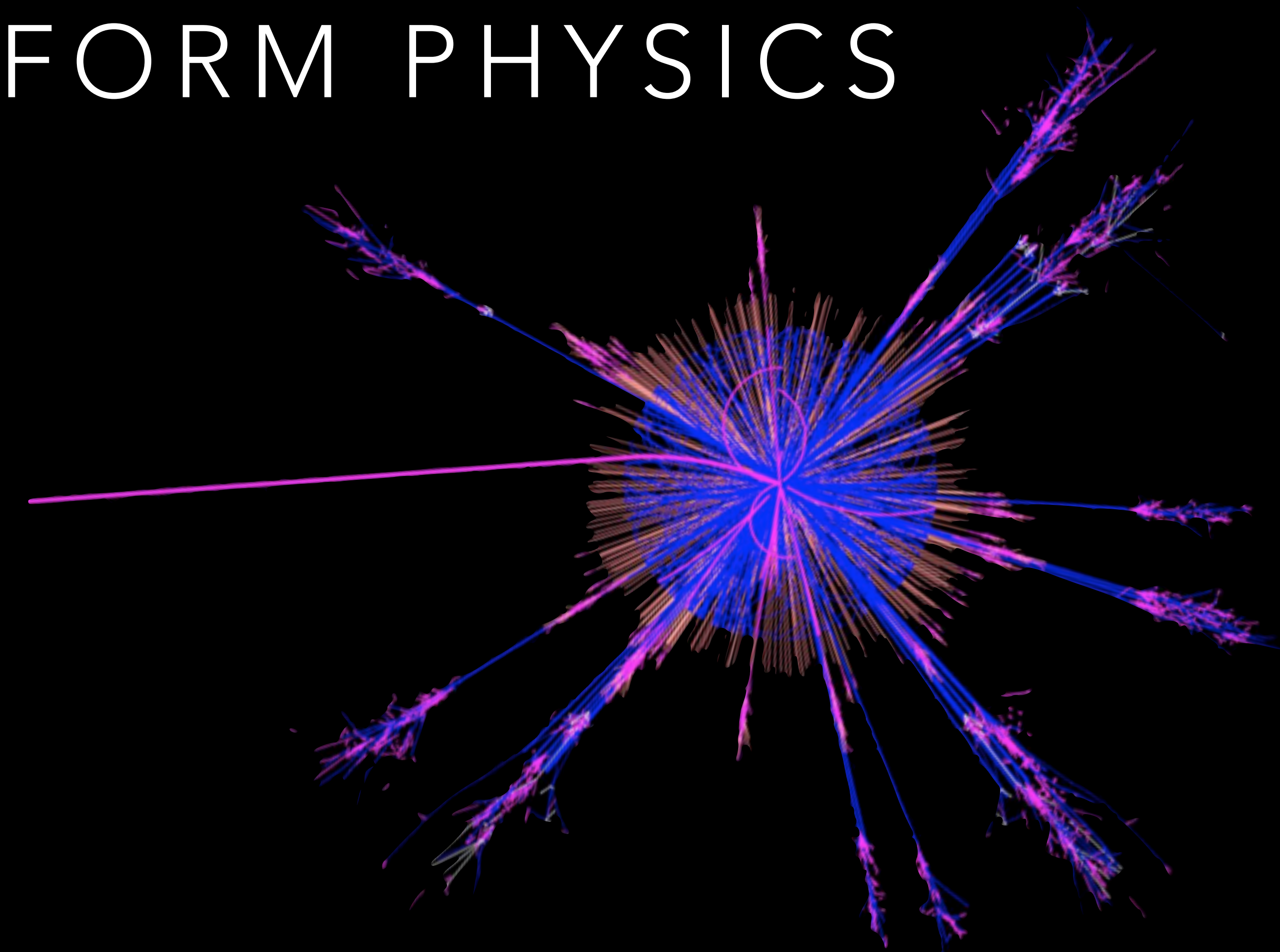
NYU Center for Data Science

Center for Cosmology and Particle Physics

WATCH THIS SPACE:

EMERGING TRENDS AND TECHNIQUES THAT COULD TRANSFORM PHYSICS AND ASTRONOMY



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

OUTLINE

Opening remarks on physics, stats & modeling

Gaussian Processes

Likelihood-free inference & implicit models

Adversarial Training & Systematics

Incorporating physics knowledge

From Reproducibility to Reusability

Black Box Optimization

BUILDING A MODEL OF THE DATA

Before one can discuss statistical tests, one must have a "**model**" for the data.

- by "model", I mean the full structure of P(data | parameters)
  - holding parameters fixed gives a PDF for data
  - provides ability to generate pseudo-data (via Monte Carlo)
  - holding data fixed gives a **likelihood function** for parameters
    - note, likelihood function is not as general as the full model because it doesn't allow you to generate pseudo-data

Both Bayesian and Frequentist methods start with the model

- it's the objective part that everyone can agree on
- it's the place where our physics knowledge, understanding, and intuition comes in
- building a better model is the best way to improve your statistical procedure

THE SCIENTIFIC NARRATIVE

The model can be seen as a quantitative summary of the analysis

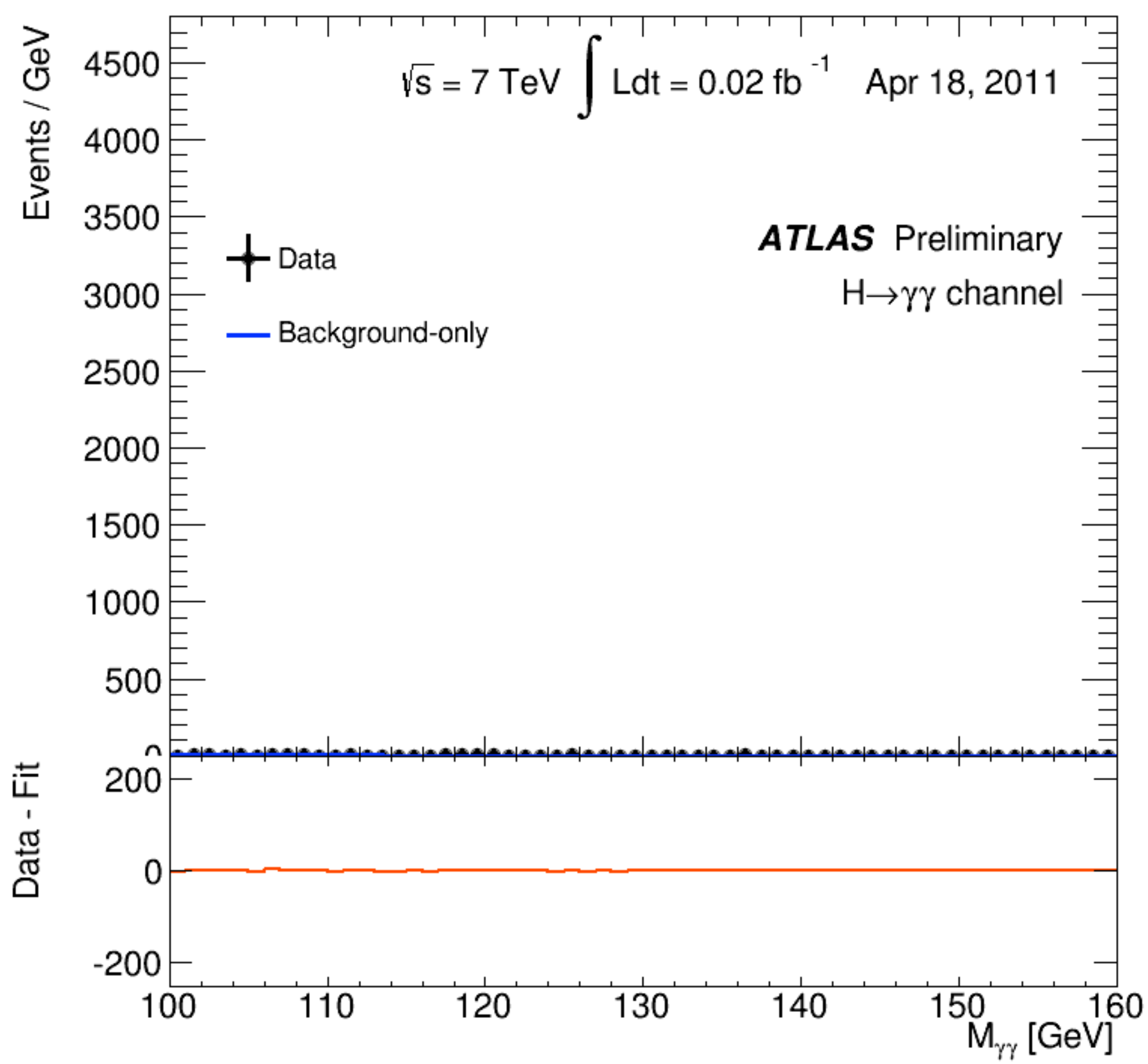
- If you were asked to justify your modeling, you would tell a **story** about why you know what you know
  - based on previous results and studies performed along the way
- the quality of the result is largely tied to how convincing this story is and how tightly it is connected to model

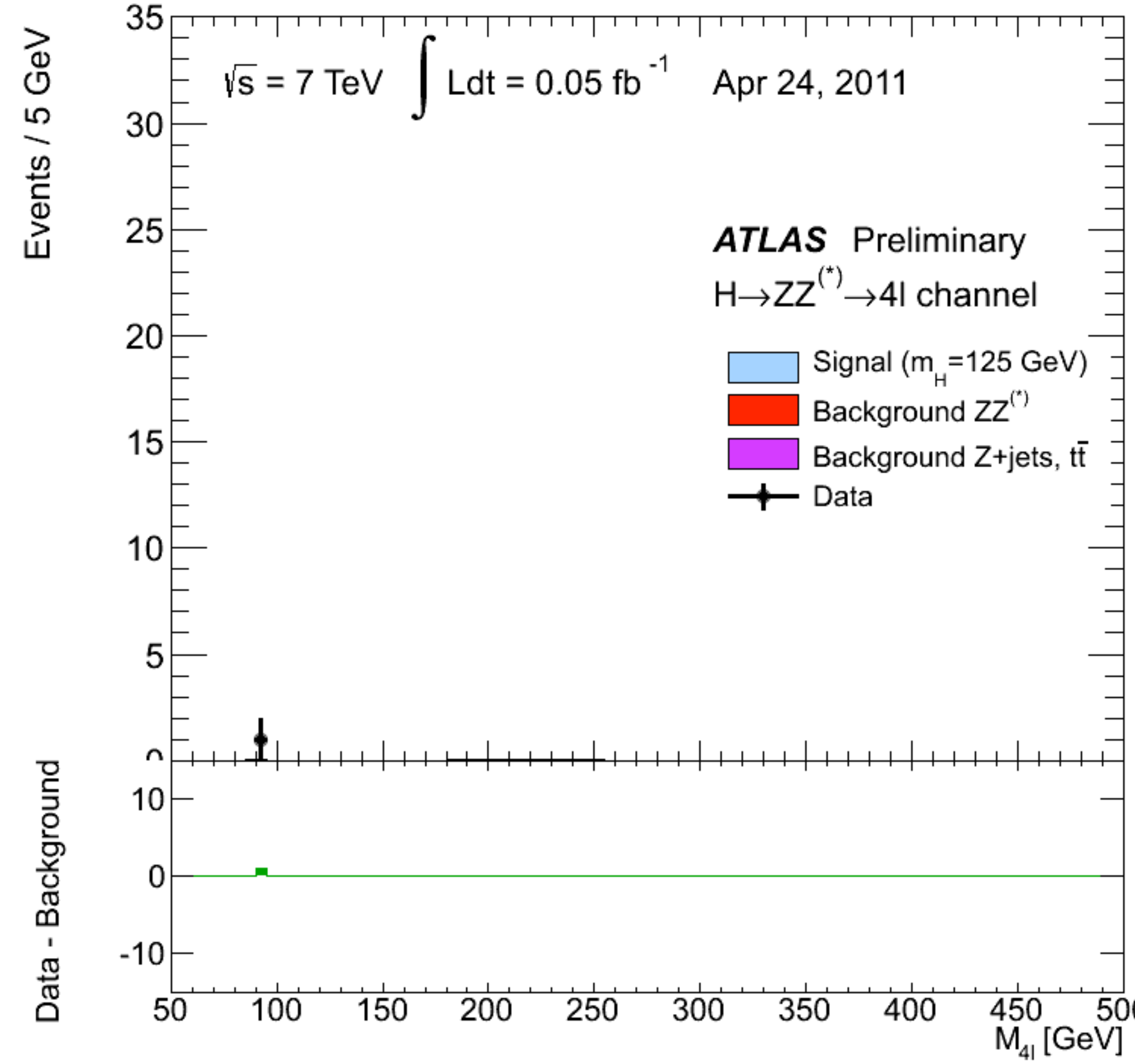
Common "narrative styles"

- The "Monte Carlo Simulation" narrative
- The "Data Driven" narrative
- The "Effective Modeling" narrative

Real-life analyses often use a mixture of these

Discovery!





Effective Model Narrative

polynomial fit

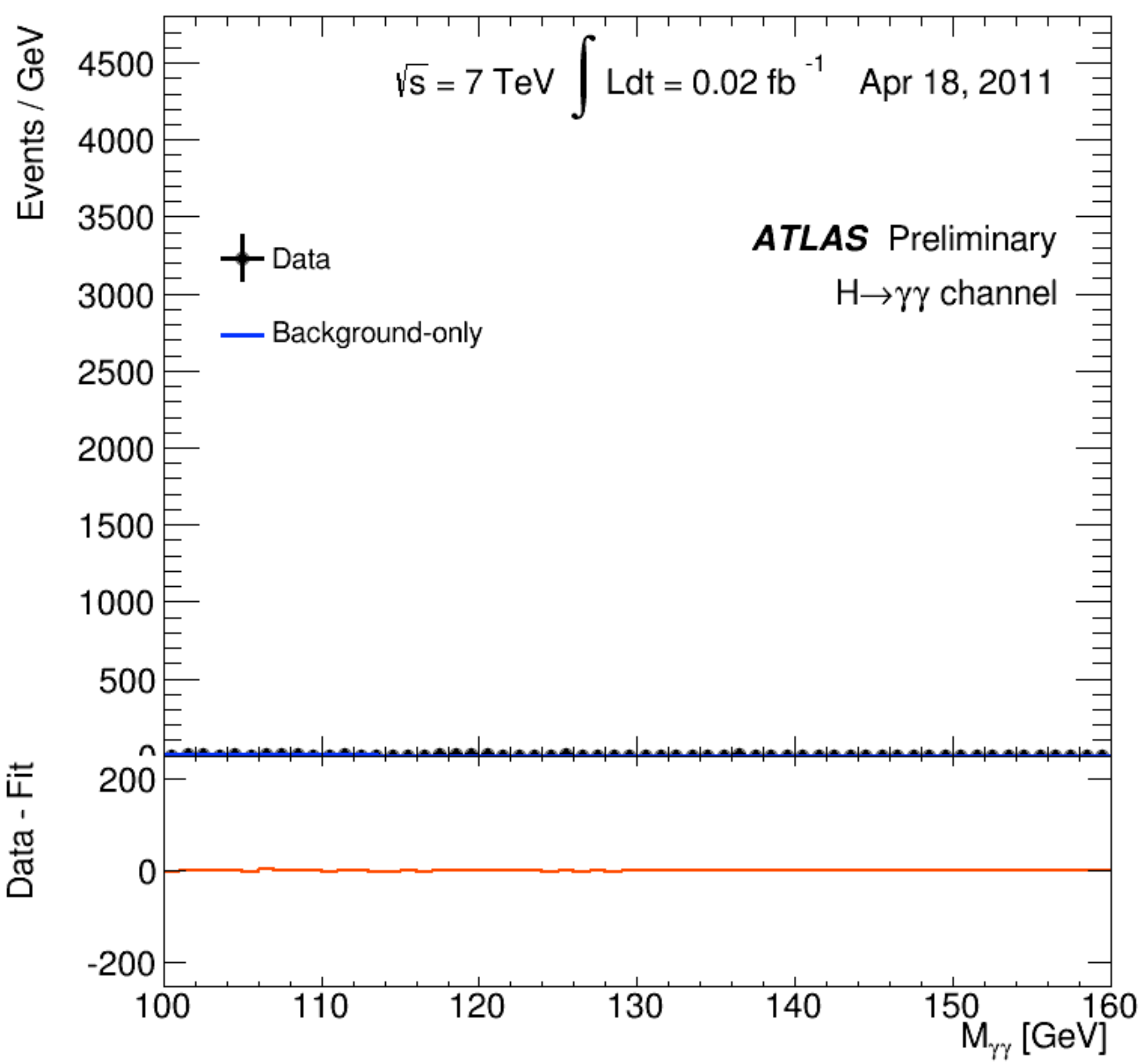
to smooth background

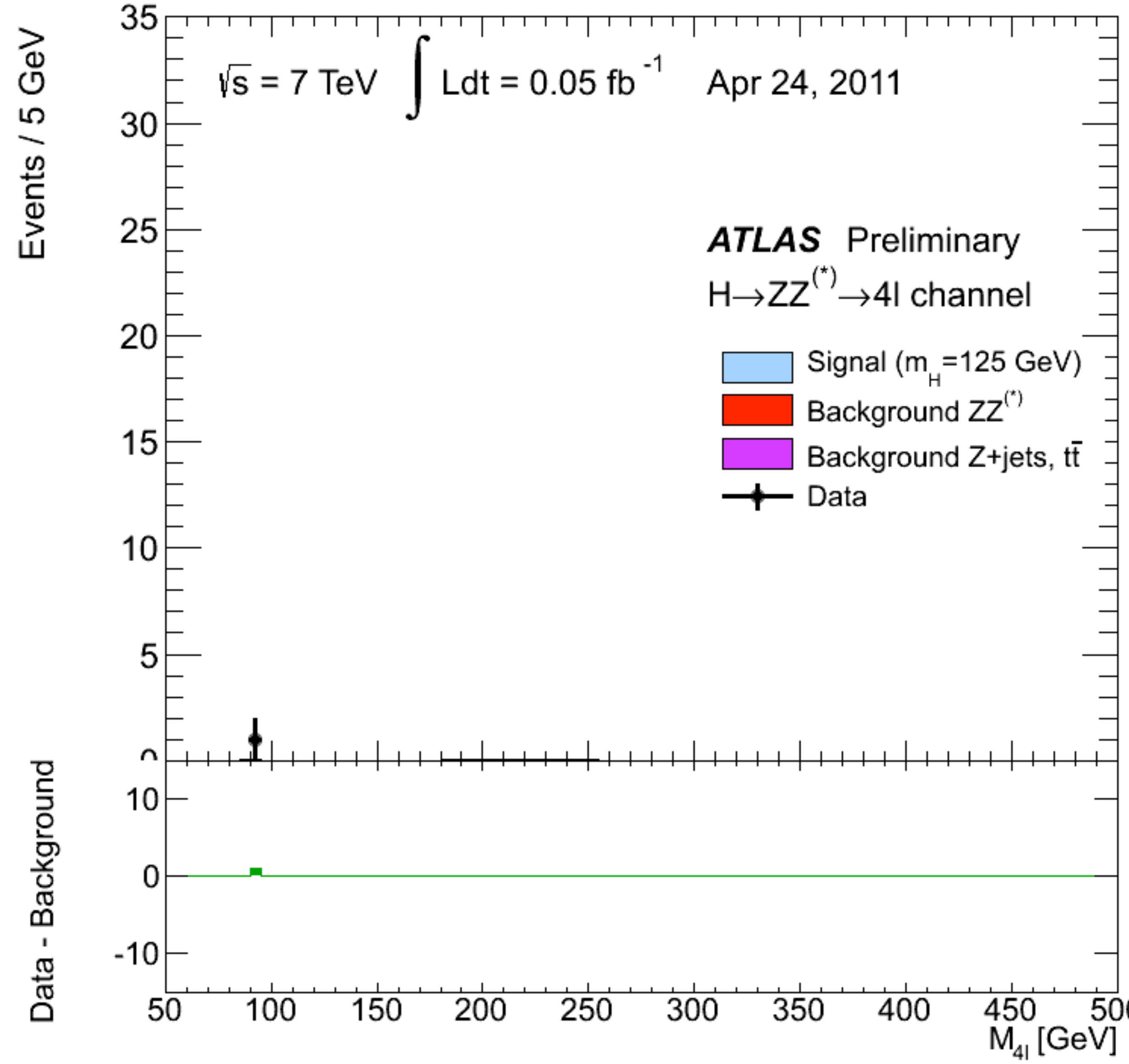
Simulation Narrative

template histograms

from simulation

Discovery!



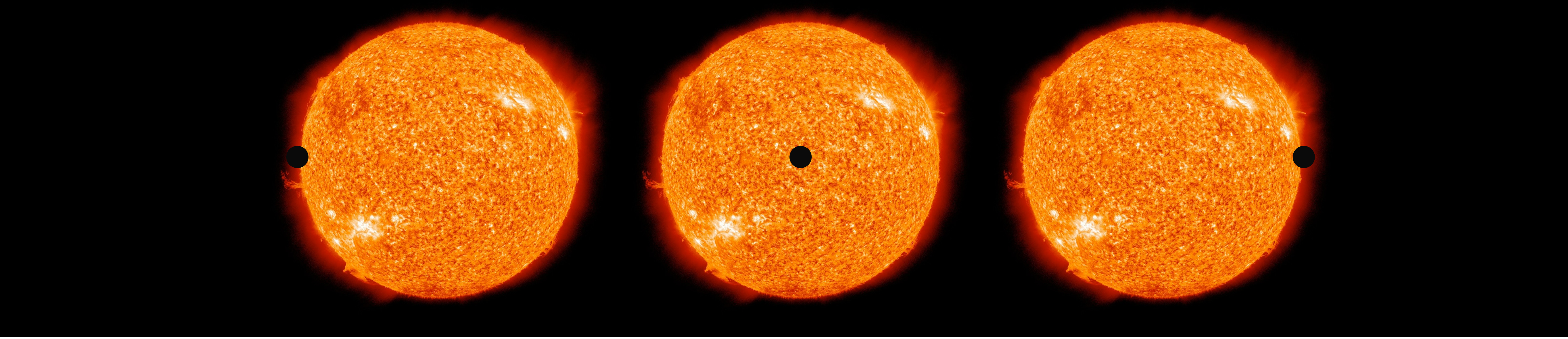


Effective Model Narrative
polynomial fit
to smooth background

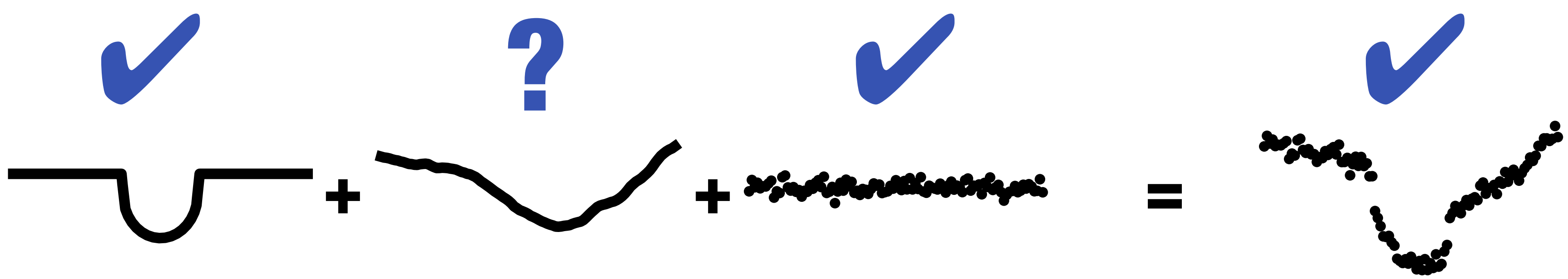
Simulation Narrative
template histograms
from simulation

Gaussian Processes
(Effective Model / Surrogates)

[a few slides by Dan Foreman-Mackey from DS@LHC]



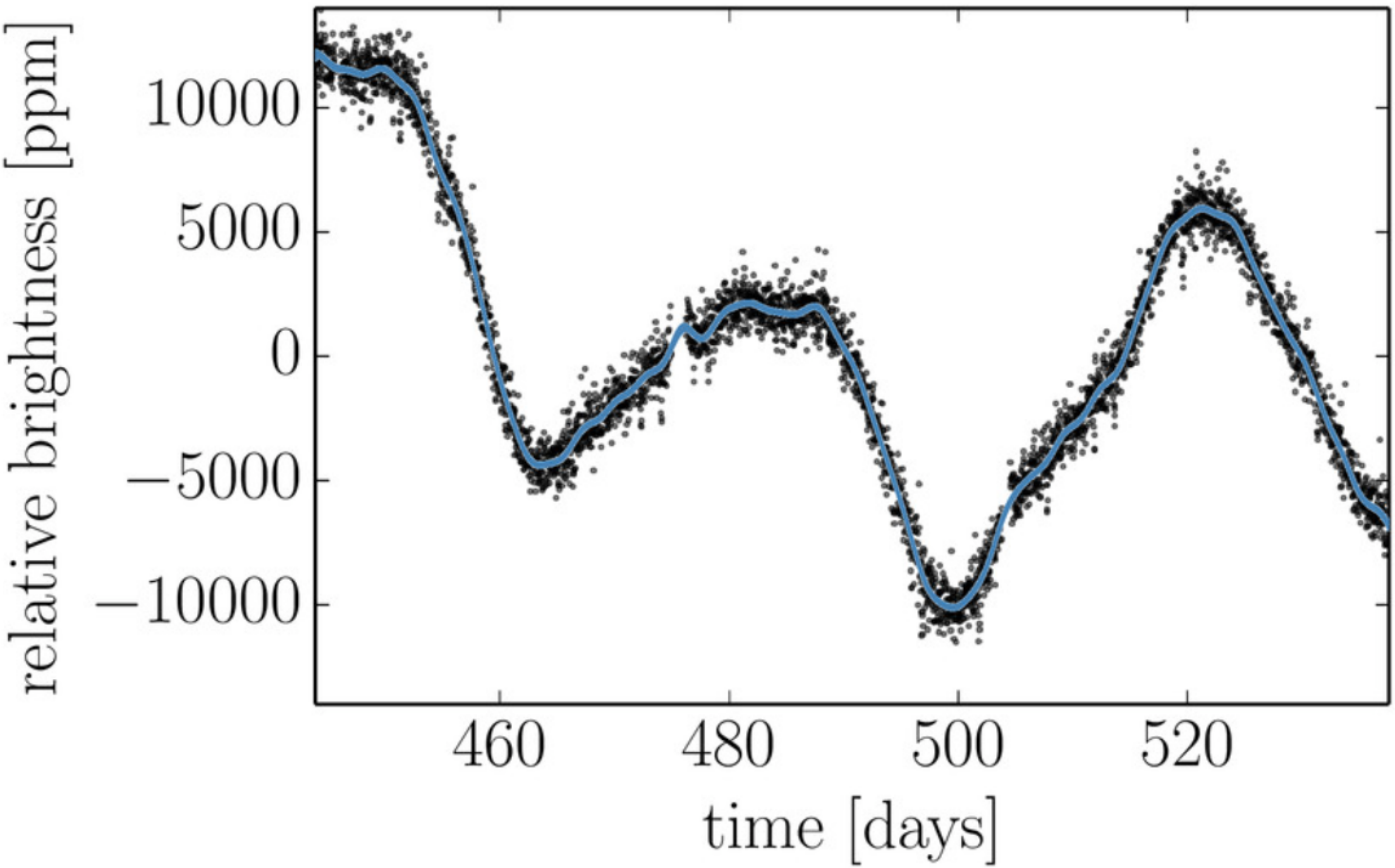
The anatomy of a transit observation



signal   variability   noise   data

astrophysics and spacecraft

AN EXOPLANET EXAMPLE



https://speakerdeck.com/dfm/pydata-time-series-analysis-gps-and-exoplanets

the data are drawn from one

HUGE*

Gaussian

* the dimension is the number of data points.

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}),\, K_{\boldsymbol{\alpha}}(\boldsymbol{x},\,\boldsymbol{\sigma})\right)$$

where

$$[K_{\boldsymbol{\alpha}}(\boldsymbol{x},\,\boldsymbol{\sigma})]_{ij} = {\sigma_i}^2\,\delta_{ij} + \underbrace{k_{\boldsymbol{\alpha}}(x_i,\,x_j)}_{\text{kernel function}}$$

kernel function

(where the magic happens)

see: gaussianprocess.org/**gpml** github.com/**dfm/george**

$$\log p(\boldsymbol{y}\,|\,\boldsymbol{x},\,\boldsymbol{\sigma},\,\boldsymbol{\theta},\,\boldsymbol{\alpha}) = -\frac{1}{2}\,[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})]^{\mathrm{T}}\,K_{\boldsymbol{\alpha}}(\boldsymbol{x},\,\boldsymbol{\sigma})^{-1}\,[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})] - \frac{1}{2}\log\det K_{\boldsymbol{\alpha}}(\boldsymbol{x},\,\boldsymbol{\sigma}) - \frac{N}{2}\log 2\,\pi$$

where

$$[K_{\boldsymbol{\alpha}}(\boldsymbol{x},\,\boldsymbol{\sigma})]_{ij} = {\sigma_i}^2\,\delta_{ij} + k_{\boldsymbol{\alpha}}(x_i,\,x_j)$$

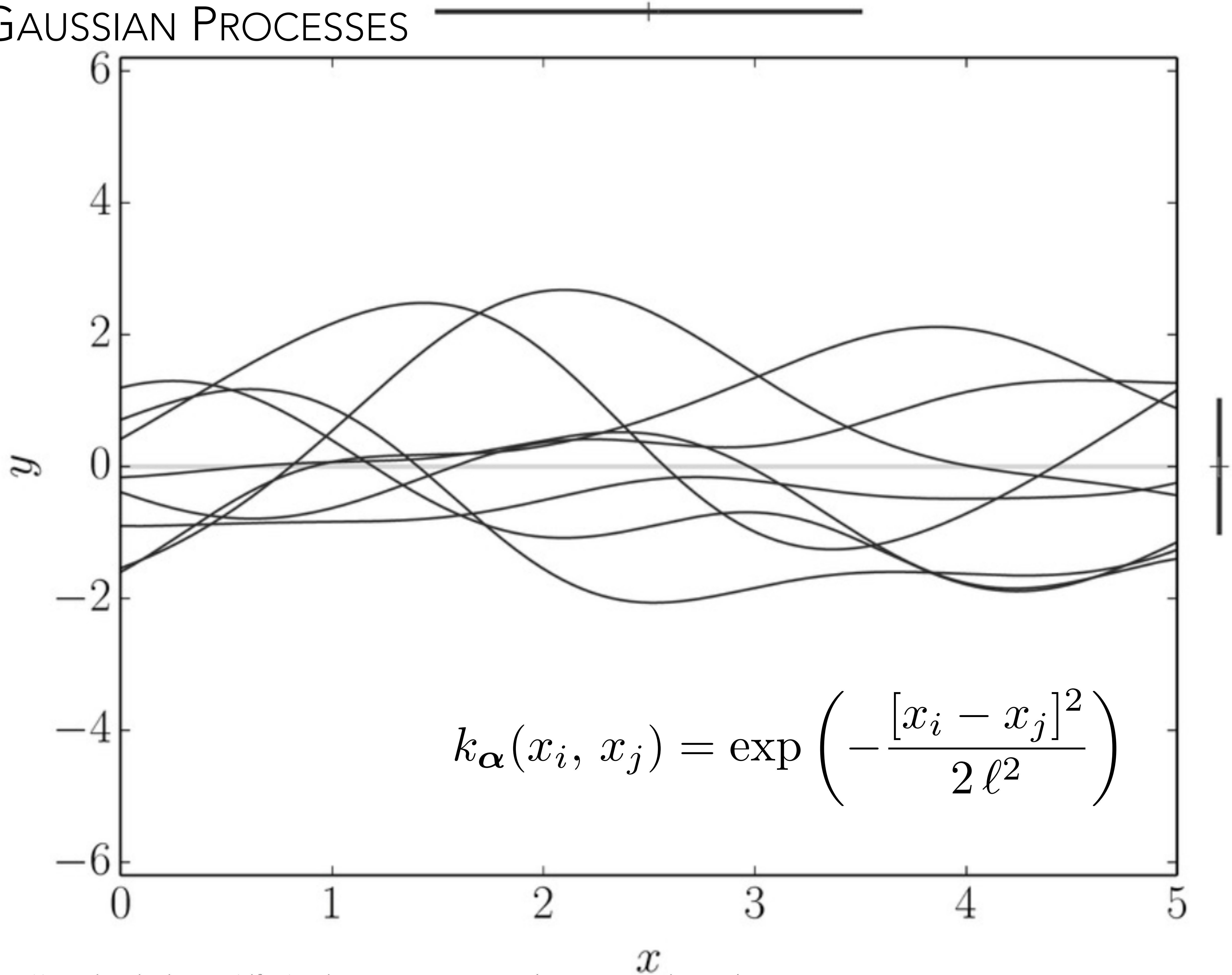
kernel function

(where the magic happens)

see: gaussianprocess.org/**gpml** github.com/**dfm/george**

$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \exp\left(-\frac{[x_i - x_j]^2}{2\,\ell^2}\right)$$

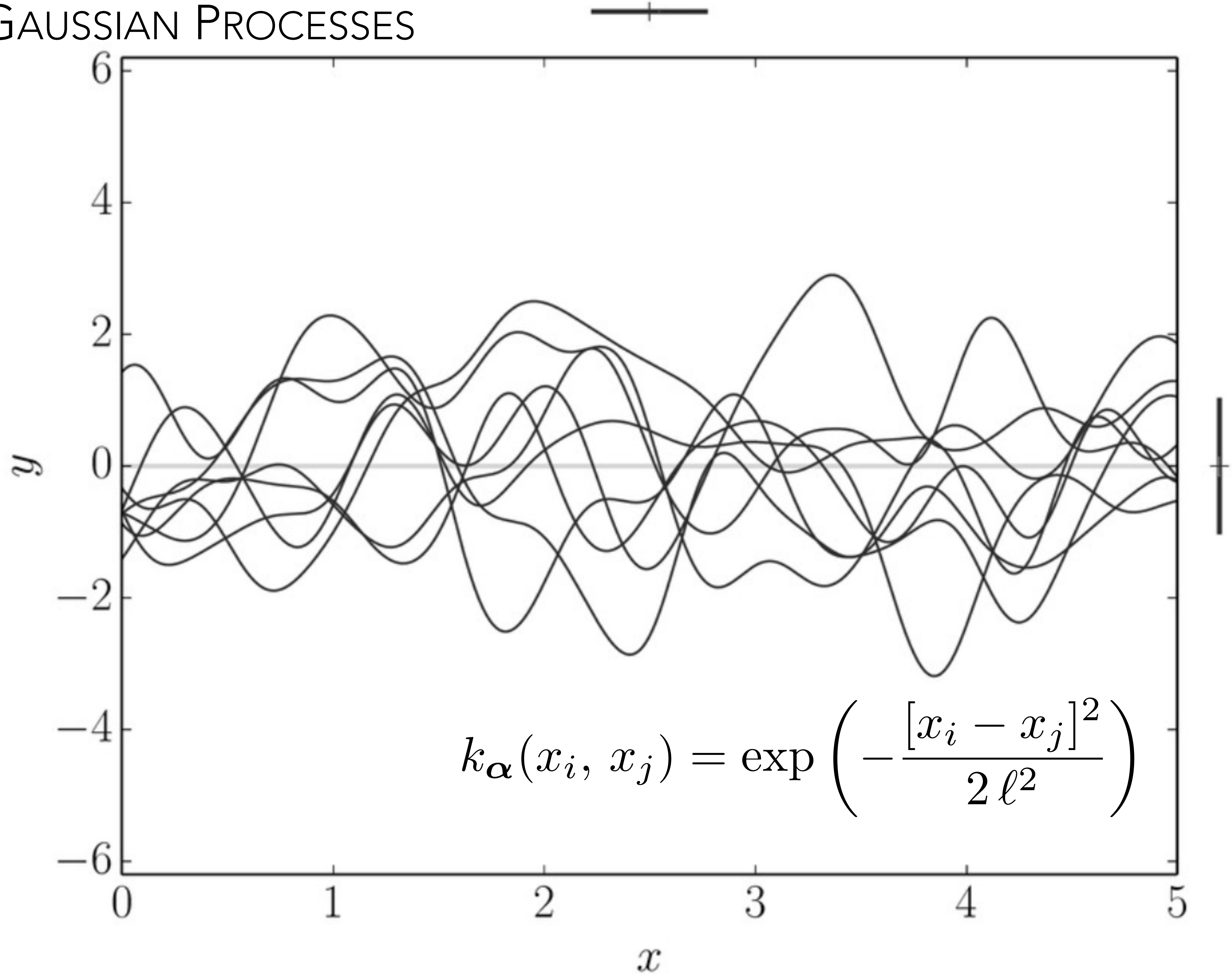
Gaussian Processes



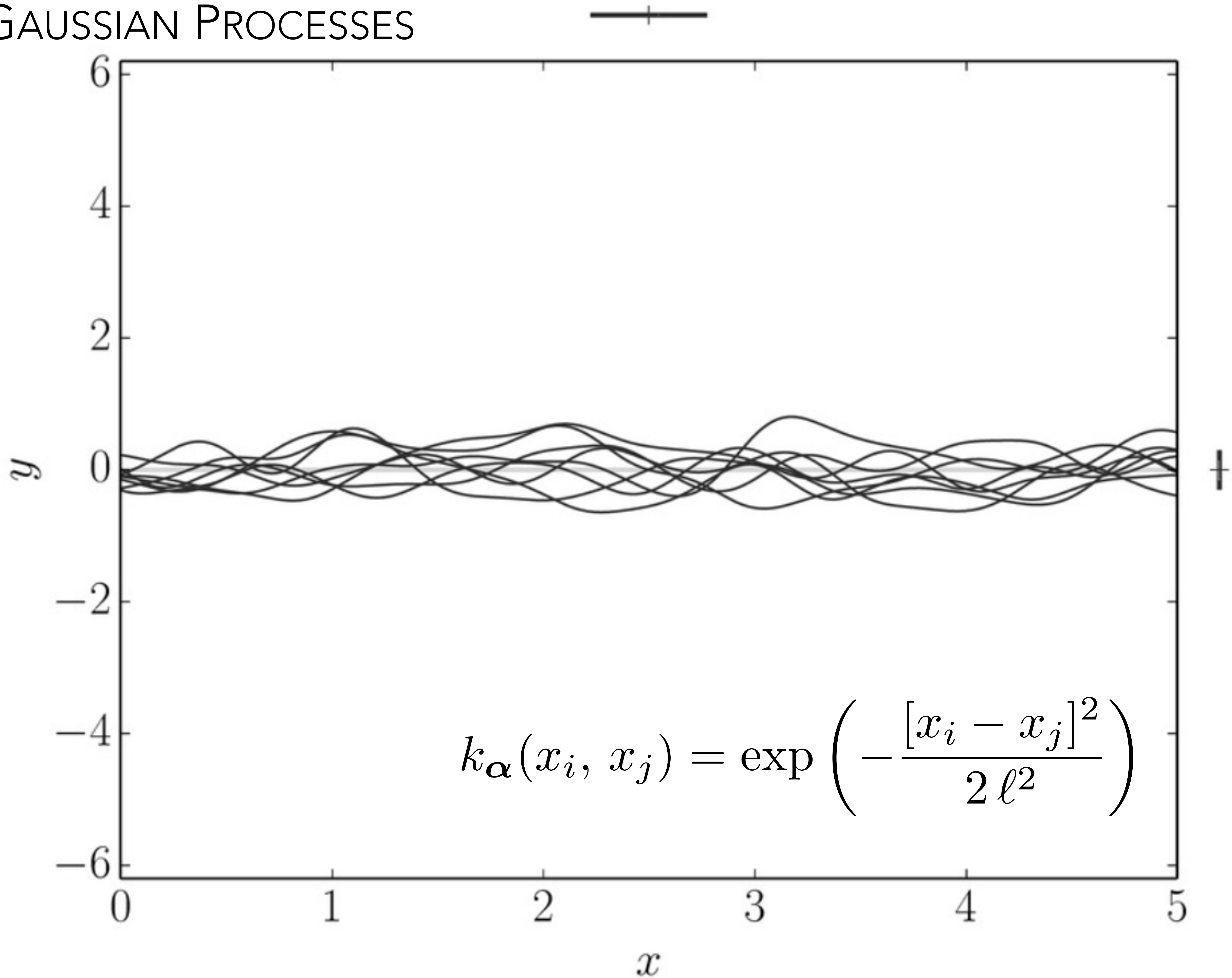
$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \exp\left(-\frac{[x_i - x_j]^2}{2\,\ell^2}\right)$$

https://speakerdeck.com/dfm/pydata-time-series-analysis-gps-and-exoplanets

Gaussian Processes



https://speakerdeck.com/dfm/pydata-time-series-analysis-gps-and-exoplanets



$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \exp\left(-\frac{[x_i - x_j]^2}{2\,\ell^2}\right)$$

