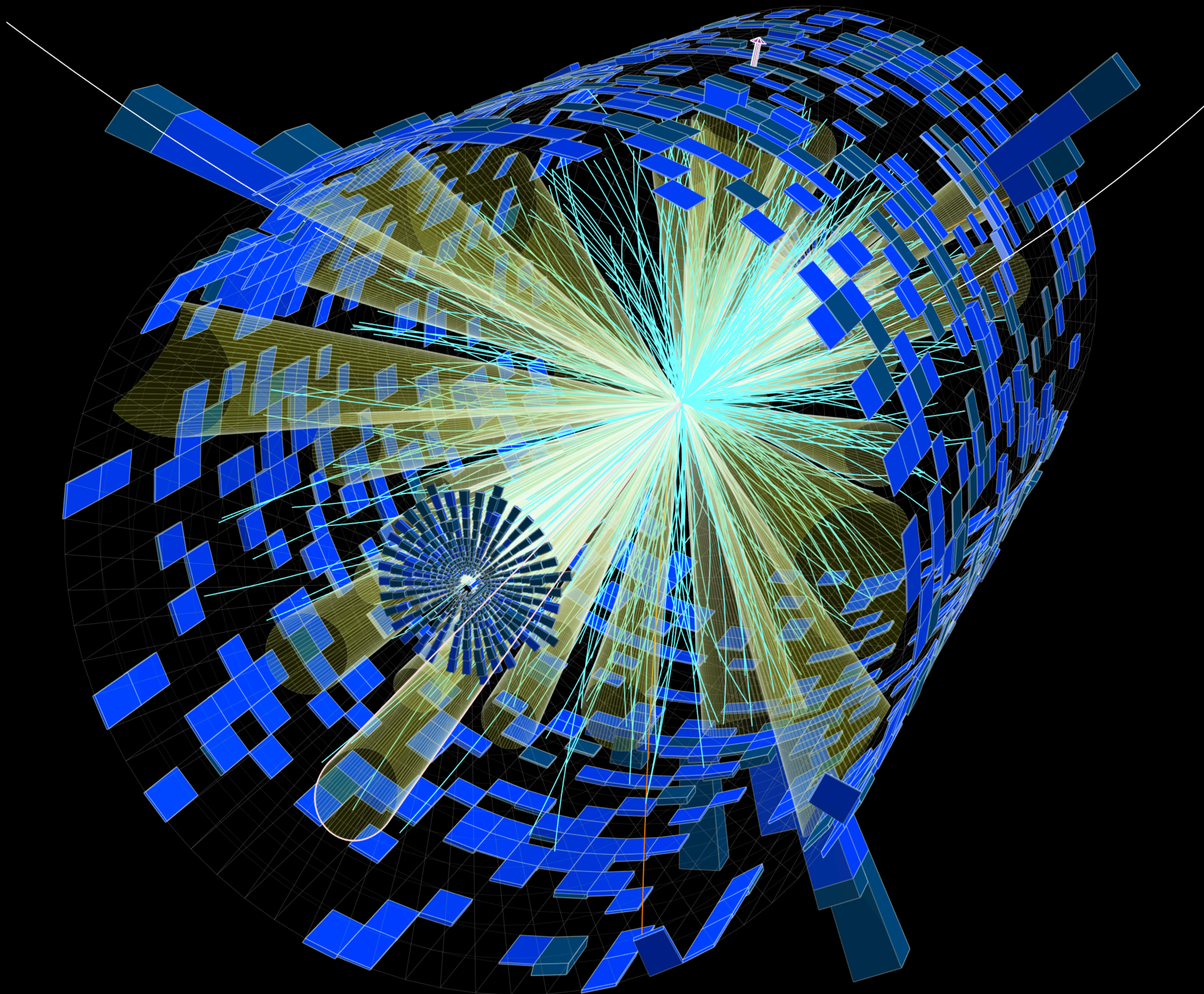
PhyStat: Statistics meets ML

Systematics: mystery or muse?



@KyleCranmer
University of Wisconsin-Madison
Data Science Institute
Physics, Computer Science, Statistics

My charge

Theme of workshop: Statistics meets Machine Learning

- This describes a lot of my research - I'm sad I can't be there
- Purest expression is probably simulation-based Inference

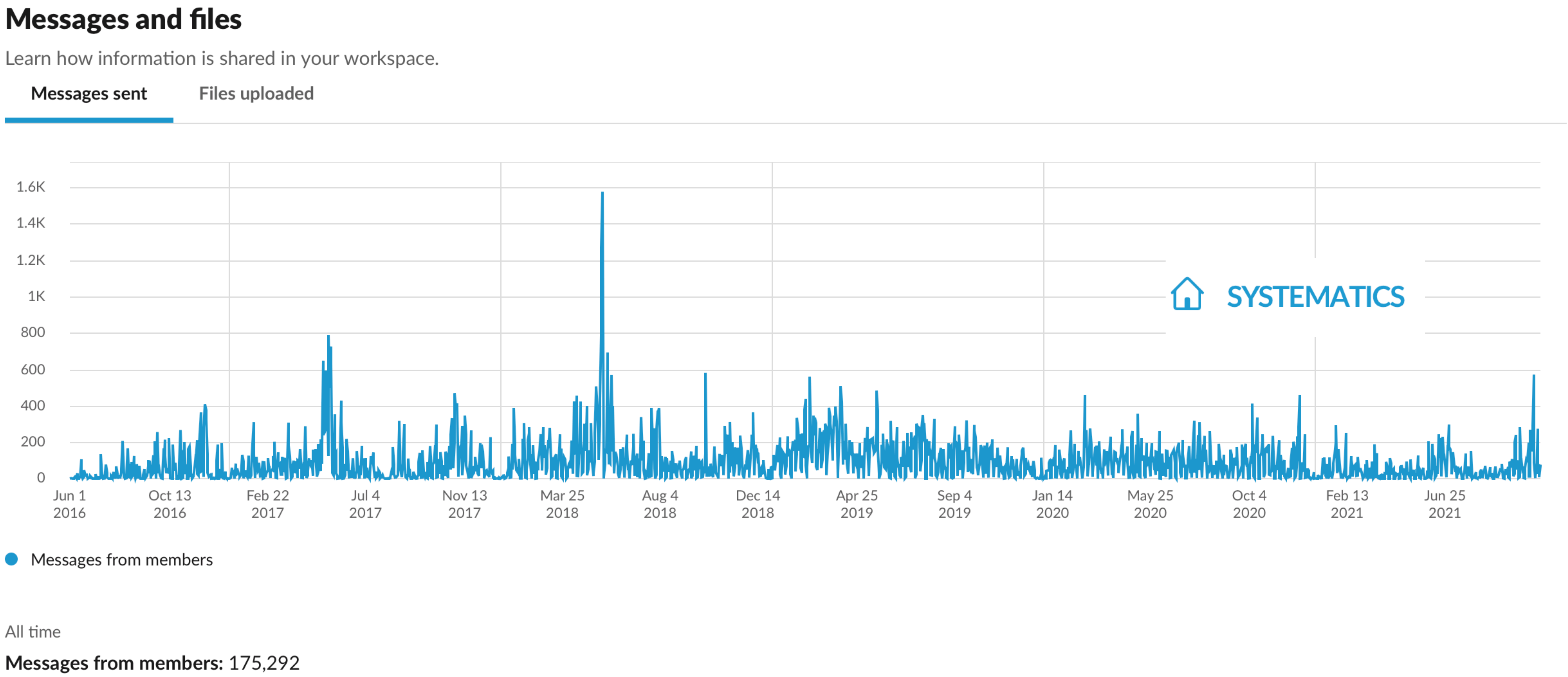
Louis specifically asked me to talk about **systematic uncertainties**

A personal note on "systematics"

Thinking about systematics in ML is what inspired my work in simulation-based inference

- Slack channel I created in 2015 for my ML work with collaborators is called "**systematics**"
  - Now has 125 members and ~200,000 messages!

Location of personal eureka moment at UC Irvine



Messages and files

Learn how information is shared in your workspace.

Messages sent | Files uploaded

● Messages from members

All time

Messages from members: 175,292

Inspirational message / initial theme of talk

Systematic uncertainties usually have a negative connotation since they reduce the sensitivity of an experiment.

However, the practical and conceptual challenges posed by various types of systematic uncertainty also have a long track record of motivating new ideas.

Original Plan for Talk: outline some examples from my own career where **systematics were my muse for innovation**

- That theme is still there, but I pivoted

The struggle is real

I want to make several points that isolate / highlight different issues

- Initial organization seemed very scattered
- Difficult to convert into a linear story

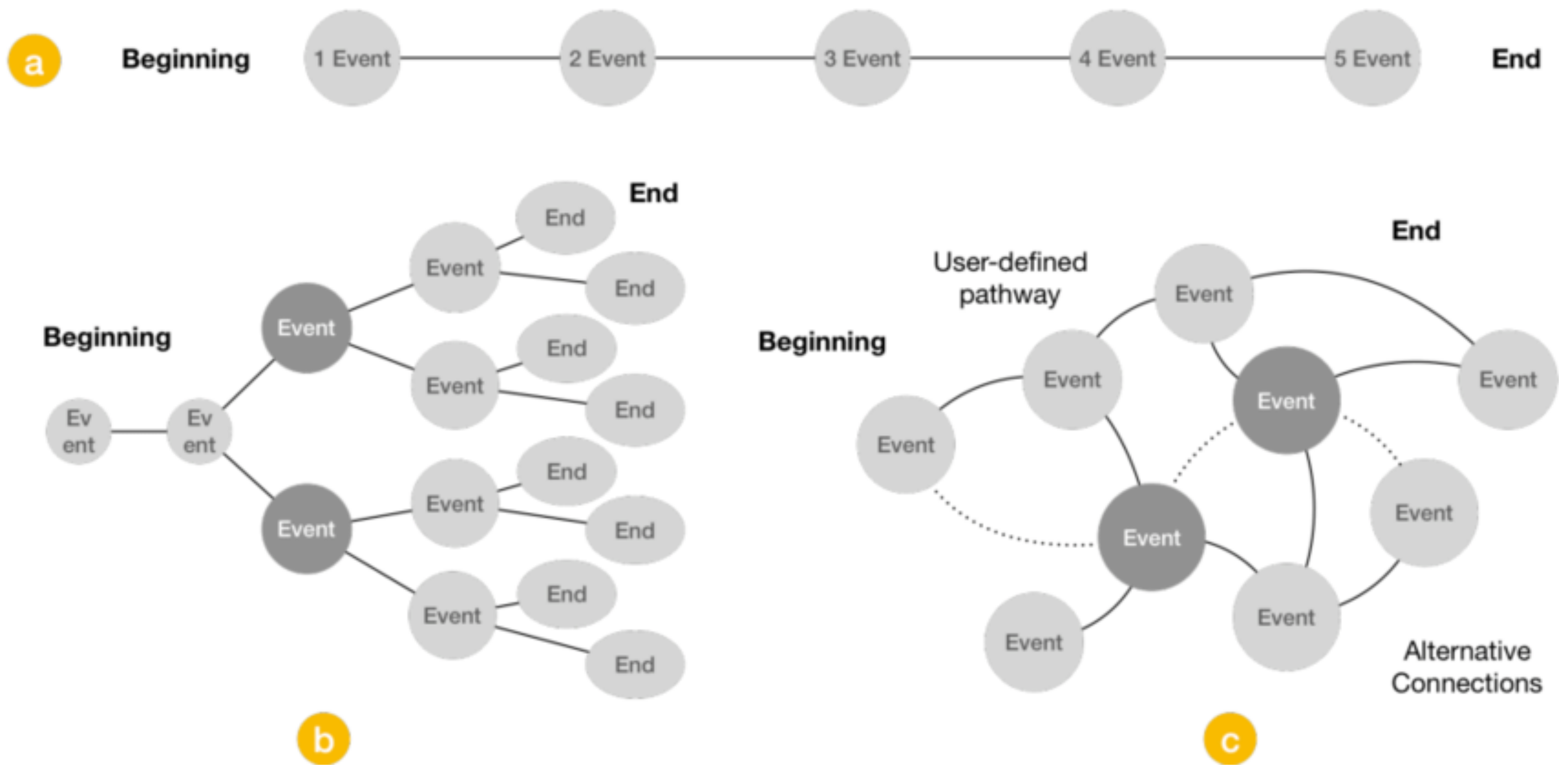


Figure source: A Design Experience for Interactive Narrative Based on The User Behavior, Yuan Yao, Hailing Mi

Organization: 5-D "Outline"

I want to make several points that isolate / highlight different issues

- An organizational principle emerged that helps isolate individual points

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Classical histogram-based SBI in HEP

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Improved treatment of systematics in traditional approach

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / “non-parametric” (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Traditional use of ML for searches

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Genetic Programming, INFERNO, Neon

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / “non-parametric” (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

(Locally) Sufficient statistics for measurements

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / “non-parametric” (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Learning to Pivot

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event (Classifier)	Experiment-wide (Adversary & Hyper parameter opt.)

SBI: Neural Likelihood Ratio Estimation

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Learning to Profile

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Traditional binned-template analysis

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / “non-parametric” (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

High-level overview of traditional binned-template analysis

Inhomogeneous Poisson Process

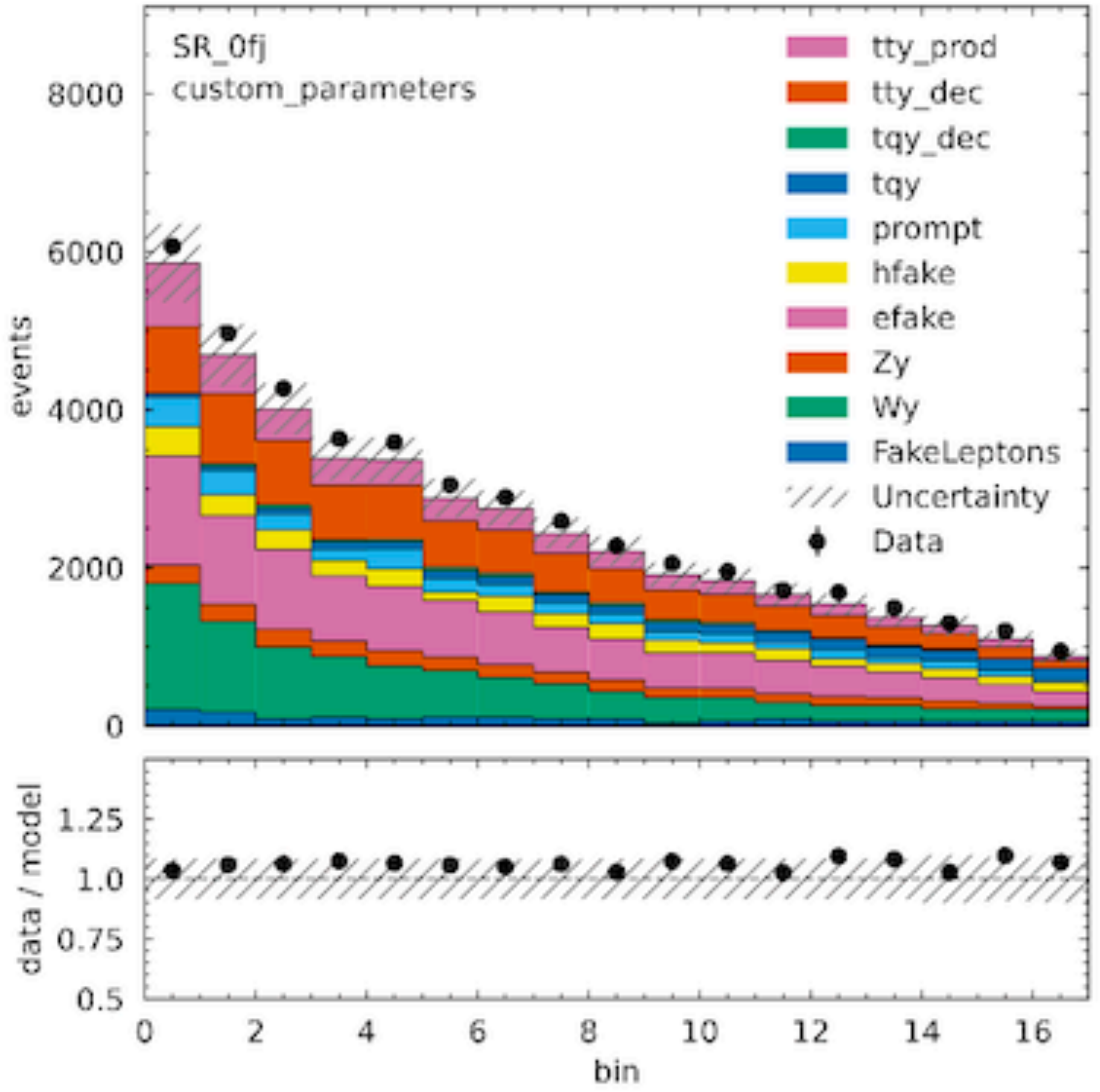
- Continuous summary statistic is binned.
  - Poisson for total number X multinomial over bins (equivalently product of Poisson distributions for each bin)
- Expected bin counts (Poisson rate) often comes from simulation

Usually modeled as a mixture of signal + multiple background processes

- Mixture coefficient on signal often parameter of interest
- Mixture coefficients for background components often uncertain and promoted to nuisance parameters

Nuisance parameters are introduced to parametrize uncertain aspects of simulation (e.g. calibration constants or parameters of underlying physics model)

- Large simulated samples are produced for systematic variations
- Fill corresponding histograms for those systematic variations
- Interpolate between these to create continuously parametrized model



HistFactory specification defines model exactly [CERN-OPEN-2012-016]

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

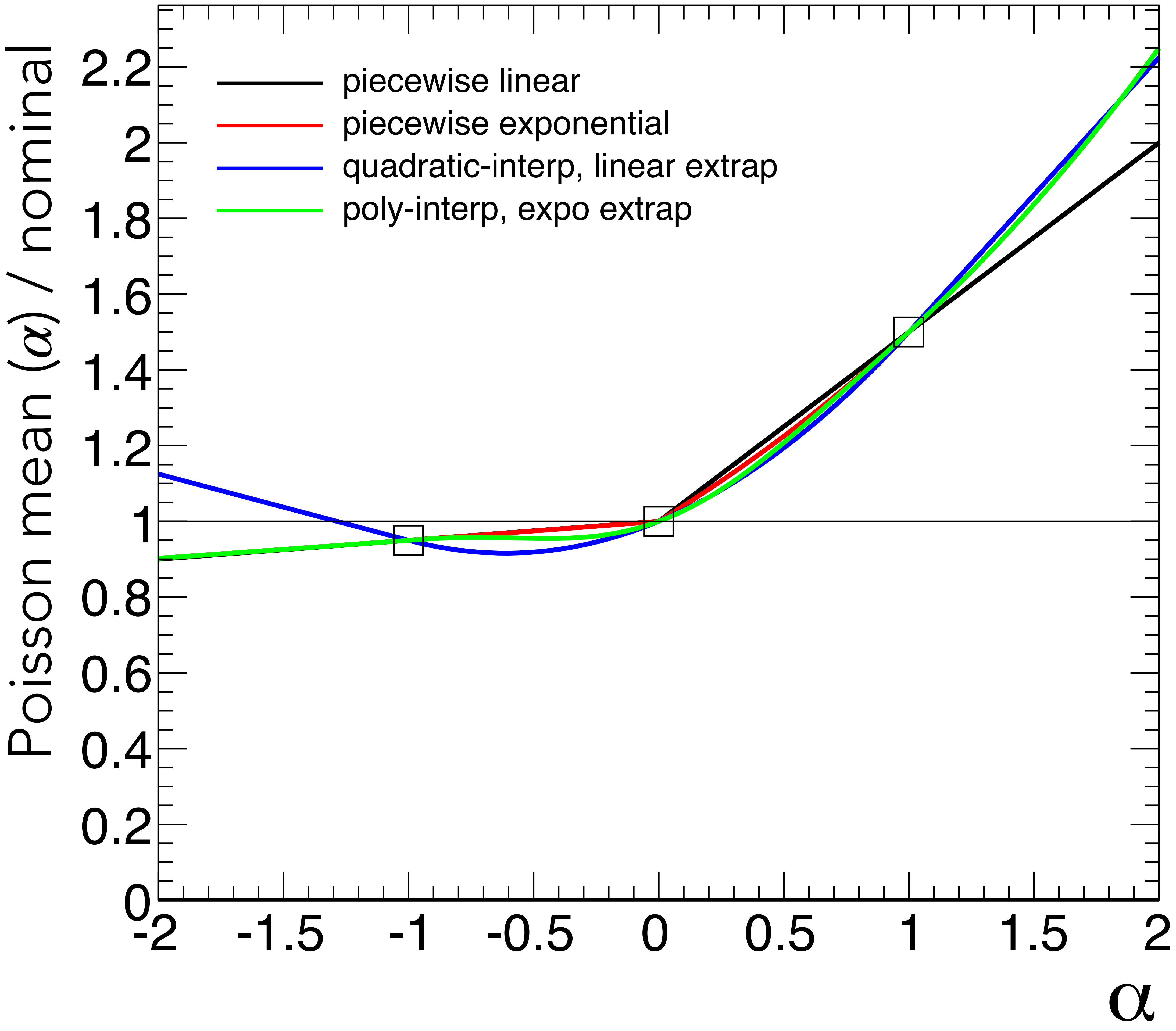
Current approach to histogram-based modeling

Input to the interpolation algorithms take

- Nominal sample
- $\pm 1\sigma$ one-at-a-time Systematic variations

The most widely used approach to modeling systematics **assumes that the effects of different systematics factorize**

Recognized as a weak point for ~10 years



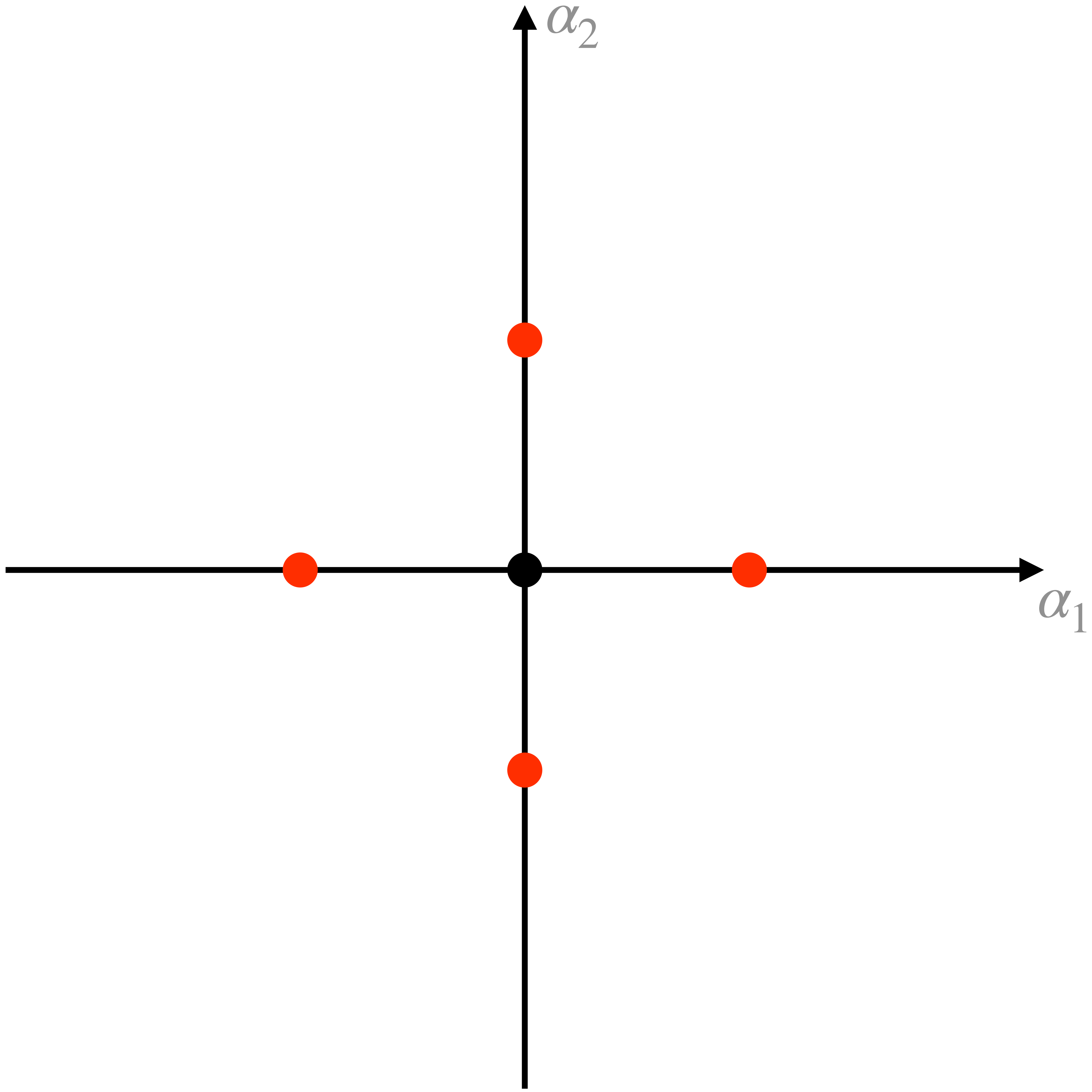
Current approach to histogram-based modeling

Input to the interpolation algorithms take

- Nominal sample
- $\pm 1\sigma$ one-at-a-time Systematic variations

The most widely used approach to modeling systematics **assumes that the effects of different systematics factorize**

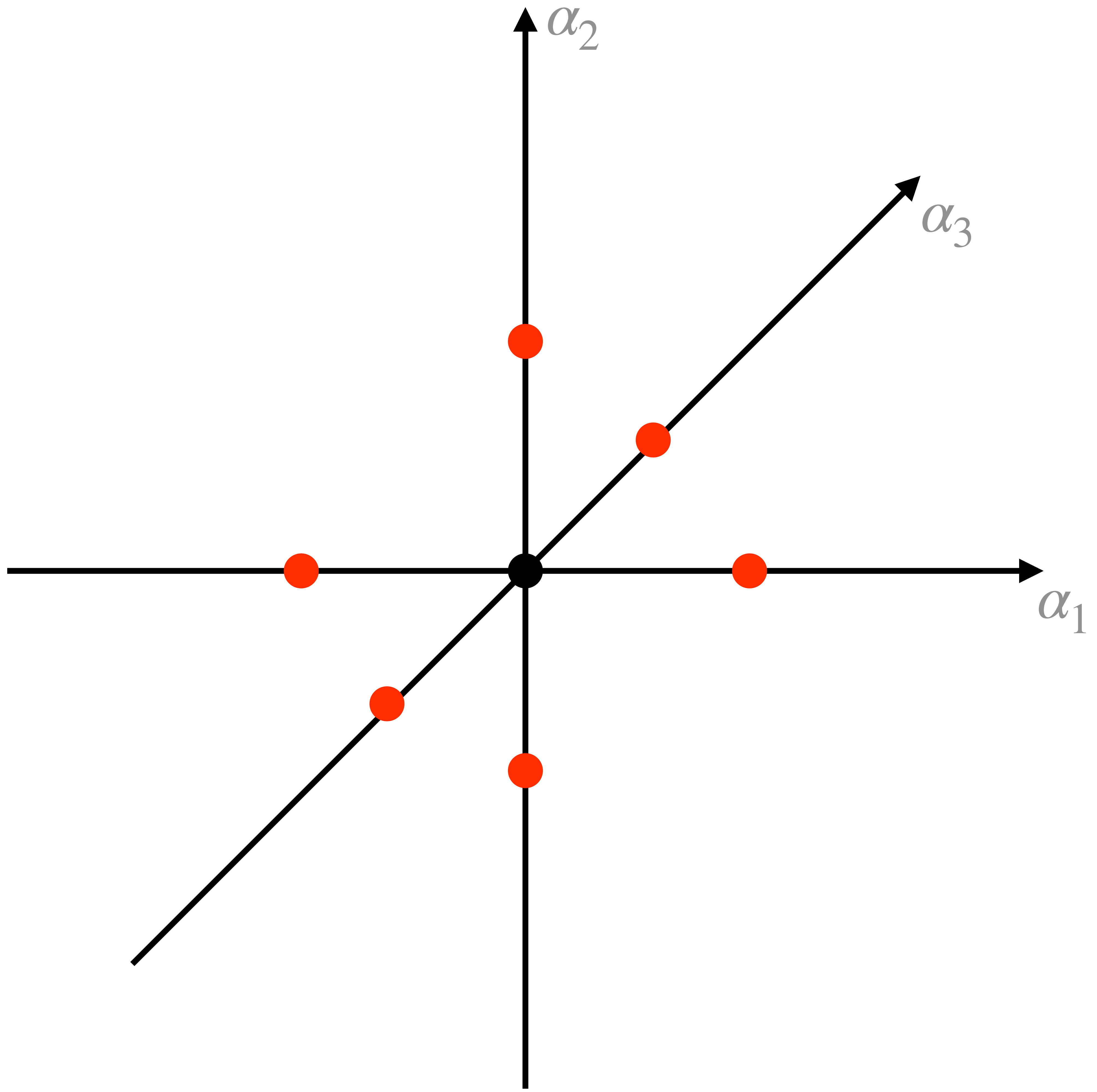
Recognized as a weak point for ~10 years



Current approach to histogram-based modeling

Input to the interpolation algorithms take

- Nominal sample
- $\pm 1\sigma$ one-at-a-time Systematic variations



The most widely used approach to modeling systematics **assumes that the effects of different systematics factorize**

Recognized as a weak point for ~10 years

Using ML to improve the treatment of systematics in traditional binned template analysis

