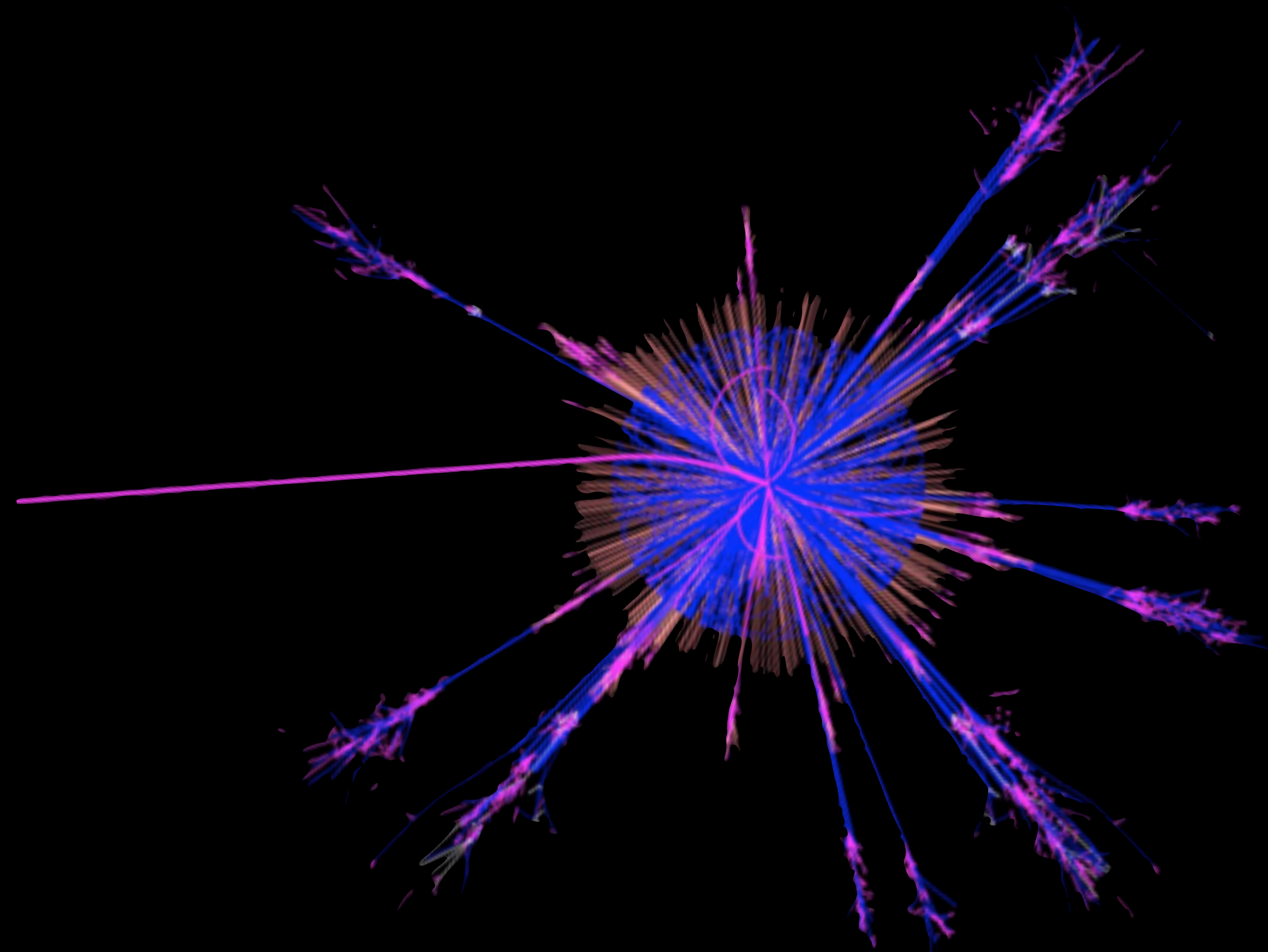
NYU CENTER FOR DATA SCIENCE

CENTER FOR COSMOLOGY AND PARTICLE PHYSICS

NEW APPROACHES TO

# LIKELIHOOD FREE INFERENCE

http://arxiv.org/abs/1506.02169



**Kyle Cranmer**
New York University
Department of Physics
Center for Data Science

# PREFACE

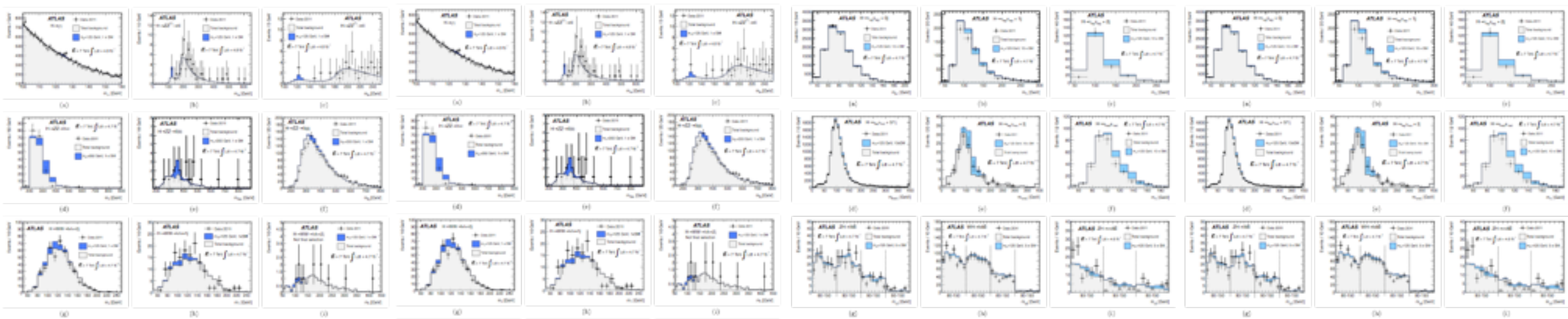
This reminds me of PhyStat series leading up to the LHC.

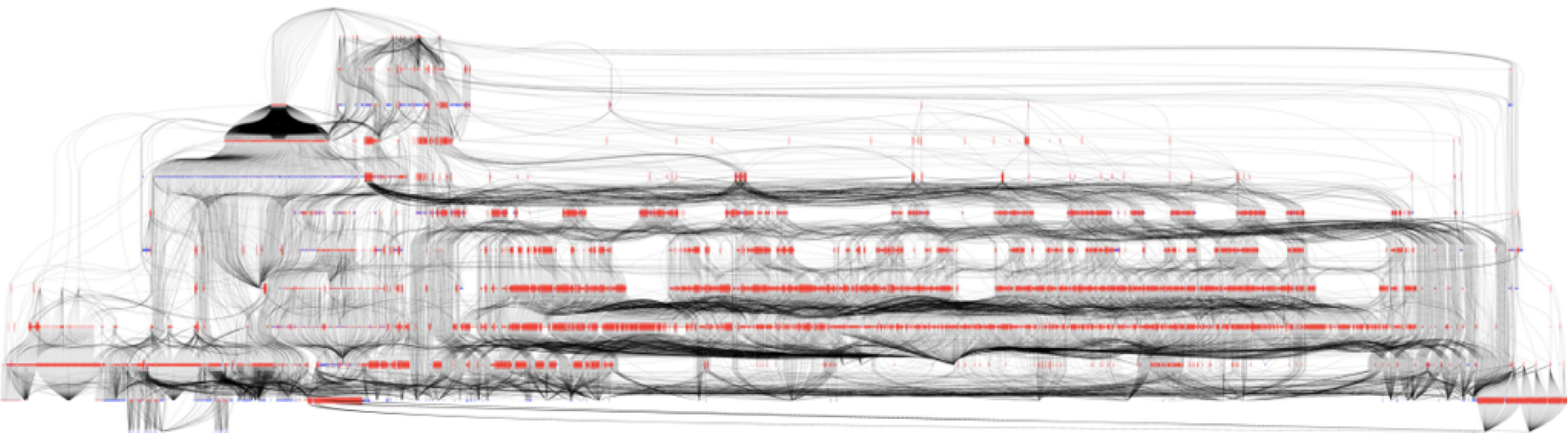
- Thanks to Louis, Tom, Bob, Richard, …
- Impressed by the sophistication of discussion

One thing I learned:

- collaboration might converge on high-level statistical procedure. Put in likelihood / probability model and turn the crank.
- Practical improvements to analysis mainly lie in techniques used for modeling the data ! (eg. systematics, ND->FD extrapolation, etc.)
- Useful to factorize discussion & software in terms of **modeling** and high-level statistical **procedure**

# THE HIGGS DISCOVERY





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

# INTRODUCTION

In particle physics, our high-level **inference goals** are

- searches (hypothesis testing)
- measurements (maximum likelihood estimate)
- constrain parameters (confidence intervals)

Typically, we use likelihood-based techniques

- surprisingly, we lack a nice technique for likelihood-based inference when we want to use high-dimensional observations and have to deal with detector response

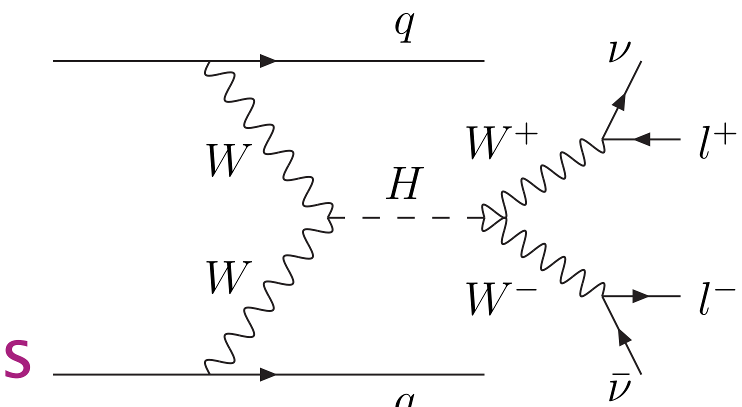
# Likelihood-free Inference

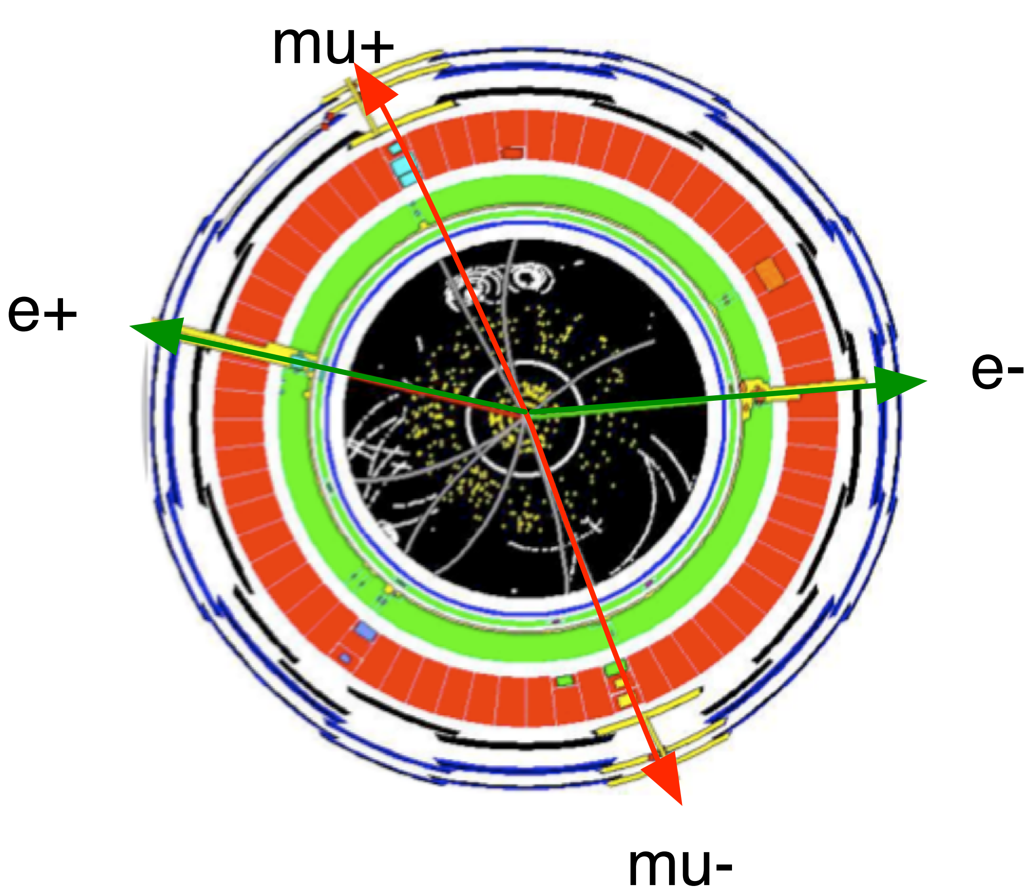
# OVERVIEW OF PREDICTIONS

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

1) The language is Quantum Field Theory

2) Feynman Diagrams are used to predict high-energy interaction among fundamental particles




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

>100 million sensors

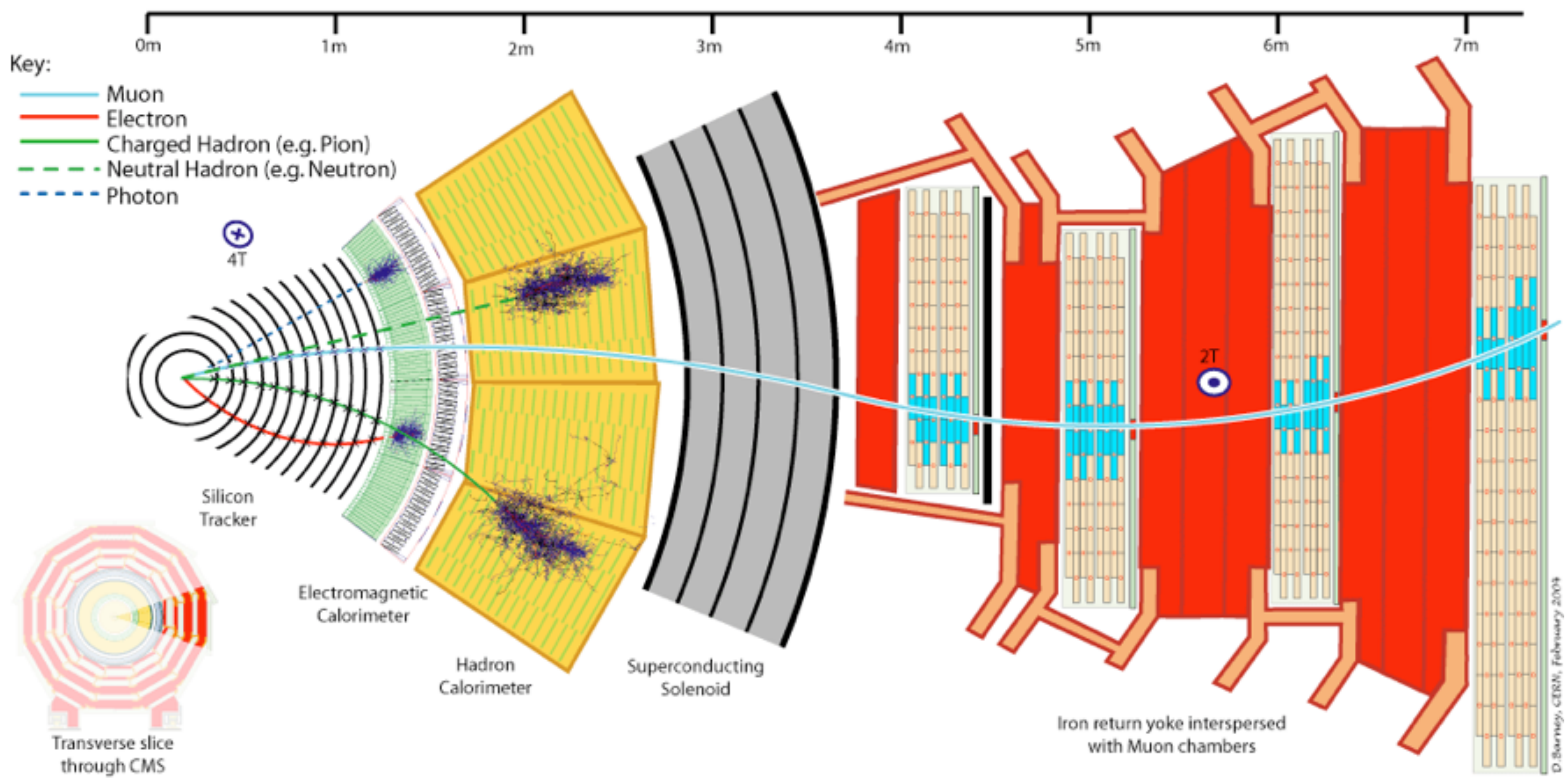
4) Finally, we run particle identification algorithms on the simulated data as if they were from real collisions.