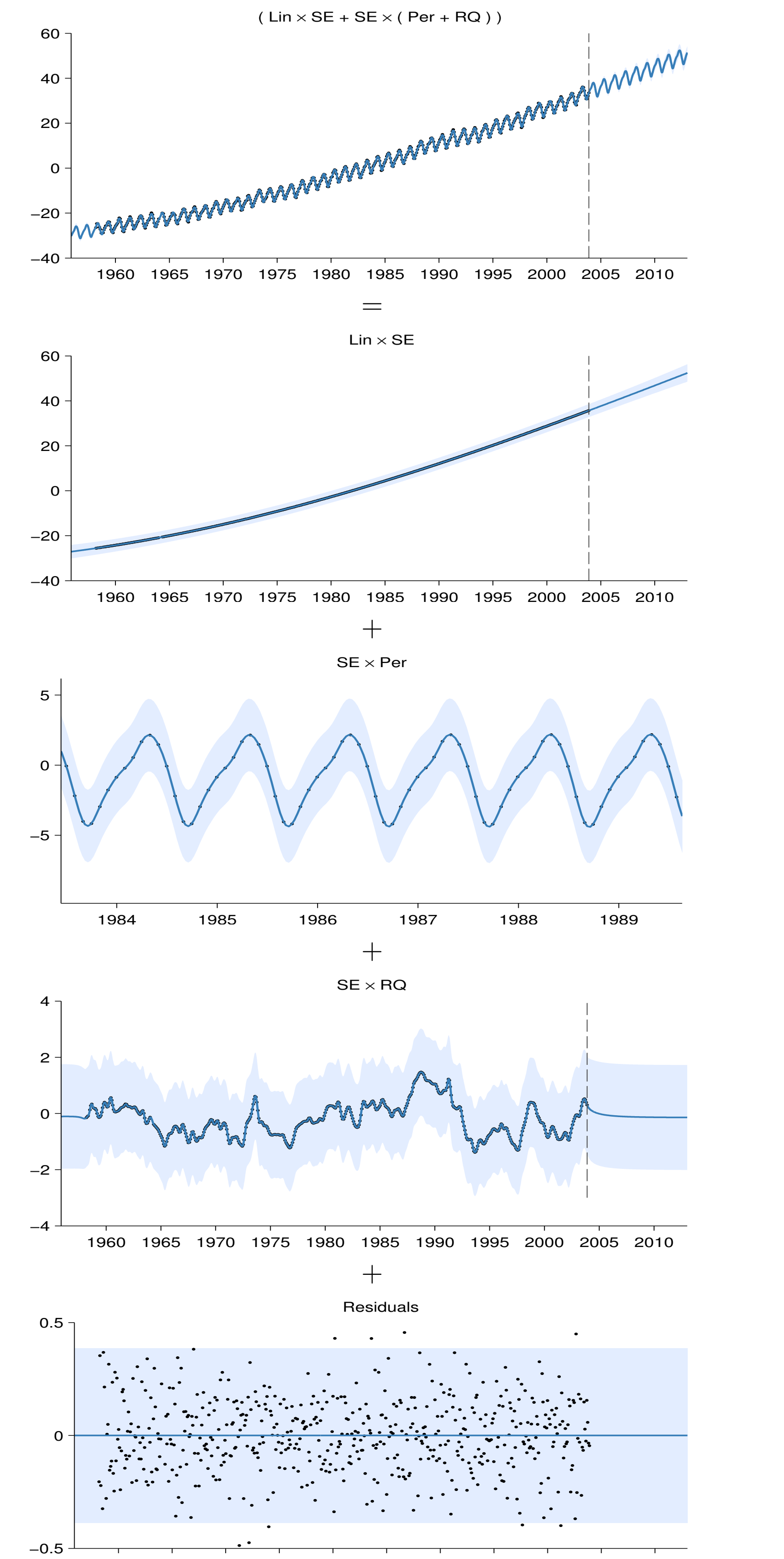
https://speakerdeck.com/dfm/pydata-time-series-analysis-gps-and-exoplanets

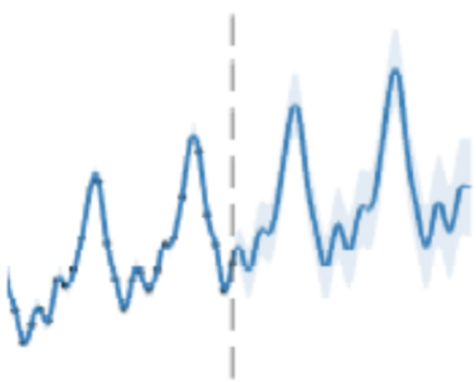
SEARCHING OVER SPACE OF MODELS

Vocabulary of kernels + grammar for composition

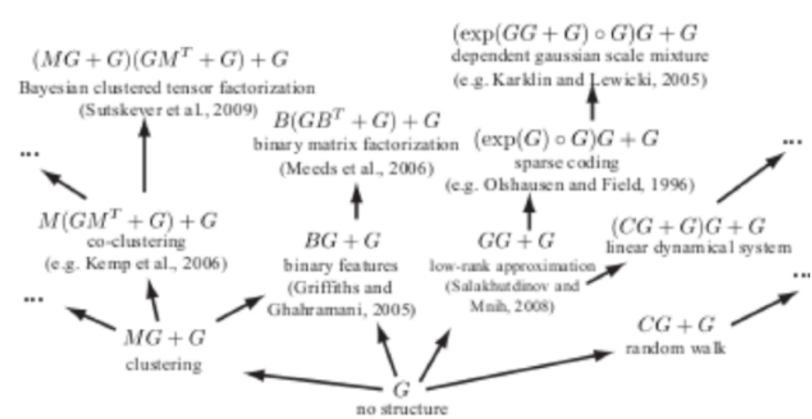
- physics goes into the construction of a "Kernel" that describes covariance of data

Mauna Loa atmospheric CO_2





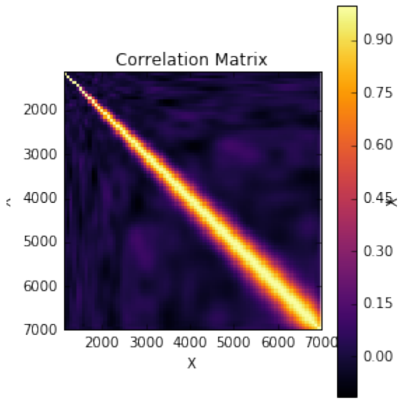
Structure Discovery in Nonparametric Regression through Compositional Kernel Search
David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani
International Conference on Machine Learning, 2013
pdf | code | poster | bibtex



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

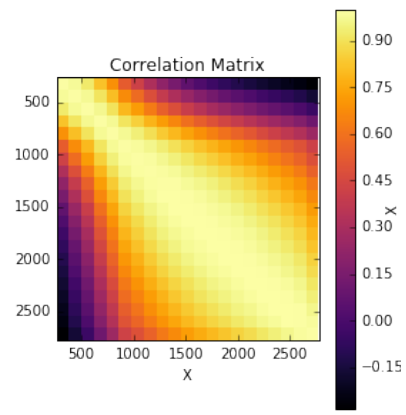
with Meghan Frate

Instead of fitting the dijet spectrum with an ad hoc 3-5 parameter function, use GP with kernel motivated from physics



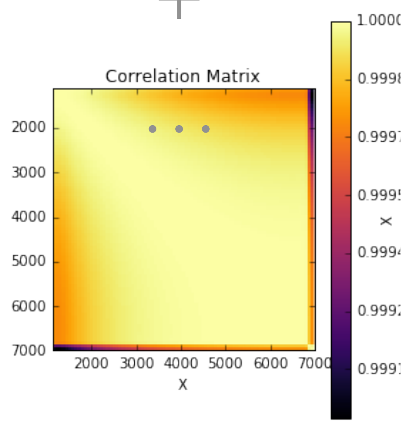
Final Kernel =

=



Poisson stats
+ Mass Resolution

+

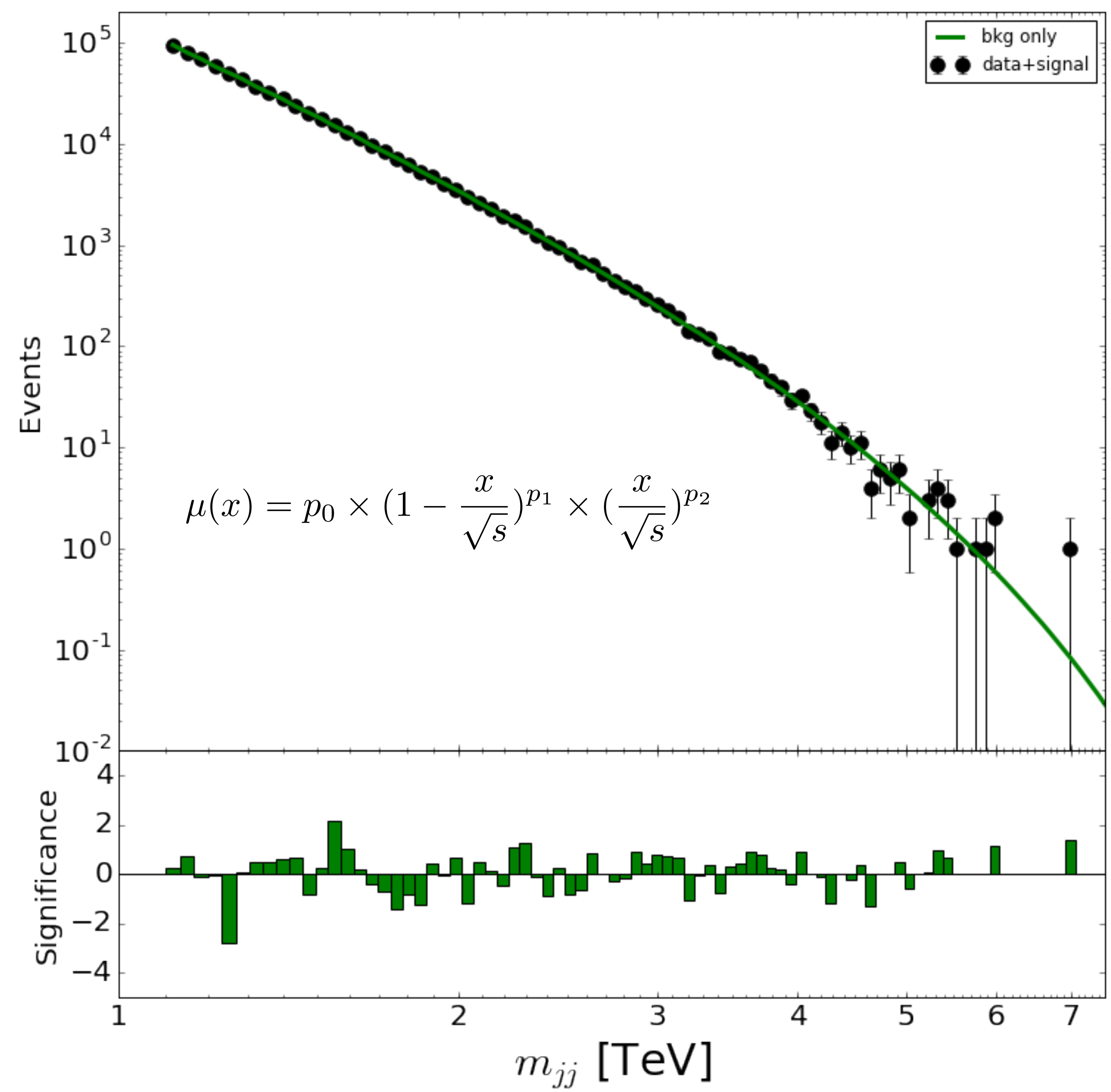


+ Parton Density Functions

+ Jet Energy Scale

+

. . .



Likelihood-free Inference /
Simulation-based Implicit Models

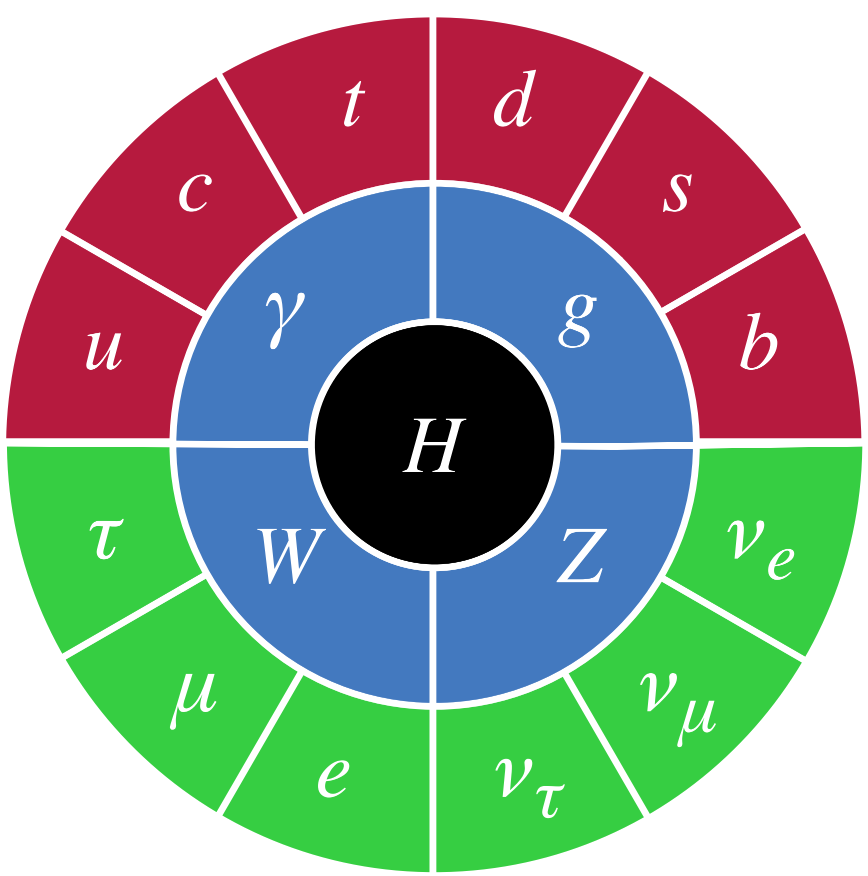
PARTICLE PHYSICS: 19 PARAMETERS

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

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

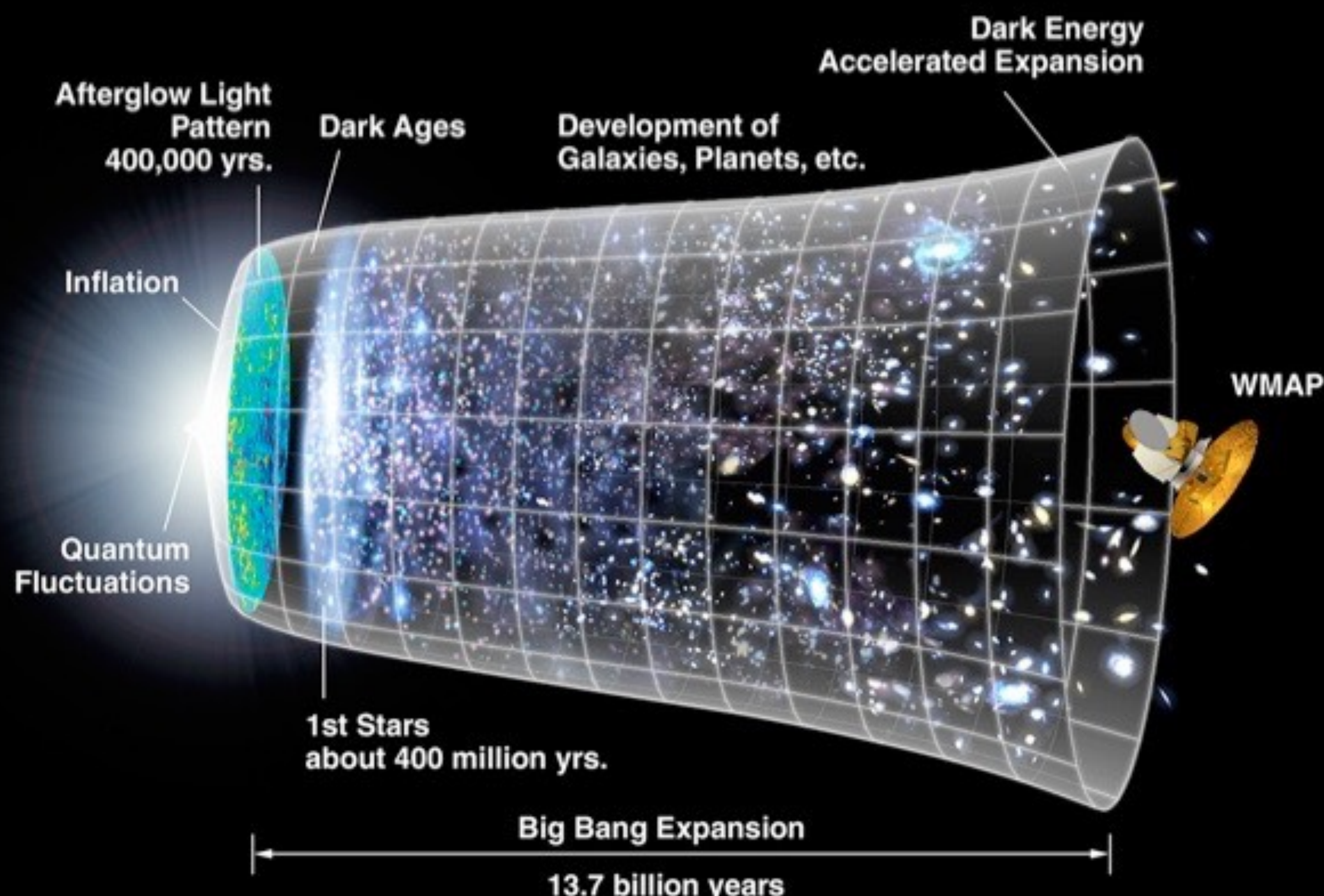
$$+ \underbrace{\frac{1}{2}\left|(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)\,\phi\right|^2 - V(\phi)}_{W^\pm, Z, \gamma, \text{and Higgs masses and couplings}}$$

$$+ \underbrace{g''(\bar{q}\gamma^\mu T_a q)\,G^a_\mu}_{\text{interactions between quarks and gluons}} + \underbrace{(G_1\bar{L}\phi R + G_2\bar{L}\phi_c R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$

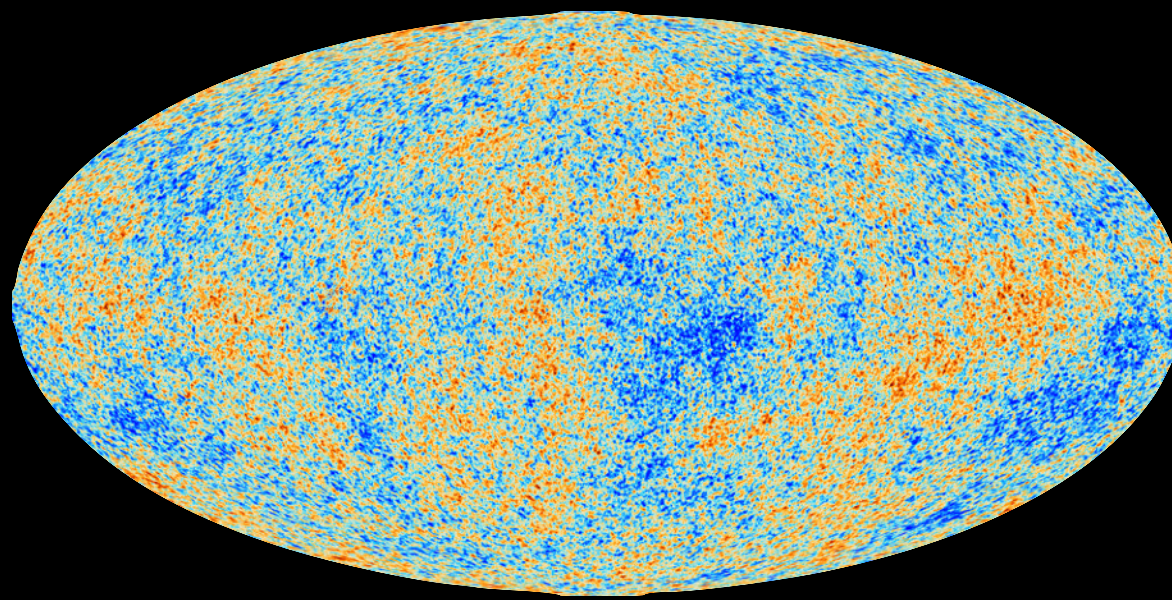


Symbol	Description	Value
m_e	Electron mass	511 keV
m_μ	Muon mass	105.7 MeV
m_τ	Tau mass	1.78 GeV
m_u	Up quark mass	1.9 MeV
m_d	Down quark mass	4.4 MeV
m_s	Strange quark mass	87 MeV
m_c	Charm quark mass	1.32 GeV
m_b	Bottom quark mass	4.24 GeV
m_t	Top quark mass	172.7 GeV
θ_{12}	CKM 12-mixing angle	13.1°
θ_{23}	CKM 23-mixing angle	2.4°
θ_{13}	CKM 13-mixing angle	0.2°
δ	CKM CP-violating Phase	0.995
g_1	U(1) gauge coupling	0.357
g_2	SU(2) gauge coupling	0.652
g_3	SU(3) gauge coupling	1.221
θ_{QCD}	QCD vacuum angle	~0
v	Higgs vacuum expectation value	246 GeV
m_H	Higgs mass	125 GeV

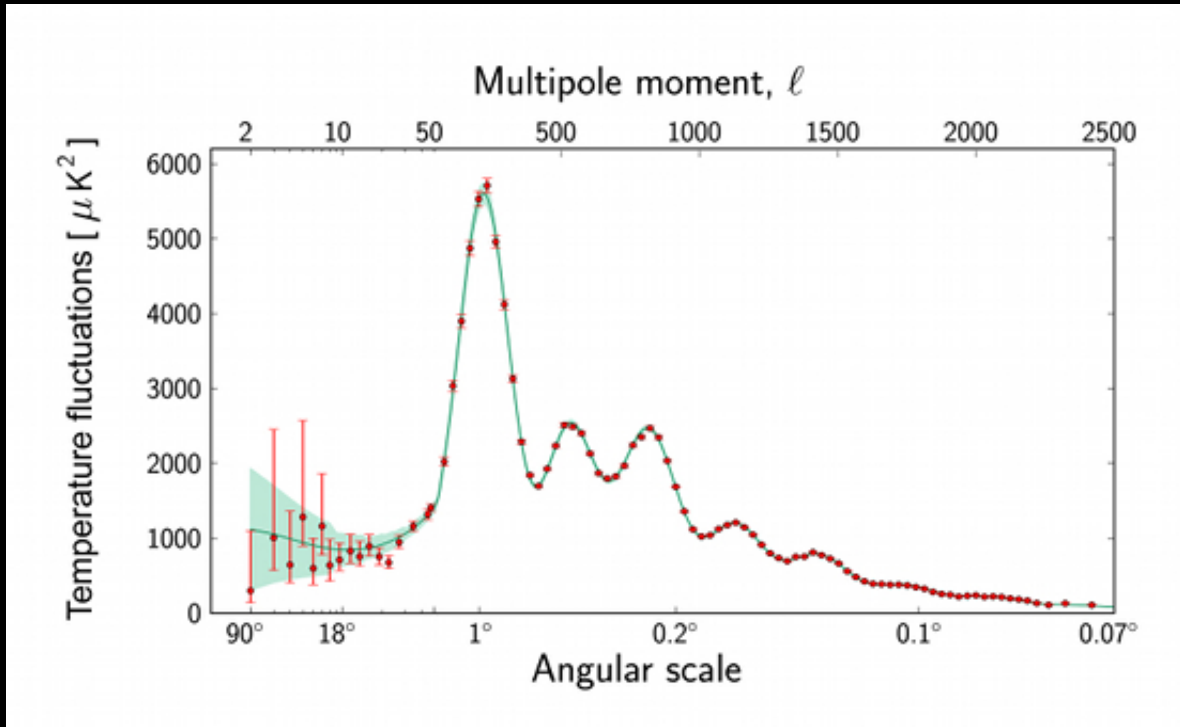
COSMOLOGY: 6 PARAMETERS



The Cosmic Microwave Background
A Gaussian Process in the Sky



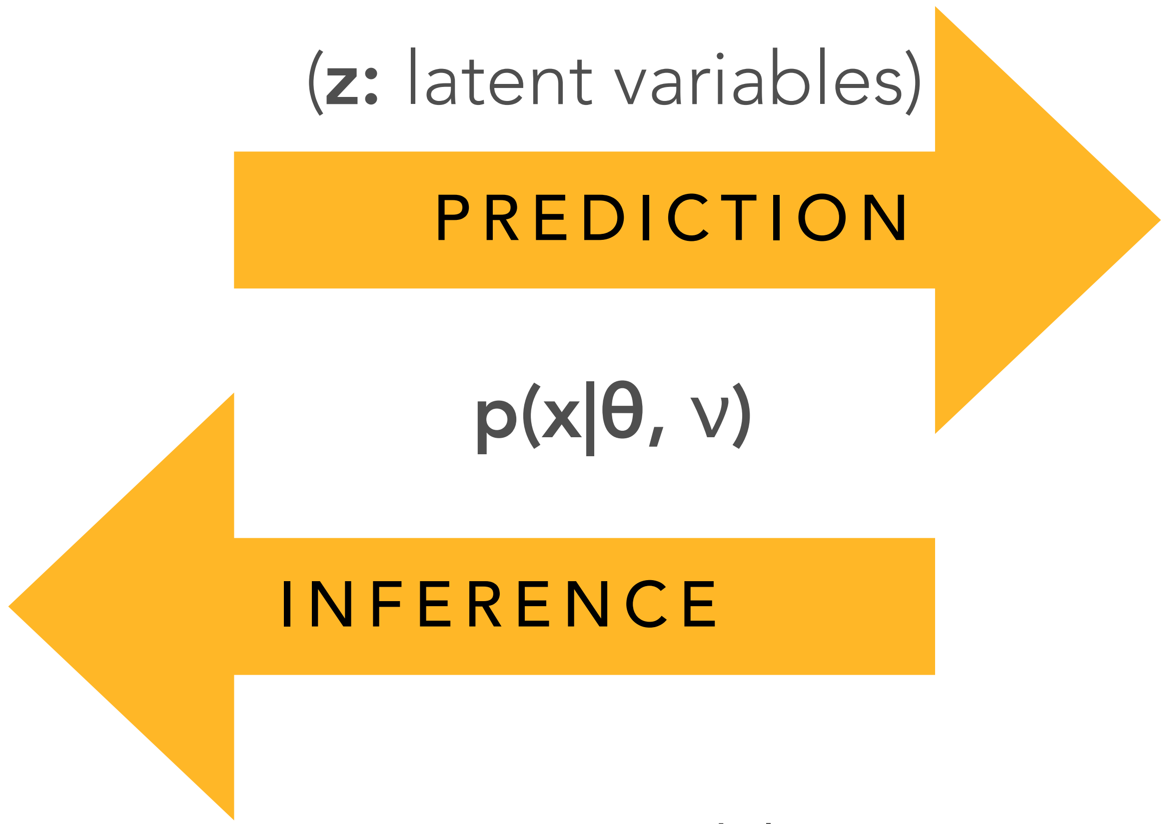
Symbol	Description	Value
$\Omega_B H^2$	Physical Baryon Density Parameter	0.02230 ± 0.00014
$\Omega_C H^2$	Physical Dark Matter Density Parameter	0.1188 ± 0.0010
T_0	Age Of The Universe	13.799 ± 0.021 × 10^9 Years
N_S	Scalar Spectral Index	0.9667 ± 0.0040
Δ_2	Curvature Fluctuation Amplitude	2.441 ± 0.09 × 10^{-9}
T	Reionization Optical Depth	0.066 ± 0.012



THE PLAYERS

forward modeling
generation
simulation

(**z:** latent variables)



θ
parameters of interest

$p(x|\theta, \nu)$

x
observed data
covariates
simulated data

ν
nuisance parameters

inverse problem
measurement
parameter estimation

The Forward Model

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

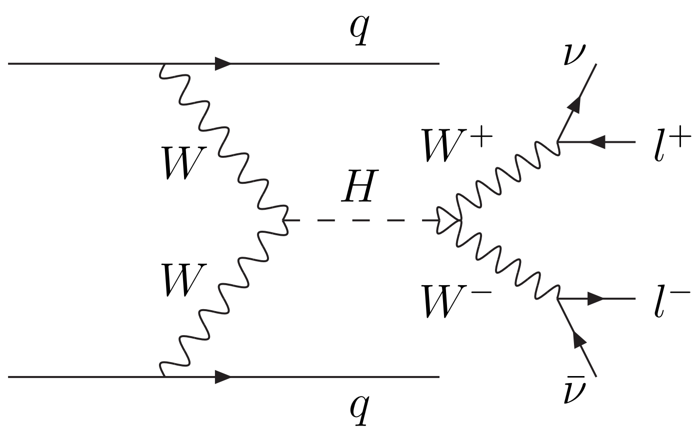
1) We begin with Quantum Field Theory

The Forward Model

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

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions



hierarchical: 2 → O(10) → O(100) particles

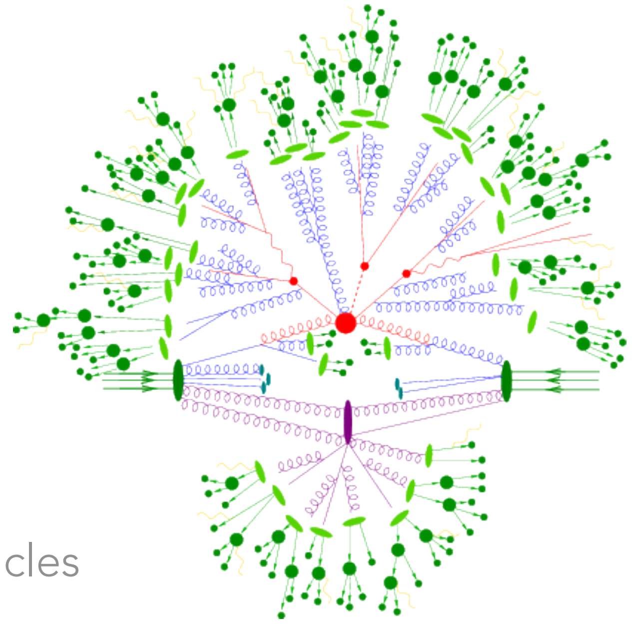
The Forward Model

$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^a_{\mu\nu}G^{\mu\nu}_a}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$
$$+ \underbrace{\bar{L}\gamma^\mu(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)L + \bar{R}\gamma^\mu(i\partial_\mu - \frac{1}{2}g'YB_\mu)R}_{\text{kinetic energies and electroweak interactions of fermions}}$$
$$+ \underbrace{\frac{1}{2}\left|(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)\,\phi\right|^2 - V(\phi)}_{W^\pm,Z,\gamma,\text{and Higgs masses and couplings}}$$
$$+ \underbrace{g''(\bar{q}\gamma^\mu T_a q)\,G^a_\mu}_{\text{interactions between quarks and gluons}} + \underbrace{(G_1\bar{L}\phi R + G_2\bar{L}\phi_c R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions

hierarchical: 2 → O(10) → O(100) particles



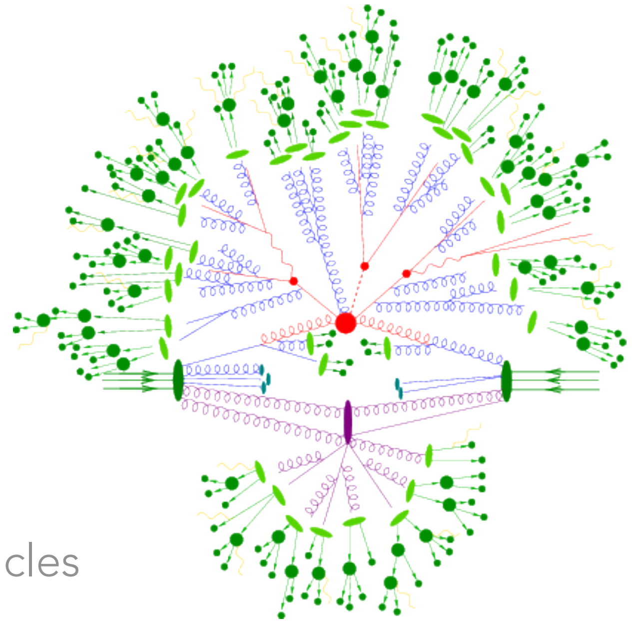
The Forward Model

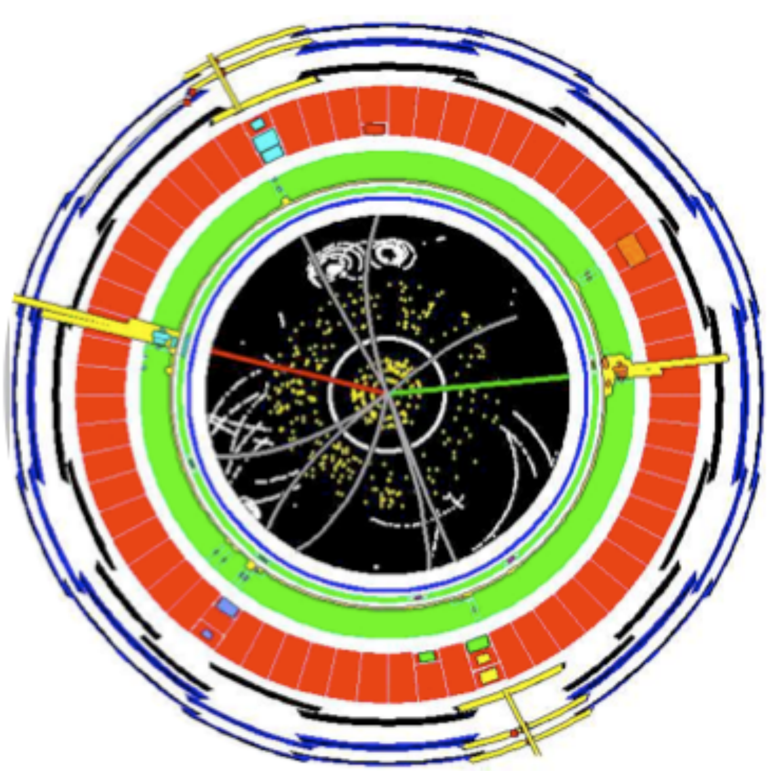
$$
\begin{aligned}
\mathcal{L}_{SM} = \; & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
+ \; & \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu}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

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions

hierarchical: 2 → O(10) → O(100) particles





3) The interaction of outgoing particles with the detector is simulated.

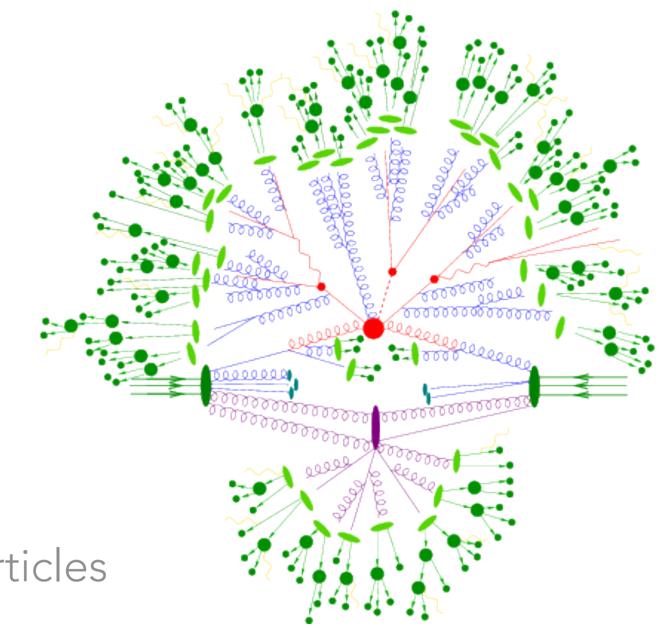
>100 million sensors

The Forward Model

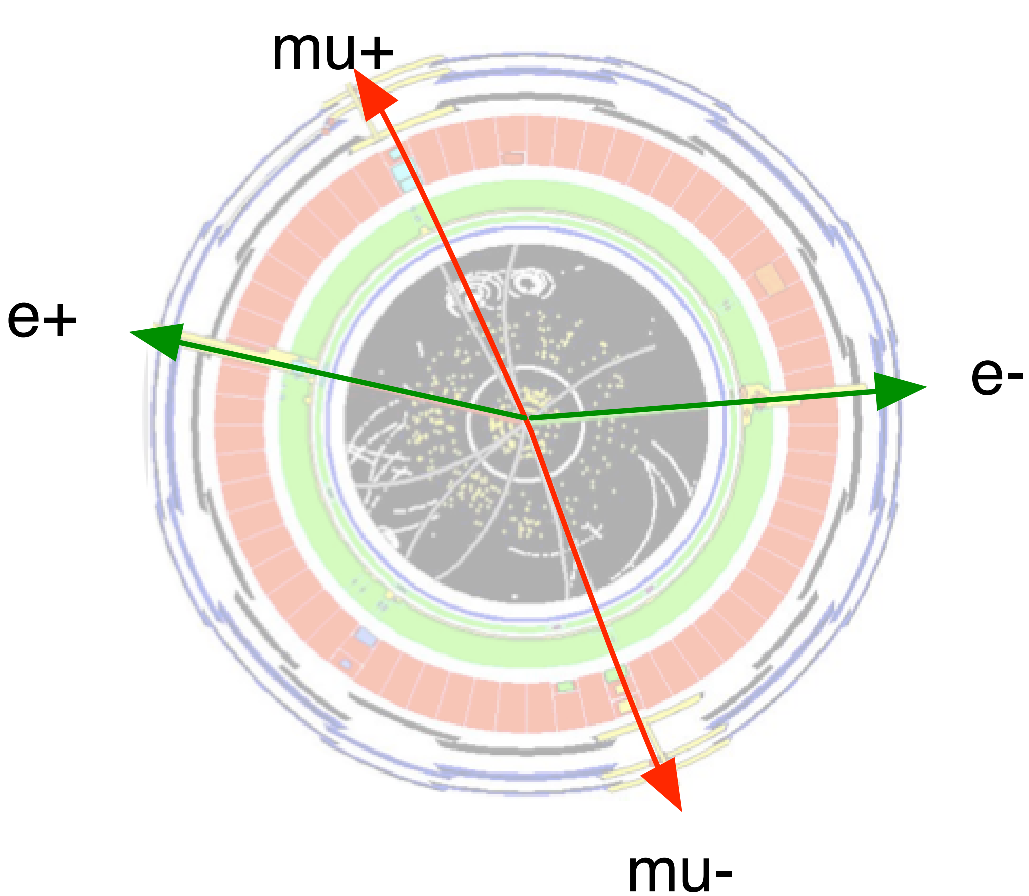
$$
\begin{aligned}
\mathcal{L}_{SM} = \; & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
& + \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu}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

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions



hierarchical: 2 → O(10) → O(100) particles



3) The interaction of outgoing particles with the detector is simulated.

>100 million sensors

4) Finally, we run particle identification and feature extraction algorithms on the simulated data as if they were from real collisions.