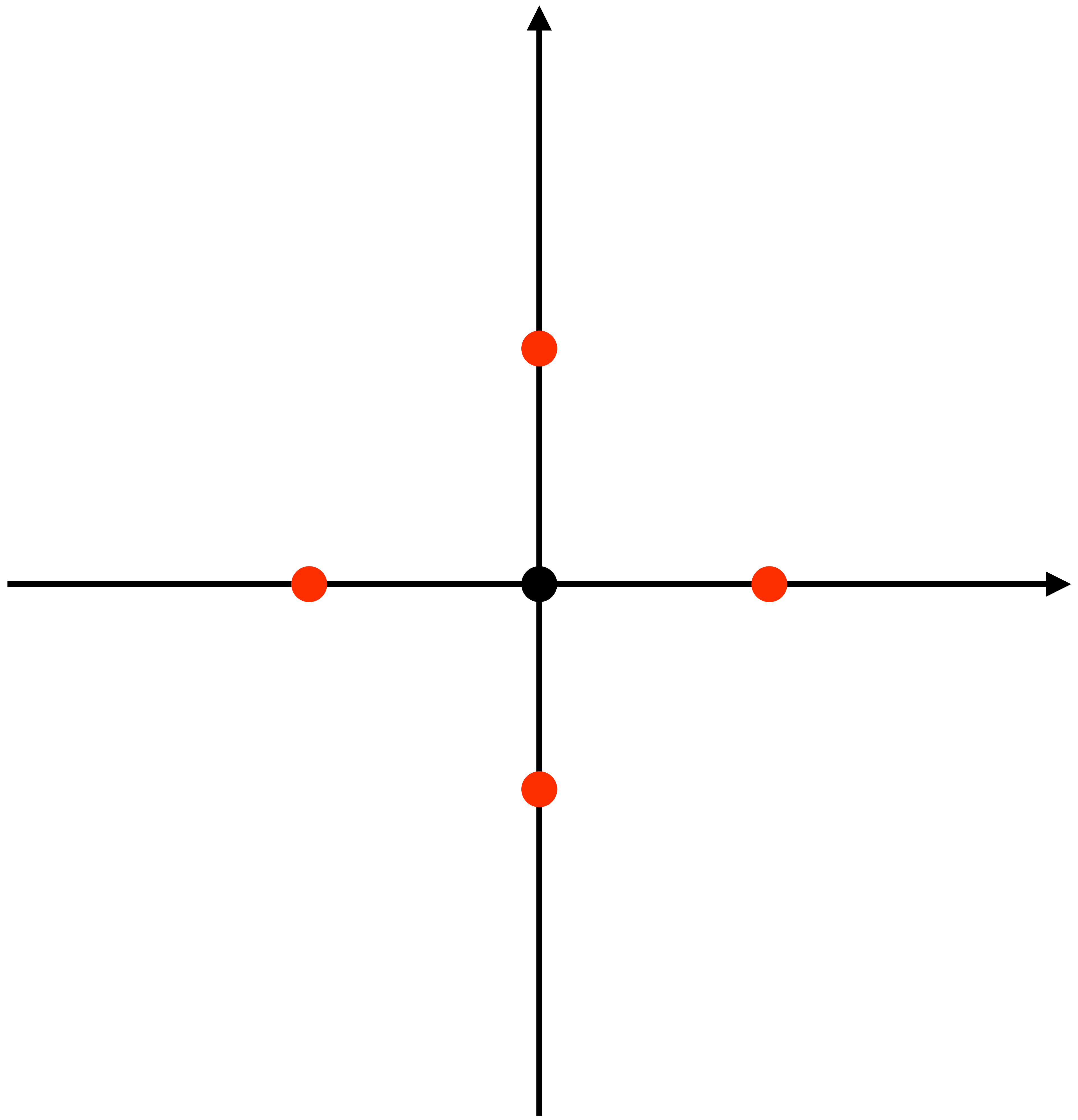
(new, unpublished work)

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Going beyond $\pm 1\sigma$ one-at-a-time variations

Nominal

$\pm 1\sigma$ one-at-a-time

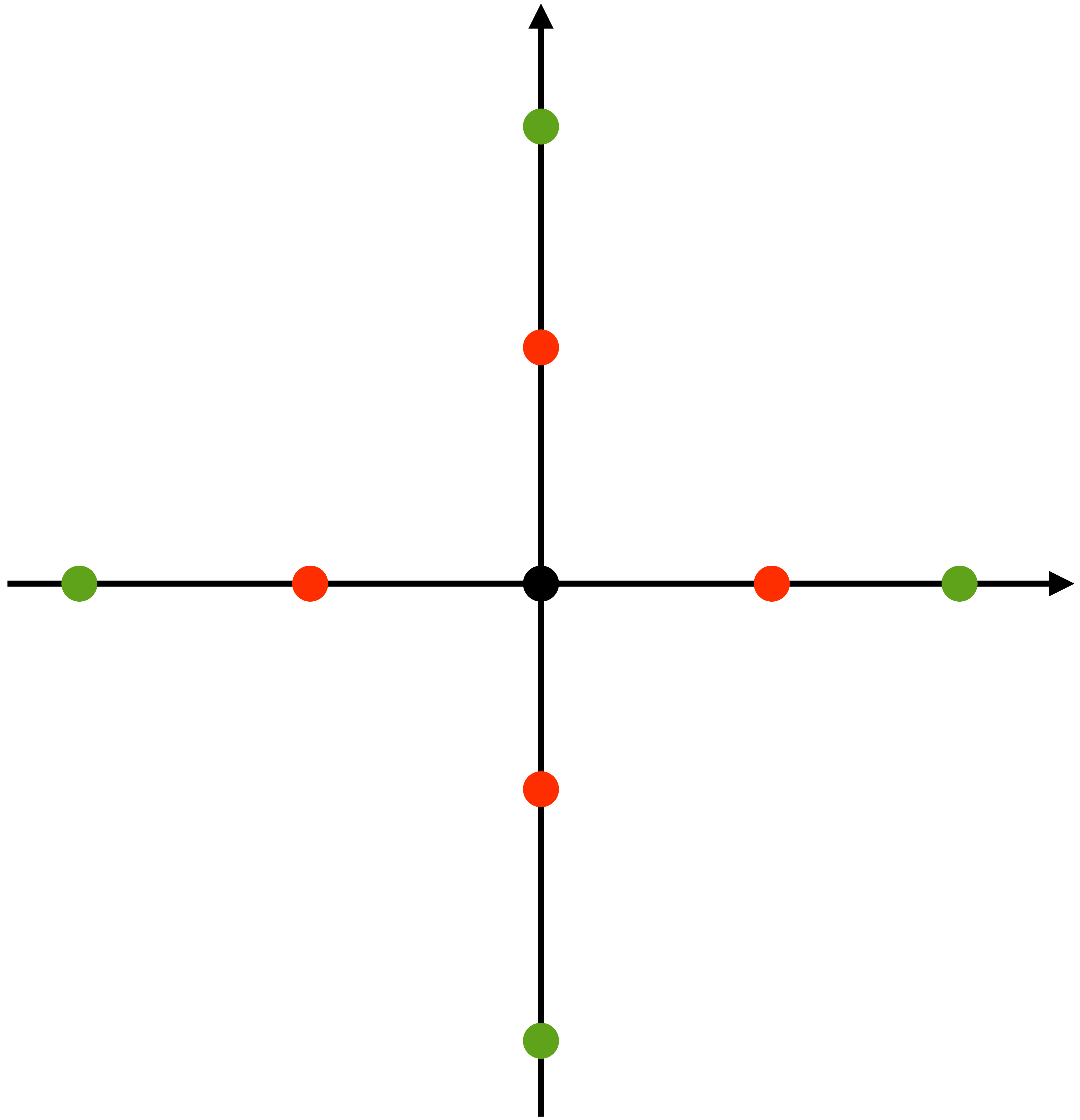


Going beyond $\pm 1\sigma$ one-at-a-time variations

Nominal

$\pm 1\sigma$ one-at-a-time

$\pm 2\sigma$ one-at-a-time



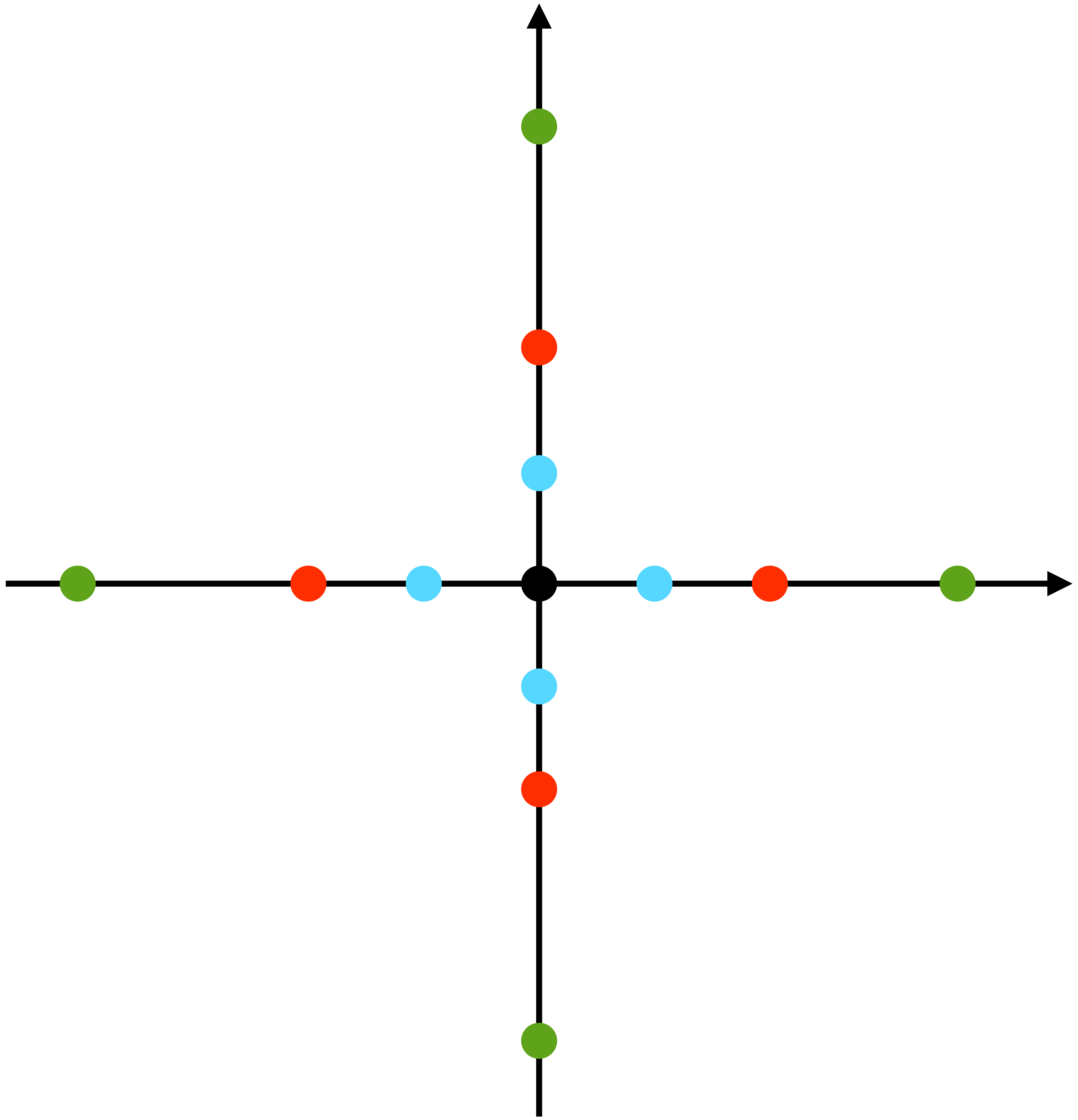
Going beyond $\pm 1\sigma$ one-at-a-time variations

Nominal

$\pm 1\sigma$ one-at-a-time

$\pm 2\sigma$ one-at-a-time

$\pm \frac{1}{2}\sigma$ one-at-a-time



Going beyond $\pm 1\sigma$ one-at-a-time variations

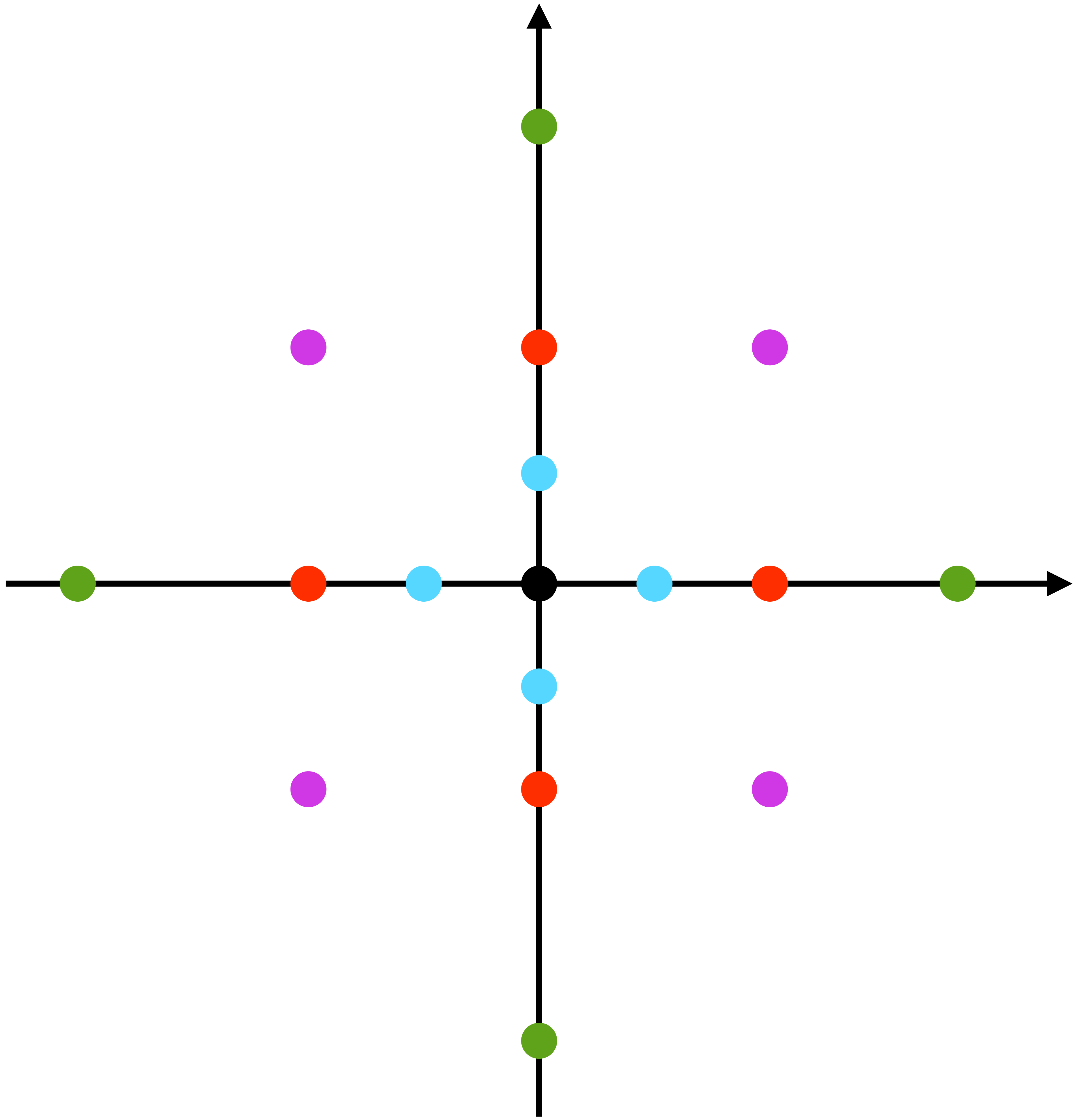
Nominal

$\pm 1\sigma$ one-at-a-time

$\pm 2\sigma$ one-at-a-time

$\pm$½σ one-at-a-time

$\pm 1\sigma$ simultaneously variation



Going beyond ±1σ one-at-a-time variations

Nominal

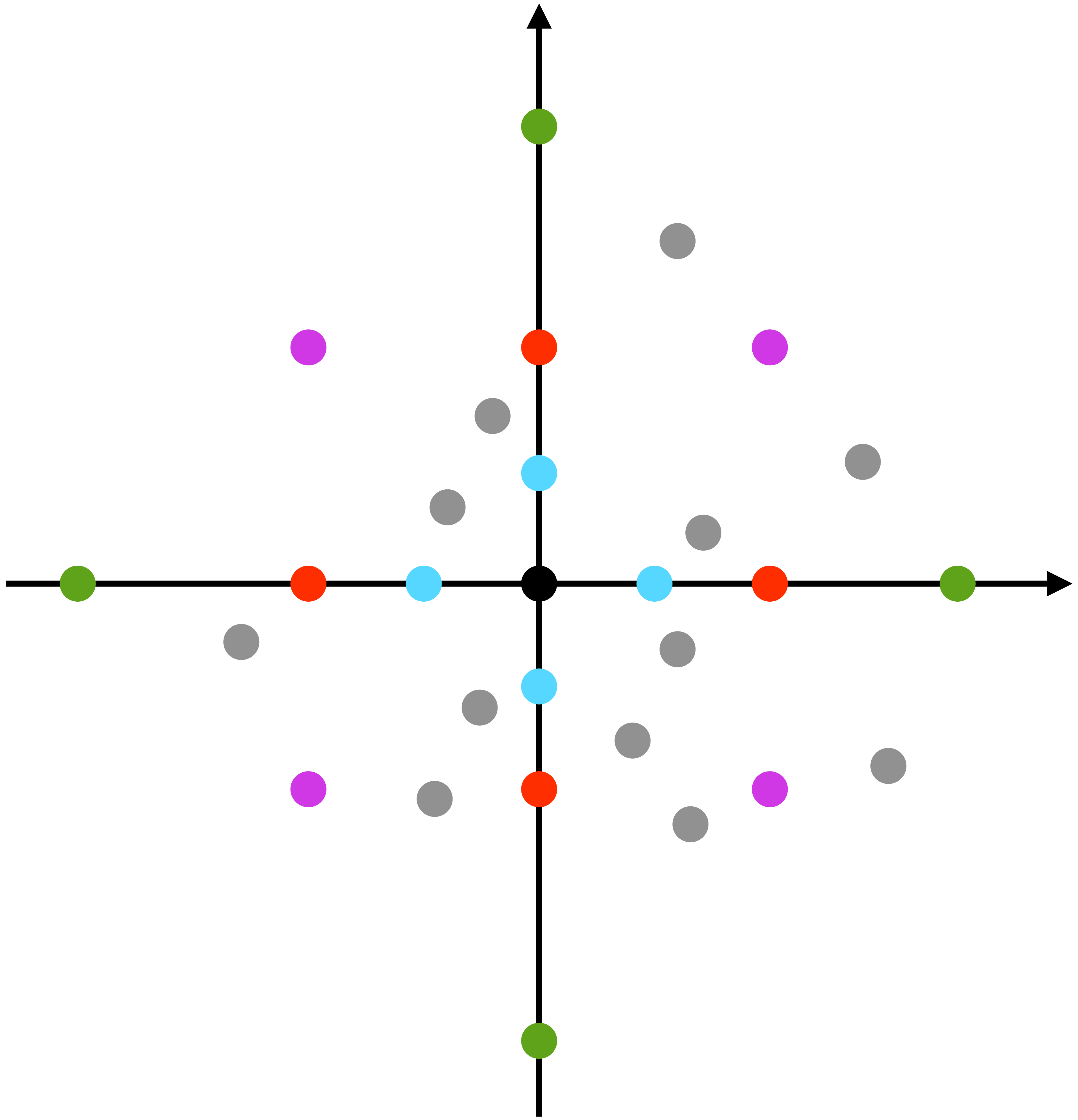
±1σ one-at-a-time

±2σ one-at-a-time

±½σ one-at-a-time

±1σ simultaneously variation

Arbitrary



Going beyond $\pm 1\sigma$ one-at-a-time variations

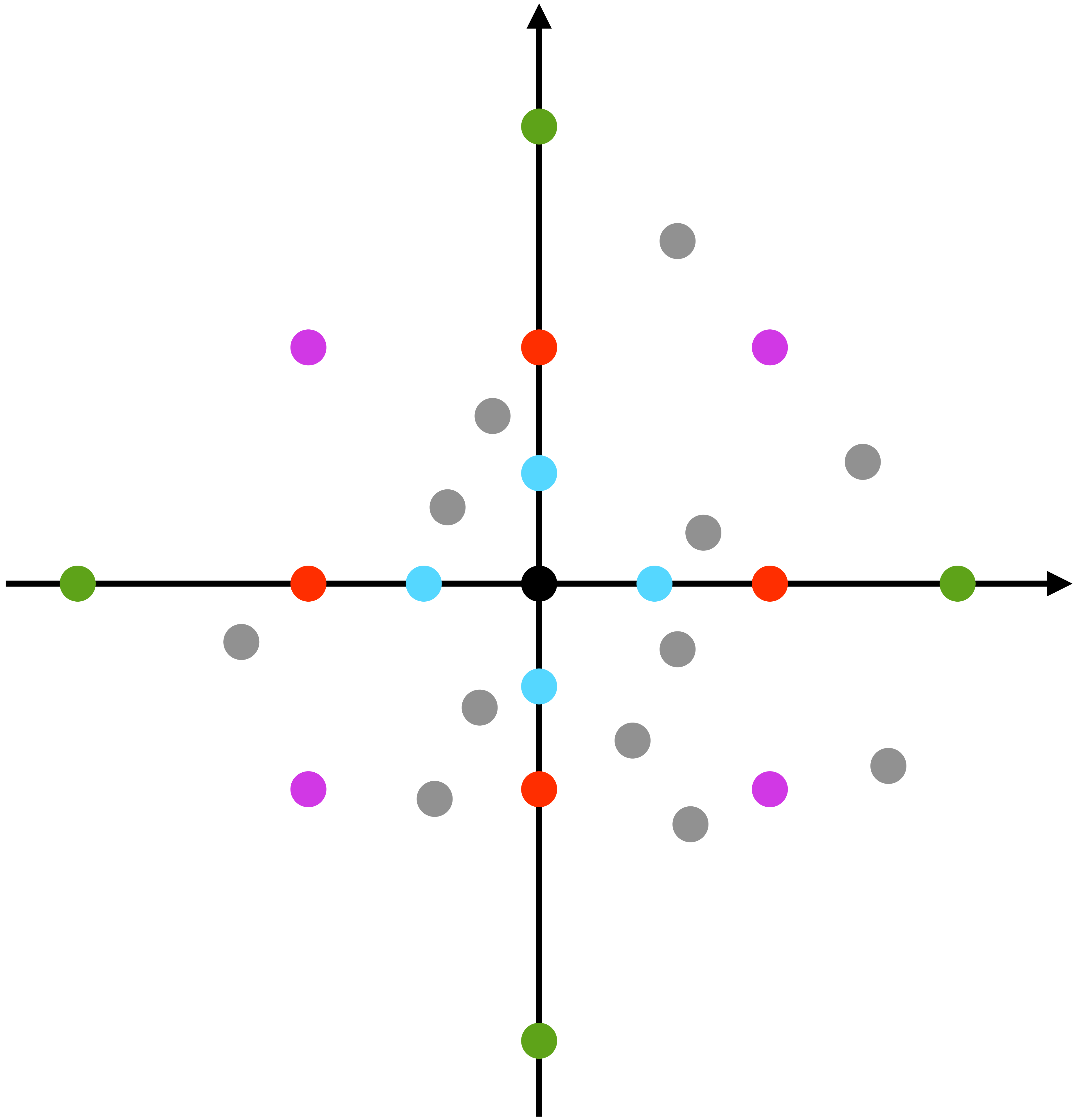
We want a way to interpolate with arbitrary samples (in arbitrary dimensions) that

- Will be smooth
- Sample efficient (few inputs)
- Trustworthy (no worries about training that doesn't converge)

Gaussian Processes are a natural choice

- They also provide a notion of uncertainty on the interpolation

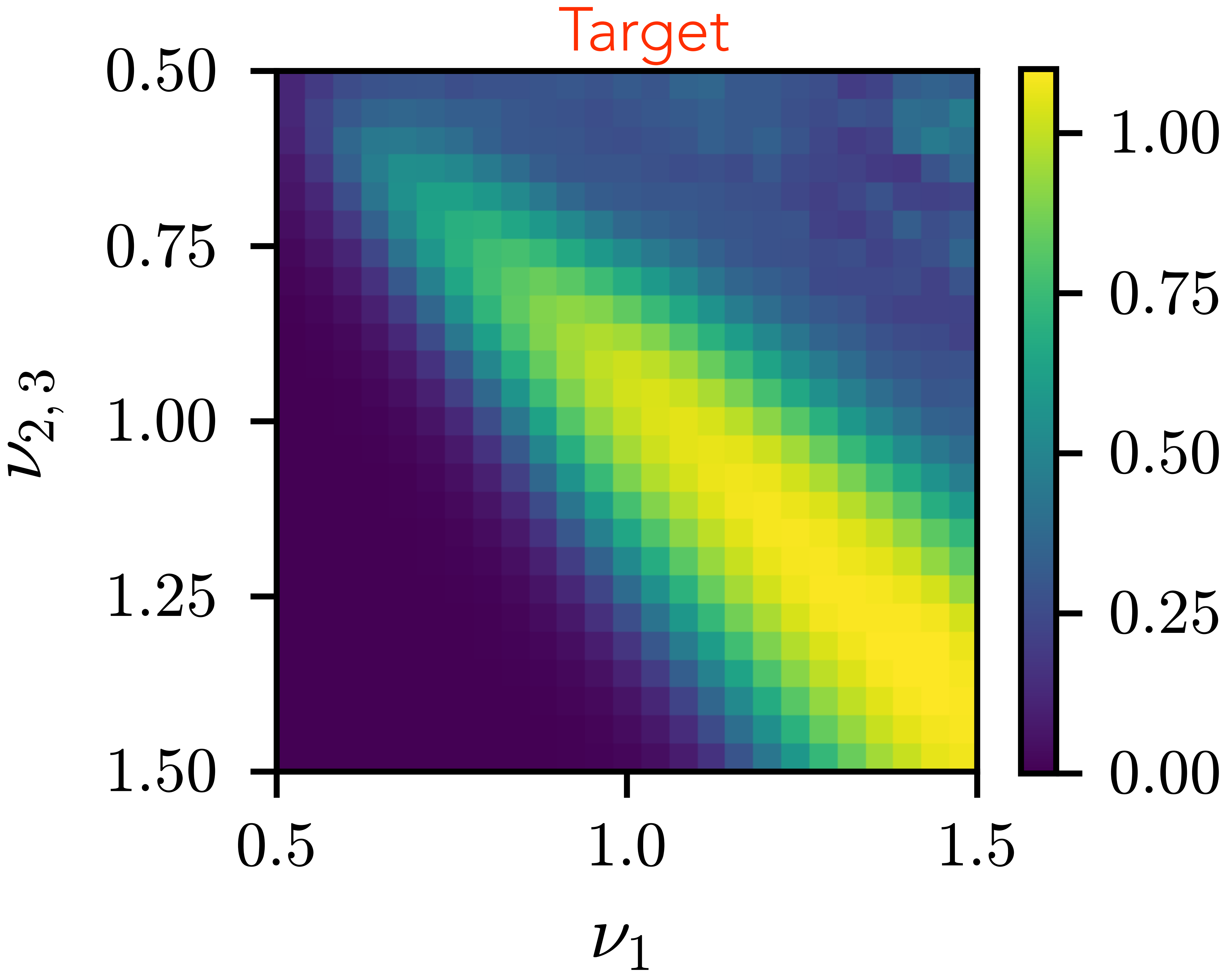
We still face the curse of dimensionality!

