There are several advantages to these rejection methods, among them the fact that they are usually easy to code, they generate independent observations (and thus can use embarrassingly parallel computation), and they readily provide estimates of Baves factors that can be used for model com-

ABC

Markov chain Monte Carlo without likelihoods

Paul Marjoram*, John Molitor*, Vincent Plagnol†, and Simon Tavaré†‡

*Biostatistics Division, Department of Preventive Medicine, Keck School of Medicine, and †Molecular and Computational Biology, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089

D1. Generate θ from $\pi(\cdot)$.
D2. Simulate $\mathcal{D}'$ from stochastic model $\mathcal{M}$ with parameter θ, and compute the corresponding statistics S'.
D3. Calculate the distance $\rho(S, S')$ between S and S'.
D4. Accept θ if $\rho \leq \varepsilon$, and return to *D1*.

discussion.

One of the basic problems in Bayesian statistics is the computation of posterior distributions. We imagine data $\mathcal{D}$ generated from a model $\mathcal{M}$ determined by parameters θ, the prior density of which is denoted by $\pi(\theta)$. We assume unless otherwise stated that the data are discrete. The posterior distribution of interest is $f(\theta|\mathcal{D})$, which is given by

$$f(\theta|\mathcal{D}) = \mathbb{P}(\mathcal{D}|\theta)\pi(\theta)/\mathbb{P}(\mathcal{D}), \quad [1]$$

where $\mathbb{P}(\mathcal{D}) = \int \mathbb{P}(\mathcal{D}|\theta)\pi(\theta)d\theta$ is the normalizing constant.

In most scientific contexts, explicit formulae for such posterior densities are few and far between, and we usually resort to stochastic simulation to generate observations from f. Perhaps the simplest approach for this is the rejection method:

A1. Generate θ from $\pi(\cdot)$.
A2. Accept θ with probability $h = \mathbb{P}(\mathcal{D}|\theta)$; return to *A1*.

typically larger than $\mathbb{P}(\mathcal{D})$, resulting in more acceptances. In practice it will be hard, if not impossible, to identity a suitable set of sufficient statistics, and we then might resort to a more heuristic approach. Thus we seek to use knowledge of the particular problem at hand to suggest summary statistics that capture information about θ. With these statistics in hand, we have the following approximate Bayesian computation scheme for data $\mathcal{D}$ summarized by S:

D1. Generate θ from $\pi(\cdot)$.
D2. Simulate $\mathcal{D}'$ from stochastic model $\mathcal{M}$ with parameter θ, and compute the corresponding statistics S'.
D3. Calculate the distance $\rho(S, S')$ between S and S'.
D4. Accept θ if $\rho \leq \varepsilon$, and return to *D1*.

There are several advantages to these rejection methods, among them the fact that they are usually easy to code, they generate independent observations (and thus can use embarrassingly parallel computation), and they readily provide estimates of Bayes factors that can be used for model com-

Markov chain Monte Carlo without likelihoods

Paul Marjoram*, John Molitor*, Vincent Plagnol†, and Simon Tavaré†‡

*Biostatistics Division, Department of Preventive Medicine, Keck School of Medicine, and †Molecular and Computational Biology, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089

Communicated by Michael S. Waterman, University of Southern California, Los Angeles, CA, October 24, 2003 (received for review June 20, 2003)

Many stochastic simulation approaches for generating observations from a posterior distribution depend on knowing a likelihood function. However, for many complex probability models, such likelihoods are either impossible or computationally prohibitive to obtain. Here we present a Markov chain Monte Carlo method for generating observations from a posterior distribution without the use of likelihoods. It can also be used in frequentist applications, in particular for maximum-likelihood estimation. The approach is illustrated by an example of ancestral inference in population genetics. A number of open problems are highlighted in the discussion.

One of the basic problems in Bayesian statistics is the computation of posterior distributions. We imagine data $\mathcal{D}$ generated from a model $\mathcal{M}$ determined by parameters θ, the prior density of which is denoted by $\pi(\theta)$. We assume unless otherwise stated that the data are discrete. The posterior distribution of interest is $f(\theta|\mathcal{D})$, which is given by

A1. Generate θ from $\pi(\cdot)$.
A2. Accept θ with probability $h = \mathbb{P}(\mathcal{D}|\theta)$; return to *A1*.

of ε therefore reflects a tension between computability and accuracy. The method is still honest in that, for a given ρ and ε, we are generating independent and identically distributed observations from $f(\theta|\rho(\mathcal{D}, \mathcal{D}') \leq \varepsilon)$.

When $\mathcal{D}$ is high-dimensional or continuous, this approach can be impractical as well, and then the comparison of $\mathcal{D}'$ with $\mathcal{D}$ can be made by using lower-dimensional summaries of the data. The motivation for this approach is that if the set of statistics $S = (S_1, \ldots, S_p)$ is sufficient for θ, in that $\mathbb{P}(\mathcal{D}|S, \theta)$ is independent of θ, then $f(\theta|\mathcal{D}) = f(\theta|S)$. The normalizing constant $\mathbb{P}(S)$ is typically larger than $\mathbb{P}(\mathcal{D})$, resulting in more acceptances. In practice it will be hard, if not impossible, to identity a suitable set of sufficient statistics, and we then might resort to a more heuristic approach. Thus we seek to use knowledge of the particular problem at hand to suggest summary statistics that capture information about θ. With these statistics in hand, we have the following approximate Bayesian computation scheme for data $\mathcal{D}$ summarized by S:

When $\mathcal{D}$ is high-dimensional or continuous, this approach can be impractical as well, and then the comparison of $\mathcal{D}'$ with $\mathcal{D}$ can be made by using lower-dimensional summaries of the data. The

θ, and

hods,

, they

generate independent observations (and thus can use embarrassingly parallel computation), and they readily provide estimates of Baves factors that can be used for model com-

Markov chain Monte Carlo without likelihoods

Paul Marjoram*, John Molitor*, Vincent Plagnol†, and Simon Tavaré†‡

*Biostatistics Division, Department of Preventive Medicine, Keck School of Medicine, and †Molecular and Computational Biology, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089

Communicated by Michael S. Waterman, University of Southern California, Los Angeles, CA, October 24, 2003 (received for review June 20, 2003)

Many stochastic simulation approaches for generating observations from a posterior distribution depend on knowing a likelihood function. However, for many complex probability models, such likelihoods are either impossible or computationally prohibitive to obtain. Here we present a Markov chain Monte Carlo method for generating observations from a posterior distribution without the use of likelihoods. It can also be used in frequentist applications, in particular for maximum-likelihood estimation. The approach is illustrated by an example of ancestral inference in population genetics. A number of open problems are highlighted in the discussion.

One of the basic problems in Bayesian statistics is the computation of posterior distributions. We imagine data $\mathcal{D}$ generated from a model $\mathcal{M}$ determined by parameters θ, the prior density of which is denoted by $\pi(\theta)$. We assume unless otherwise stated that the data are discrete. The posterior distribution of interest is $f(\theta|\mathcal{D})$, which is given by

A1. Generate θ from $\pi(\cdot)$.
A2. Accept θ with probability $h = \mathbb{P}(\mathcal{D}|\theta)$; return to *A1*.

of ε therefore reflects a tension between computability and accuracy. The method is still honest in that, for a given ρ and ε, we are generating independent and identically distributed observations from $f(\theta|\rho(\mathcal{D}, \mathcal{D}') \leq \varepsilon)$.

When $\mathcal{D}$ is high-dimensional or continuous, this approach can be impractical as well, and then the comparison of $\mathcal{D}'$ with $\mathcal{D}$ can be made by using lower-dimensional summaries of the data. The motivation for this approach is that if the set of statistics $S = (S_1, \ldots, S_p)$ is sufficient for θ, in that $\mathbb{P}(\mathcal{D}|S, \theta)$ is independent of θ, then $f(\theta|\mathcal{D}) = f(\theta|S)$. The normalizing constant $\mathbb{P}(S)$ is typically larger than $\mathbb{P}(\mathcal{D})$, resulting in more acceptances. In practice it will be hard, if not impossible, to identity a suitable set of sufficient statistics, and we then might resort to a more heuristic approach. Thus we seek to use knowledge of the particular problem at hand to suggest summary statistics that capture information about θ. With these statistics in hand, we have the following approximate Bayesian computation scheme for data $\mathcal{D}$ summarized by S:

practice it will be hard, if not impossible, to identity a suitable set of sufficient statistics, and we then might resort to a more heuristic approach.

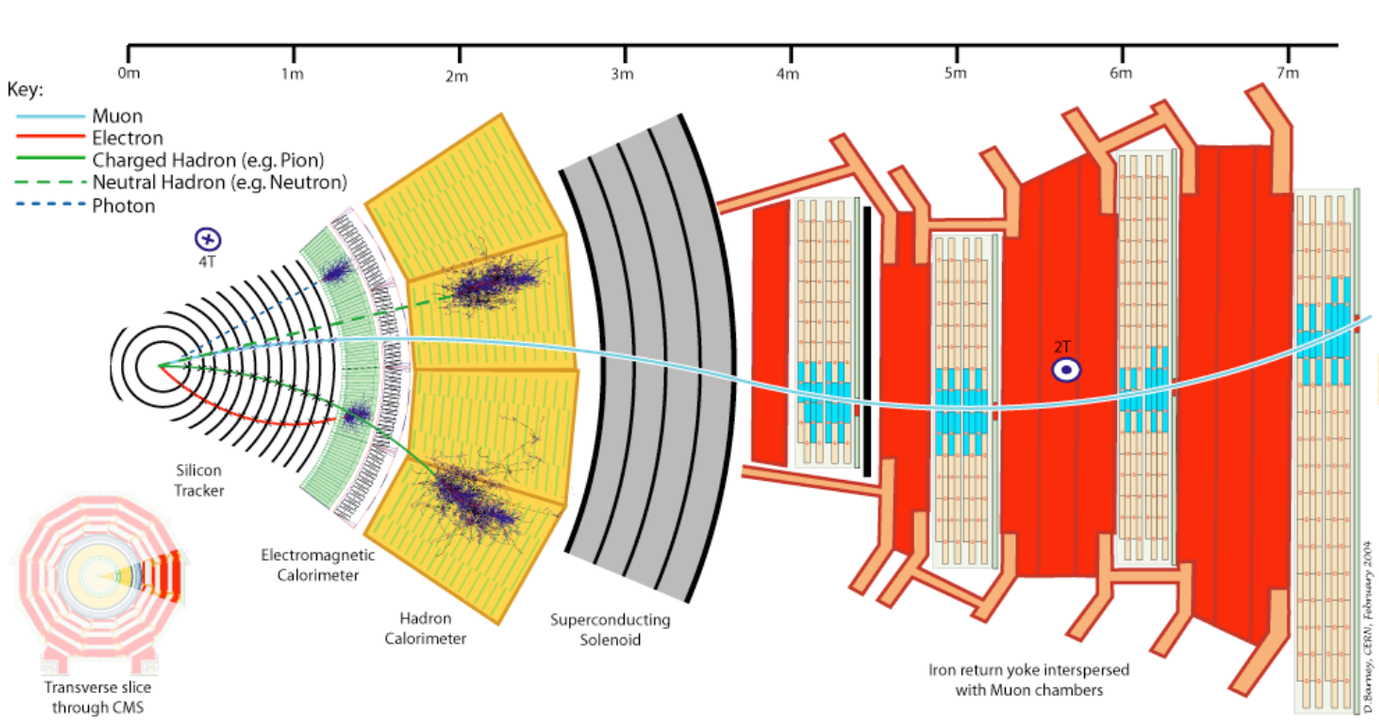
θ, and

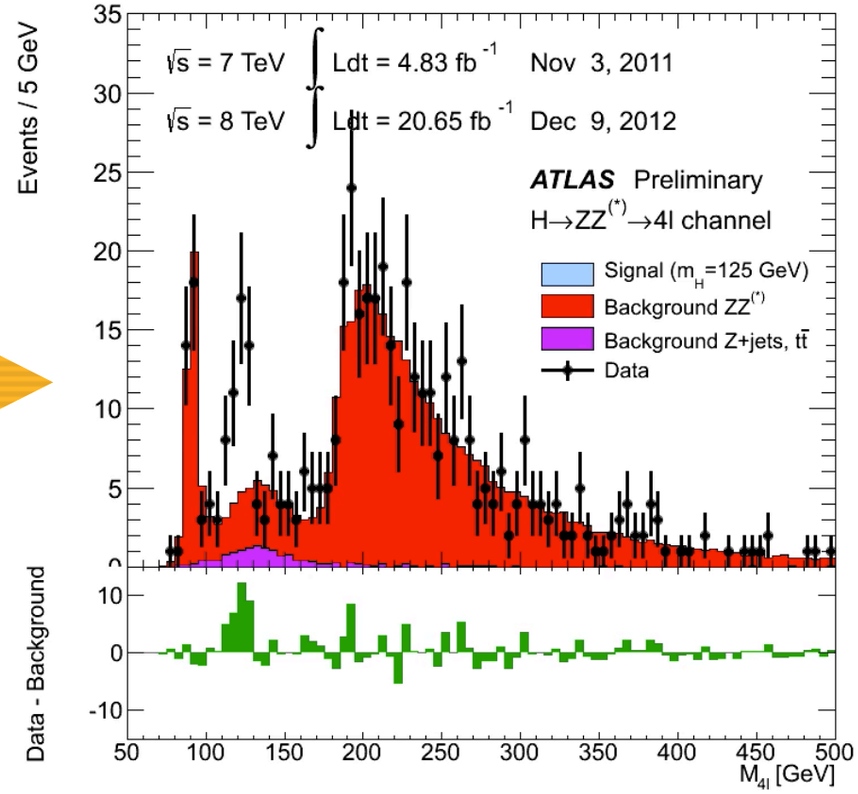
hods,
, they
generate independent observations (and thus can use embarrassingly parallel computation), and they readily provide estimates of Baves factors that can be used for model com-

10^8 SENSORS → 1 REAL-VALUED QUANTITY

Most measurements and searches for new particles at the LHC are based on the distribution of a single summary statistic

- choosing a good summary statistic (feature engineering) is a task for a skilled physicist and tailored to the goal of measurement or new particle search
- likelihood $p(x|\theta)$ **approximated** using histograms (univariate density estimation)

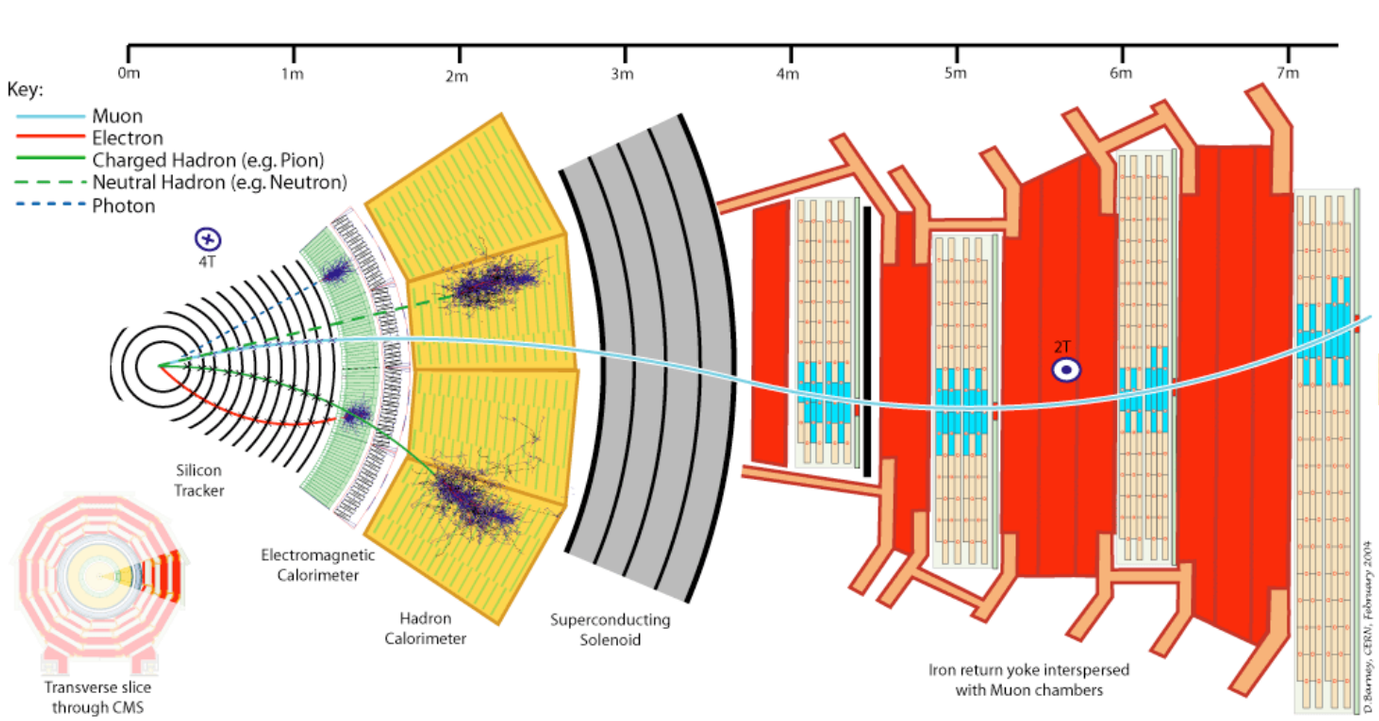


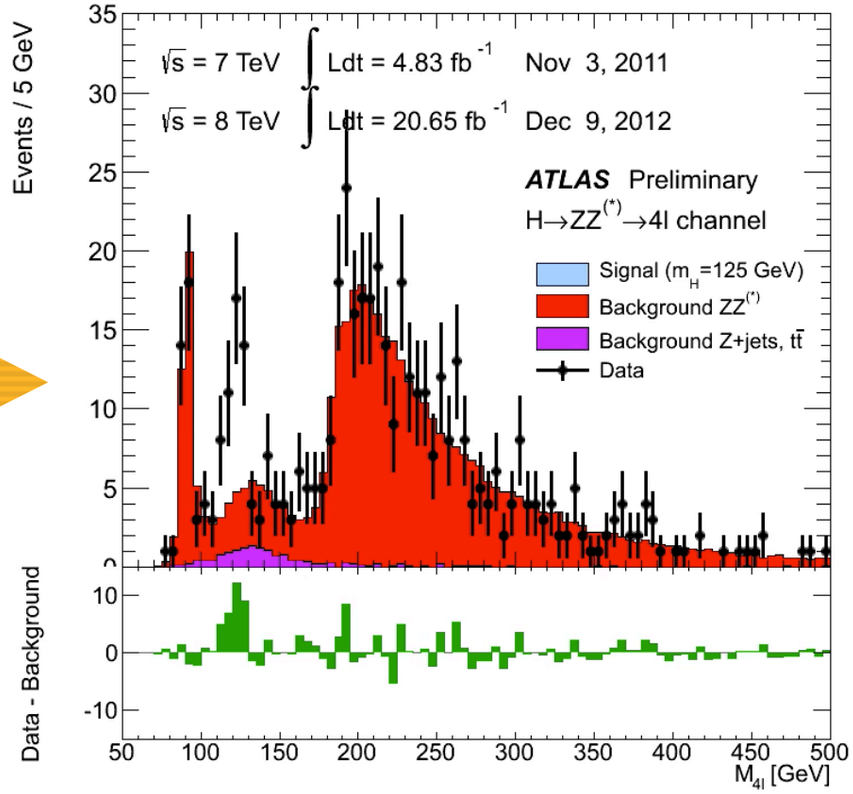


10^8 SENSORS → 1 REAL-VALUED QUANTITY

Most measurements and searches for new particles at the LHC are based on the distribution of a single summary statistic

- choosing a good summary statistic (feature engineering) is a task for a skilled physicist and tailored to the goal of measurement or new particle search
- likelihood $p(x|\theta)$ **approximated** using histograms (univariate density estimation)



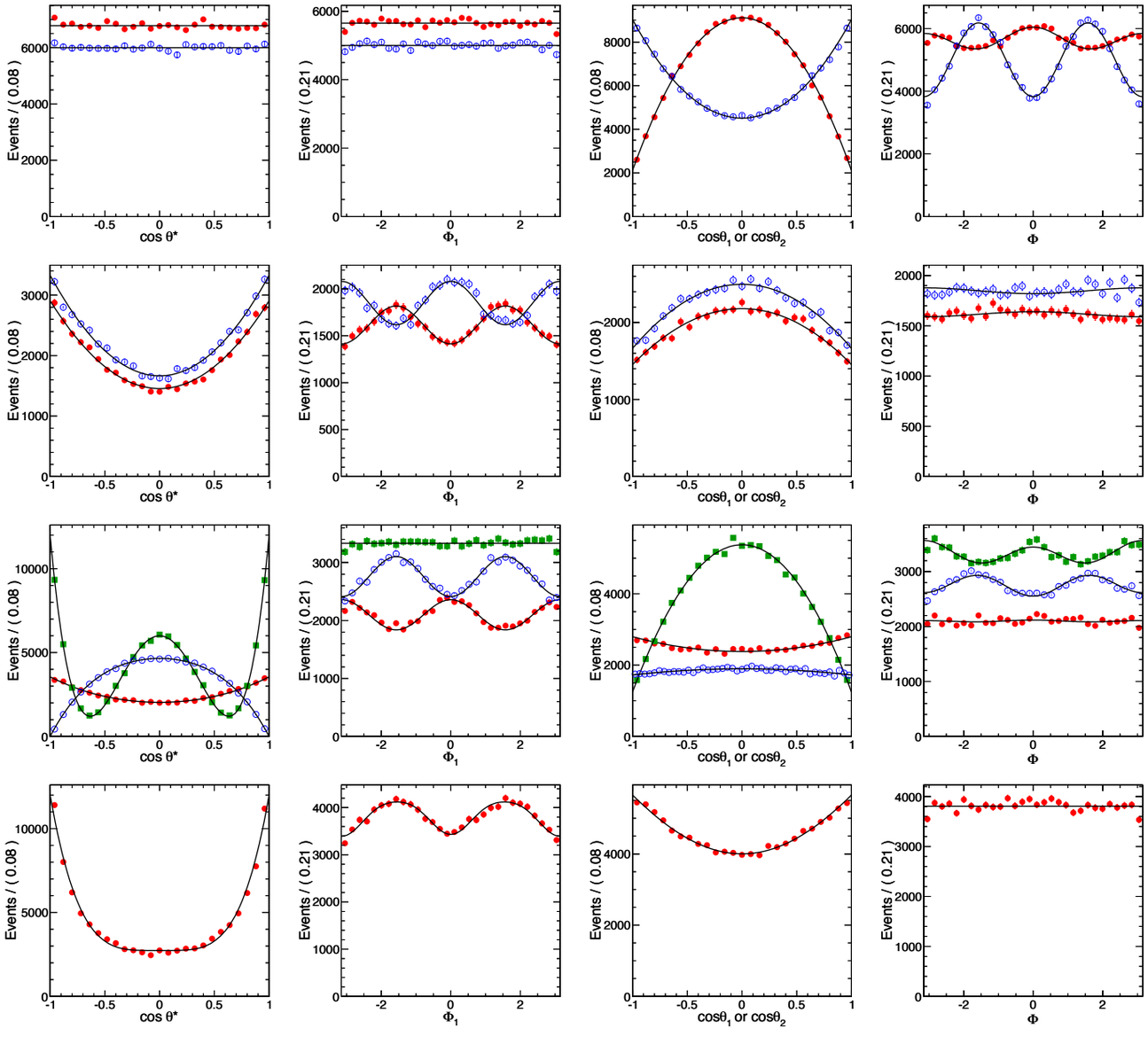


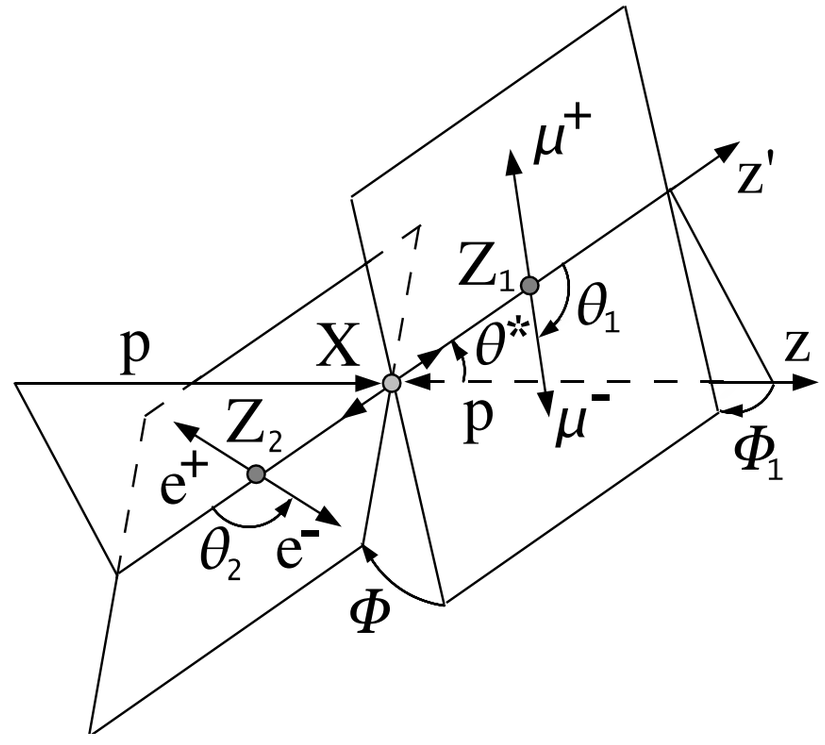
This doesn't scale if x is high dimensional!