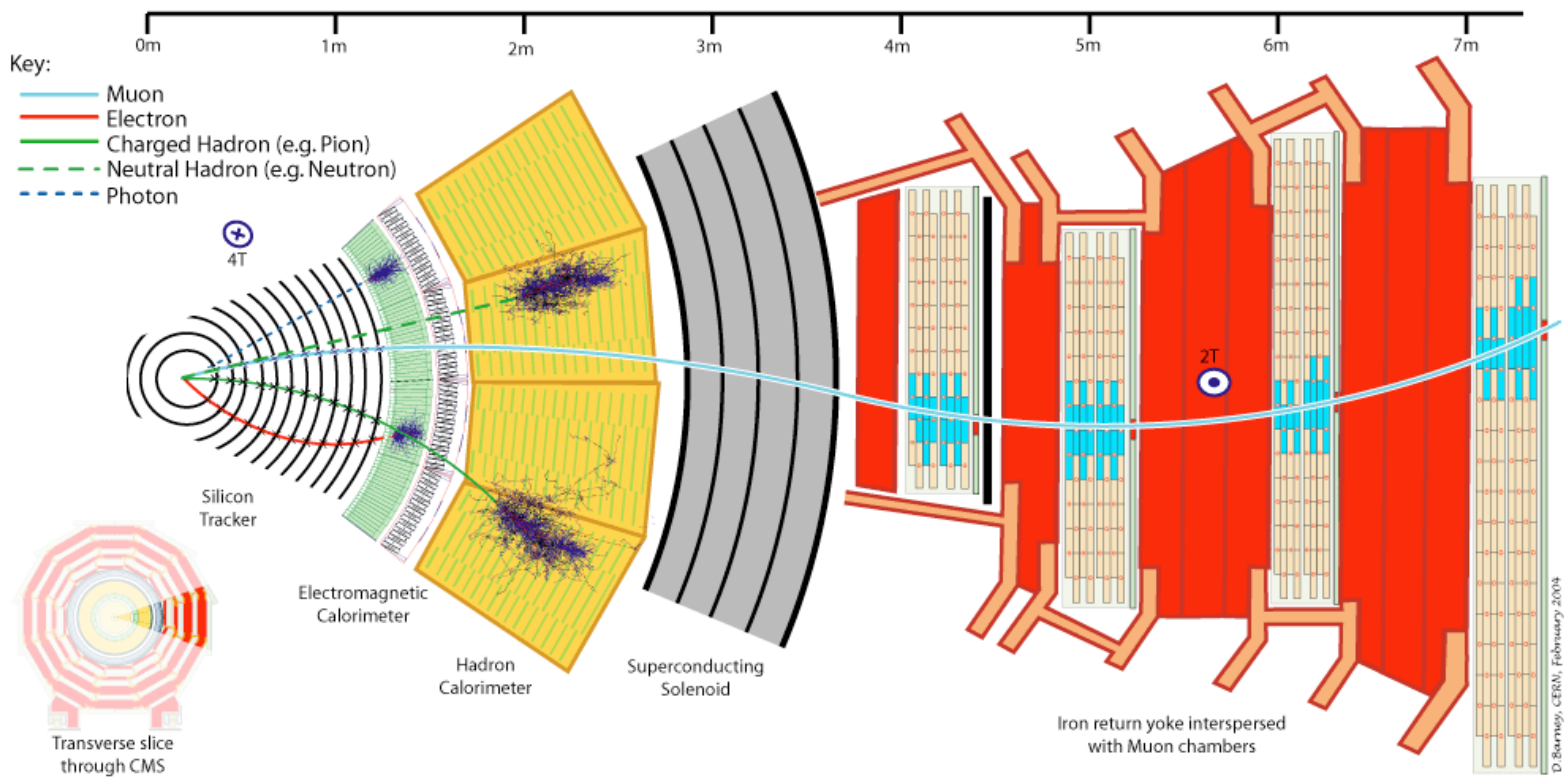
~10-30 features describe interesting part

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

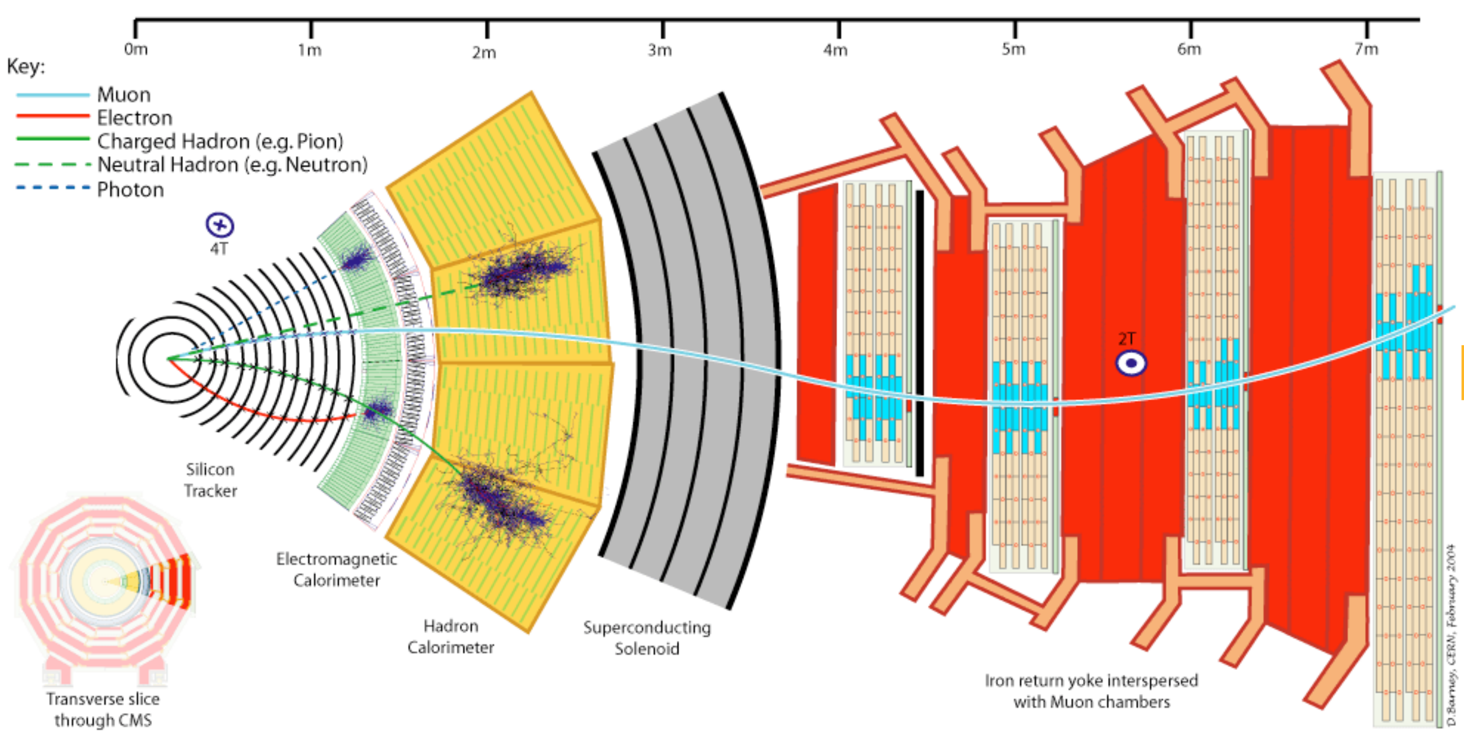
Consequence: evaluation of the likelihood is intractable

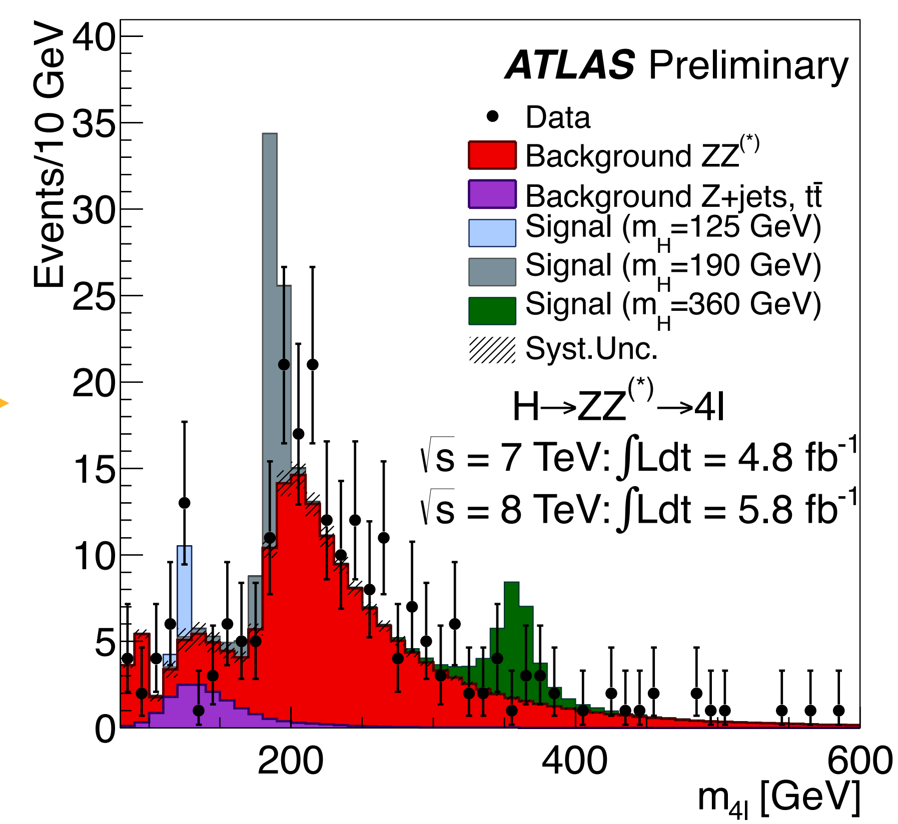
This motivates a new class of algorithms for what is called **likelihood-free inference**, which only require ability to generate samples from the simulation in the “forward mode”

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 variable or feature

- choosing a good variable (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)

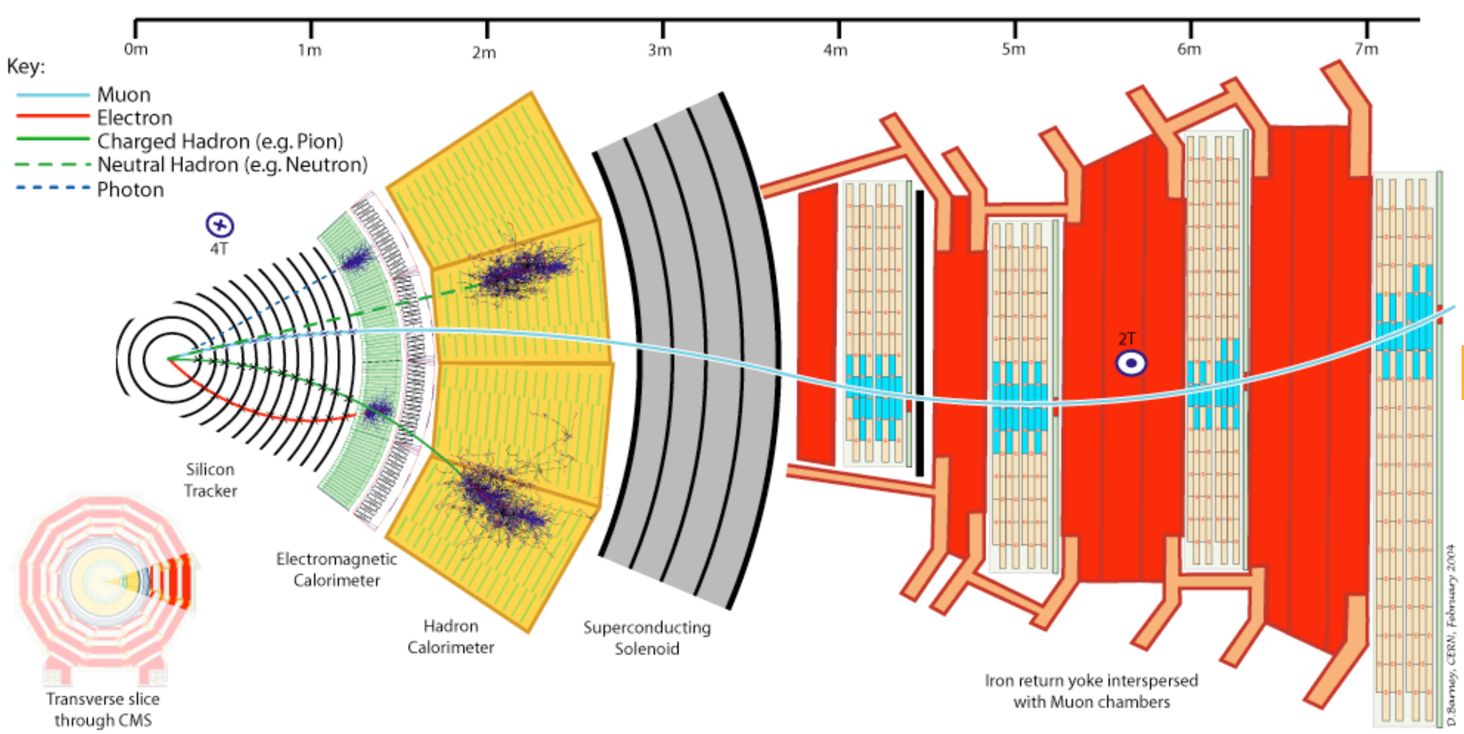


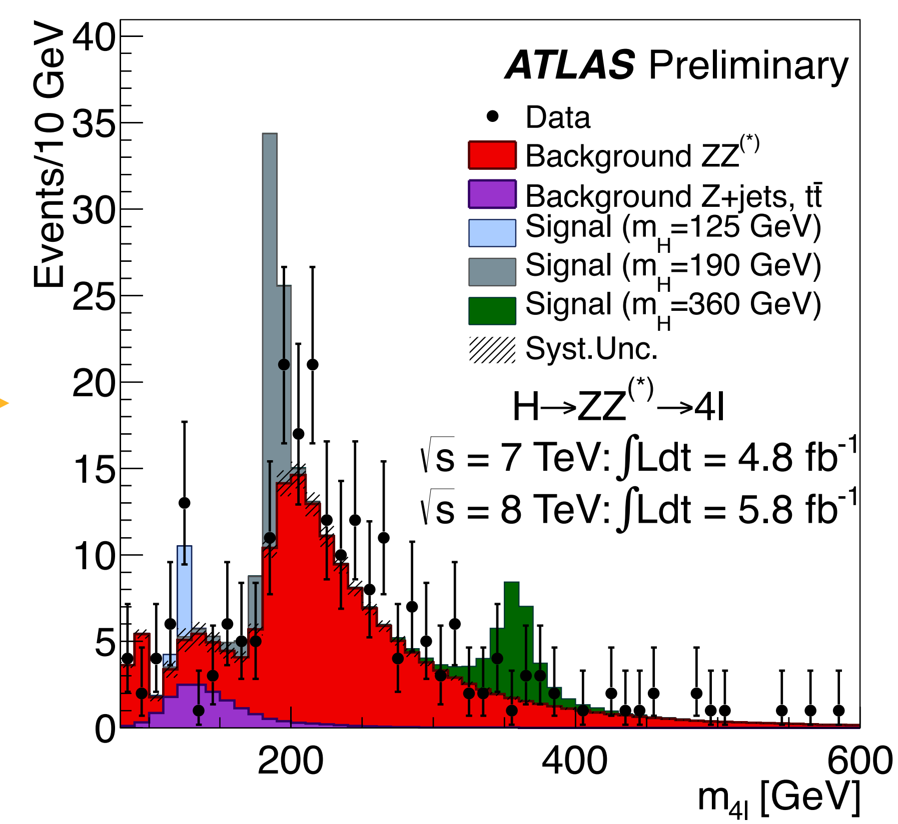


10^8 SENSORS → 1 REAL-VALUED QUANTITY

Most measurements and searches for new particles at the LHC are based on the distribution of a single variable or feature

- choosing a good variable (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!

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.

‘Likelihood-Free’ Inference

Rejection Algorithm

- Draw θ from prior $\pi(\cdot)$
- Accept θ with probability $\pi(D \mid \theta)$

Accepted θ are independent draws from the posterior distribution, $\pi(\theta \mid D)$.
If the likelihood, $\pi(D|\theta)$, is unknown:

‘Mechanical’ Rejection Algorithm

- Draw θ from $\pi(\cdot)$
- Simulate $X \sim f(\theta)$ from the computer model
- Accept θ if $D = X$, i.e., if computer output equals observation

The acceptance rate is $\int \mathbb{P}(D|\theta)\pi(\theta)\mathrm{d}\theta = \mathbb{P}(D)$.

*From Richard Wilkinson’s talk at Data science @LHC

Rejection ABC

If $\mathbb{P}(D)$ is small (or D continuous), we will rarely accept any θ. Instead, there is an approximate version:

Uniform Rejection Algorithm

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(D, X) \leq \epsilon$

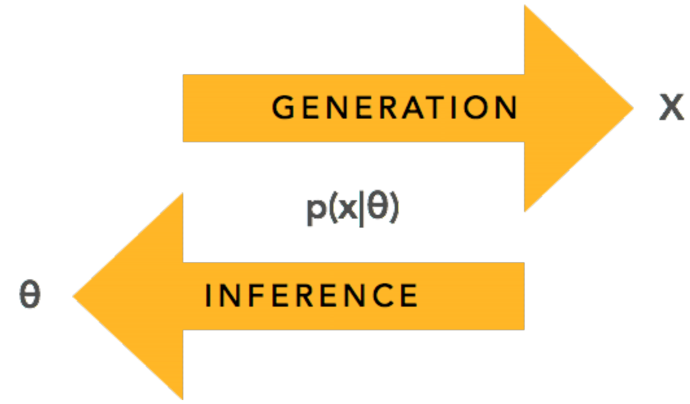
ϵ reflects the tension between computability and accuracy.

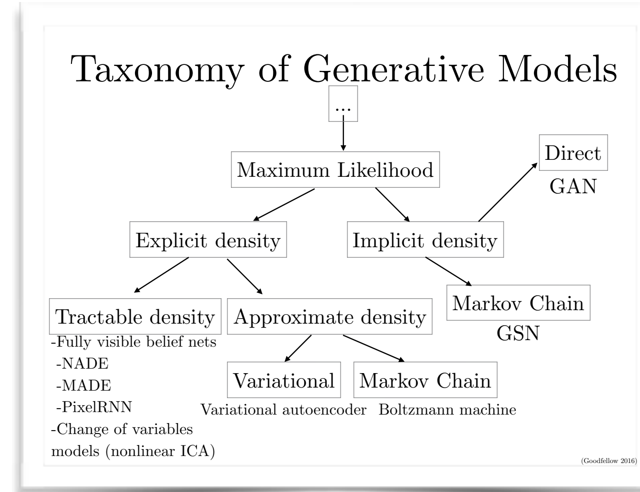
- As $\epsilon \to \infty$, we get observations from the prior, $\pi(\theta)$.
- If $\epsilon = 0$, we generate observations from $\pi(\theta \mid D)$.

For reasons that will become clear later, we call this *uniform-ABC*.

*From Richard Wilkinson's talk at Data science @LHC

COMPARISON



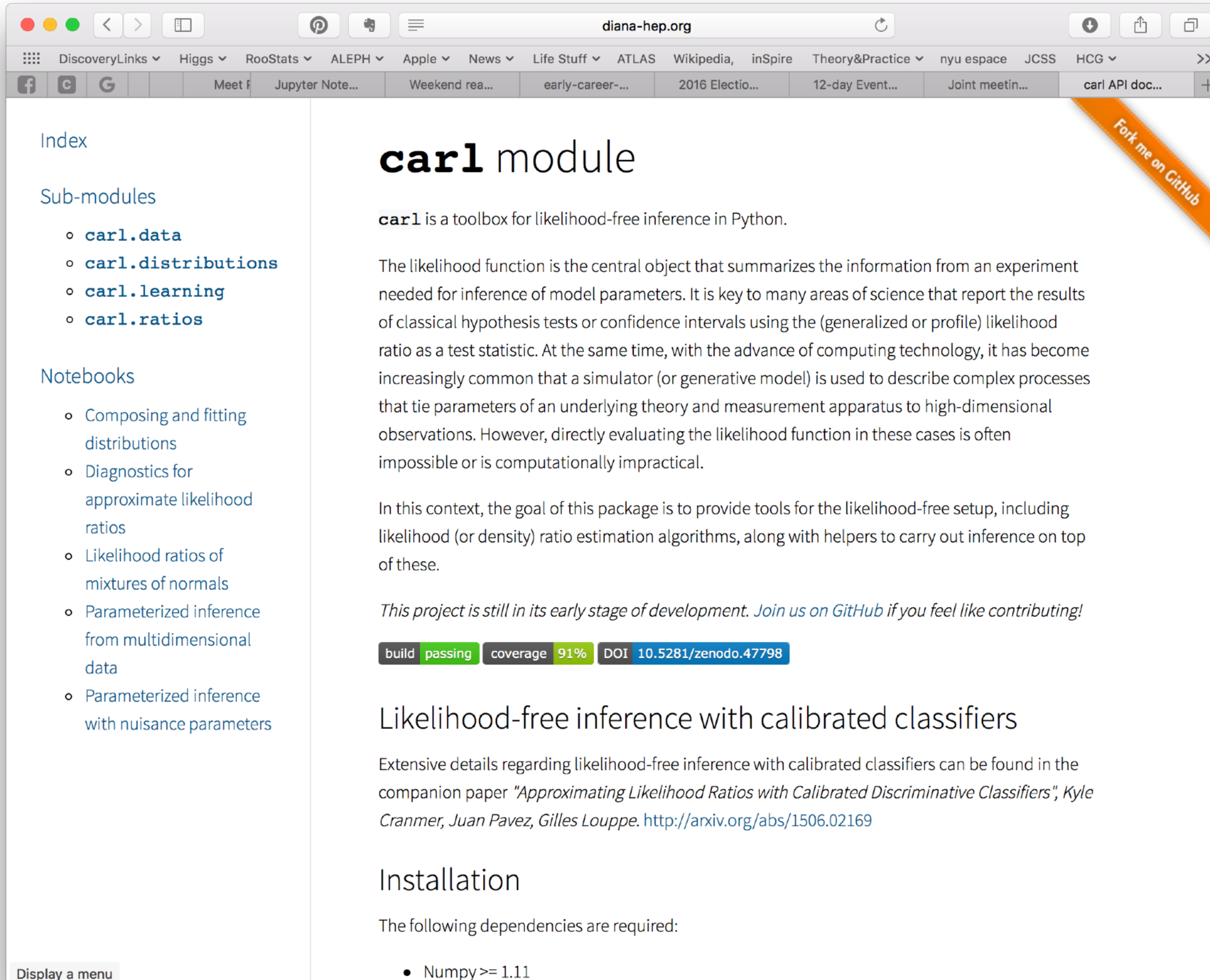


	Goal is to estimate	likelihood-free	θ inference	Generator $p(x\|\theta)$
ABC	$p(\theta \mid x_0)$	yes	approximate	—
BBVI	$p(\theta, z \mid x)$	no	—	—
AEVB	$p(\varphi, z \mid x)$	yes	approximate on φ not θ	surrogate
c-GAN	$p(x\|\theta)$	yes	—	surrogate
NVP/IAF	$p(x)$	yes	—	surrogate
CARL	$p(x \mid \theta)/p(x \mid \theta_1)$	yes	exact	simulation@ θ_1 x importance sampling to θ
"c-NVP"	$p(x\|\theta)$ via bijections $x(z\|\theta)$	yes	exact	surrogate

exact = asymptotically consistent in infinite capacity limit

CARL SOFTWARE

http://diana-hep.org/carl/



Index

Sub-modules

- `carl.data`
- `carl.distributions`
- `carl.learning`
- `carl.ratios`

Notebooks

- Composing and fitting distributions
- Diagnostics for approximate likelihood ratios
- Likelihood ratios of mixtures of normals
- Parameterized inference from multidimensional data
- Parameterized inference with nuisance parameters

`carl` module

`carl` is a toolbox for likelihood-free inference in Python.

The likelihood function is the central object that summarizes the information from an experiment needed for inference of model parameters. It is key to many areas of science that report the results of classical hypothesis tests or confidence intervals using the (generalized or profile) likelihood ratio as a test statistic. At the same time, with the advance of computing technology, it has become increasingly common that a simulator (or generative model) is used to describe complex processes that tie parameters of an underlying theory and measurement apparatus to high-dimensional observations. However, directly evaluating the likelihood function in these cases is often impossible or is computationally impractical.

In this context, the goal of this package is to provide tools for the likelihood-free setup, including likelihood (or density) ratio estimation algorithms, along with helpers to carry out inference on top of these.

This project is still in its early stage of development. Join us on GitHub if you feel like contributing!

build passing | coverage 91% | DOI 10.5281/zenodo.47798

Likelihood-free inference with calibrated classifiers

Extensive details regarding likelihood-free inference with calibrated classifiers can be found in the companion paper *"Approximating Likelihood Ratios with Calibrated Discriminative Classifiers", Kyle Cranmer, Juan Pavez, Gilles Louppe.* http://arxiv.org/abs/1506.02169

Installation

The following dependencies are required:

- Numpy >= 1.11

Hierarchical Graphical Models

"LA MIA PARABOLA"

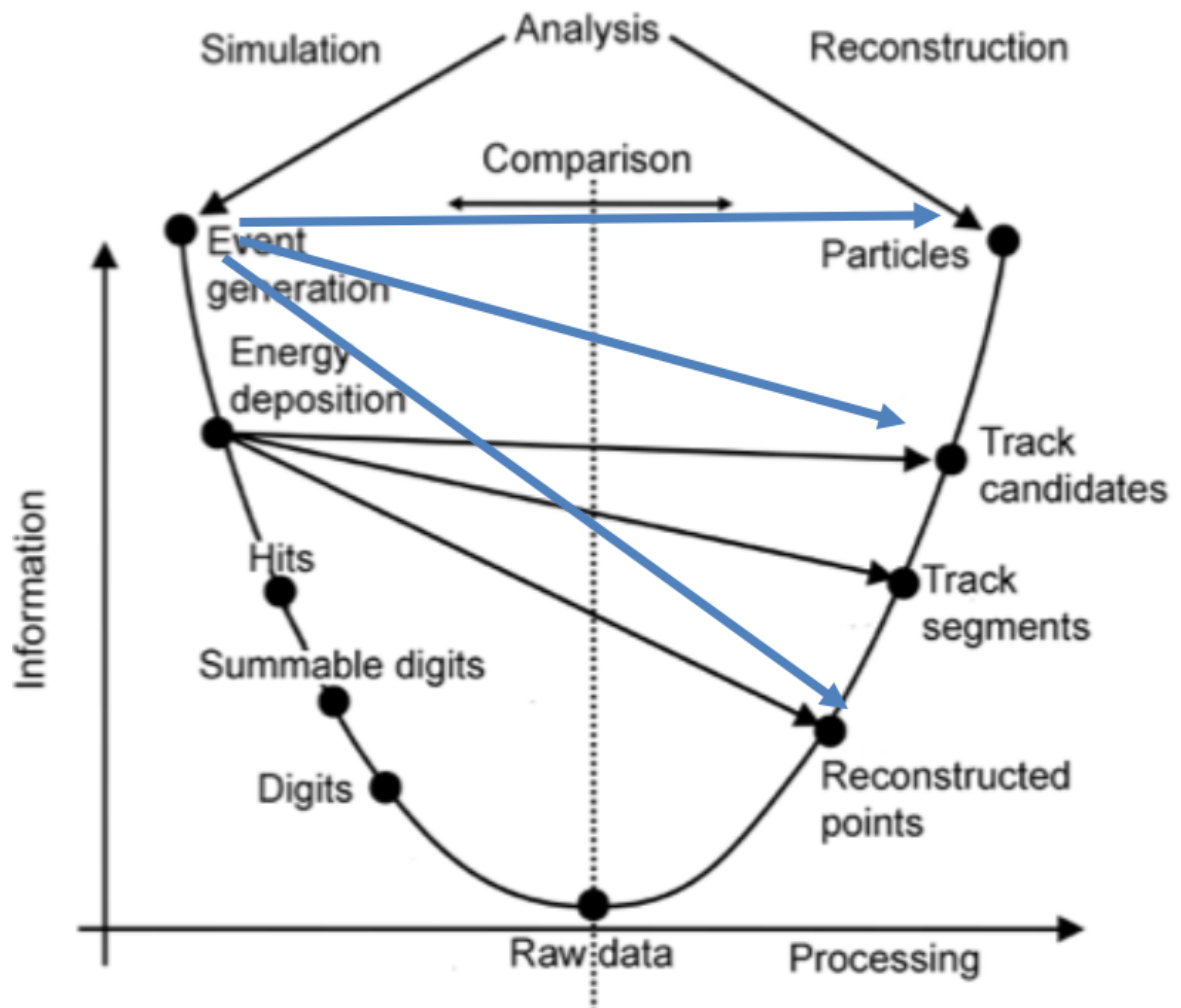
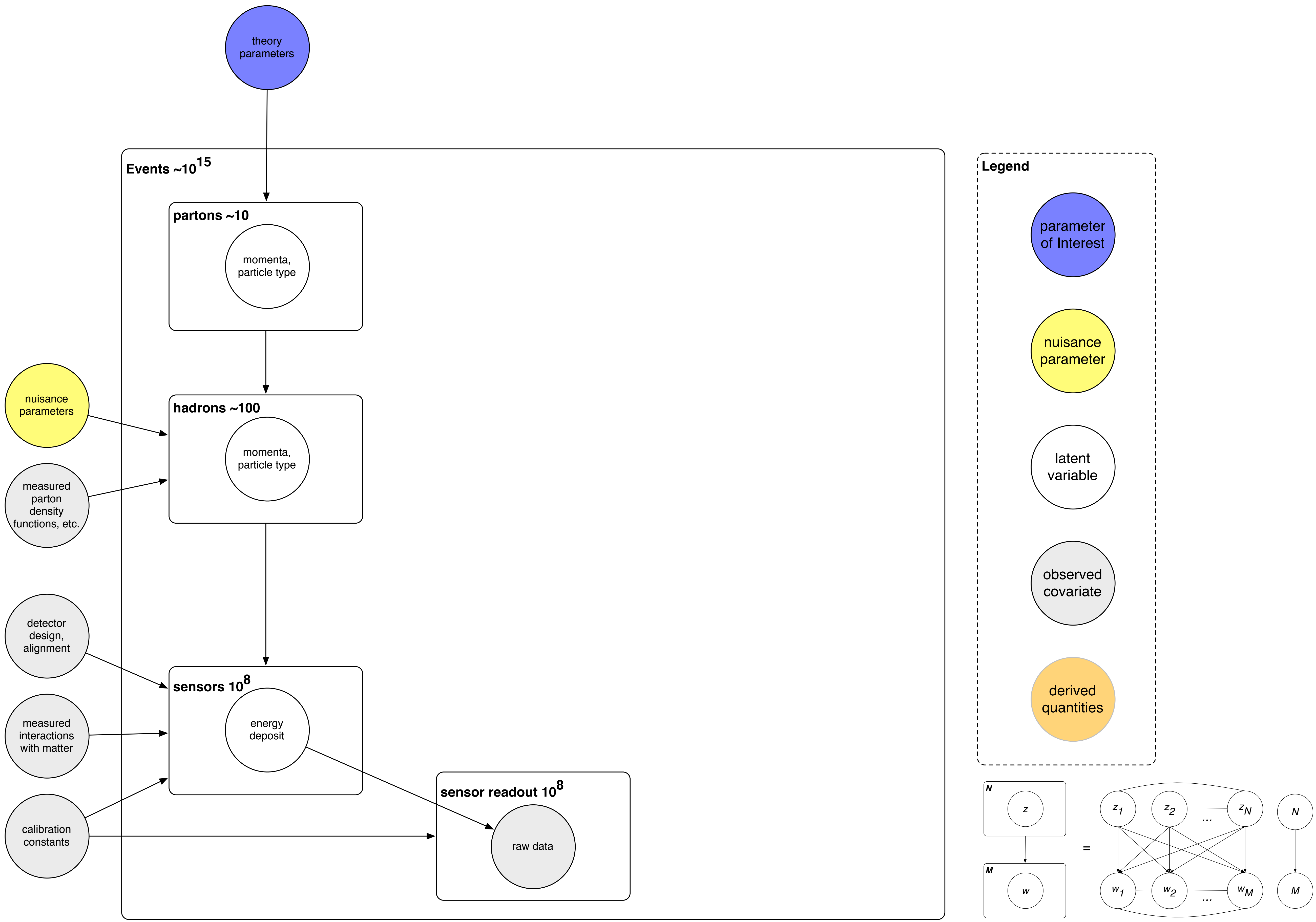
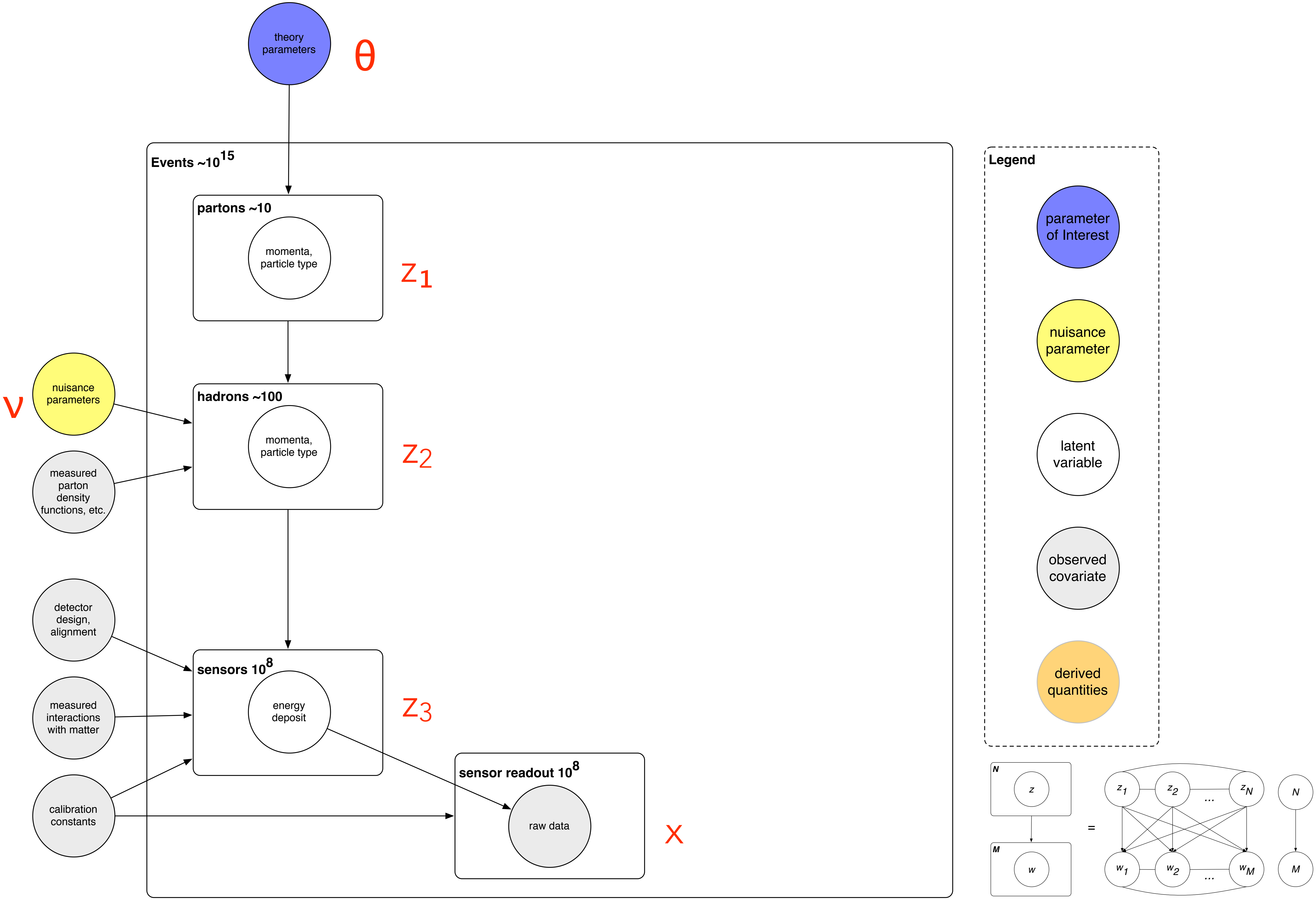


Figure by Federico Carminati, independent parallel inventions by Vincenzo Innocente & K.C.