~10-30 features describe interesting part

**Conceptually:** Prob(detector response | particles )

**Implementation:** Monte Carlo integration over micro-physics

**Consequence:** cannot evaluate likelihood for a given event



# DETECTOR SIMULATION

**Conceptually:** Prob(detector response | particles )

**Implementation:** Monte Carlo integration over micro-physics

**Consequence:** cannot evaluate likelihood for a given event

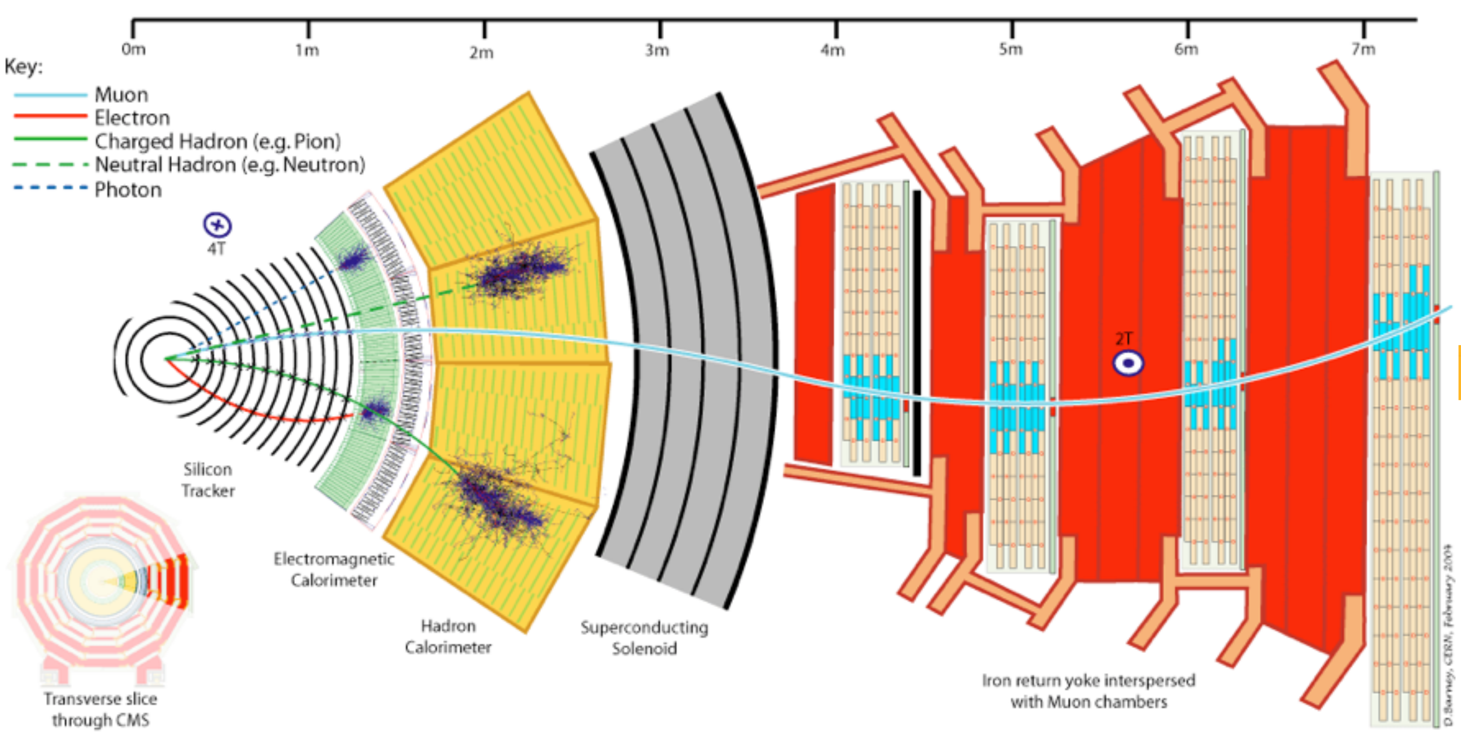
This motivates a new class of algorithms for what is called **likelihood-free inference**, which only require ability to generate samples from the simulation in the "forward mode"

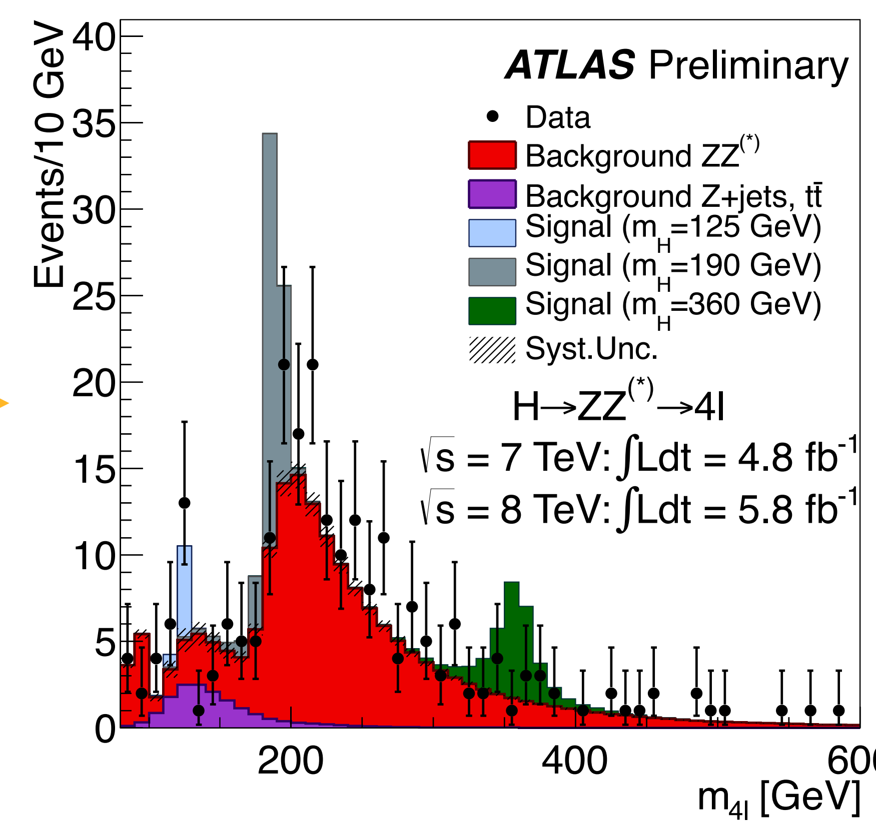
# $10^8$ SENSORS → 1 REAL-VALUED QUANTITY

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|θ) **approximated** using histograms (univariate density estimation)