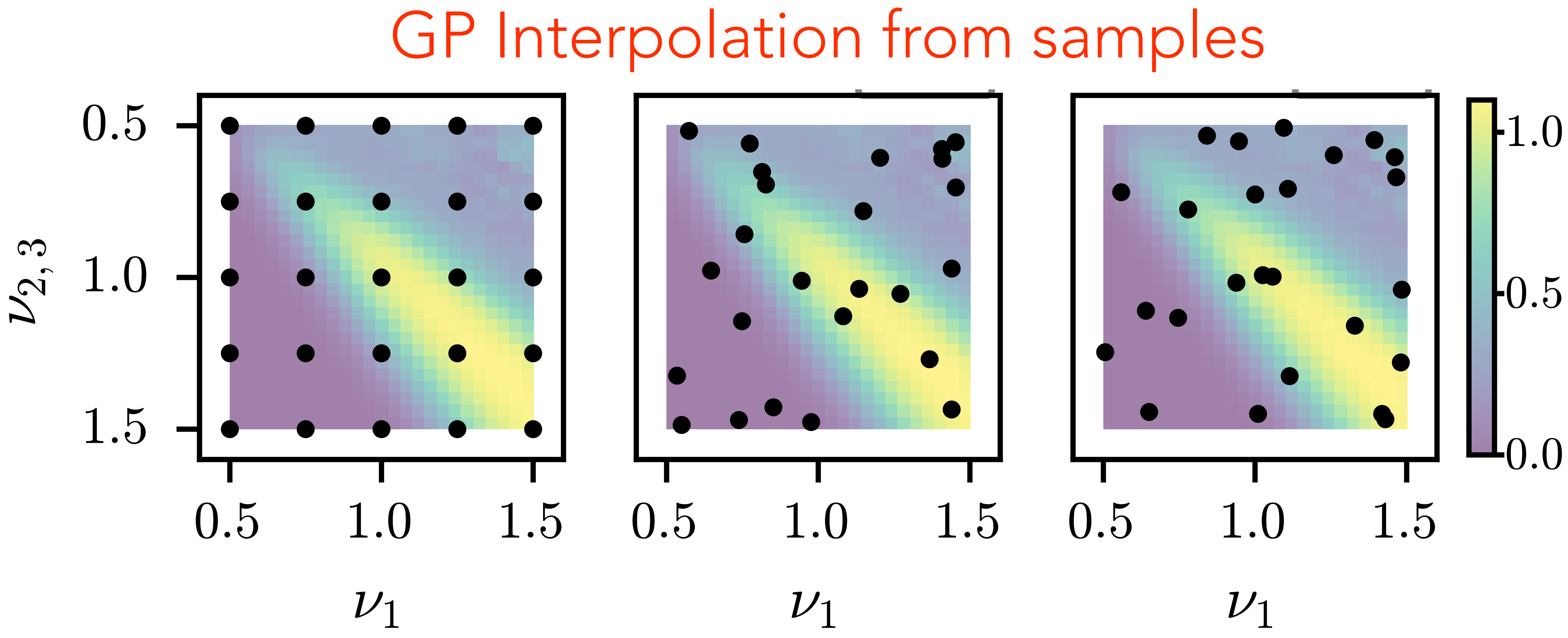


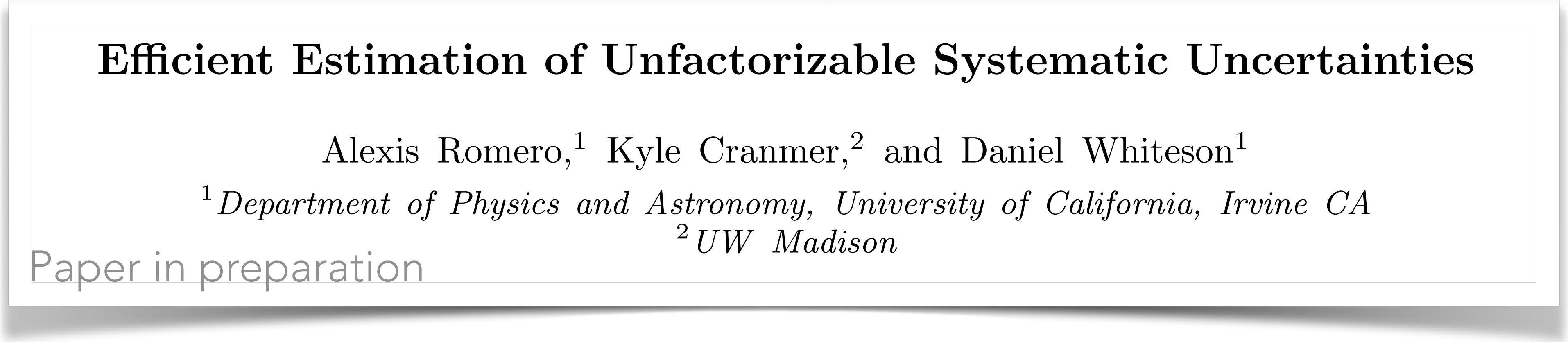
GPs on a 2D example that doesn’t factorize

Here are some examples of a GP fitting samples of a target efficiency function that doesn’t factorize

- **Physics:** efficiency of cut on MET. Systematics are jet energy scale uncertainties for low-pT and high-pT jets







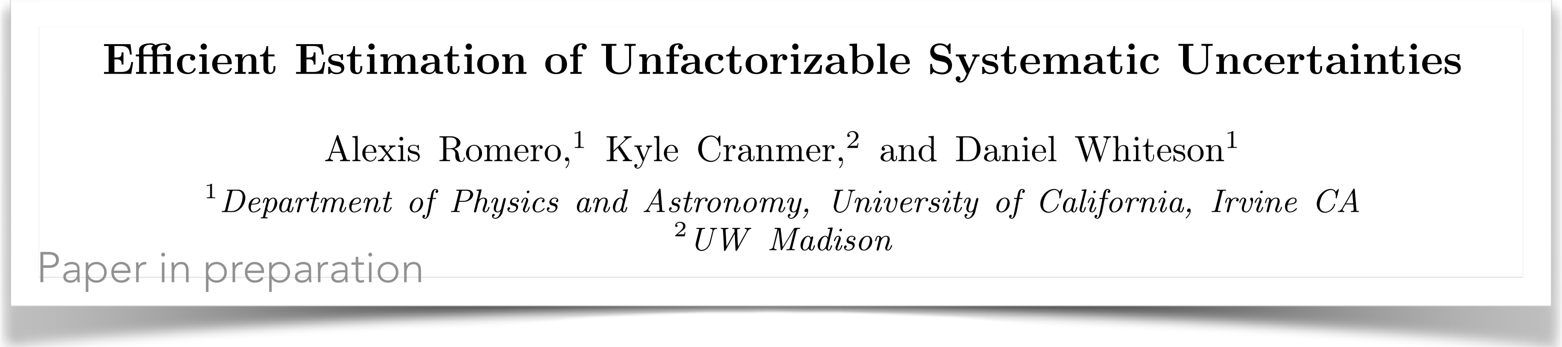
Efficient Estimation of Unfactorizable Systematic Uncertainties

Alexis Romero,[1] Kyle Cranmer,[2] and Daniel Whiteson[1]

[1] *Department of Physics and Astronomy, University of California, Irvine CA*
[2] *UW Madison*

Paper in preparation

GPs with Derivative observations



Efficient Estimation of Unfactorizable Systematic Uncertainties

Alexis Romero,[1] Kyle Cranmer,[2] and Daniel Whiteson[1]

[1] *Department of Physics and Astronomy, University of California, Irvine CA*
[2] *UW Madison*

Paper in preparation

Gaussian processes produce a smooth interpolation between data points and provide a notion of uncertainty on the interpolation

- If we in addition to y(x) pairs, we also have **derivative observations** dy(x)/dx, then the GP will converge more quickly (best fit interpolation changes and uncertainty of interpolation reduced)

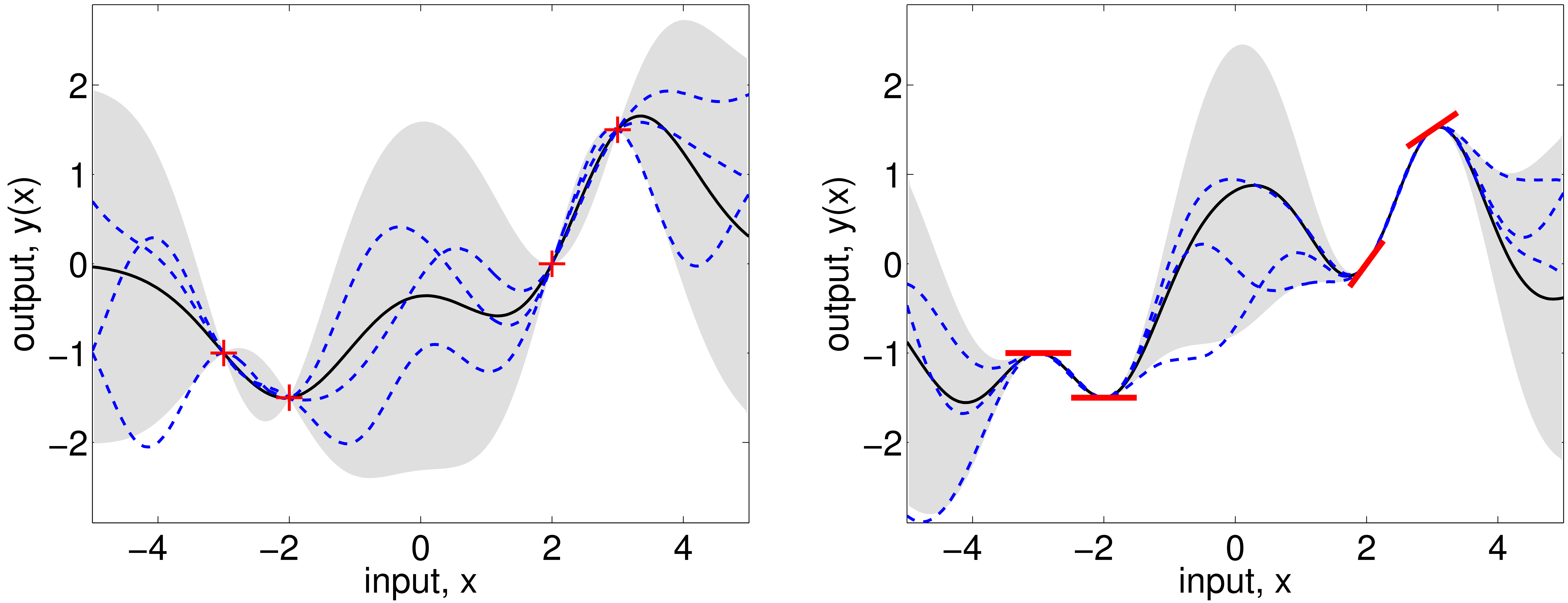
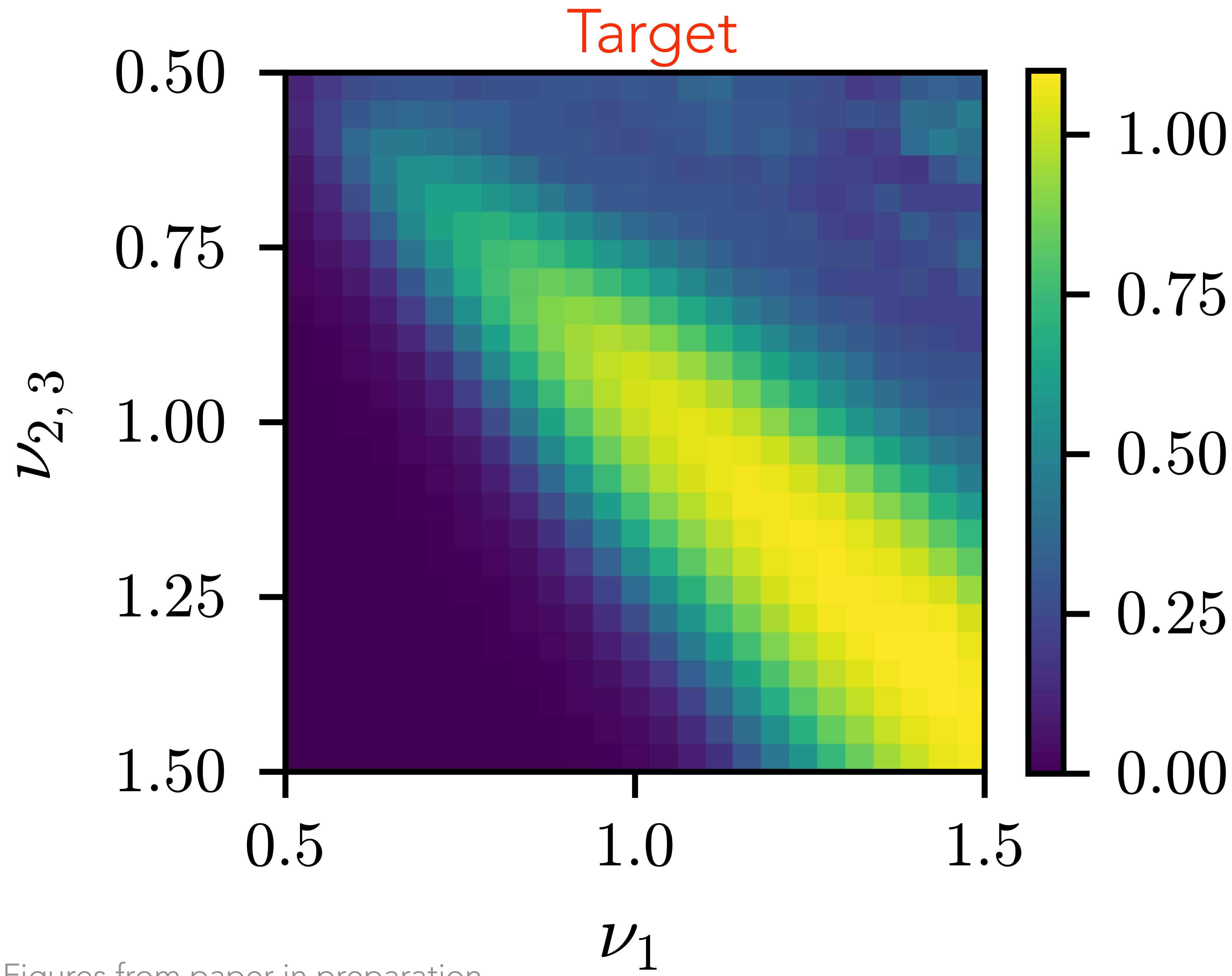
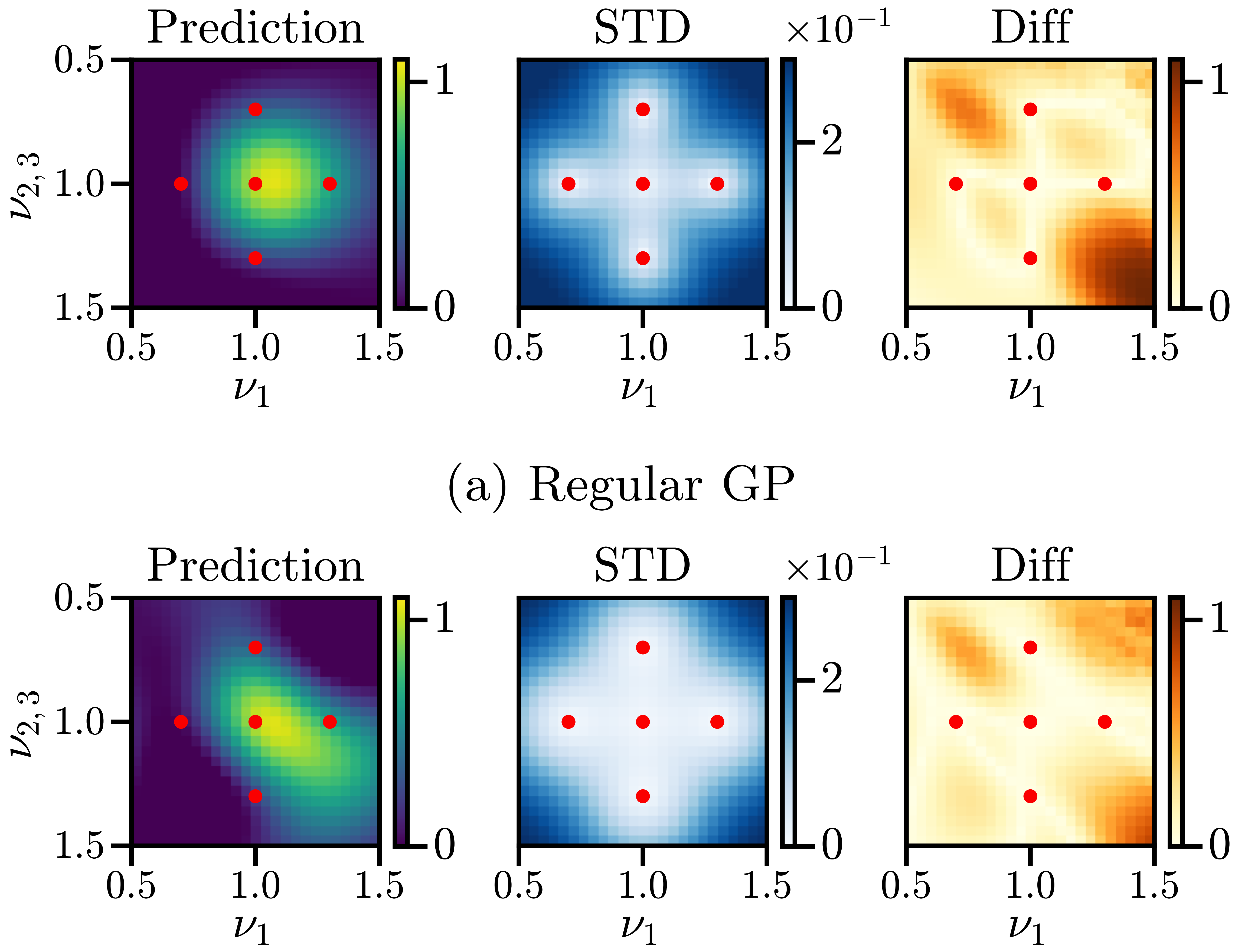


Figure from Rasmussen & Williams, Gaussian Processes for Machine Learning,

Derivative GPs on a 2D example that doesn't factorize

Even with the nominal and $\pm 1\sigma$ one-at-a-time variations, the derivative GP is able to capture the effect of simultaneous variation of the two nuisance parameters





(a) Regular GP

(b) Derivative GP

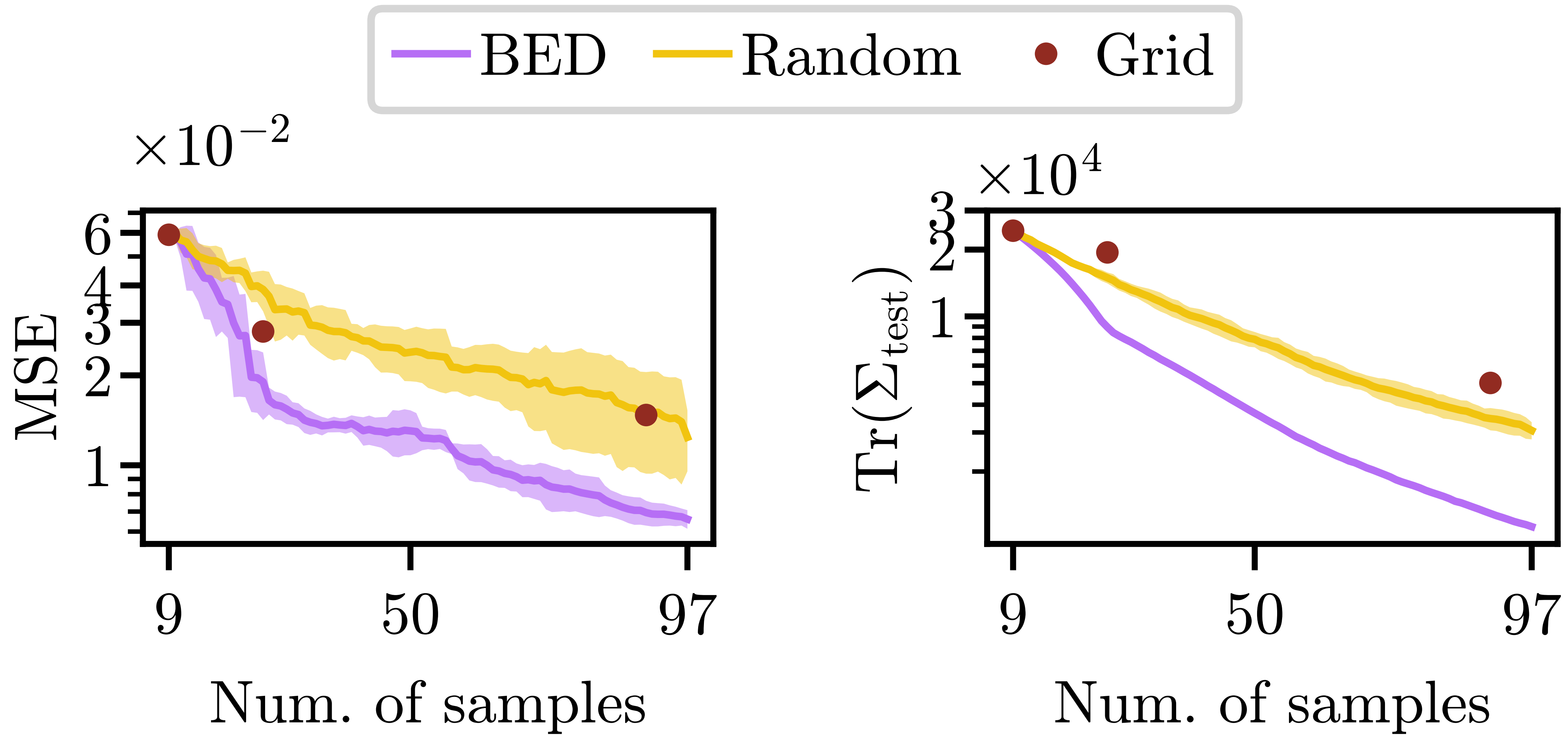
Figures from paper in preparation

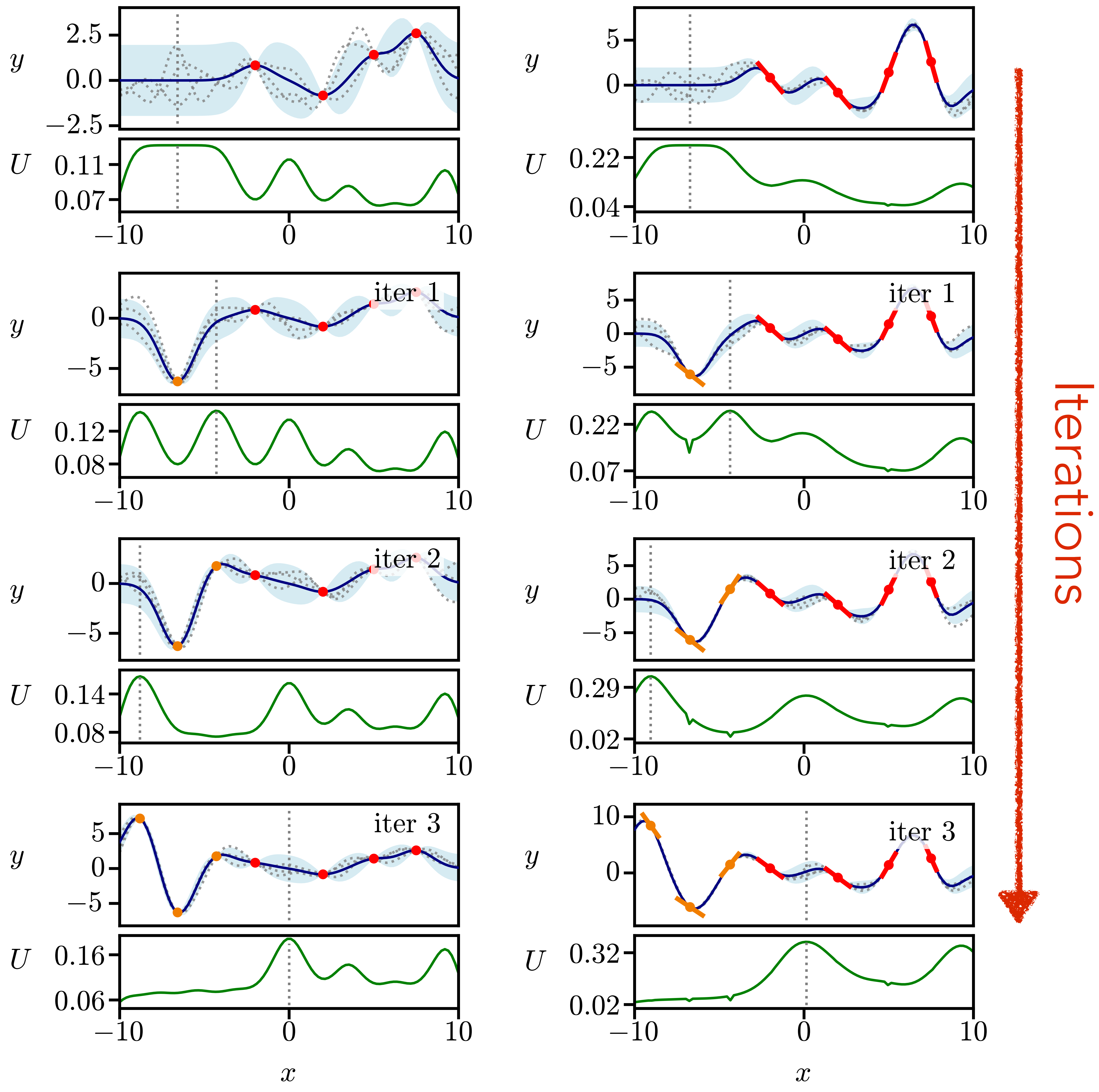
Active learning / Bayesian Experimental Design (BED)

We can also be smart about where we sample in nuisance parameter space

- Sample in places where uncertainty is large. Iterate.

Active learning (BED) more efficient





(a) Regular GP

(b) Derivative GP

Figures from paper in preparation

Traditional use of ML for searches

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / “non-parametric” (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

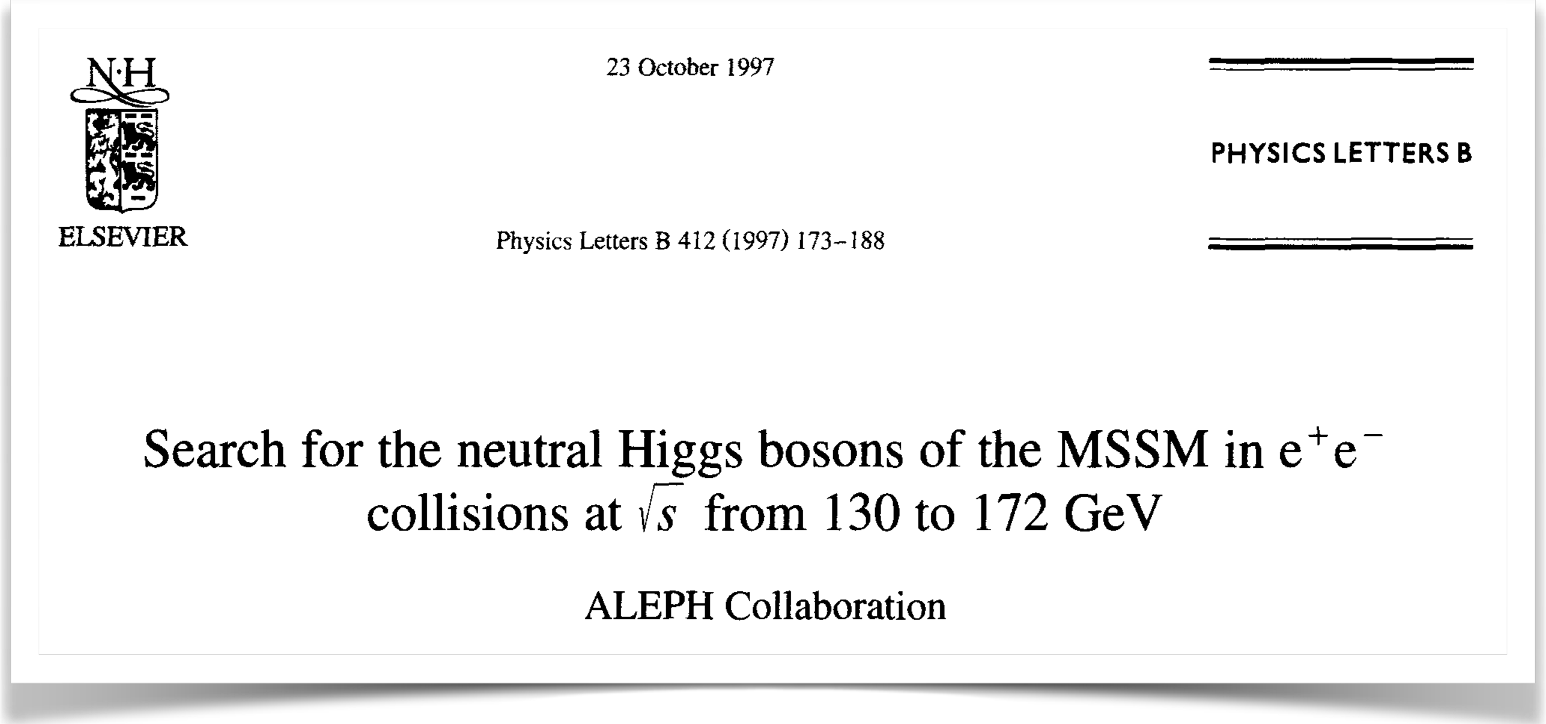
Traditional use of ML for searches

Most searches for new particles are cast into a hypothesis testing framework

- Likelihood ratio is well motivated, but the likelihood for high dimensional, low-level observations from simulation is intractable

Instead of designing a summary statistic by hand, can **use a neural network to learn a more powerful summary statistic** (that approximates likelihood ratio)

- From that point on, the NN output is treated like any other summary statistic in the down-stream statistical analysis



N·H
ELSEVIER

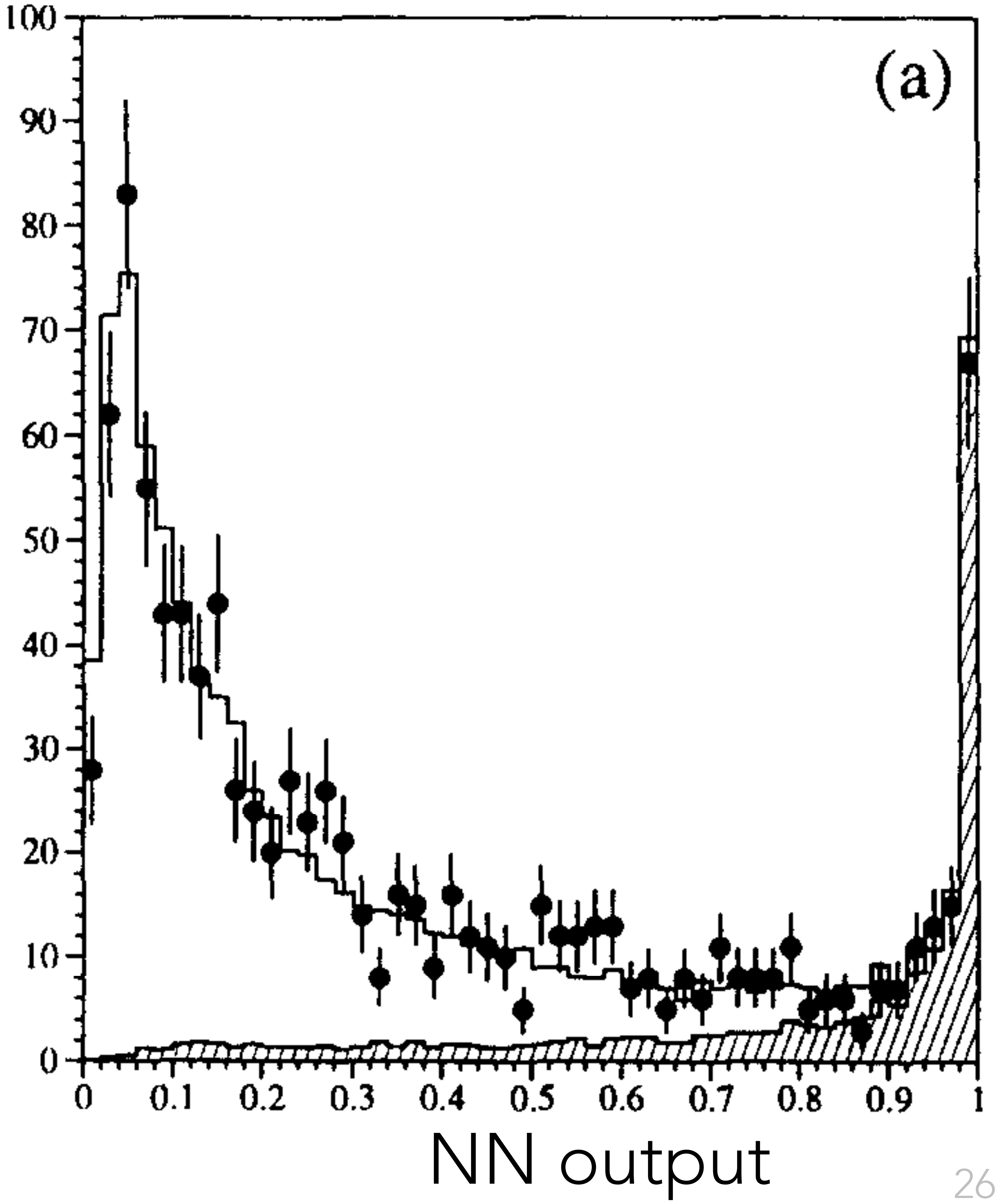
23 October 1997

PHYSICS LETTERS B

Physics Letters B 412 (1997) 173–188

Search for the neutral Higgs bosons of the MSSM in e^+e^- collisions at $\sqrt{s}$ from 130 to 172 GeV

ALEPH Collaboration



Incorporating Systematics

We want to take advantage of the power of machine learning, but we need to incorporate systematic uncertainties.