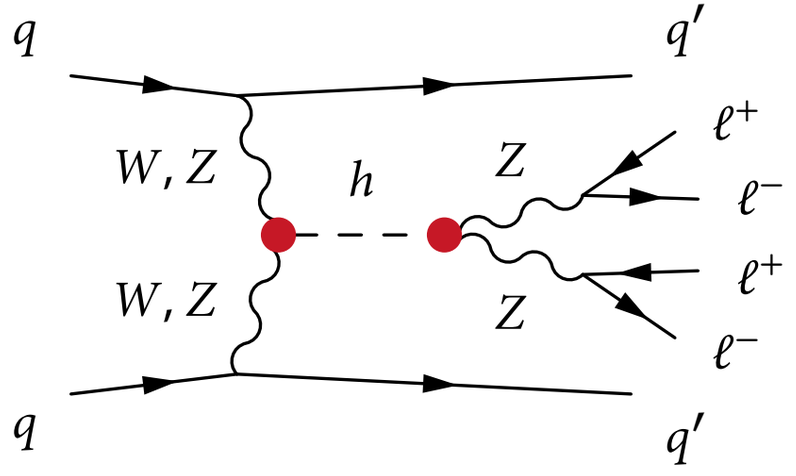
HIGH DIMENSIONAL EXAMPLE

When looking for deviations from the standard model Higgs, we would like to look at all sorts of kinematic correlations

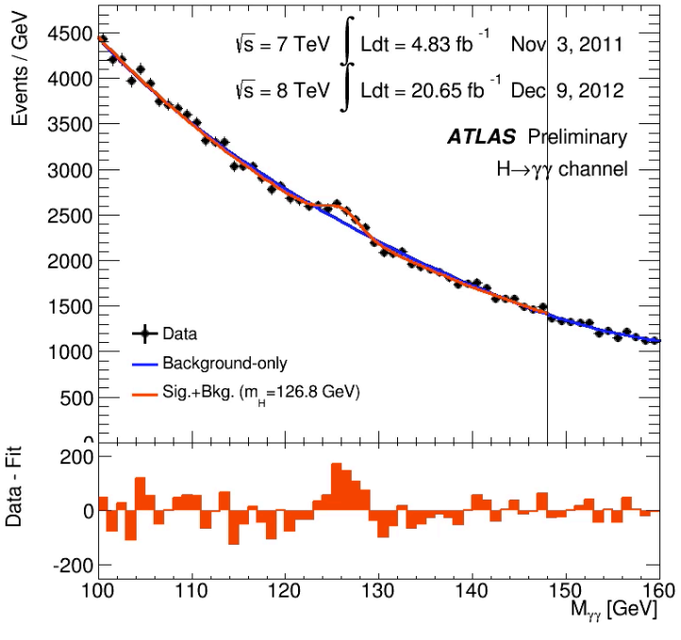
- thus each observation **x** is high-dimensional

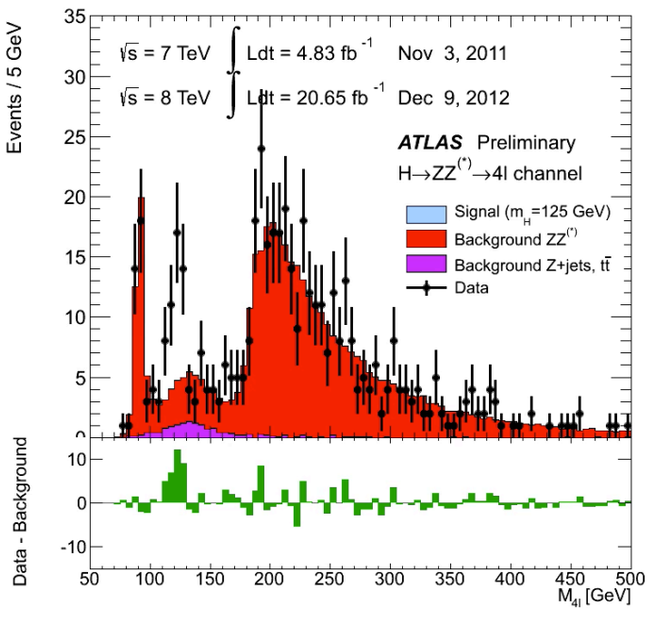






JULY 2012





ImageNet Classification with Deep Convolutional Neural Networks

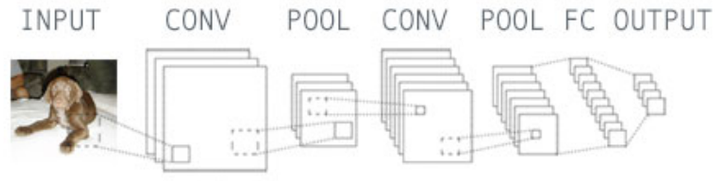
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

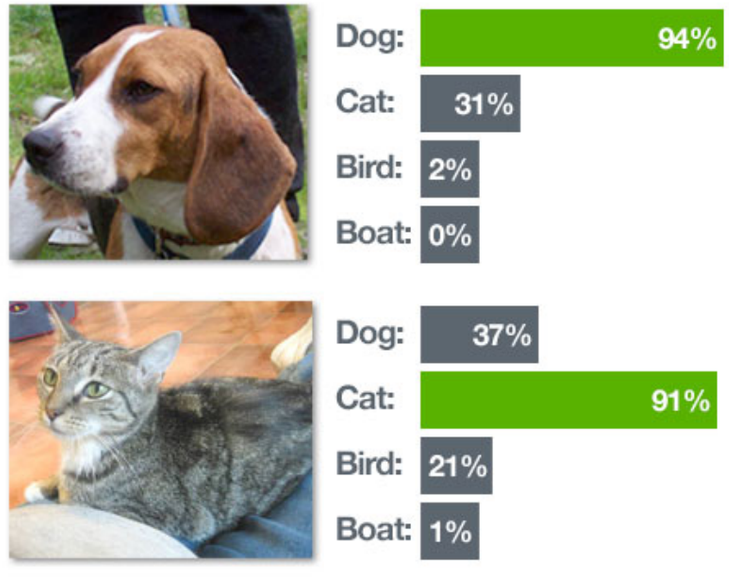
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

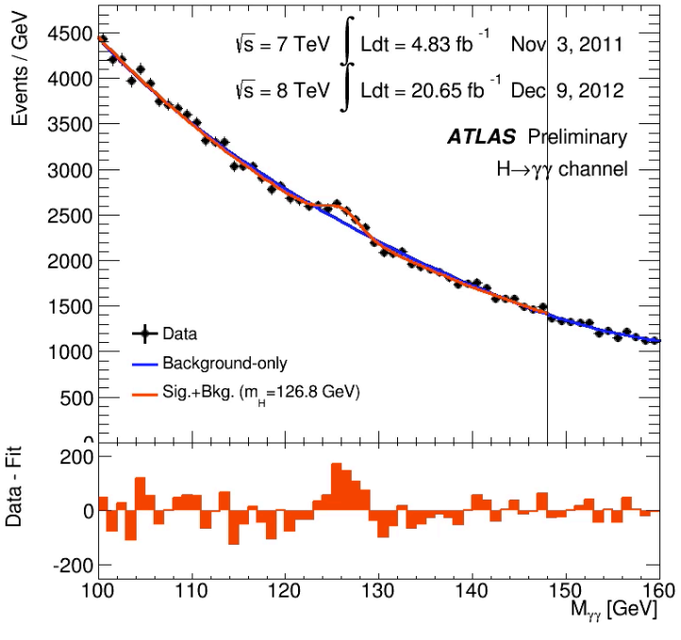
Abstract

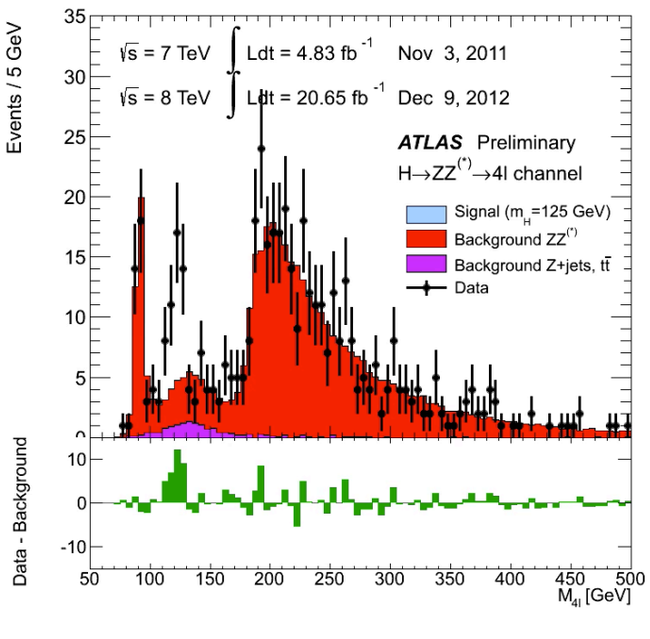
We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.





JULY 2012





ImageNet Classification with Deep Convolutional Neural Networks

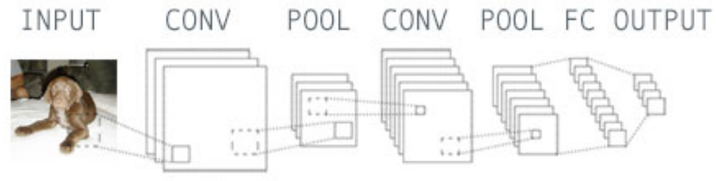
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

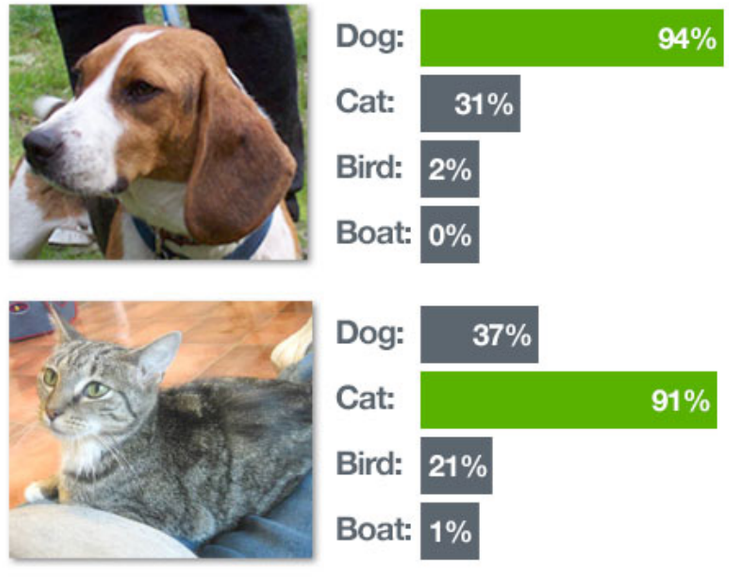
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.





ICML 2017 Workshop on Implicit Models

Workshop Aims

Probabilistic models are an important tool in machine learning. They form the basis for models that generate realistic data, uncover hidden structure, and make predictions. Traditionally, probabilistic models in machine learning have focused on prescribed models. Prescribed models specify a joint density over observed and hidden variables that can be easily evaluated. The requirement of a tractable density simplifies their learning but limits their flexibility --- several real world phenomena are better described by simulators that do not admit a tractable density. Probabilistic models defined only via the simulations they produce are called implicit models.

Arguably starting with generative adversarial networks, research on implicit models in machine learning has exploded in recent years. This workshop's aim is to foster a discussion around the recent developments and future directions of implicit models.

Implicit models have many applications. They are used in ecology where models simulate animal populations over time; they are used in phylogeny, where simulations produce hypothetical ancestry trees; they are used in physics to generate particle simulations for high energy processes. Recently, implicit models have been used to improve the state-of-the-art in image and content generation. Part of the workshop's focus is to discuss the commonalities among applications of implicit models.

Of particular interest at this workshop is to unite fields that work on implicit models. For example:

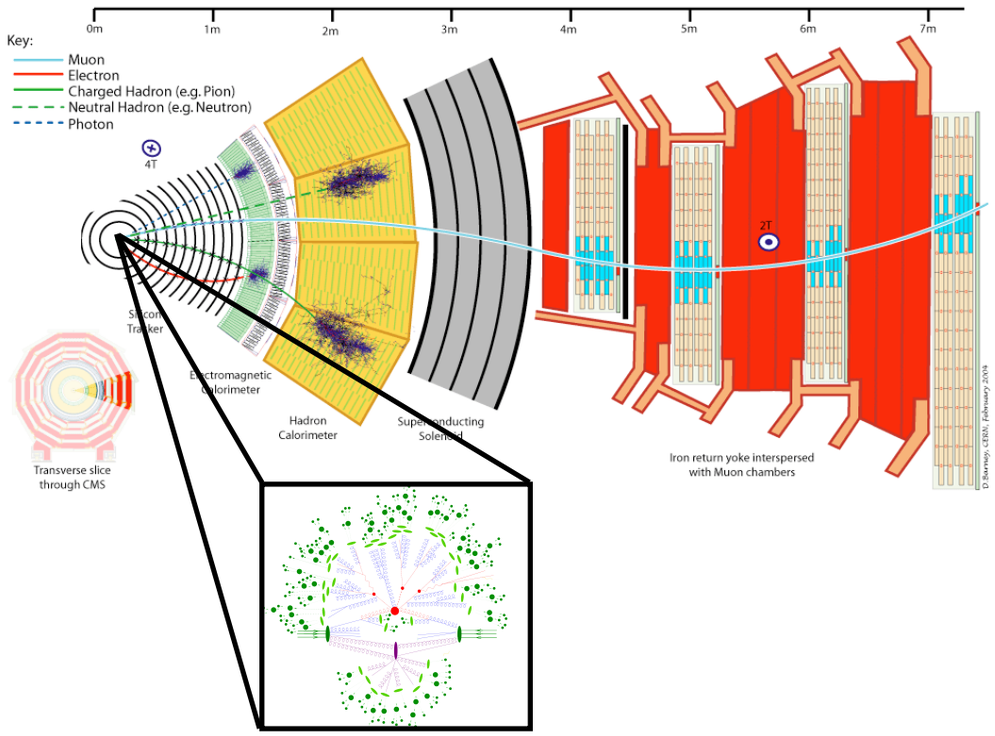
- **Generative adversarial networks** (a NIPS 2016 workshop) are implicit models with an adversarial training scheme.
- Recent advances in **variational inference** (a NIPS 2015 and 2016 workshop) have leveraged implicit models for more accurate approximations.
- **Approximate Bayesian computation** (a NIPS 2015 workshop) focuses on posterior inference for models with implicit likelihoods.
- Learning implicit models is deeply connected to **two sample testing, density ratio and density difference** estimation.

We hope to bring together these different views on implicit models, identifying their core challenges and combining their innovations.

TWO APPROACHES TO LIKELIHOOD FREE INFERENCE

Use simulator

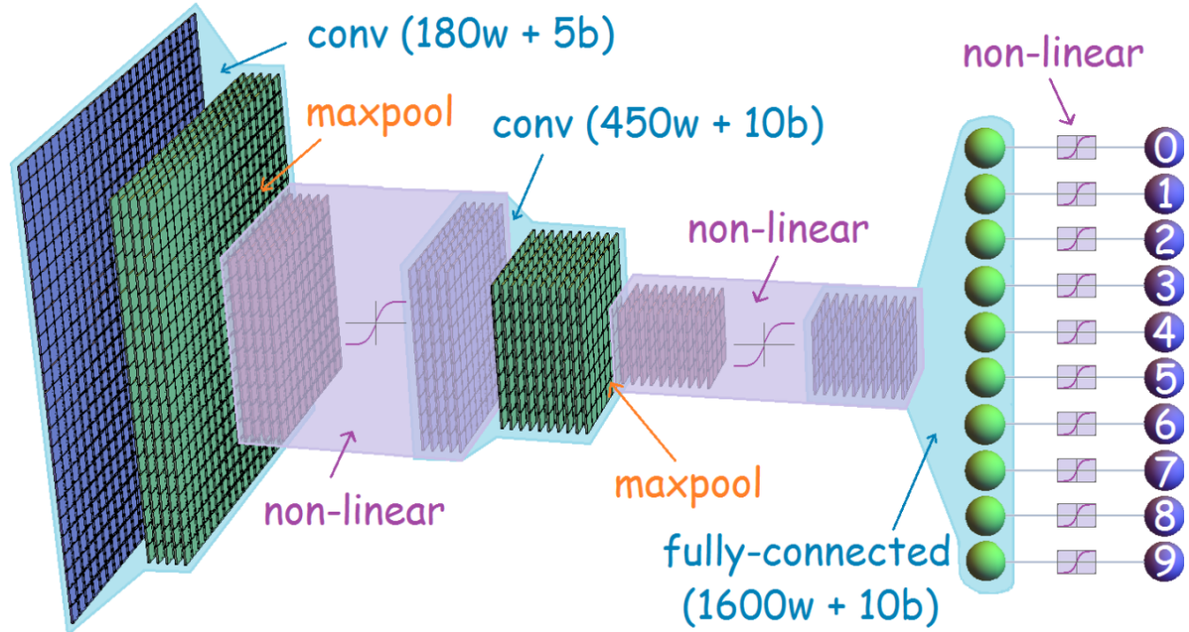
(much more efficiently)



- Approximate Bayesian Computation (ABC)
- Probabilistic Programming
- Adversarial Variational Optimization (AVO)

Learn simulator

(with deep learning)

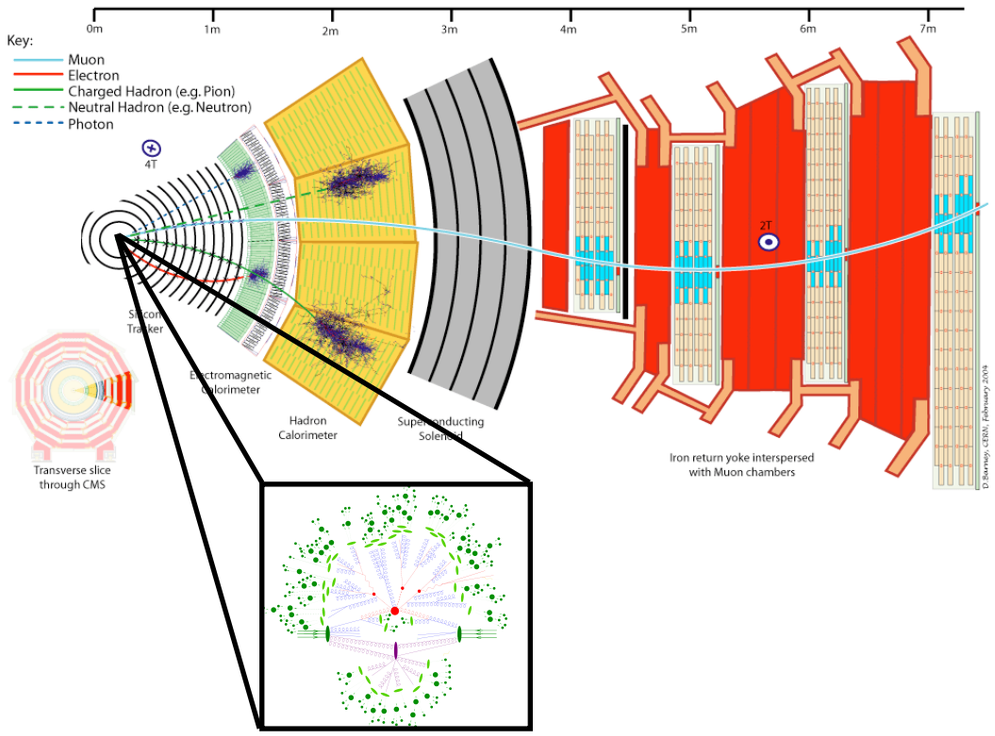


- Generative Adversarial Networks (GANs), Variational Auto-Encoders (VAE)
- Likelihood ratio from classifiers (CARL)
- Autogregressive models, Normalizing Flows

[image credit: A.P. Goucher]