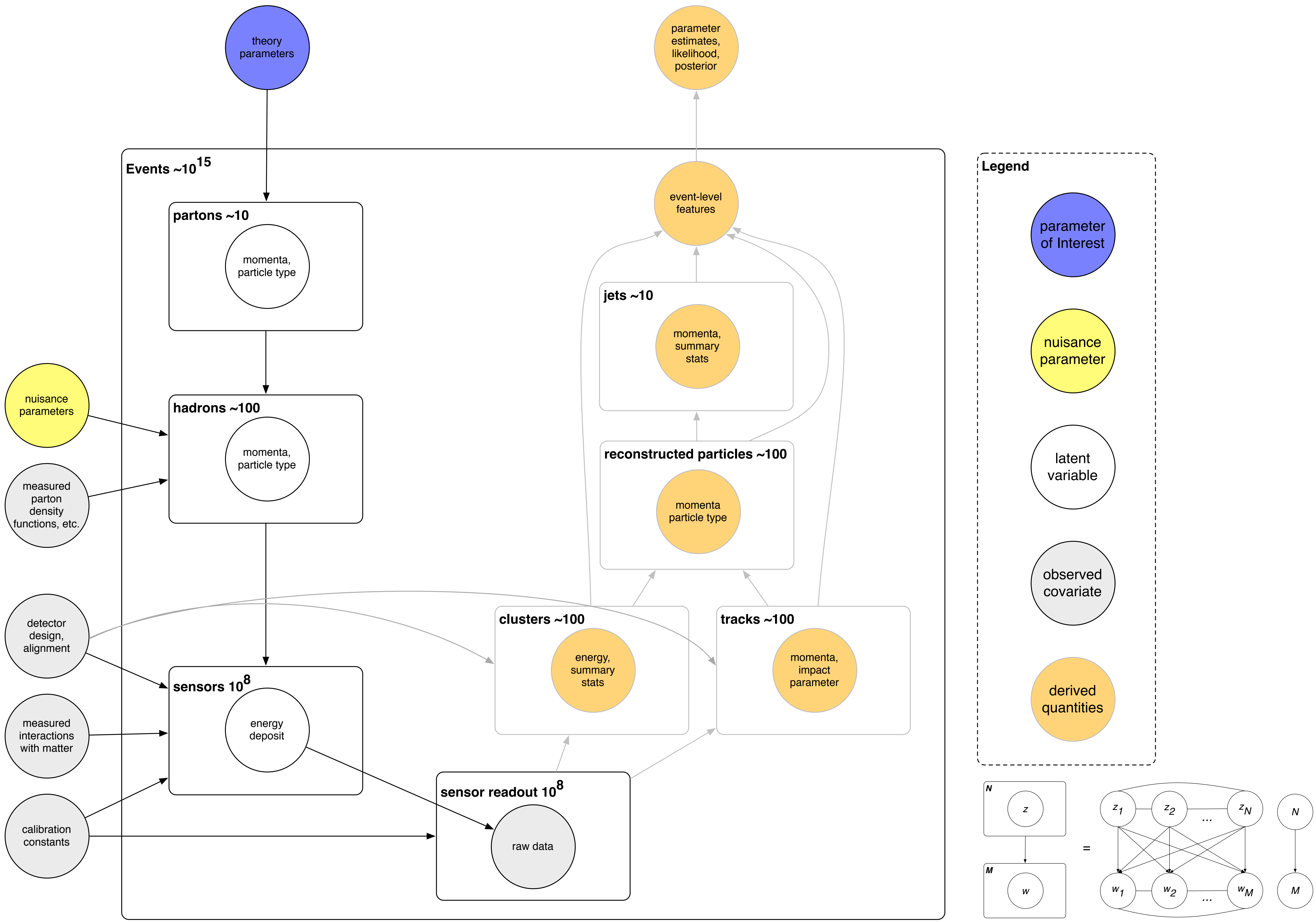
FULL SIMULATION

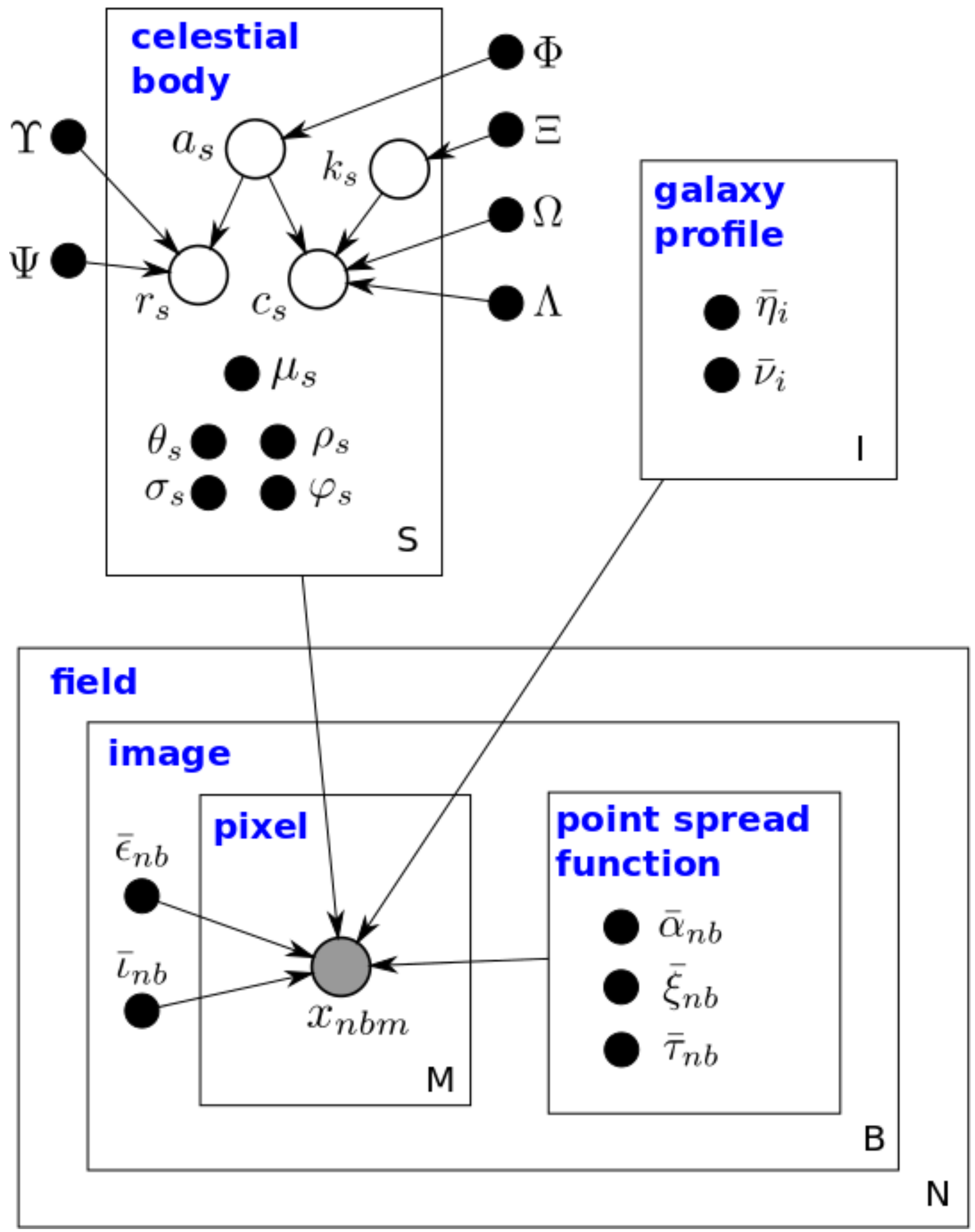


FULL SIMULATION



FULL SIMULATION + RECONSTRUCTION





Celeste: Variational inference for a generative model of astronomical images

Regler, et al 33rd ICMLR, 2016

Learning

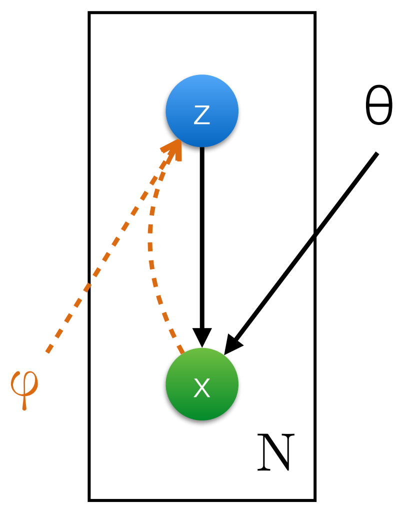
Generative Models / Implicit Models

VARIATIONAL AUTO-ENCODER

[Slides from D. Kingma NIPS 2015]

Auto-Encoding Variational Bayes

[Kingma and Welling, 2013/2014]
[Rezende et al, 2014]



- $q_\varphi(z|x) = N(\mu,\sigma^2)$
  $[\mu, \sigma^2] = f^{(z|x)}(x,\varphi) = \text{multilayer neural net}$
- Objective: lower bound of log p(x).
  - Jointly optimized w.r.t. φ and θ
  - This is approx. maximum likelihood
  - Simple SGD:
    - Sampling small minibatches of data
    - Sampling from approx. posterior
- This also minimizes an expected KL divergence
  $D_{KL}(q_\varphi(z|x)||p(z|x))$
  -> gives us cheap approx. inference for new datapoints

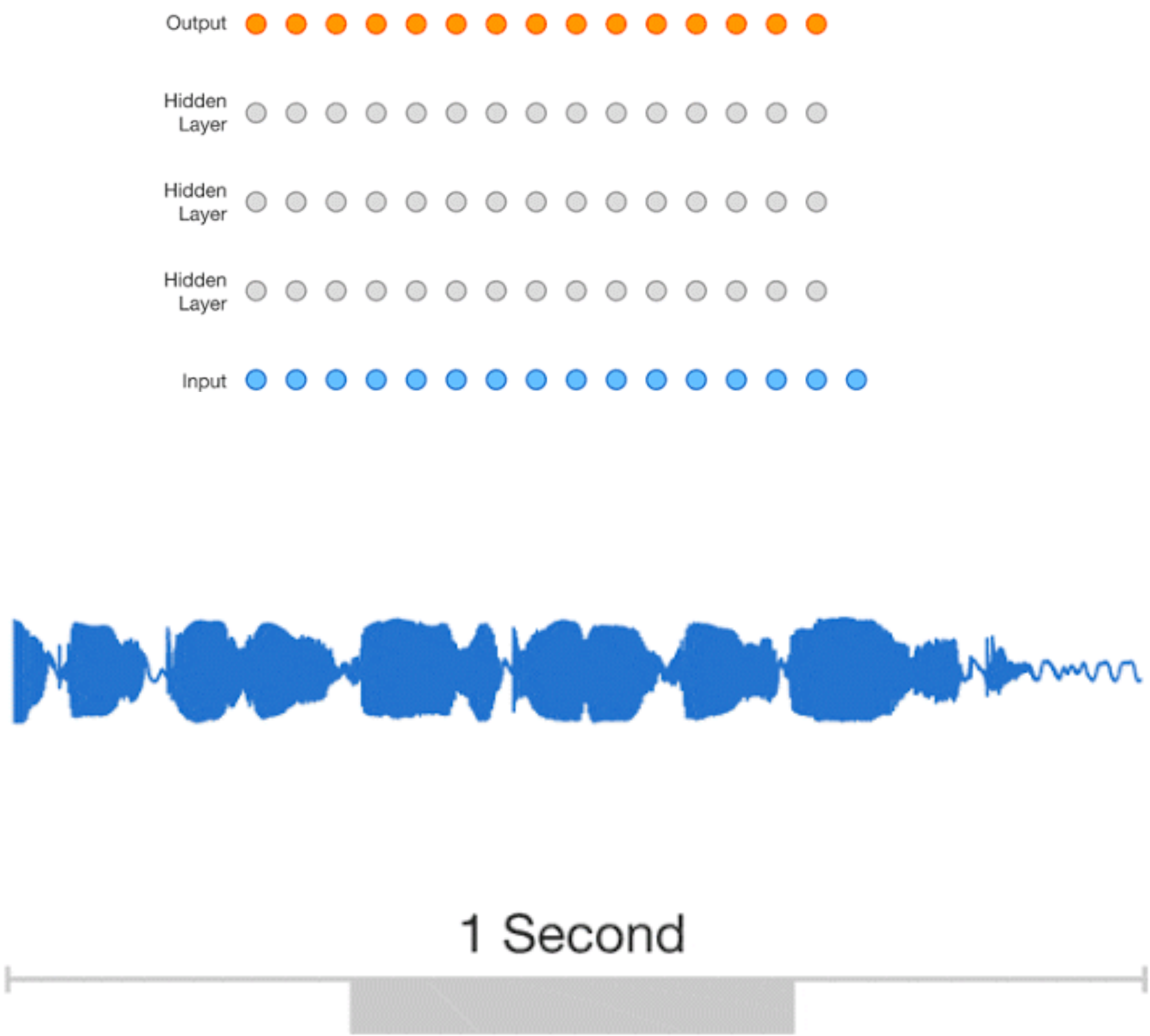
Kingma and Welling, Auto-encoding Variational Bayes, ICLR 2014
Rezende, Mohamed and Wierstra, Stochastic back-propagation and variational inference in deep latent Gaussian models, ICML 2014

Diederik (Durk) Kingma

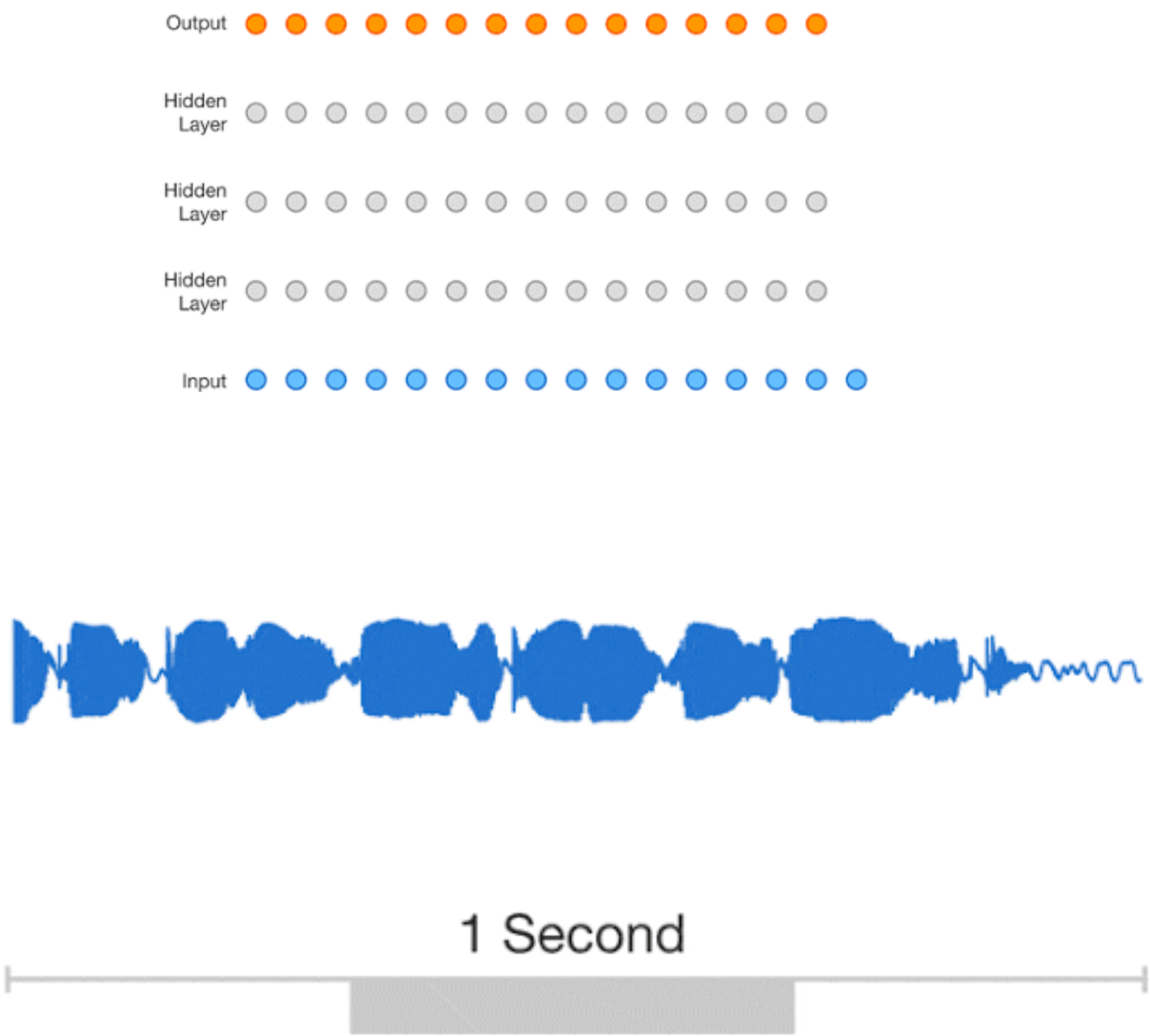
Max Welling

Conv. net as encoder/decoder, trained on faces

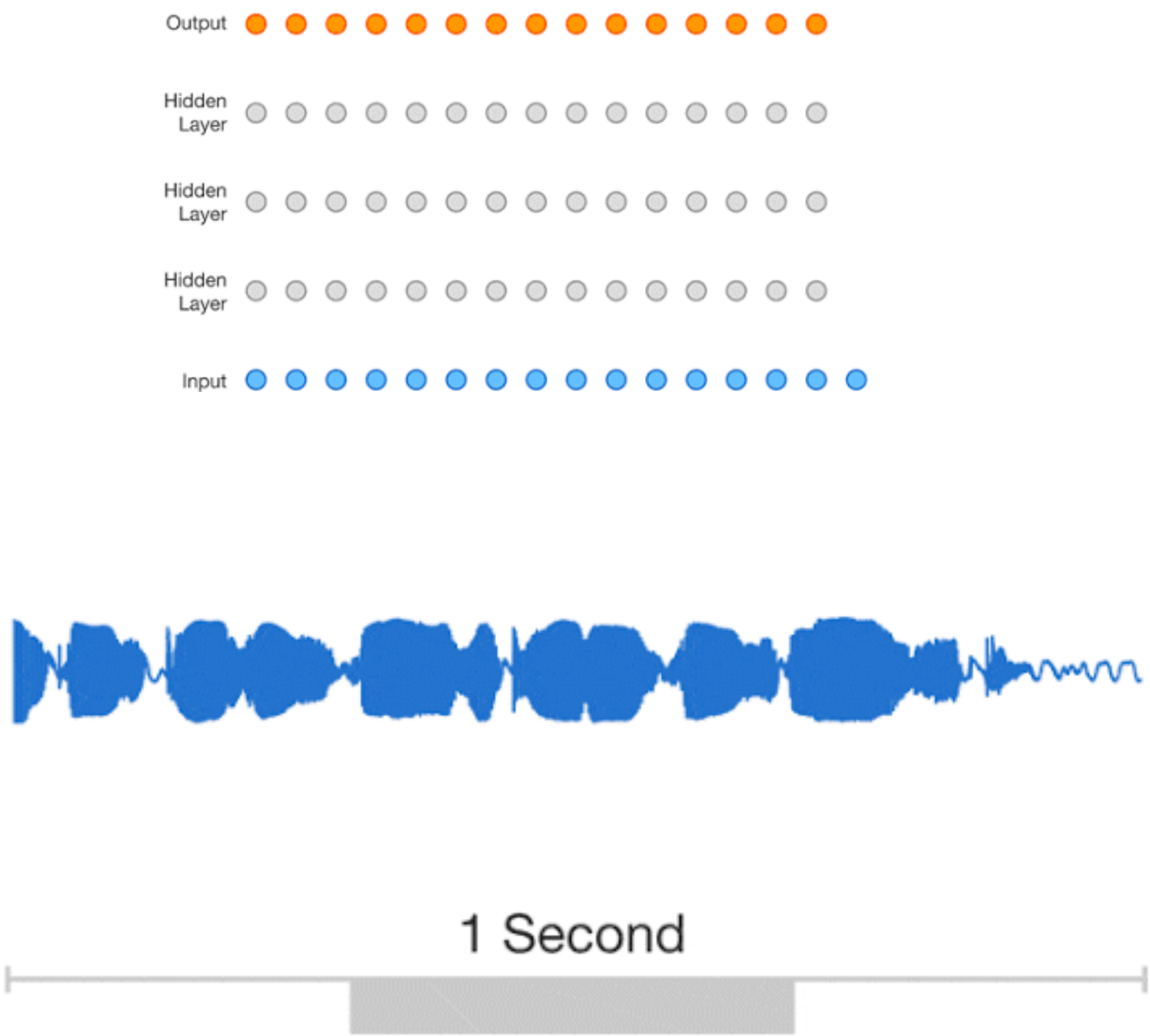
WAVENET: A GENERATIVE MODEL FOR RAW AUDIO



WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

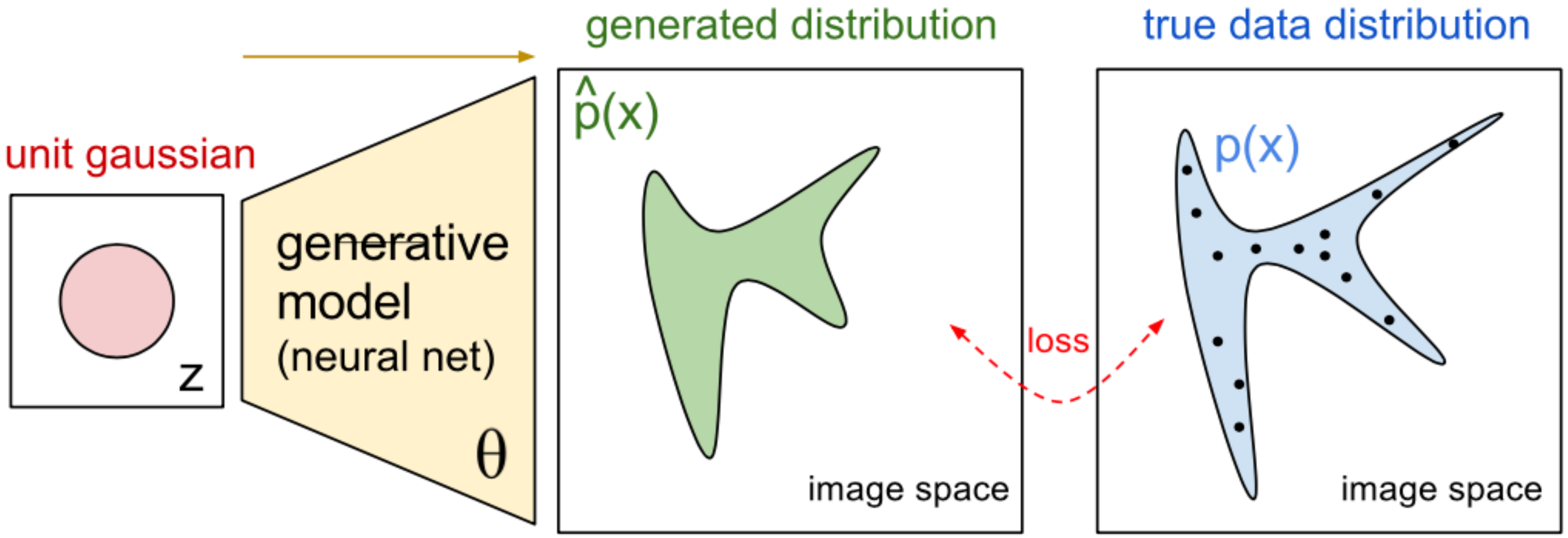


WAVENET: A GENERATIVE MODEL FOR RAW AUDIO



Goodfellow, et al arXiv:1406.2661
slide Gilles Louppe

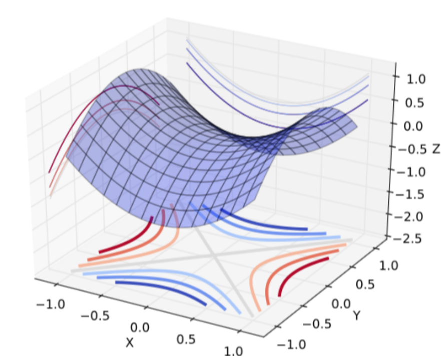
GENERATIVE ADVERSARIAL NETWORKS



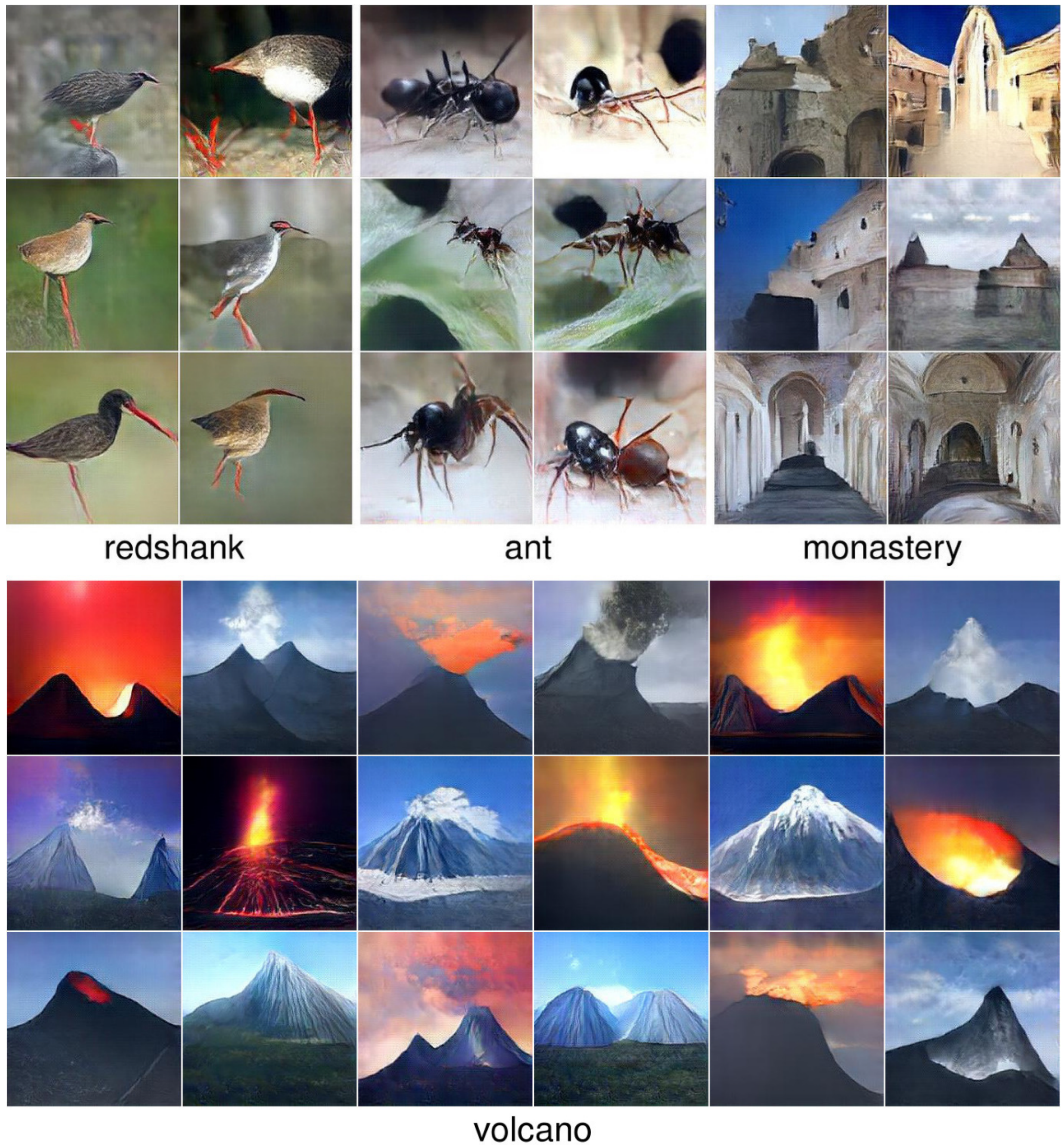
- Two-player game:
  - a discriminator D,
  - a generator G;
- D is a classifier $\mathcal{X} \mapsto \{0,1\}$ that tries to distinguish between
  - a sample from the data distribution ($D(\mathbf{x}) = 1$, for $\mathbf{x} \sim p_{\text{data}}$),
  - and a sample from the model distribution ($D(G(\mathbf{z})) = 0$, for $\mathbf{z} \sim p_{\text{noise}}$);
- G is a generator $\mathcal{Z} \mapsto \mathcal{X}$ trained to produce samples $G(\mathbf{z})$ (for $\mathbf{z} \sim p_{\text{noise}}$) that are difficult for D to distinguish from data.

$$(D^*, G^*) = \max_D \min_G V(D, G).$$

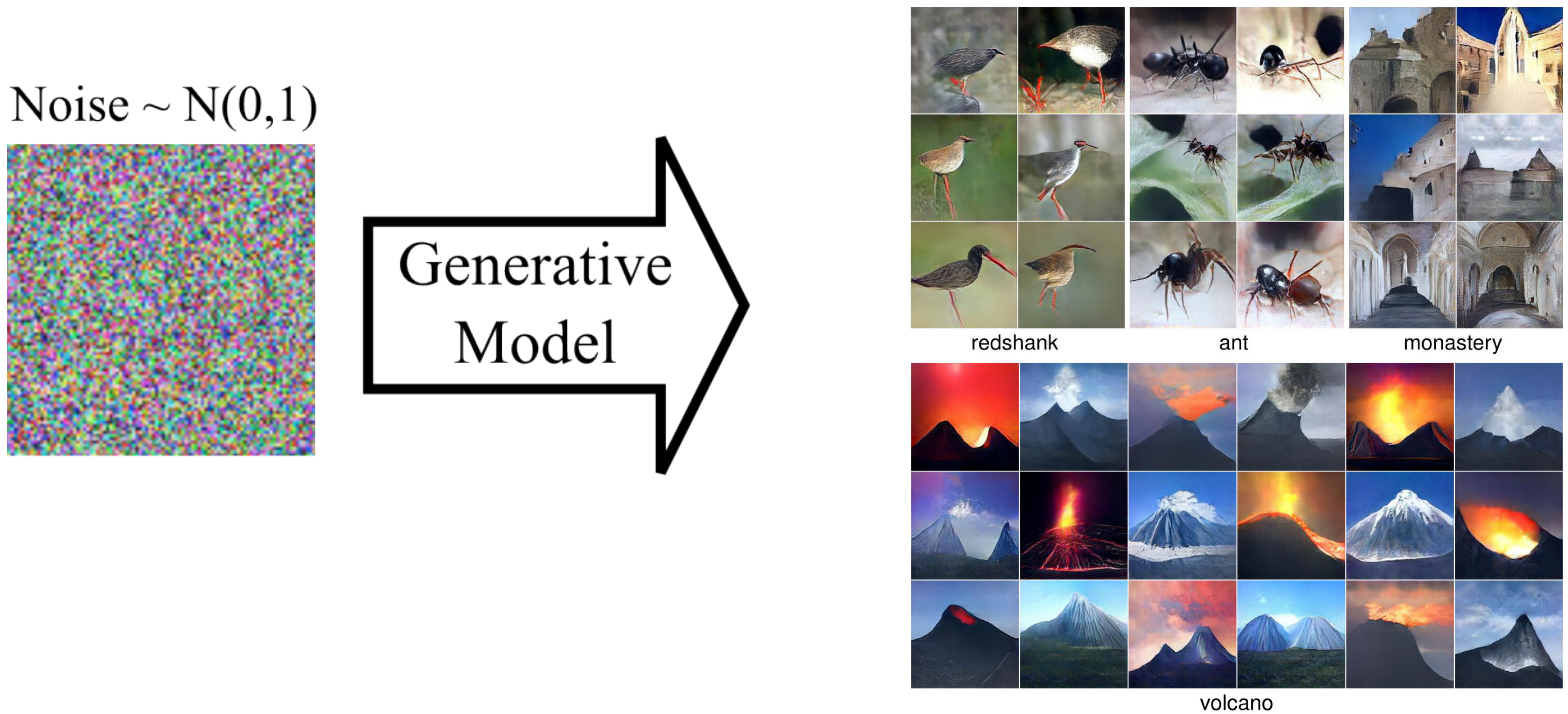
Leo is G Tom is D

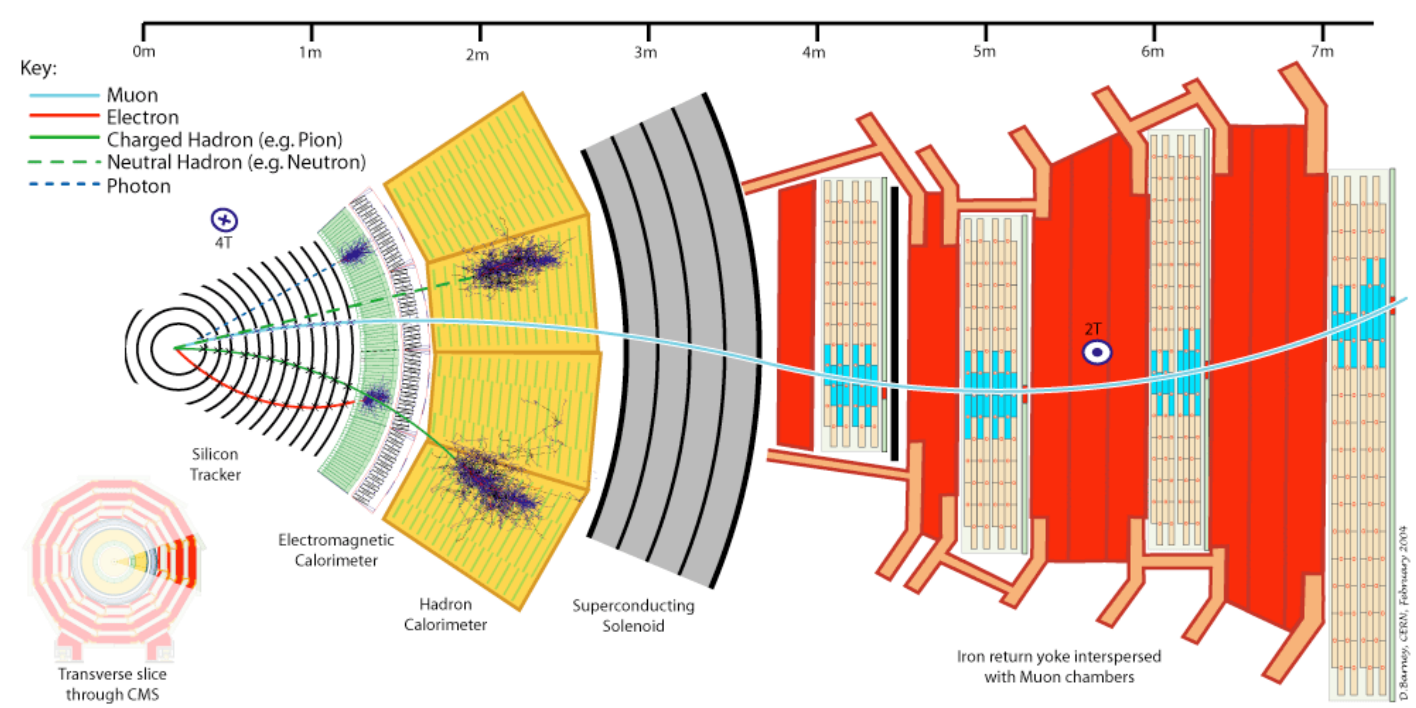


GENERATED IMAGES



LEARNING THE (SIMULATED) DATA DISTRIBUTION





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

GANs FOR PHYSICS

CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks

Michela Paganini[a,b], **Luke de Oliveira**[a], **and Benjamin Nachman**[a]

[a] *Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA, 94720, USA*
[b] *Department of Physics, Yale University, New Haven, CT 06520, USA*

E-mail: michela.paganini@yale.edu, lukedeoliveira@lbl.gov, bnachman@cern.ch

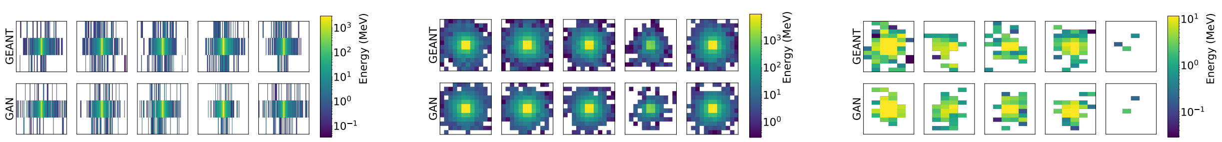


Figure 9: Five randomly selected e^+ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

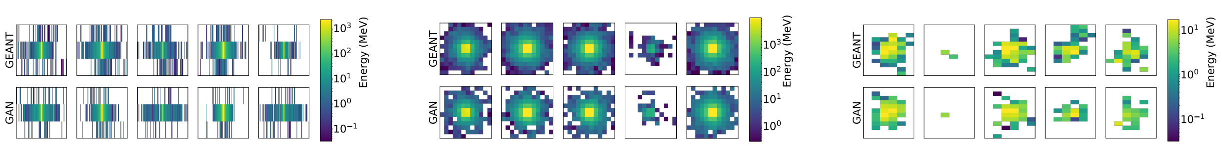


Figure 10: Five randomly selected γ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

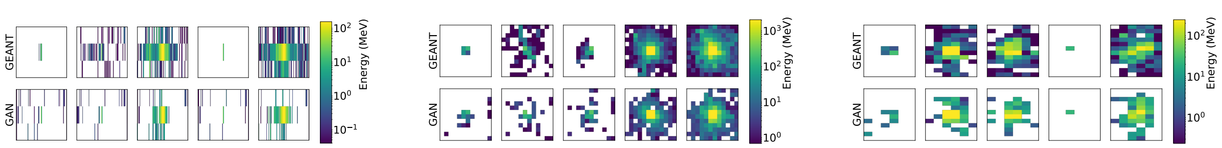
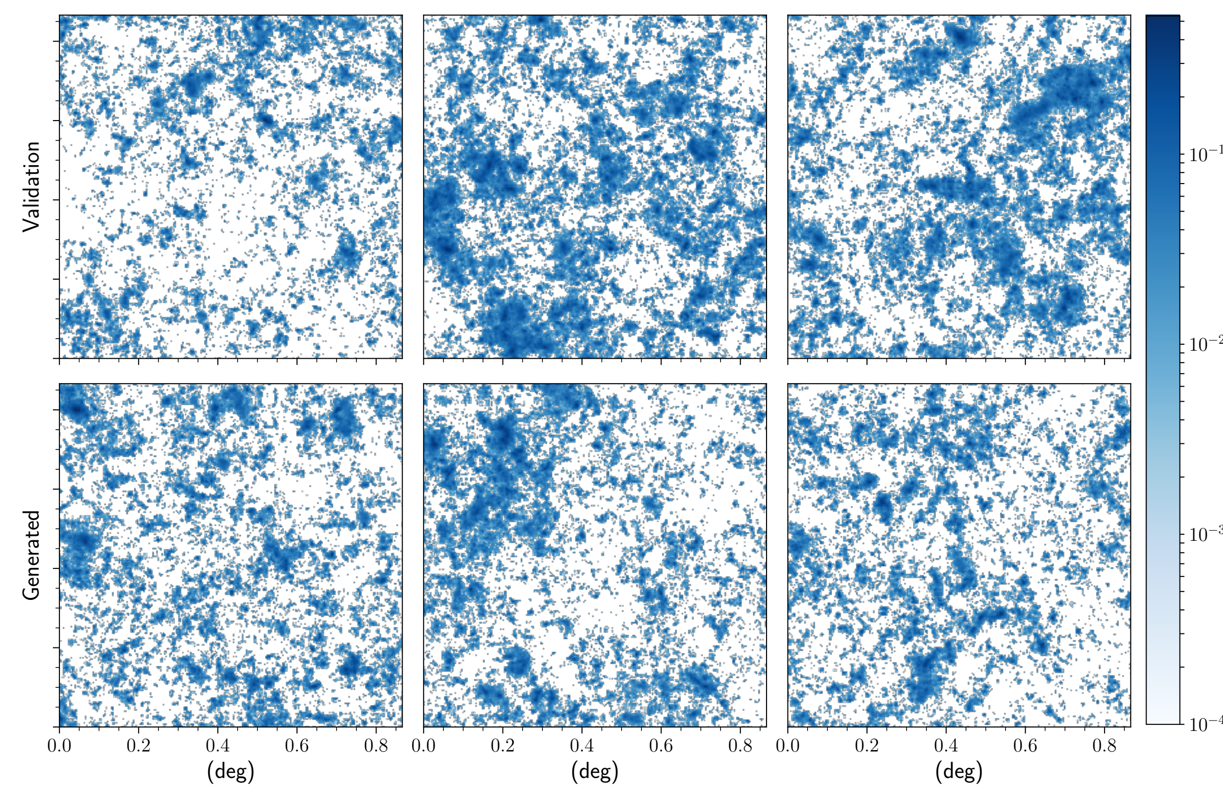


Figure 11: Five randomly selected π^+ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

Creating Virtual Universes Using Generative Adversarial Networks

Mustafa Mustafa*[1], Deborah Bard[1], Wahid Bhimji[1], Rami Al-Rfou[2], and Zarija Lukić[1]

[1]Lawrence Berkeley National Laboratory, Berkeley, CA 94720
[2]Google Research, Mountain View, CA 94043



GENERATIVE MODELS FOR CALIBRATION

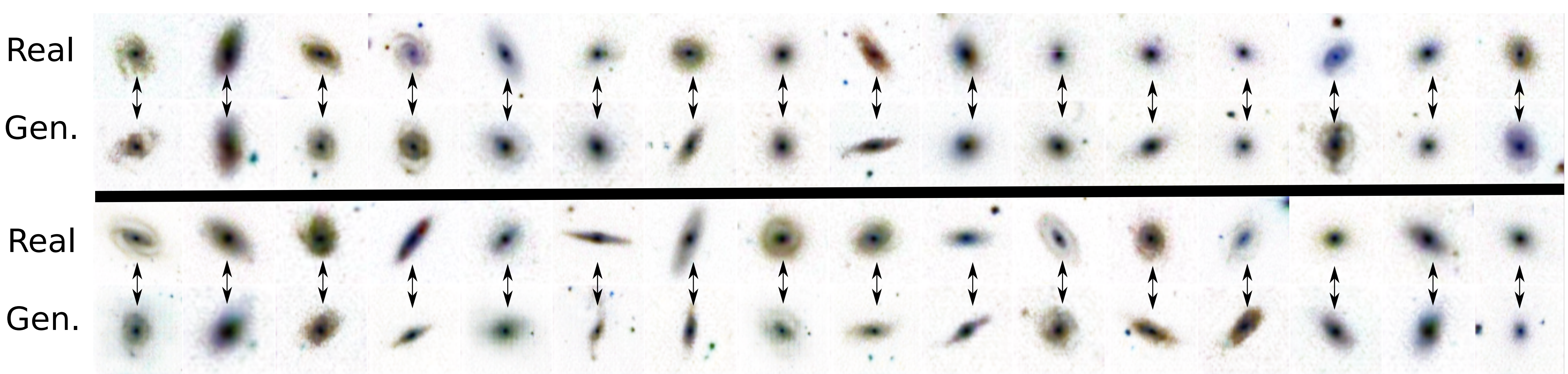
Use of generative models of galaxy images to help calibrate next-generation surveys.