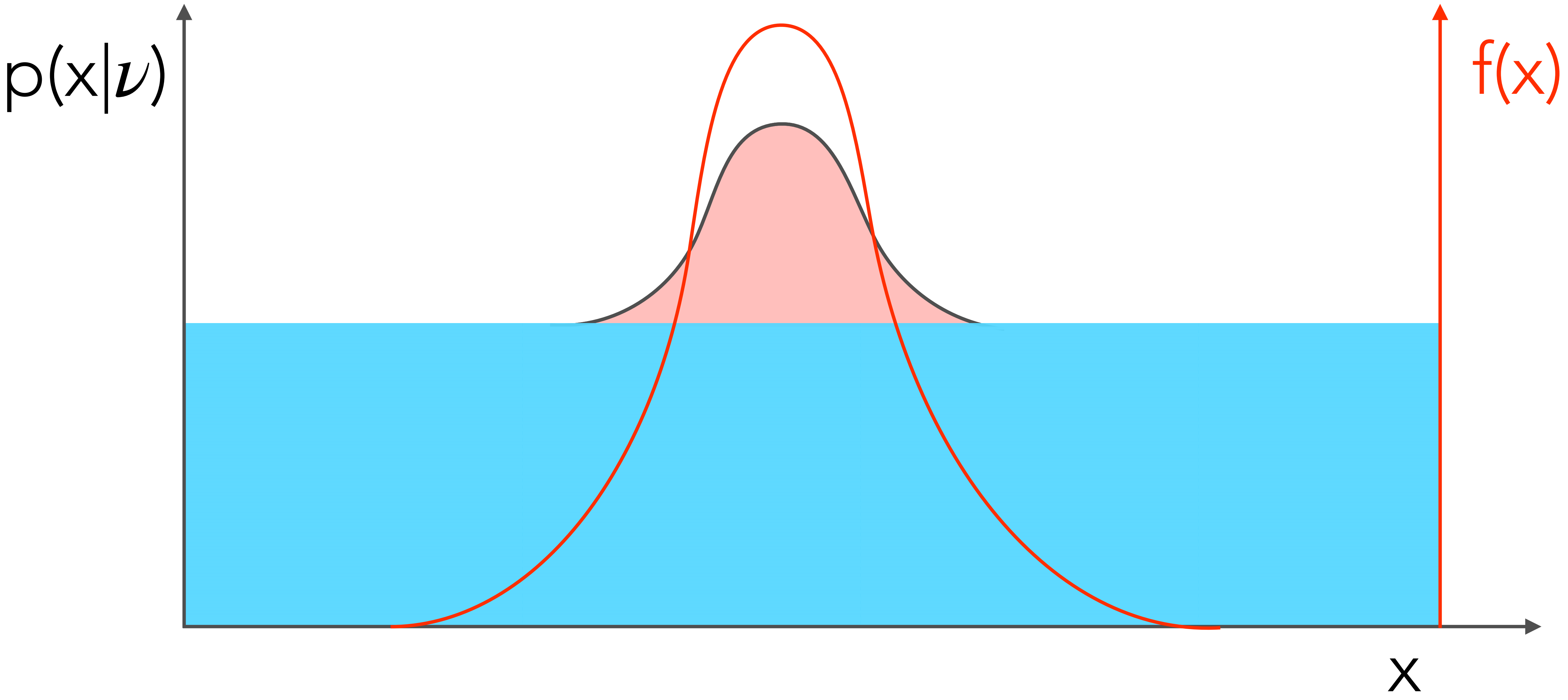
Two notions of "incorporate":

- **Don't be wrong:** view analysis chain as fixed and propagate systematic uncertainty through it.
  - e.g. control rate of type-I error in the presence of nuisance parameters
- **Try to be "optimal":** adjust the training of ML components so that the analysis is sensitive after accounting for systematics
  - e.g. minimize rate of type-II error / maximize power

Fixed classifier is not optimal

Imagine a simple example of bump on flat background

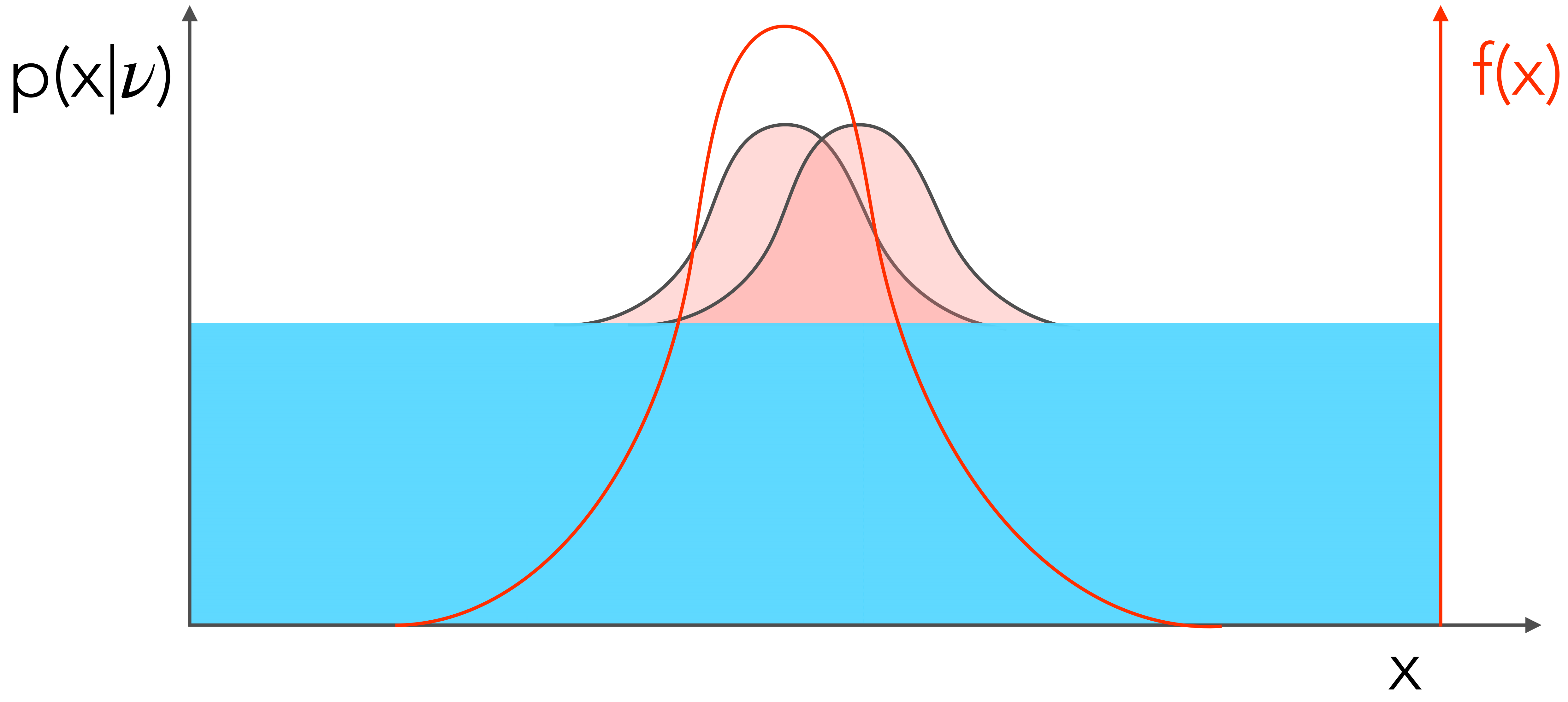
- train on nominal samples with $\nu = \nu_0$ to obtain fixed classifier $f(x)$
- uncertainty in ν modifies location and width of peak
- the classifier not optimal for $\nu \neq \nu_0$, **but we can propagate uncertainty**



Fixed classifier is not optimal

Imagine a simple example of bump on flat background

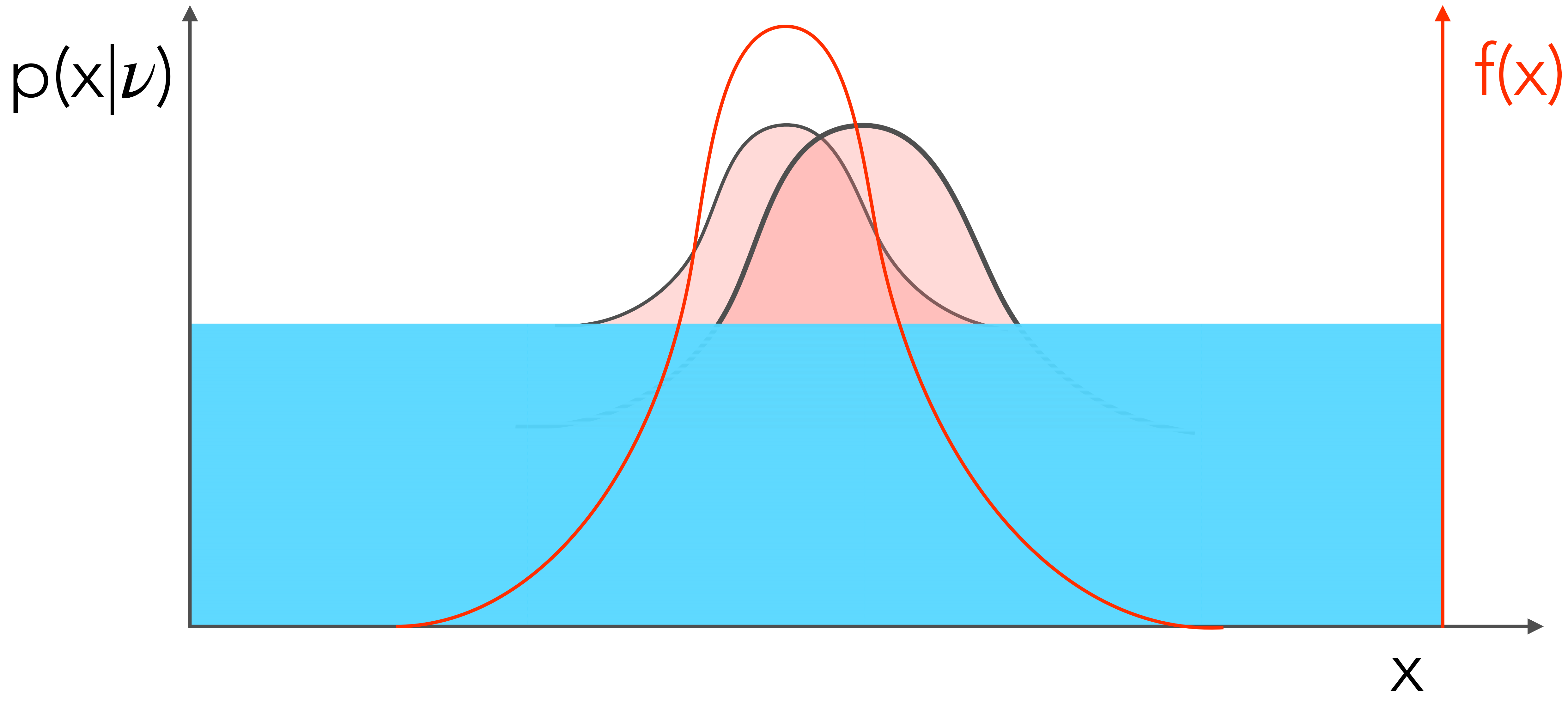
- train on nominal samples with $\nu = \nu_0$ to obtain fixed classifier $f(x)$
- uncertainty in ν modifies location and width of peak
- the classifier not optimal for $\nu \neq \nu_0$, **but we can propagate uncertainty**



Fixed classifier is not optimal

Imagine a simple example of bump on flat background

- train on nominal samples with $\nu = \nu_0$ to obtain fixed classifier $f(x)$
- uncertainty in ν modifies location and width of peak
- the classifier not optimal for $\nu \neq \nu_0$, **but we can propagate uncertainty**

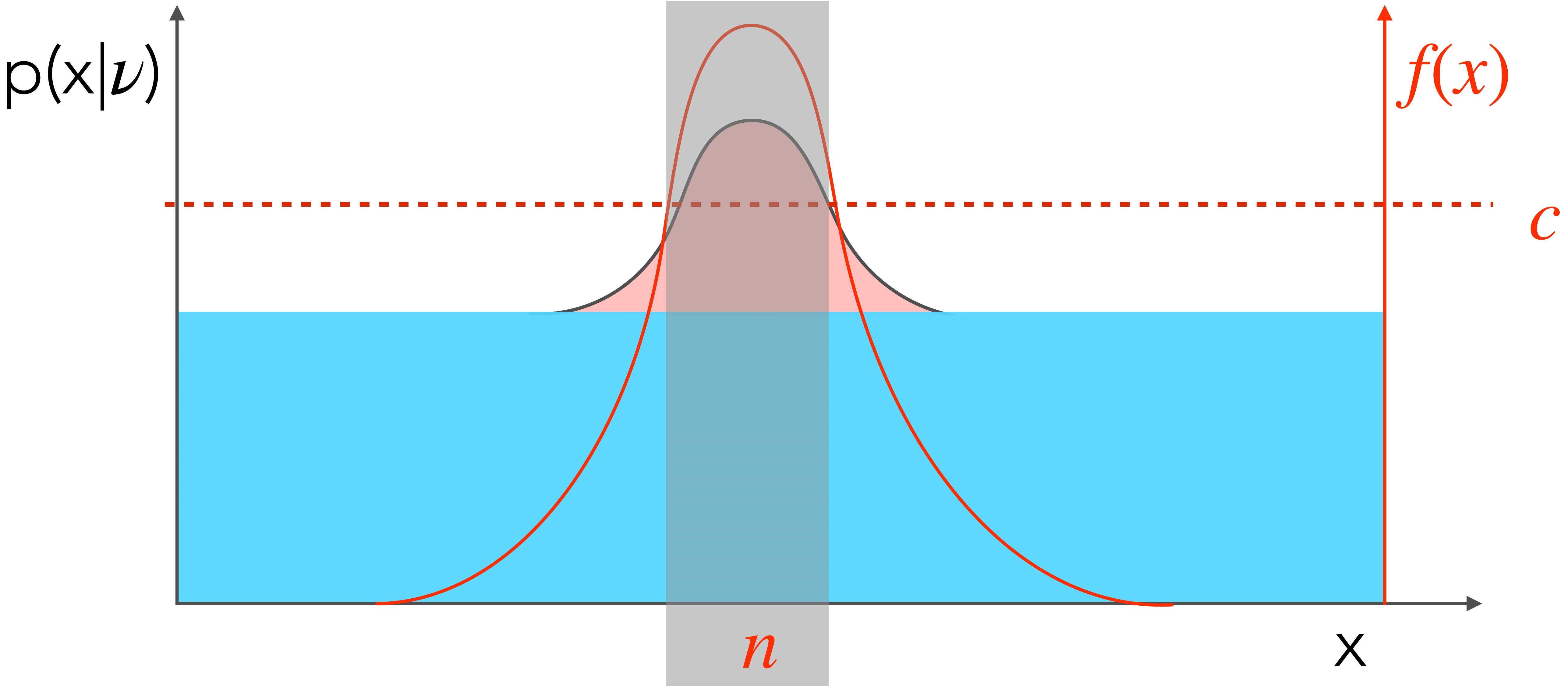


Propagation of uncertainty

One might form a statistical model for the number of events n that have $f(x) > c$

$$\epsilon_{\mathrm{sig}}(\nu) = \int_c^\infty p(f\,|\,y = 1,\nu)\mathrm{d}f \qquad \epsilon_{\mathrm{bkg}}(\nu) = \int_c^\infty p(f\,|\,y = 0,\nu)\mathrm{d}f$$

$$p(n, a\,|\,\mu, \nu) = \mathrm{Pois}(n\,|\,\mu\epsilon_{\mathrm{sig}}(\nu)s + \epsilon_{\mathrm{bkg}}(\nu)b)p(a\,|\,\nu)$$

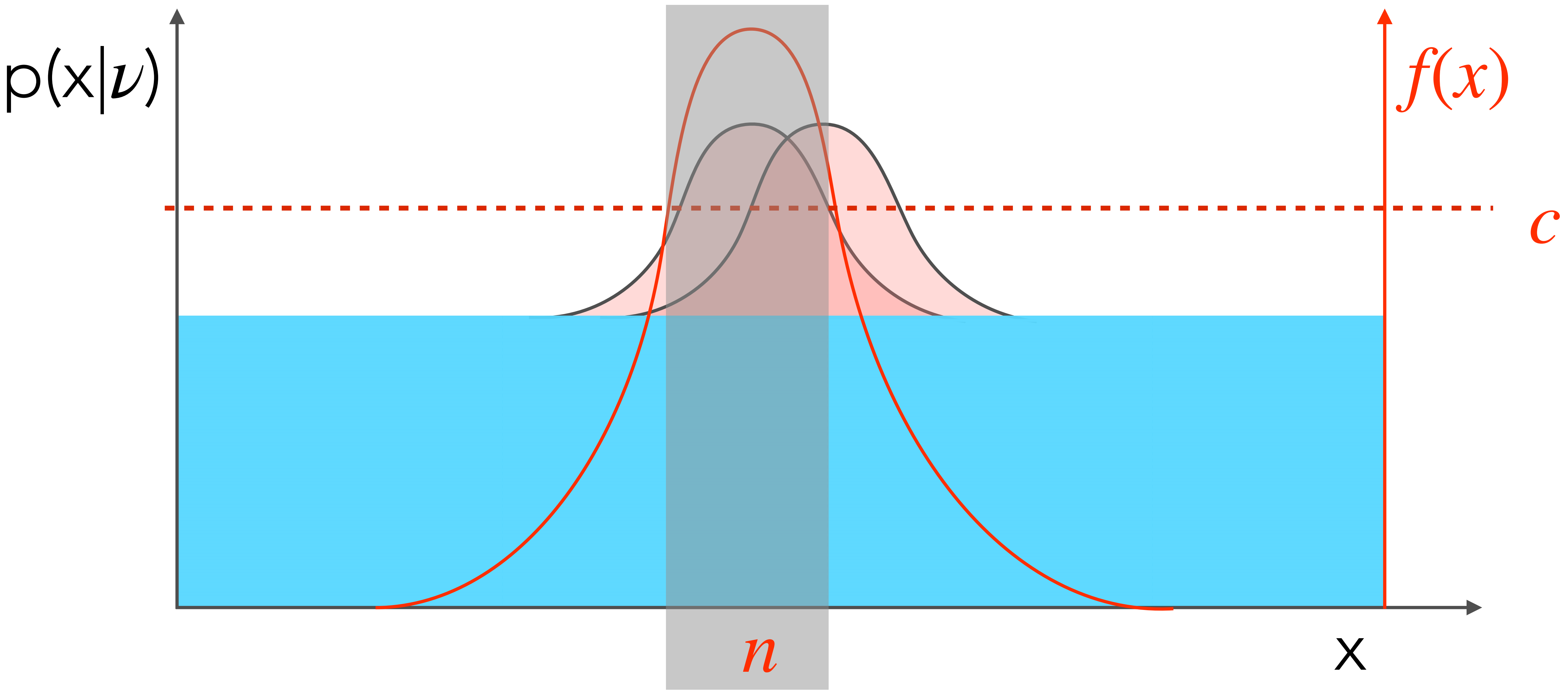


Propagation of uncertainty

One might form a statistical model for the number of events n that have $f(x) > c$

$$\epsilon_{\mathrm{sig}}(\nu) = \int_c^\infty p(f \,|\, y = 1, \nu)\mathrm{d}f \qquad \epsilon_{\mathrm{bkg}}(\nu) = \int_c^\infty p(f \,|\, y = 0, \nu)\mathrm{d}f$$

$$p(n, a \,|\, \mu, \nu) = \mathrm{Pois}(n \,|\, \mu\epsilon_{\mathrm{sig}}(\nu)s + \epsilon_{\mathrm{bkg}}(\nu)b)p(a \,|\, \nu)$$

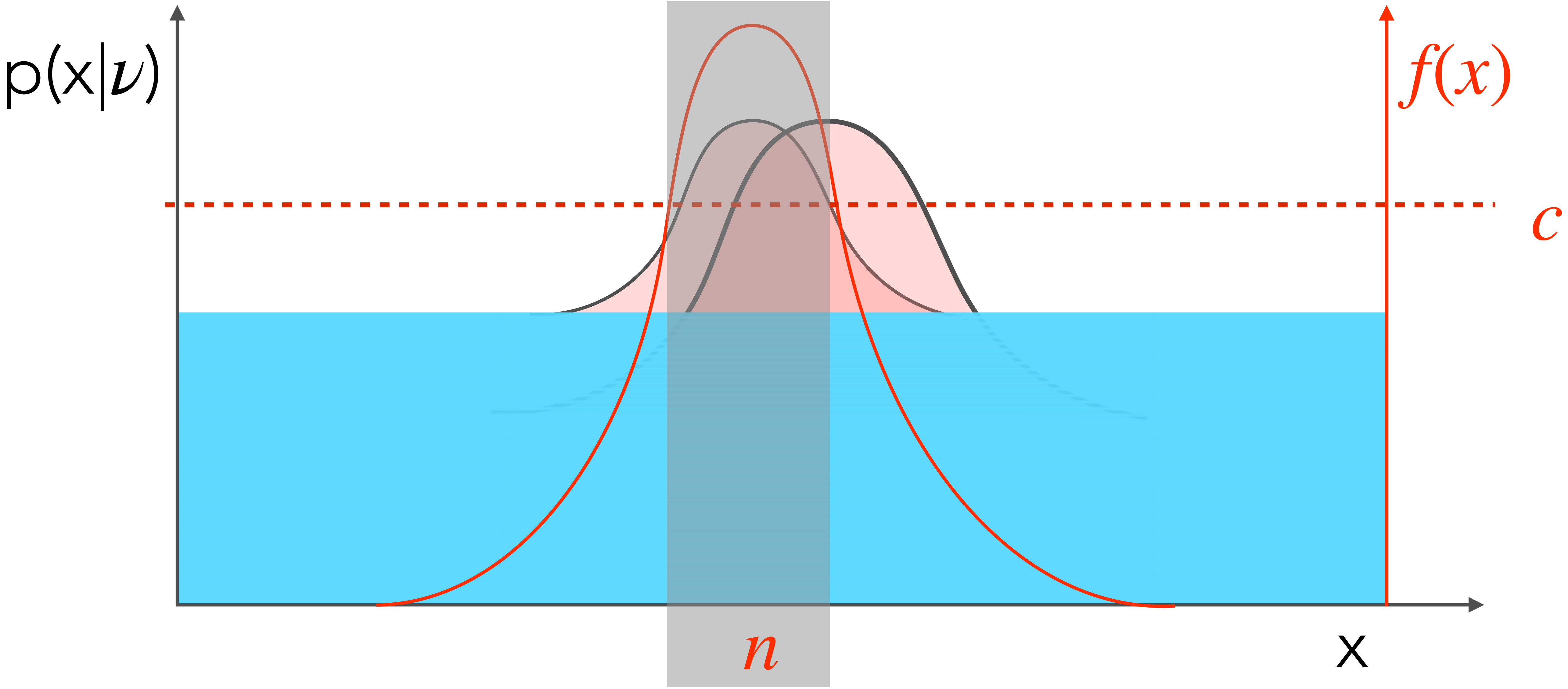


Propagation of uncertainty

One might form a statistical model for the number of events n that have $f(x) > c$

$$\epsilon_{\text{sig}}(\nu) = \int_c^\infty p(f\,|\,y=1,\nu)\mathrm{d}f \qquad \epsilon_{\text{bkg}}(\nu) = \int_c^\infty p(f\,|\,y=0,\nu)\mathrm{d}f$$

$$p(n, a\,|\,\mu, \nu) = \mathrm{Pois}(n\,|\,\mu\epsilon_{\text{sig}}(\nu)s + \epsilon_{\text{bkg}}(\nu)b)p(a\,|\,\nu)$$



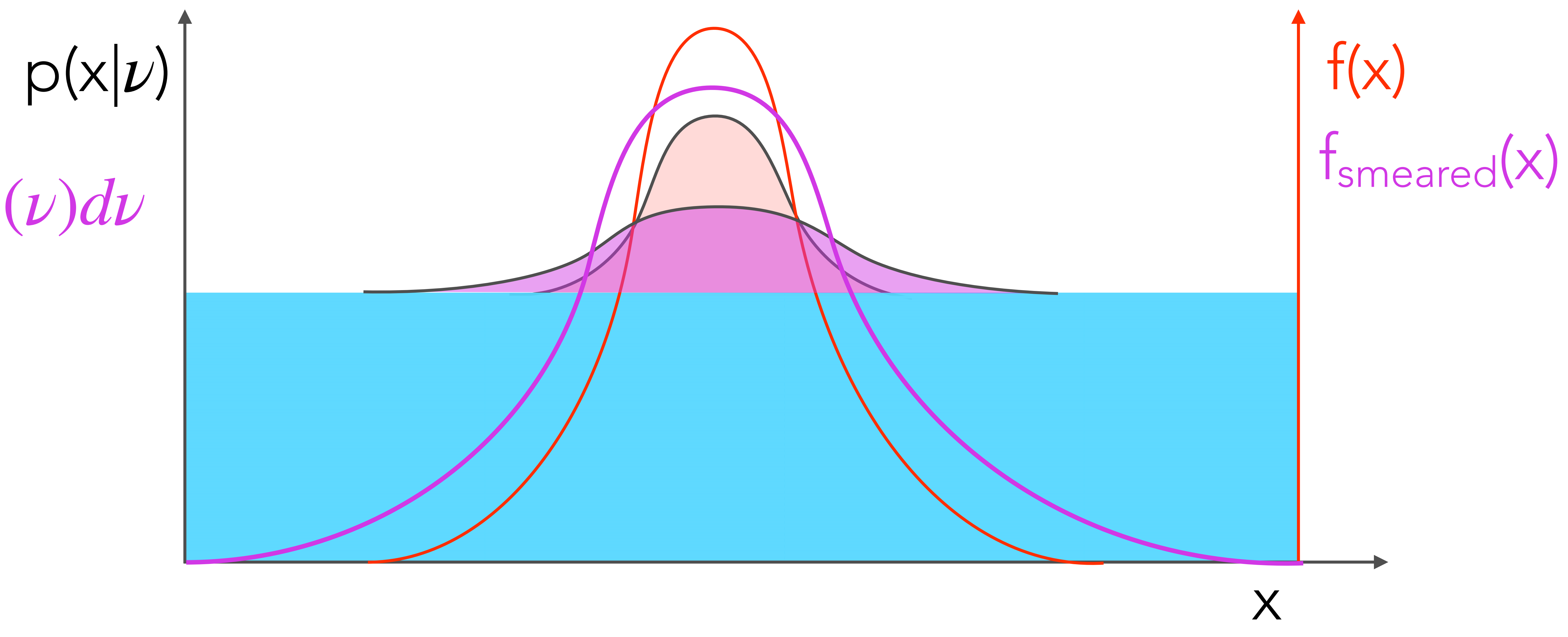
An alternate idea: Data augmentation

An intuitive approach to incorporate systematics into training is to train on “smeared data”, or data generated from a marginal model

$$x_i, y_i \sim p(x,y) = \int d\nu p(x,y|\nu)p(\nu)$$

- Note: this requires a prior / proposal distribution $p(\nu)$

$$p(x) = \int p(x\,|\,\nu)p(\nu)d\nu$$

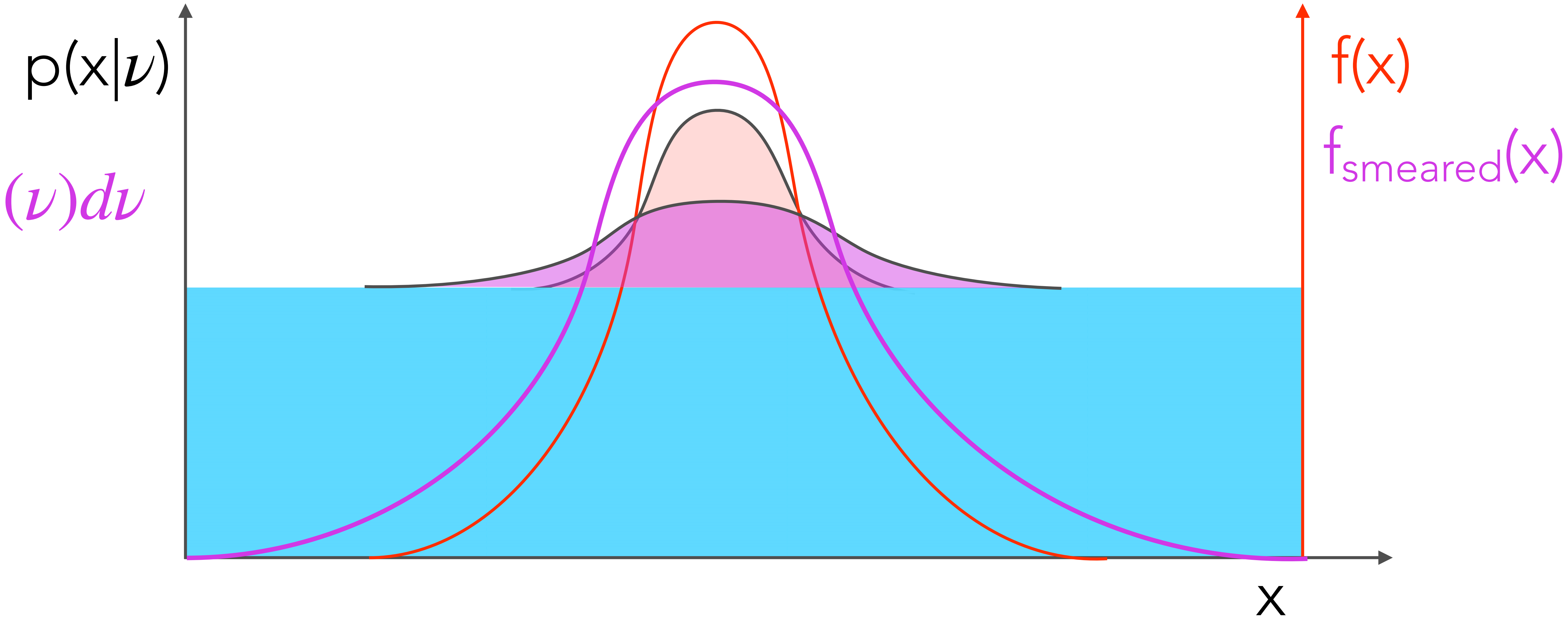


Fixed classifier is not optimal

Training on **smeared** samples with $\nu \sim p(\nu)$ still results in a fixed classifier $f_{\text{smeared}}(x)$