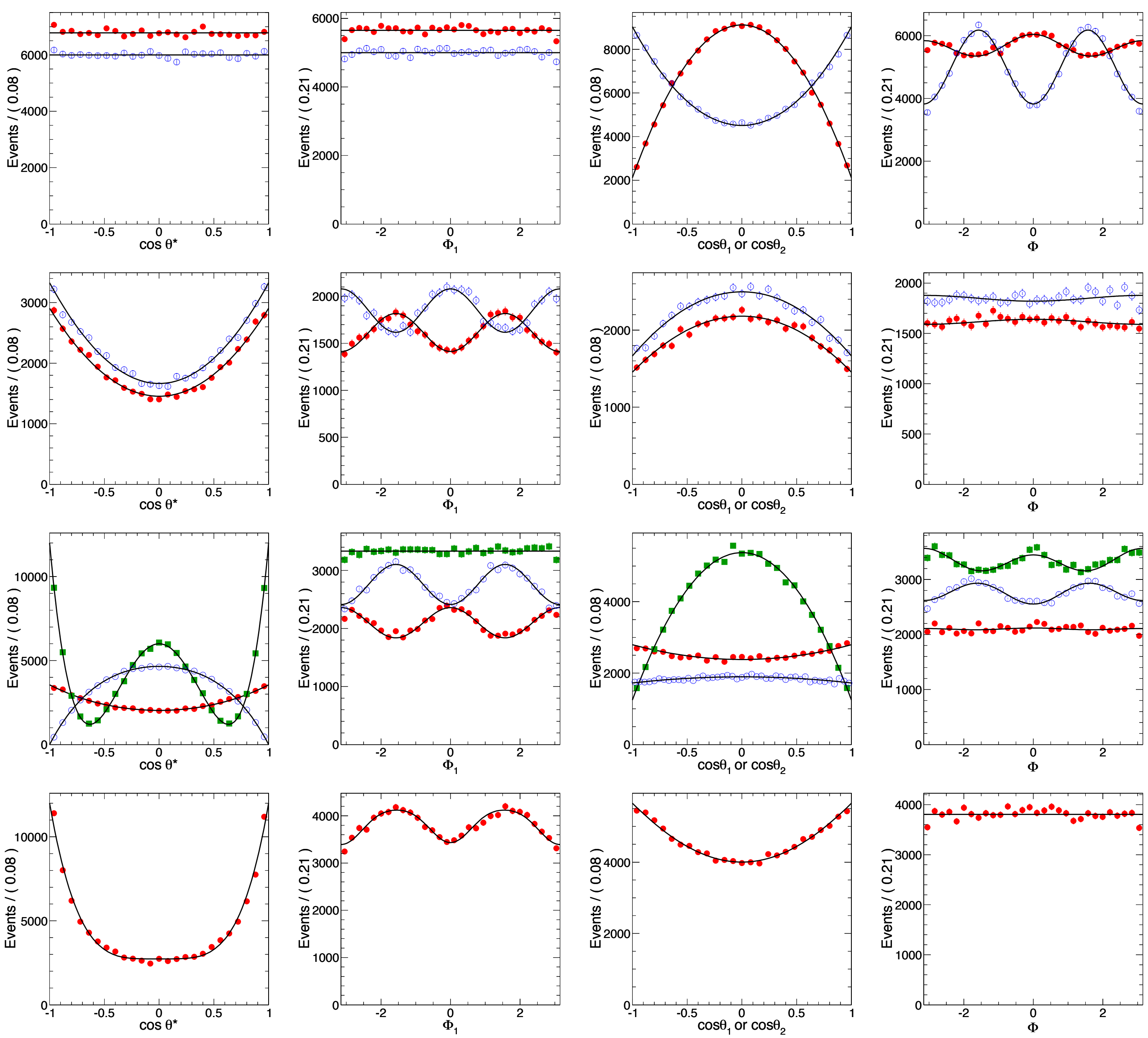



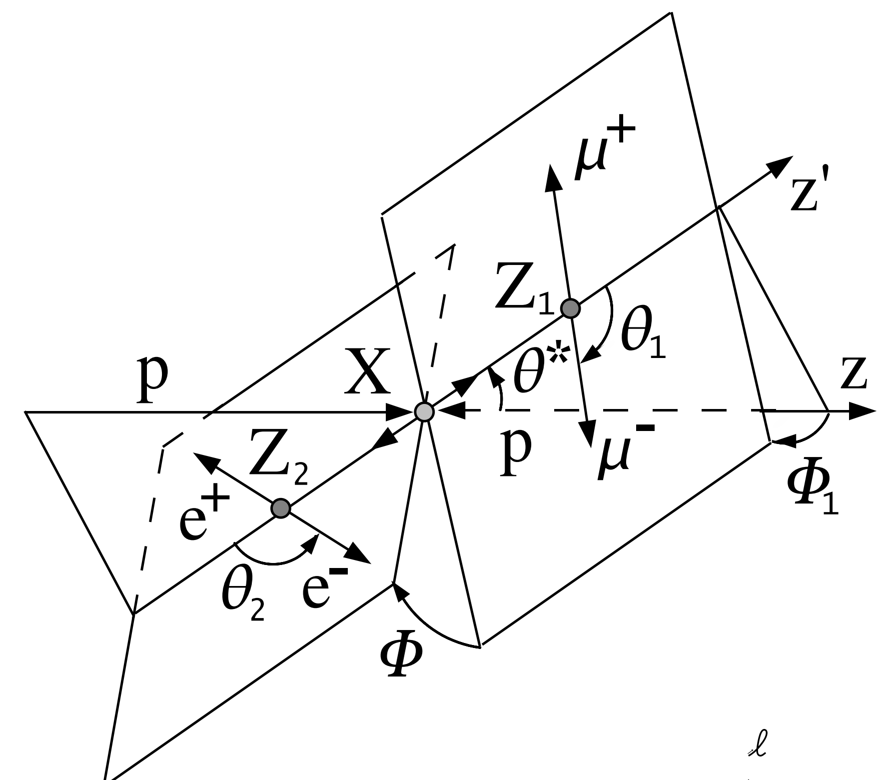

**This doesn't scale if x is high dimensional!**

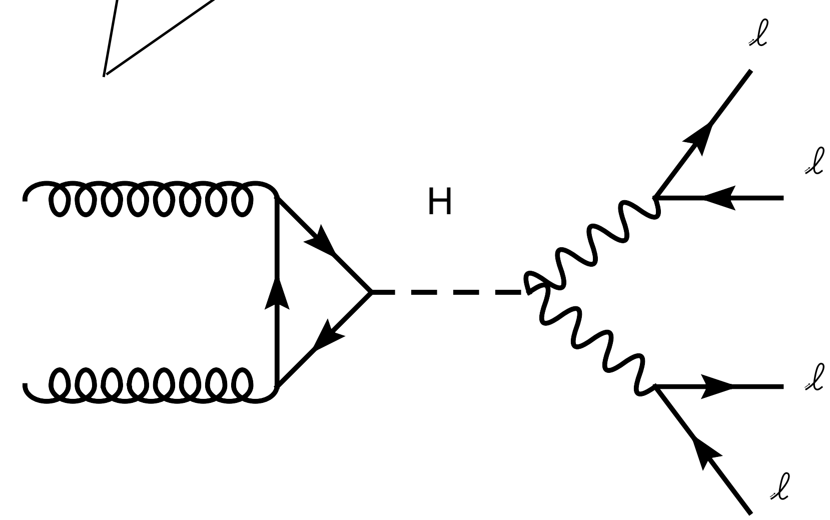
# HIGH DIMENSIONAL EXAMPLE

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

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

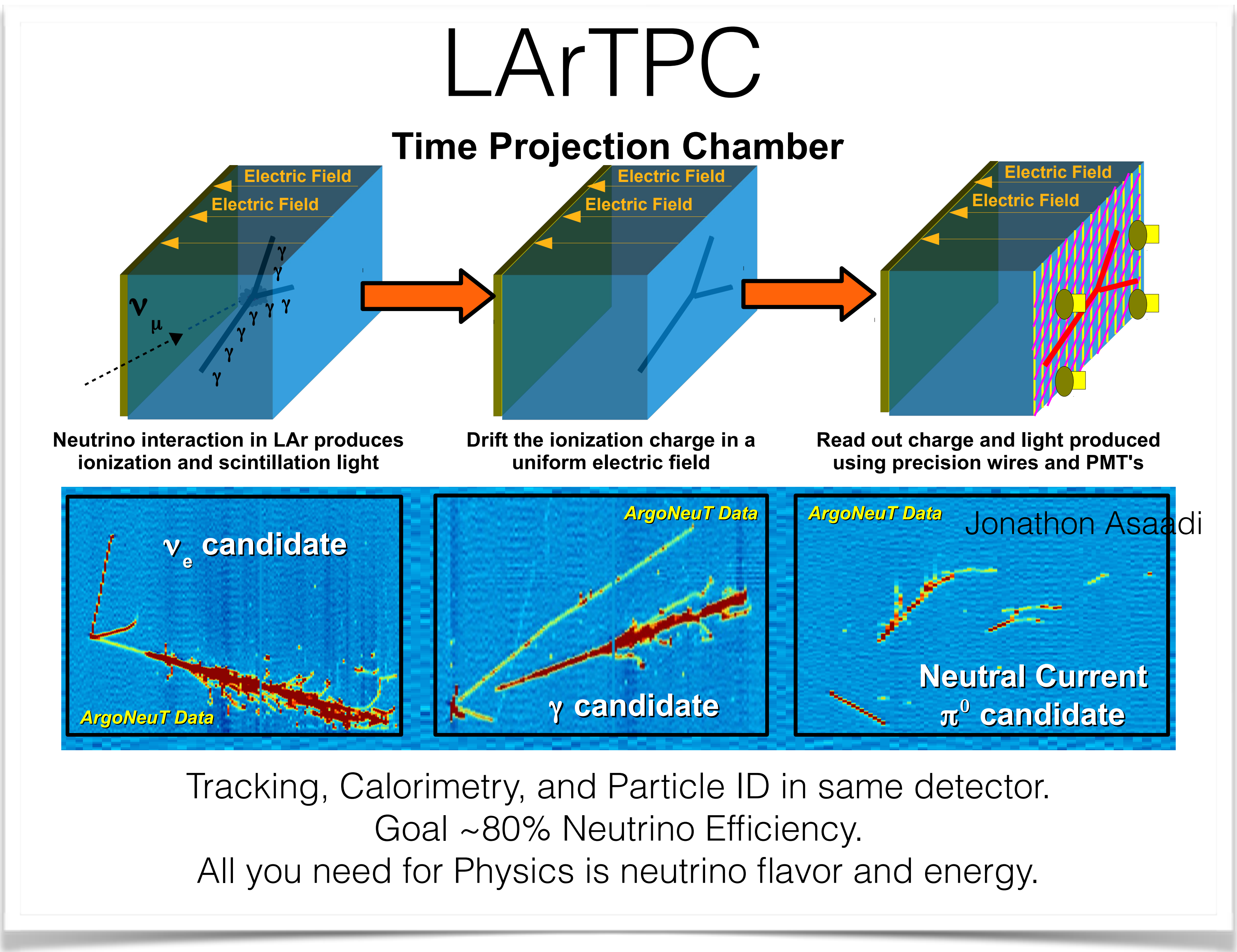


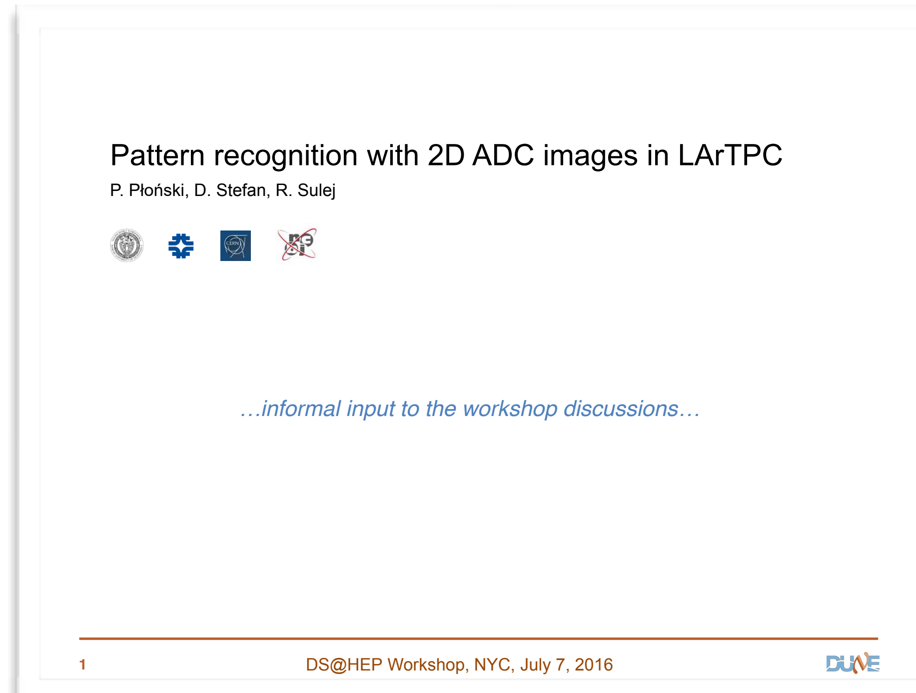



## A more extreme example is to work with lower-level data

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

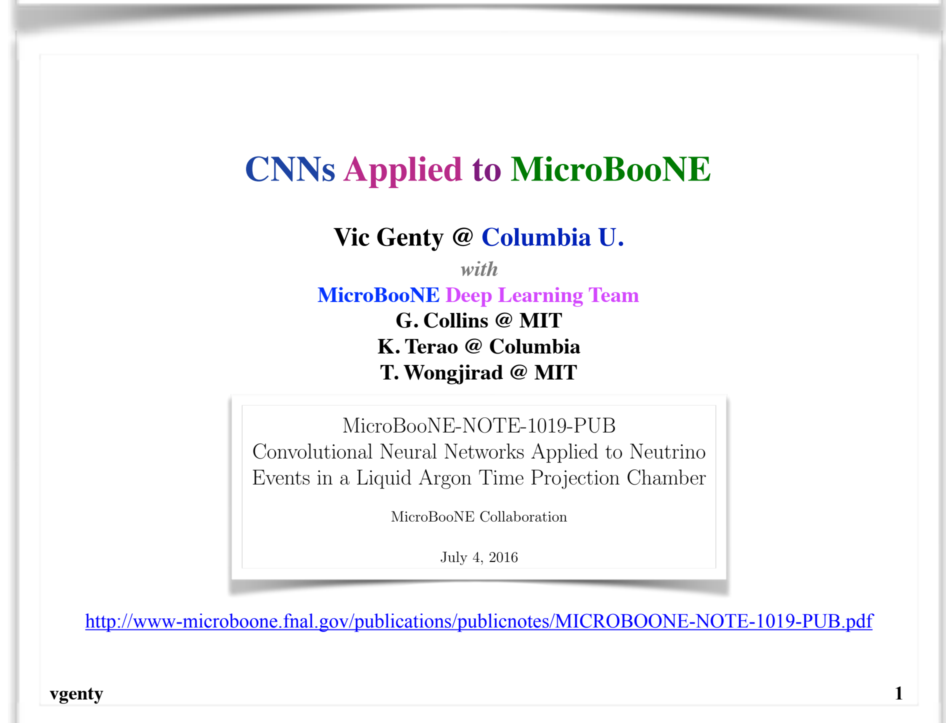




Pattern recognition with 2D ADC images in LArTPC

P. Płoński, D. Stefan, R. Sulej

…informal input to the workshop discussions…

1 DS@HEP Workshop, NYC, July 7, 2016


CNNs Applied to MicroBooNE

Vic Genty @ Columbia U.

*with*

MicroBooNE Deep Learning Team

G. Collins @ MIT

K. Terao @ Columbia

T. Wongjirad @ MIT

MicroBooNE-NOTE-1019-PUB

Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber

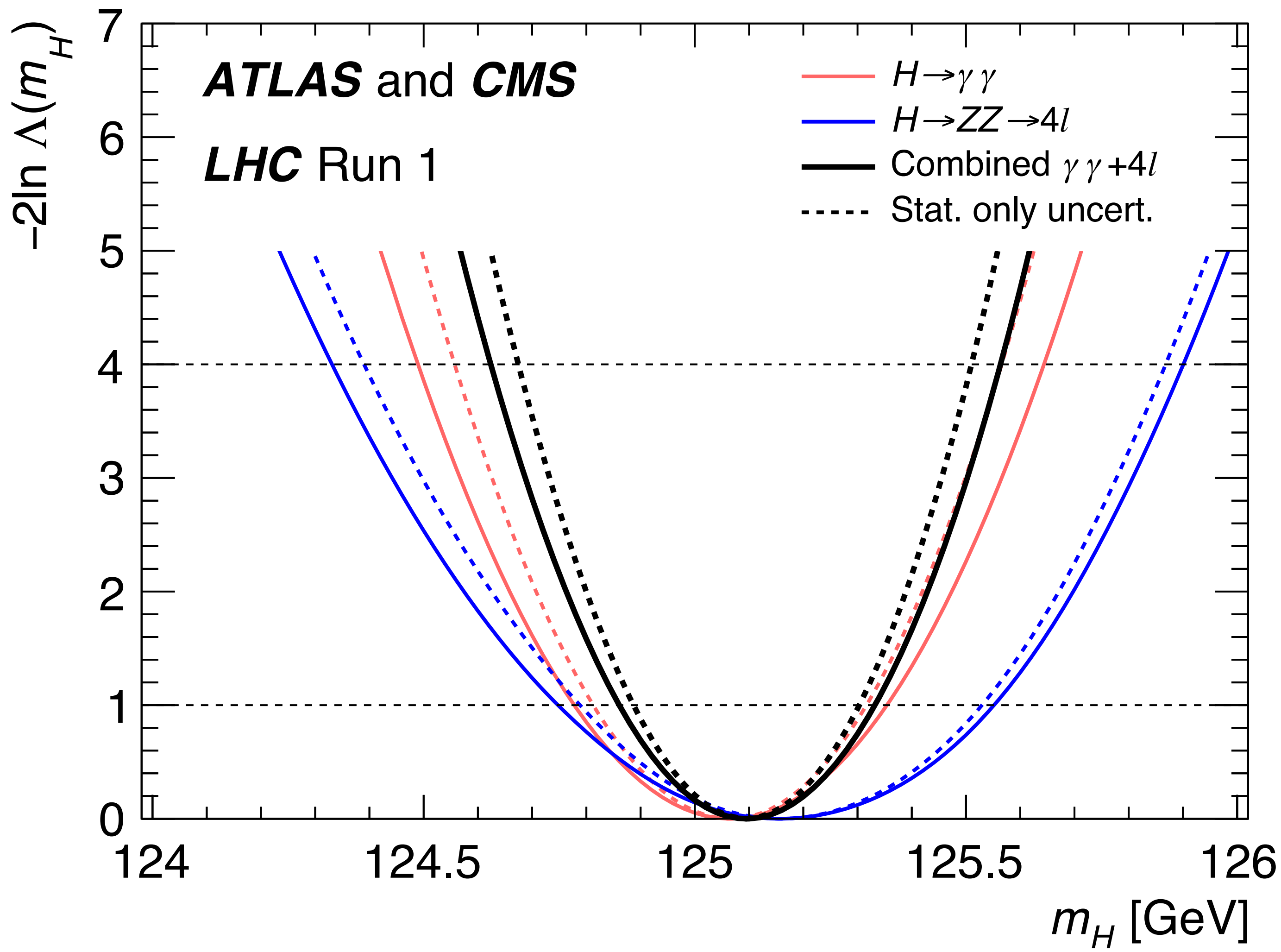
MicroBooNE Collaboration

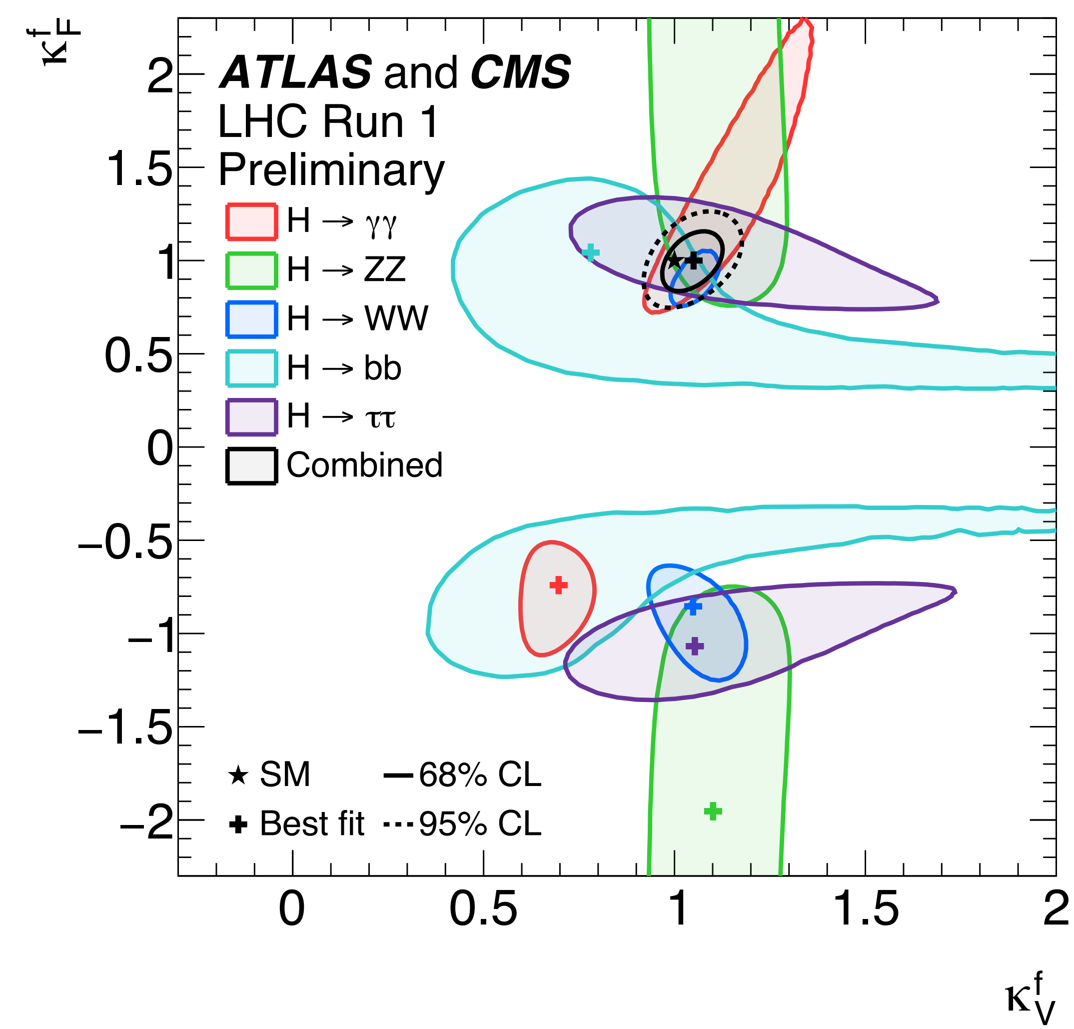
July 4, 2016

http://www-microboone.fnal.gov/publications/publicnotes/MICROBOONE-NOTE-1019-PUB.pdf

vgenty 1

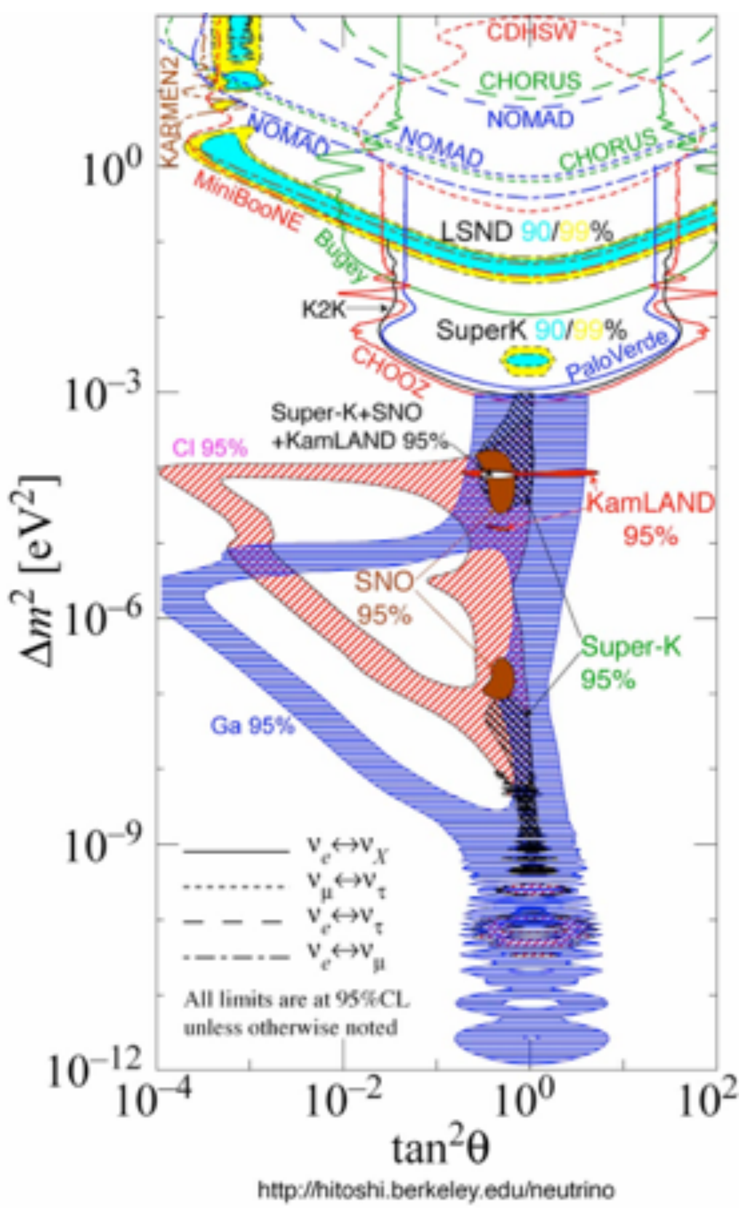
**Goal**: approximate the likelihood p(x|θ) for high dimensional feature x using a generative model for the data