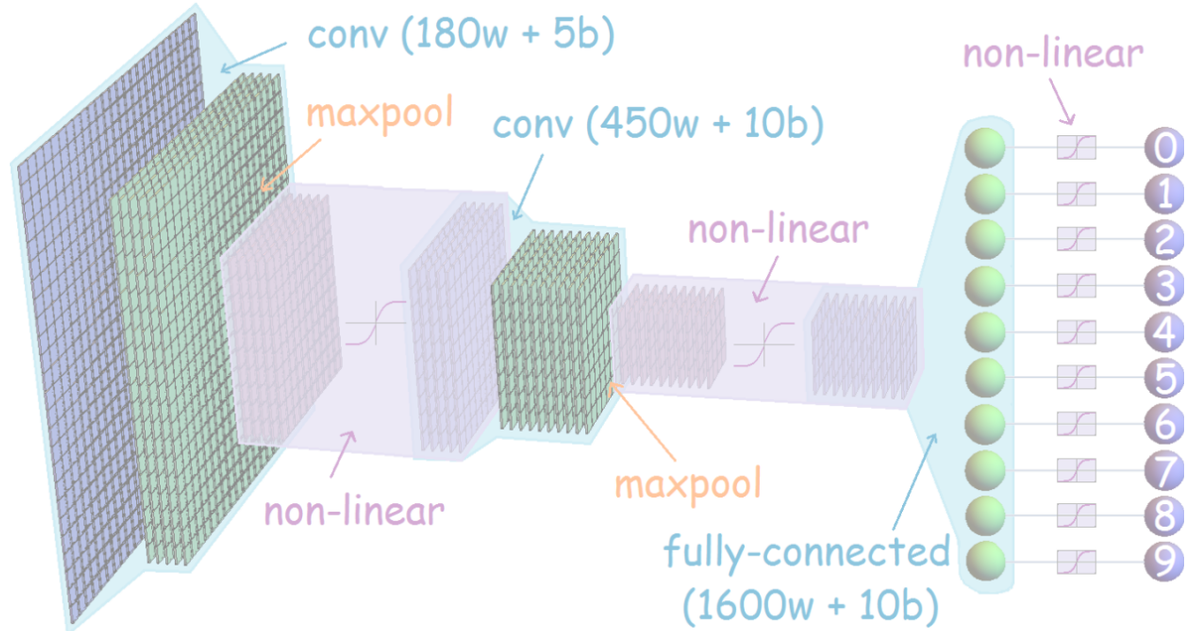
TWO APPROACHES TO LIKELIHOOD FREE INFERENCE

Use simulator
(much more efficiently)



- Approximate Bayesian Computation (ABC)
- Probabilistic Programming
- Adversarial Variational Optimization (AVO)

Learn simulator
(with deep learning)

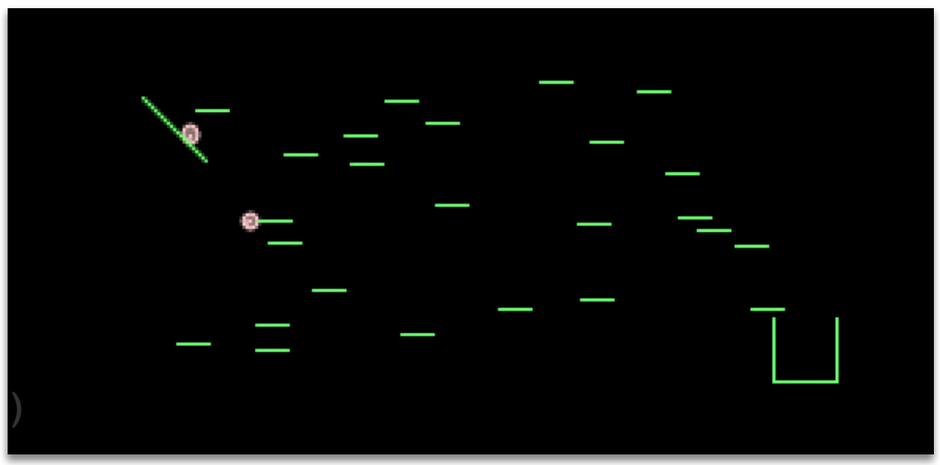


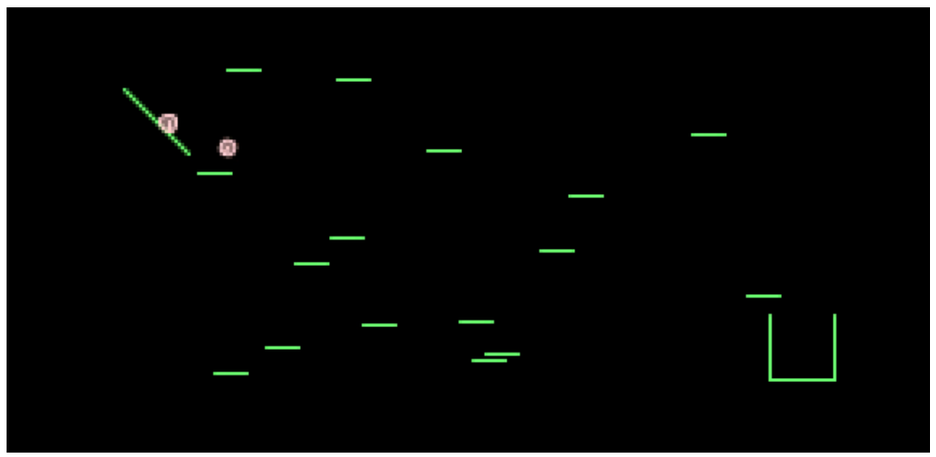
- Generative Adversarial Networks (GANs), Variational Auto-Encoders (VAE)
- Likelihood ratio from classifiers (CARL)
- Autogregressive models, Normalizing Flows

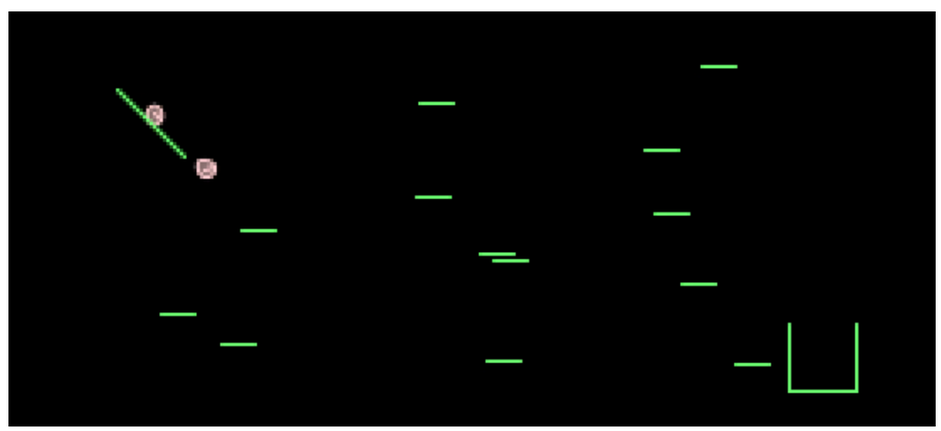
[image credit: A.P. Goucher]

PROBABILISTIC PROGRAMMING EXAMPLE

```
(defquery arrange-bumpers []
    (let [number-of-bumpers (sample (poisson 20))
          bumpydist (uniform-continuous 0 10)
          bumpxdist (uniform-continuous -5 14)
          bumper-positions (repeatedly
                            number-of-bumpers
                            #(vector (sample bumpxdist)
                                     (sample bumpydist)))

          ;; code to simulate the world
          world (create-world bumper-positions)
          end-world (simulate-world world)
          balls (:balls end-world)

          ;; how many balls entered the box?
          num-balls-in-box (balls-in-box end-world)]

      {:balls balls
       :num-balls-in-box num-balls-in-box
       :bumper-positions bumper-positions}))
```





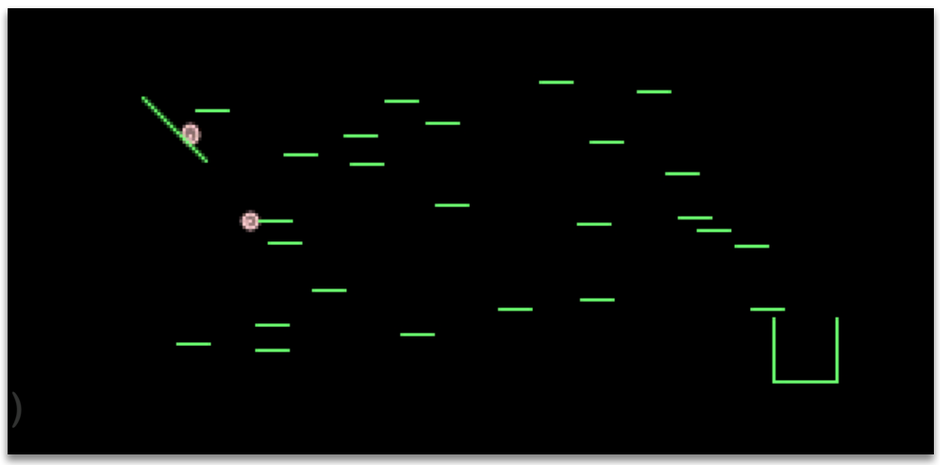


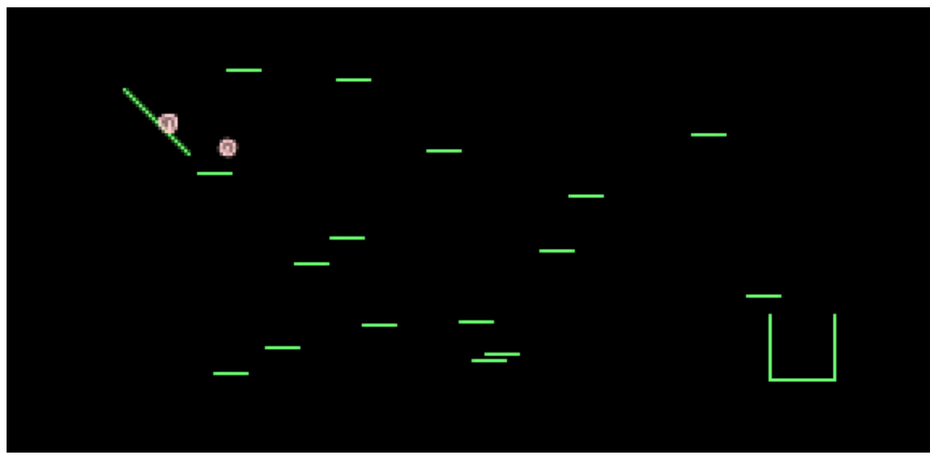
3 examples generated from simulator

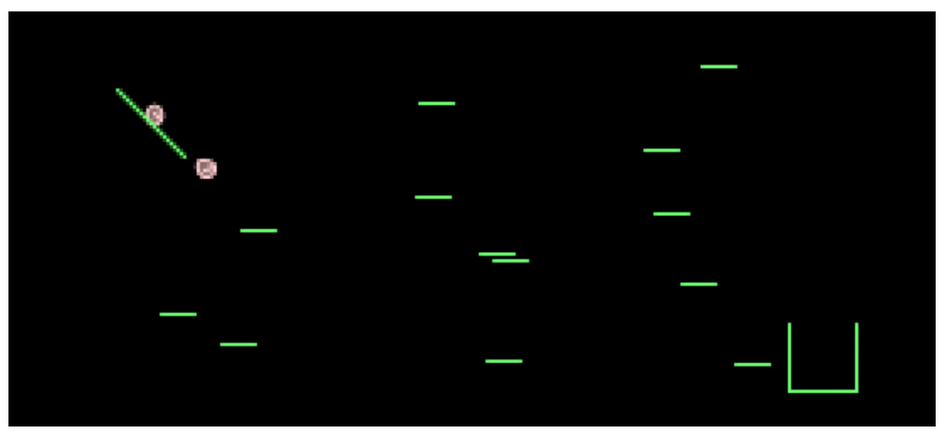
[slides, Frank Wood]

PROBABILISTIC PROGRAMMING EXAMPLE

```
(defquery arrange-bumpers []
    (let [number-of-bumpers (sample (poisson 20))
          bumpydist (uniform-continuous 0 10)
          bumpxdist (uniform-continuous -5 14)
          bumper-positions (repeatedly
                            number-of-bumpers
                            #(vector (sample bumpxdist)
                                     (sample bumpydist)))

          ;; code to simulate the world
          world (create-world bumper-positions)
          end-world (simulate-world world)
          balls (:balls end-world)

          ;; how many balls entered the box?
          num-balls-in-box (balls-in-box end-world)]

      {:balls balls
       :num-balls-in-box num-balls-in-box
       :bumper-positions bumper-positions}))
```





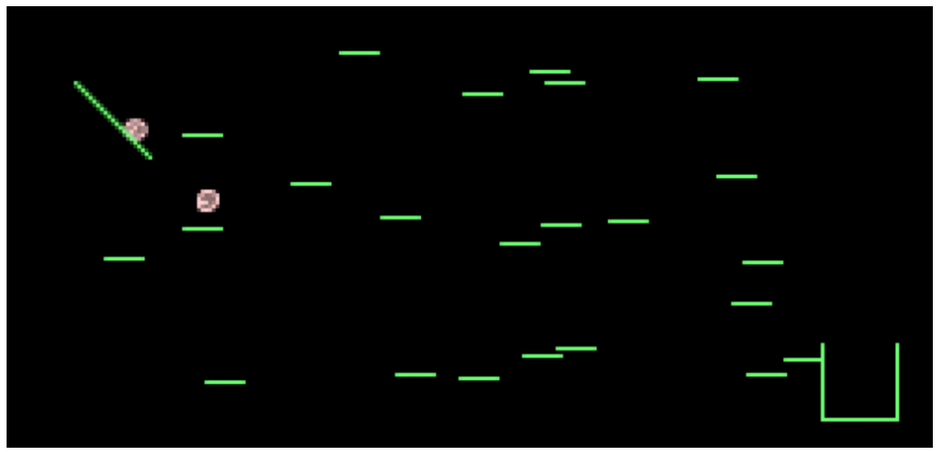


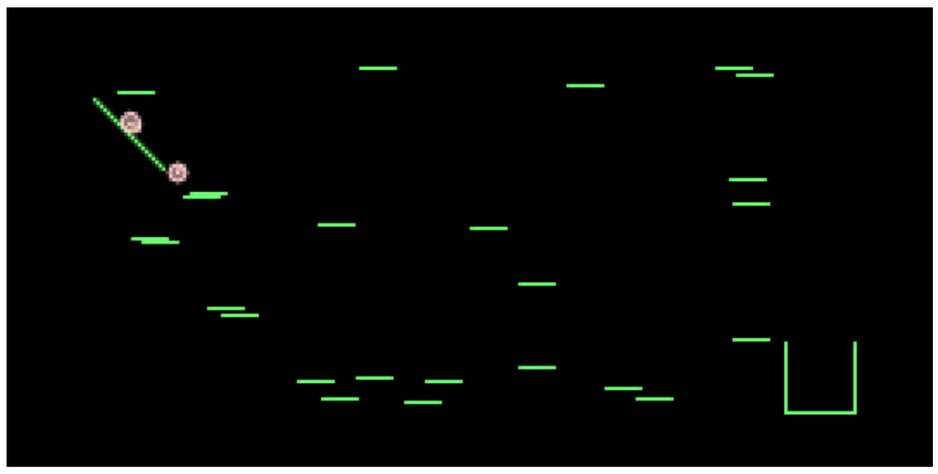
3 examples generated from simulator

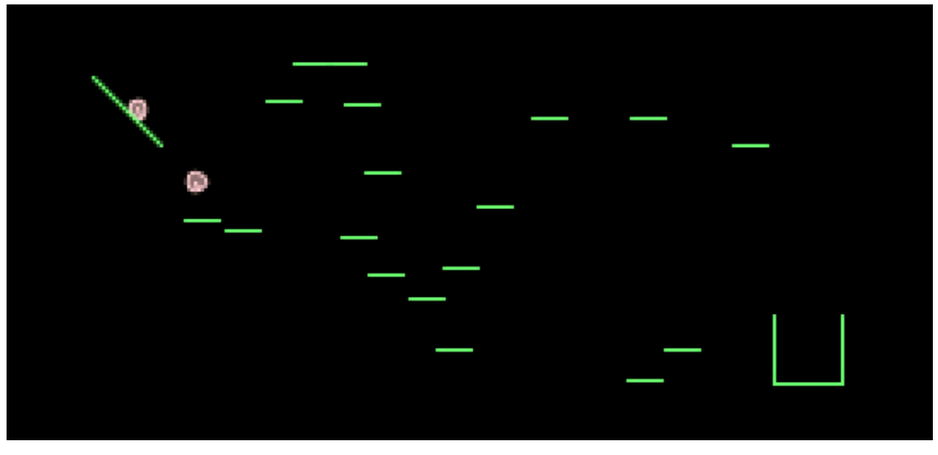
[slides, Frank Wood]

PROBABILISTIC PROGRAMMING EXAMPLE

```
(defquery arrange-bumpers []
    (let [number-of-bumpers (sample (poisson 20))
          bumpydist (uniform-continuous 0 10)
          bumpxdist (uniform-continuous -5 14)
          bumper-positions (repeatedly
                            number-of-bumpers
                            #(vector (sample bumpxdist)
                                     (sample bumpydist)))

          ;; code to simulate the world
          world (create-world bumper-positions)
          end-world (simulate-world world)
          balls (:balls end-world)

          ;; how many balls entered the box?
          num-balls-in-box (balls-in-box end-world)

          obs-dist (normal 4 0.1)]

    (observe obs-dist num-balls-in-box)
```





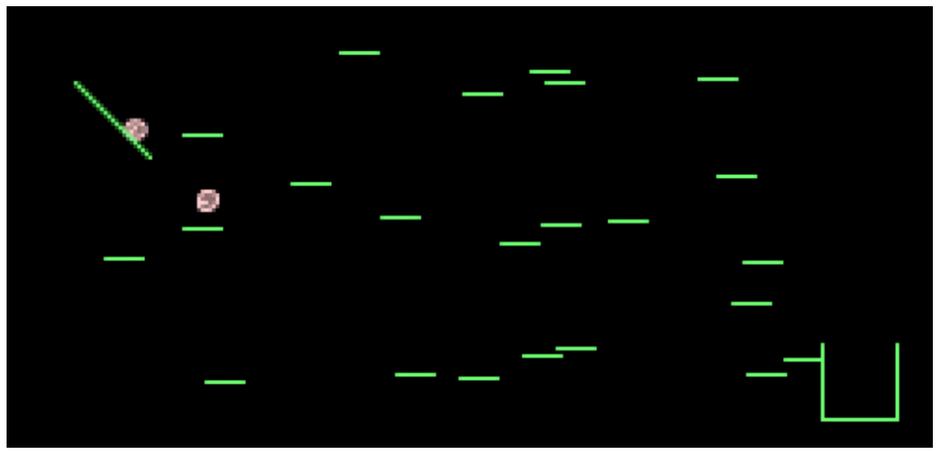


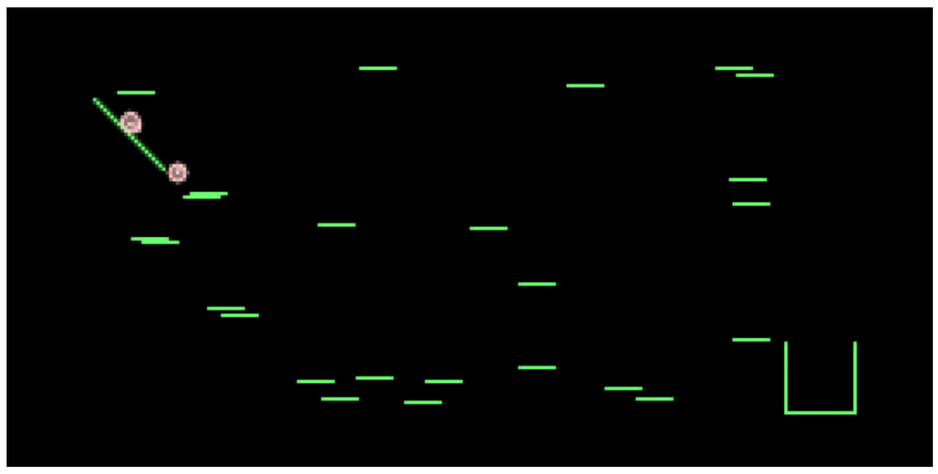
3 examples generated from simulator
conditioned on ~20% of balls land in box

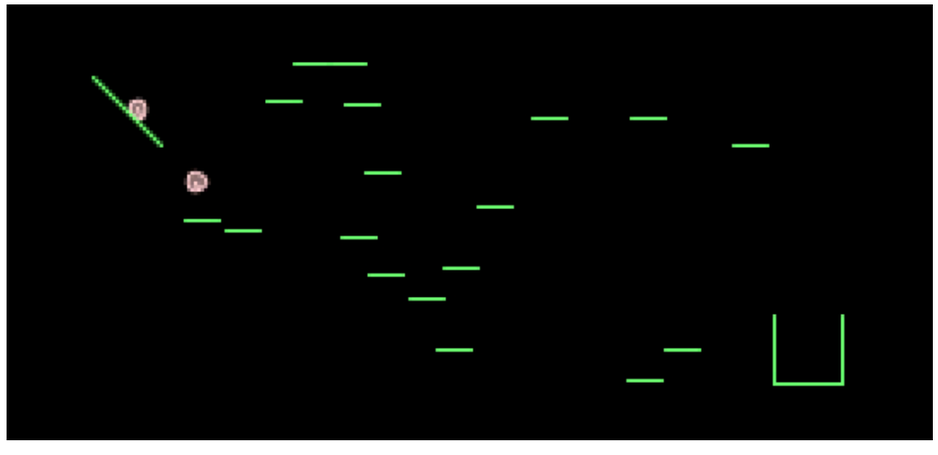
[slides, Frank Wood]

PROBABILISTIC PROGRAMMING EXAMPLE

```
(defquery arrange-bumpers []
    (let [number-of-bumpers (sample (poisson 20))
          bumpydist (uniform-continuous 0 10)
          bumpxdist (uniform-continuous -5 14)
          bumper-positions (repeatedly
                            number-of-bumpers
                            #(vector (sample bumpxdist)
                                     (sample bumpydist)))

          ;; code to simulate the world
          world (create-world bumper-positions)
          end-world (simulate-world world)
          balls (:balls end-world)

          ;; how many balls entered the box?
          num-balls-in-box (balls-in-box end-world)

          obs-dist (normal 4 0.1)]

    (observe obs-dist num-balls-in-box)
```