- classifier not optimal for any ν
- we can still propagate uncertainty through the fixed classifier as before

$$p(x) = \int p(x\,|\,\nu)p(\nu)d\nu$$

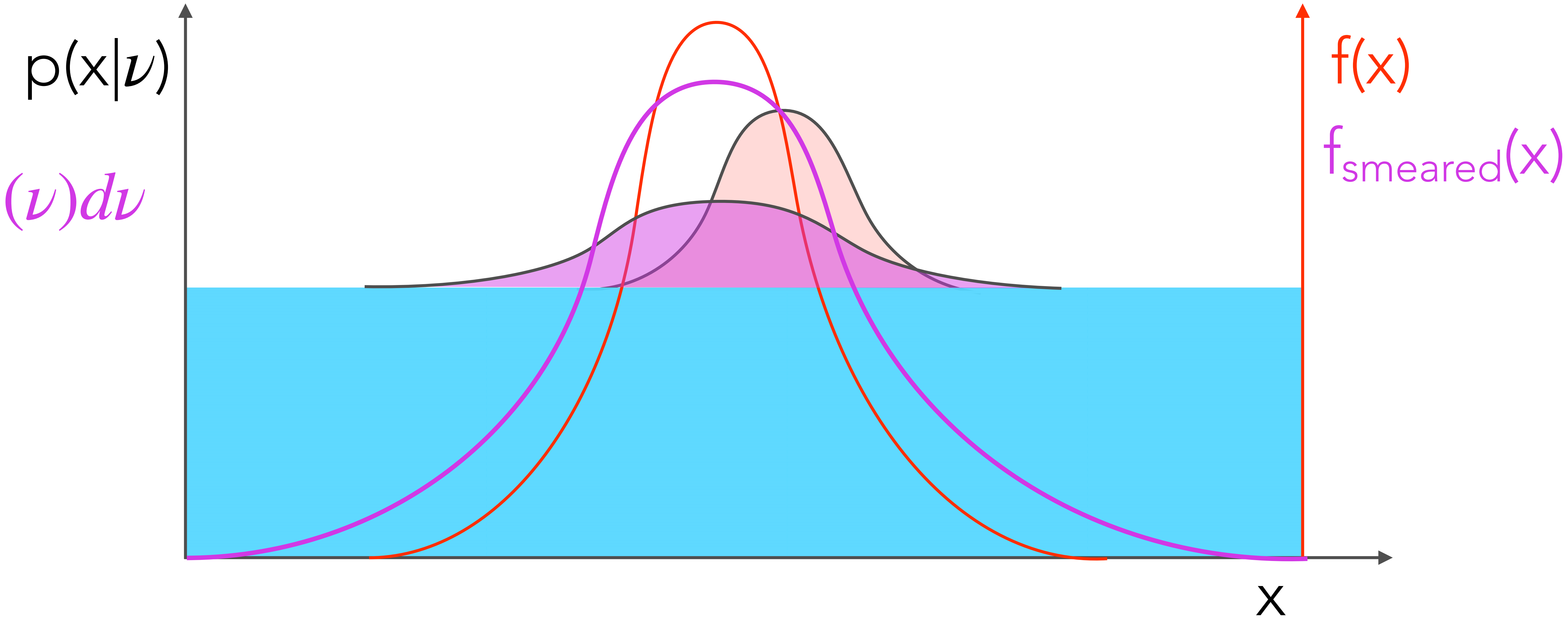


Fixed classifier is not optimal

Training on **smeared** samples with $\nu \sim p(\nu)$ still results in a fixed classifier $f_{\text{smeared}}(x)$

- classifier not optimal for any ν
- we can still propagate uncertainty through the fixed classifier as before

$$p(x) = \int p(x\,|\,\nu)p(\nu)d\nu$$



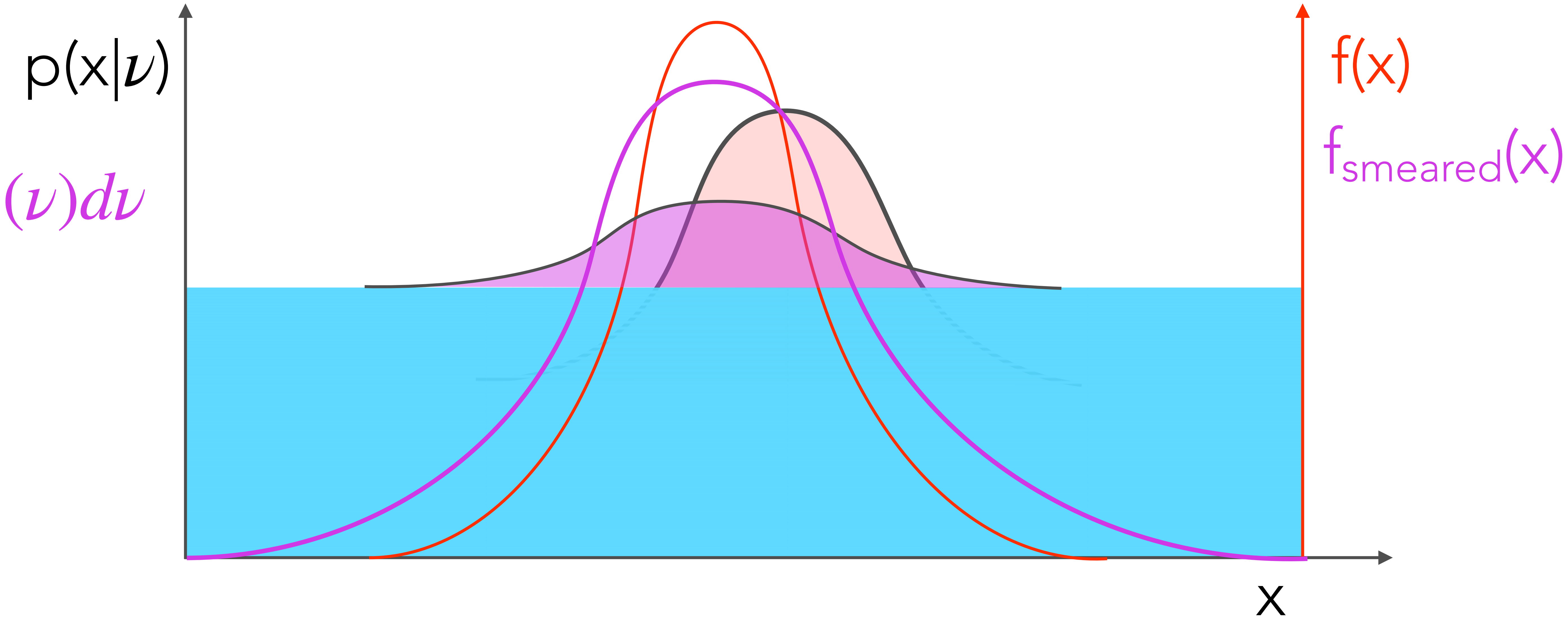
Fixed classifier is not optimal

Training on **smeared** samples with $\nu \sim p(\nu)$ still results in a fixed classifier $f_{\text{smeared}}(x)$

- classifier not optimal for any ν

- we can still propagate uncertainty through the fixed classifier as before

$$p(x) = \int p(x|\nu)p(\nu)d\nu$$



Learning to Pivot

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event (Classifier)	Experiment-wide (Adversary & Hyper parameter opt.)

Learning to pivot with adversarial networks

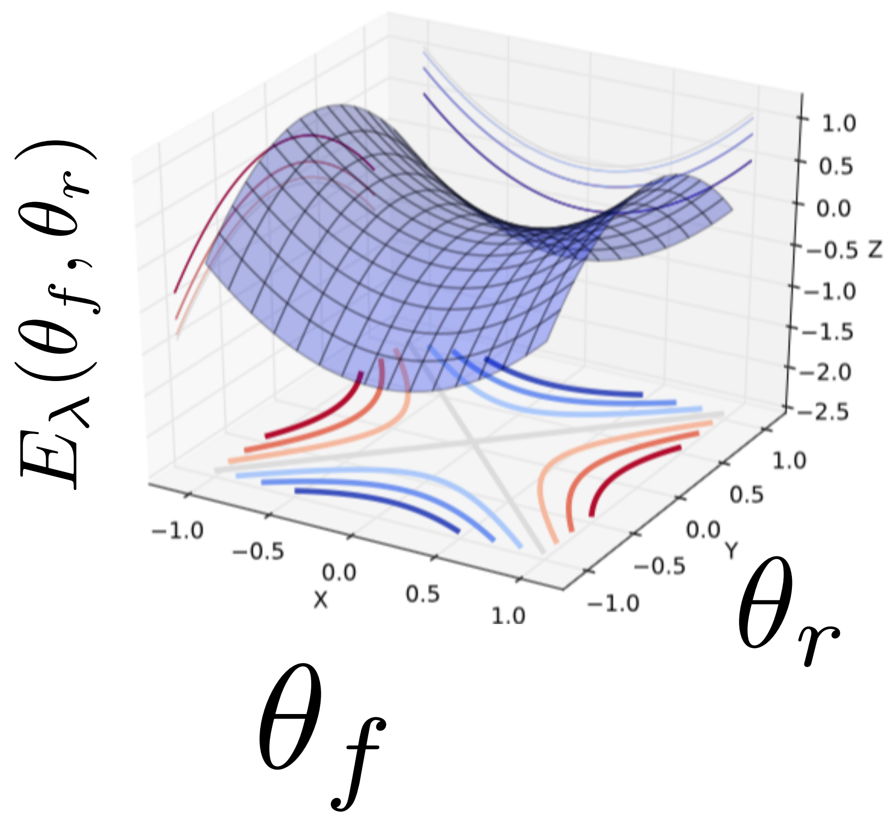
G. Louppe, M. Kagan, K. Cranmer,
arXiv:1611.01046

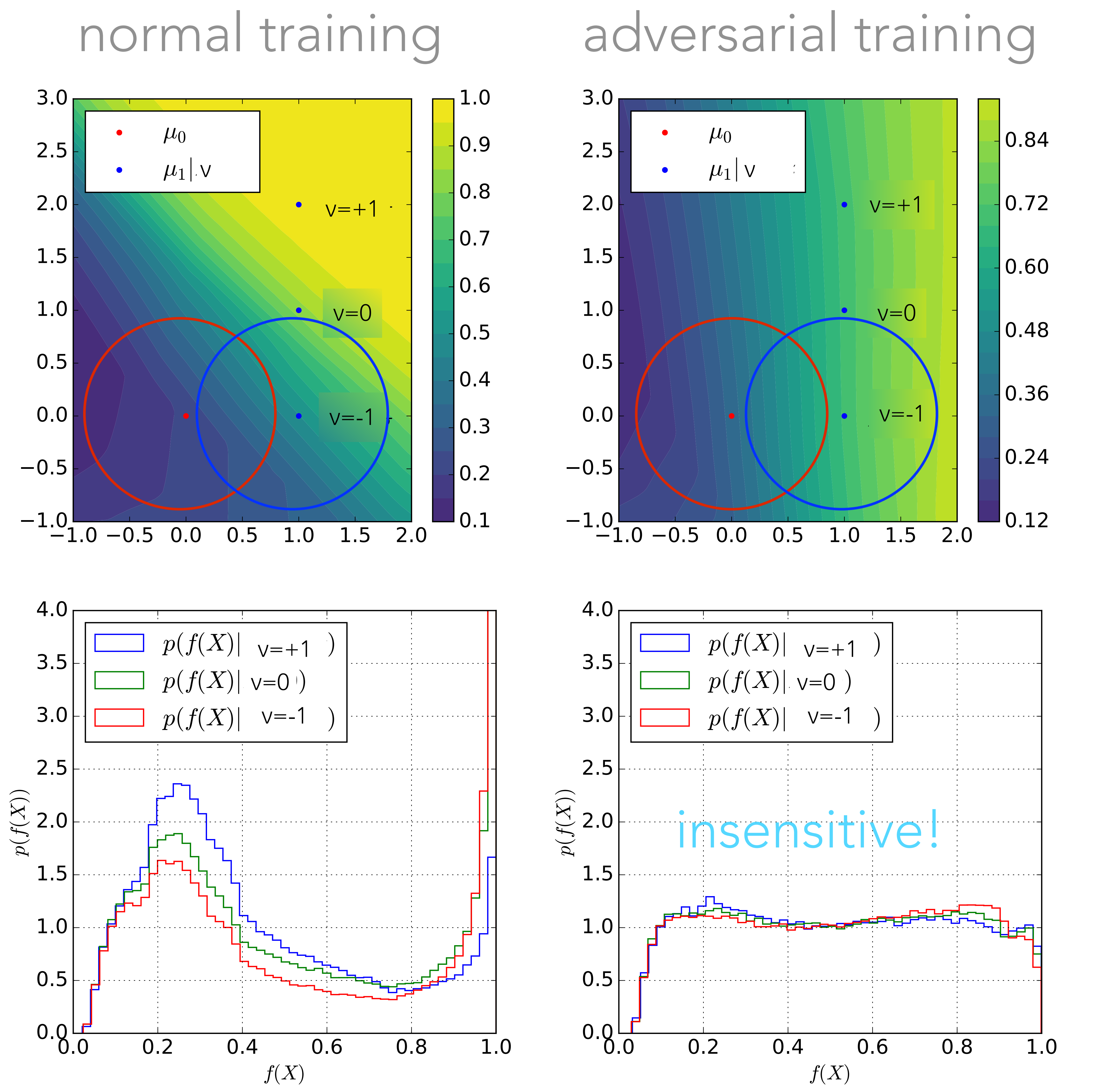
Typically classifier $f(x)$ trained to minimize loss $\mathbf{L_f}$.

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

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

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





An example of learning to pivot

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

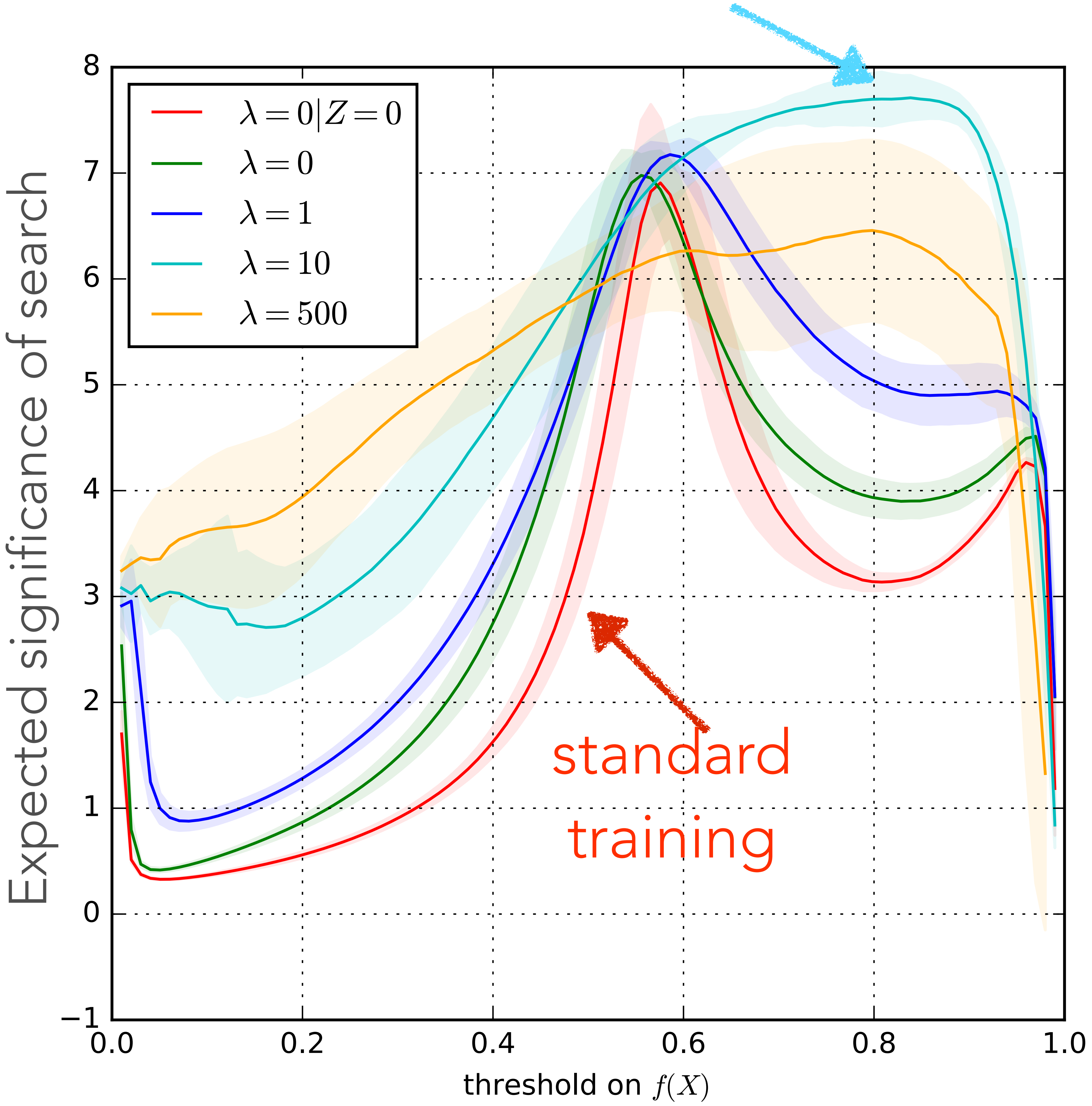
An example:
background: 1000 QCD jets
signal: 100 boosted W's

Train W vs. QCD classifier

Pileup as source of uncertainty

Simple cut-and-count analysis **with background uncertainty.**

optimal tradeoff of classification vs. & robustness

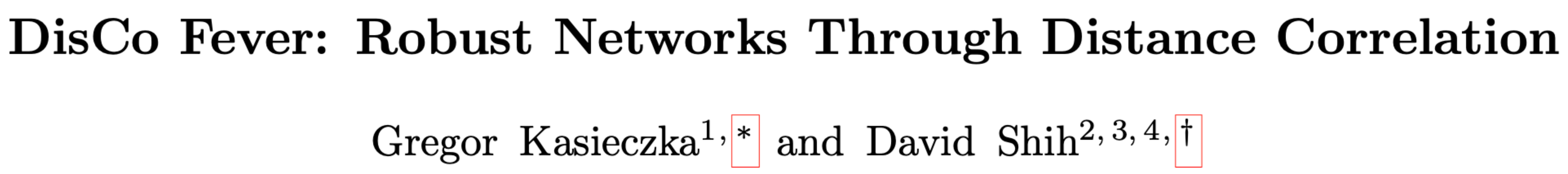


standard training

Learned adversary → explicit regularization

One way of interpreting the mini-max game $\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f}\max_{\theta_r} E(\theta_f, \theta_r).$
is to minimize a **regularized** loss term $\tilde{L}(\theta_f) = \arg\max_{\theta_r} E_\lambda(\theta_f, \theta_r)$ where the optimization with respect to θ_r is not exposed

This motivates another approach in which the regularization is not achieved through a learned adversary, but some other measure of discrepancy



DisCo Fever: Robust Networks Through Distance Correlation

Gregor Kasieczka[1,*] and David Shih[2,3,4,†]

$$L = L_{classifier}(\vec{y}, \vec{y}_{true}) + \lambda\, \mathrm{dCorr}^2_{y_{true}=0}(\vec{m}, \vec{y})$$

$$\begin{aligned}\mathrm{dCov}^2(X,Y) = &\langle |X - X'||Y - Y'|\rangle \\ &+ \langle |X - X'|\rangle\langle |Y - Y'|\rangle \\ &- 2\langle |X - X'||Y - Y''|\rangle\end{aligned}$$

Phys.Rev.Lett. 125 (2020) 12, 122001

Learned adversary → explicit regularization

One way of interpreting th

is to minimize a **regularize**

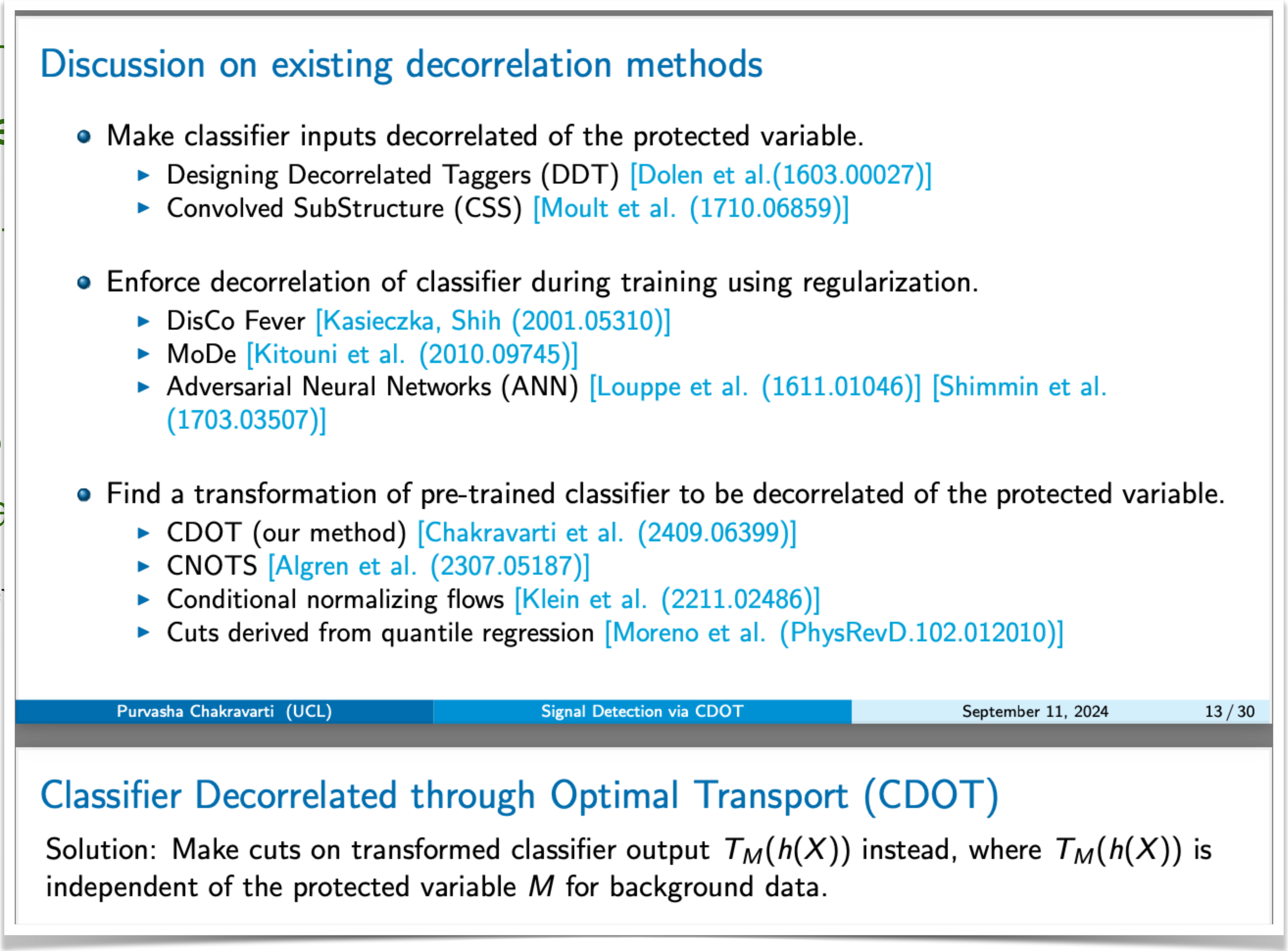
optimization with respect

This motivates another ap

through a learned adversa

DisCo Fe

$$L = L_{classifier}(\vec{y}, \vec{y}_{true}) + \lambda$$



Discussion on existing decorrelation methods

- Make classifier inputs decorrelated of the protected variable.
  - Designing Decorrelated Taggers (DDT) [Dolen et al.(1603.00027)]
  - Convolved SubStructure (CSS) [Moult et al. (1710.06859)]
- Enforce decorrelation of classifier during training using regularization.
  - DisCo Fever [Kasieczka, Shih (2001.05310)]
  - MoDe [Kitouni et al. (2010.09745)]
  - Adversarial Neural Networks (ANN) [Louppe et al. (1611.01046)] [Shimmin et al. (1703.03507)]
- Find a transformation of pre-trained classifier to be decorrelated of the protected variable.
  - CDOT (our method) [Chakravarti et al. (2409.06399)]
  - CNOTS [Algren et al. (2307.05187)]
  - Conditional normalizing flows [Klein et al. (2211.02486)]
  - Cuts derived from quantile regression [Moreno et al. (PhysRevD.102.012010)]

Purvasha Chakravarti (UCL) Signal Detection via CDOT September 11, 2024 13 / 30

Classifier Decorrelated through Optimal Transport (CDOT)

Solution: Make cuts on transformed classifier output $T_M(h(X))$ instead, where $T_M(h(X))$ is independent of the protected variable M for background data.

Phys.Rev.Lett. 125 (2020) 12, 122001

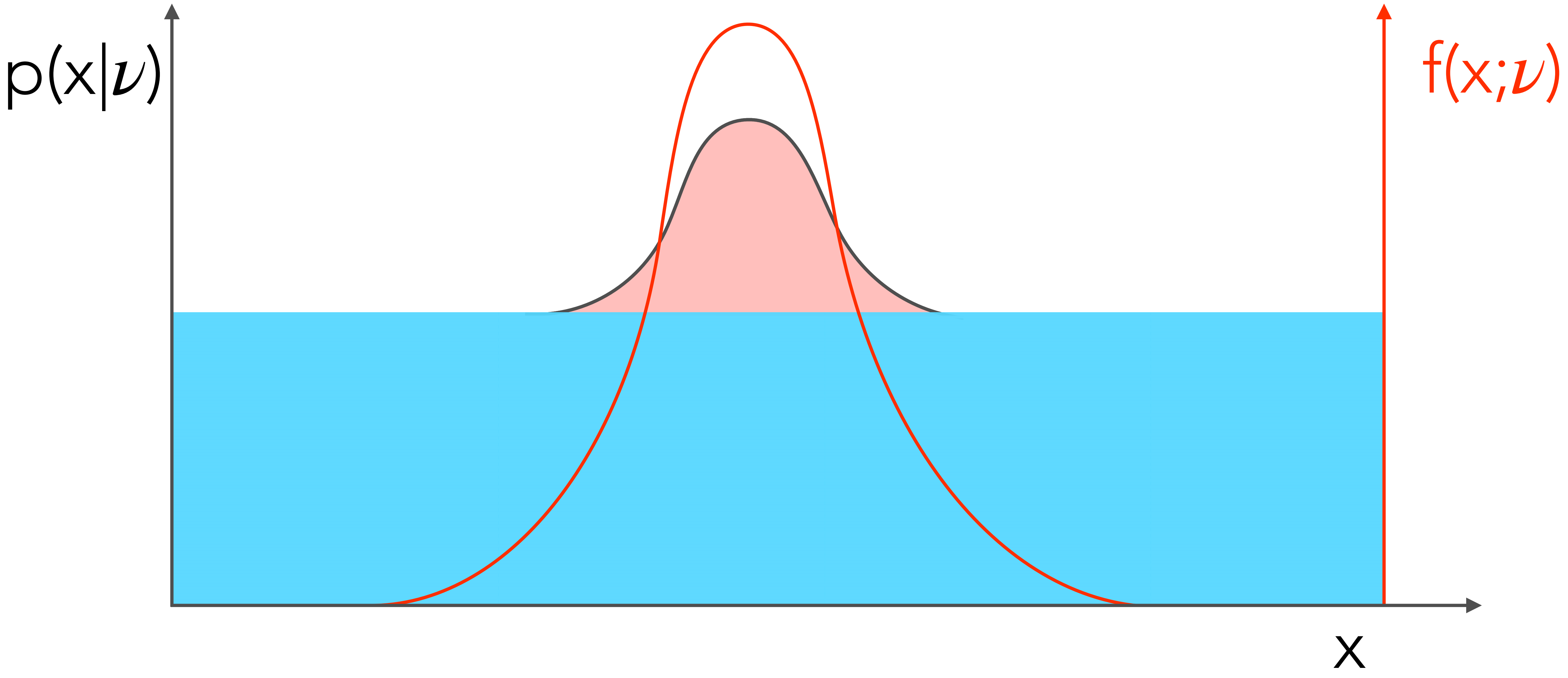
Parametrized Classifier & Parametrized Likelihood Ratio Trick

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Idea: what about a parameterized classier?