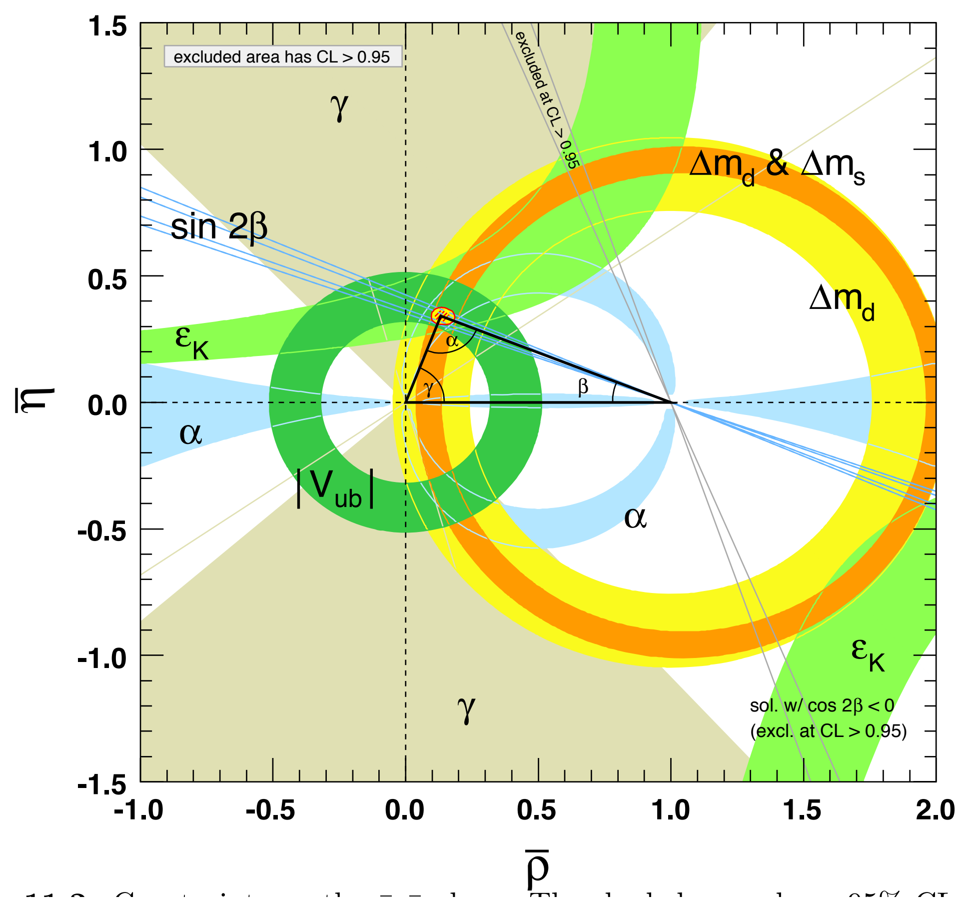


# LIKELIHOOD FREE INFERENCE

**Goal**: approximate the likelihood p(x|θ) for high dimensional feature x using a generative model for the data



# THE RAPID RISE OF "ABC"

**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

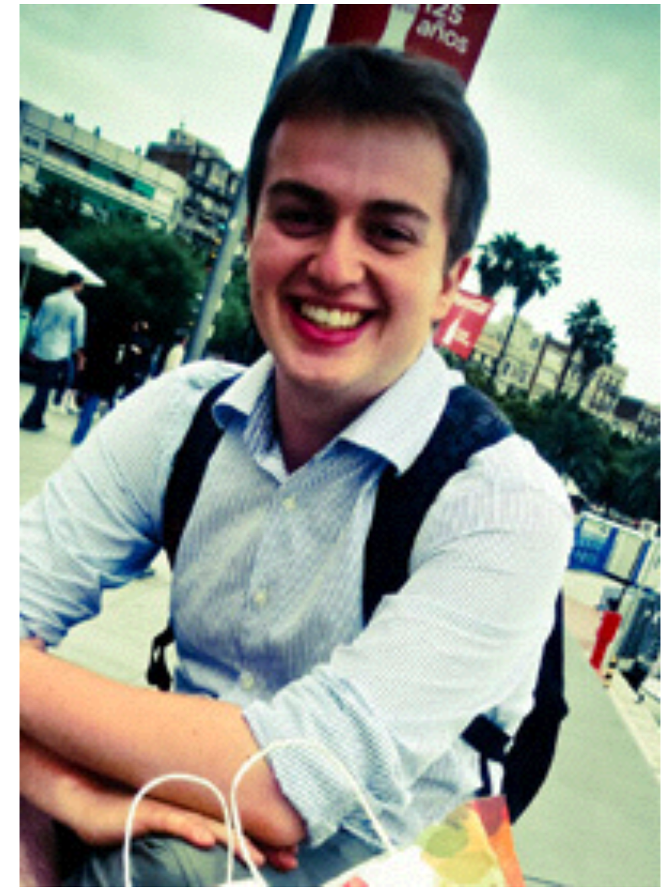
Approximate Bayesian computation (ABC) or likelihood-free (LF) methods have developed mostly beyond the radar of the machine learning community, but are important tools for a large and diverse segment of the scientific community. This is particularly true for systems and population biology, computational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

# AN ALTERNATIVE TO ABC

K.C., http://arxiv.org/abs/1506.02169

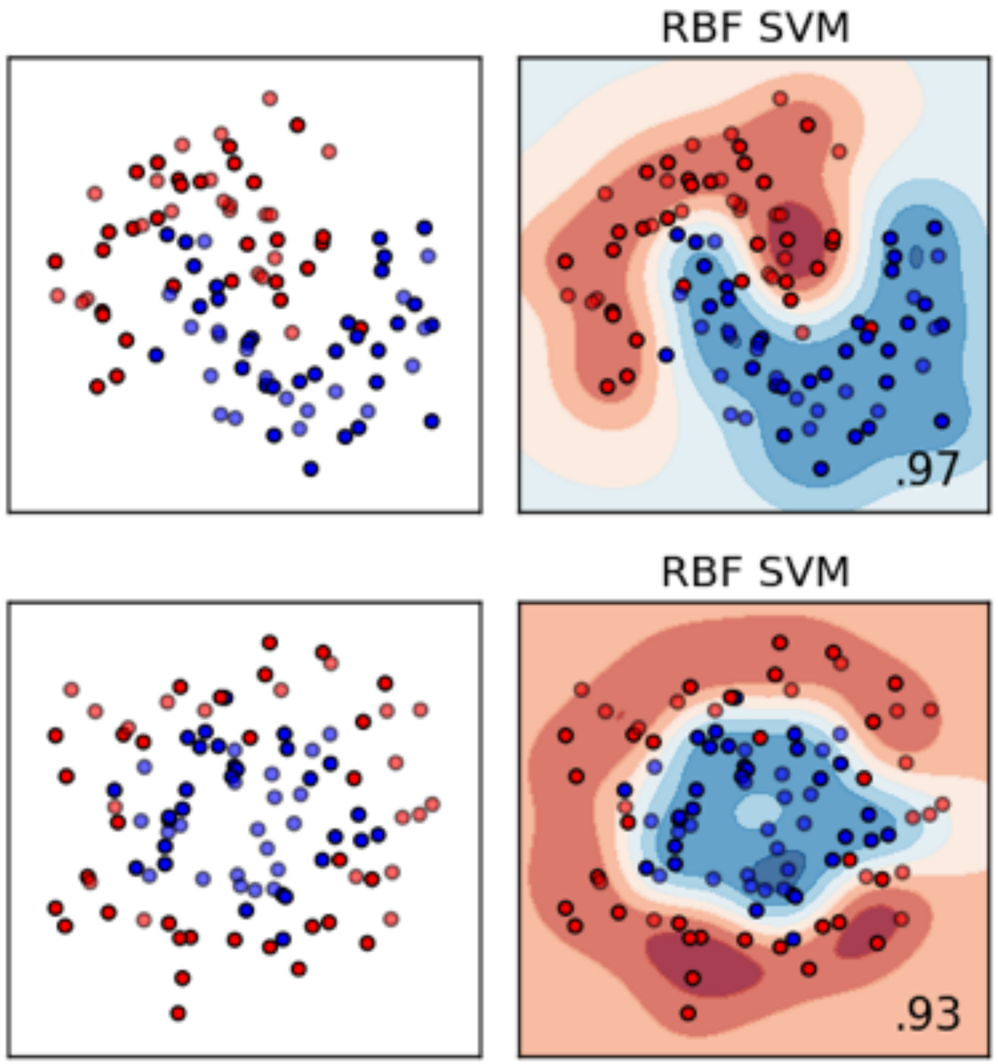
# COLLABORATORS

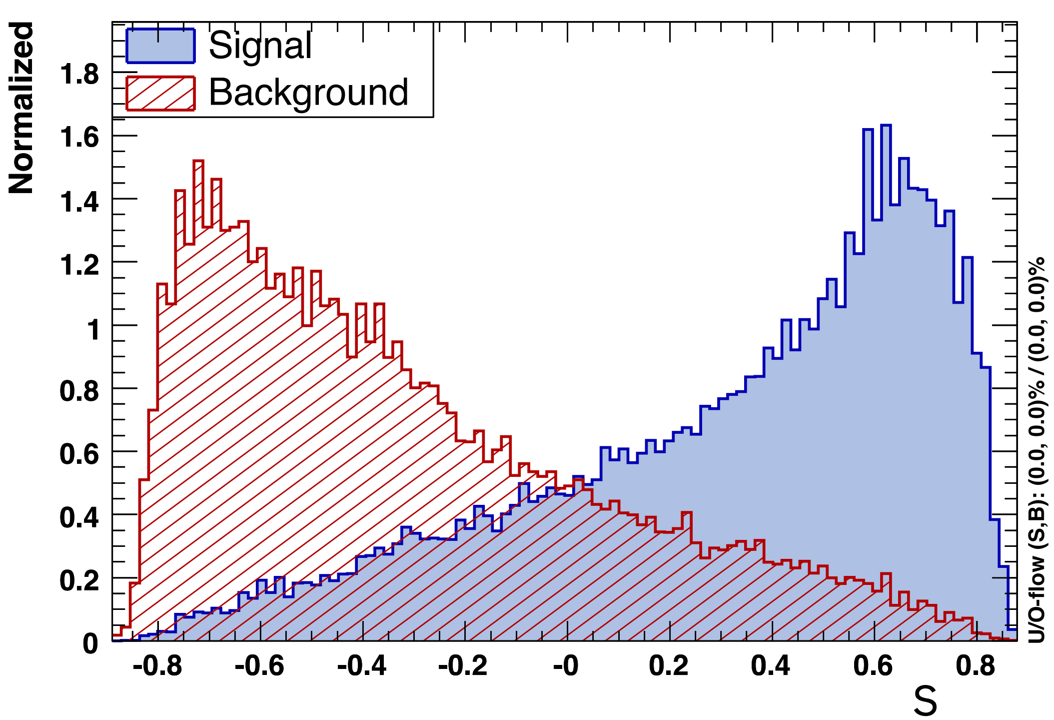


**Juan Pavez**
CS graduate student in Chile
Fellowship to work @ CERN summer ’15
@jgpavez

**Gilles Louppe**
Data Science Fellow
Funded via NSF DIANA/HEP
Based at CERN
PhD in machine learning
scikit-learn developer
@glouppe