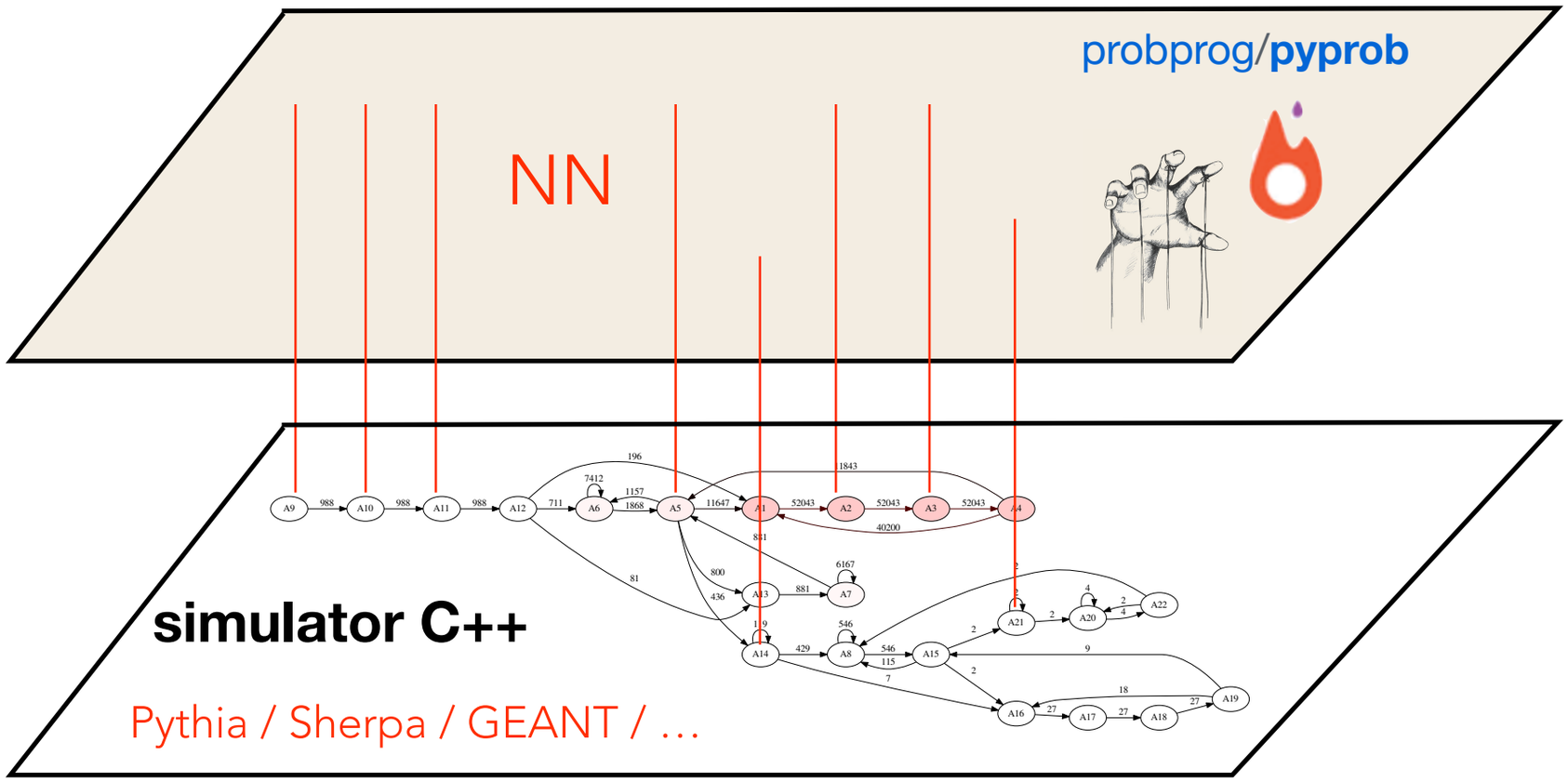


3 examples generated from simulator
conditioned on ~20% of balls land in box

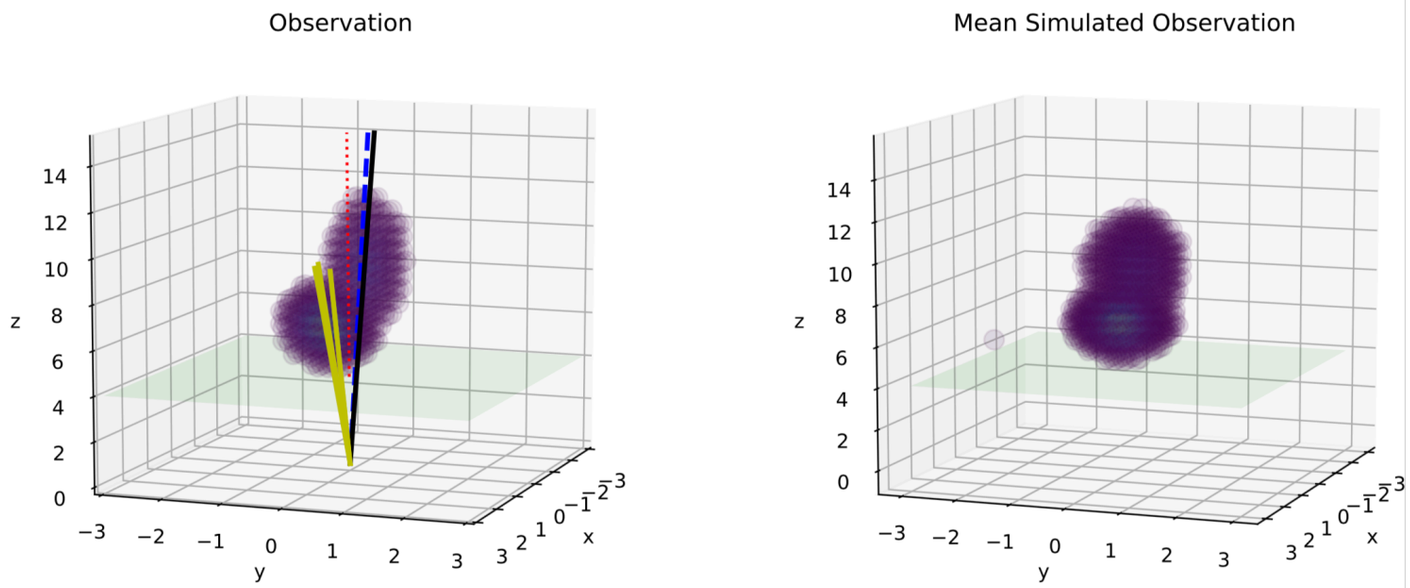
[slides, Frank Wood]

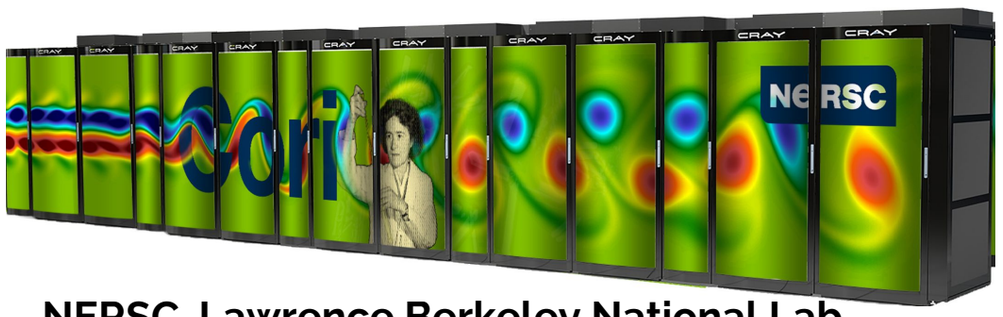
PROBABILISTIC PROGRAMING

Idea: hijack the random number generators and use Neural Network to perform a *very* fancy type of importance sampling



- Neural Network powered inference engine (python)
- real-world scientific simulator (C++)



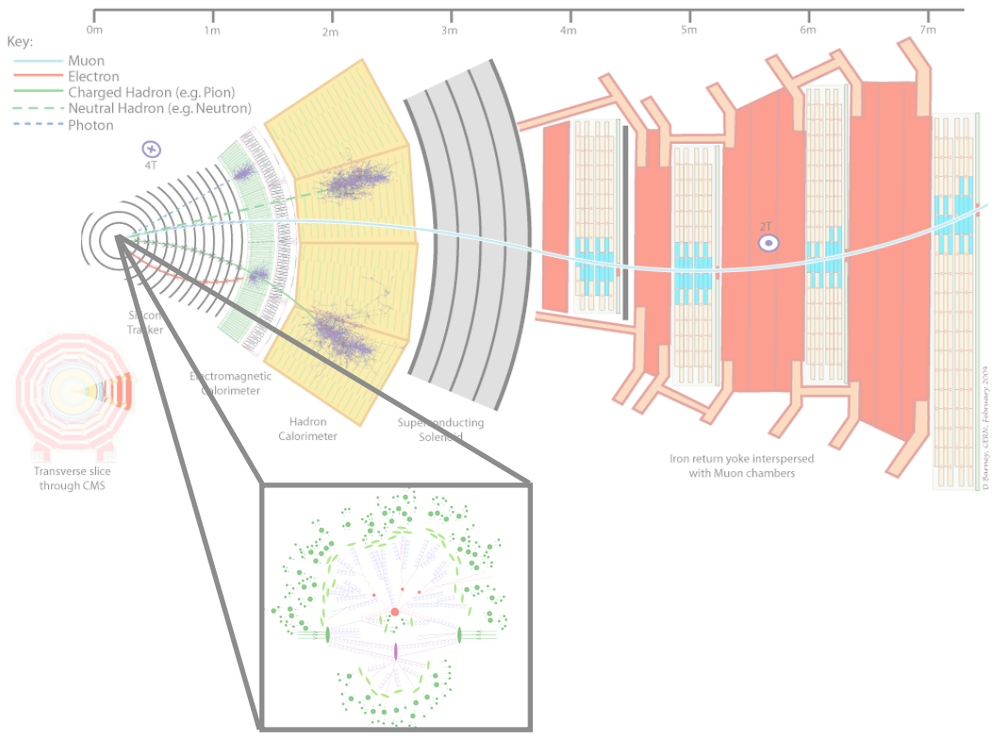


NERSC, Lawrence Berkeley National Lab

arXiv:1807.07706

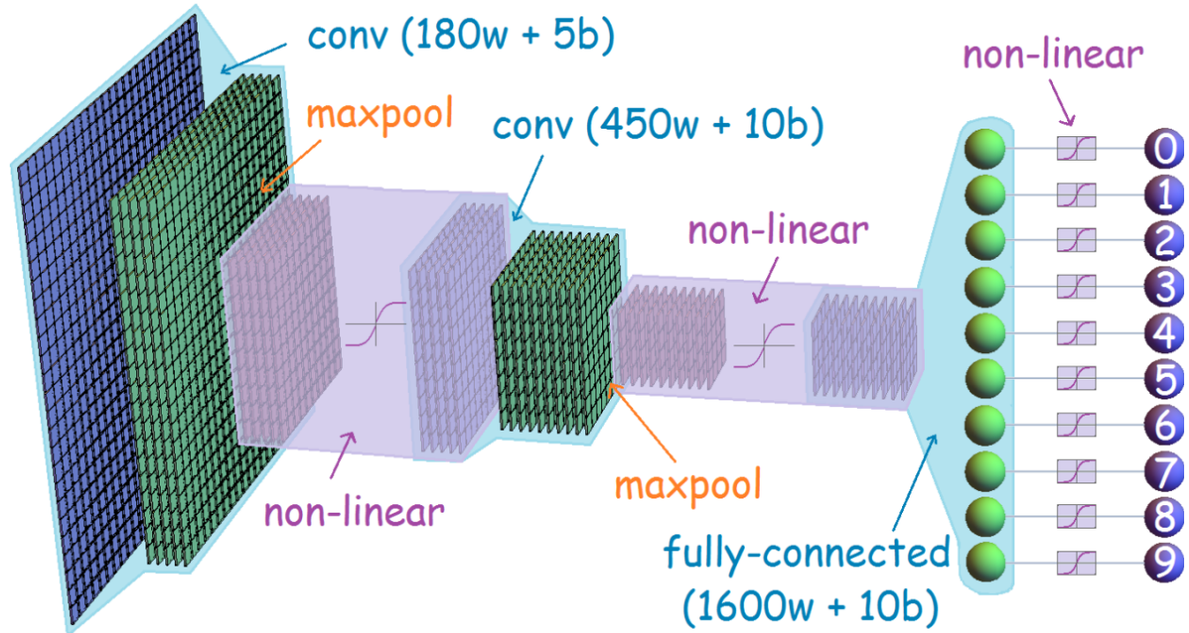
TWO APPROACHES TO LIKELIHOOD FREE INFERENCE

Use simulator
(much more efficiently)



- Approximate Bayesian Computation (ABC)
- Probabilistic Programming
- Adversarial Variational Optimization (AVO)

Learn simulator
(with deep learning)



- Generative Adversarial Networks (GANs), Variational Auto-Encoders (VAE)
- Likelihood ratio from classifiers (CARL)
- Autogregressive models, Normalizing Flows

[image credit: A.P. Goucher]

GENERATIVE MODEL FOR IMAGES

Note, same NN can model birds, ants, and volcanos! Is that good or bad?

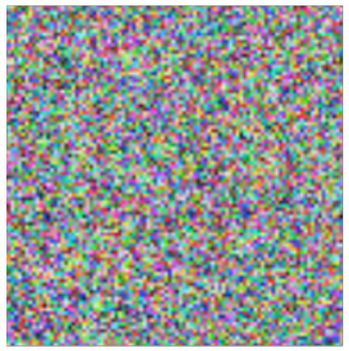
LEARNING THE SIMULATOR

note, same NN can model birds, ants, and volcanos! (lack of implicit bias)

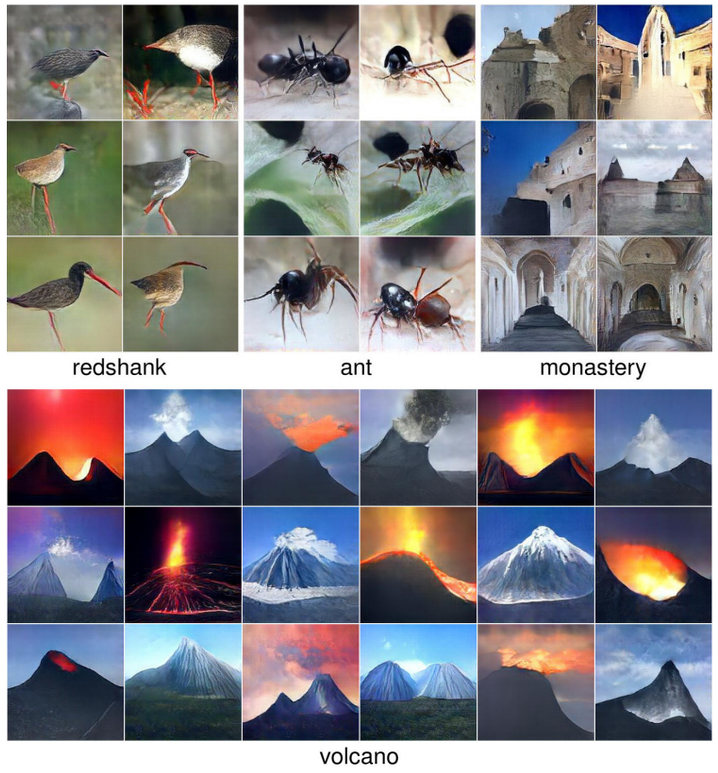
Z

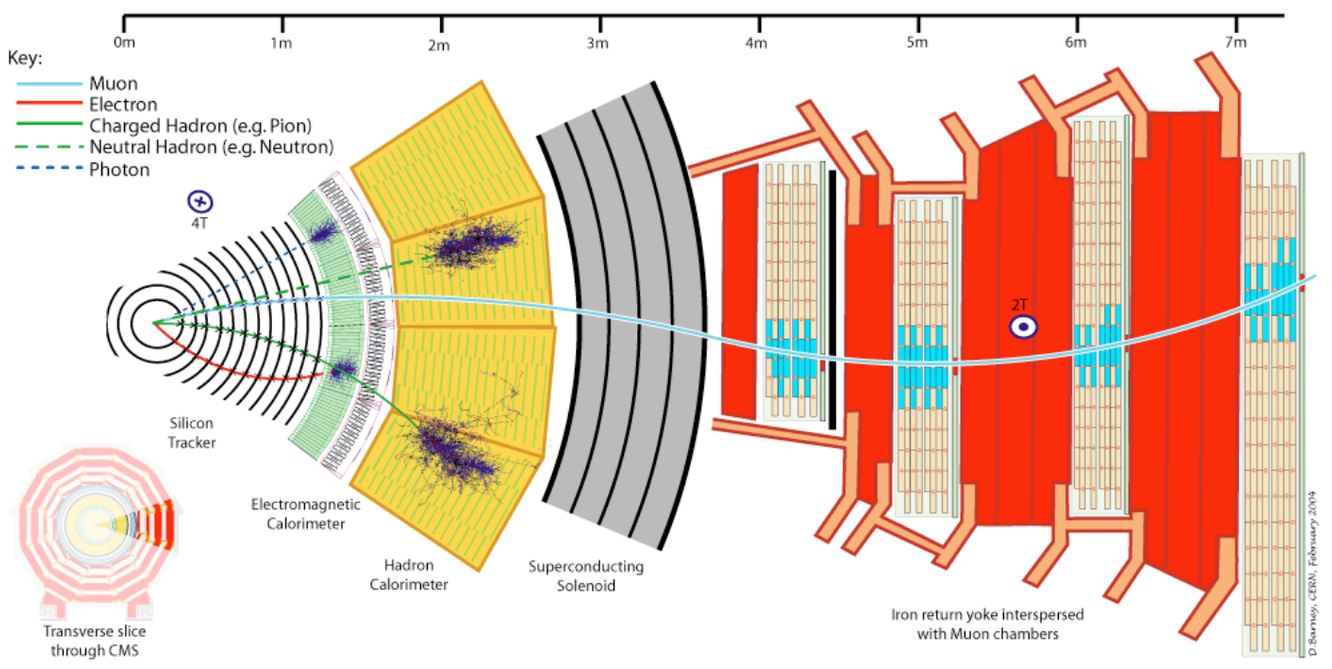
X

Noise ~ N(0,1)



Generative Model





http://torch.ch/blog/2015/11/13/gan.html

Paganini, et. al. Phys.Rev.Lett. 120 (2018) no.4, 042003 ; & ATL-SOFT-PUB-2018-001

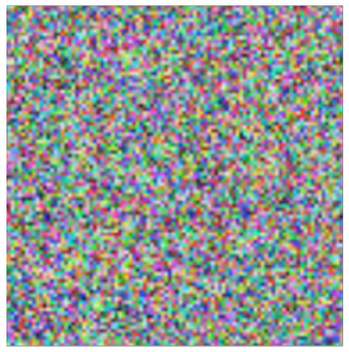
LEARNING THE SIMULATOR

Z

X

note, same NN can model birds, ants, and volcanos! (lack of implicit bias)

Noise ~ N(0,1)



Generative Model

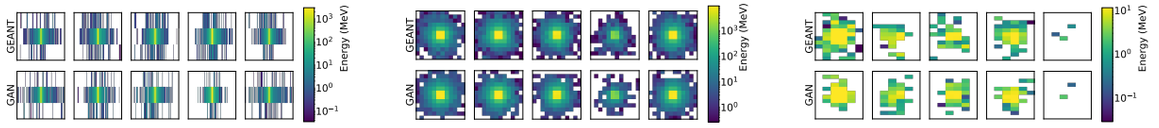


Figure 9: Five randomly selected e^+ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

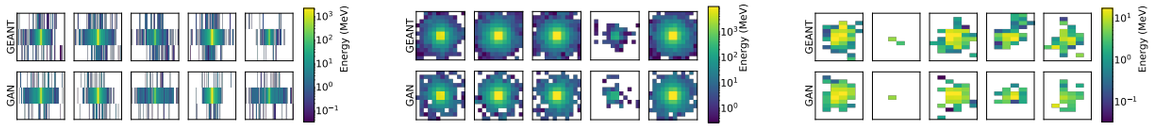


Figure 10: Five randomly selected γ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

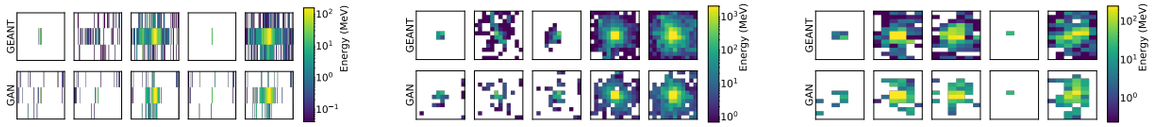
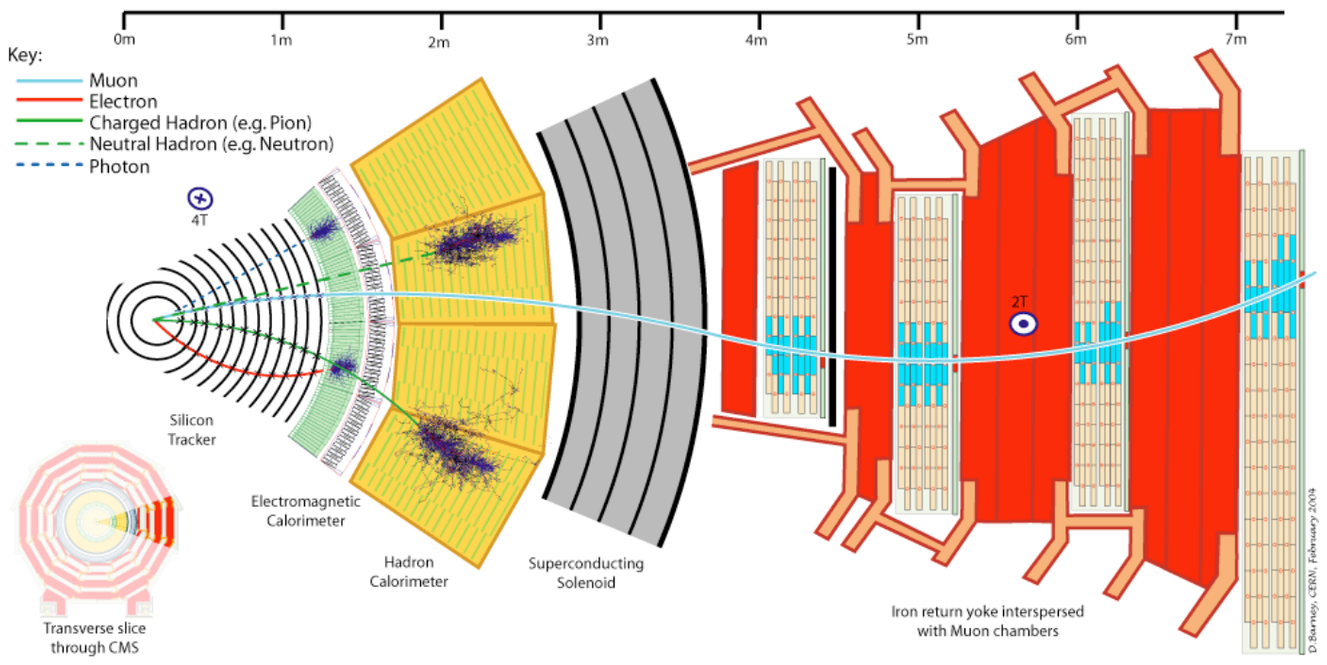


Figure 11: Five randomly selected π^+ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