We want a classifier that depends on / is parametrized by ν

- augment training data (x,y) → (x,ν,y) to obtain f(x;ν)

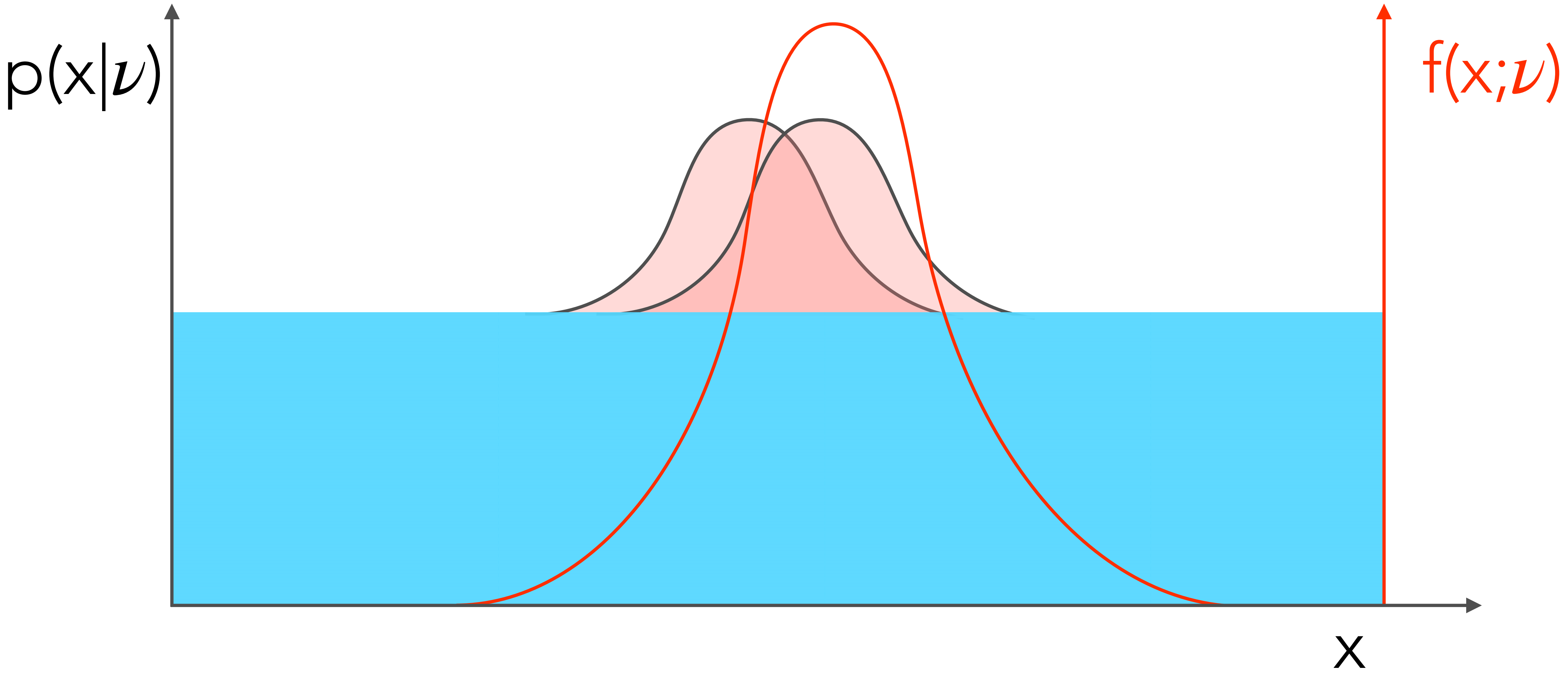


- **Confusing**: how do we evaluate on real data when ν is unknown?

Idea: what about a parameterized classier?

We want a classifier that depends on / is parametrized by ν

- augment training data (x,y) ➞ (x,ν,y) to obtain f(x;ν)

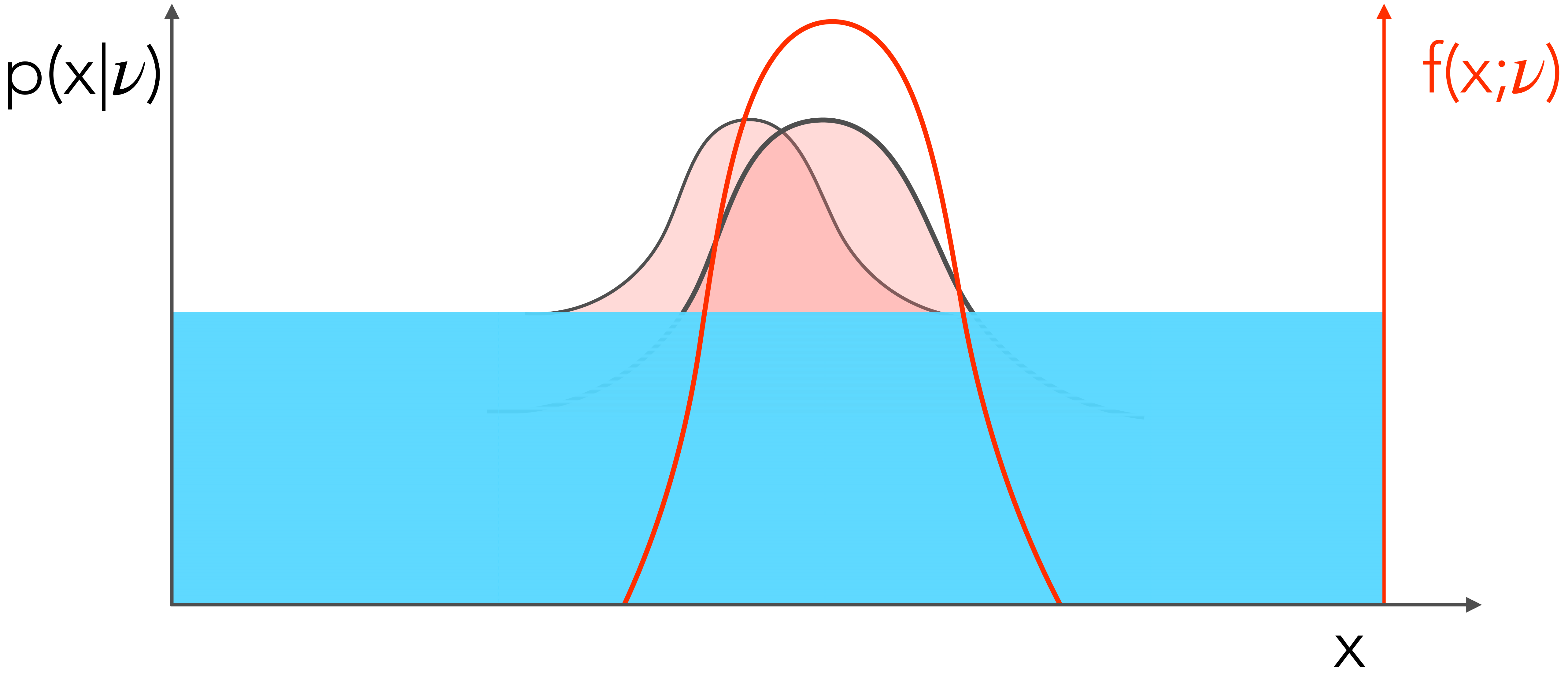


- **Confusing**: how do we evaluate on real data when ν is unknown?

Idea: what about a parameterized classier?

We want a classifier that depends on / is parametrized by ν

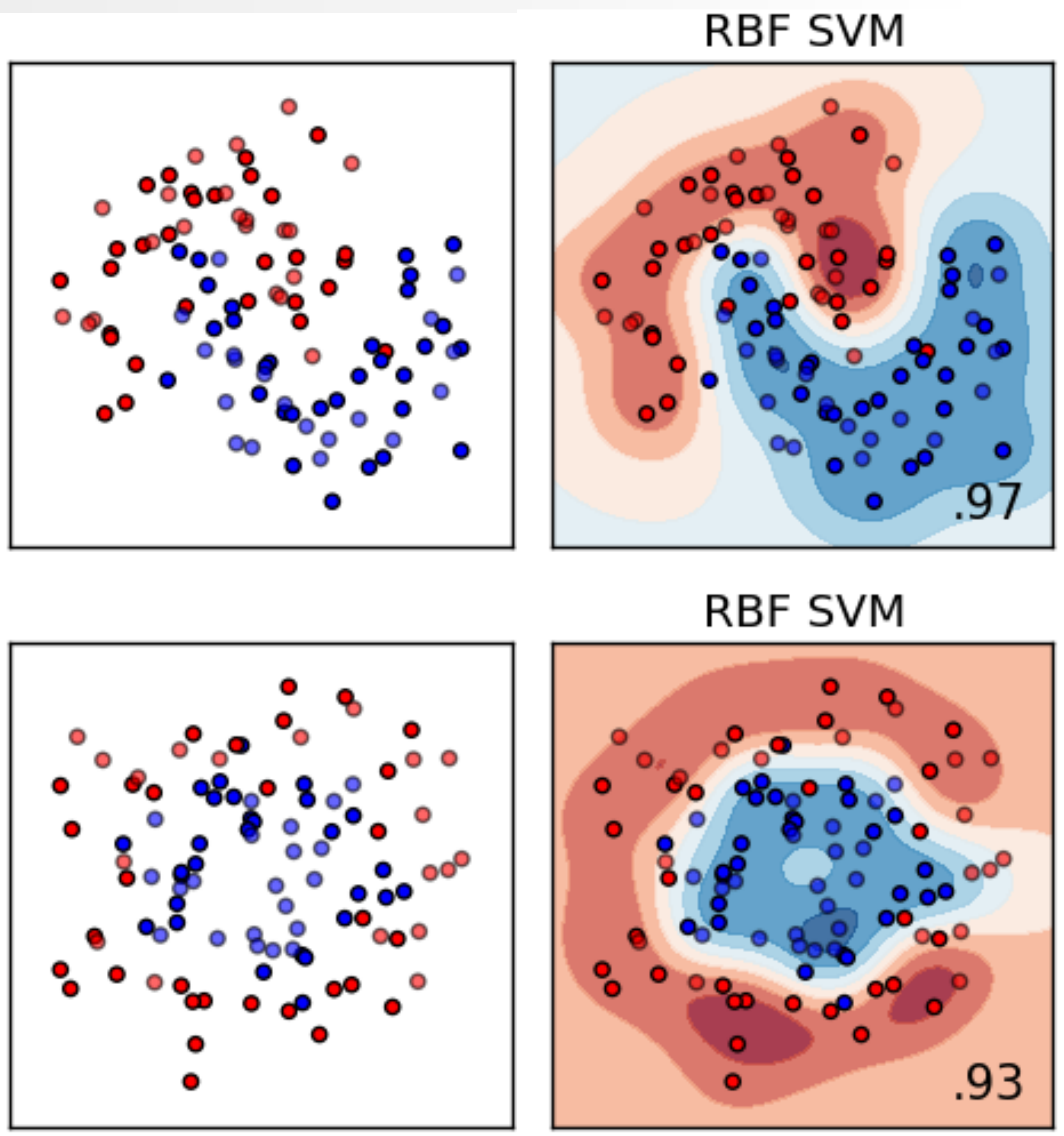
- augment training data (x,y) → (x,ν,y) to obtain f(x;ν)

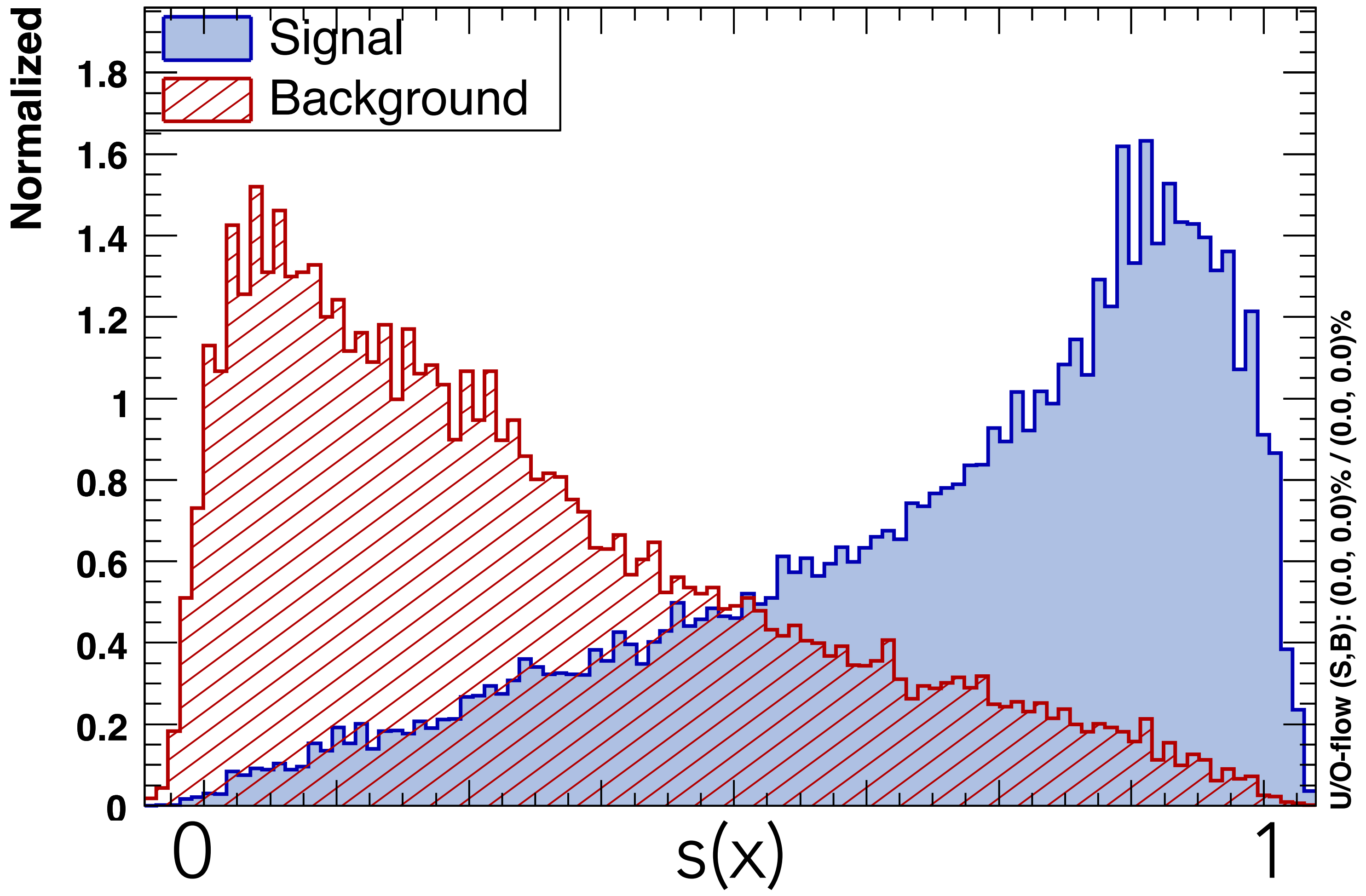


- **Confusing**: how do we evaluate on real data when ν is unknown?

Likelihood Ratio Trick

[Hastie et al., 2001; Sugiyama et al., 2012; Cranmer et. al., 2015]





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

$$L[s] = \mathbb{E}_{p(x|H_1)}[-\log s(x)] + \mathbb{E}_{p(x|H_0)}[-\log(1 - s(x))]$$

- i.e. approximate the optimal classifier

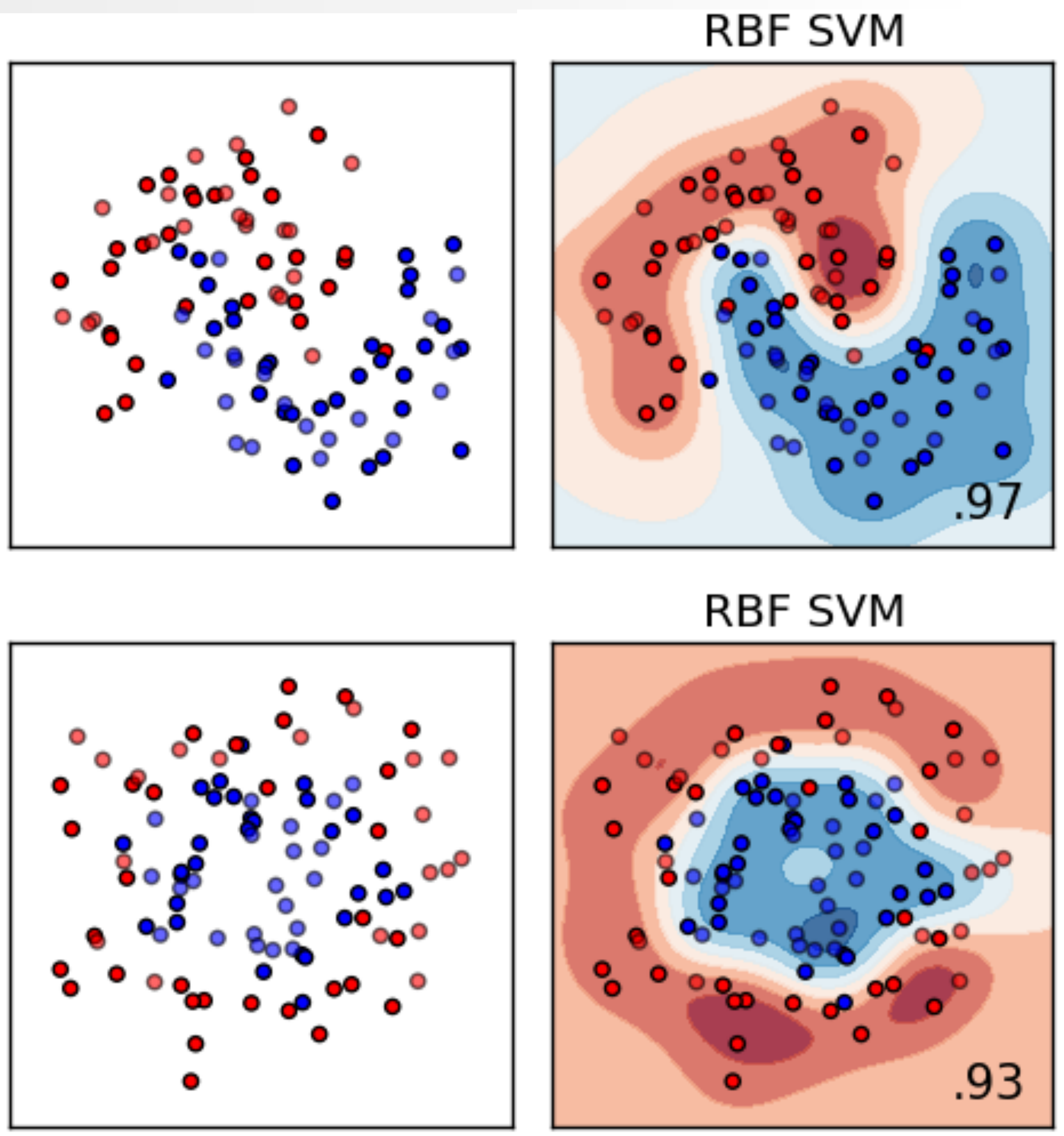
$$s(x) = \frac{p(x|H_1)}{p(x|H_0) + p(x|H_1)}$$

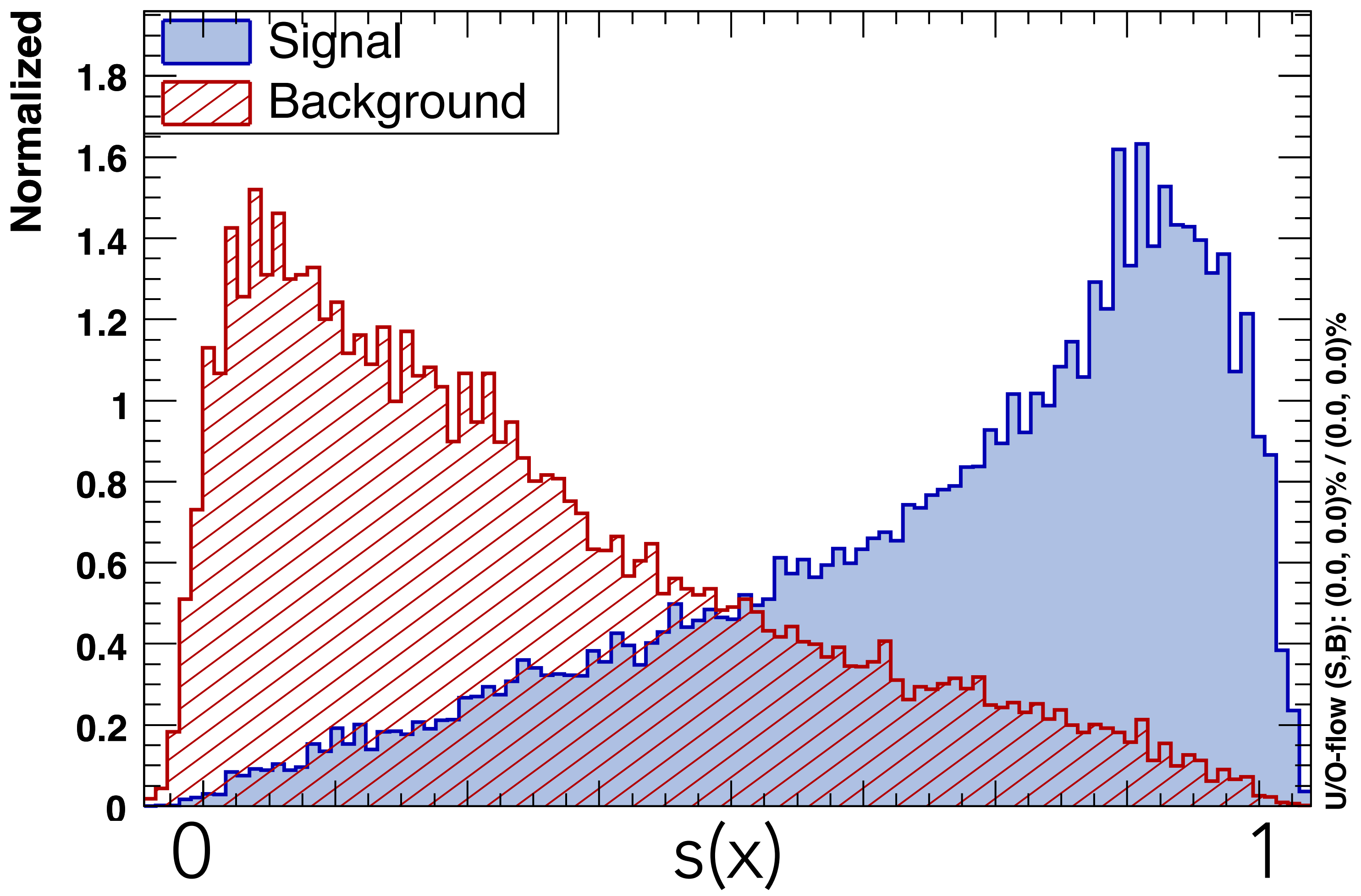
- which is 1-to-1 with the likelihood ratio

$$r(x) = \frac{p(x|H_1)}{p(x|H_0)} = 1 - \frac{1}{s(x)}$$

Likelihood Ratio Trick

[Hastie et al., 2001; Sugiyama et al., 2012; Cranmer et. al., 2015]





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

$$L[s] = \mathbb{E}_{p(x|H_1)}[-\log s(x)] + \mathbb{E}_{p(x|H_0)}[-\log(1 - s(x))]$$

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

- i.e. approximate the optimal classifier

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

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

$$r(x) = \frac{p(x|H_1)}{p(x|H_0)} = 1 - \frac{1}{s(x)}$$

Parametrizing the Likelihood Ratio Trick

Can do the same thing for any two points θ_0 & θ_1 in parameter space $\boldsymbol{\Theta}$.

$$r(x;\theta_0,\theta_1) = \frac{p(x \mid \theta_0)}{p(x \mid \theta_1)} = 1 - \frac{1}{s(x;\theta_0,\theta_1)}$$

Or train to classify data from $p(x\,|\,\theta)$ versus some fixed reference $p_{\rm ref}(x)$

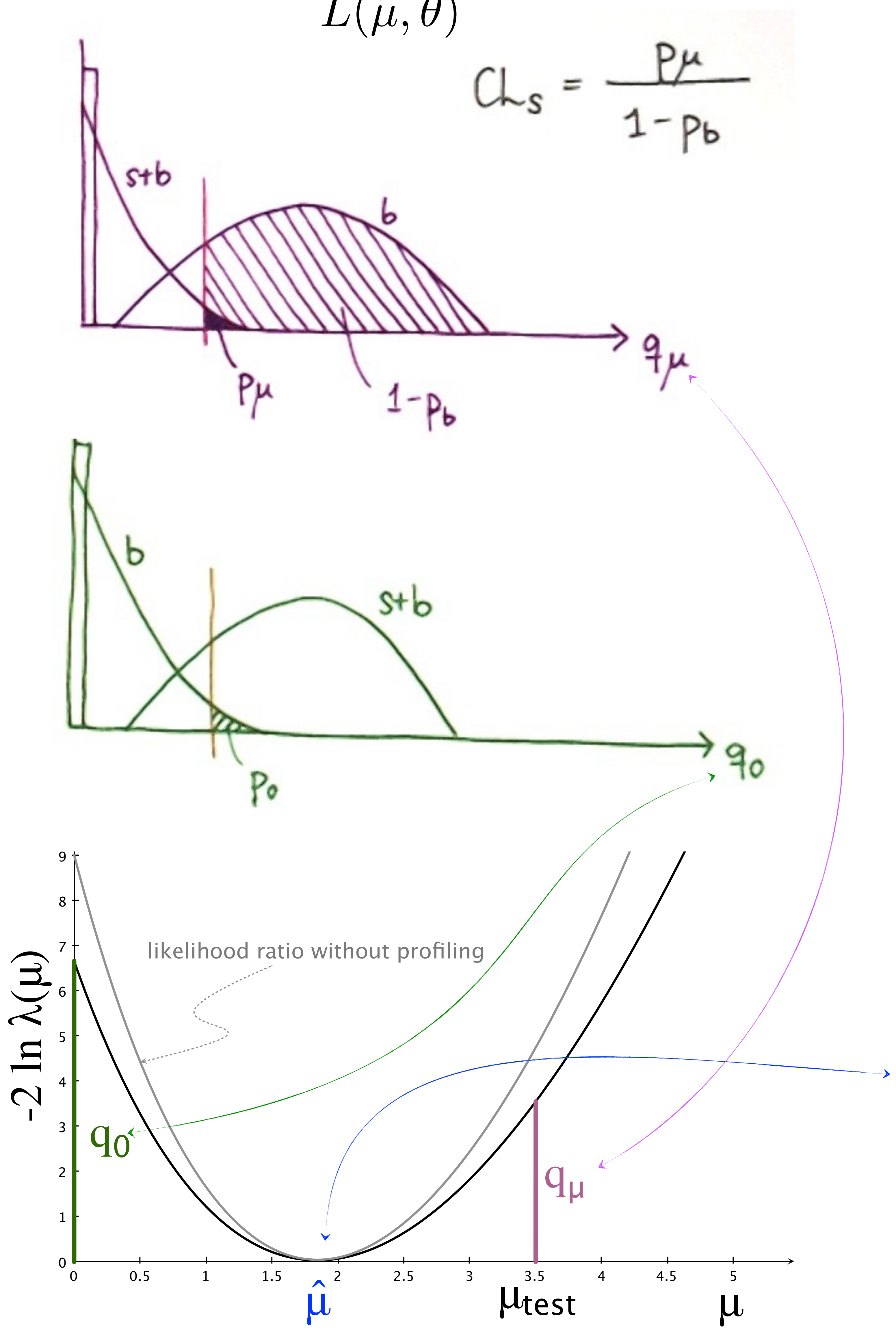
$$r(x;\theta) = \frac{p(x|\theta)}{p_{\rm ref}(x)} = 1 - \frac{1}{s(x;\theta)}$$

I call this a **parametrized classifier.**

K.C., G. Louppe, J. Pavez: Approximating Likelihood Ratios with Calibrated Discriminative Classifiers [arXiv:1506.02169]

Thumbnail of the LHC statistical procedures

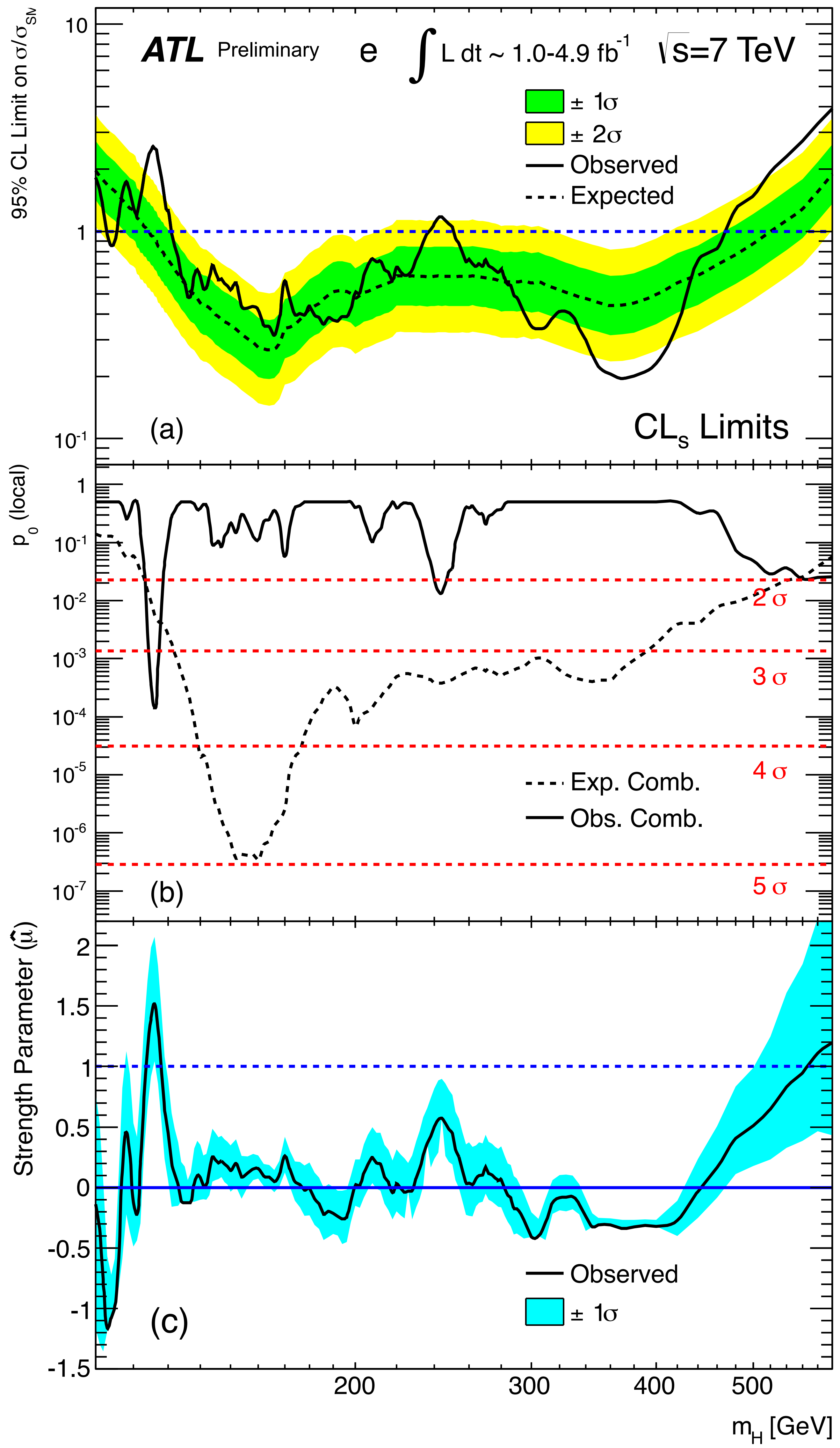
$$\lambda(\mu) = \frac{L(\mu, \hat{\hat{\theta}}(\mu))}{L(\hat{\mu}, \hat{\theta})}$$