# MACHINE LEARNING: CLASSIFIERS

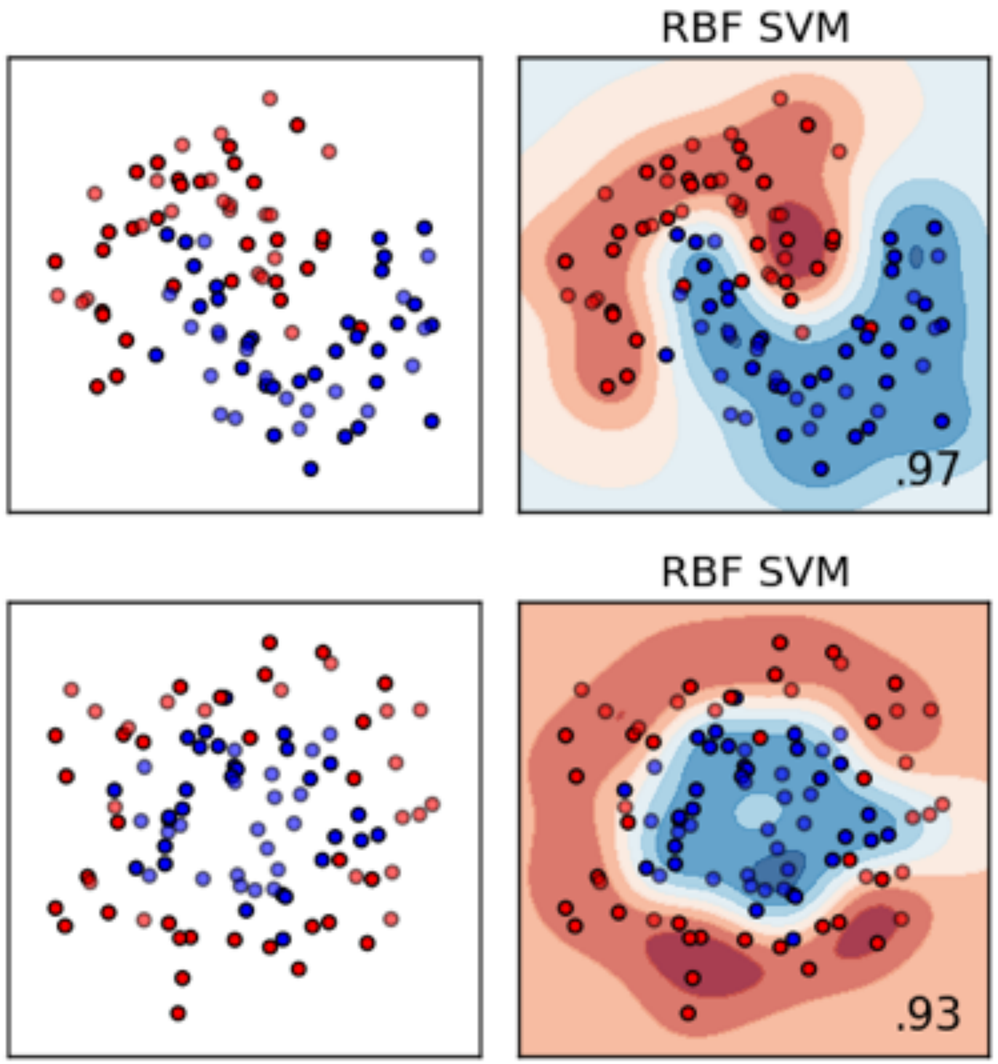


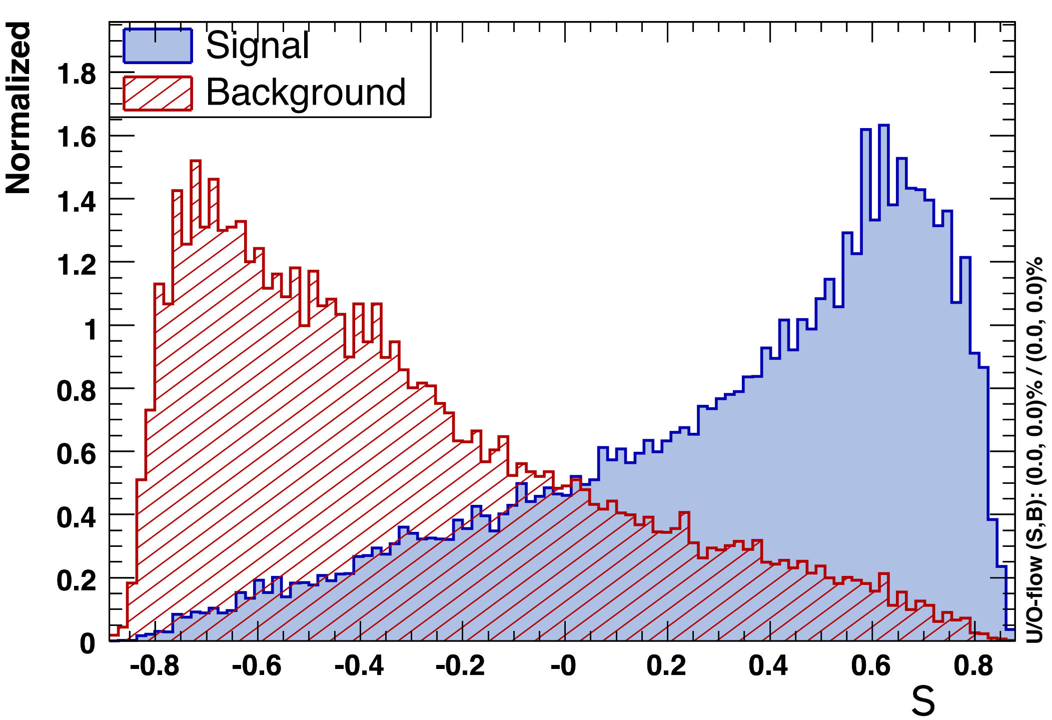




Common to use machine learning classifiers to separate signal ($H_1$) vs. background ($H_0$)

- want a function that maps signal to y=1 and background to y=0
- think of it as applied calculus of variations: find function s(x) that minimizes *loss*:

$$
\begin{aligned}
L[s] &= \int p(x|H_0)\,(0 - s(x))^2\,dx \\
&+ \int p(x|H_1)\,(1 - s(x))^2 dx \\
&\approx \sum_i (y_i - s(x_i))^2
\end{aligned}
$$




- applied calculus of variations: find function s(x) that minimizes *loss*:

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

- the optimal classifier would learn the regression function

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

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

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

We started with a classifier that was learning

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

Implicitly that classifier depends on $H_0$ and $H_1$ used to generate the training data. Make that explicit

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

Can do the same thing for any two points in parameter space. I call this a **parametrized classifier**

$$s(x; \theta_0, \theta_1) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$

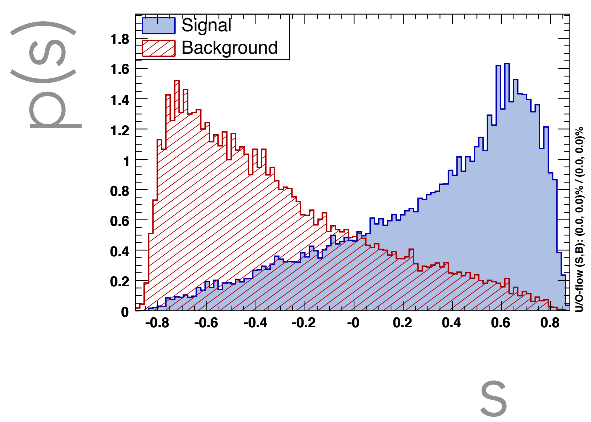
# GENERALIZED LIKELIHOOD RATIO TESTS

The target likelihood ratio test based on high-dimensional features x is:

$$T(D;\theta_0,\theta_1) = \prod_{e=1}^{n} \frac{p(x_e|\theta_0)}{p(x_e|\theta_1)}$$

I can show that an **equivalent test** can be made from 1-D projection

$$T(D;\theta_0,\theta_1) = \prod_{e} \frac{p(x_e|\theta_0)}{p(x_e|\theta_1)} = \prod_{e} \frac{p(s(x_e;\theta_0,\theta_1)|\theta_0)}{p(s(x_e;\theta_0,\theta_1)|\theta_1)}$$




**if** the map s: X → ℝ has the same level sets as the likelihood ratio

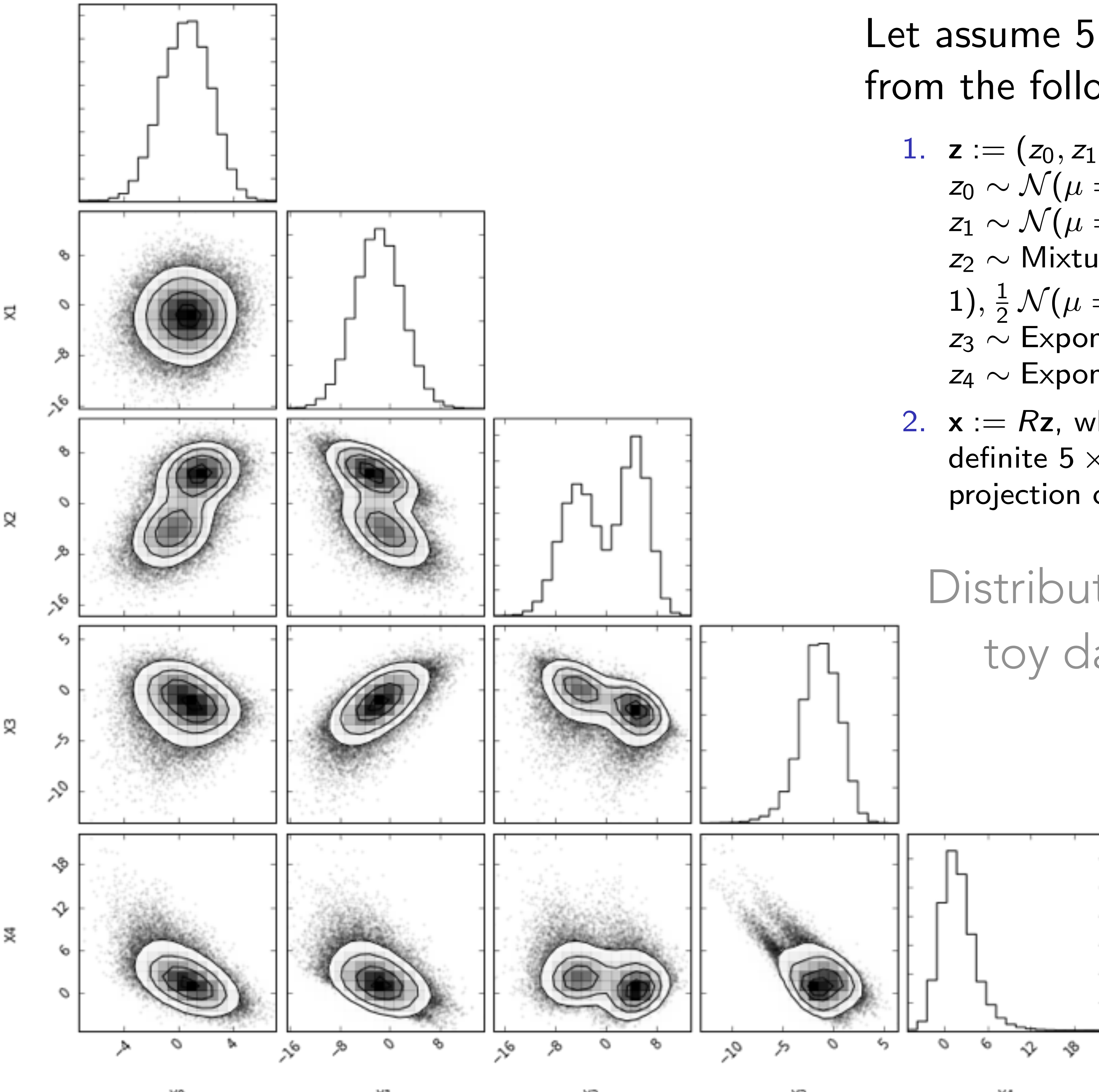
$$s(x;\theta_0;\theta_1) = \text{monotonic}[\ p(x|\theta_0)/p(x|\theta_1)\ ]$$

Remember that a **classifier** that minimizes squared loss ∑ [ $y_i$ - $s(x_i)$ ]$^2$ approximates the regression function, which has the same level sets!

K.C., G. Louppe, J. Pavez: http://arxiv.org/abs/1506.02169

# AN EXAMPLE

# THE DATA



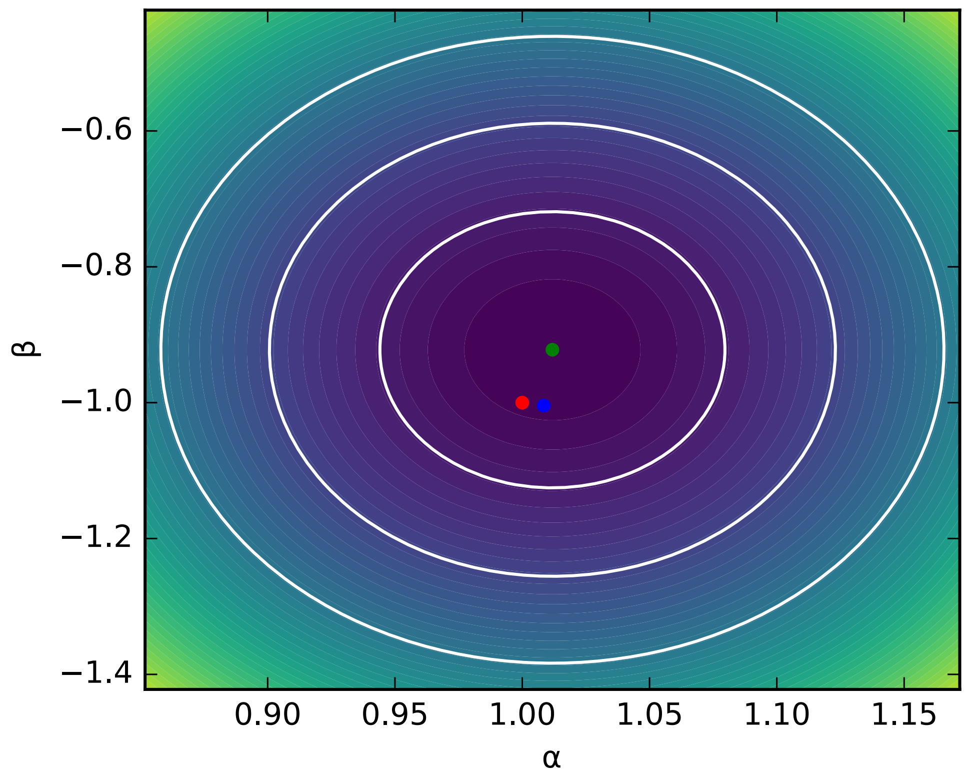

Let assume 5D data $\mathbf{x}$ generated from the following process $p_0$:

1. $\mathbf{z} := (z_0, z_1, z_2, z_3, z_4)$, such that
   $z_0 \sim \mathcal{N}(\mu = \alpha, \sigma = 1)$,
   $z_1 \sim \mathcal{N}(\mu = \beta, \sigma = 3)$,
   $z_2 \sim \text{Mixture}(\frac{1}{2}\mathcal{N}(\mu = -2, \sigma = 1), \frac{1}{2}\mathcal{N}(\mu = 2, \sigma = 0.5))$,
   $z_3 \sim \text{Exponential}(\lambda = 3)$, and
   $z_4 \sim \text{Exponential}(\lambda = 0.5)$;
2. $\mathbf{x} := R\mathbf{z}$, where $R$ is a fixed semi-positive definite $5 \times 5$ matrix defining a fixed projection of $\mathbf{z}$ into the observed space.

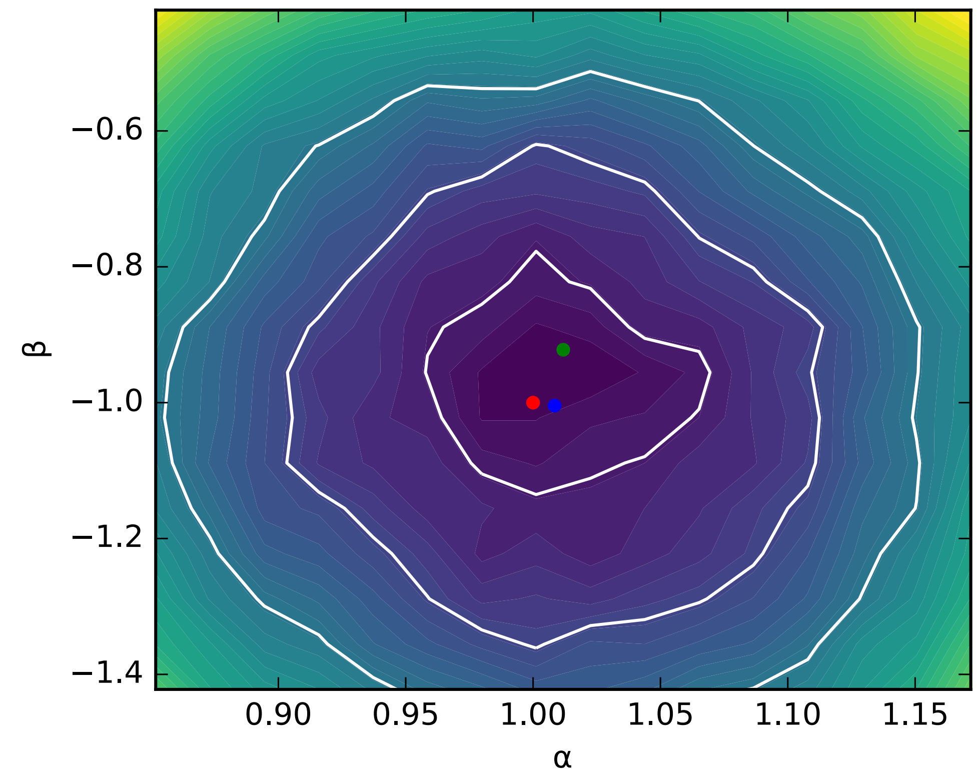
Distribution depends on α, β

toy data with α=1, β=-1

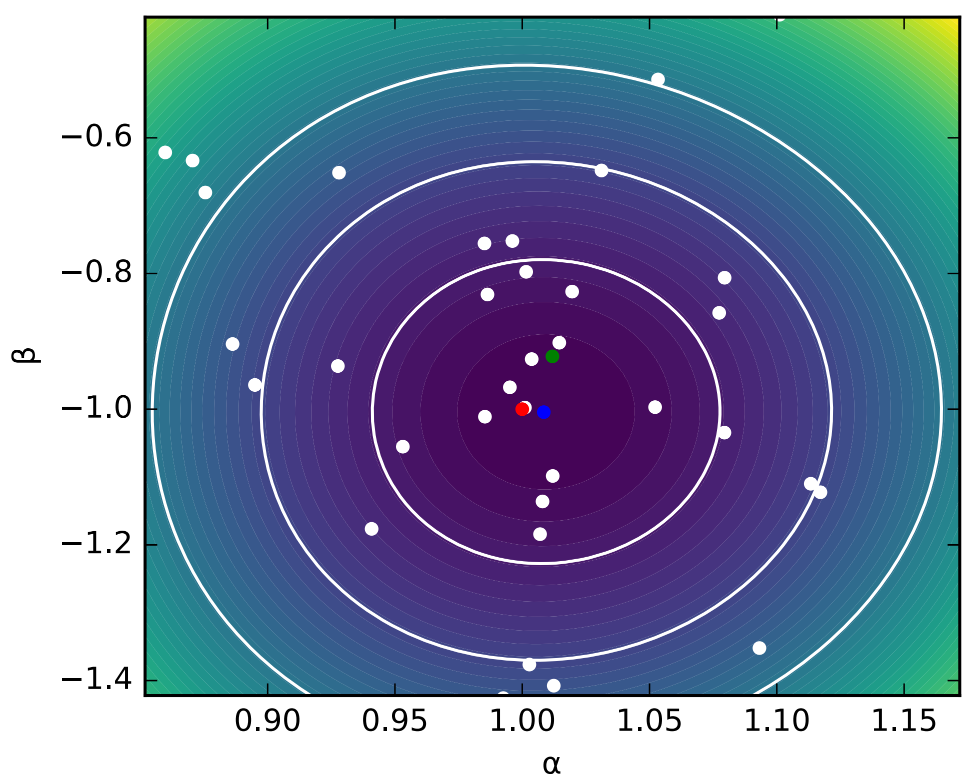
# LIKELIHOOD CONTOURS



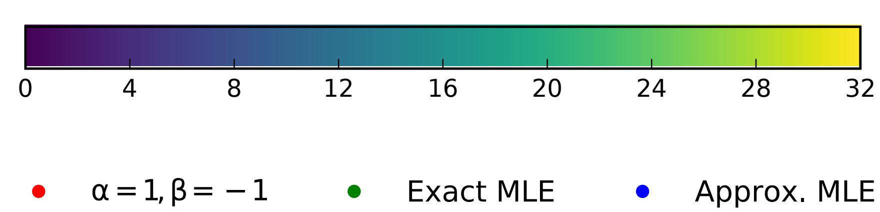

Exact likelihood




Approximate likelihood




Approximate likelihood (smoothed)