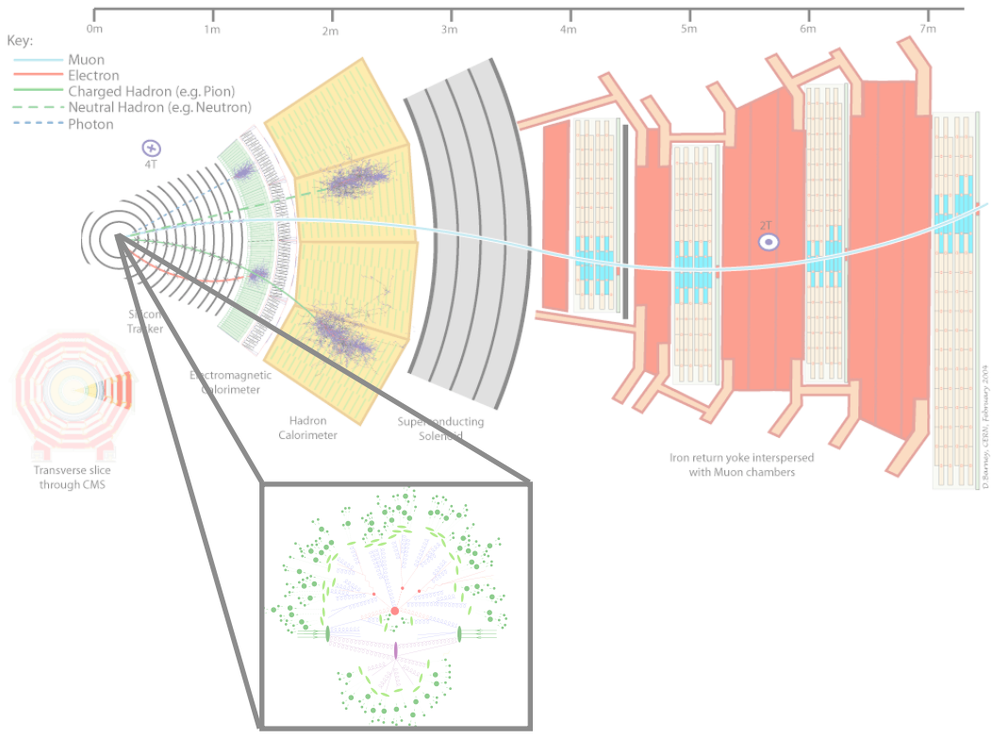


http://torch.ch/blog/2015/11/13/gan.html

Paganini, et. al. Phys.Rev.Lett. 120 (2018) no.4, 042003 ; & ATL-SOFT-PUB-2018-001

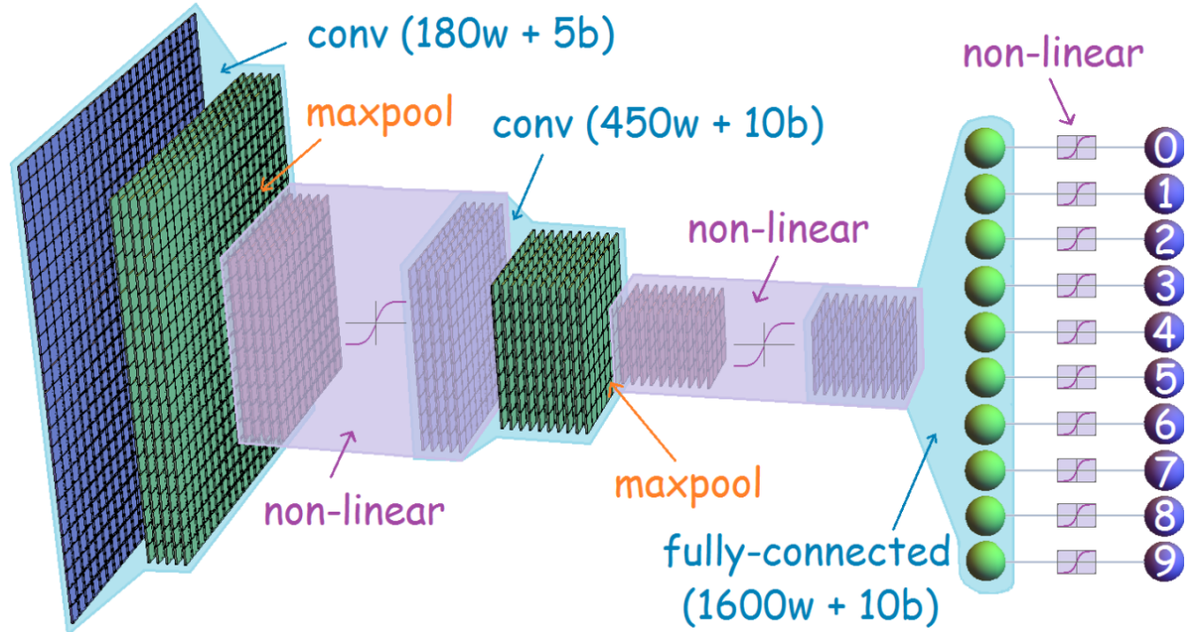
TWO APPROACHES TO LIKELIHOOD FREE INFERENCE

Use simulator
(much more efficiently)



- Approximate Bayesian Computation (ABC)
- Probabilistic Programming
- Adversarial Variational Optimization (AVO)

Learn simulator
(with deep learning)



- Generative Adversarial Networks (GANs), Variational Auto-Encoders (VAE)
- Likelihood ratio from classifiers (CARL)
- Autogregressive models, Normalizing Flows

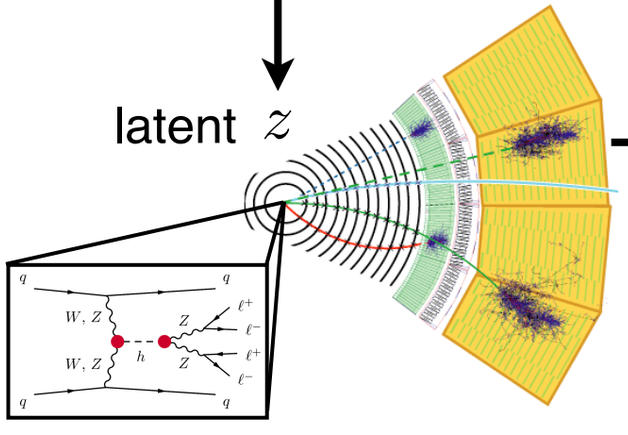
[image credit: A.P. Goucher]

LEARNING THE LIKELIHOOD RATIO

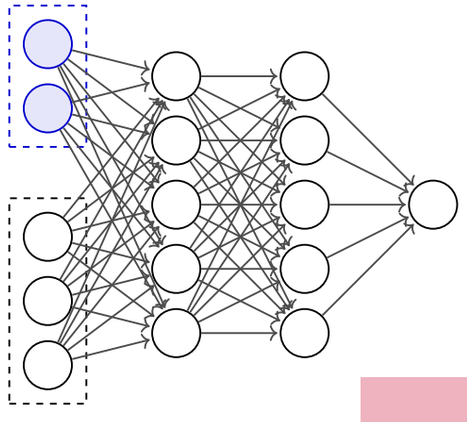
arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90

parameter θ

latent z

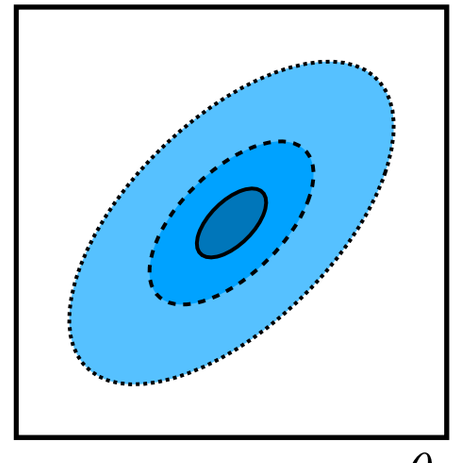


observable x



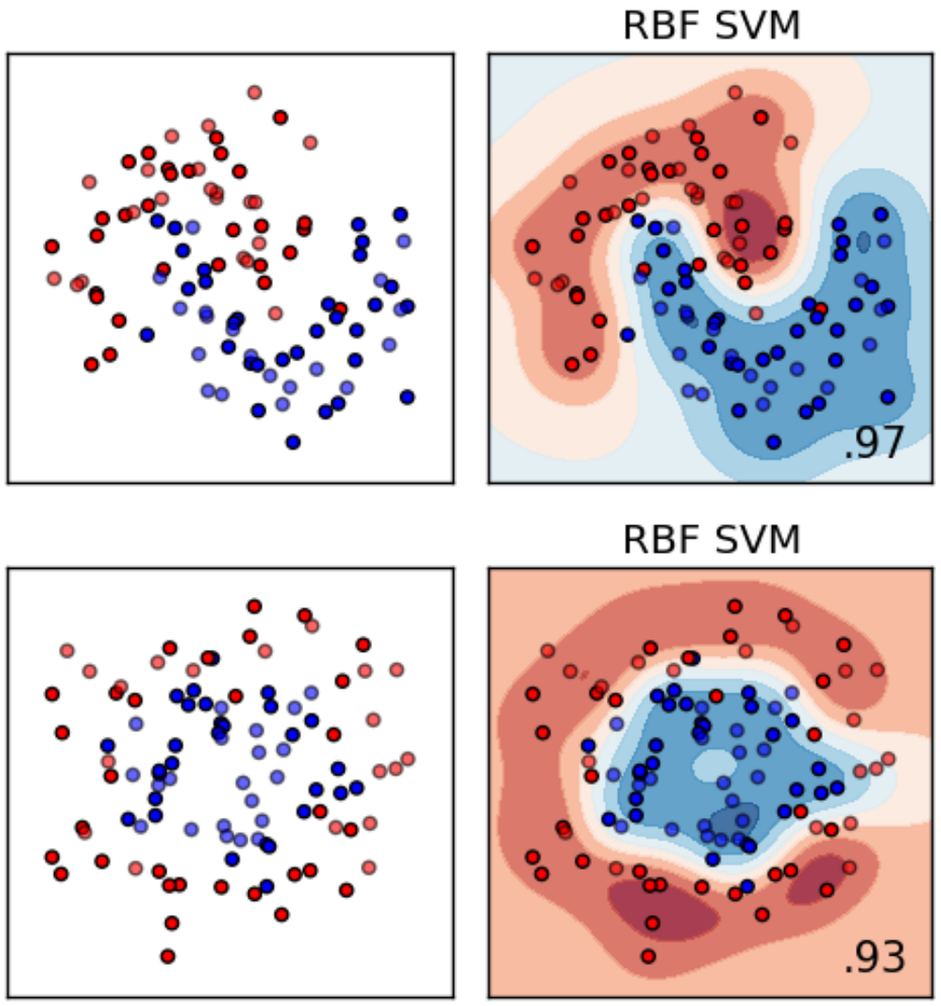
$\arg\min_g L[g]$

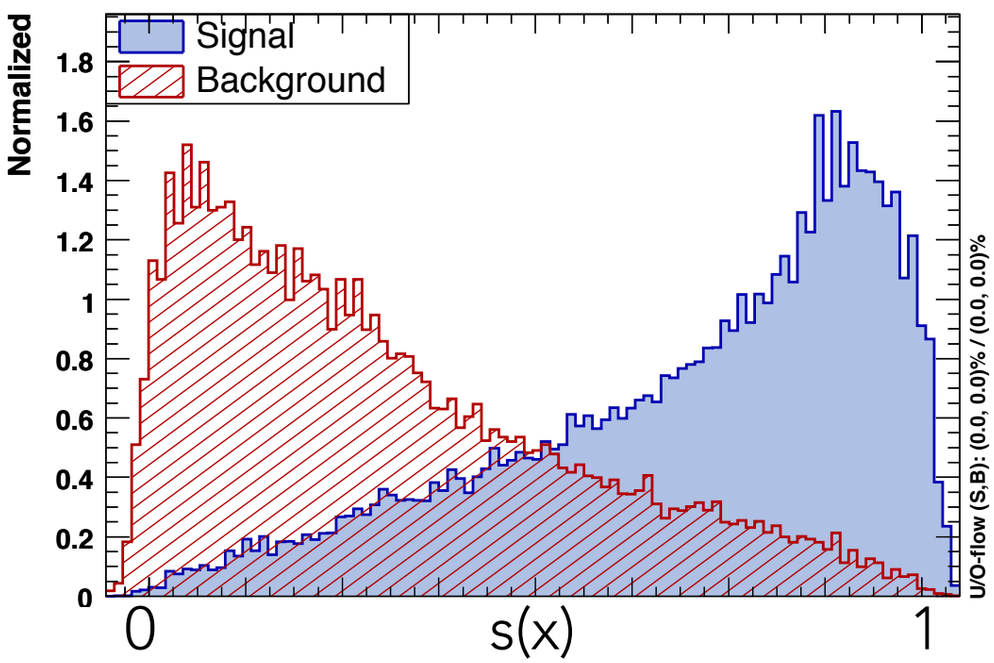
approximate likelihood ratio $\hat{r}(x|\theta)$



Simulation | Machine Learning | Inference

LIKELIHOOD RATIO TRICK





- **binary classifier**: find function s(x) that minimizes ***loss***:

$$L[s] = \int p(x|H_0)\,(0 - s(x))^2\,dx + \int p(x|H_1)\,(1 - s(x))^2 dx$$

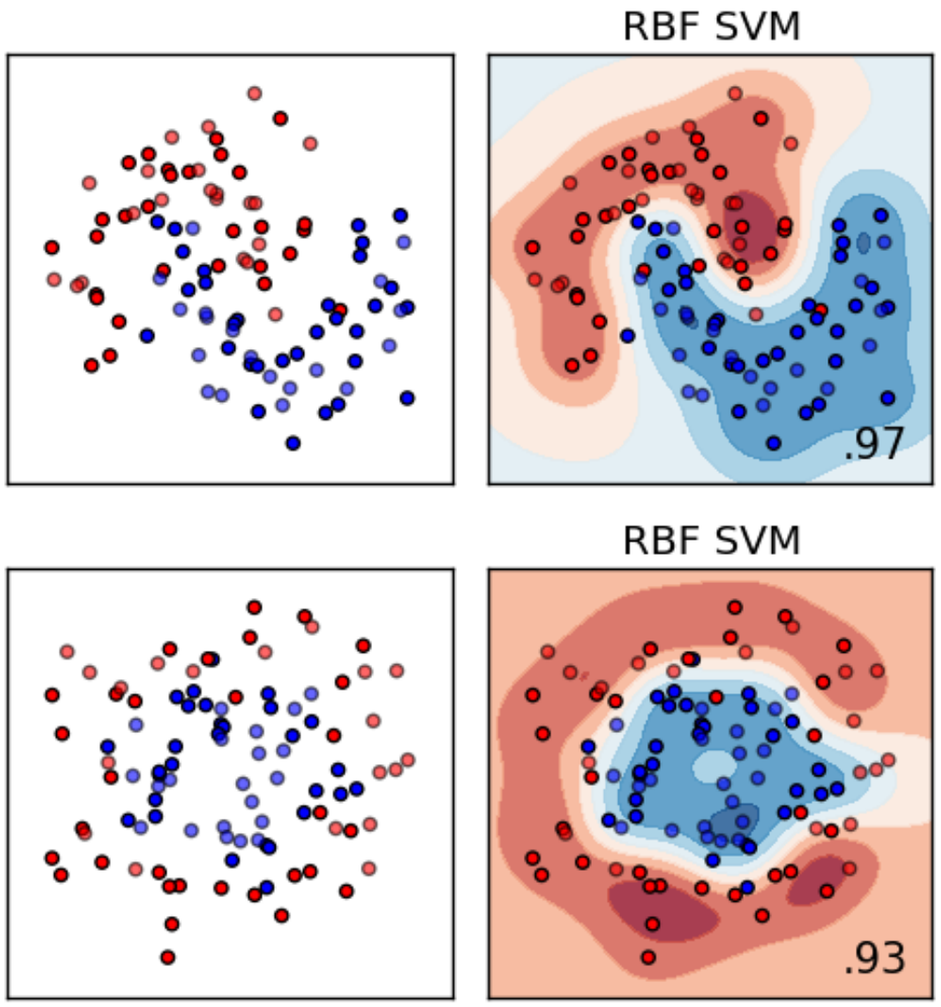
- i.e. approximate the optimal classifier

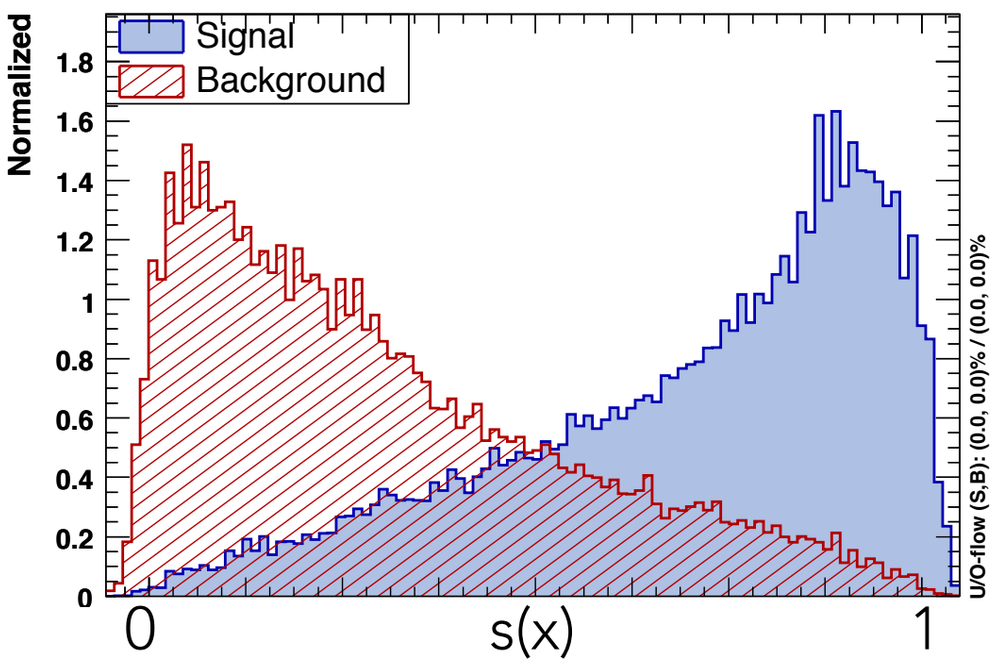
$$s(x) = \frac{p(x|H_1)}{p(x|H_0) + p(x|H_1)}$$

- which is 1-to-1 with the likelihood ratio

$$\frac{p(x|H_1)}{p(x|H_0)}$$

LIKELIHOOD RATIO TRICK





- **binary classifier**: find function s(x) that minimizes ***loss***:

$$L[s] = \int p(x|H_0)\,(0 - s(x))^2\,dx + \int p(x|H_1)\,(1 - s(x))^2 dx$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}(y_i - s(x_i))^2$$

- i.e. approximate the optimal classifier

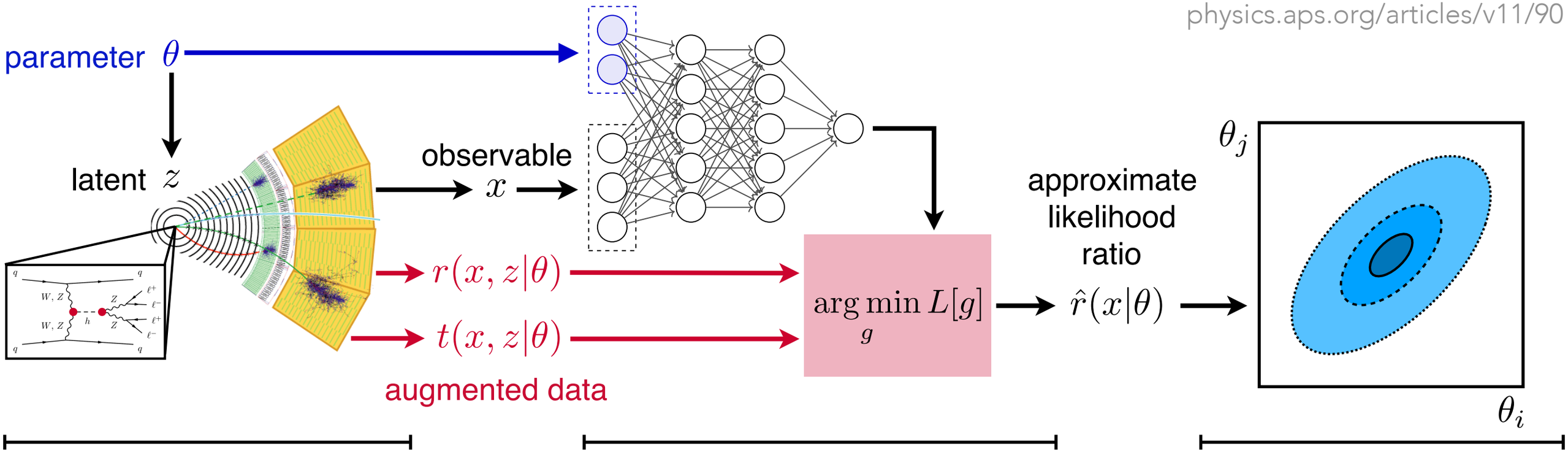
$$s(x) = \frac{p(x|H_1)}{p(x|H_0) + p(x|H_1)}$$