Enabling Dark Energy Science with Deep Generative Models of Galaxy Images

Siamak Ravanbakhsh[1], François Lanusse[2], Rachel Mandelbaum[2], Jeff Schneider[1], and Barnabás Póczos[1]

[1]School of Computer Science, Carnegie Mellon University
[2]McWilliams Center for Cosmology, Carnegie Mellon University

***Abstract*—Understanding the nature of dark energy, the mysterious force driving the accelerated expansion of the Universe, is a major challenge of modern cosmology. The next generation of cosmological surveys, specifically designed to address this issue, rely on accurate measurements of the apparent shapes of distant galaxies. However, shape measurement methods suffer from various unavoidable biases and therefore will rely on a precise calibration to meet the accuracy requirements of the science analysis. This calibration process remains an open challenge as it requires large sets of high quality galaxy images. To this end, we study the application of deep conditional generative models in generating realistic galaxy images. In particular we consider variations on conditional variational autoencoder and introduce a new adversarial objective for training of conditional generative networks. Our results suggest a reliable alternative to the acquisition of expensive high quality observations for generating the calibration data needed by the next generation of cosmological surveys.**



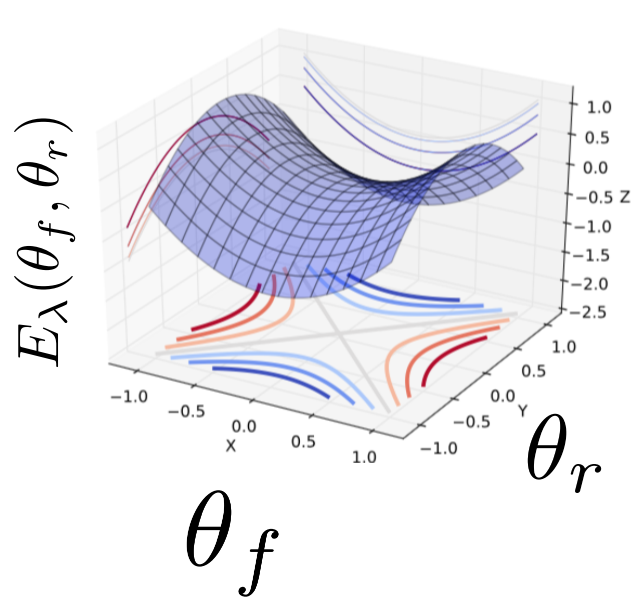
Adversarial Training for Systematics
(aka Domain Adaptation)

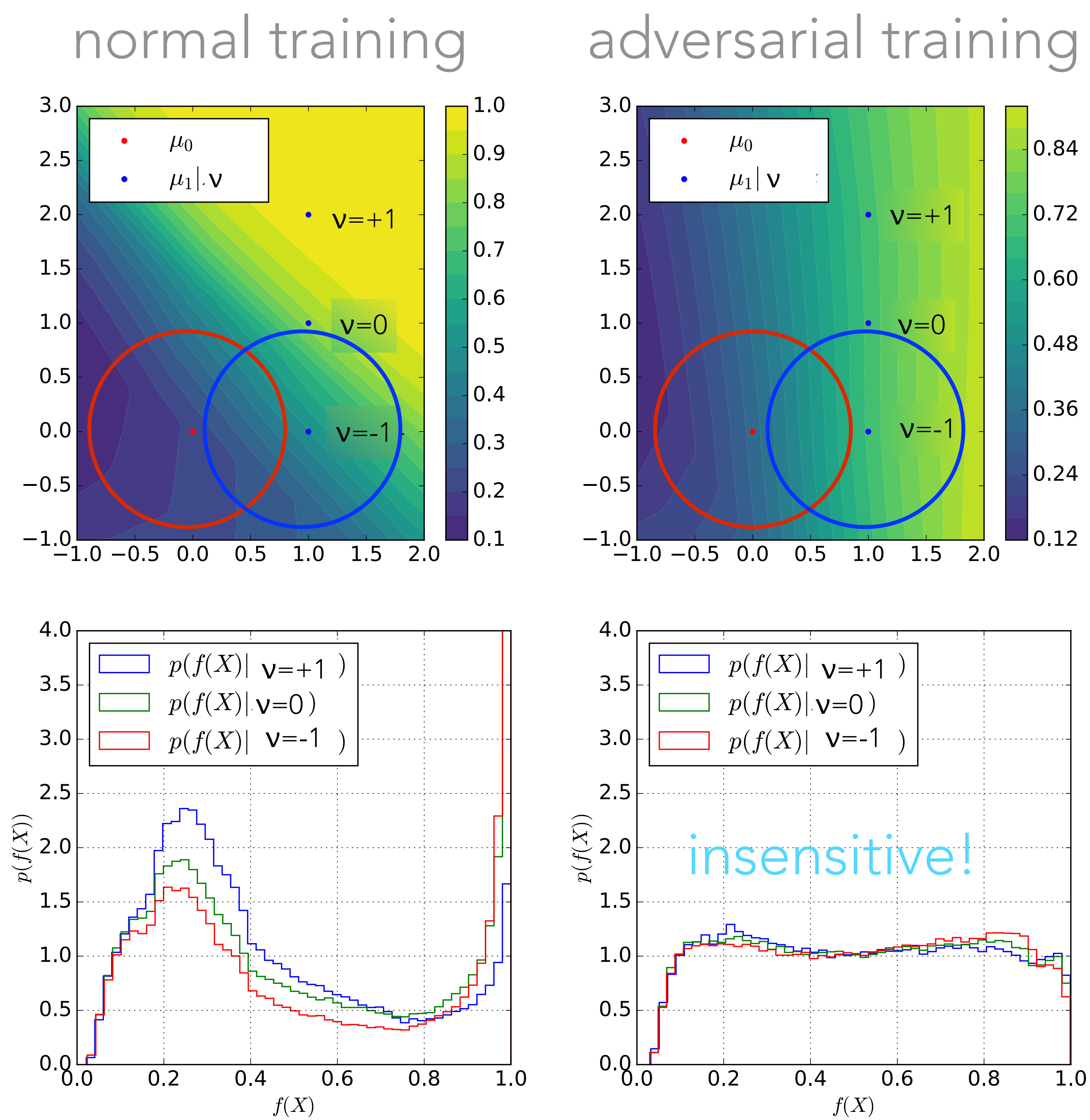
Typically classifier **f(x)** trained to minimize loss **L_f**.

- want classifier output to be insensitive to systematics (nuisance parameter **ν**)
- introduce an **adversary r** that tries to predict ν based on f.
- setup as a minimax game:

$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f}\max_{\theta_r} E(\theta_f, \theta_r).$$

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r)$$





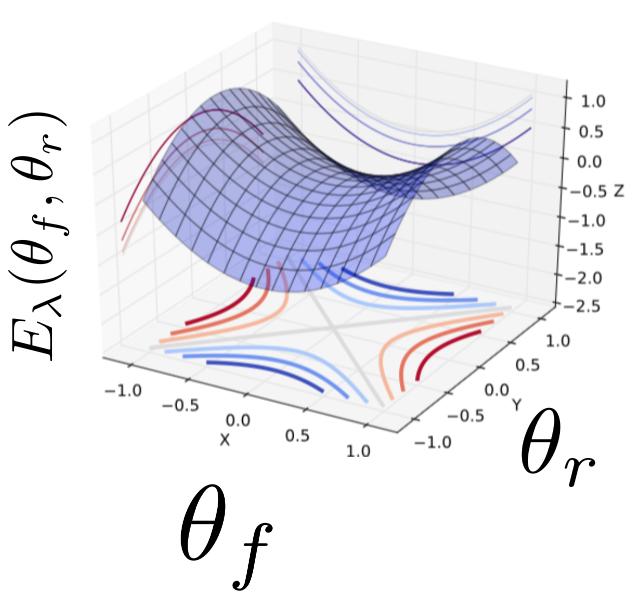
G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

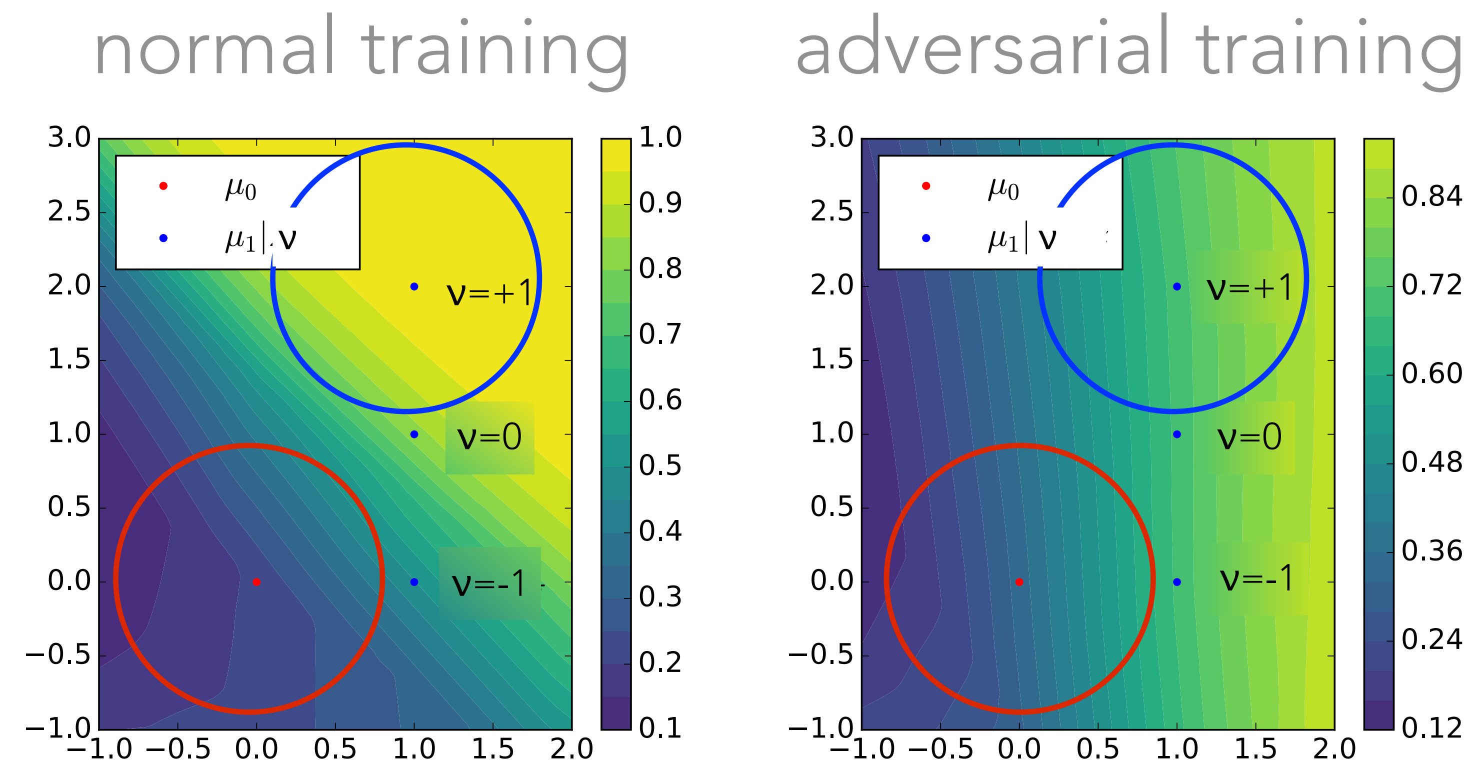
Typically classifier **f(x)** trained to minimize loss $\mathbf{L_f}$.

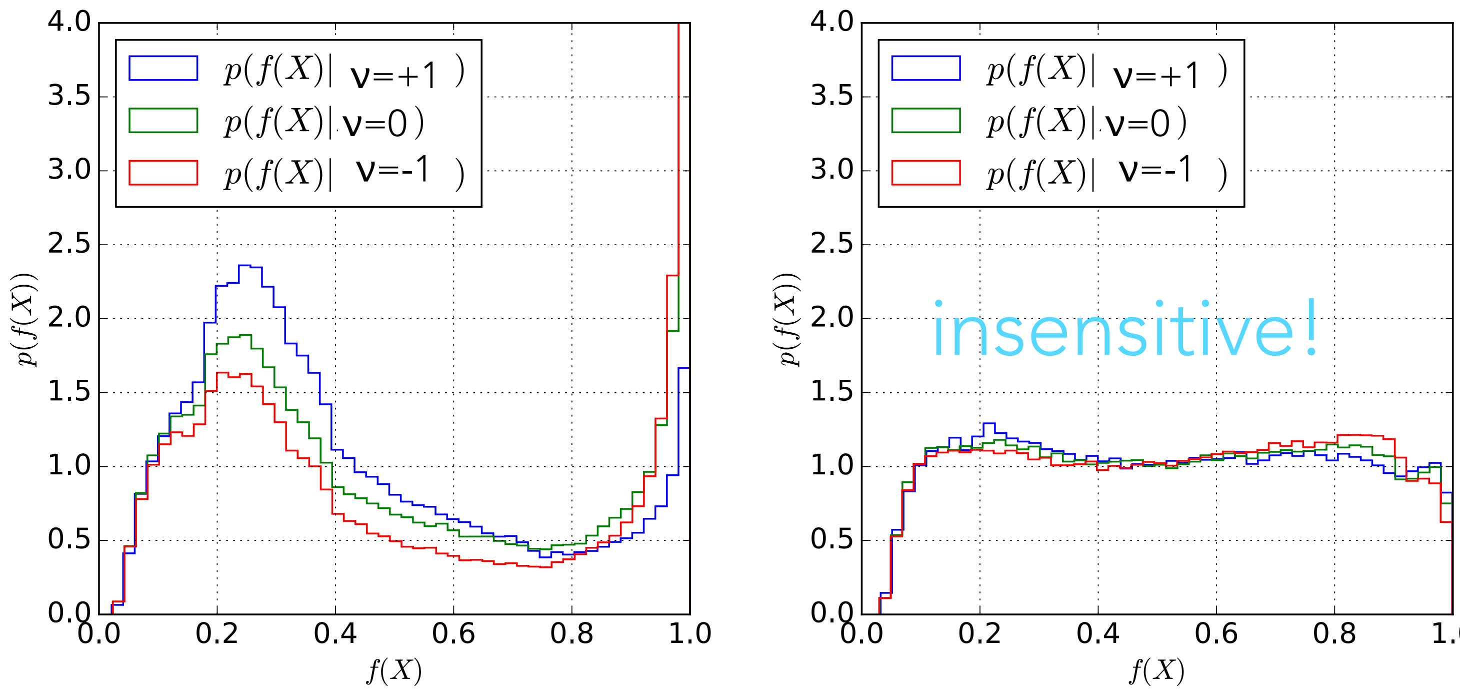
- want classifier output to be insensitive to systematics (nuisance parameter **ν**)
- introduce an **adversary r** that tries to predict ν based on f.
- setup as a minimax game:

$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f}\max_{\theta_r} E(\theta_f, \theta_r).$$

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda\mathcal{L}_r(\theta_f, \theta_r)$$







G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

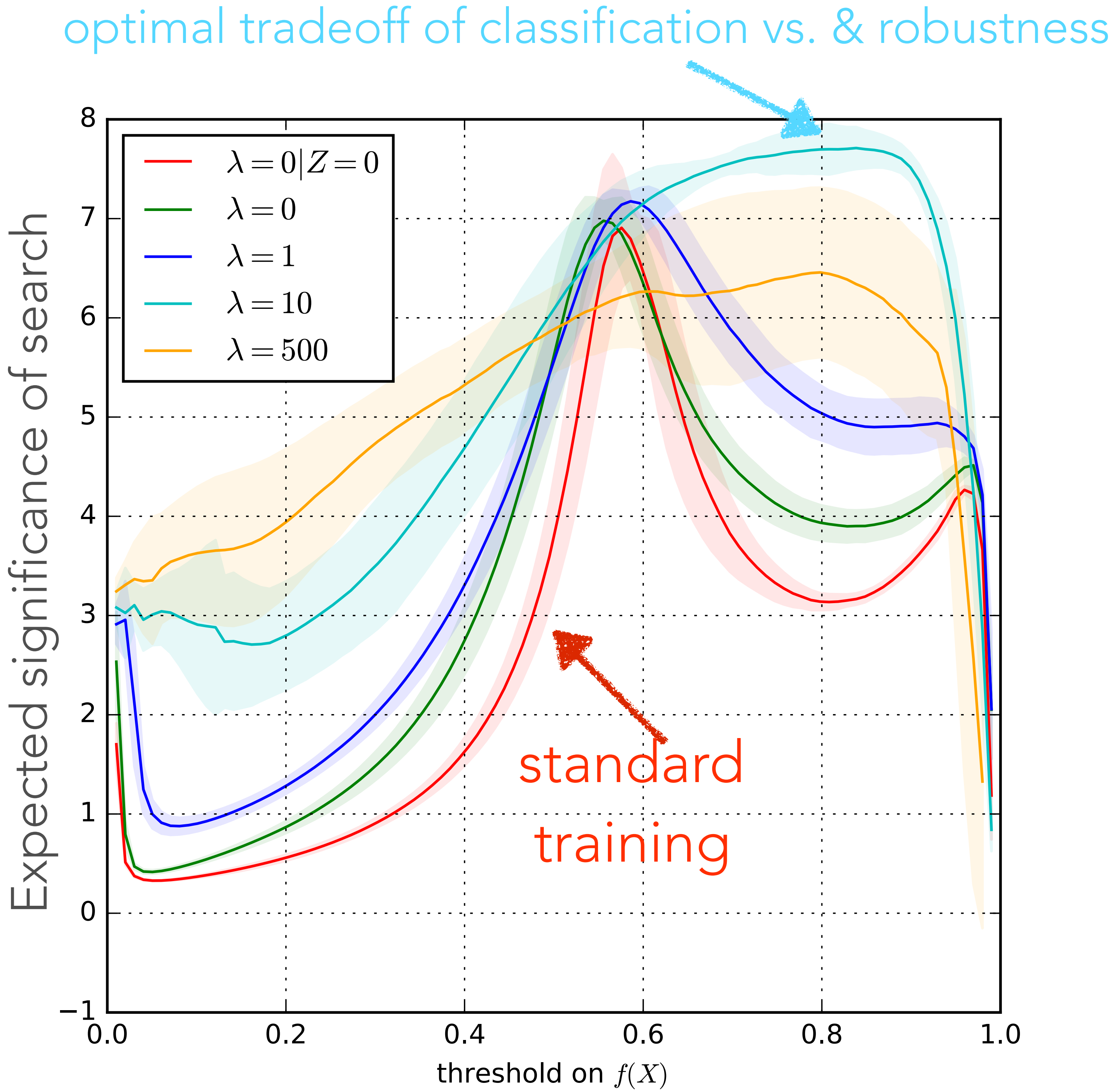
G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

Technique allows us to tune λ, the tradeoff between classification power and robustness to systematic uncertainty

An example:
background: 1000 QCD jets
signal: 100 boosted W's

Train W vs. QCD classifier

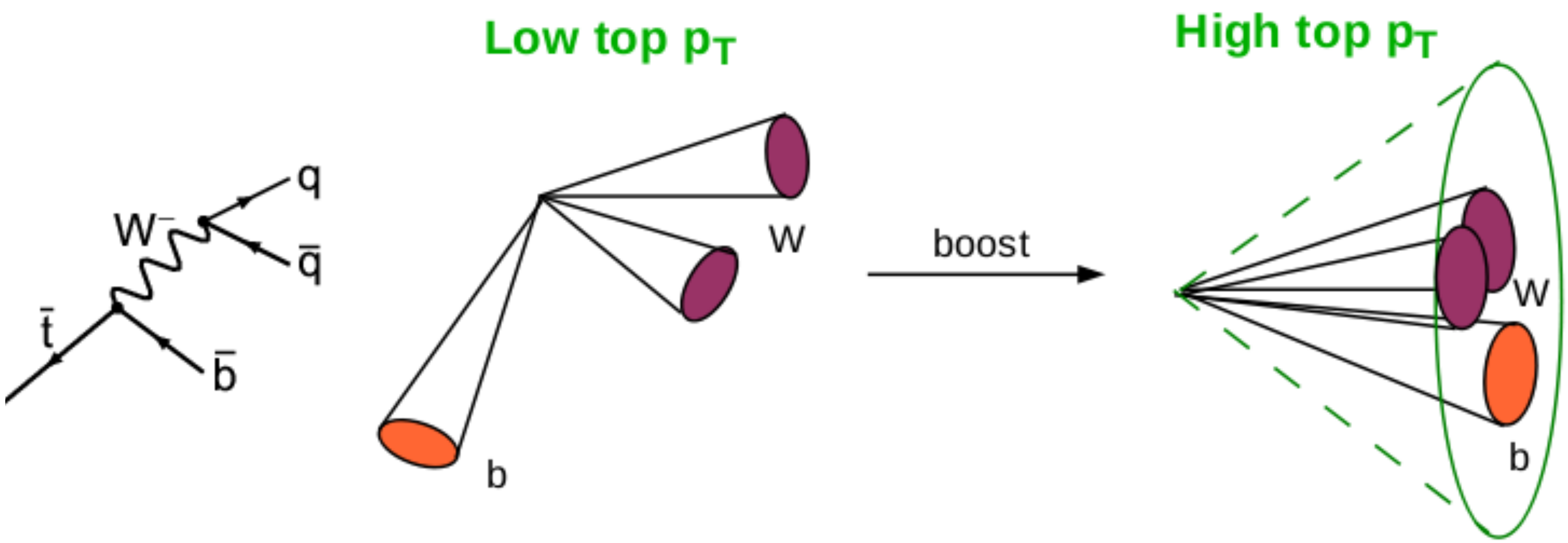
Simple cut-and-count analysis with background uncertainty.



From off-the-shelf algorithms
to physics-aware algorithms

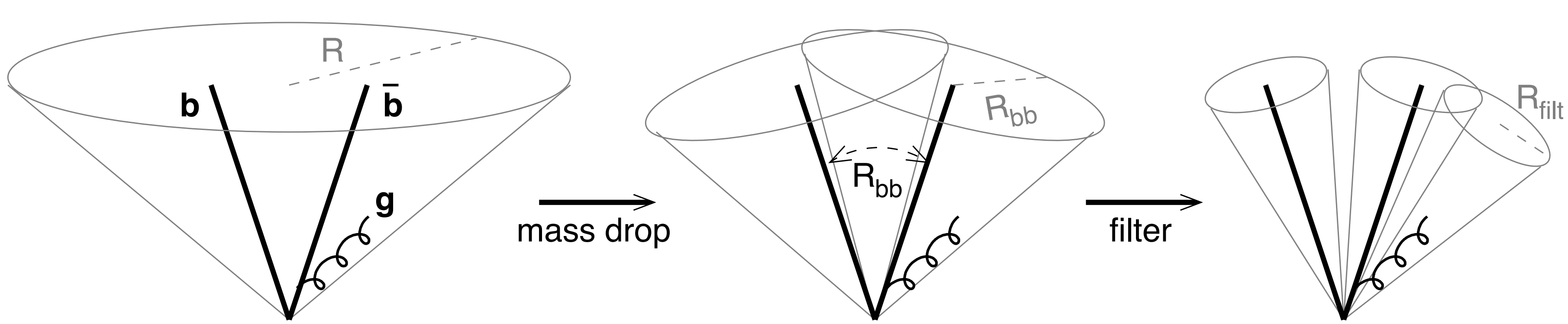
Example: Jet Substructure

Many scenarios for physics Beyond the Standard Model include highly boosted W, Z, H bosons or top quarks



Identifying these rests on subtle substructure inside jets

- an enormous number of theoretical effort in developing observables and techniques to tag jets like this



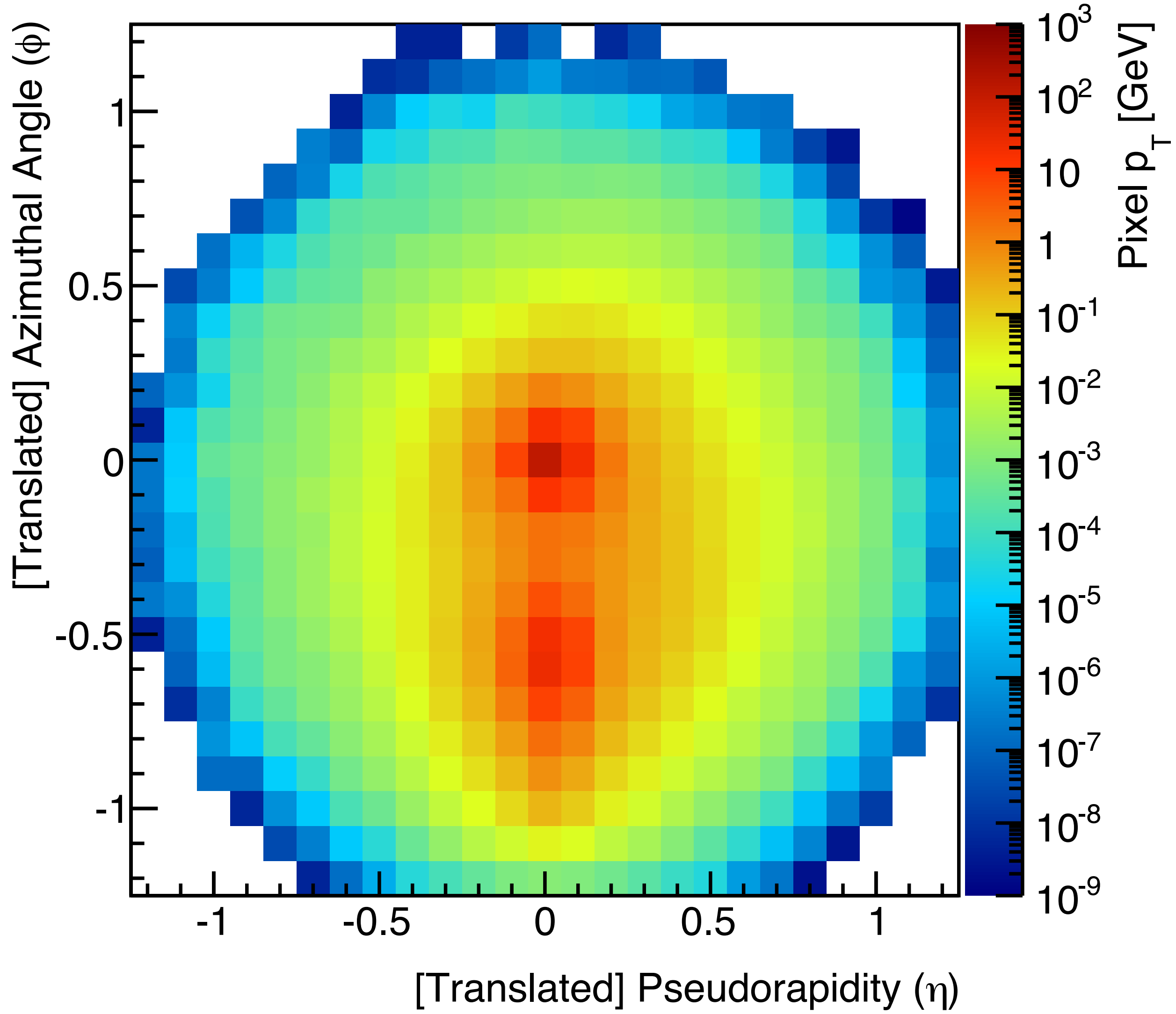
JET IMAGES

Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Dawe, et al arXiv:1609.00607

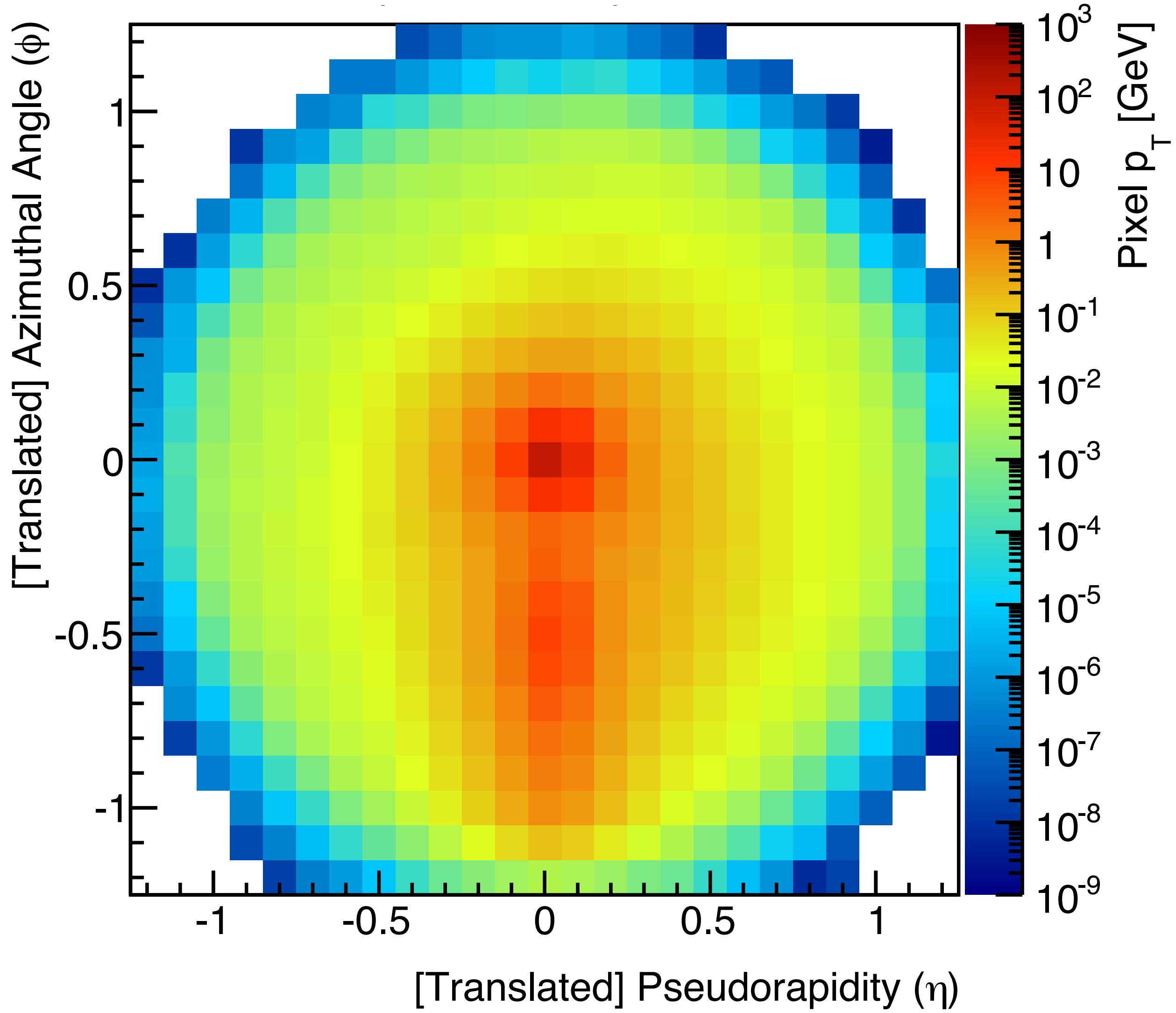
Last year deep learning algorithms applied to "jet images"

- based on fast simulation & idealized uniform calorimeter
- preprocessed to recenter (η, φ) & rotated

Average Boosted W Jet



Average QCD Jet



JET IMAGES

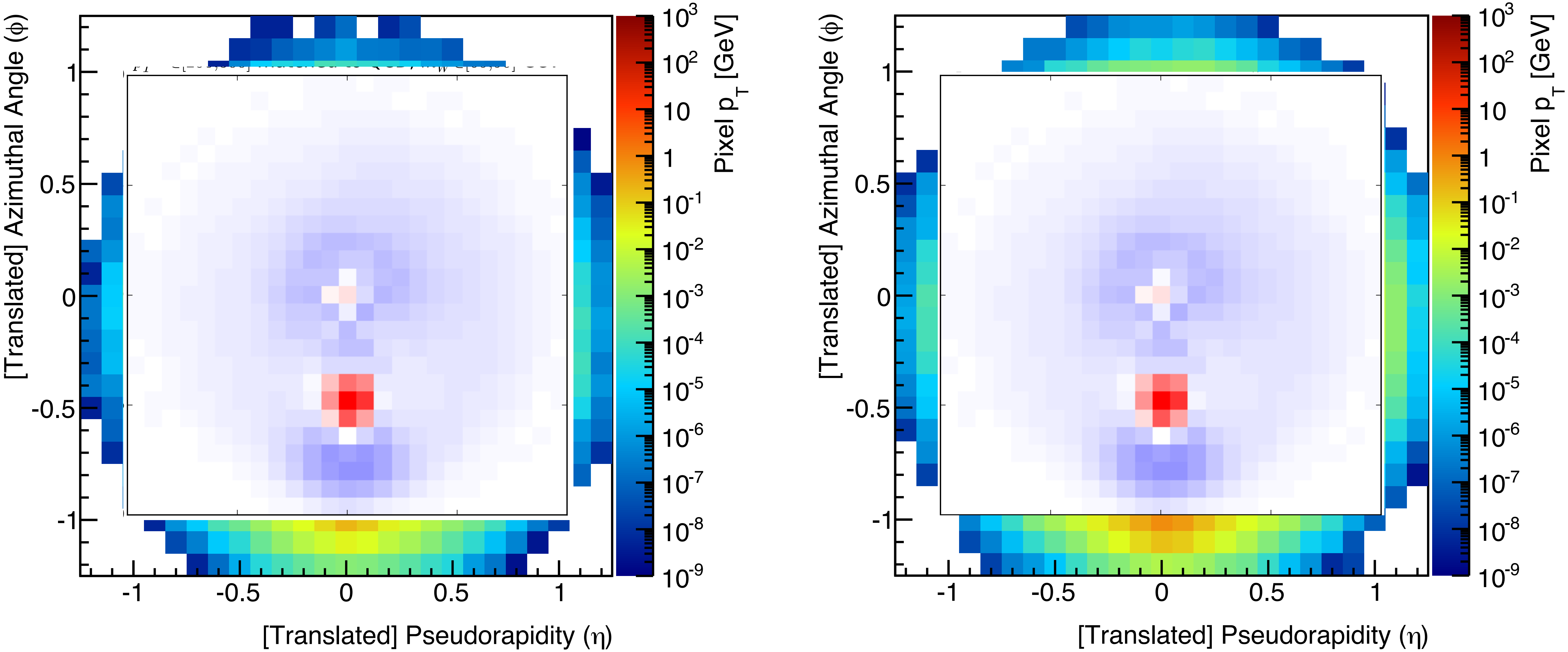
Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Dawe, et al arXiv:1609.00607

Inspecting the classifier shows parts of image that favor the W→ jj interpretation are consistent with physics intuition

- W-like QCD-like

Average Boosted W Jet

Average QCD Jet



[Slides by Max Welling]

Physics is ripe with symmetries, we should incorporate that knowledge into our models

- difficulty: often detector breaks symmetries

Symmetry in Deep Learning

What makes CNNs so effective?

- Weight sharing: exploits translation symmetry
- Depth: exploits equivariance

Network design principle:
Equivariance to symmetry transformations

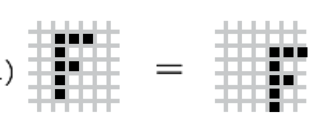
Conv vs G-Conv

Planar Convolution

"translate filter and compute inner product"

Translation

$$T_s f(x) = f(x - s)$$

$T_{(2,1)}$ 

Z^2-Convolution

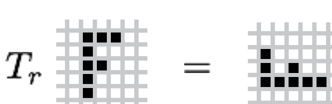
$$[f \star \psi](s) = \sum_{x \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(x) [T_s \psi]_k(x)$$

Group Convolution

"transform filter and compute inner product"

Transformation

$$T_r f(x) = f(r^{-1} x)$$

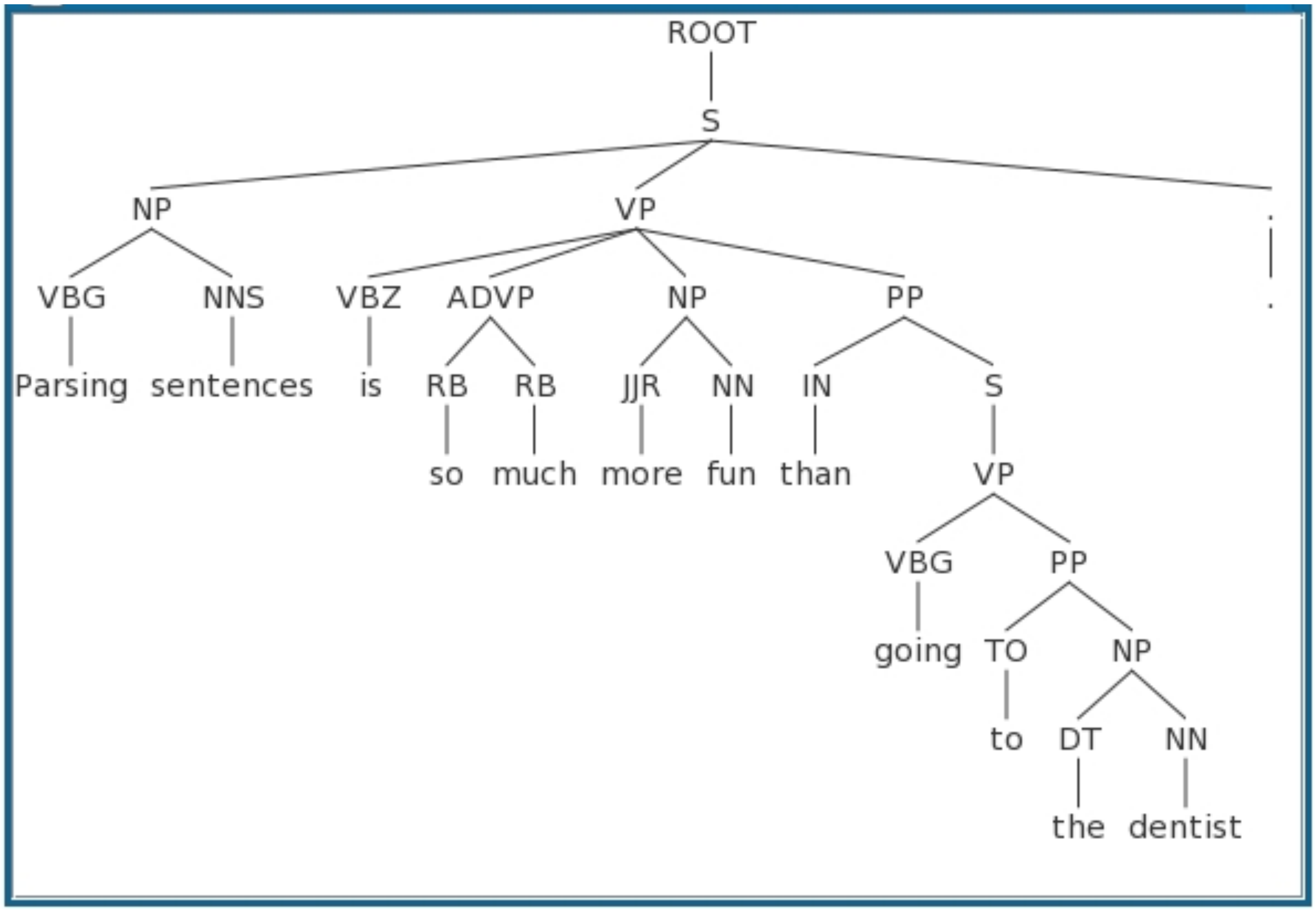
T_r 

G-Convolution

$$[f \star \psi](g) = \sum_{x \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(x) [T_g \psi]_k(x)$$

Recursive Neural Networks showing great performance for Natural Language Processing tasks

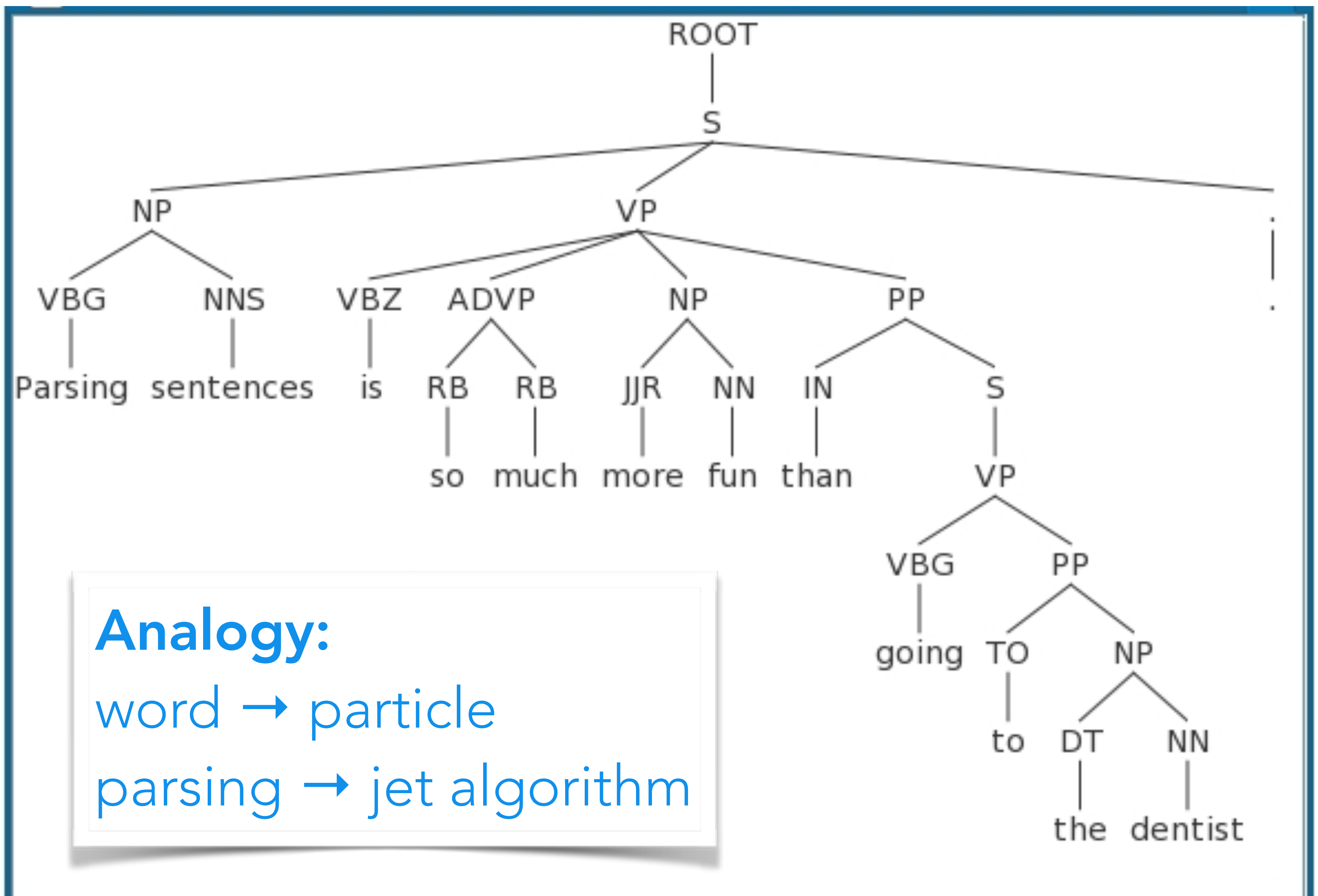
- neural network's topology given by parsing of sentence!