(d)

# DIAGNOSTICS

In practice $\hat{r}(\hat{s}(\mathbf{x};\theta_0,\theta_1))$ will not be exact. Diagnostic procedures are needed to assess the quality of this approximation.

1. For inference, the value of the MLE $\hat{\theta}$ should be independent of the value of $\theta_1$ used in the denominator of the ratio.

## The denominator in the likelihood ratio is just a shift

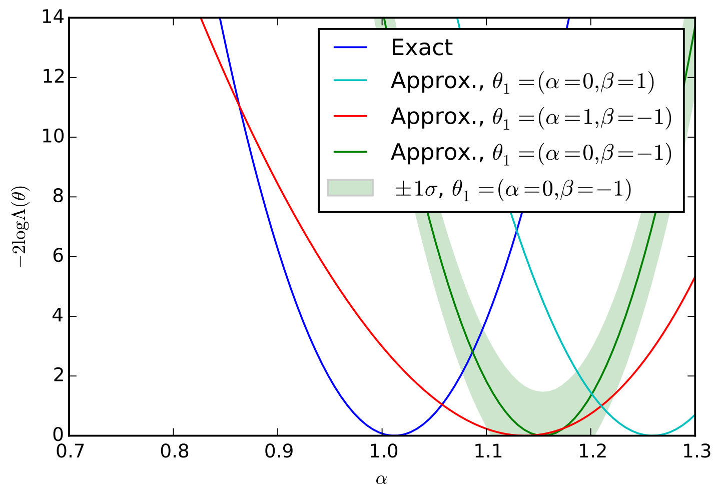
$$\hat{\theta} = \arg\max_{\theta} \sum \ln \frac{p(x_e|\theta)}{p(x_e|\theta_1)} = \arg\max_{\theta} \sum \ln \frac{p(s(x_e;\theta,\theta_1)|\theta)}{p(s(x_e;\theta,\theta_1)|\theta_1)} \,. \tag{4.4}$$

It is important that we include the denominator $p(s(x_e;\theta,\theta_1)|\theta_1)$ because this cancels Jacobian factors that vary with $\theta$.

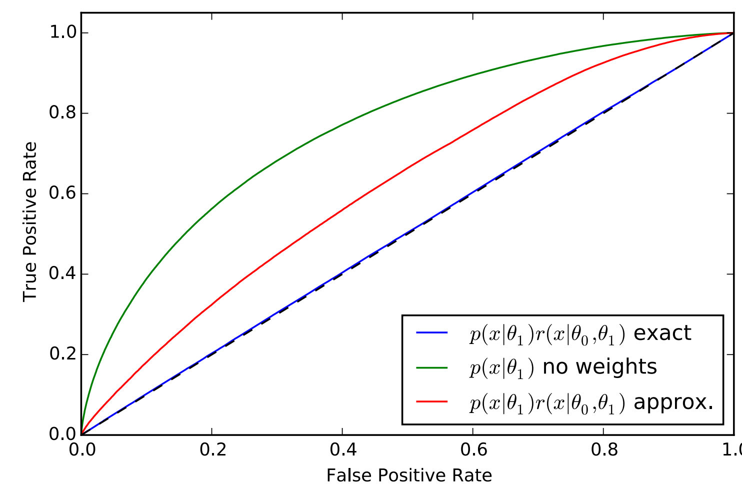
## Provides a non-trivial diagnostic:

$$\frac{p_1(s^*)}{p_0(s^*)} = \frac{p_1(x)}{p_0(x)} \boxed{\frac{\int d\Omega_{s^*} p_0(x)/|\hat{n}\cdot\nabla s|}{\int d\Omega_{s^*} p_0(x)/|\hat{n}\cdot\nabla s|}} = \frac{p_1(x)}{p_0(x)}$$

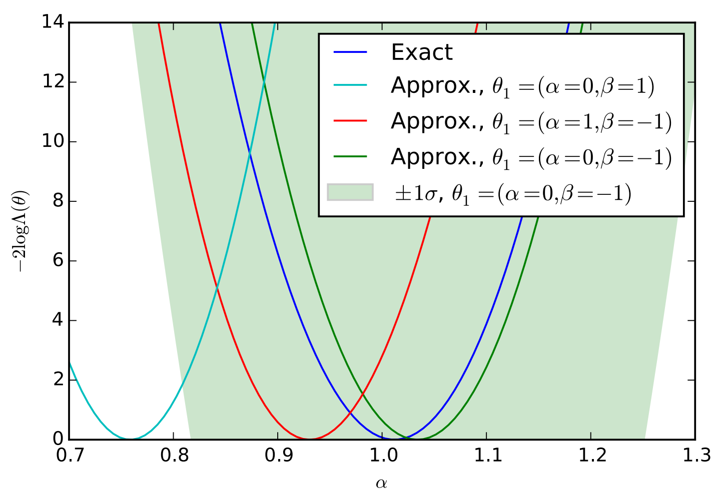
# DIAGNOSTICS



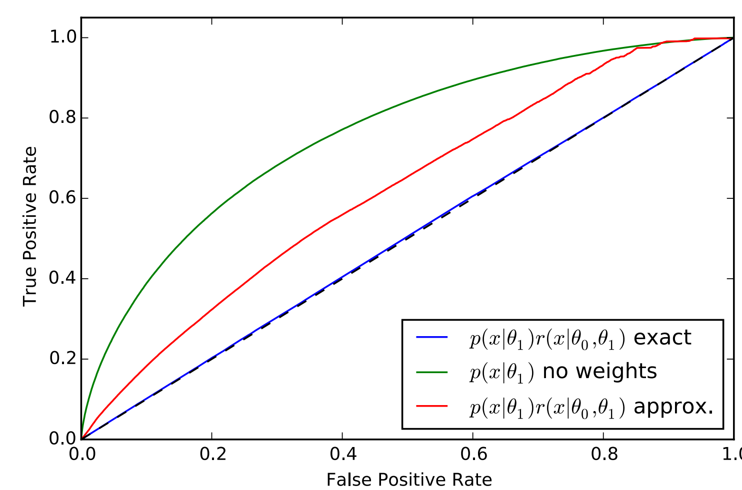

(a) Poorly trained, well calibrated.



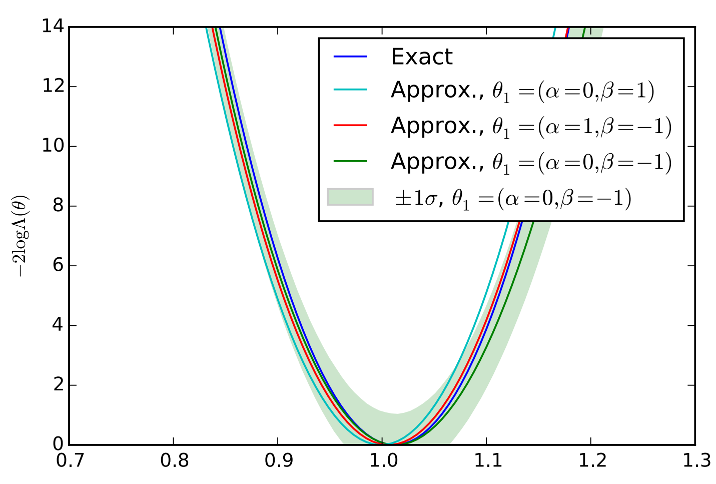

(b) Poorly trained, well calibrated.



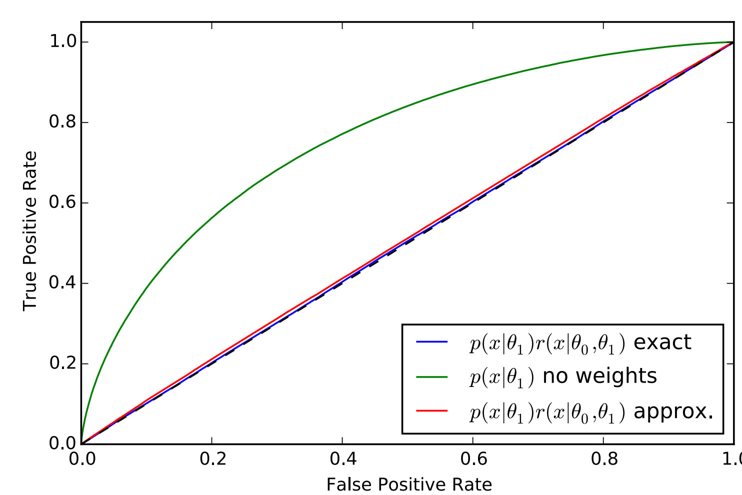

(c) Poorly calibrated, well trained.




(d) Poorly calibrated, well trained.




(e) Well trained, well calibrated.

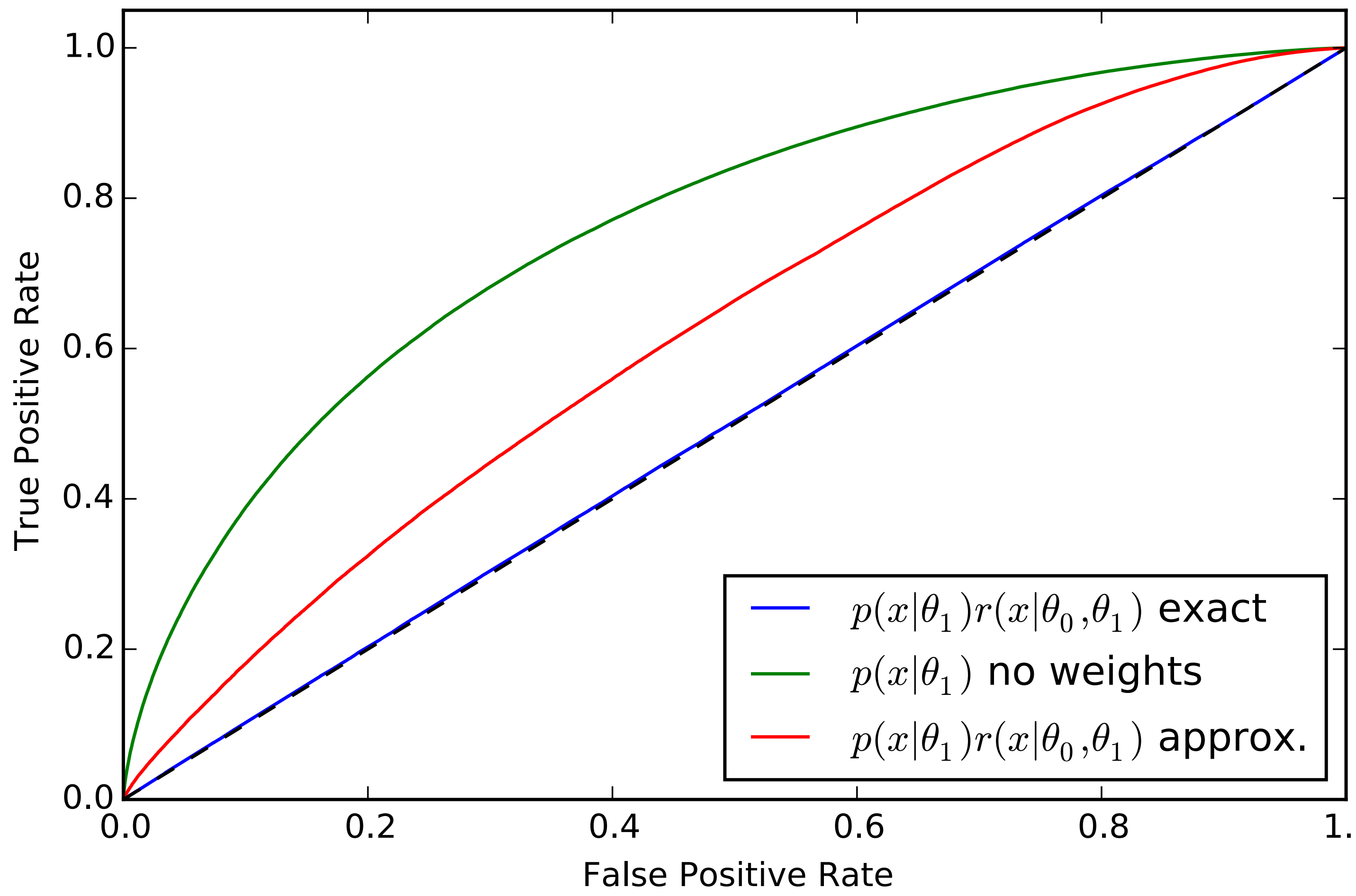



(f) Well trained, well calibrated.