- which is 1-to-1 with the likelihood ratio

$$\frac{p(x|H_1)}{p(x|H_0)}$$

LEARNING THE LIKELIHOOD RATIO

arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90

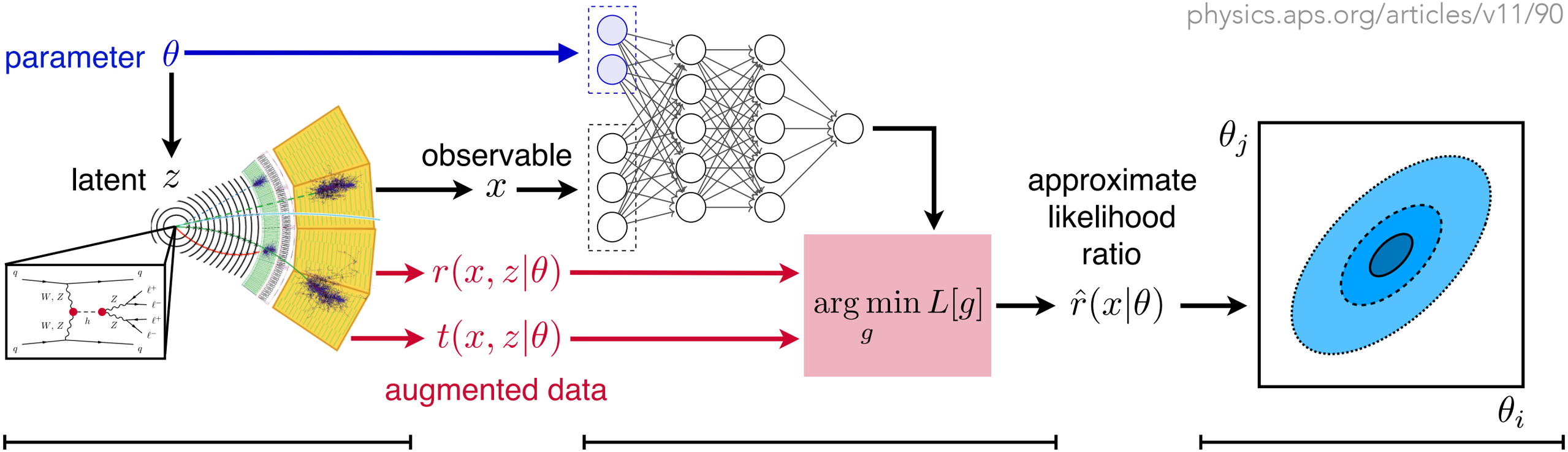


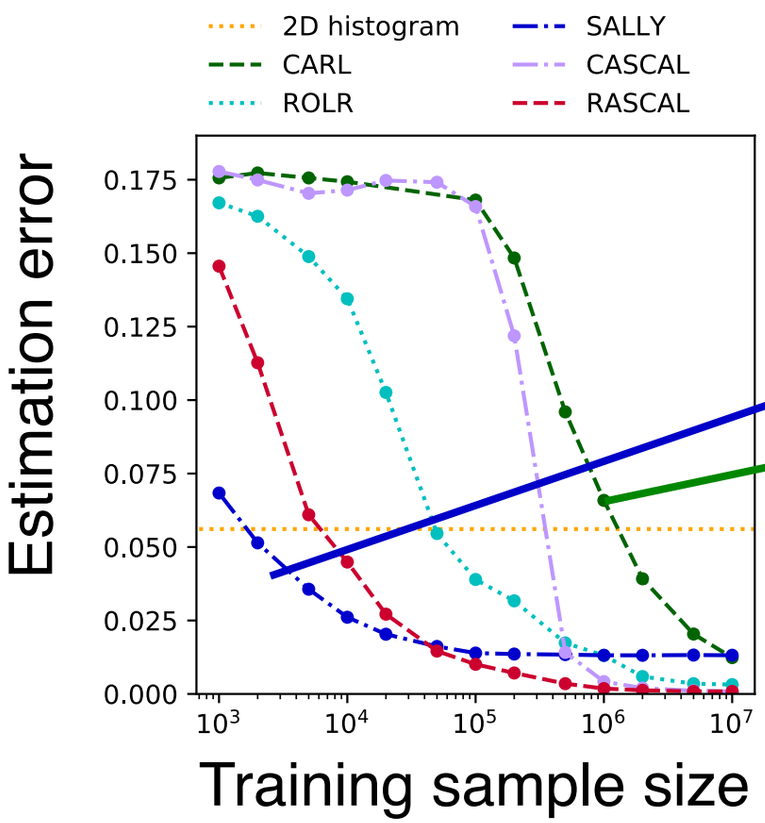
Recently, we realized we can **extract more from the simulator**.
We can use **augmented data** to improve training

(connections to reinforcement learning)

LEARNING THE LIKELIHOOD RATIO

arXiv:1805.12244
PRL, arXiv:1805.00013
PRD, arXiv:1805.00020
physics.aps.org/articles/v11/90



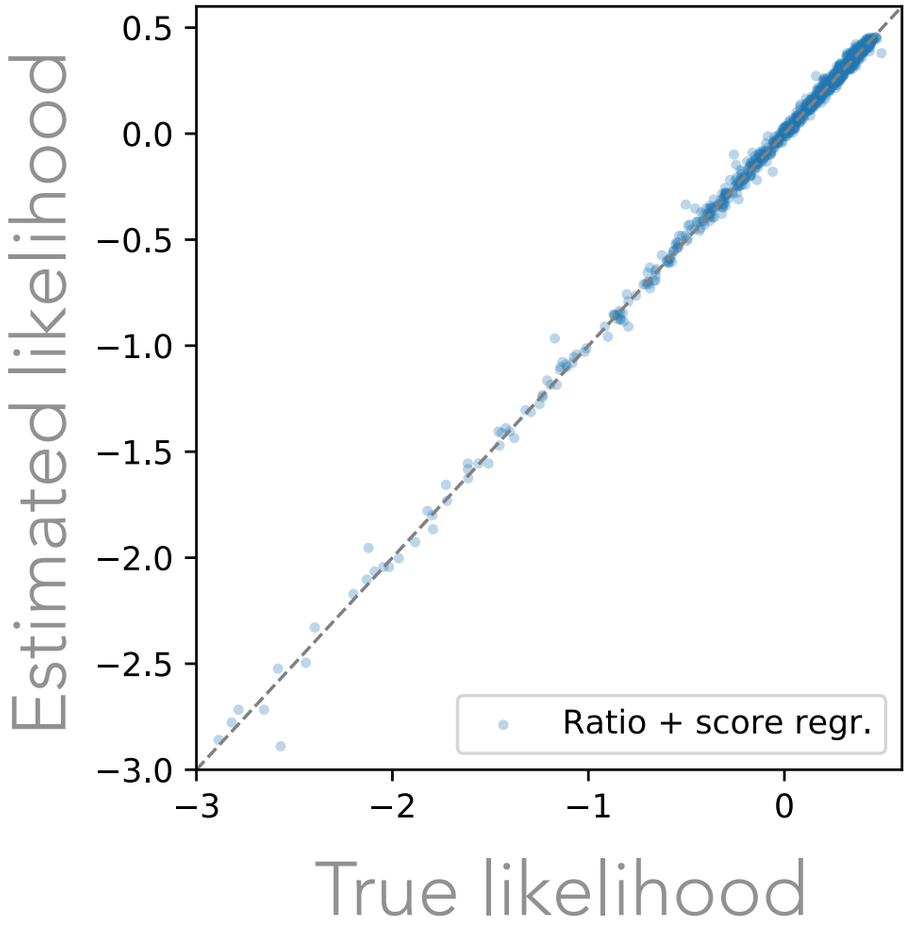


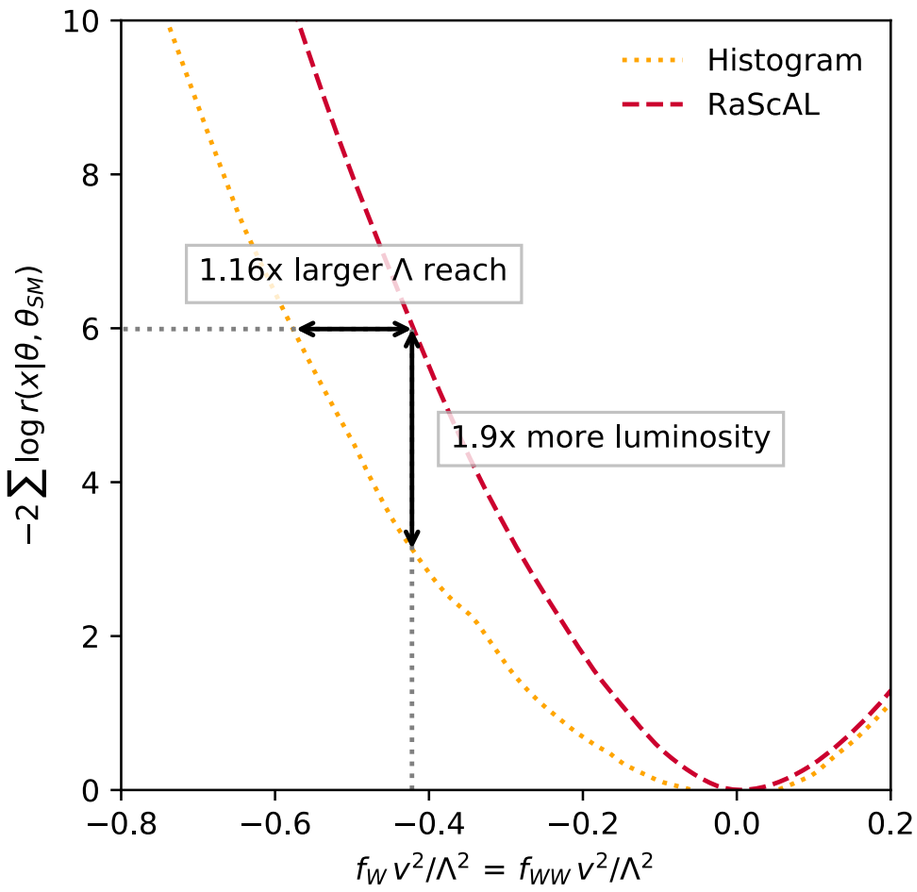
New techniques
require less data than
without augmented data

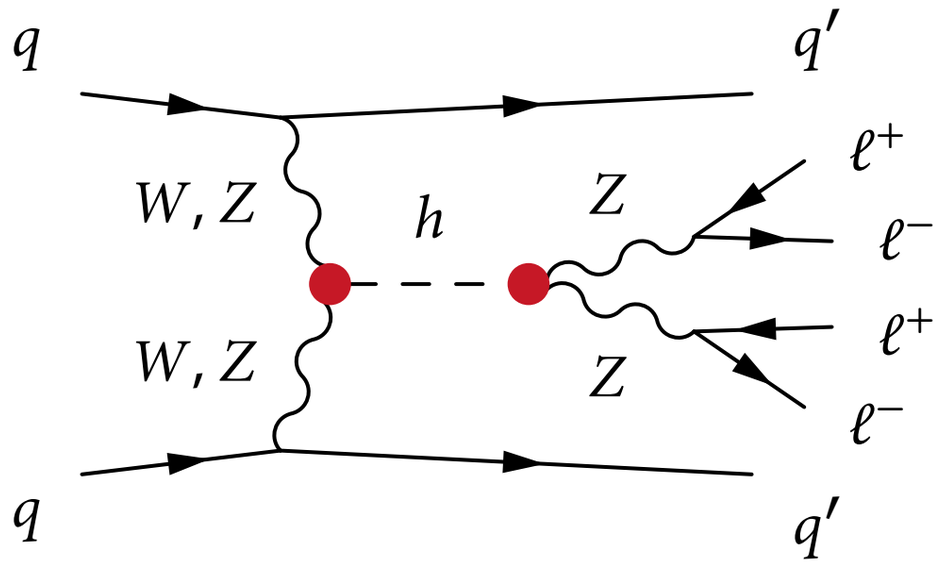
Traditional Approach no NN

IMPACT ON STUDIES OF THE HIGGS BOSON

(based on a 42-Dim observation ***x***)







J Brehmer, J Pavez, G Louppe, K.C. PRL & PRD 2018 [arXiv:1805.00013 & arXiv:1805.00020]
"Better Higgs Measurements Through Information Geometry" [arXiv:1612.05261] & CARL [arxiv:1506.02169]

TAKE AWAYS

Many areas of science have simulations based on some well-motivated mechanistic model.

However, the aggregate effect of many interactions between these low-level components leads to an intractable inverse problem.

The developments in machine learning and AI have the potential to effectively bridge the microscopic - macroscopic divide & aid in the inverse problem.

- they can provide effective statistical models that describe emergent macroscopic phenomena that are tied back to the low-level microscopic (reductionist) model

- generative models and likelihood-free inference are two particularly exciting areas

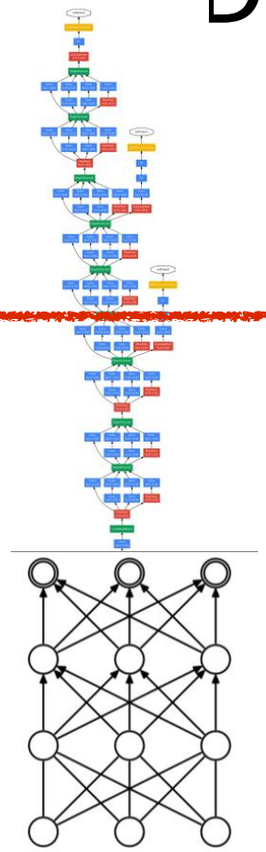
Physics-Aware Machine Learning

PHYSICS AT THE INTERSECTION

Max Welling

We can leverage both the power of deep learning and inject our expert physics knowledge

Discriminative or Generative?



-Deep Learning

-Kernel Methods

-Random Forests

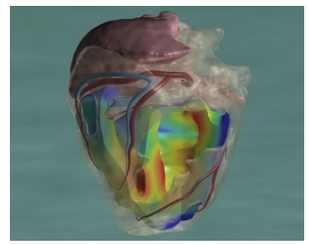
-Boosting

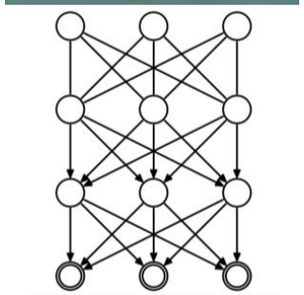
Exciting

-Bayesian Networks

-Probabilistic Programs

-Simulator Models





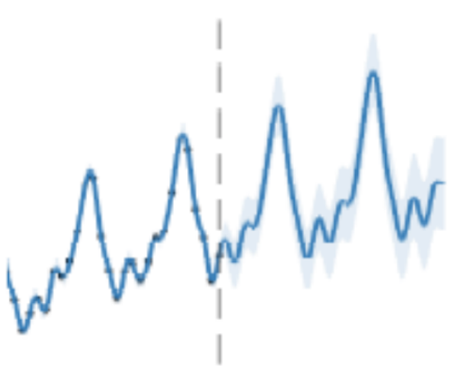
- Advantages discriminative models:
  - Flexible map from input to target (low bias)
  - Efficient training algorithms available
  - Solve the problem you are evaluating on.
  - Very successful and accurate!

- Advantages generative models:
  - Inject expert knowledge
  - Model causal relations
  - Interpretable
  - Data efficient
  - More robust to domain shift
  - Facilitate un/semi-supervised learning

NARRATIVE MODELING

Physics goes into the construction of a "Kernel" that defines the model

- Vocabulary of kernels + grammar for composition = powerful modeling

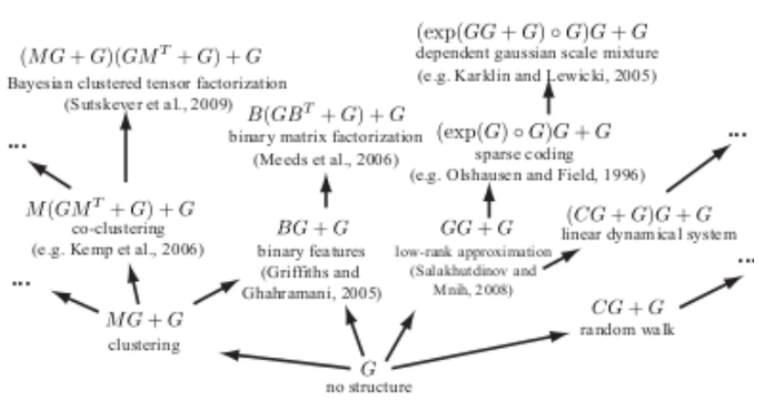


Structure Discovery in Nonparametric Regression through Compositional Kernel Search

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani

International Conference on Machine Learning, 2013

pdf | code | poster | bibtex

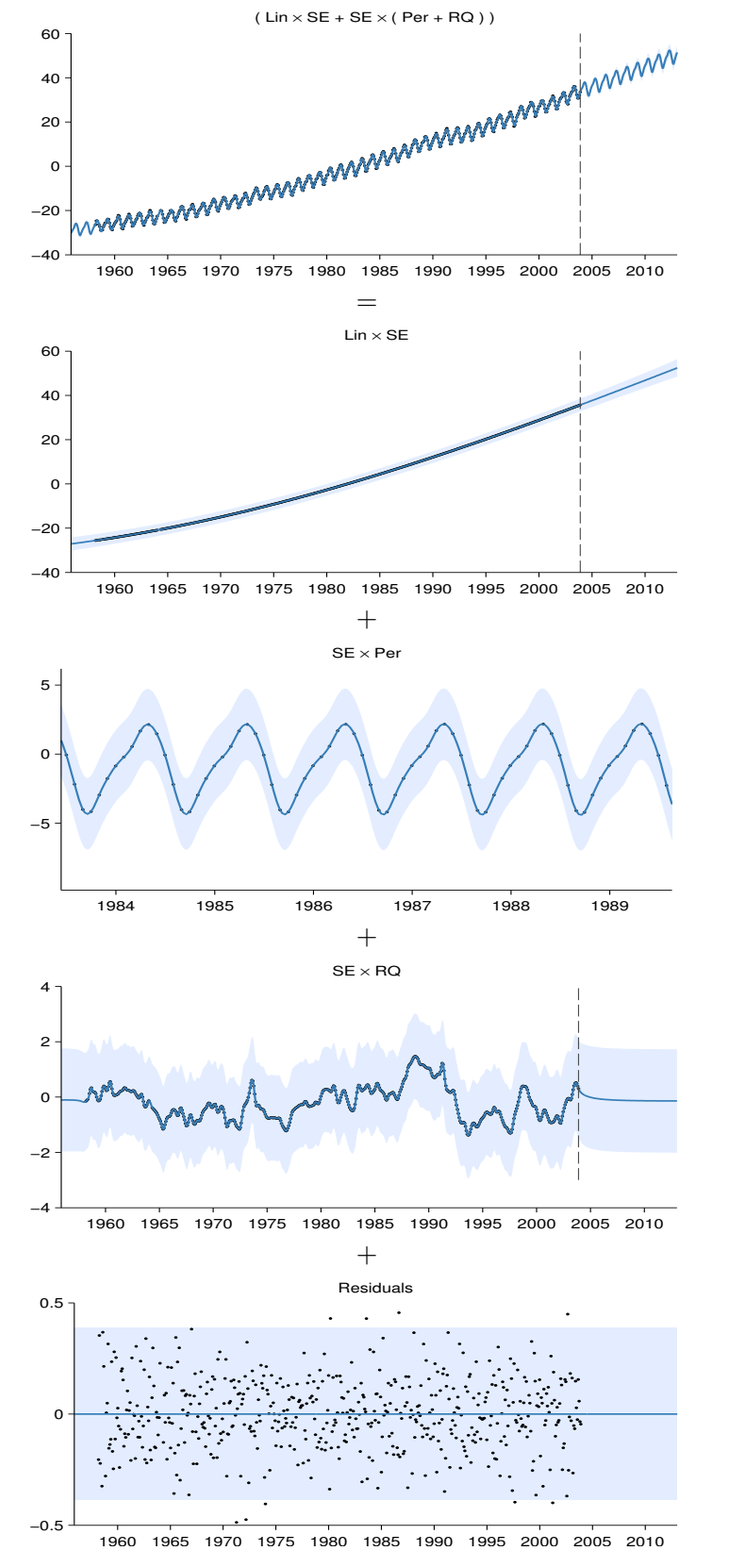


Exploiting compositionality to explore a large space of model structures

Roger Grosse, Ruslan Salakhutdinov, William T. Freeman, Joshua B. Tenenbaum

Conference on Uncertainty in Artificial Intelligence, 2012

pdf | code | bibtex

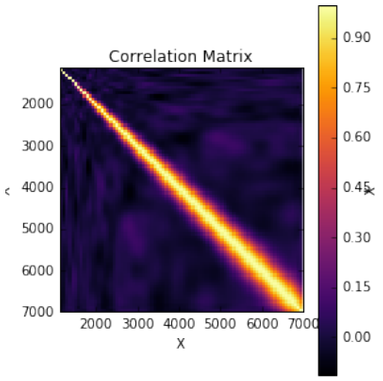


PHYSICS-AWARE MACHINE LEARNING

We can **inject** our knowledge of physics into the machine learning models!
We can **extract** knowledge learned from the data!

Physics-aware Gaussian Processes

arXiv:1709.05681

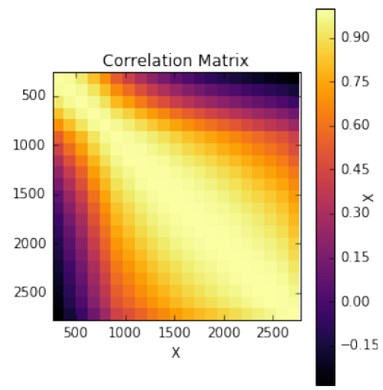


Final Kernel =

Poisson fluctuations

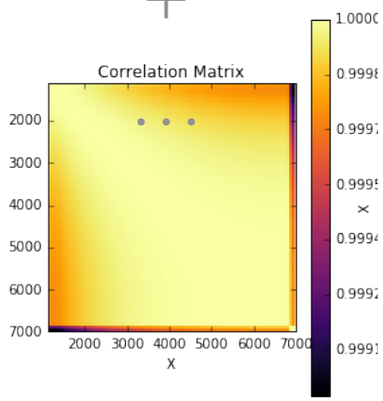
=

+ Mass Resolution



+

+ Parton Density Functions

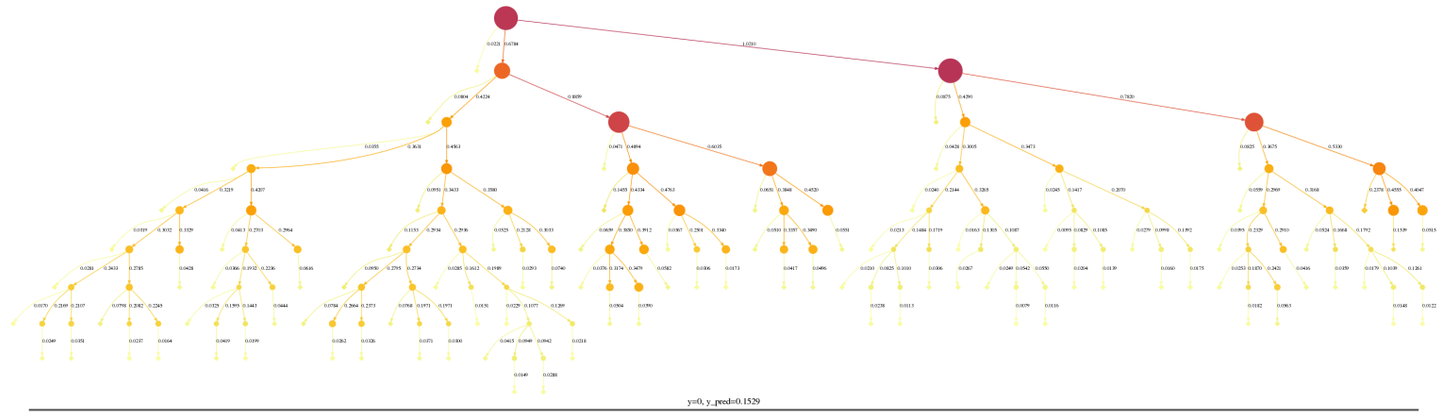


+

+ Jet Energy Scale

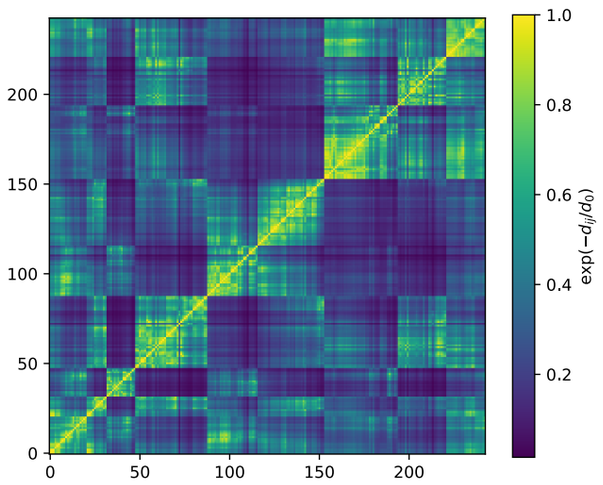
QCD-Aware recursive neural networks

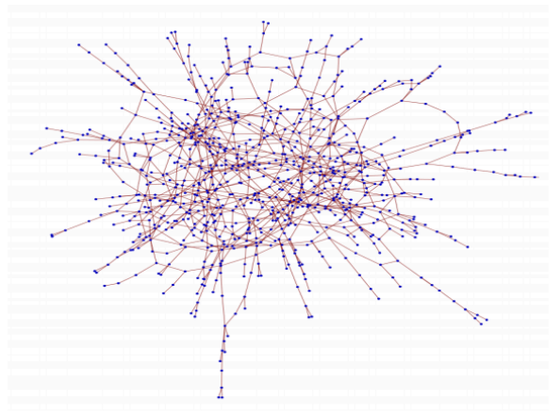
arXiv:1702.00748



QCD-Aware graph convolutional neural networks

NIPS2017 workshop [http://bit.ly/2AkwYRG]