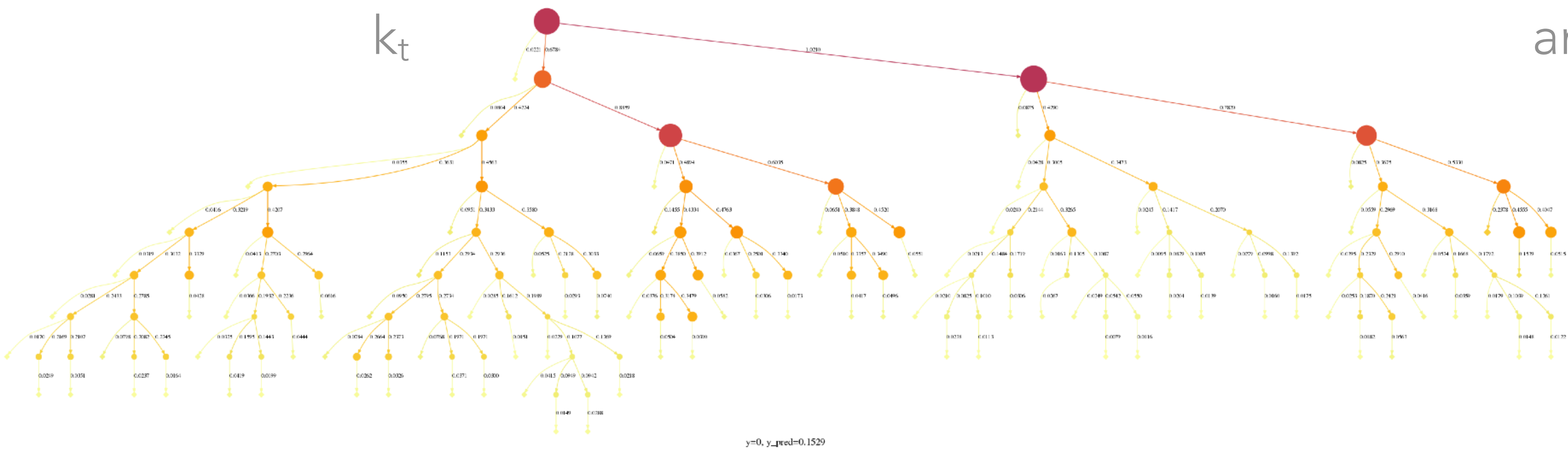
FROM IMAGES TO SENTENCES

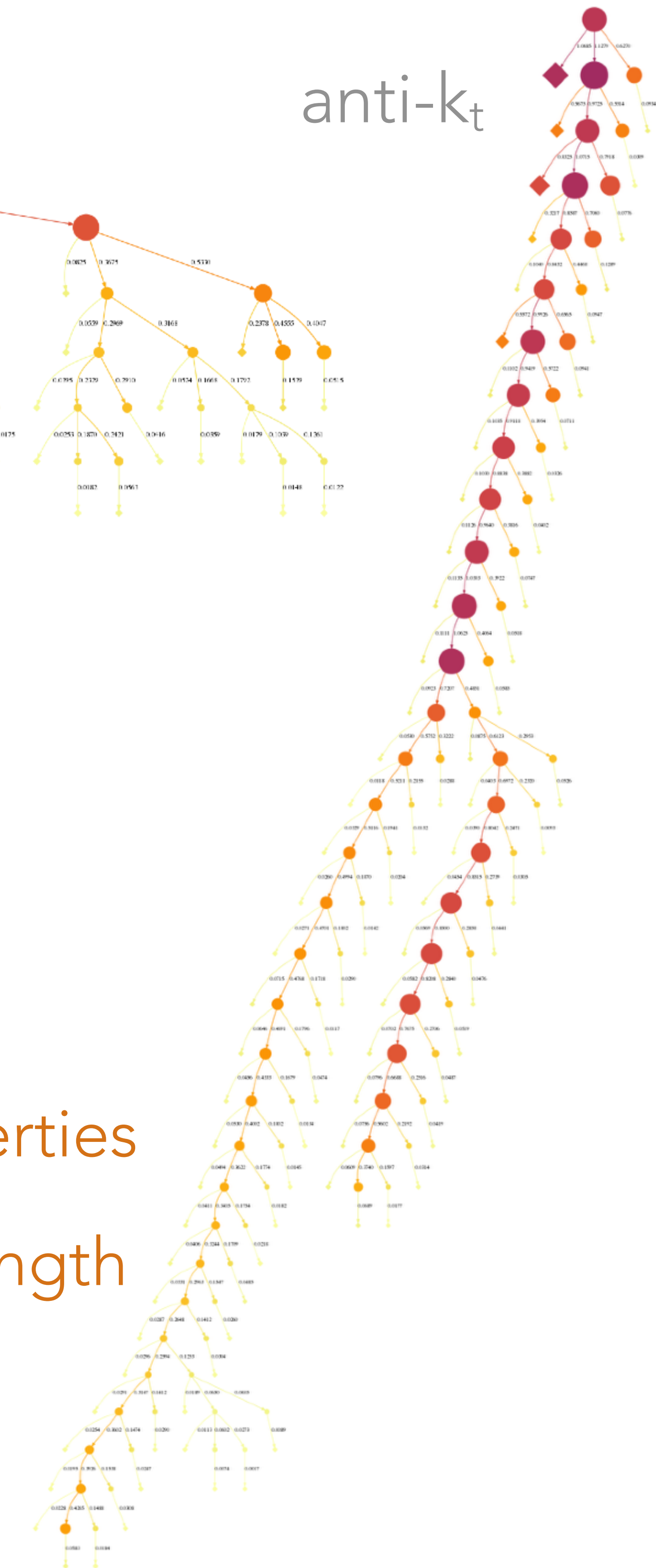
Recursive Neural Networks showing great performance for Natural Language Processing tasks

- neural network's topology given by parsing of sentence!



QCD-INSPIRED RECURSIVE NEURAL NETWORKS





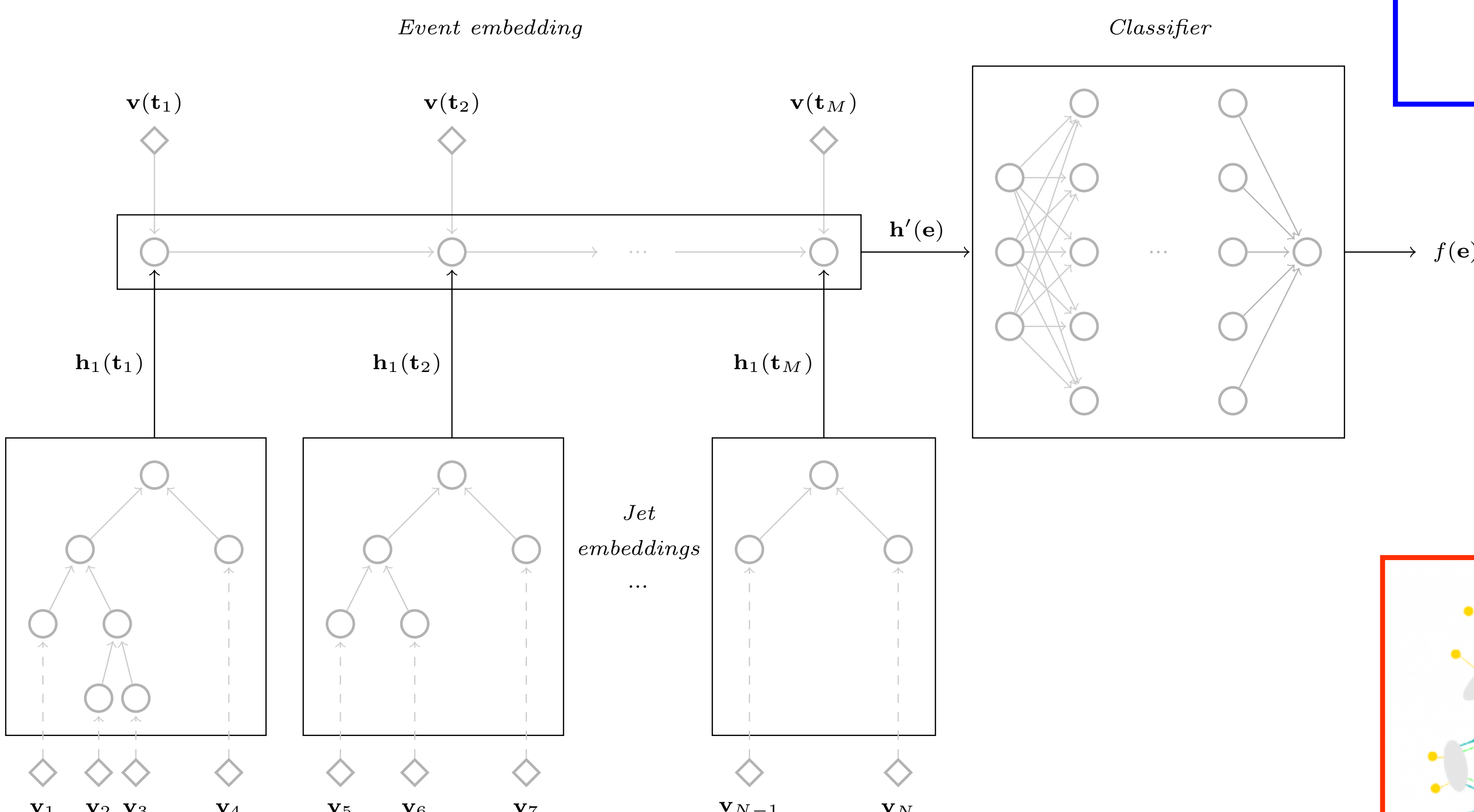
Work with Gilles Louppe, Kyunghyun Cho, Cyril Becot (arXiv:1702.00748)

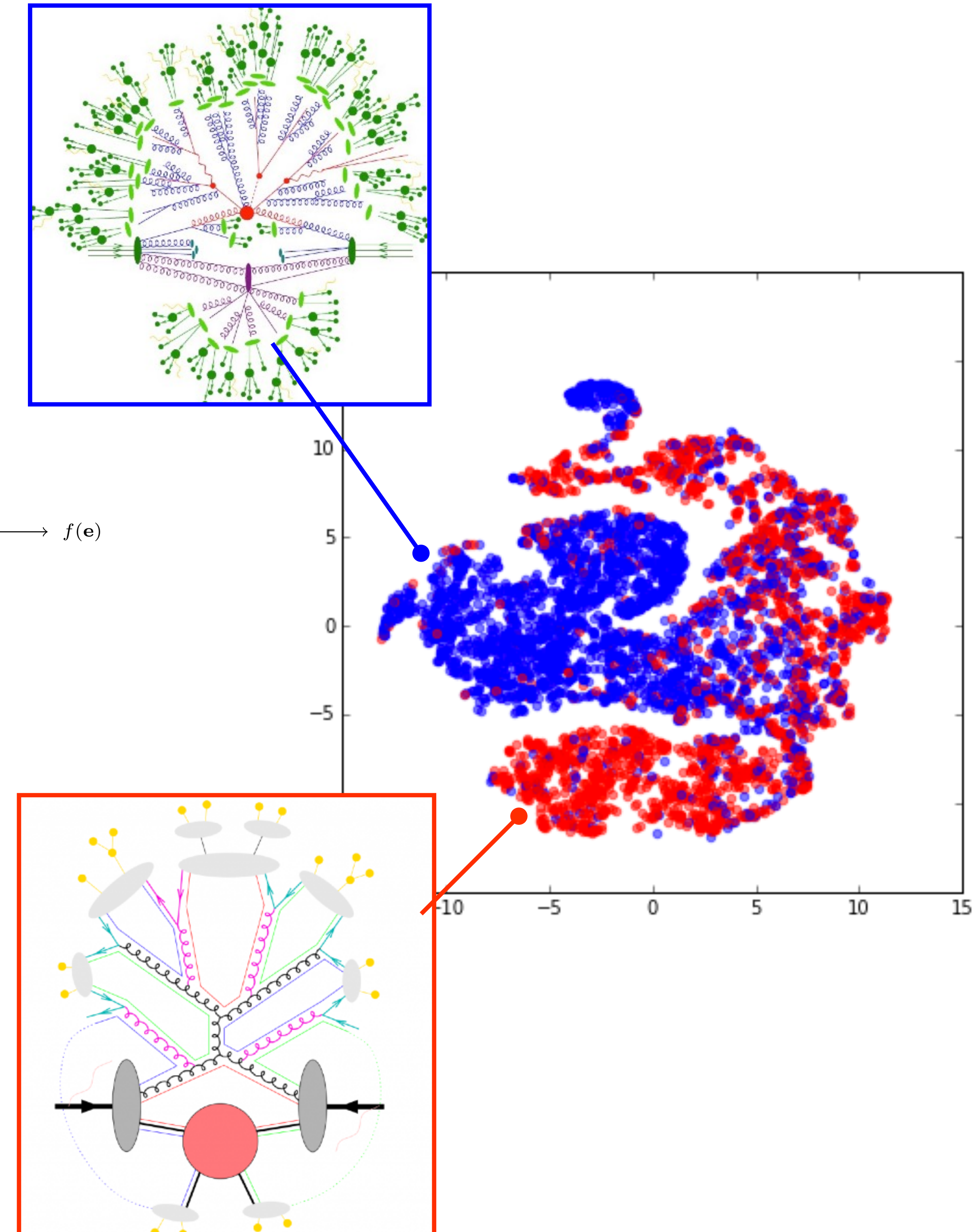
- Use sequential recombination jet algorithms to provide network topology (on a per-jet basis)
- path towards ML models with good physics properties
- Top node of recursive network provides a fixed-length **embedding** of a jet that can be fed to a classifier

EVENT EMBEDDINGS

Jointly optimize jet embedding → event embedding → classifier

It scales!



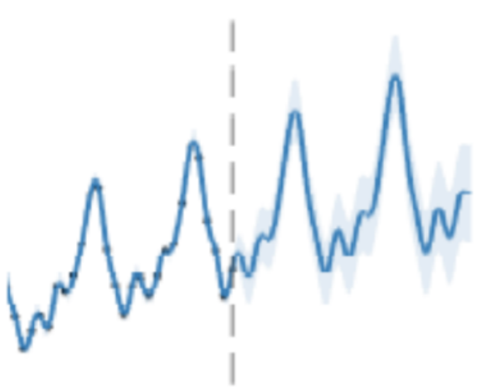


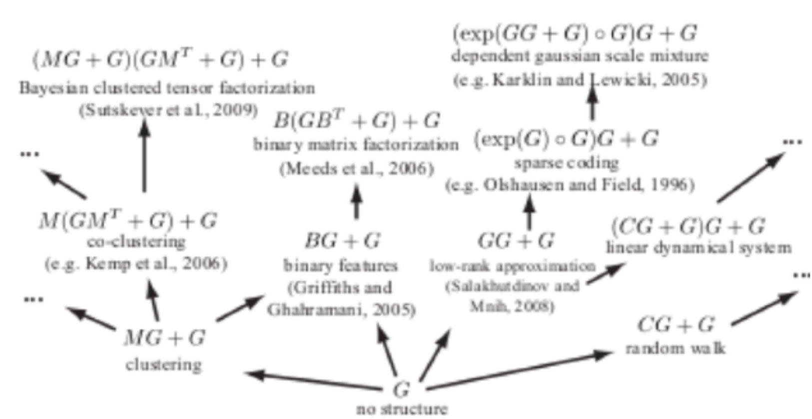
SEARCHING OVER SPACE OF MODELS

Using a class of models known as Gaussian Processes to model data

- physics goes into the construction of a "Kernel" that describes covariance of data

Vocabulary of kernels + grammar for composition

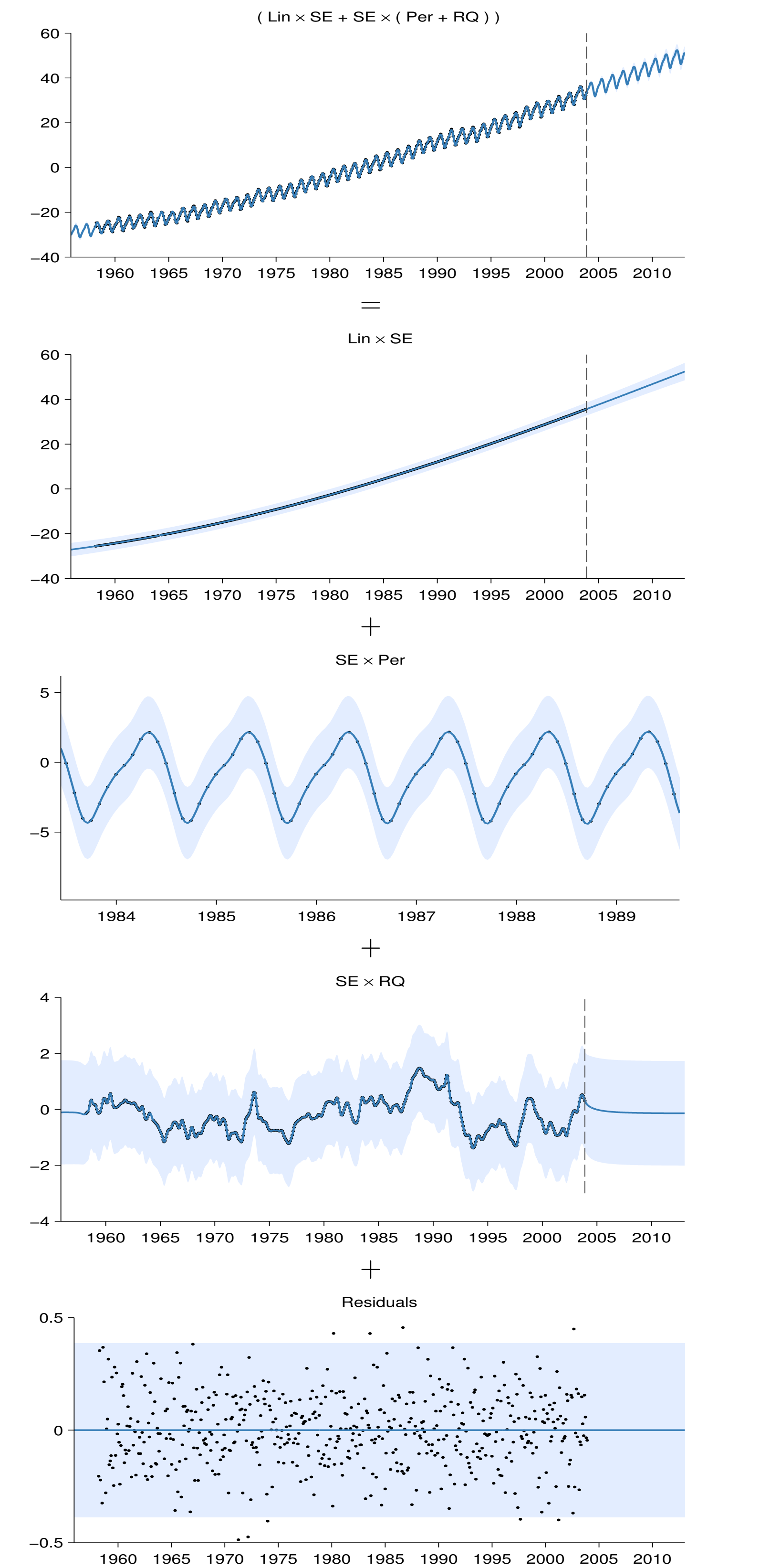




Structure Discovery in Nonparametric Regression through Compositional Kernel Search
David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani
International Conference on Machine Learning, 2013
pdf | code | poster | bibtex

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

Mauna Loa atmospheric CO_2



From Reproducibility
To Reusability

[work with Lukas Heinrich]

The BSM-AI project: SUSY-AI – generalizing LHC limits on supersymmetry with machine learning

Sascha Caron,[a,b] **Jong Soo Kim,**[c] **Krzysztof Rolbiecki,**[c,d]
Roberto Ruiz de Austri,[e] **Bob Stienen**[a]

[a] *Institute for Mathematics, Astro- and Particle Physics IMAPP, Radboud Universiteit, Nijmegen, The Netherlands*

[b] *Nikhef, Amsterdam, The Netherlands*

[c] *Instituto de Física Teórica UAM/CSIC, Madrid, Spain*

[d] *Faculty of Physics, University of Warsaw, Warsaw, Poland*

[e] *Instituto de Física Corpuscular, IFIC-UV/CSIC, Valencia, Spain*

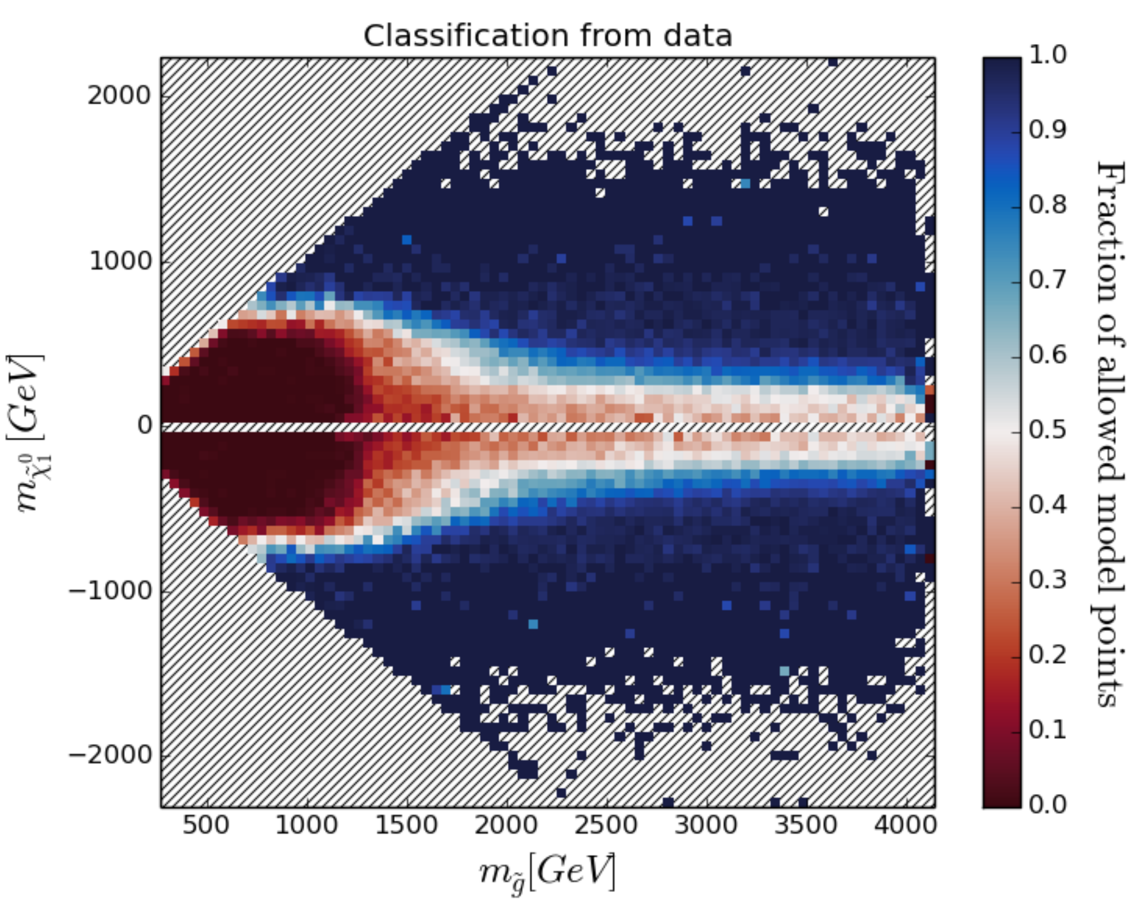
Accelerating the BSM interpretation of LHC data with machine learning

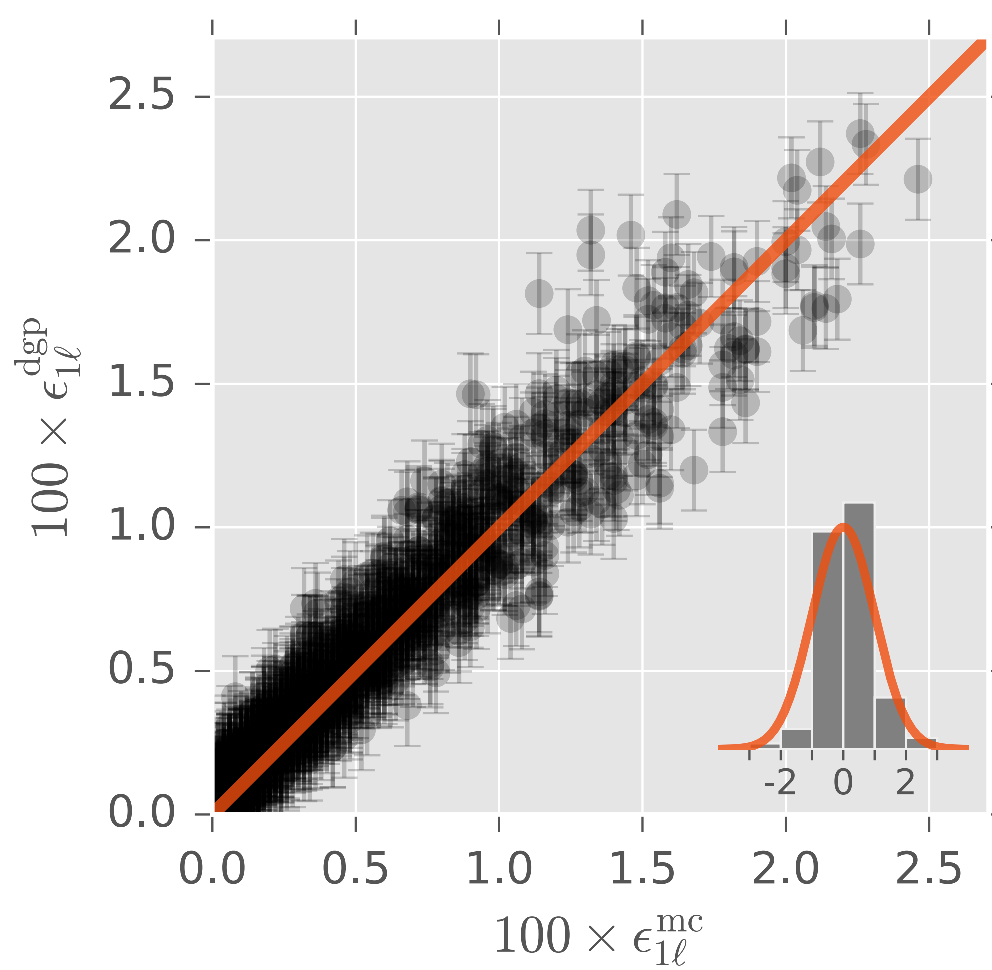
Gianfranco Bertone,[1] Marc Peter Deisenroth,[2] Jong Soo Kim,[3]
Sebastian Liem,[1] Roberto Ruiz de Austri,[4] and Max Welling[5]

[1] *GRAPPA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, Netherlands*
[2] *Department of Computing, Imperial College London, 180 Queen's Gate, SW7 2AZ London, United Kingdom*
[3] *Center for Theoretical Physics of the Universe, Institute for Basic Science (IBS), Daejeon, 34051, Korea and Instituto de Física Teórica UAM/CSIC, Madrid, Spain*
[4] *Instituto de Física Corpuscular IFIC-UV/CSIC, Valencia, Spain*
[5] *Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, Netherlands*

(Dated: November 10, 2016)

The interpretation of Large Hadron Collider (LHC) data in the framework of Beyond the Standard Model (BSM) theories is hampered by the need to run computationally expensive event generators and detector simulators. Performing statistically convergent scans of high-dimensional BSM theories is consequently challenging, and in practice unfeasible for very high-dimensional BSM theories. We present here a new machine learning method that accelerates the interpretation of LHC data, by learning the relationship between BSM theory parameters and data. As a proof-of-concept, we demonstrate that this technique accurately predicts natural SUSY signal events in two signal regions at the High Luminosity LHC, up to four orders of magnitude faster than standard techniques. The new approach makes it possible to rapidly and accurately reconstruct the theory parameters of complex BSM theories, should an excess in the data be discovered at the LHC.






It’s the difference between if you had airplanes
where you threw away an airplane after every flight,
versus you could reuse them multiple times.
— Elon Musk


analyses pipeline
analysis pipeline
testing
one theory
It’s the difference between if you had ~~airplanes~~
where you threw away an ~~airplane~~ after ~~every flight~~,
versus you could reuse them multiple times.
— Elon Musk

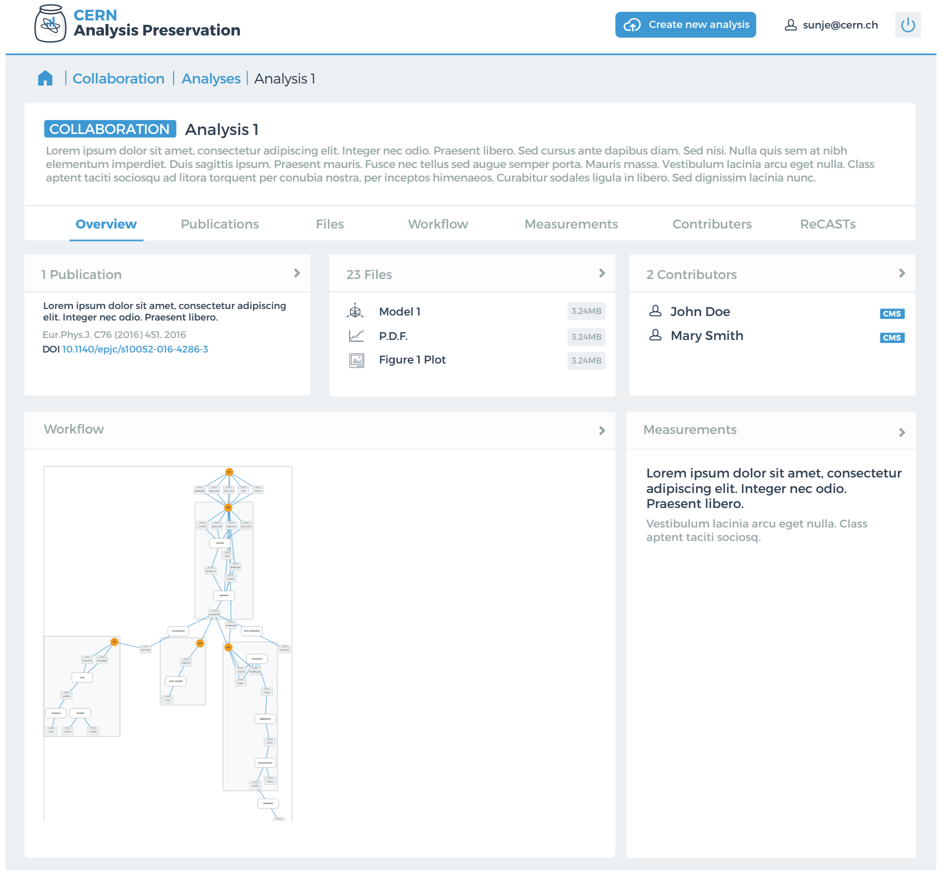
Technical Solution:

Workflow (i.e. logic which steps to run in which order: reconstruction → analysis→ fit)

- in easy to write / read text based format (YAML)
- generic workflow language “**yadage**” based on graphs. No assumption on how you run your analysis. Should be able to accommodate your workflows.
- integrated into CERN Analysis Preservation.
- re-run workflow using tool that interprets info stored in CAP

workflow	software	data
eventsel.yml	docker img	data, bkgds
fit.yml	docker img	
workflow.yml		

import analysis workflow



CERN Analysis Preservation

https://arxiv.org/pdf/1706.01878.pdf

SOFTWARE

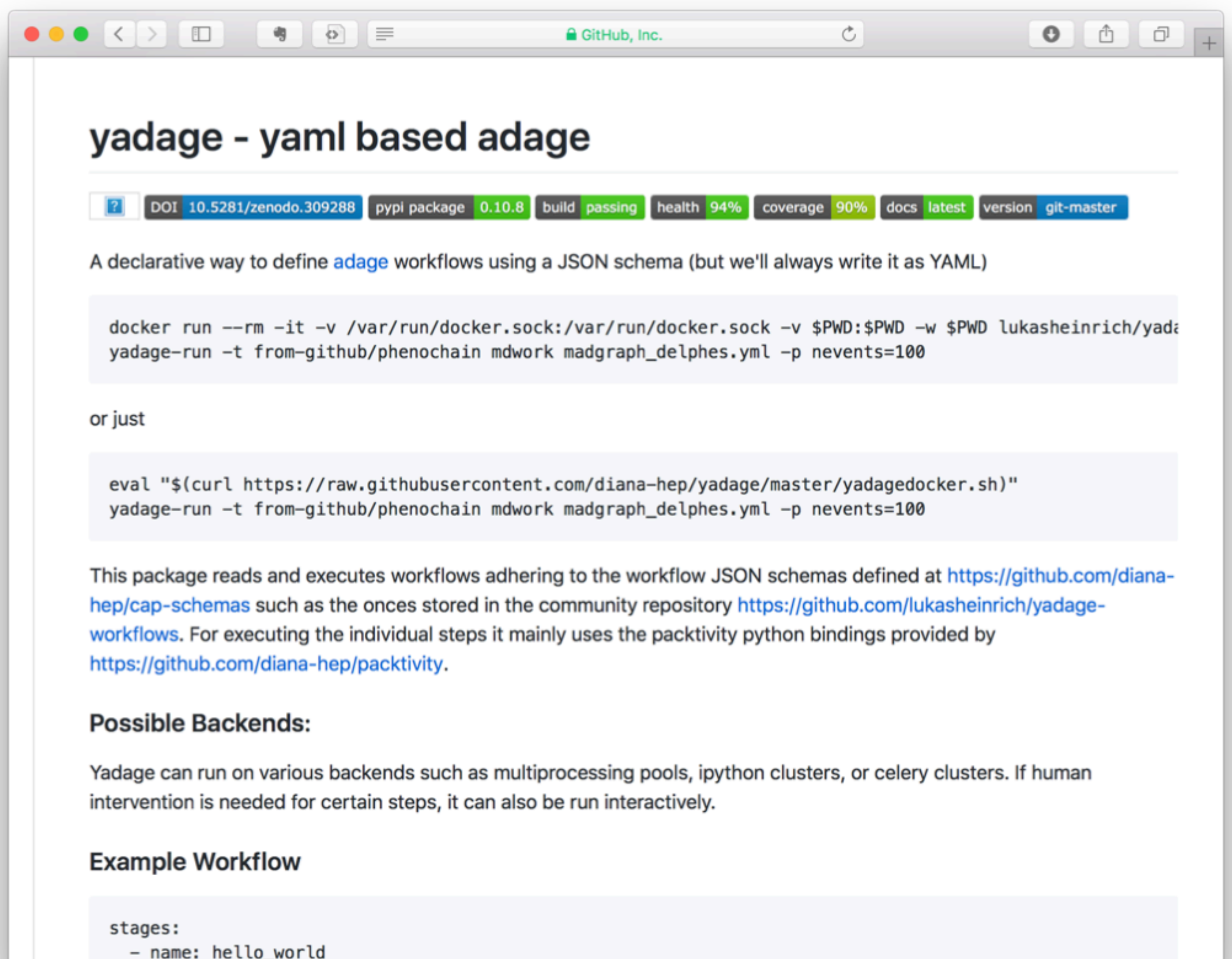
Yadage and Packtivity – analysis preservation using parametrized workflows

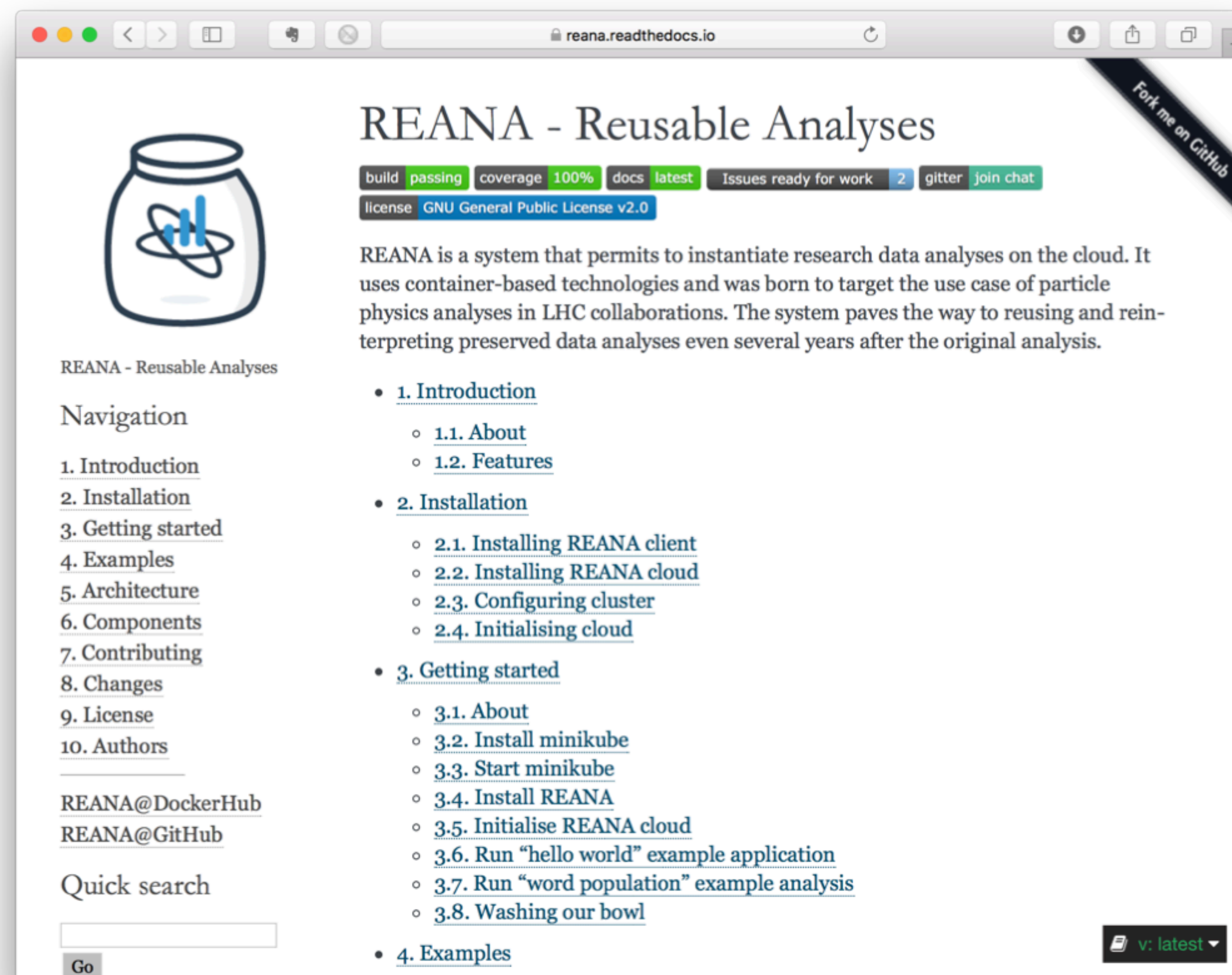
Kyle Cranmer[1] **and Lukas Heinrich**[1]