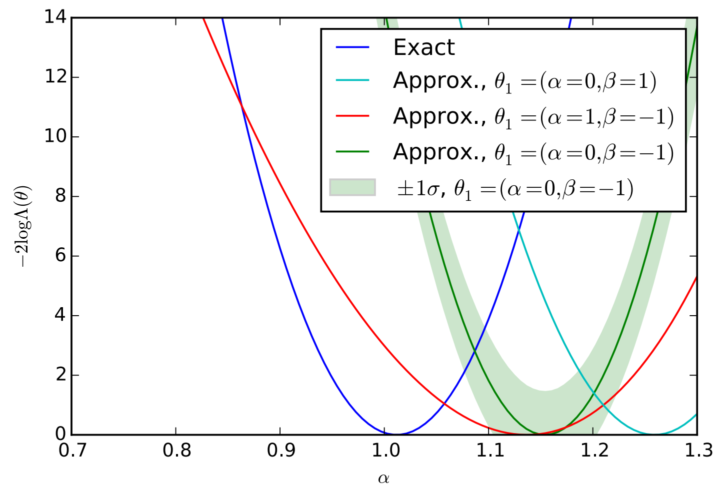
Train a new classifier to **discriminate** between events from target $p(x|\theta_0)$ and events resampled from original distribution $p(x|\theta_1)$ with probabilities given by the predicted weights $\hat{r}(x|\theta_0, \theta_1) \approx p(x|\theta_0)/p(x|\theta_1)$

- classifier can easily distinguish unweighted distributions;
- exact weights are perfect (AUC~0.5)

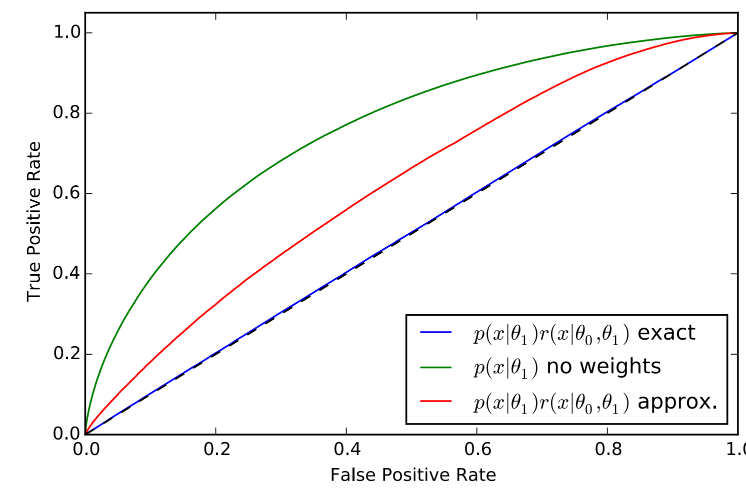
**Important**:
Performance evaluated on independent testing sample



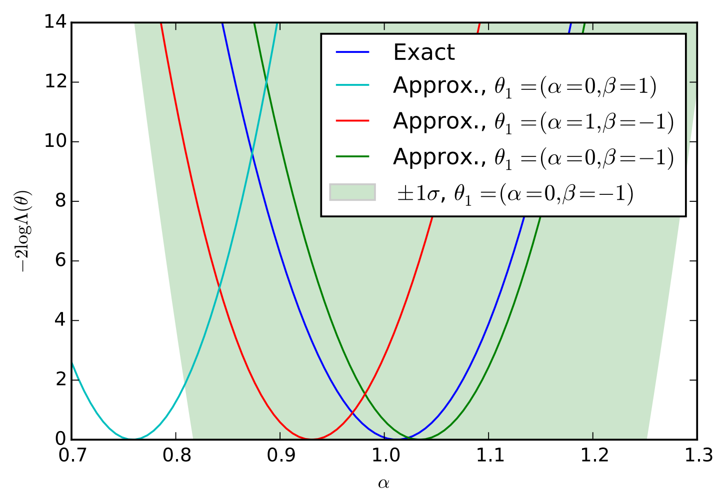
# DIAGNOSTICS



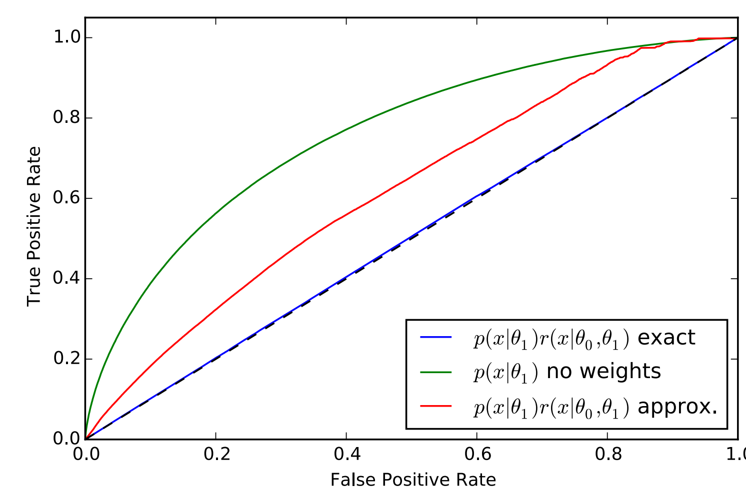

(a) Poorly trained, well calibrated.



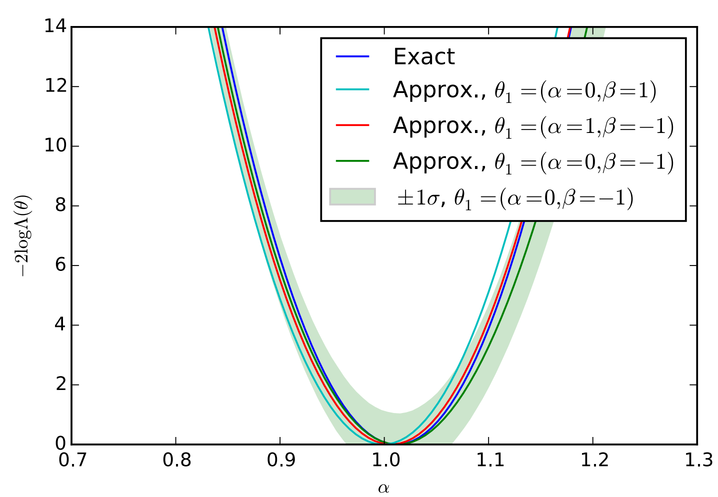

(b) Poorly trained, well calibrated.



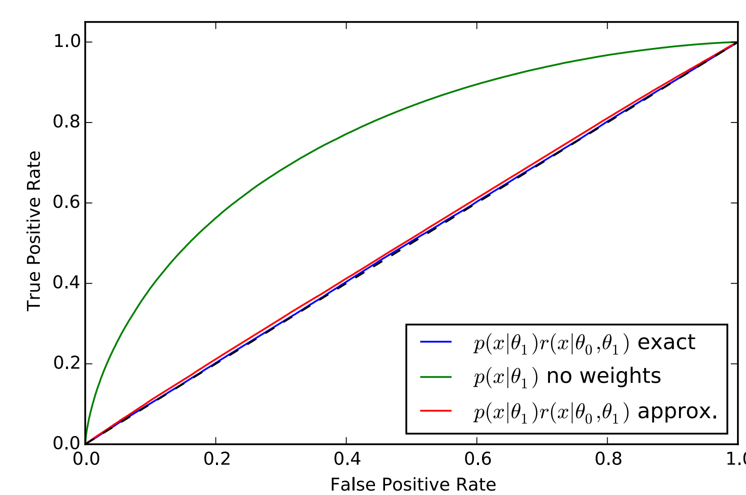

(c) Poorly calibrated, well trained.




(d) Poorly calibrated, well trained.




(e) Well trained, well calibrated.




(f) Well trained, well calibrated.

# SPECIAL CASE:
# MIXTURE MODELS

# MIXTURE MODEL

Often the model for the data is a mixture of different components $w_c$

- to be more generic, consider parametrized coefficients $w_c(\theta)$

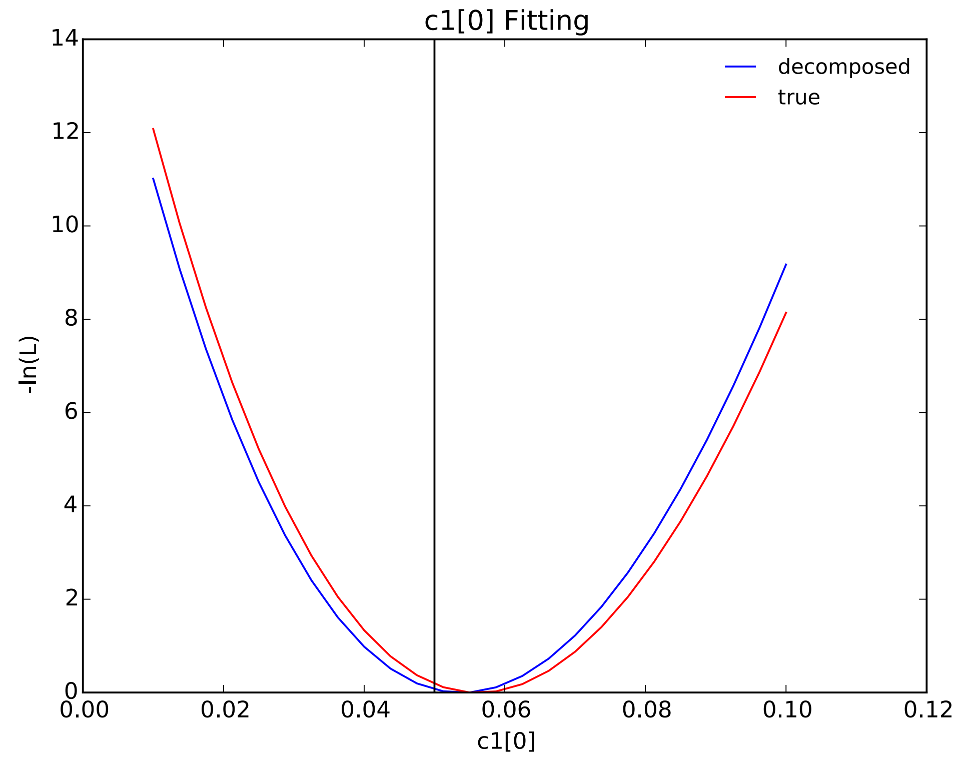
$$p(x|\theta) = \sum_c w_c(\theta) p_c(x)$$

I worked out a way to decompose the training into pairwise comparisons:

$$\frac{p(x|\theta_0)}{p(x|\theta_1)} = \frac{\sum_c w_c(\theta_0) p_c(x)}{\sum_{c'} w_{c'}(\theta_1) p_{c'}(x)}$$
$$= \sum_c \left[ \sum_{c'} \frac{w_{c'}(\theta_1)}{w_c(\theta_0)} \frac{p_{c'}(x)}{p_c(x)} \right]^{-1}$$
$$= \sum_c \left[ \sum_{c'} \frac{w_{c'}(\theta_1)}{w_c(\theta_0)} \frac{p_{c'}(s_{c,c'})}{p_c(s_{c,c'})} \right]^{-1}$$

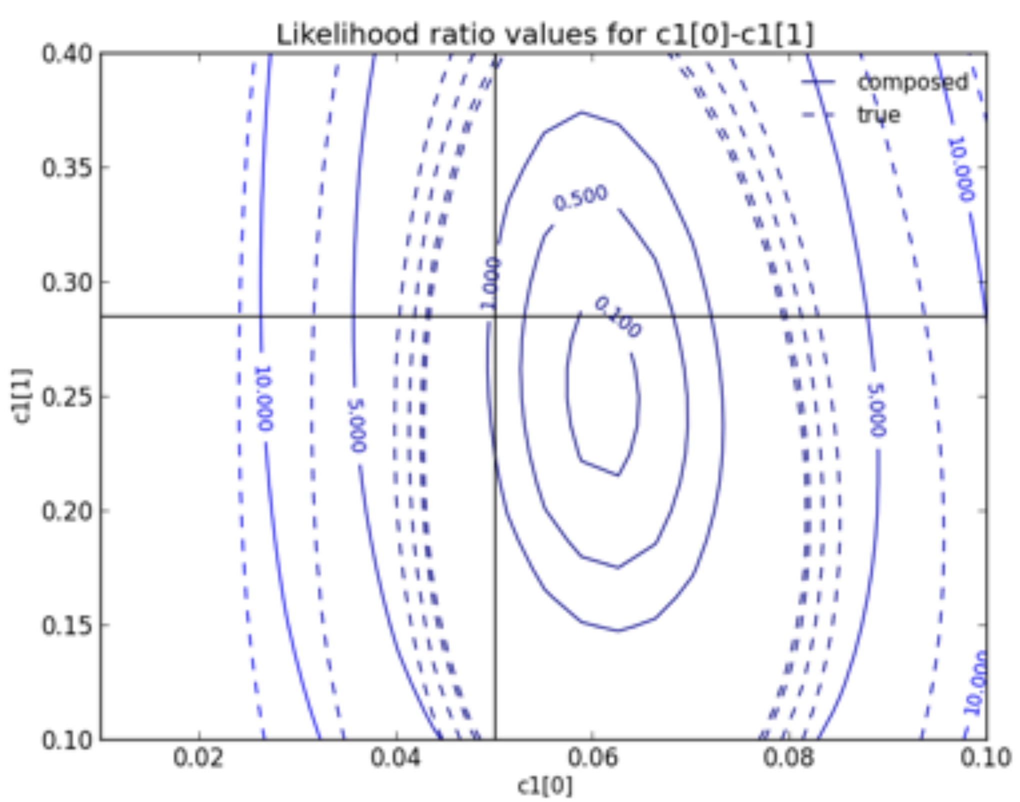
Last line uses the main result of the paper, need a classifier for each pairwise ( $c$ vs. $c'$ ) comparison (n (n-1)/2 of them)