CLs to test signal hypothesis

p0 to test background hypothesis

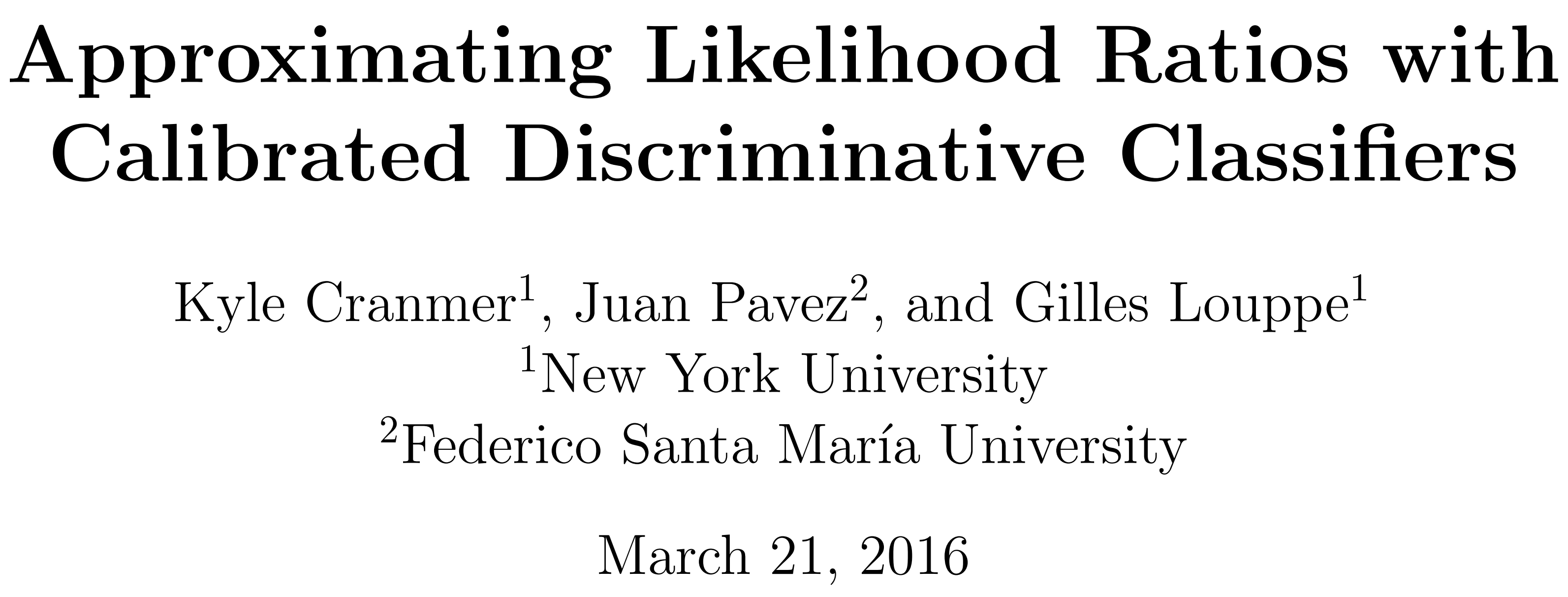
μ̂ to estimate signal strength



SBI & Profile Likelihood Ratio

The original arXiv:1506.02169 paper lays out and demonstrates how the parametrized classifier approach can be used to model the profile likelihood ratio

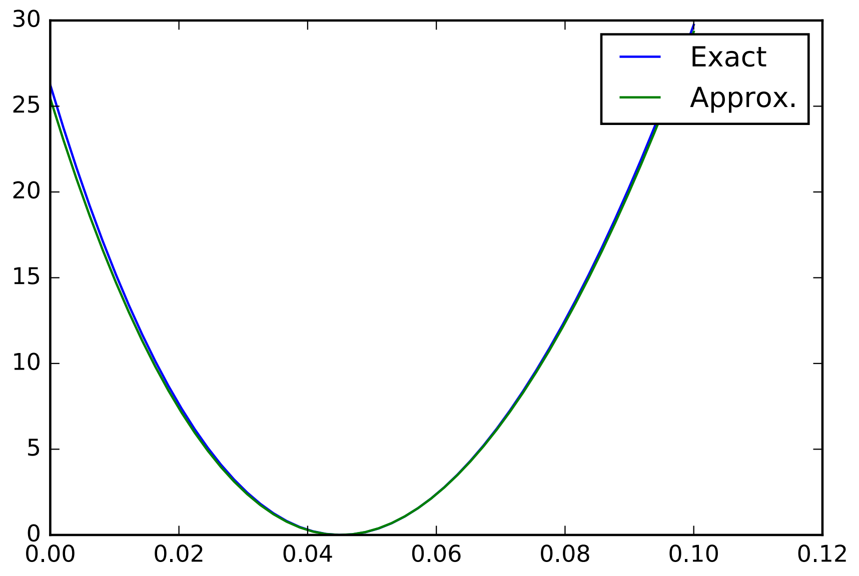
- The basis for later work in MadMiner
- In my mind, cleanest conceptually
  - Also called "uncertainty-aware" in review by Ghosh, Nachman, and Whiteson [arXiv:2105.08742]



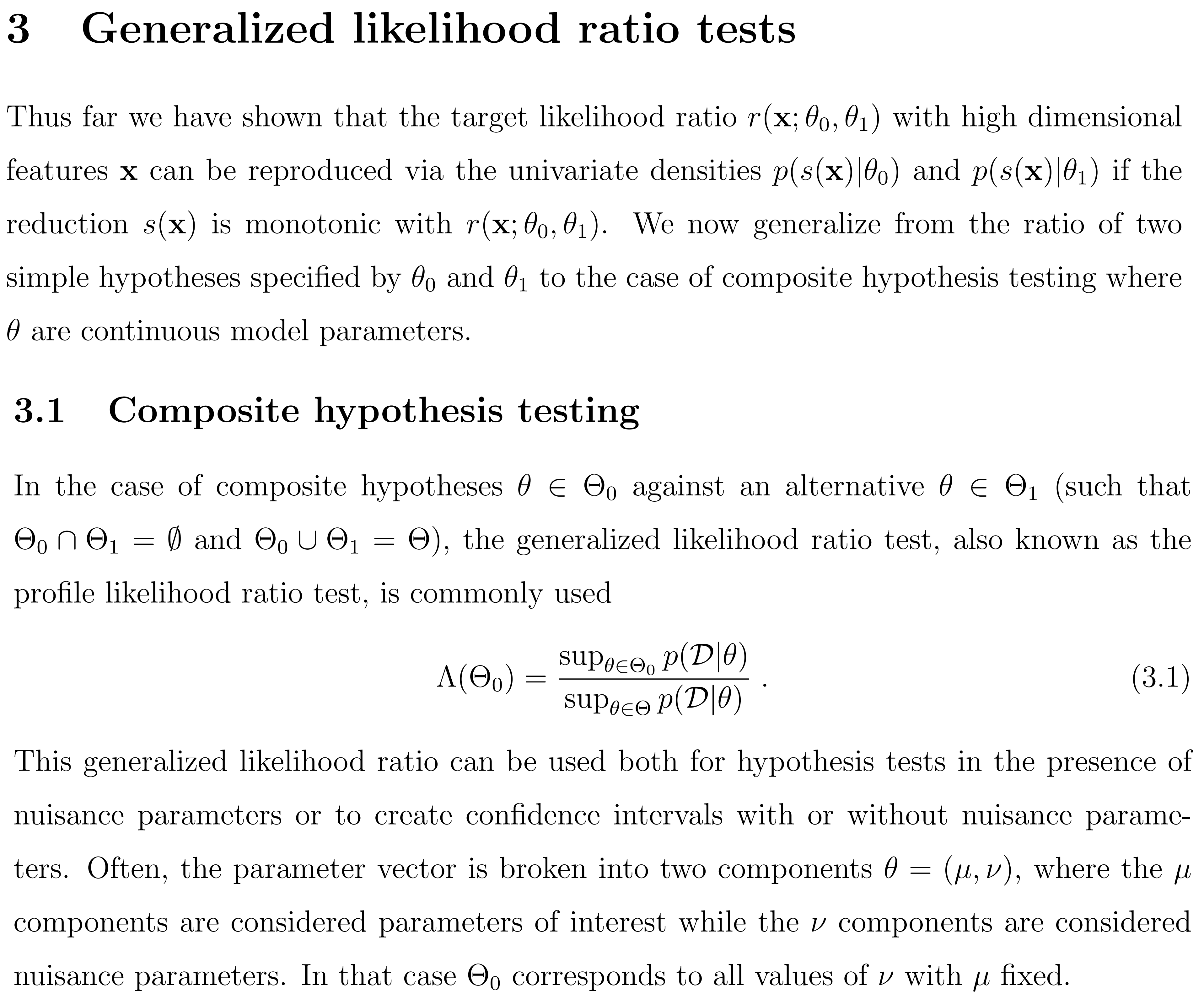
Approximating Likelihood Ratios with Calibrated Discriminative Classifiers

Kyle Cranmer[1], Juan Pavez[2], and Gilles Louppe[1]
[1]New York University
[2]Federico Santa María University

March 21, 2016



(f) $-2\log\Lambda(\gamma)$



3 Generalized likelihood ratio tests

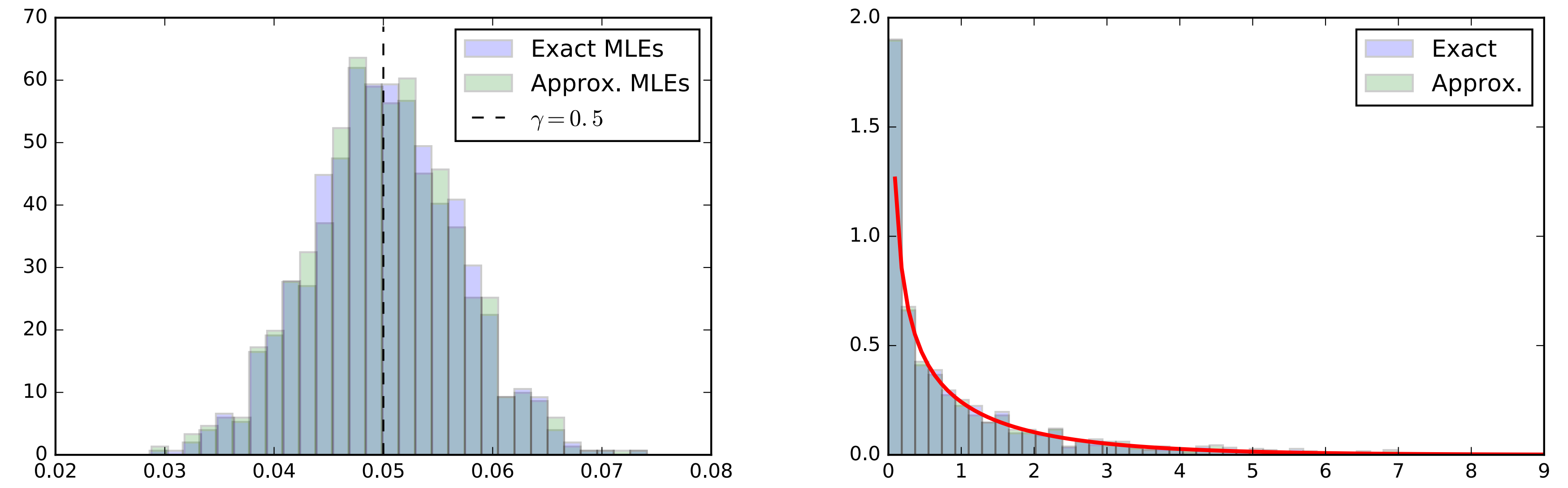
Thus far we have shown that the target likelihood ratio $r(\mathbf{x};\theta_0,\theta_1)$ with high dimensional features $\mathbf{x}$ can be reproduced via the univariate densities $p(s(\mathbf{x})|\theta_0)$ and $p(s(\mathbf{x})|\theta_1)$ if the reduction $s(\mathbf{x})$ is monotonic with $r(\mathbf{x};\theta_0,\theta_1)$. We now generalize from the ratio of two simple hypotheses specified by θ_0 and θ_1 to the case of composite hypothesis testing where θ are continuous model parameters.

3.1 Composite hypothesis testing

In the case of composite hypotheses $\theta \in \Theta_0$ against an alternative $\theta \in \Theta_1$ (such that $\Theta_0 \cap \Theta_1 = \emptyset$ and $\Theta_0 \cup \Theta_1 = \Theta$), the generalized likelihood ratio test, also known as the profile likelihood ratio test, is commonly used

$$\Lambda(\Theta_0) = \frac{\sup_{\theta\in\Theta_0} p(\mathcal{D}|\theta)}{\sup_{\theta\in\Theta} p(\mathcal{D}|\theta)} \,. \tag{3.1}$$

This generalized likelihood ratio can be used both for hypothesis tests in the presence of nuisance parameters or to create confidence intervals with or without nuisance parameters. Often, the parameter vector is broken into two components $\theta = (\mu,\nu)$, where the μ components are considered parameters of interest while the ν components are considered nuisance parameters. In that case Θ_0 corresponds to all values of ν with μ fixed.



(a) Exact vs. approximated MLEs.

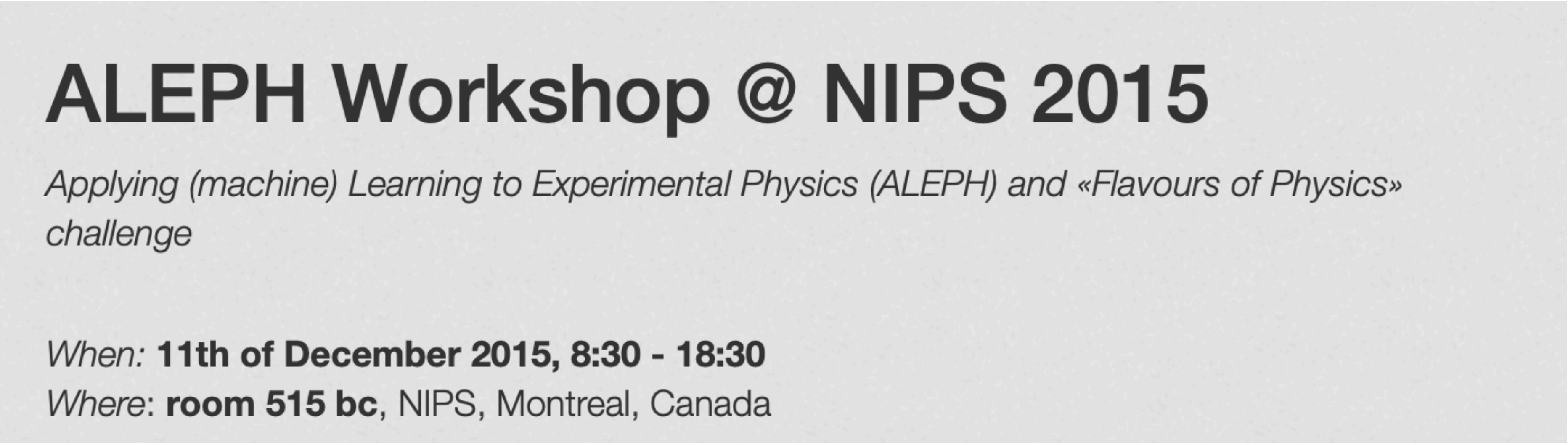
(b) $p(-2\log\Lambda(\gamma=0.05)\,|\,\gamma=0.05)$

Figure 2: Using approximated likelihood ratios for parameter inference yields an unbiased maximum likelihood estimator $\hat{\gamma}$, as empirically estimated from an ensemble of 1000 artificial datasets.

Some parameterized classifier history

2015 NeurIPS ML & Physics workshop:

- http://yandexdataschool.github.io/aleph2015/
- https://indico.cern.ch/event/465572/



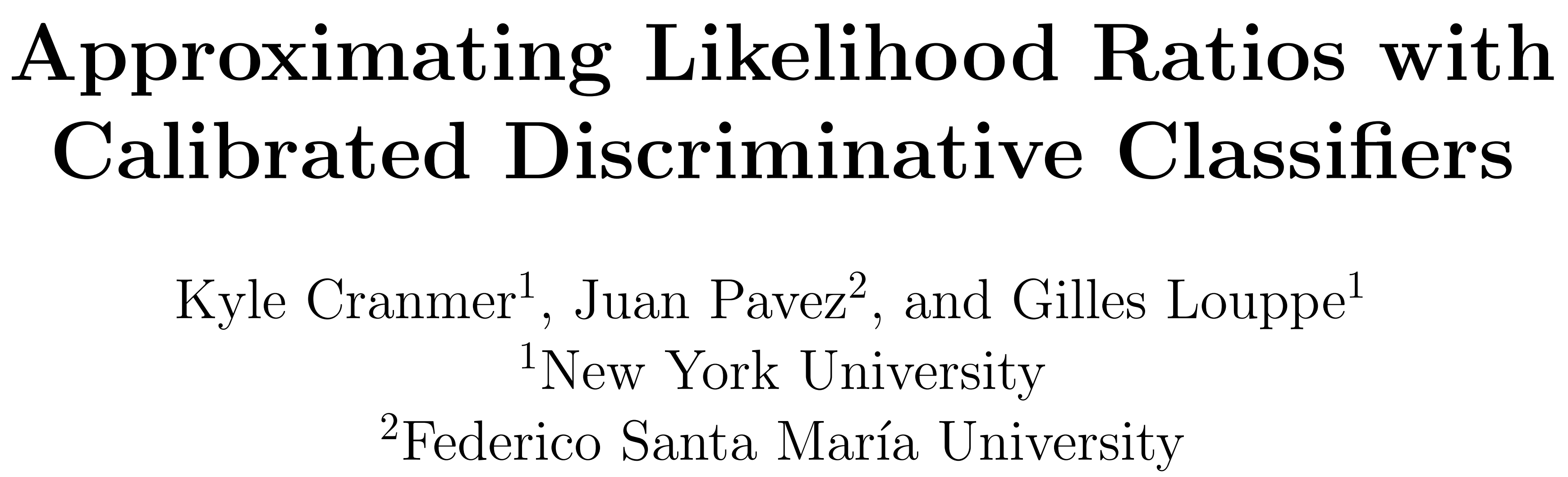
ALEPH Workshop @ NIPS 2015

Applying (machine) Learning to Experimental Physics (ALEPH) and «Flavours of Physics» challenge

When: **11th of December 2015, 8:30 - 18:30**
Where: **room 515 bc**, NIPS, Montreal, Canada

First SBI paper with Neural Likelihood Ratios

- "CARL" paper arXiv:1506.02169



Approximating Likelihood Ratios with Calibrated Discriminative Classifiers

Kyle Cranmer[1], Juan Pavez[2], and Gilles Louppe[1]
[1]New York University
[2]Federico Santa María University

2016 NeurIPS Keynote

- https://doi.org/10.6084/m9.figshare.4291565.v1

NIPS 2016

BARCELONA · SPAIN · DECEMBER 5 - 10, 2016 | http://nips.cc/

TUTORIALS

Deep Reinforcement Learning Through Policy Optimization
Pieter Abbeel (OpenAI, UC Berkeley) and John Schulman (OpenAI)

Large-scale Optimization: Beyond Stochastic Gradient Descent and Convexity
Francis Bach (INRIA, ENS) and Suvrit Sra (MIT)

Variational Inference: Foundations and Modern Methods
David Blei (Columbia), Shakir Mohamed (Google Deepmind) and Rajesh Ranganath (Princeton)

Natural Language Processing for Computational Social Science
Cristian Danescu-Niculescu-Mizil (Cornell) and Lillian Lee (Cornell)

Generative Adversarial Networks
Ian Goodfellow (OpenAI)

Theory and Algorithms for Forecasting Non-stationary Time Series
Vitaly Kuznetsov (Google) and Mehryar Mohri (Courant Institute, Google Research)

Deep Learning for Building AI Systems
Andrew Ng (Baidu, Stanford University)

ML Foundations and Methods for Precision Medicine and Healthcare
Suchi Saria (Johns Hopkins) and Peter Schulam (Johns Hopkins)

Crowdsourcing: Beyond Label Generation
Jenn Wortman Vaughan (Microsoft Research)

INVITED SPEAKERS

Reproducible Research: the Case of the Human Microbiome
Susan Holmes (Stanford University)

Dynamic Legged Robots
Marc Raibert (Boston Dynamics)

Intelligent Biosphere
Drew Purves (Google DeepMind)

Predictive Learning
Yann LeCun (Facebook and New York University)

Machine Learning and Likelihood-Free Inference in Particle Physics
Kyle Cranmer (New York University)

Learning About the Brain: Neuroimaging and Beyond
Irina Rish (IBM T.J. Watson Research Center)

Engineering Principles From Stable And Developing Brains
Saket Navlakha (The Salk Institute for Biological Studies)

SYMPOSIA

Recurrent Neural Networks and other Machines that Learn Algorithms
Alex Graves (Google DeepMind)
Juergen Schmidhuber (IDSIA)
Rupesh Srivastava (IDSIA)
Sepp Hochreiter (Johannes Kepler University)

Deep Learning
Navdeep Jaitly (Google)
Roger Grosse (University of Toronto)
Yann LeCun (New York University & Facebook)

Machine Learning and the Law
Adrian Weller (Cambridge, Alan Turing Inst.)
Conrad McDonnell (Gray's Inn Tax Chambers)
Jatinder Singh (University of Cambridge)
Thomas Grant (University of Cambridge)

ORGANIZING COMMITTEE

General Chairs:
Daniel D Lee (University of Pennsylvania)
Masashi Sugiyama (The University of Tokyo)

Program Chairs:
Ulrike von Luxburg (University of Tübingen)
Isabelle Guyon (Clopinet)

Tutorials Chair:
Joelle Pineau (McGill University)
Hanna Wallach (Microsoft)

Workshop Chairs:
Ralf Herbrich (Amazon)

Demonstration Chair:
Raia Hadsell (Google DeepMind)

Publications Chair & Electronic Proceedings Chair:
Roman Garnett (Washington University)

Program Managers:
Krikamol Muandet (Mahidol University and MPI)
Rohit Babbar, Behzad Tabibian (MPI for Intelligent Systems)

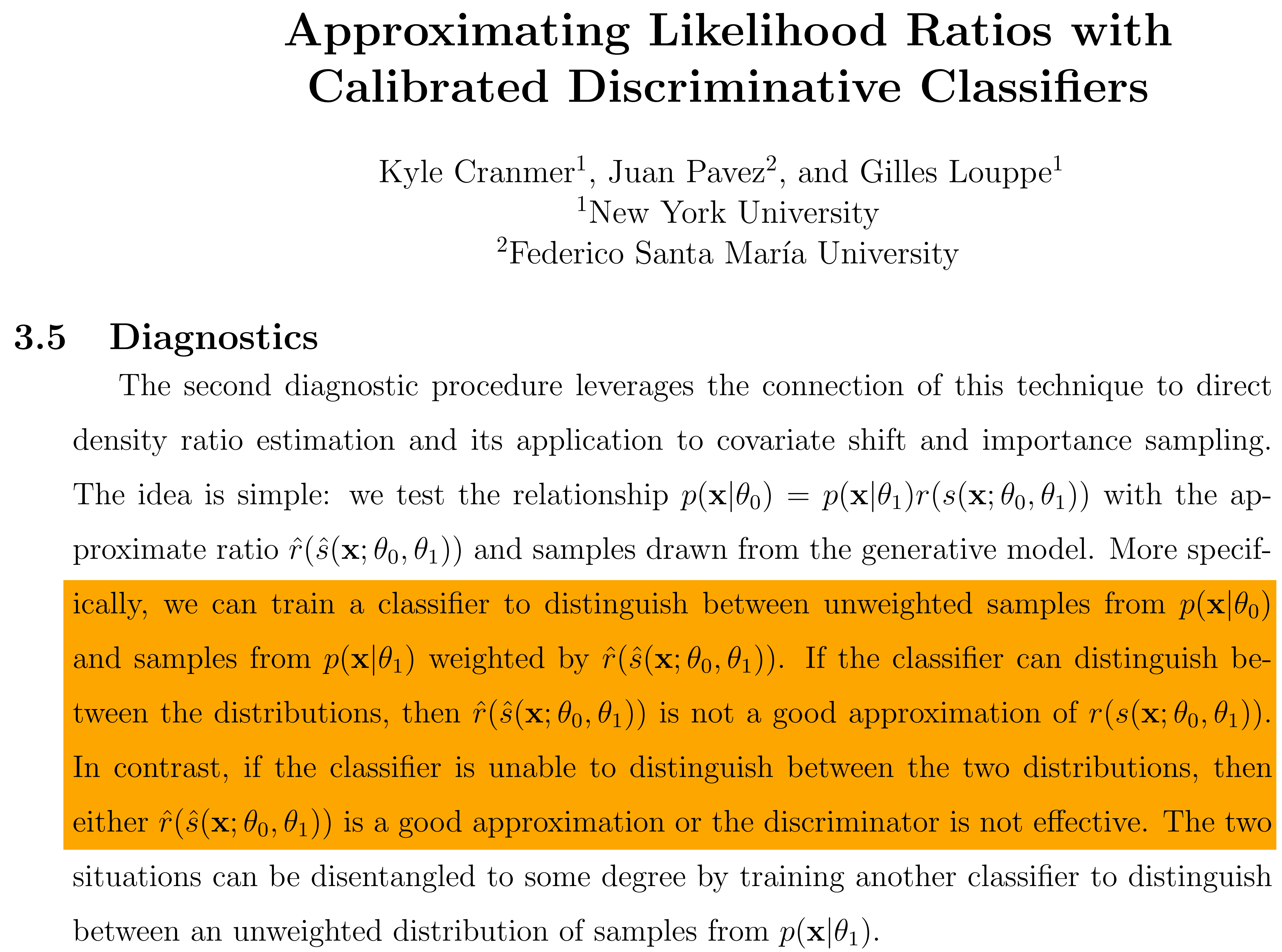
PROGRAM COMMITTEE

Emmanuel Abbe, Princeton Univ.
Alekh Agarwal, Microsoft
Anima Anandkumar, UC Irvine
Chloé-Agathe Azencott, MINES ParisTech
Shai Ben-David, Univ. Waterloo
Alina Beygelzimer, Yahoo Research
Jeff Bilmes, Univ. of Washington, Seattle
Gilles Blanchard, Univ. of Potsdam
Matthew Blaschko, KU Leuven
Tamara Broderick, MIT
Sebastien Bubeck, Princeton
Alexandra Carpentier, Univ. Potsdam
Miguel Carreira-Perpinan, UC Merced
Kamalika Chaudhuri, UC San Diego
Gal Chechik, Google, Bar-Ilan Univ.
Kyunghyun Cho, New York Univ.
Aaron Courville, Univ. of Montreal
Koby Crammer, Technion
Florence d'Alché-Buc, Telecom Paris Tech
Arnak Dalalyan, ENSAE ParisTech
Marc Deisenroth, Imperial College London
Francesco Dinuzzo, Amazon
Finale Doshi-Velez, Harvard
Ran El-Yaniv, Technion
Hugo Jair Escalante, INAOE
Sergio Escalera, Univ. of Barcelona
Maryam Fazel, Univ. of Washington
Aasa Feragen, Univ. of Copenhagen
Rob Fergus, New York Univ.
Xiaoli Fern, Oregon State Univ.
Francois Fleuret, Idiap Research Institute
Surya Ganguli, Stanford
Peter Gehler, Univ. of Tübingen
Claudio Gentile, DiSTA, Universita dell'Insubria
Lise Getoor, UC Santa Cruz
Mark Girolami, Imperial College London
Amir Globerson, Tel Aviv Univ.
Yoav Goldberg, Bar Ilan Univ.
Manuel Gomez, Max Planck Institute
Yves Grandvalet, Univ. of Compiègne & CNRS
Moritz Grosse-Wentrup, MPI
Zaid Harchaoui, Univ. of Washington
Moritz Hardt, Google
Matthias Hein, Saarland Univ.
Philipp Hennig, MPI IS Tübingen
Frank Hutter, Univ. of Freiburg
Prateek Jain, Microsoft Research
Navdeep Jaitly, Google Brain
Stefanie Jegelka, MIT
Samuel Kaski, Aalto Univ.
Koray Kavukcuoglu, Google DeepMind
Jens Kober, TU Delft
Samory Kpotufe, Princeton Univ.
Sanjiv Kumar, Google Research
James Kwok, Hong Kong Univ.
Simon Lacoste-Julien, U. of Montreal
Christoph Lampert, IST Austria
Hugo Larochelle, Twitter
Francois Laviolette, L'Université Laval
Honglak Lee, Univ. of Michigan
Christoph Lippert, Human Longevity
Po-Ling Loh, UW-Madison
Phil Long, Sentient Technologies
Jakob Macke, Caesar Bonn
Julien Mairal, Inria
Shie Mannor, Technion
Marina Meila, Univ. of Washington
Claire Monteleoni, George Washington Univ.
Remi Munos, Google DeepMind
Guillaume Obozinski, Ecole Paris
Cheng Soon Ong, Data61 and ANU
Francesco Orabona, Stony Brook U.
Fernando Perez-Cruz, Universidad Carlos III de Madrid, Bell Labs (Nokia)
Jonathan Pillow, Princeton Univ.
Doina Precup, McGill Montreal
Alain Rakotomamonjy, Univ. of Rouen
Manuel Rodriguez, Max Planck Inst.
Rómer Rosales, Linkedin
Lorenzo Rosasco, U. of Genova, MIT
Sivan Sabato, Ben-Gurion Univ.
Mehreen Saeed, FAST, Univ of CES
Ruslan Salakhutdinov, CMU
Purnamrita Sarkar, Univ. T. Austin
Fei Sha, USC
Ohad Shamir Weizmann, Inst of Science
Jonathon Shlens, Google Brain
David Sontag, New York Univ.
Suvrit Sra, MIT
Karthik Sridharan, Cornell Univ.
Bharath Sriperumbudur, Pennsylvania State Univ.
Erik Sudderth, Brown Univ.
Csaba Szepesvari, Univ. of Alberta
Graham Taylor, Univ. of Guelph
Ambuj Tewari, Univ. of Michigan
Ruth Urner, MPI Tübingen
Benjamin Van Roy, Stanford
Jean-Philippe Vert, MINES ParisTech
Bob Williamson, Data61 and ANU
Jennifer Wortman, Vaughan Microsoft Research
Lin Xiao, Microsoft Research
Kun Zhang, CMU