[1] Department of Physics, New York University, New York, USA

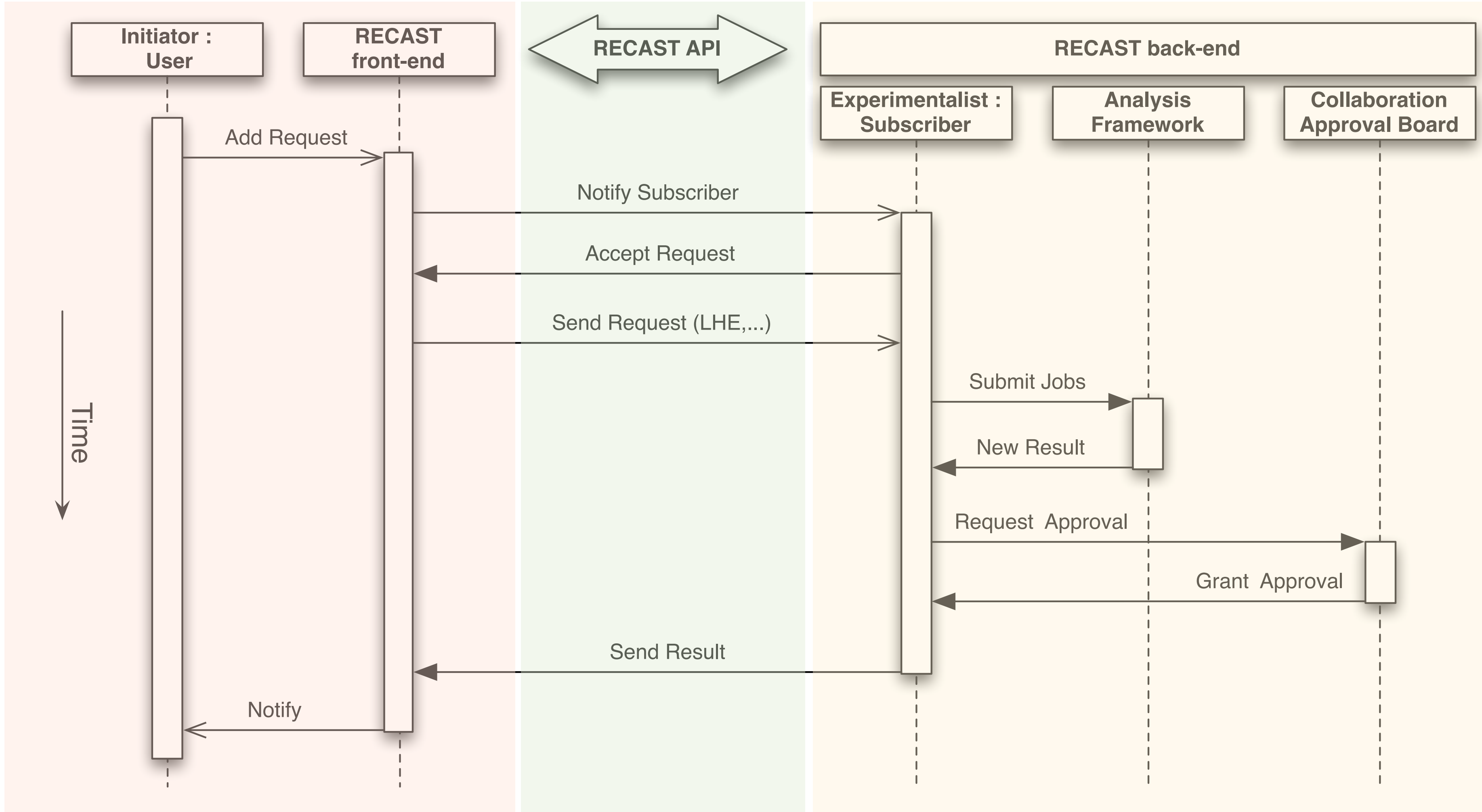
E-mail: `lukas.heinrich@cern.ch`

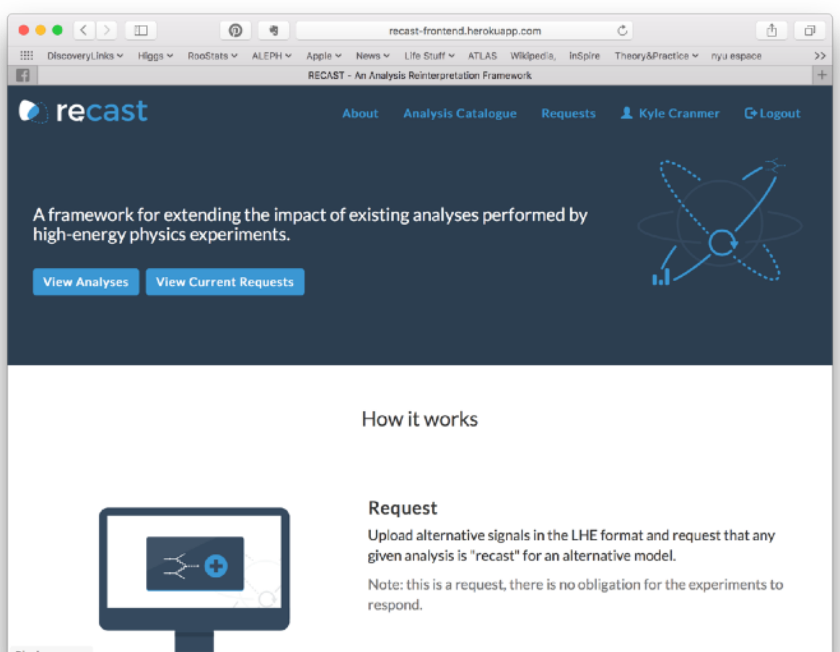
Abstract. Preserving data analyses produced by the collaborations at LHC in a parametrized fashion is crucial in order to maintain reproducibility and re-usability. We argue for a declarative description in terms of individual processing steps – "packtivities" – linked through a dynamic directed acyclic graph (DAG) and present an initial set of JSON schemas for such a description and an implementation – "yadage" – capable of executing workflows of analysis preserved via Linux containers.



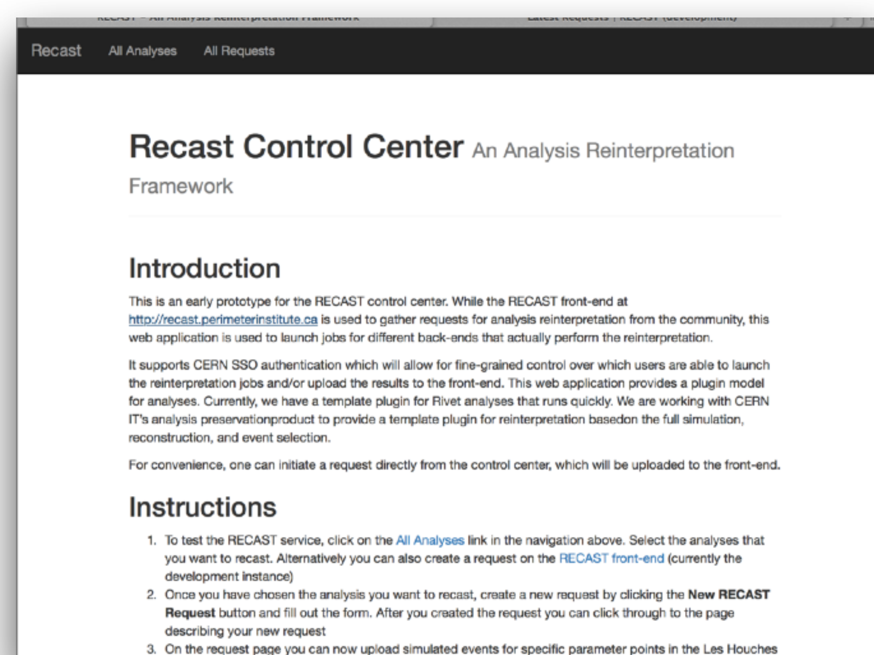


https://arxiv.org/pdf/1706.01878.pdf

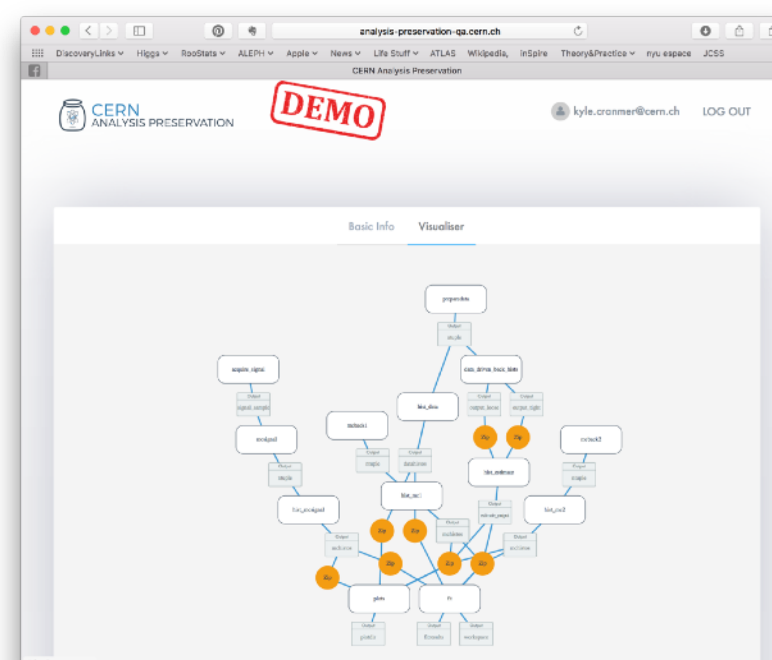




Front-End: public facing collects requests



Control Center: not public, uses CERN auth., oversees processing of jobs on back-end



CERN Analysis Preservation: Stores workflows, provides back-end computing resources

Black Box Optimization
Bayesian Optimization

Bayesian optimisation

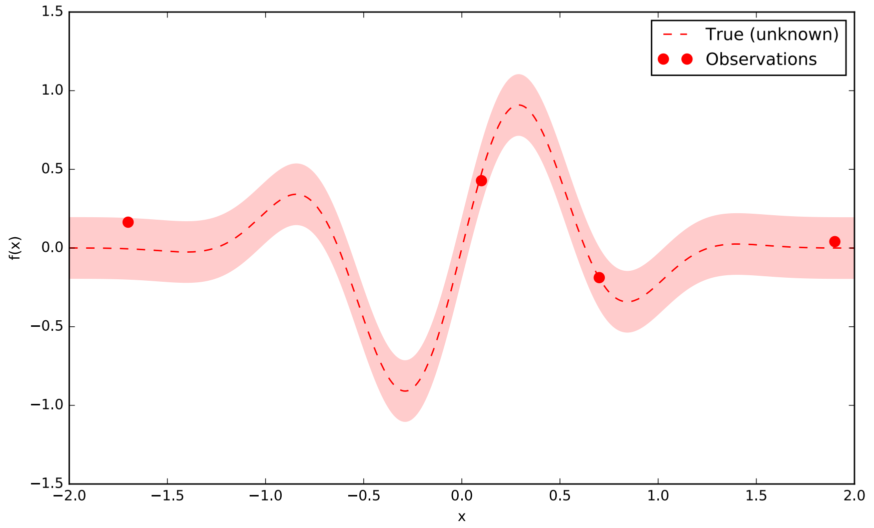
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

$$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Where shall we sample next?



Bayesian optimisation

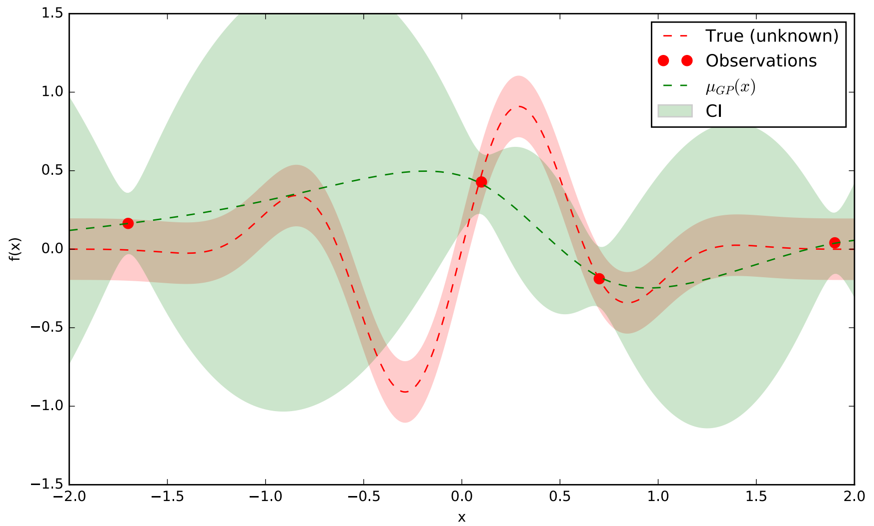
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

$$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

BAYES OPT IN A NUTSHELL

[slides from Gilles Loupppe]

Bayesian optimisation

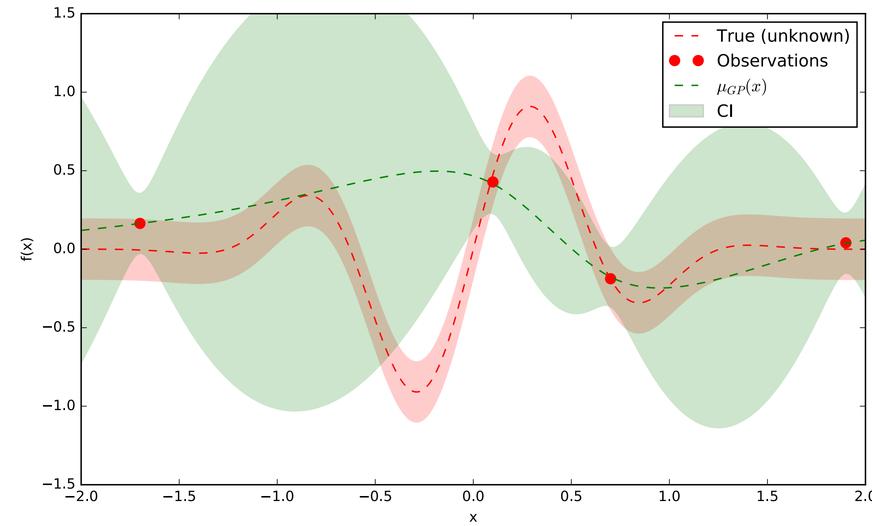
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

$$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

Acquisition functions

Acquisition functions u(x) specify which sample x should be tried next:

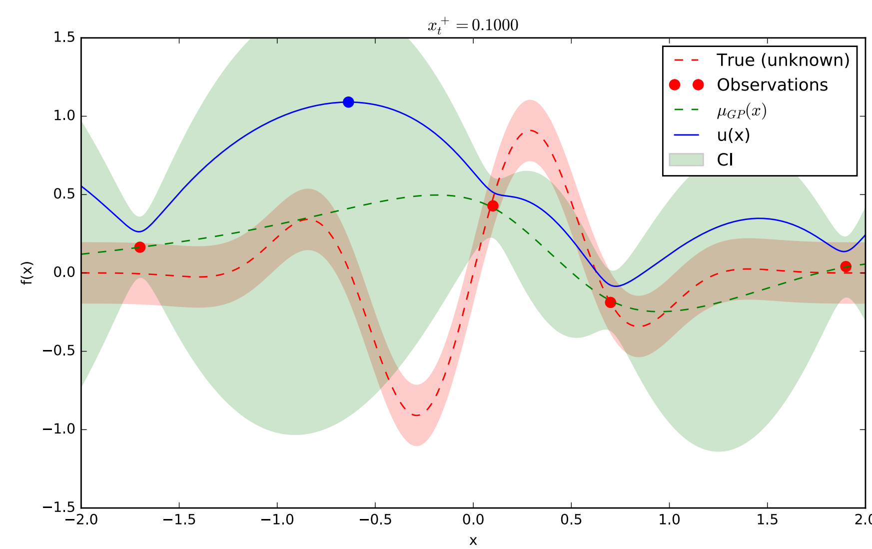
- Upper confidence bound $\text{UCB}(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$;
- Probability of improvement $\text{PI}(x) = P(f(x) \geq f(x_t^+) + \kappa)$;
- Expected improvement $\text{EI}(x) = \mathbb{E}[f(x) - f(x_t^+)]$;
- ... and many others.

where x_t^+ is the best point observed so far.

In most cases, acquisition functions provide knobs (e.g., κ) for controlling the exploration-exploitation trade-off.

- Search in regions where $\mu_{GP}(x)$ is high (exploitation)
- Probe regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

Plugging everything together ($t = 0$)



$$x_{t+1} = \arg\max_x \text{UCB}(x)$$

BAYES OPT IN A NUTSHELL

[slides from Gilles Louppe]

Bayesian optimisation

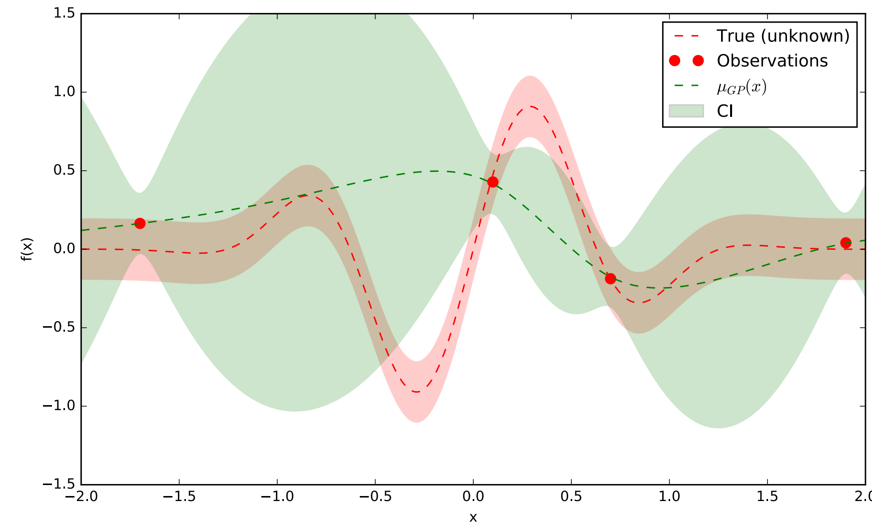
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

   $$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

Acquisition functions

Acquisition functions $u(x)$ specify which sample x should be tried next:

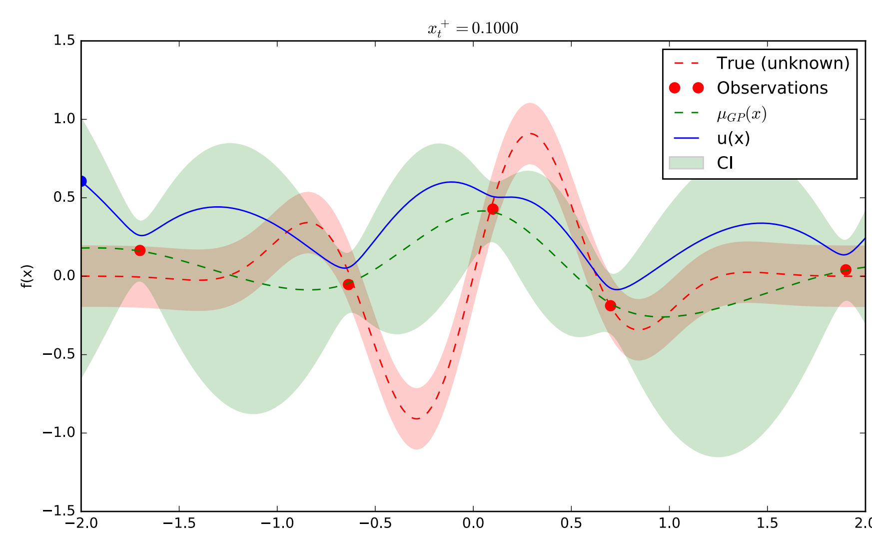
- Upper confidence bound $UCB(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$;
- Probability of improvement $PI(x) = P(f(x) \geq f(x_t^+) + \kappa)$;
- Expected improvement $EI(x) = \mathbb{E}[f(x) - f(x_t^+)]$;
- ... and many others.

where x_t^+ is the best point observed so far.

In most cases, acquisition functions provide knobs (e.g., κ) for controlling the exploration-exploitation trade-off.

- Search in regions where $\mu_{GP}(x)$ is high (exploitation)
- Probe regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

... and repeat until convergence ($t = 1$)



BAYES OPT IN A NUTSHELL

[slides from Gilles Louppe]

Bayesian optimisation

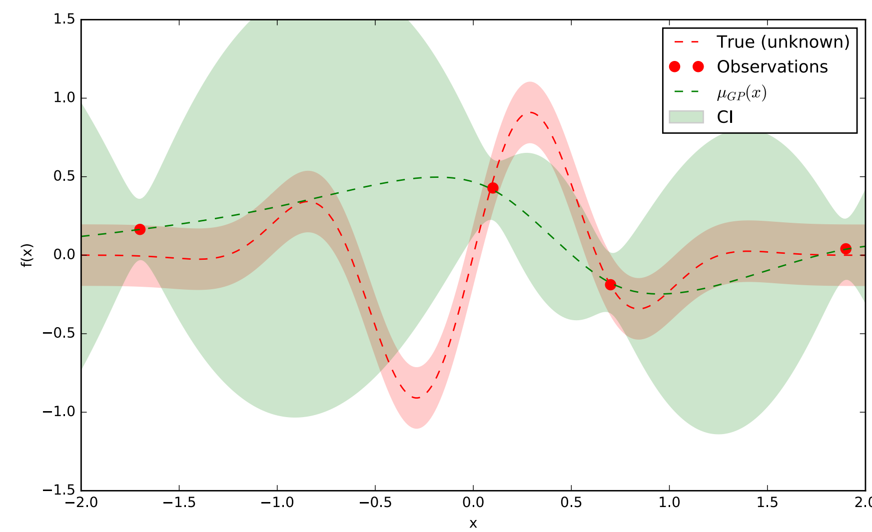
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

   $$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

Acquisition functions

Acquisition functions u(x) specify which sample x should be tried next:

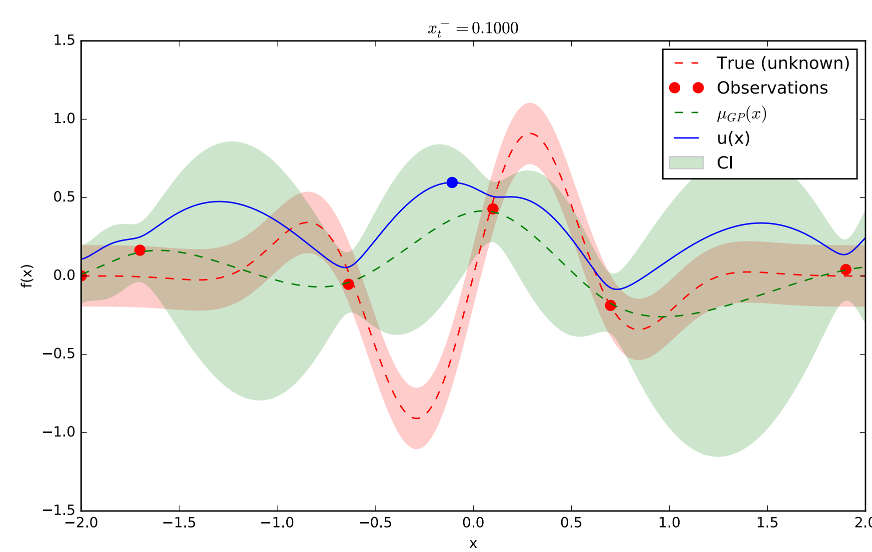
- Upper confidence bound $\text{UCB}(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$;
- Probability of improvement $\text{PI}(x) = P(f(x) \geq f(x_t^+) + \kappa)$;
- Expected improvement $\text{EI}(x) = \mathbb{E}[f(x) - f(x_t^+)]$;
- ... and many others.

where x_t^+ is the best point observed so far.

In most cases, acquisition functions provide knobs (e.g., κ) for controlling the exploration-exploitation trade-off.

- Search in regions where $\mu_{GP}(x)$ is high (exploitation)
- Probe regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

... and repeat until convergence ($t = 2$)



BAYES OPT IN A NUTSHELL

[slides from Gilles Louppe]

Bayesian optimisation

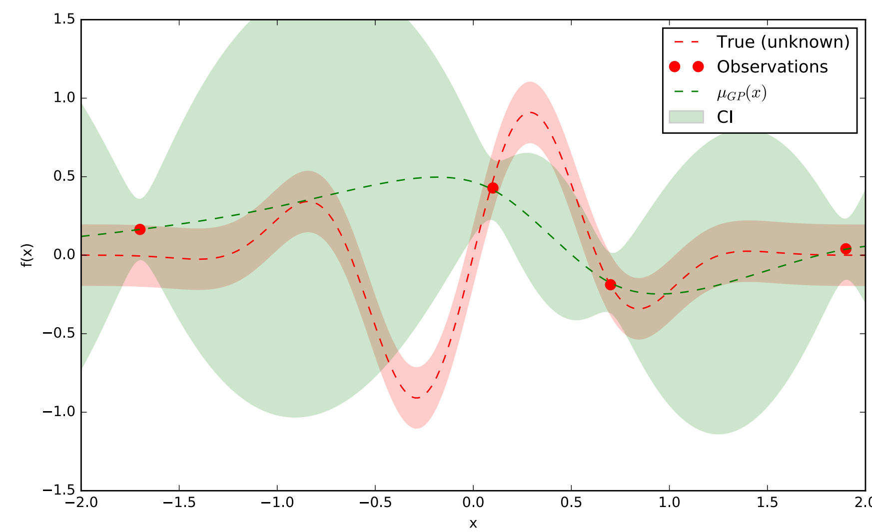
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

   $$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

Acquisition functions

Acquisition functions u(x) specify which sample x should be tried next:

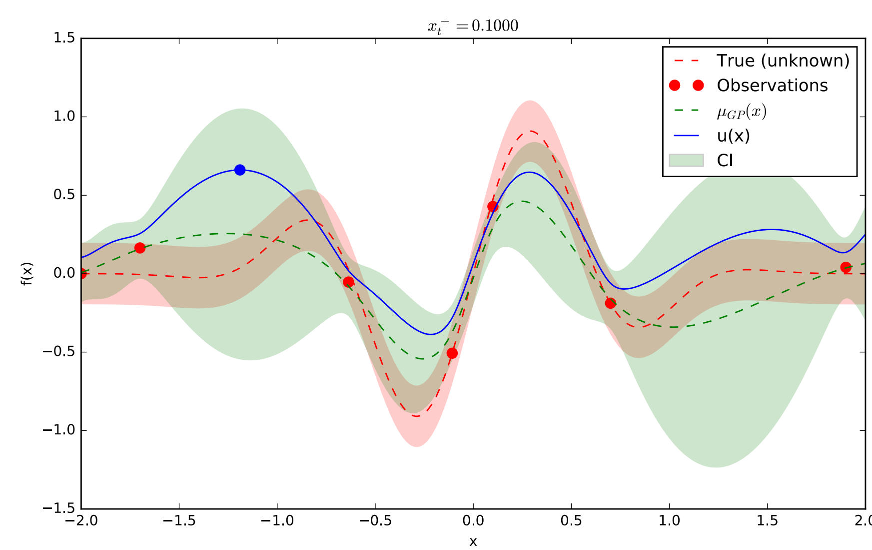
- Upper confidence bound $\text{UCB}(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$;
- Probability of improvement $\text{PI}(x) = P(f(x) \geq f(x_t^+) + \kappa)$;
- Expected improvement $\text{EI}(x) = \mathbb{E}[f(x) - f(x_t^+)]$;
- ... and many others.

where x_t^+ is the best point observed so far.

In most cases, acquisition functions provide knobs (e.g., κ) for controlling the exploration-exploitation trade-off.

- Search in regions where $\mu_{GP}(x)$ is high (exploitation)
- Probe regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

... and repeat until convergence ($t = 3$)



BAYES OPT IN A NUTSHELL

[slides from Gilles Loupe]

Bayesian optimisation

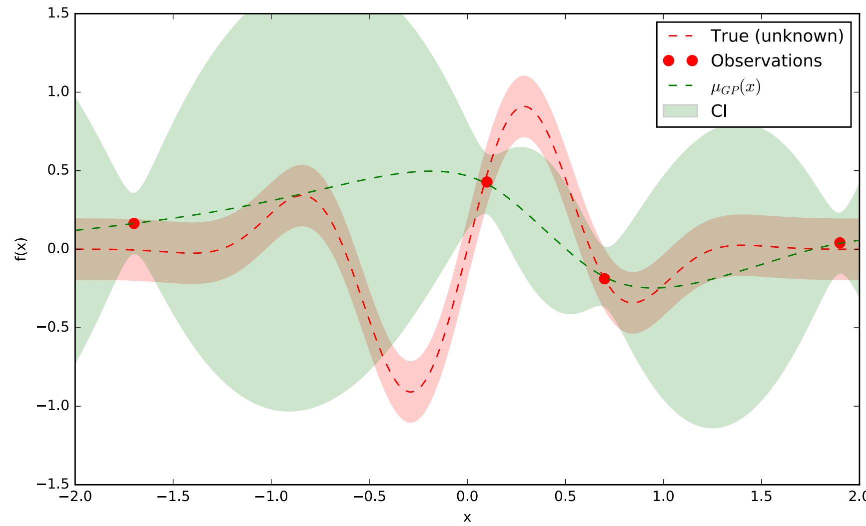
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

   $$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

Acquisition functions

Acquisition functions u(x) specify which sample x should be tried next:

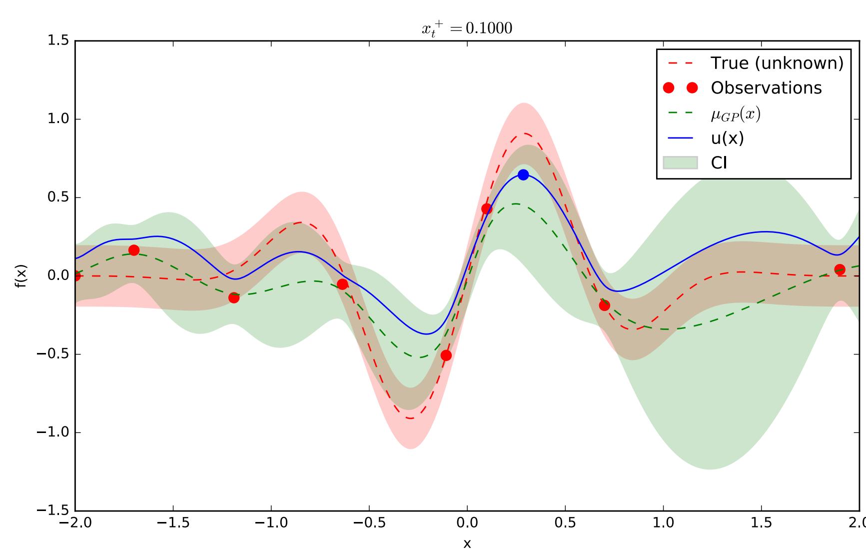
- Upper confidence bound $\text{UCB}(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$;
- Probability of improvement $\text{PI}(x) = P(f(x) \geq f(x_t^+) + \kappa)$;
- Expected improvement $\text{EI}(x) = \mathbb{E}[f(x) - f(x_t^+)]$;
- ... and many others.

where x_t^+ is the best point observed so far.

In most cases, acquisition functions provide knobs (e.g., κ) for controlling the exploration-exploitation trade-off.

- Search in regions where $\mu_{GP}(x)$ is high (exploitation)
- Probe regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

... and repeat until convergence ($t = 4$)



BAYES OPT IN A NUTSHELL

[slides from Gilles Louppe]

Bayesian optimisation

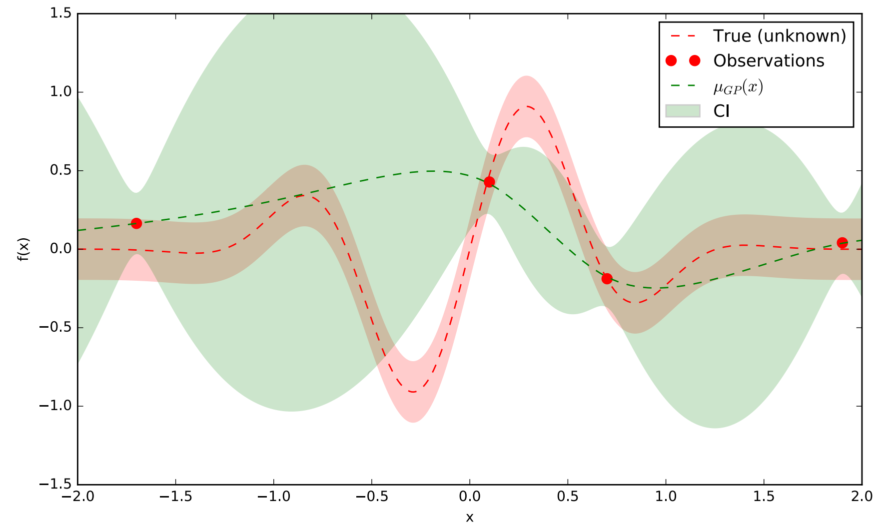
for $t = 1 : T$,

1. Given observations (x_i, y_i) for $i = 1 : t$, build a probabilistic model for the objective f.
   - Integrate out all possible true functions, using Gaussian process regression.
2. Optimise a cheap utility function u based on the posterior distribution for sampling the next point.

$$x_{t+1} = \arg\max_x u(x)$$

   Exploit uncertainty to balance exploration against exploitation.
3. Sample the next observation y_{t+1} at x_{t+1}.

Build a probabilistic model for the objective function



This gives a posterior distribution over functions that could have generated the observed data.

Acquisition functions

Acquisition functions u(x) specify which sample x should be tried next:

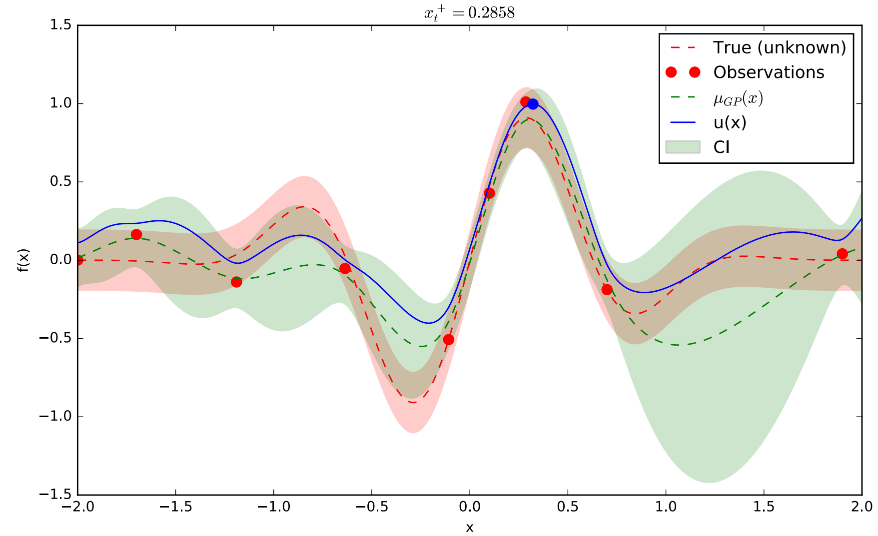
- Upper confidence bound $\text{UCB}(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$;
- Probability of improvement $\text{PI}(x) = P(f(x) \geq f(x_t^+) + \kappa)$;
- Expected improvement $\text{EI}(x) = \mathbb{E}[f(x) - f(x_t^+)]$;
- ... and many others.

where x_t^+ is the best point observed so far.

In most cases, acquisition functions provide knobs (e.g., κ) for controlling the exploration-exploitation trade-off.