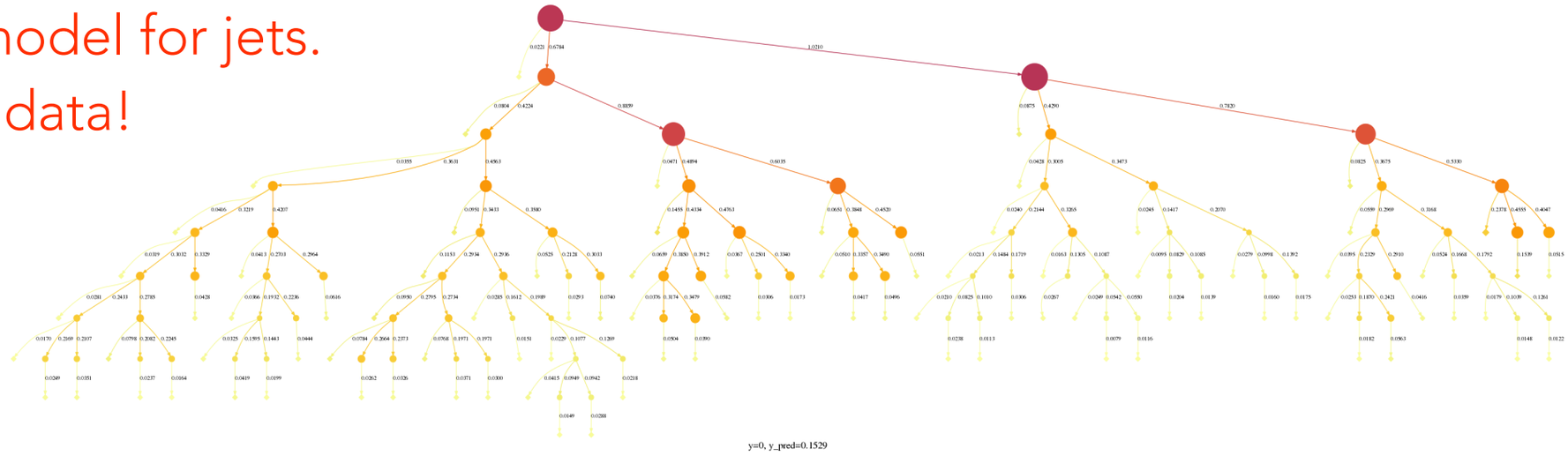


$$d_{ii'}^{\alpha} = \min(p_{ti}^{2\alpha}, p_{ti'}^{2\alpha}) \frac{\Delta R_{ii'}^2}{R^2}$$

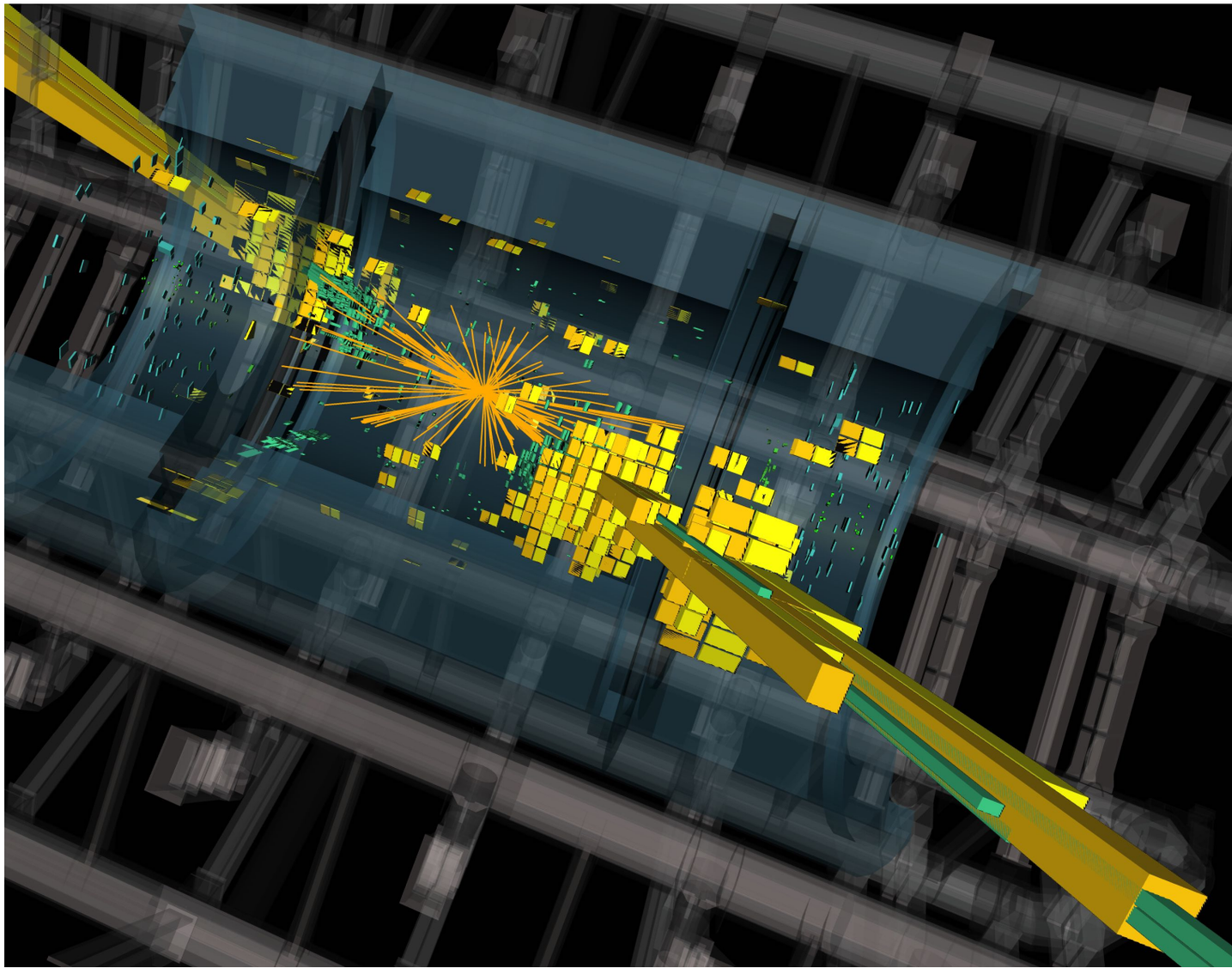
CAUSAL GENERATIVE MODELS

JUNIPR is a generative model for jets.

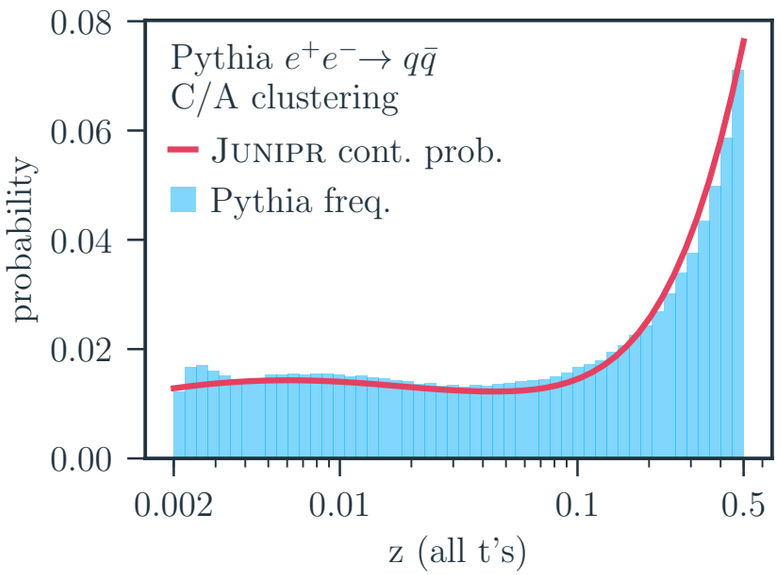
Can train on real data!



$$P_{\text{jet}}(\{p_1, \dots p_n\}) = \left[\prod_{t=1}^{n-1} P_t\left(k_1^{(t+1)}, \dots, k_{t+1}^{(t+1)} \middle| k_1^{(t)}, \dots, k_t^{(t)}\right) \right] \times P_n\left(\text{end} \middle| k_1^{(n)}, \dots, k_n^{(n)}\right).$$



… and it is interpretable



Andreassen, Feige, Frye, Schwartz arXiv:1804.09720

CONCLUSIONS

Our understanding of how to leverage our prior physics knowledge while letting machine learning do what it's good at is maturing.

- build in robustness to systematic uncertainties
- ability to inject and extract physics knowledge from models
- exploit symmetries, hierarchical structure of data

Harnessing the full potential of these techniques will require deep integration into our scientific workflow

COLLABORATORS

Gilles Louppe
U. Liège

Kyunghyun Cho

Joan Bruna

Brenden Lake

Meghan Frate

Juan Pavez

Tilman Plehn

Johann Brehmer

Isaac Henrion

Lukas Heinrich

Heiko Müller

Tim Head

Michael Kagan

Irina Espejo

Peter Sadowski

Daniel Whiteson

Pierre Baldi

Lezcano Casado

Atılım Güneş Baydin
University of Oxford

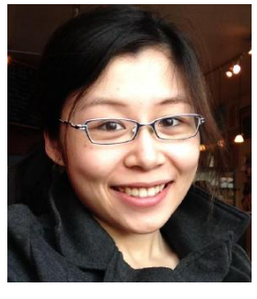
Prabhat
NERSC, Berkeley Lab

Wahid Bhimji
NERSC, Berkeley Lab

Frank Wood
University of Oxford

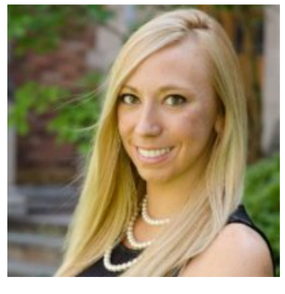
Phiala Shanahan

William Detmold



Karen Ng

Tuan Anh Le



Michela Paganini
Yale University

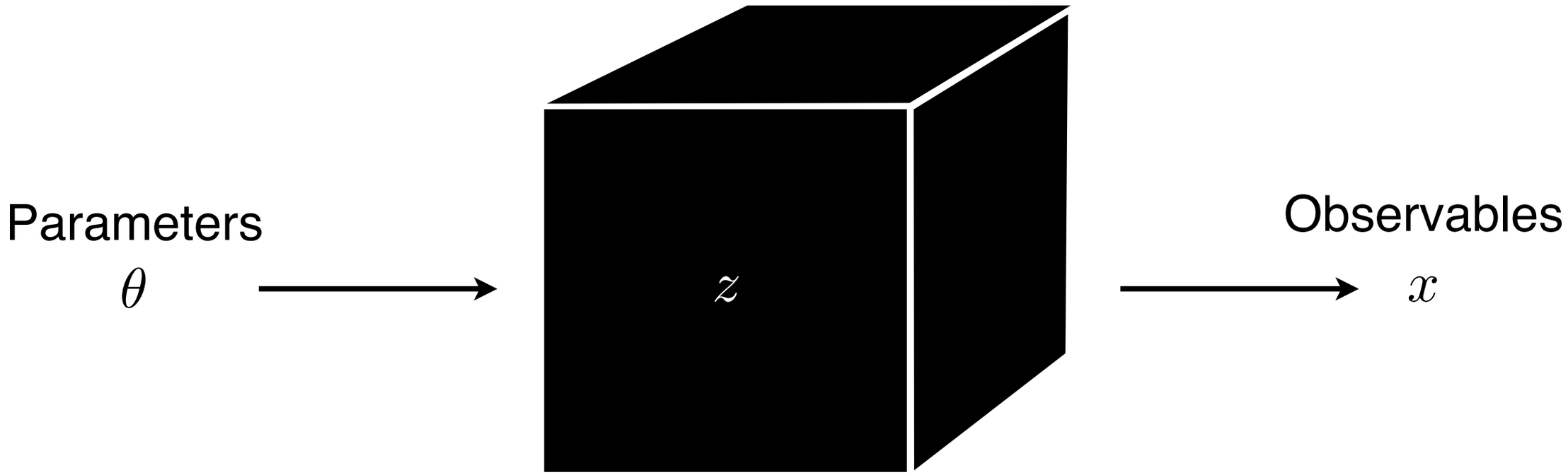
Daniela Huppenkothen
New York University

Savannah Thais
Yale University

Ruth Angus
Columbia University

Other examples

LIKELIHOOD-FREE INFERENCE



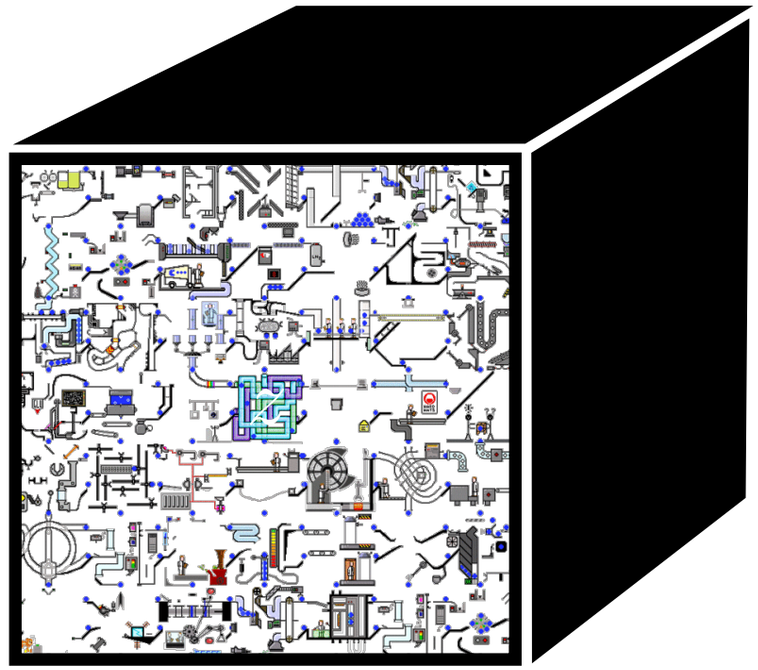
Prediction (simulation):

- Well-understood mechanistic model
- Simulator can generate samples

LIKELIHOOD-FREE INFERENCE

Parameters

θ



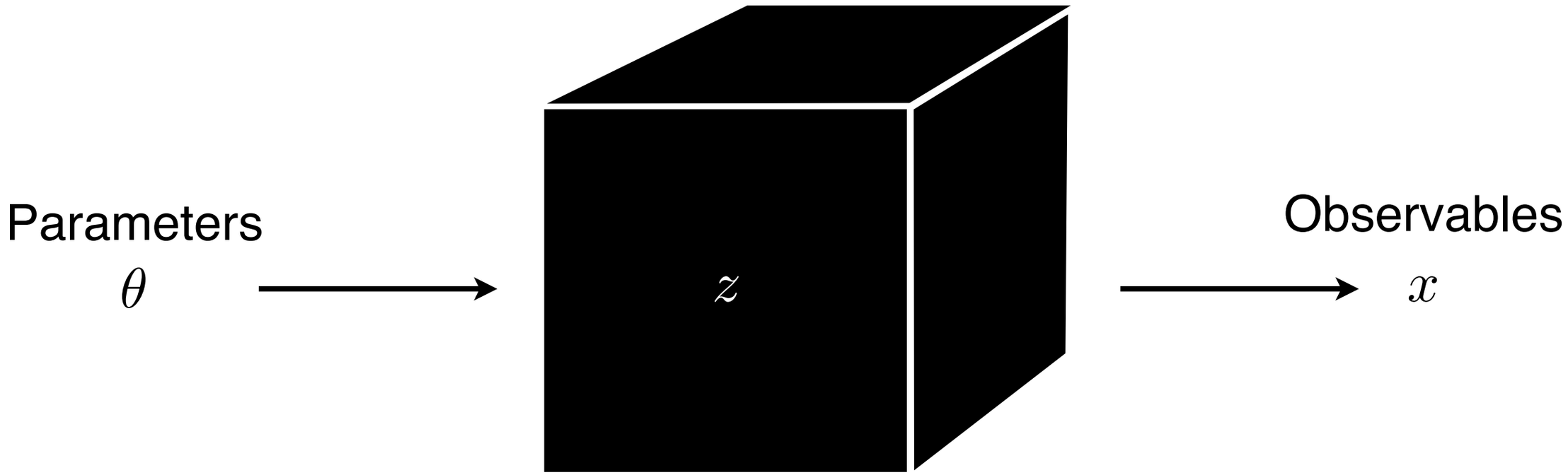
Observables

x

Prediction (simulation):

- Well-understood mechanistic model
- Simulator can generate samples

LIKELIHOOD-FREE INFERENCE



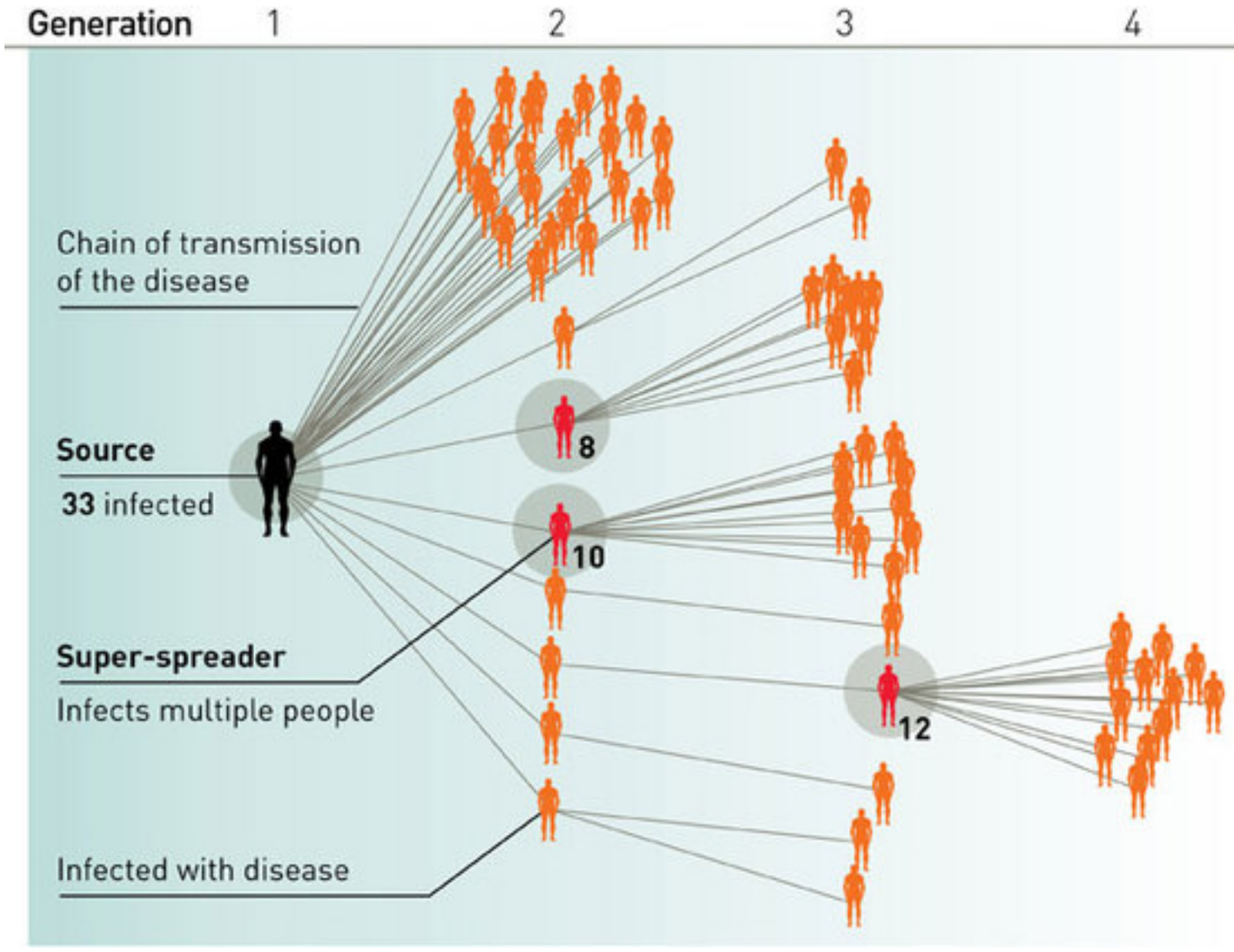
Prediction (simulation):