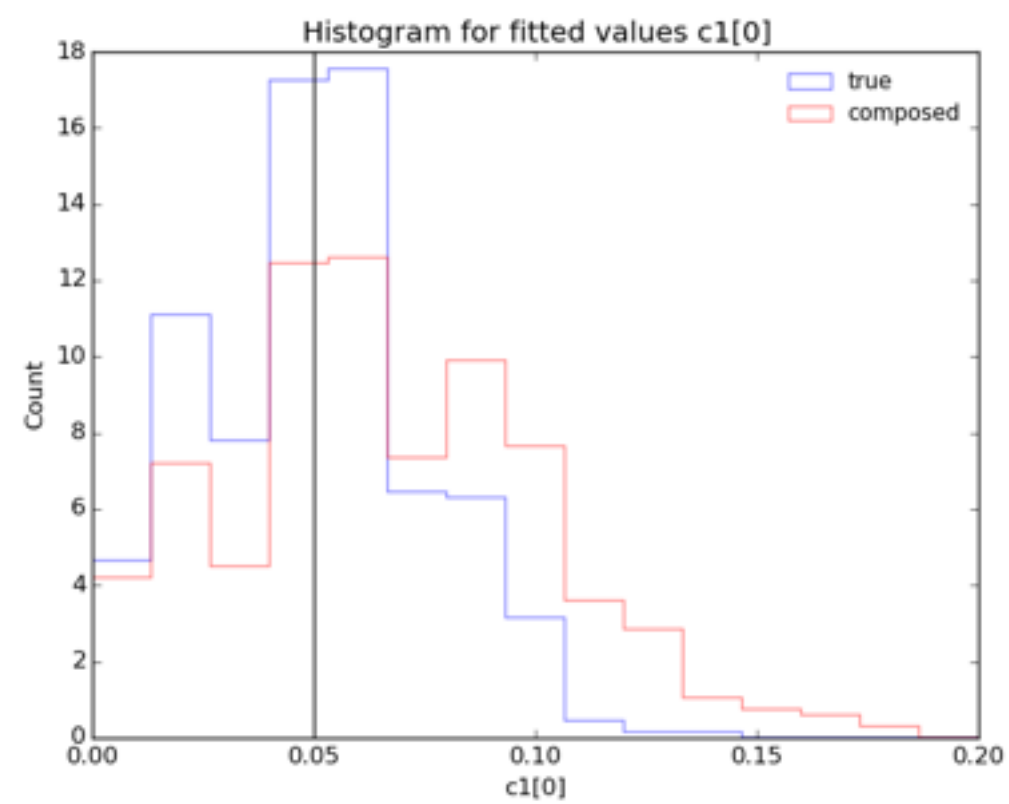
# RESULTS FOR 10-DIM EXAMPLE




**Left**: fit to mixture coefficients for single pseudo-experiment

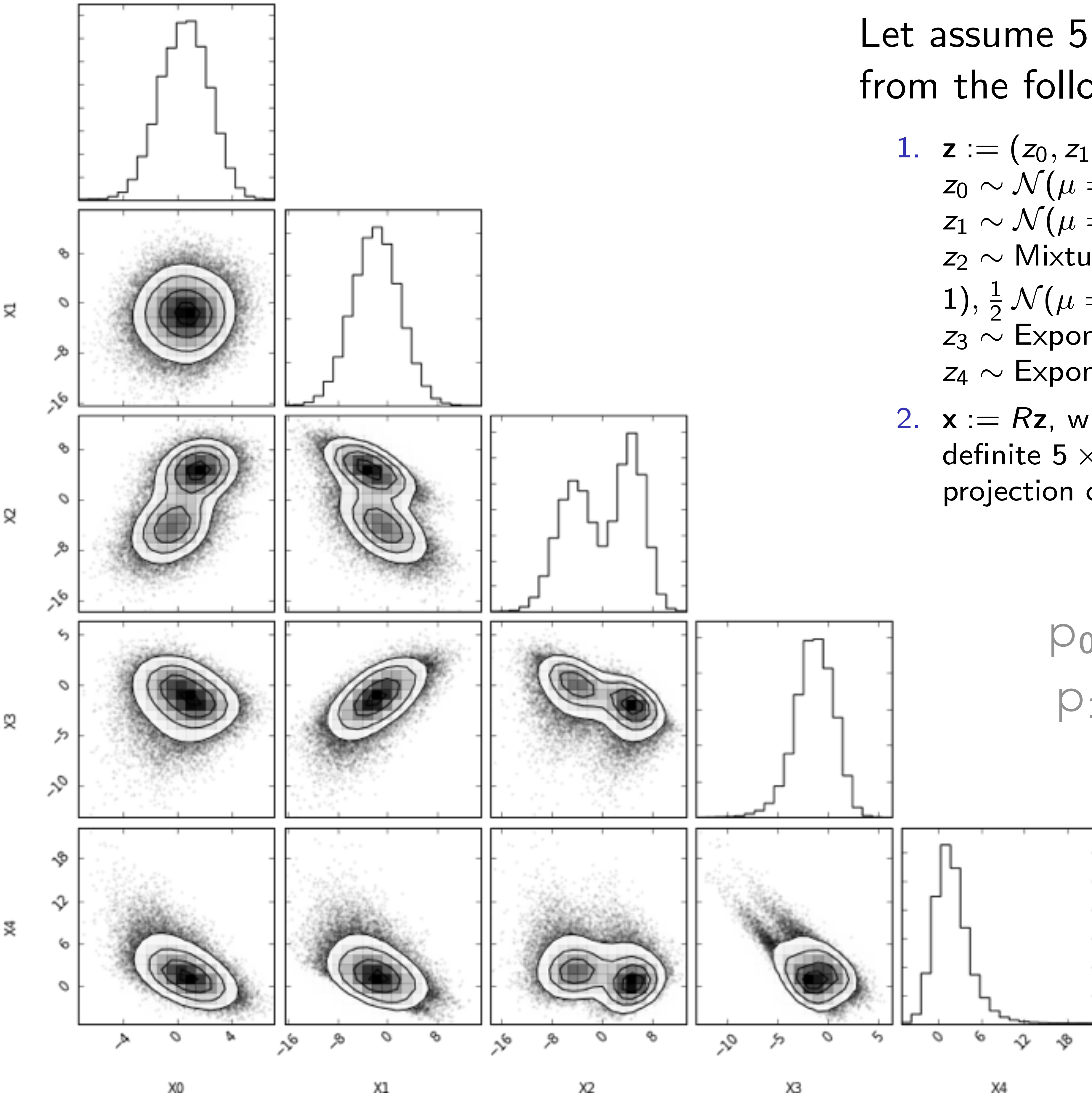
**Right**: histogram of best fit of one coefficient for many pseudo-experiments



# CONNECTION TO REWEIGHTING

# THE DATA



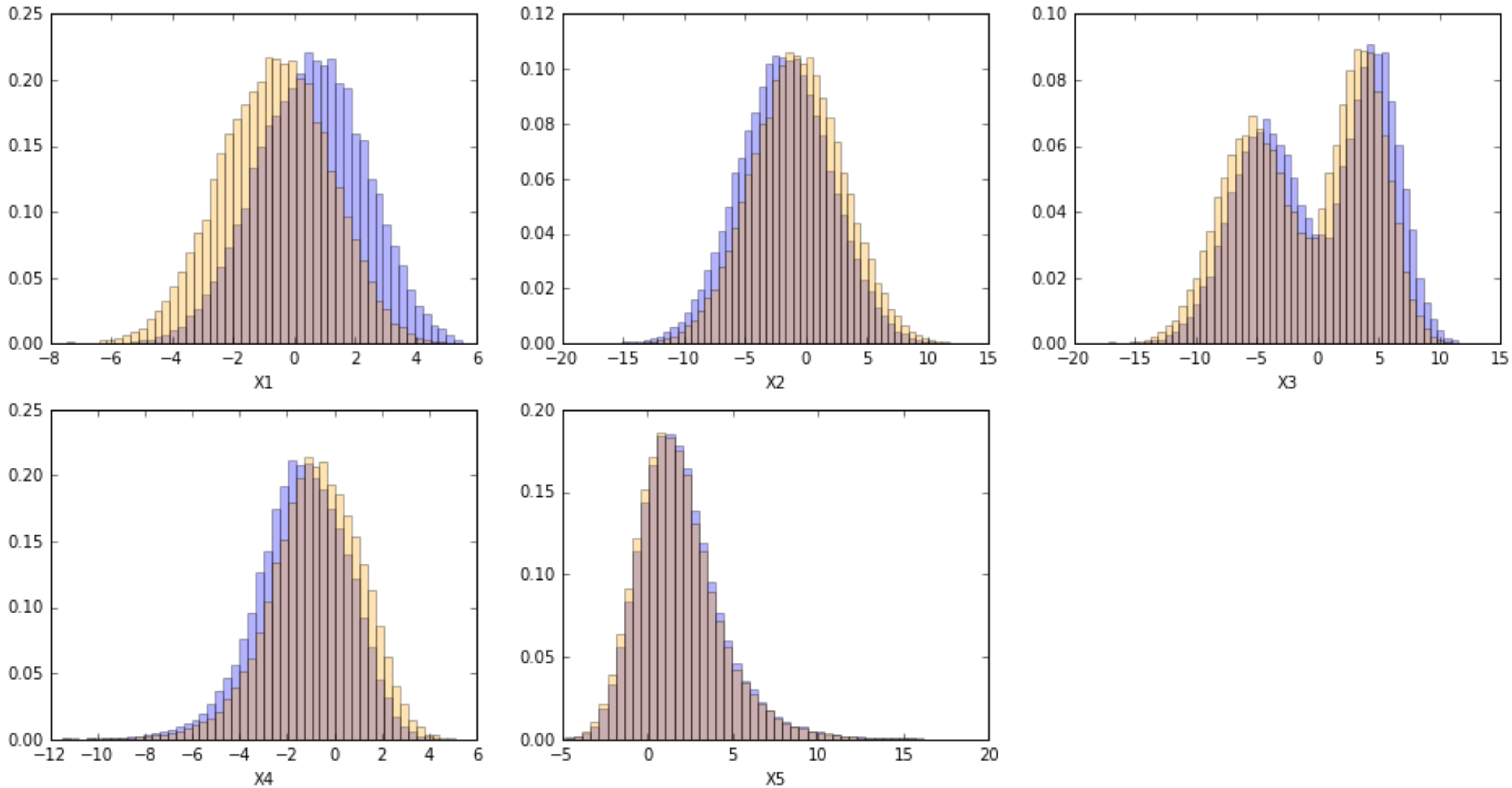

Let assume 5D data $\mathbf{x}$ generated from the following process $p_0$:

1. $\mathbf{z} := (z_0, z_1, z_2, z_3, z_4)$, such that $z_0 \sim \mathcal{N}(\mu = \alpha, \sigma = 1)$, $z_1 \sim \mathcal{N}(\mu = \beta, \sigma = 3)$, $z_2 \sim \text{Mixture}(\frac{1}{2}\mathcal{N}(\mu = -2, \sigma = 1), \frac{1}{2}\mathcal{N}(\mu = 2, \sigma = 0.5))$, $z_3 \sim \text{Exponential}(\lambda = 3)$, and $z_4 \sim \text{Exponential}(\lambda = 0.5)$;
2. $\mathbf{x} := R\mathbf{z}$, where $R$ is a fixed semi-positive definite $5 \times 5$ matrix defining a fixed projection of $\mathbf{z}$ into the observed space.

$p_0$ has $\alpha$=1, $\beta$=-1

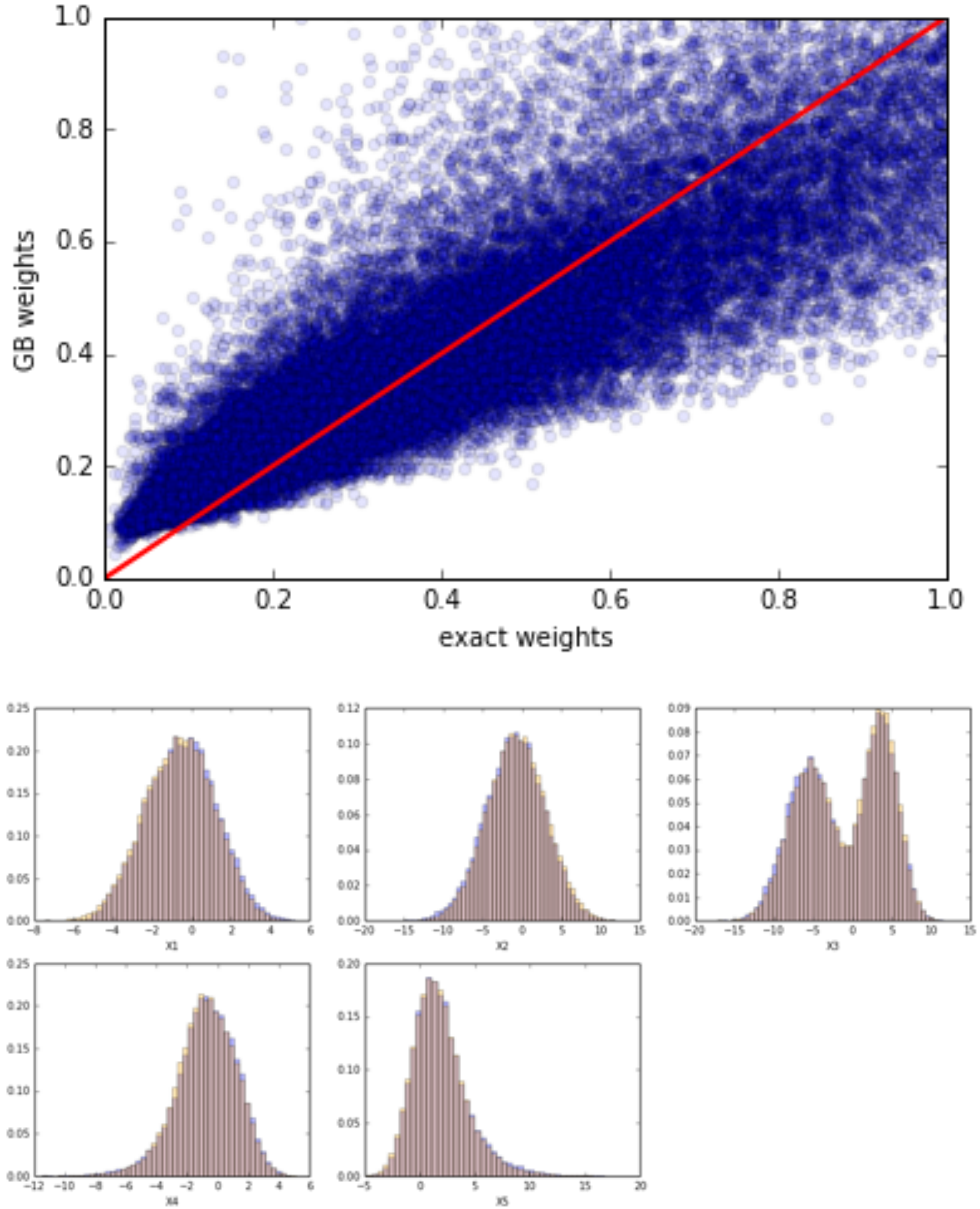
$p_1$ has $\alpha$=0, $\beta$=0

## 1-d projections of the original and target distributions

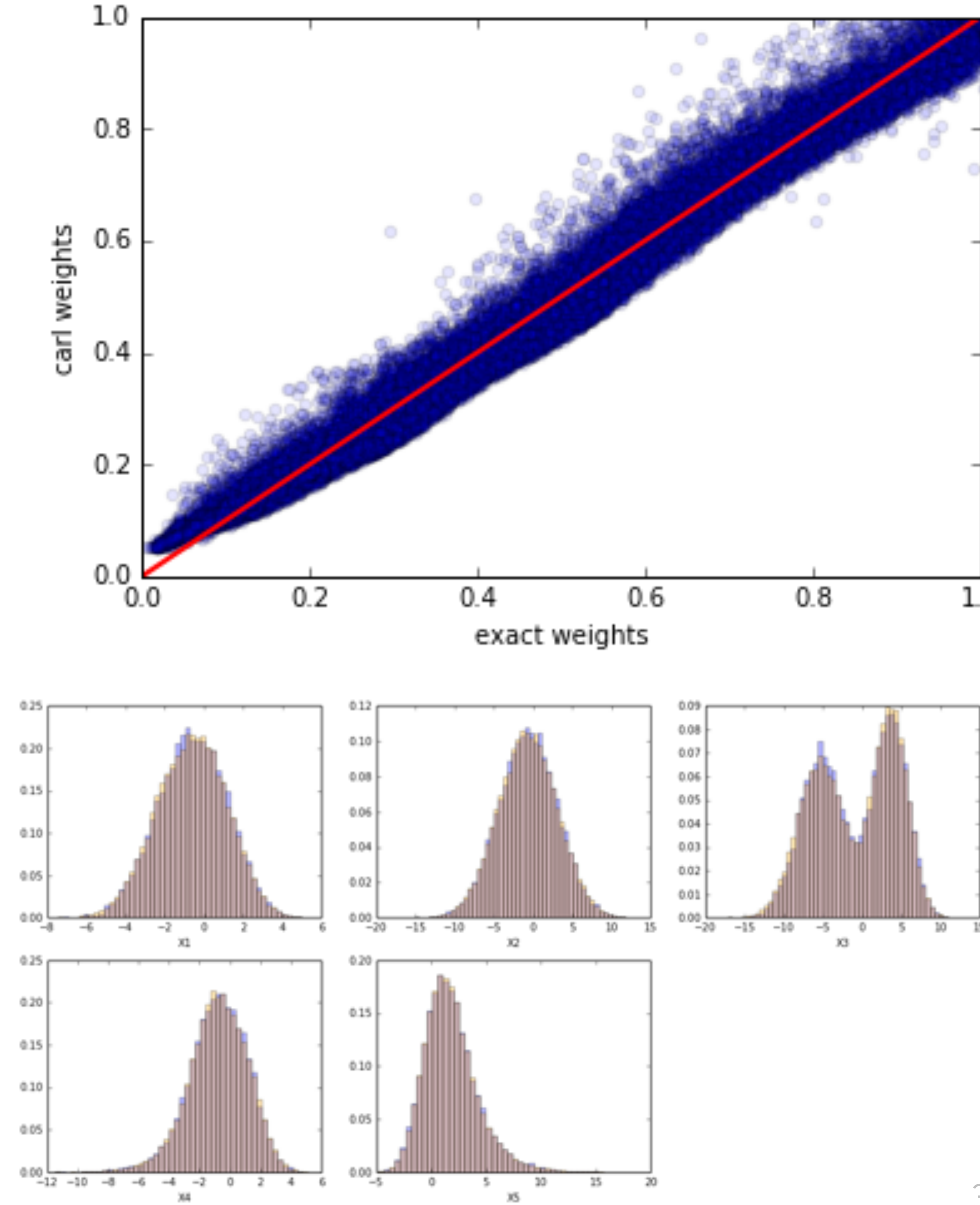


# TWO REWEIGHING METHODS: 100K SAMPLES

## hep_ml.GBReweigher




## carl with calibrated MLP



# EVALUATING THE QUALITY OF THE REWEIGHTING

Train a new classifier to **discriminate** between events from target and events resampled from original distribution with probabilities given by the predicted weights

- classifier can easily distinguish unweighted distributions;
- exact weights are perfect (AUC~0.5)
- carl doing a little better than GBReweighter on this problem (no special effort to tune either)
- neither is perfect

**Important**:
Performance evaluated on independent testing sample