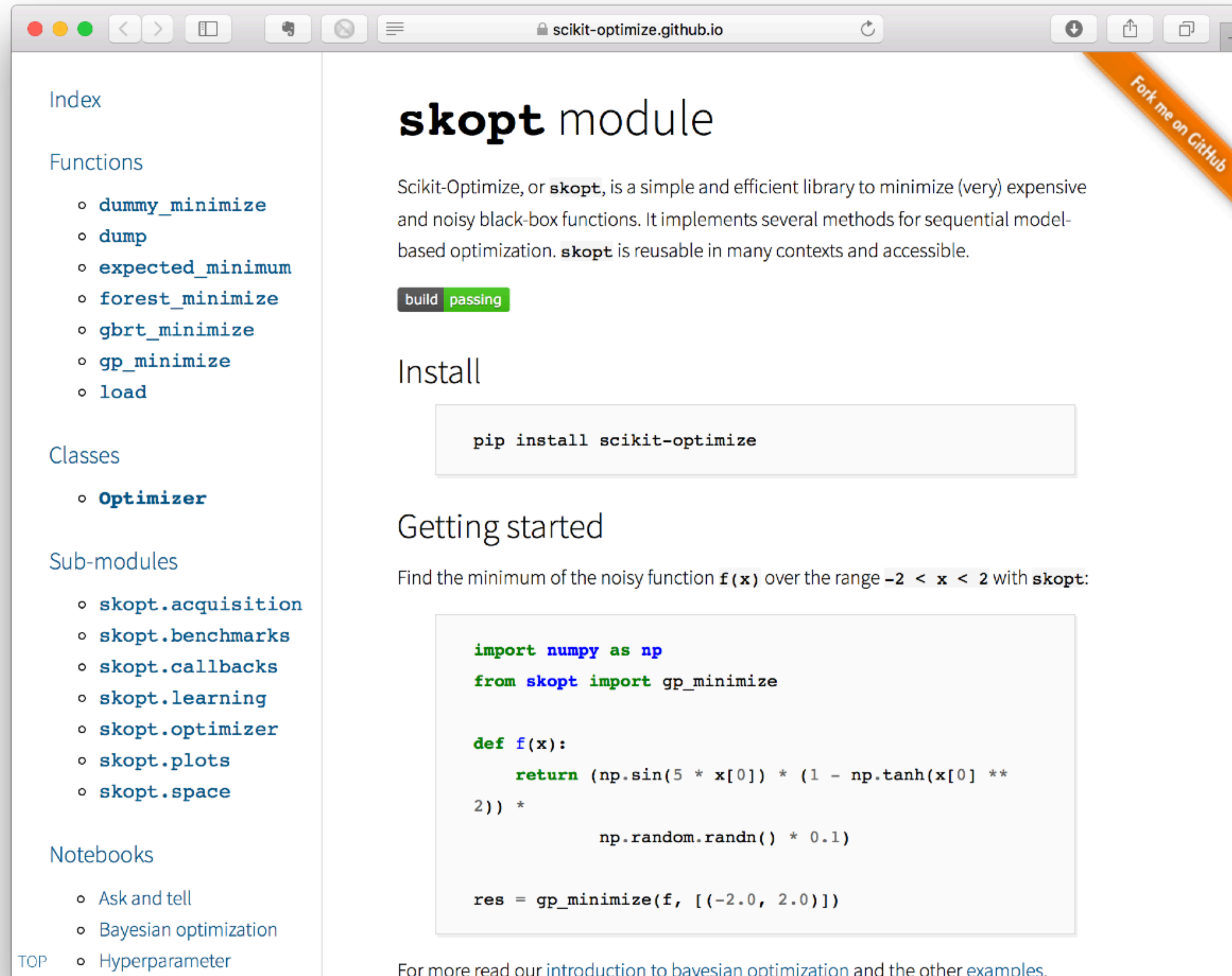
- Search in regions where $\mu_{GP}(x)$ is high (exploitation)
- Probe regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

... and repeat until convergence ($t = 5$)



SOFTWARE

- Python
  - Spearmint https://github.com/JasperSnoek/spearmint
  - GPyOpt https://github.com/SheffieldML/GPyOpt
  - RoBO https://github.com/automl/RoBO
  - scikit-optimize https://github.com/MechCoder/scikit-optimize (work in progress)
- C++
  - MOE https://github.com/yelp/MOE

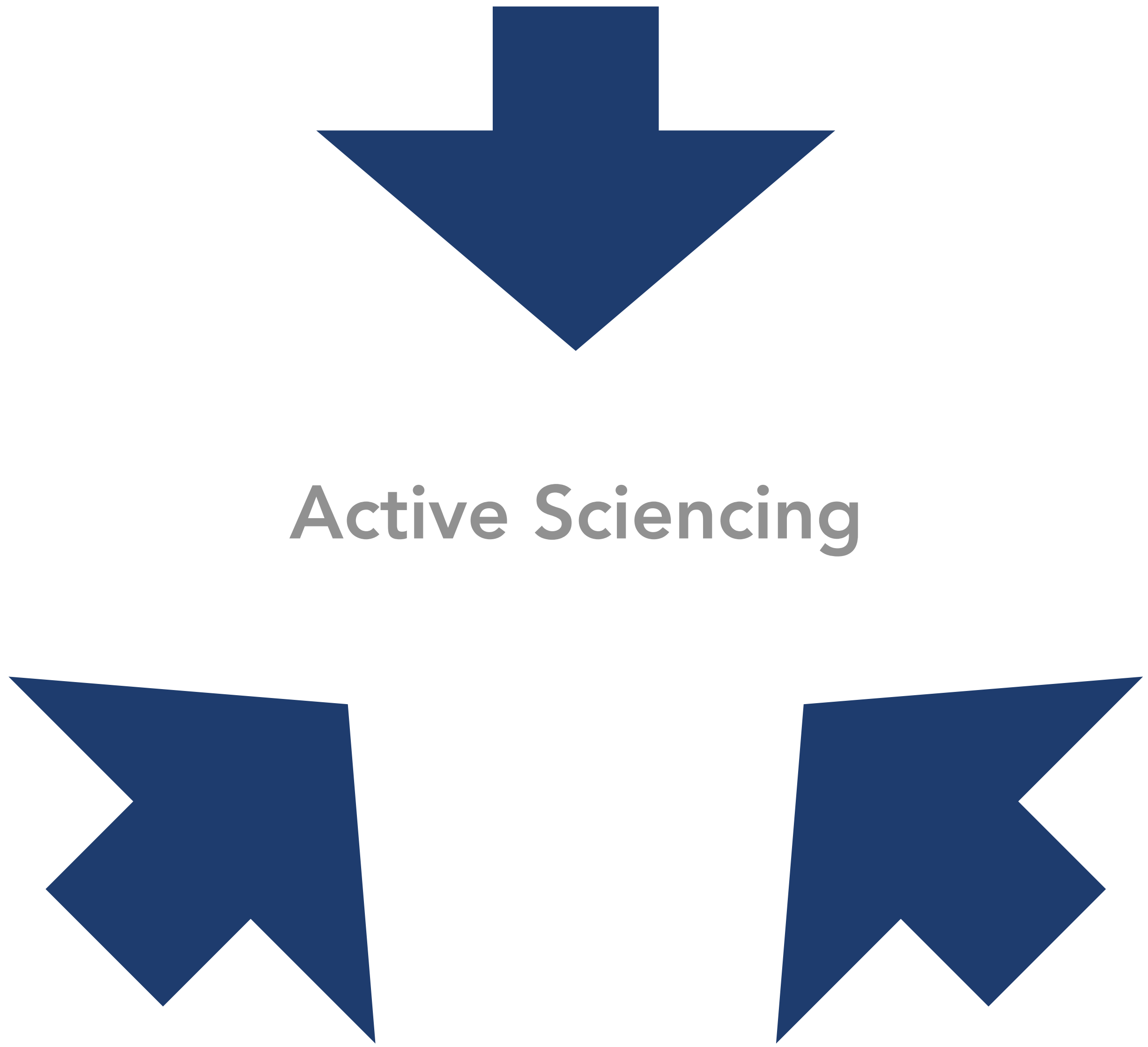


GitHub Repo for previous slides:

https://github.com/glouppe/talk-bayesian-optimisation

Putting it all together

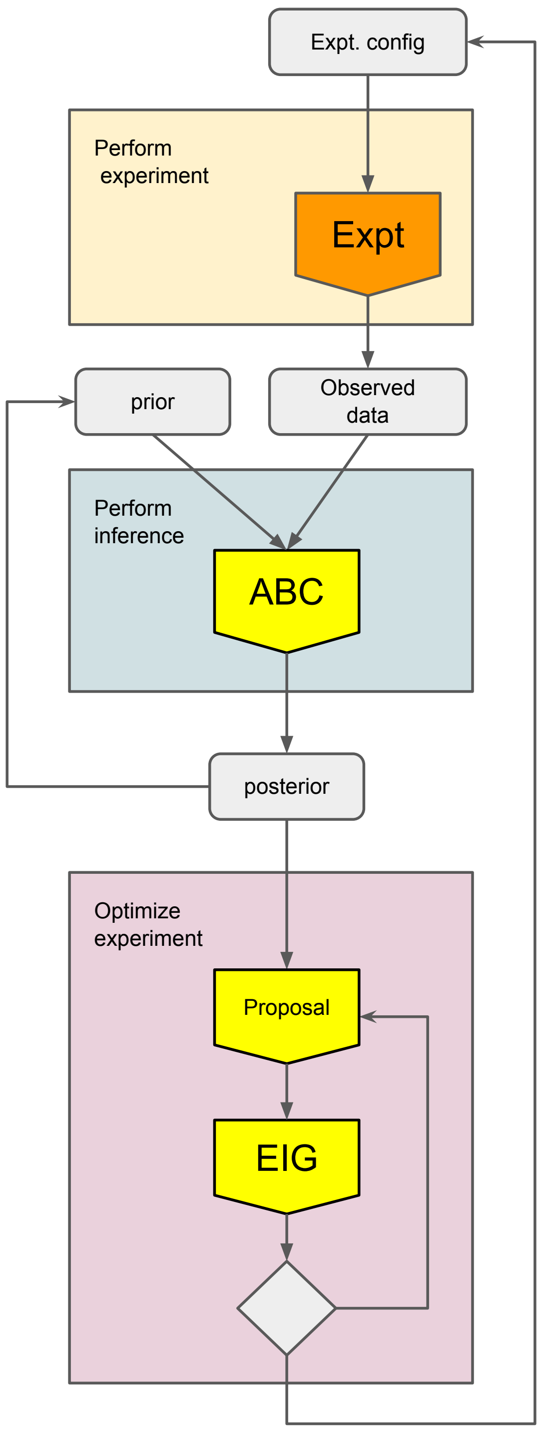
https://github.com/cranmer/active_sciencing active learning / sequential design / black box optimization

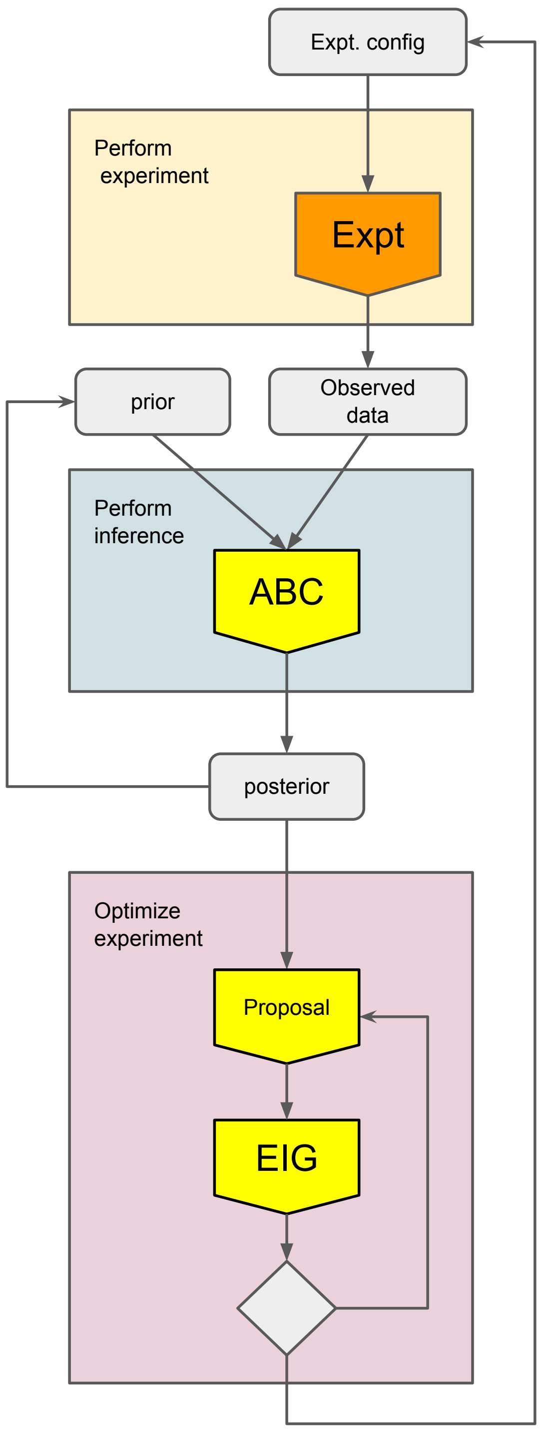


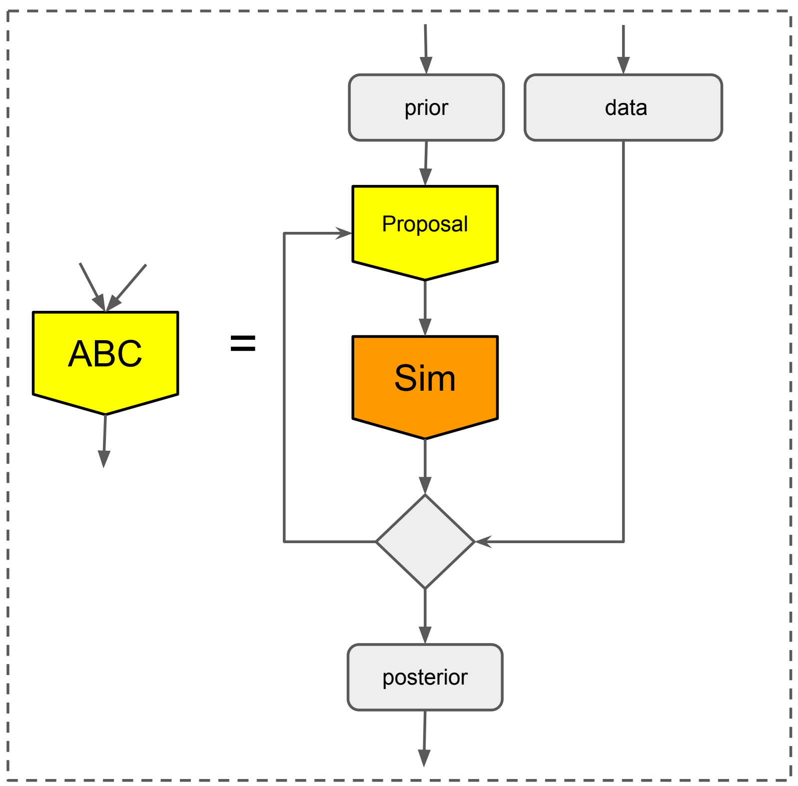
Active Sciencing

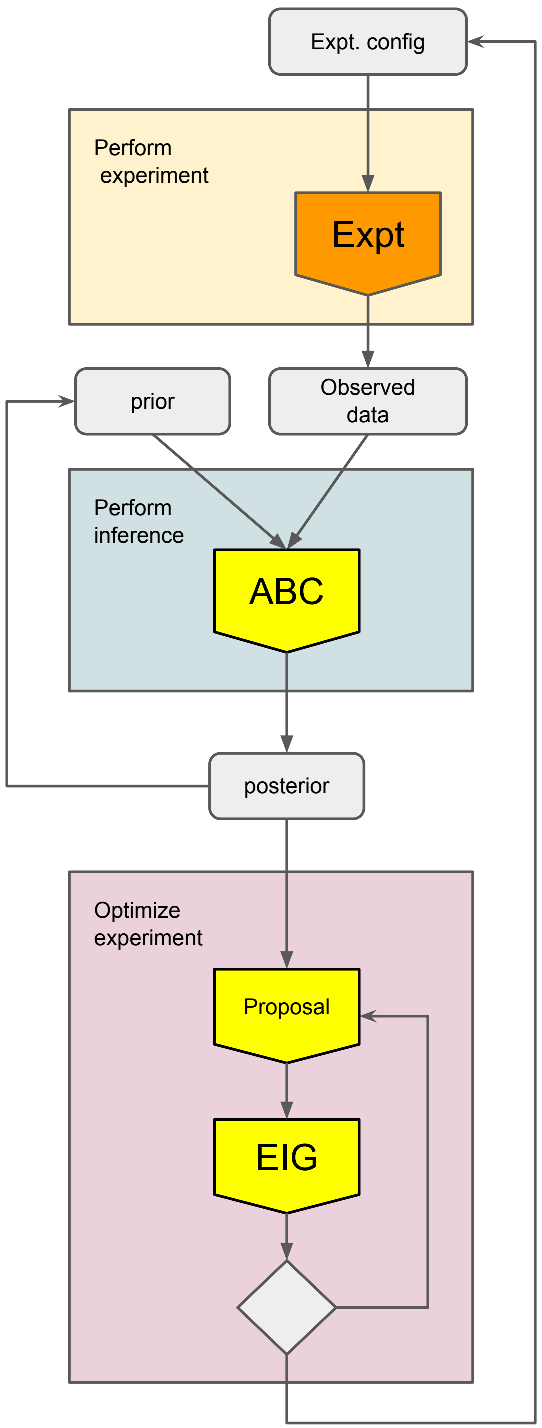
reusable workflows

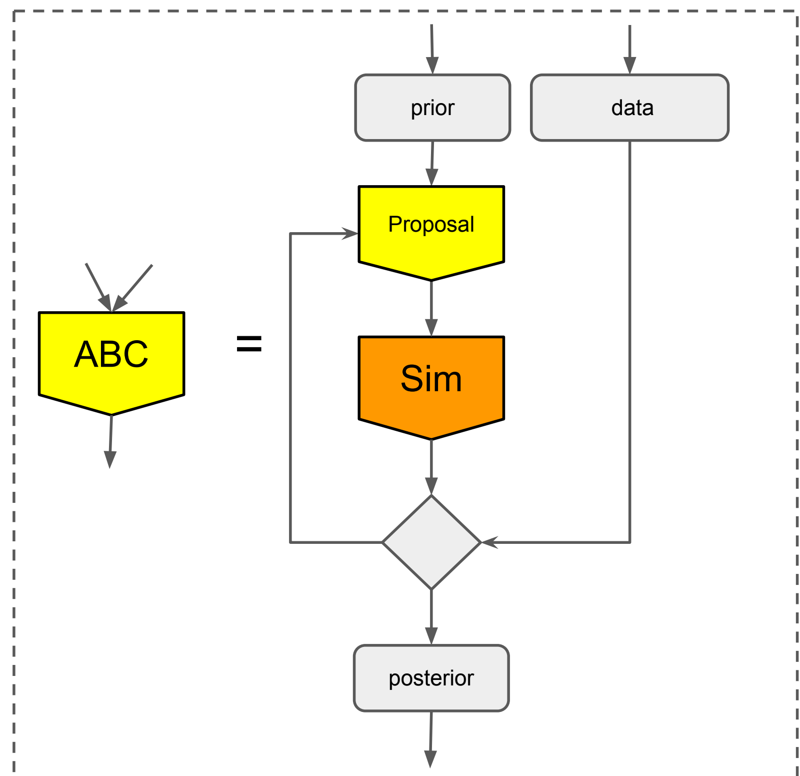
simulation-based
inference engines

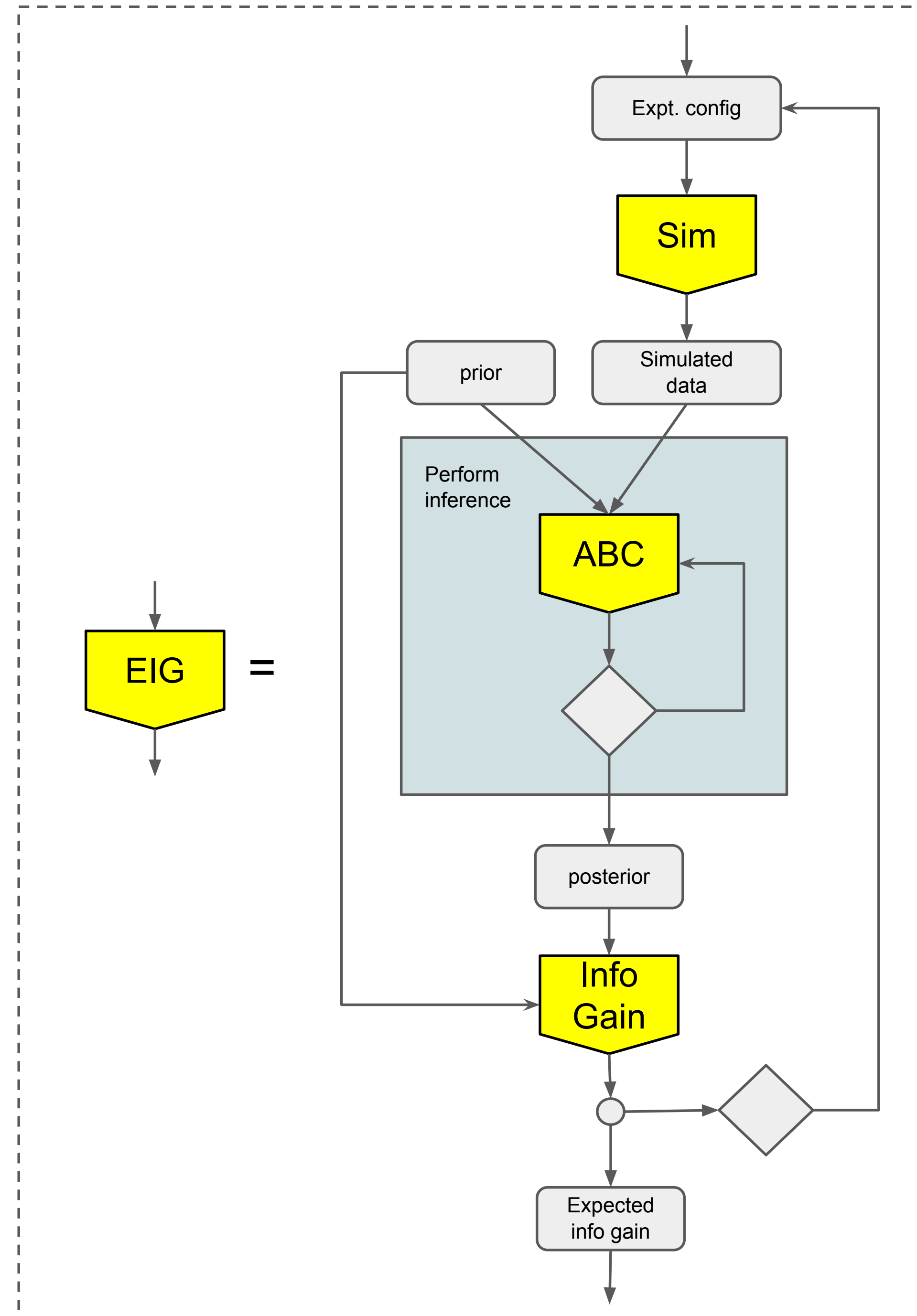

Expt. config
Perform experiment
Expt
prior
Observed data
Perform inference
ABC
posterior
Optimize experiment
Proposal
EIG


Expt. config
Perform
experiment
Expt
prior
Observed
data
Perform
inference
ABC
posterior
Optimize
experiment
Proposal
EIG


ABC
=
prior
data
Proposal
Sim
posterior


Expt. config
Perform experiment
Expt
prior
Observed data
Perform inference
ABC
posterior
Optimize experiment
Proposal
EIG


ABC
=
prior
data
Proposal
Sim
posterior

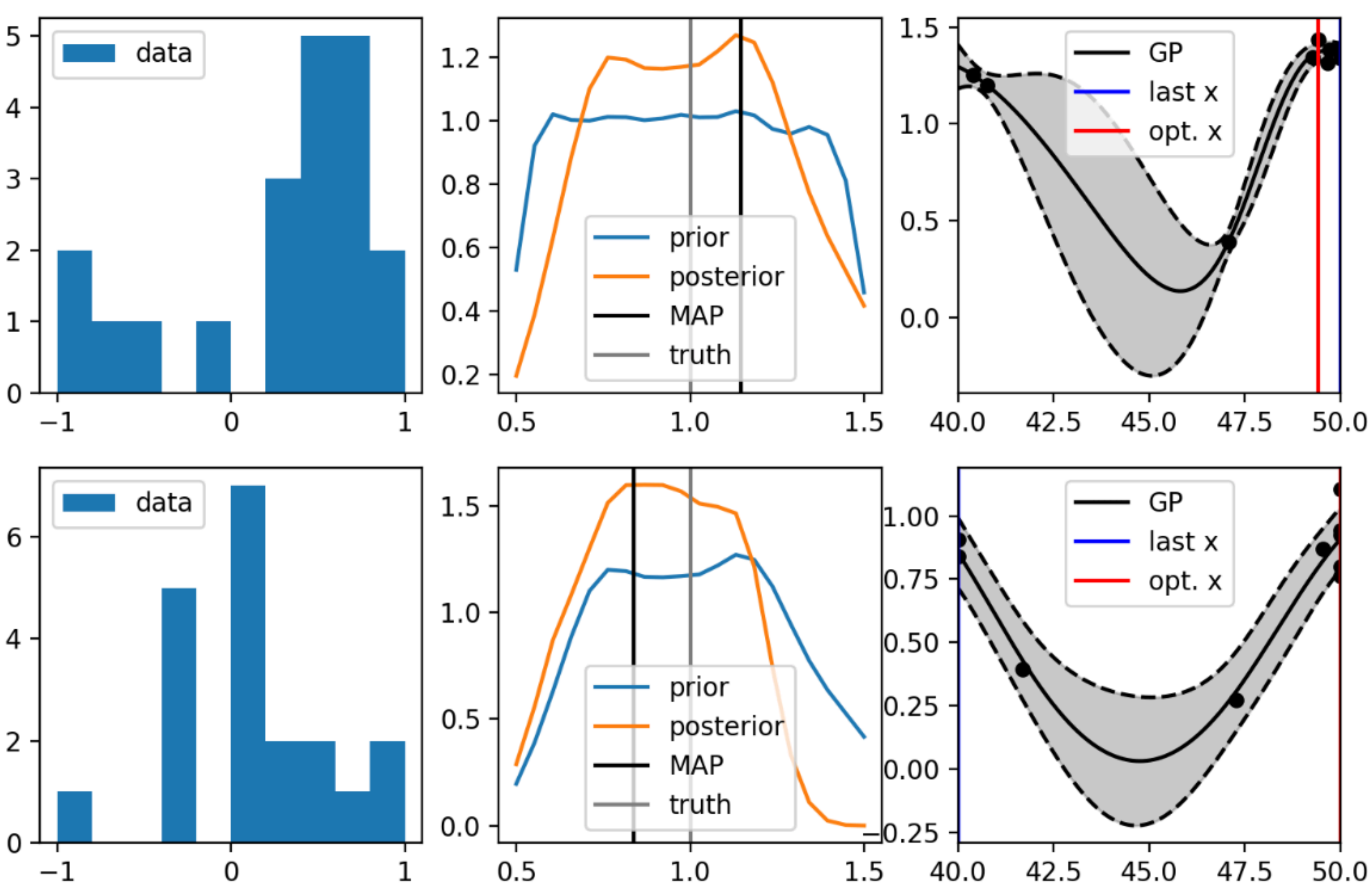

EIG
=
Expt. config
Sim
prior
Simulated data
Perform inference
ABC
posterior
Info Gain
Expected info gain

ACTIVE SCIENCING DEMO

Input:

- workflow for performing "real" experiment that returns data
- workflow for running simulator given parameters of theory and experimental configruration

Demo shows use of likelihood-free inference technique & Bayesian Optimization to measure the Weinberg angle and optimize beam energy (eg. just above or below $M_Z/2$)



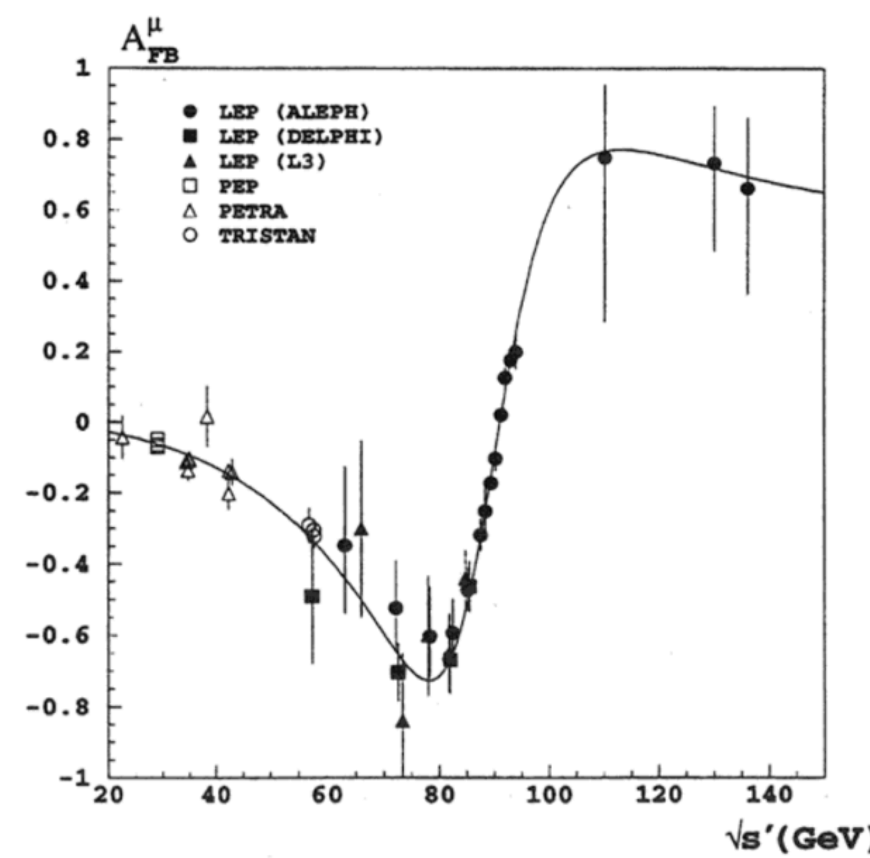


Figure 2: Measured forward-backward asymmetries of muon-pair production compared with the model independent fit results.

ENCAPSULATING THE SIMULATION

https://github.com/lukasheinrich/weinberg-test

README.md

Run HEP workflows from the web.

by Kyle Cranmer and Lukas Heinrich

An example notebook on how to generate simulated high energy physics collision events using the generator package MadGraph. Simulated datasets obtained from this notebook can then be used to train and evaluate the performance of generative models for physics.

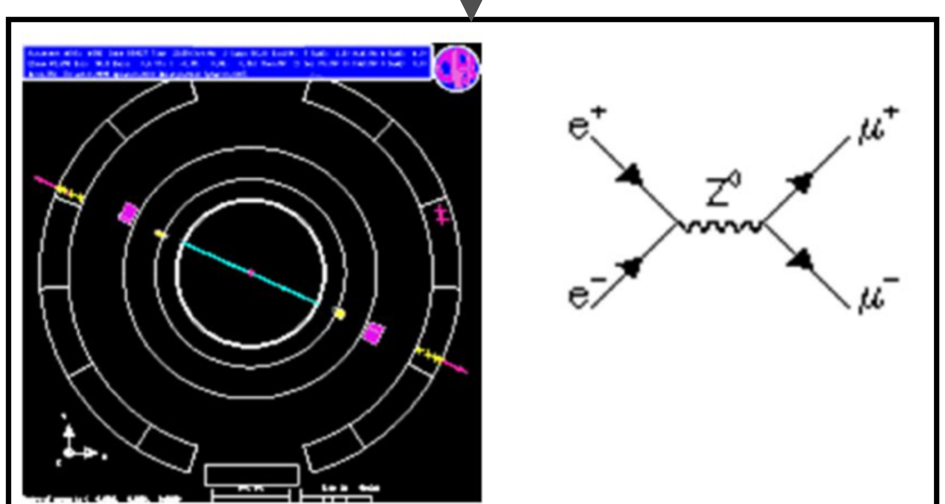
Usage:

This repository has been equipped with a Dockerfile to encapsulate its software environment. It can be used with the mybinder service to launch an ephemeral jupyter notebook server to run the notebook.

Click on the below badge and open the notebook `adage.ipynb`.

launch binder

$$
\begin{aligned}
\mathcal{L}_{SM} = & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^a_{\mu\nu}G^{\mu\nu}_a}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
& + \underbrace{\bar{L}\gamma^\mu(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)L + \bar{R}\gamma^\mu(i\partial_\mu - \frac{1}{2}g'YB_\mu)R}_{\text{kinetic energies and electroweak interactions of fermions}} \\
& + \underbrace{\frac{1}{2}\left|(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)\,\phi\right|^2 - V(\phi)}_{W^\pm, Z, \gamma \text{ and Higgs masses and couplings}} \\
& + \underbrace{g''(\bar{q}\gamma^\mu T_a q)G^a_\mu}_{\text{interactions between quarks and gluons}} + \underbrace{(G_1\bar{L}\phi R + G_2\bar{L}\phi_c R + h.c.)}_{\text{fermion masses and couplings to Higgs}}
\end{aligned}
$$



other electroweak parameters. This can be shown with Eq. (2.96), giving

$$
A^f_{FB}(s) \simeq A^f_{FB}(m_Z^2) + \frac{(s - m_Z^2)}{s}\,\frac{3\pi\alpha(s)}{\sqrt{2}G_F m_Z^2}\,\frac{2Q_e Q_f g_{Ae} g_{Af}}{(g^2_{Ve} + g^2_{Ae})(g^2_{Vf} + g^2_{Af})}\;. \quad (8.30)
$$

ENCAPSULATING THE SIMULATION

https://github.com/lukasheinrich/weinberg-test

README.md

Run HEP workflows from the web.

by Kyle Cranmer and Lukas Heinrich

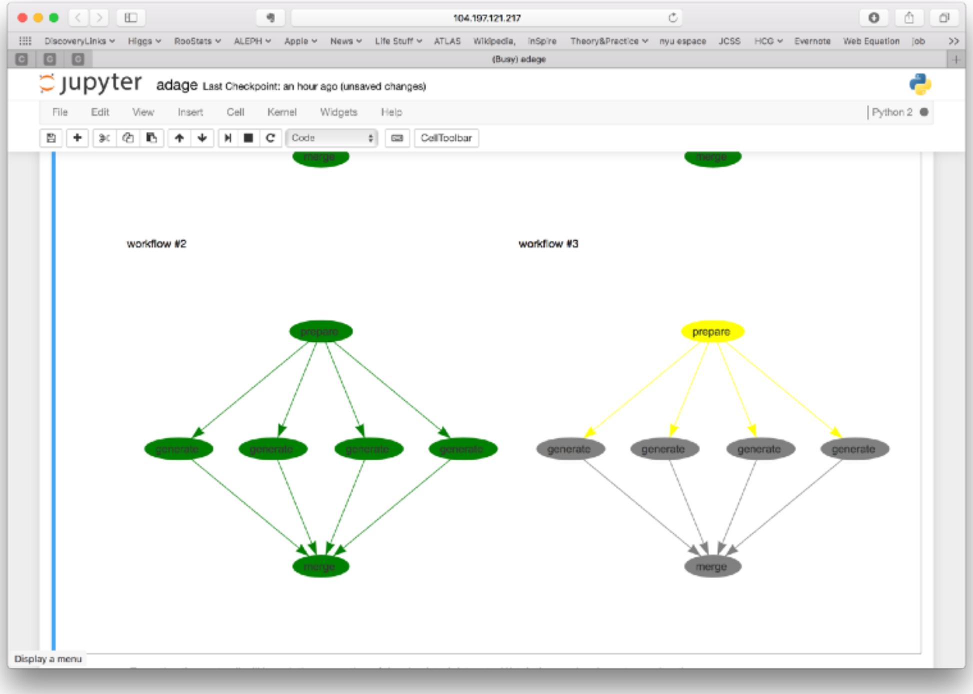
An example notebook on how to generate simulated high energy physics collision events using the generator package MadGraph. Simulated datasets obtained from this notebook can then be used to train and evaluate the performance of generative models for physics.

Usage:

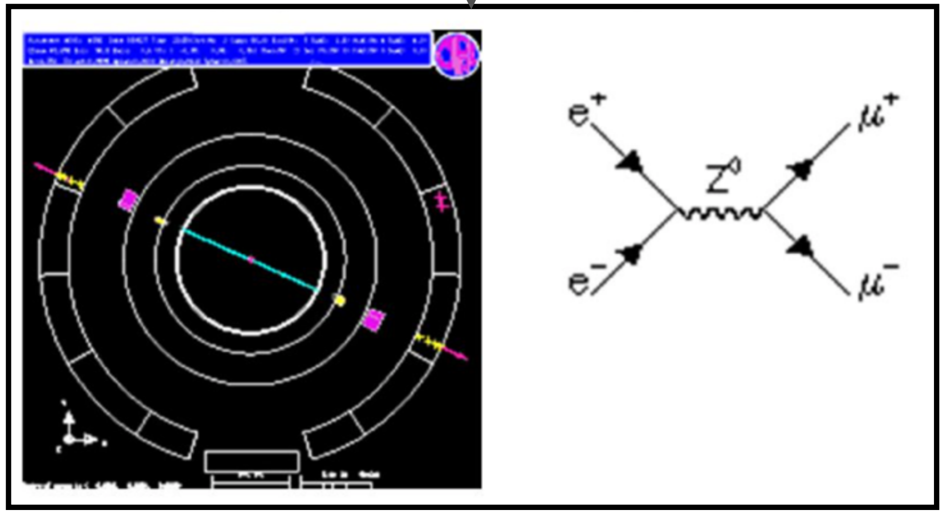
This repository has been equipped with a Dockerfile to encapsulate its software environment. It can be used with the mybinder service to launch an ephemeral jupyter notebook server to run the notebook.

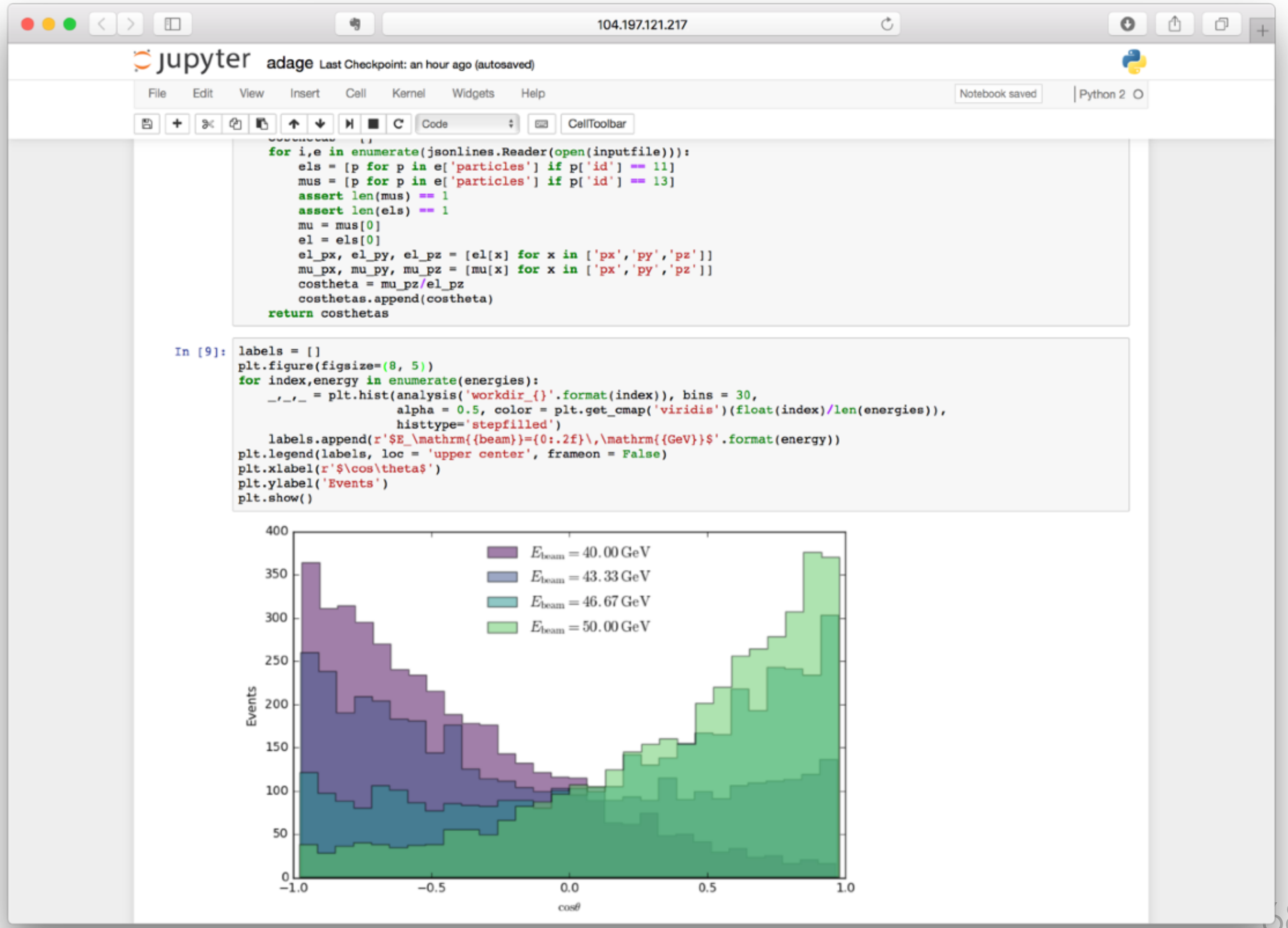
Click on the below badge and open the notebook `adage.ipynb`.

launch binder



$$
\begin{aligned}
\mathcal{L}_{SM} = & \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^a_{\mu\nu}G^{\mu\nu}_a}_{\text{kinetic energies and self-interactions of the gauge bosons}} \\
& + \underbrace{\bar{L}\gamma^\mu(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)L + \bar{R}\gamma^\mu(i\partial_\mu - \frac{1}{2}g'YB_\mu)R}_{\text{kinetic energies and electroweak interactions of fermions}} \\
& + \underbrace{\frac{1}{2}\left|(i\partial_\mu - \frac{1}{2}g\tau\cdot\mathbf{W}_\mu - \frac{1}{2}g'YB_\mu)\,\phi\right|^2 - V(\phi)}_{W^\pm, Z, \gamma, \text{ and Higgs masses and couplings}} \\
& + \underbrace{g''(\bar{q}\gamma^\mu T_a q)G^a_\mu}_{\text{interactions between quarks and gluons}} + \underbrace{(G_1\bar{L}\phi R + G_2\bar{L}\phi_c R + h.c.)}_{\text{fermion masses and couplings to Higgs}}
\end{aligned}
$$





CONCLUSIONS

(verbal)