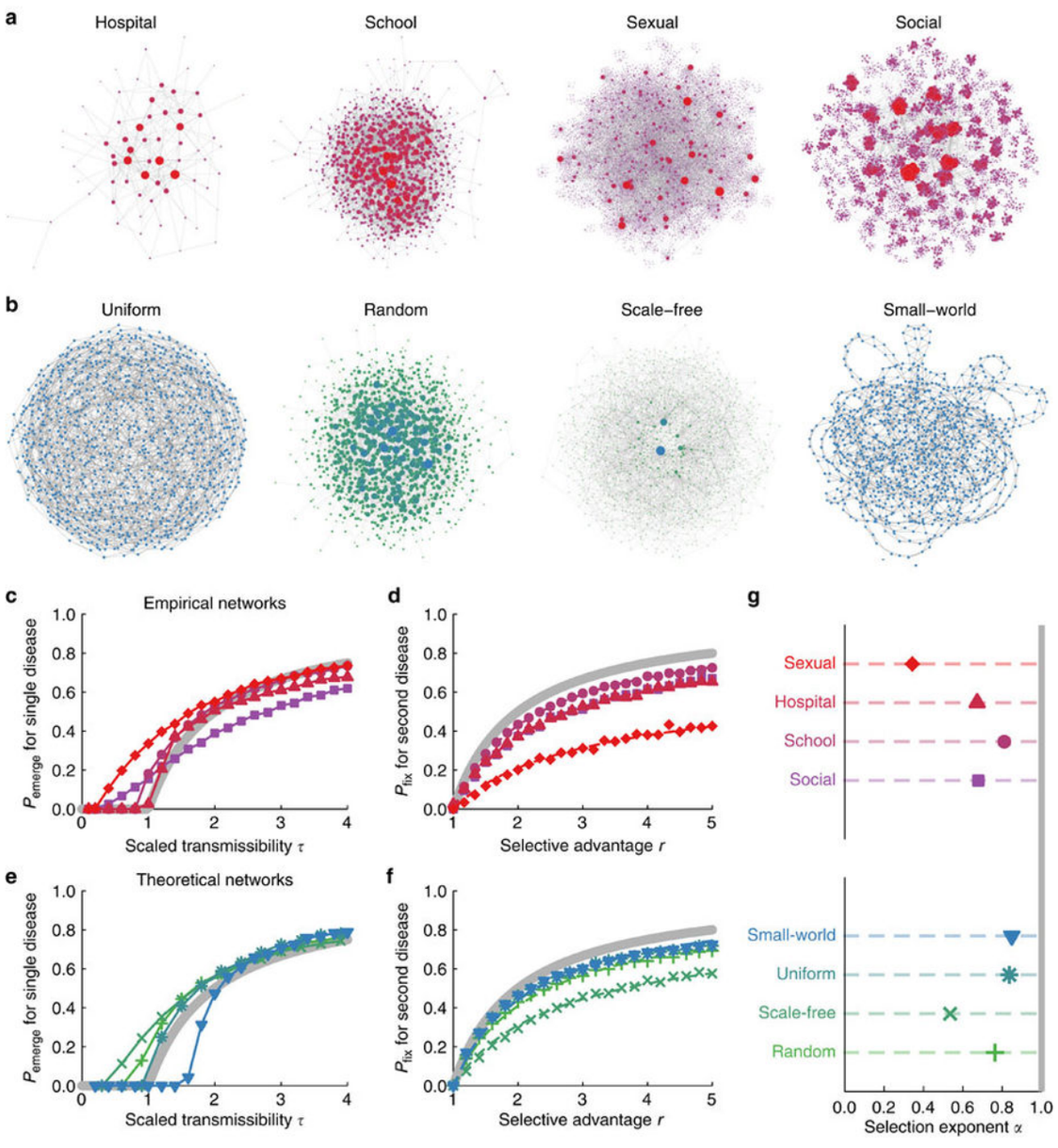
- Well-understood mechanistic model
- Simulator can generate samples

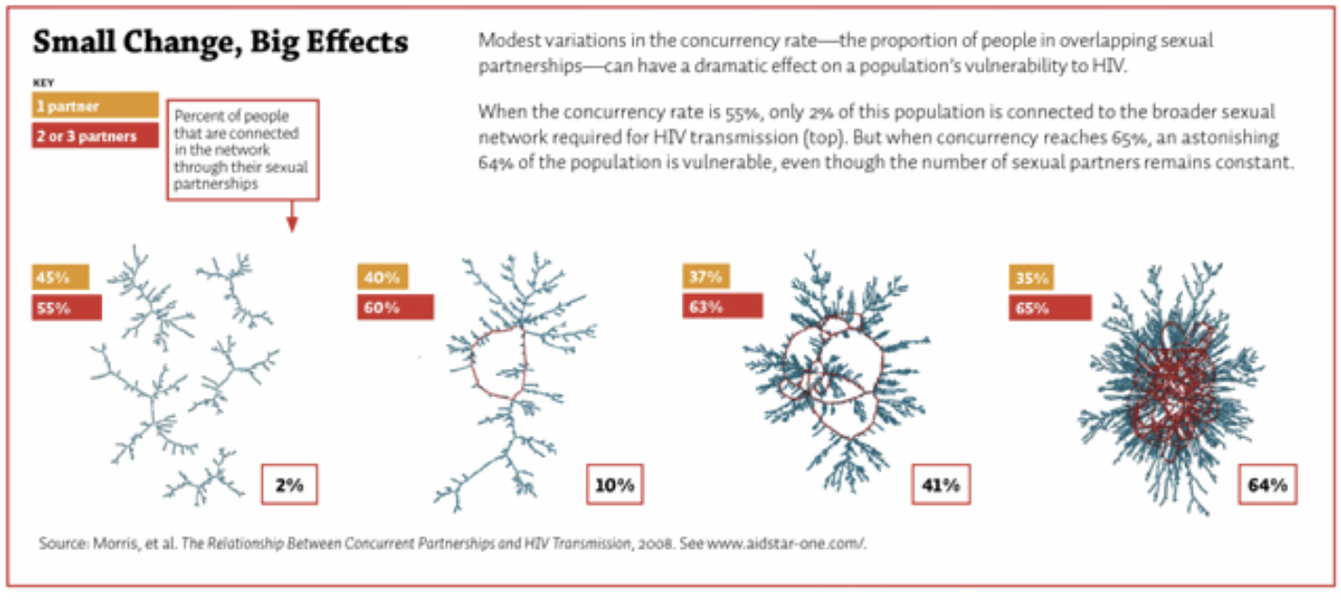
Inference:

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

EPIDEMIOLOGY & POPULATION GENETICS



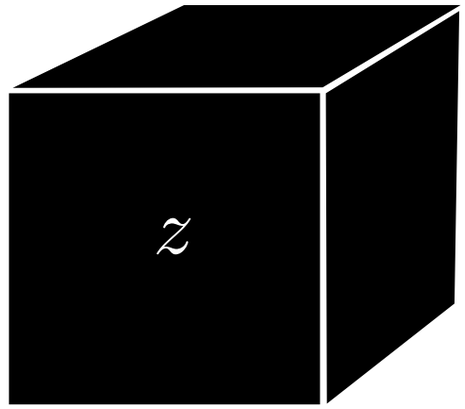




COSMOLOGICAL N-BODY SIMULATIONS

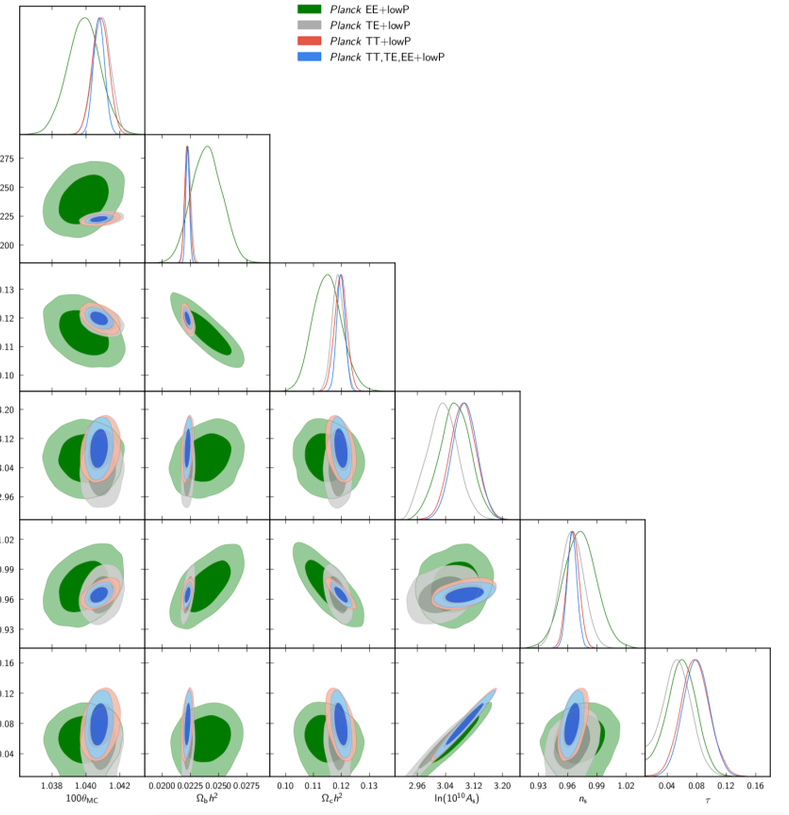
Parameters

θ

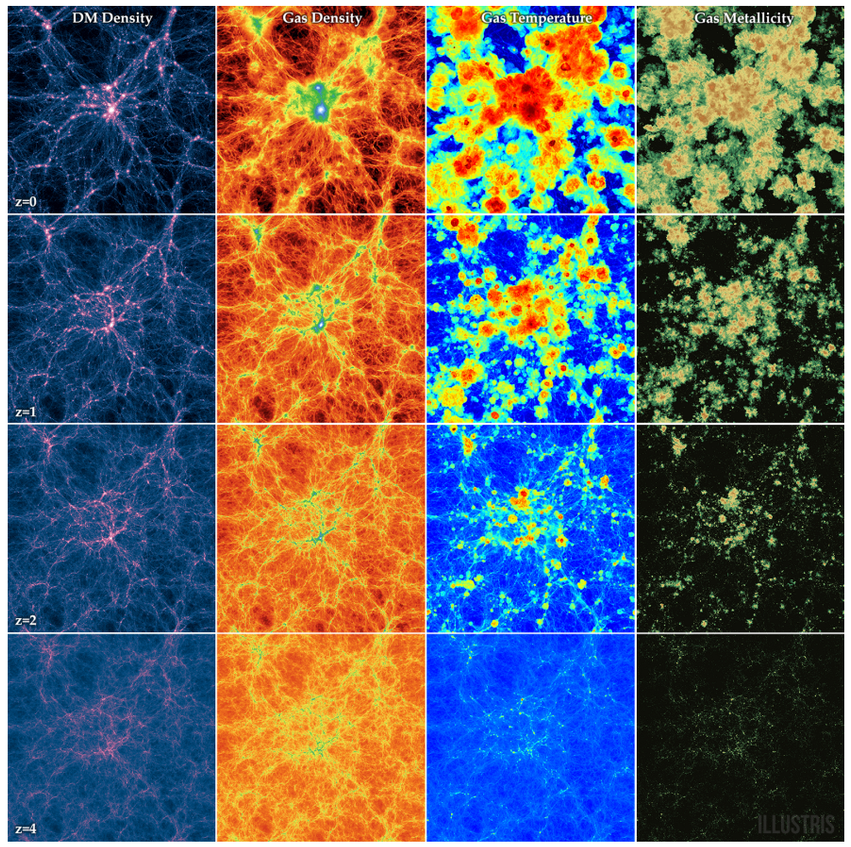


Observables

x



[Source: Planck 1502.01589]



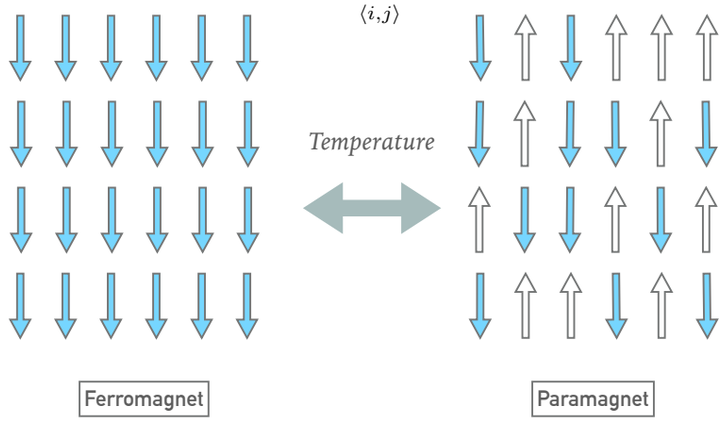
[Source: Illustris 1405.2921]

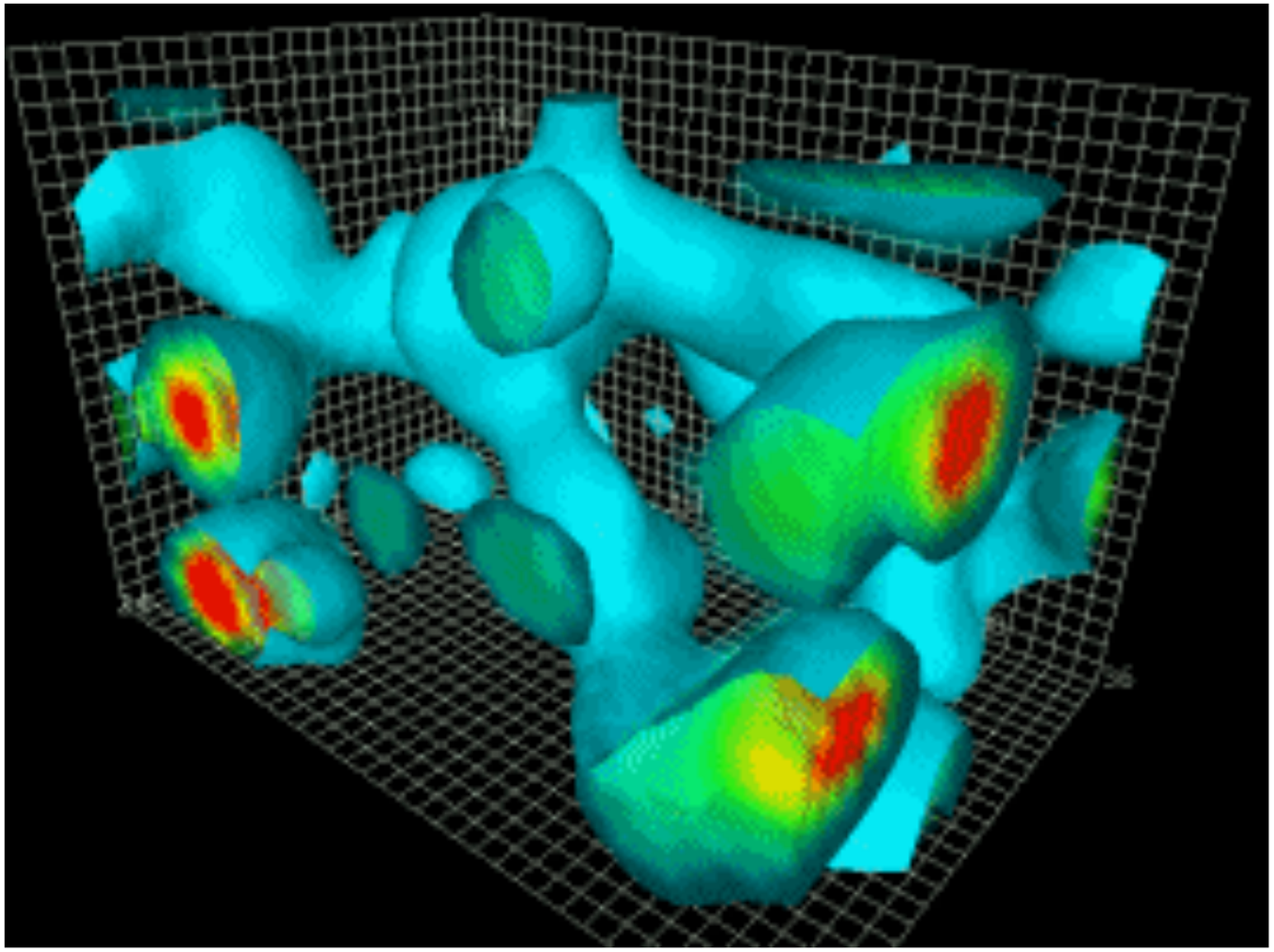
LATTICE FIELD THEORY

PHASES, PHASE TRANSITIONS, AND THE ORDER PARAMETER

Ising ferromagnet in two dimensions

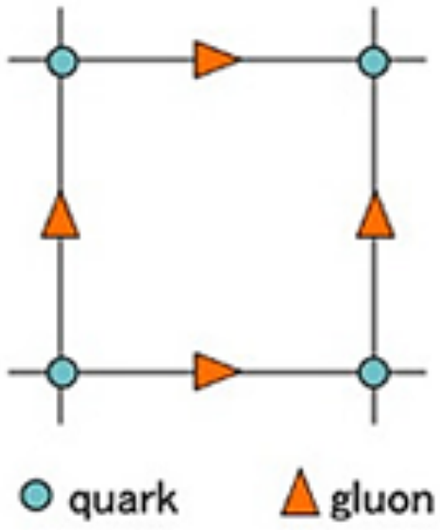
$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

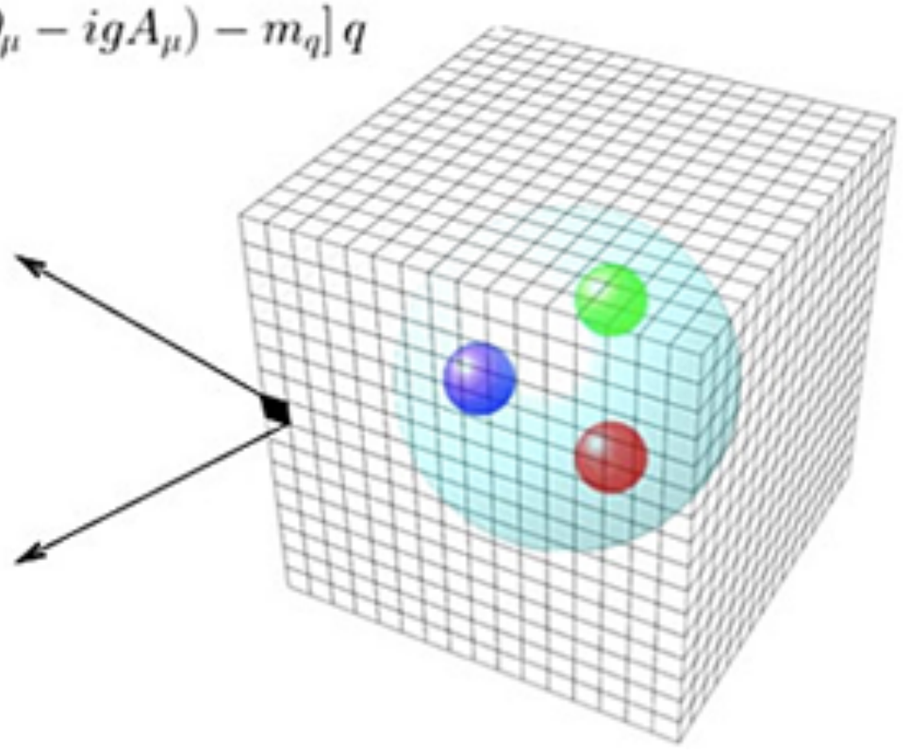




QCD Lagrangian

$$\mathcal{L} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} + \sum_{q=u,d,s,c,b,t} \bar{q} \left[i\gamma^\mu (\partial_\mu - igA_\mu) - m_q \right] q$$



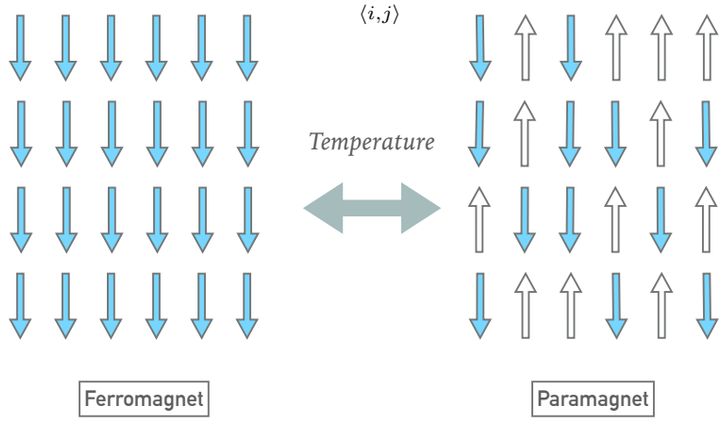


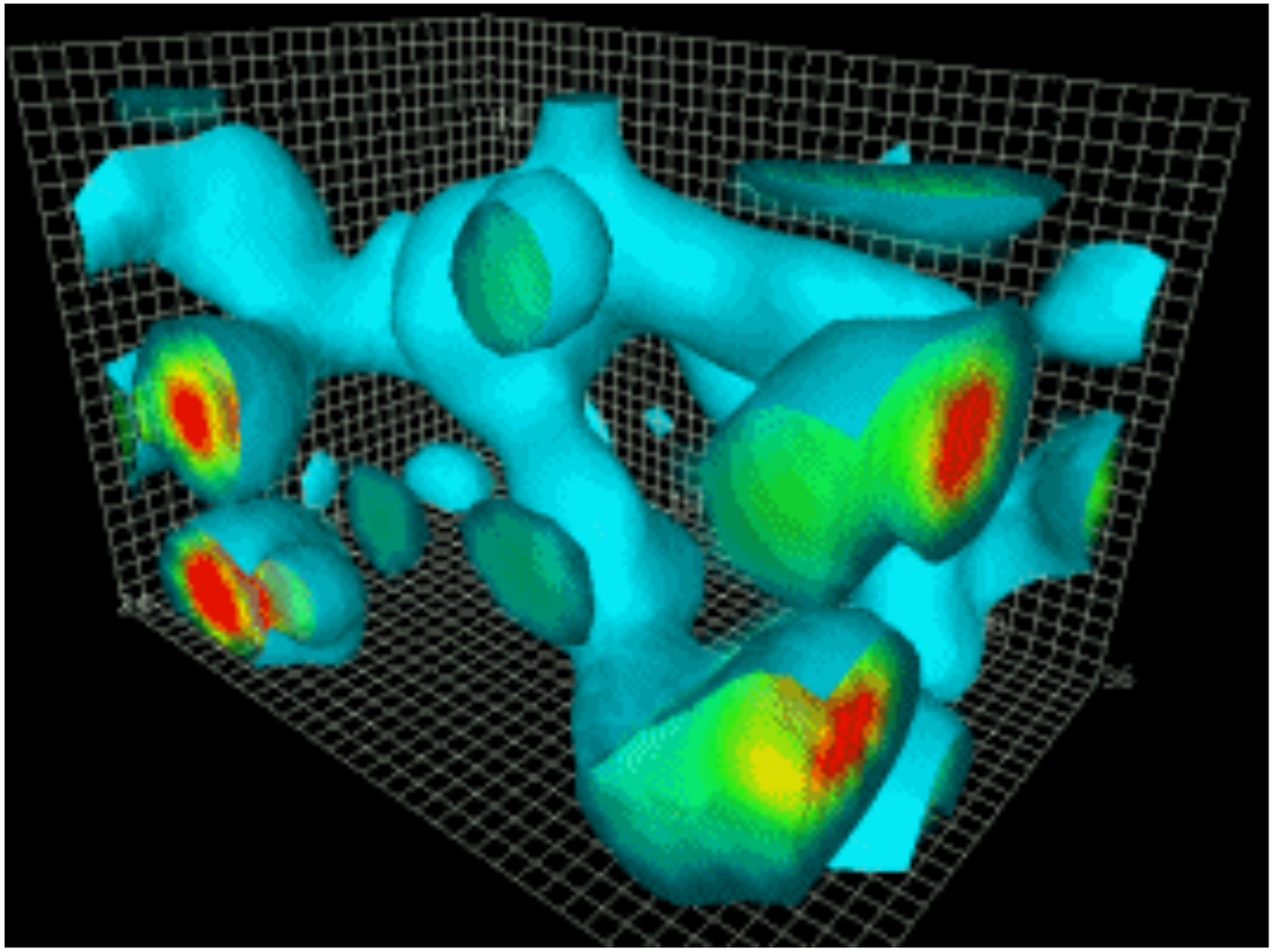
LATTICE FIELD THEORY

PHASES, PHASE TRANSITIONS, AND THE ORDER PARAMETER

Ising ferromagnet in two dimensions

$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$





QCD Lagrangian

$$\mathcal{L} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} + \sum_{q=u,d,s,c,b,t} \bar{q} \left[i\gamma^\mu (\partial_\mu - igA_\mu) - m_q \right] q$$