Diagnostics with classifiers

This paper also introduced two diagnostics

- classifier tests with data reweighed

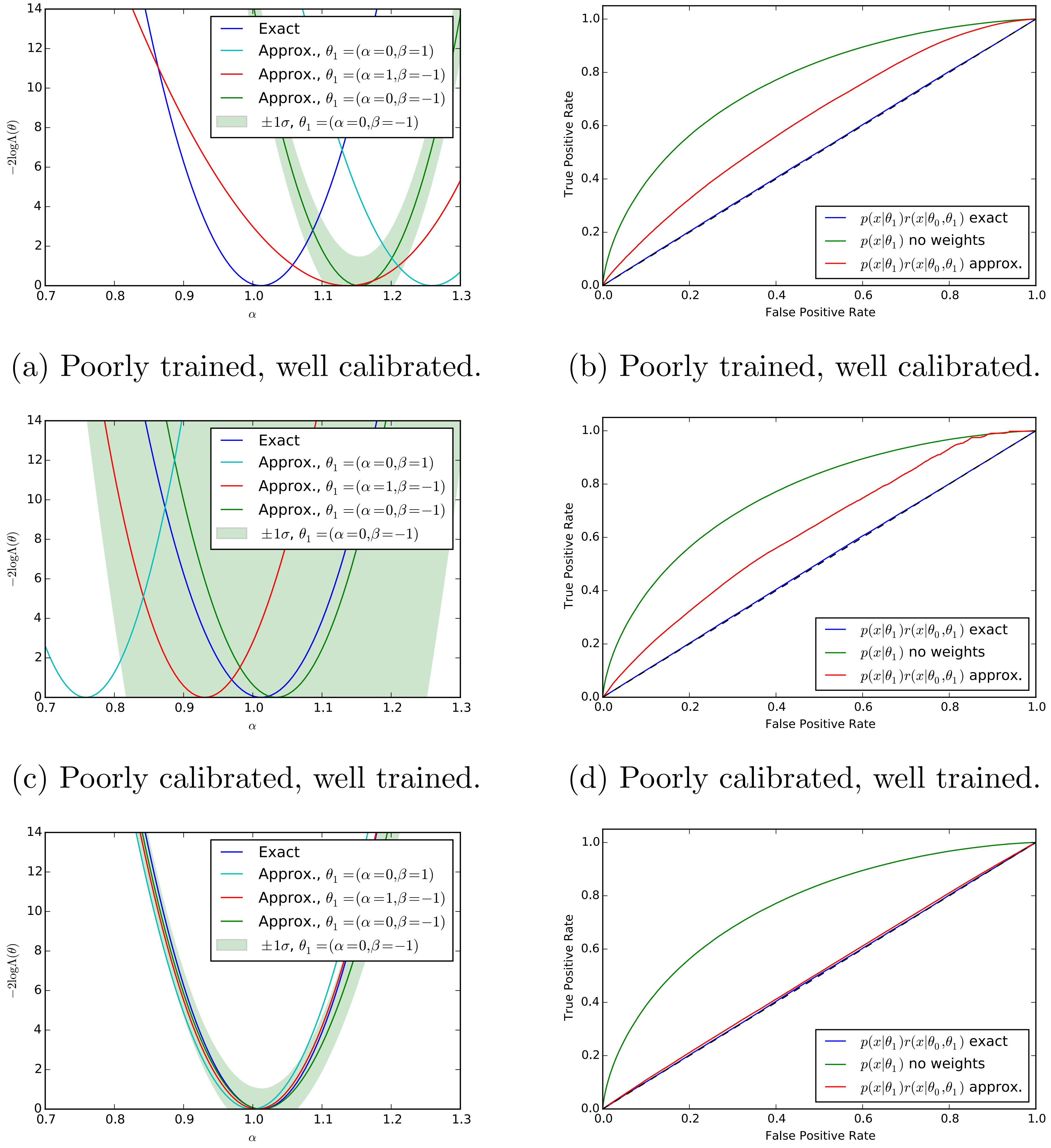


Approximating Likelihood Ratios with Calibrated Discriminative Classifiers

Kyle Cranmer[1], Juan Pavez[2], and Gilles Louppe[1]
[1]New York University
[2]Federico Santa María University

3.5 Diagnostics

The second diagnostic procedure leverages the connection of this technique to direct density ratio estimation and its application to covariate shift and importance sampling. The idea is simple: we test the relationship $p(\mathbf{x}|\theta_0) = p(\mathbf{x}|\theta_1)r(s(\mathbf{x};\theta_0,\theta_1))$ with the approximate ratio $\hat{r}(\hat{s}(\mathbf{x};\theta_0,\theta_1))$ and samples drawn from the generative model. More specifically, we can train a classifier to distinguish between unweighted samples from $p(\mathbf{x}|\theta_0)$ and samples from $p(\mathbf{x}|\theta_1)$ weighted by $\hat{r}(\hat{s}(\mathbf{x};\theta_0,\theta_1))$. If the classifier can distinguish between the distributions, then $\hat{r}(\hat{s}(\mathbf{x};\theta_0,\theta_1))$ is not a good approximation of $r(s(\mathbf{x};\theta_0,\theta_1))$. In contrast, if the classifier is unable to distinguish between the two distributions, then either $\hat{r}(\hat{s}(\mathbf{x};\theta_0,\theta_1))$ is a good approximation or the discriminator is not effective. The two situations can be disentangled to some degree by training another classifier to distinguish between an unweighted distribution of samples from $p(\mathbf{x}|\theta_1)$.



(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.

Figure 5: Results from the diagnostics described in Sec. 3.5. The rows correspond to the quality of the training and calibration of the classifier. The left plots probe the sensitivity to θ_1, while the right plots show the ROC curve for a calibrator trained to discriminate samples from $p(\mathbf{x}|\theta_0)$ and samples from $p(\mathbf{x}|\theta_1)$ weighted as indicated in the legend.

K.C., G. Louppe, J. Pavez: Approximating Likelihood Ratios with Calibrated Discriminative Classifiers [arXiv:1506.02169].

See also: 2016 talk on reweighting at the DS@HEP

Parameter dependence in Neural Likelihood Ratio Estimation

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / “non-parametric” (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Parameter dependence

Say we want to model either $p(x|\theta)$ or $r(x|\theta) = \dfrac{p(x|\theta)}{p_{\text{ref}}(x)}$.

A few approaches that change structure of the model

- **Point-by-Point**: model $p(x|\theta_i)$ for a set of points $\{\theta_i\}$
  - Not explicitly parametrized in θ, no structure
- **Parametrized Network:** NN models both x-dependence and θ-dependence
  - Most flexible, but doesn't exploit any physics knowledge
- **Fixed Interpolation:** multiple NNs model x-dependence, but form of θ-dependence is fixed & defined by physicist
  - e.g. this is possible for EFT coefficients (exact)
  - This is what HistFactory etc. do for nuisance parameters but this makes assumptions

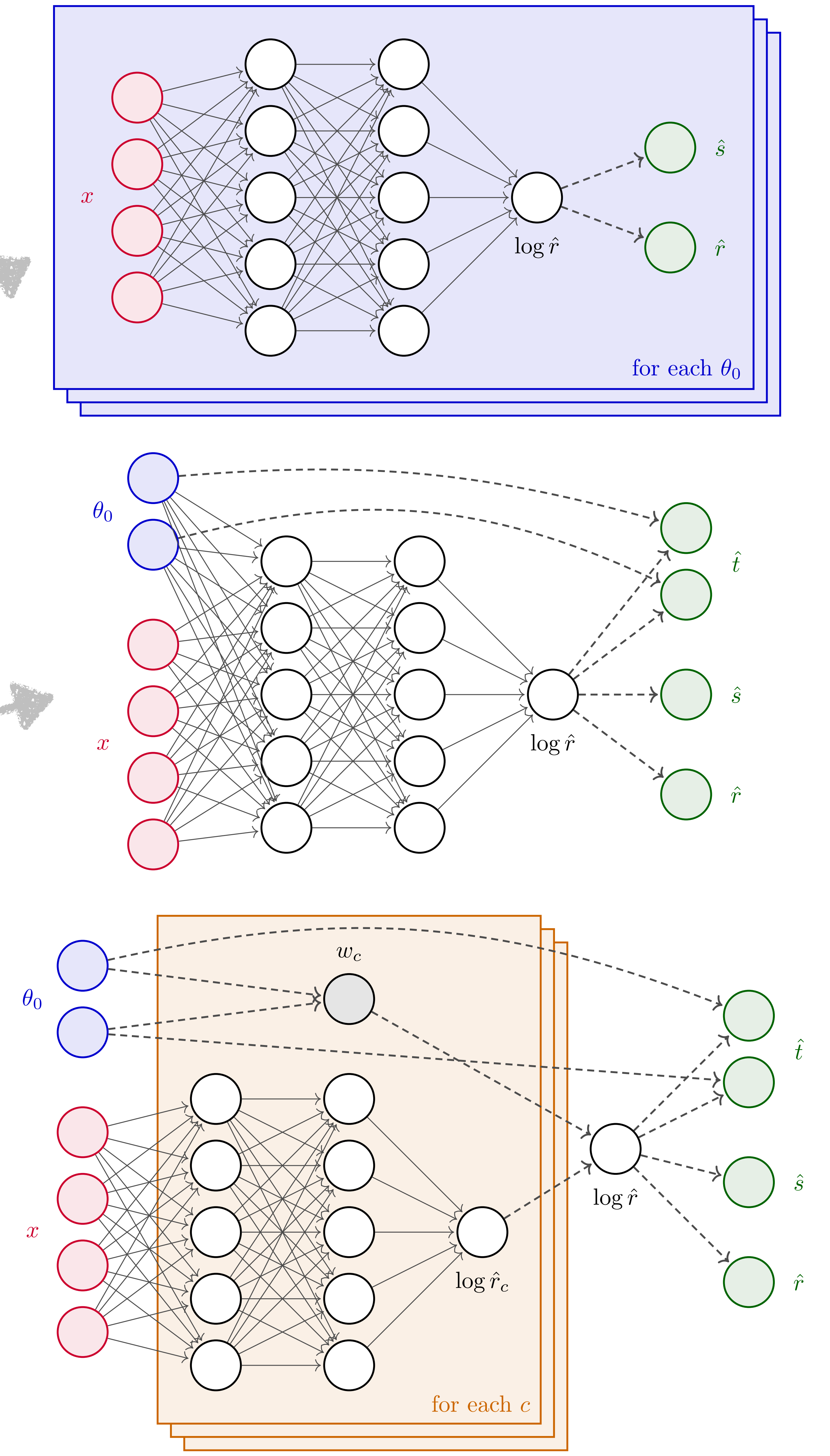


Figure 8: Schematic neural network architectures for point-by-point (top), agnostic parameterized (middle), and morphing-aware parameterized (bottom) estimators. Solid lines denote dependencies with learnable weights, dashed lines show fixed functional dependencies.

Fig from: https://arxiv.org/abs/1805.00020

Curse of dimensionality for nuisance parameters

The traditional binned-template analysis approach uses a fixed interpolation / "template morphing" strategy

- Dependence on the parameters of interest are usually very well motivated
- makes assumptions about factorization of systematics that might not be true
- … either way, fixed parametric form makes it VERY sample efficient

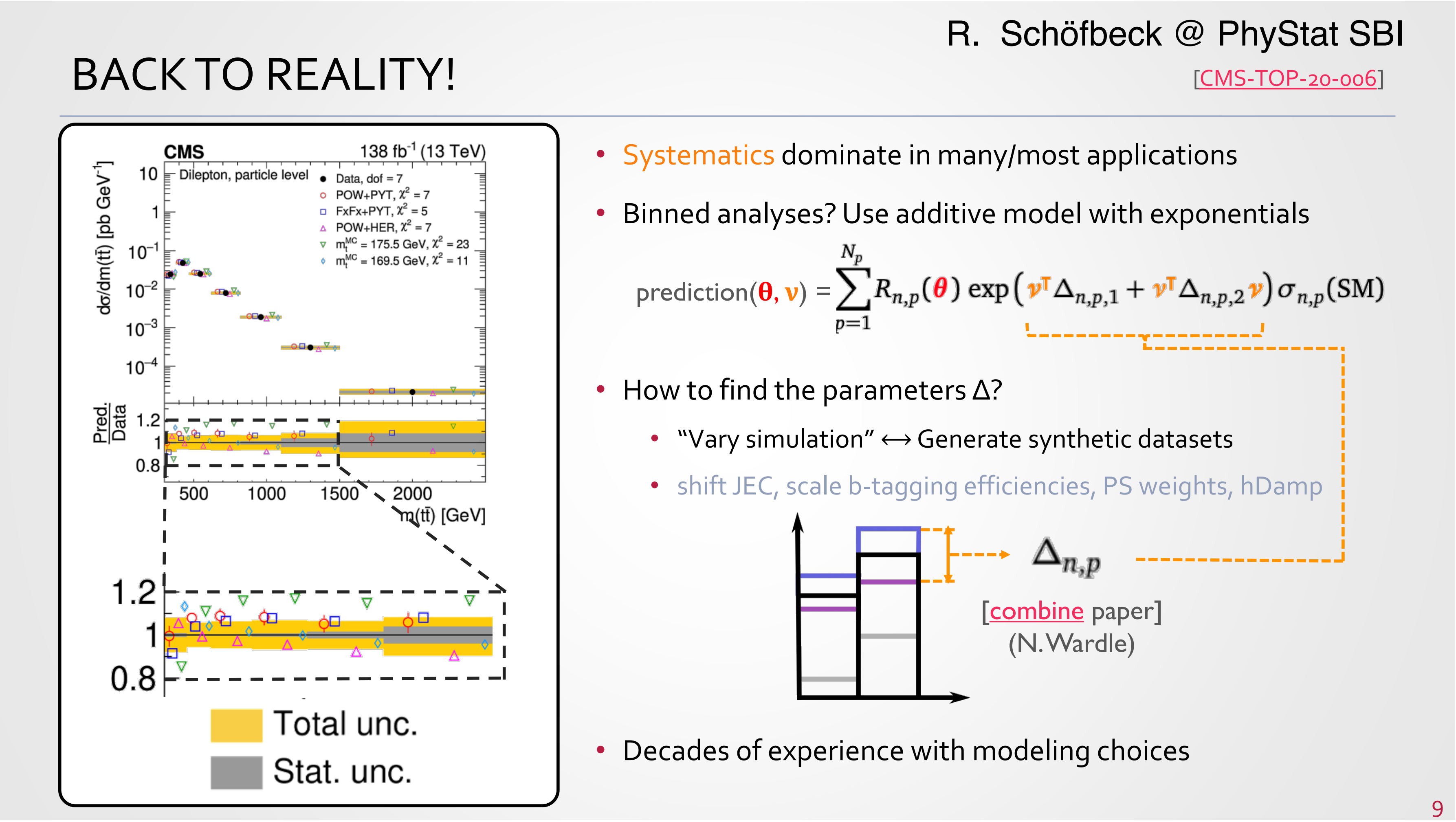
In contrast, parametrized NN is physics-agnostic and the interpolation is non-parametric

- Flexible, but requires many samples for a high-dimensional nuisance parameter space
- Curse of dimensionality

Is there a way to apply similar assumptions as template-based morphing strategy in neural SBI context?

Recent developments shown at PhyStat SBI

R. Schöfbeck & Jay Sandesara both showed work in that direction at PhyStat SBI



R. Schöfbeck @ PhyStat SBI

[CMS-TOP-20-006]

BACK TO REALITY!

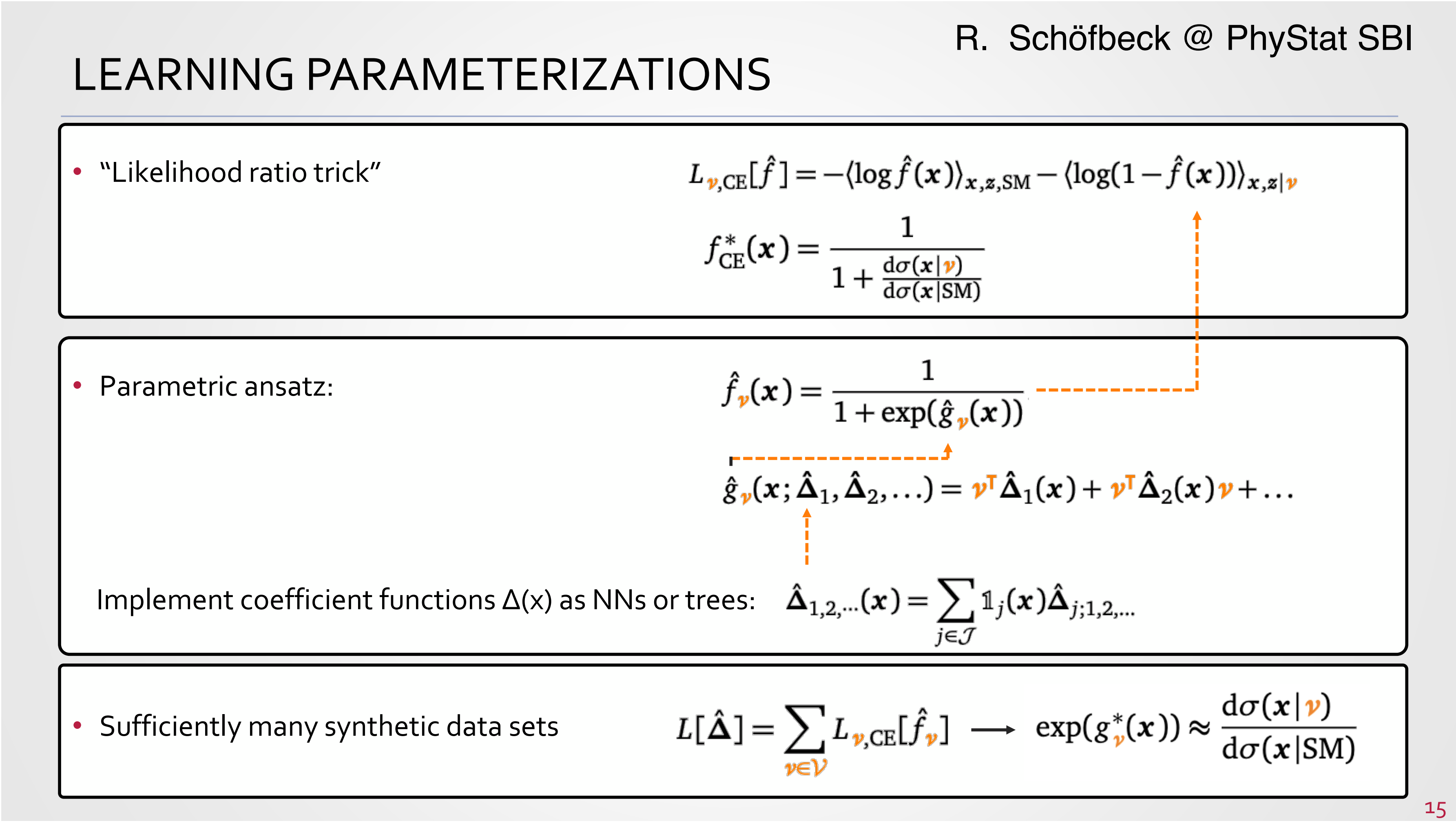
- Systematics dominate in many/most applications
- Binned analyses? Use additive model with exponentials

$$\text{prediction}(\boldsymbol{\theta}, \boldsymbol{\nu}) = \sum_{p=1}^{N_p} R_{n,p}(\boldsymbol{\theta}) \exp\left(\boldsymbol{\nu}^{\mathsf{T}}\Delta_{n,p,1} + \boldsymbol{\nu}^{\mathsf{T}}\Delta_{n,p,2}\boldsymbol{\nu}\right)\sigma_{n,p}(\text{SM})$$

- How to find the parameters Δ?
  - "Vary simulation" ⟷ Generate synthetic datasets
  - shift JEC, scale b-tagging efficiencies, PS weights, hDamp

[combine paper]
(N.Wardle)

- Decades of experience with modeling choices



R. Schöfbeck @ PhyStat SBI

LEARNING PARAMETERIZATIONS

- "Likelihood ratio trick"

$$L_{\boldsymbol{\nu},\text{CE}}[\hat{f}] = -\langle \log \hat{f}(\boldsymbol{x})\rangle_{\boldsymbol{x},\boldsymbol{z},\text{SM}} - \langle \log(1-\hat{f}(\boldsymbol{x}))\rangle_{\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\nu}}$$

$$f^*_{\text{CE}}(\boldsymbol{x}) = \frac{1}{1+\frac{d\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{d\sigma(\boldsymbol{x}|\text{SM})}}$$

- Parametric ansatz:

$$\hat{f}_{\boldsymbol{\nu}}(\boldsymbol{x}) = \frac{1}{1+\exp(\hat{g}_{\boldsymbol{\nu}}(\boldsymbol{x}))}$$

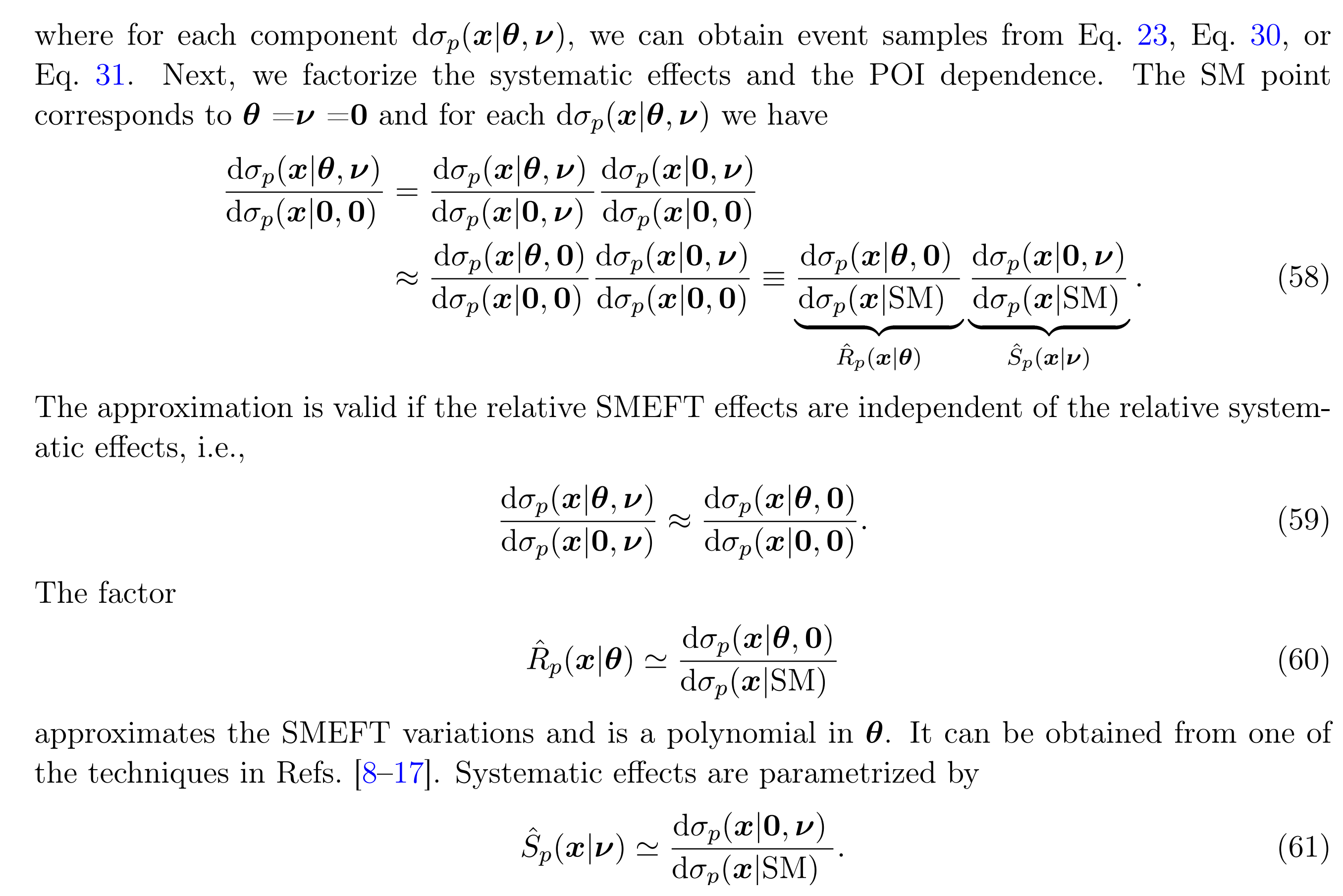
$$\hat{g}_{\boldsymbol{\nu}}(\boldsymbol{x}; \hat{\boldsymbol{\Delta}}_1, \hat{\boldsymbol{\Delta}}_2, \ldots) = \boldsymbol{\nu}^{\mathsf{T}}\hat{\boldsymbol{\Delta}}_1(\boldsymbol{x}) + \boldsymbol{\nu}^{\mathsf{T}}\hat{\boldsymbol{\Delta}}_2(\boldsymbol{x})\boldsymbol{\nu} + \ldots$$

Implement coefficient functions Δ(x) as NNs or trees: $\hat{\boldsymbol{\Delta}}_{1,2,\ldots}(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \mathbb{1}_j(\boldsymbol{x})\hat{\boldsymbol{\Delta}}_{j;1,2,\ldots}$

- Sufficiently many synthetic data sets $L[\hat{\boldsymbol{\Delta}}] = \sum_{\boldsymbol{\nu}\in\mathcal{V}} L_{\boldsymbol{\nu},\text{CE}}[\hat{f}_{\boldsymbol{\nu}}] \longrightarrow \exp(g^*_{\boldsymbol{\nu}}(\boldsymbol{x})) \approx \frac{d\sigma(\boldsymbol{x}|\boldsymbol{\nu})}{d\sigma(\boldsymbol{x}|\text{SM})}$

15

- **Basic idea:** use same interpolation point wise in x, replace histogram bin with function of x.



where for each component $d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$, we can obtain event samples from Eq. 23, Eq. 30, or Eq. 31. Next, we factorize the systematic effects and the POI dependence. The SM point corresponds to $\boldsymbol{\theta} = \boldsymbol{\nu} = \mathbf{0}$ and for each $d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})$ we have

$$\begin{aligned}\frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\mathbf{0})} &= \frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\boldsymbol{\nu})}\frac{d\sigma_p(\boldsymbol{x}|\mathbf{0},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\mathbf{0})} \\ &\approx \frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\mathbf{0})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\mathbf{0})}\frac{d\sigma_p(\boldsymbol{x}|\mathbf{0},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\mathbf{0})} \equiv \underbrace{\frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\mathbf{0})}{d\sigma_p(\boldsymbol{x}|\text{SM})}}_{\hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta})}\underbrace{\frac{d\sigma_p(\boldsymbol{x}|\mathbf{0},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\text{SM})}}_{\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})}.\end{aligned} \tag{58}$$

The approximation is valid if the relative SMEFT effects are independent of the relative systematic effects, i.e.,

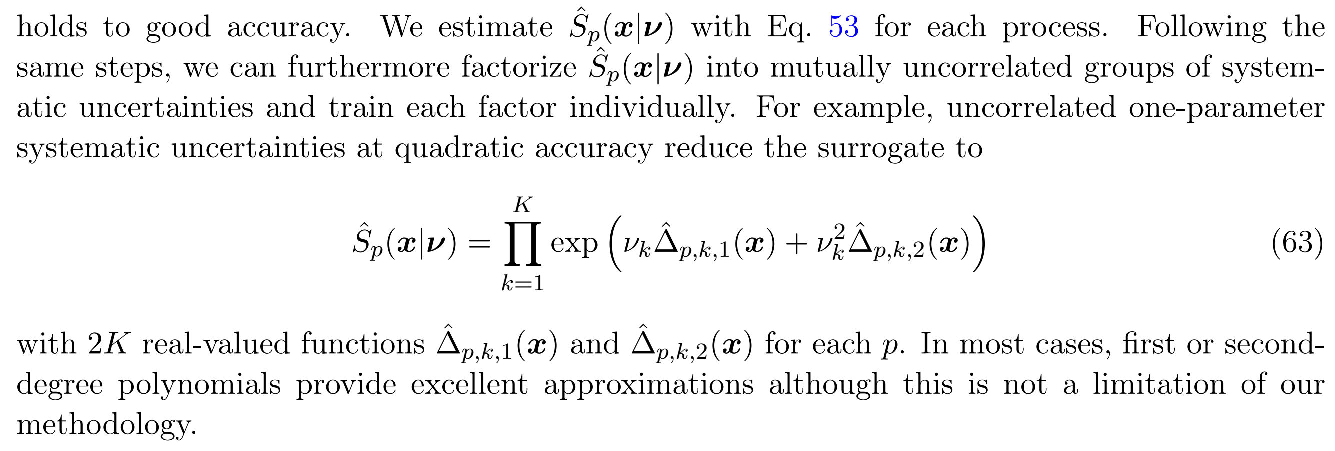
$$\frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\boldsymbol{\nu})} \approx \frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\mathbf{0})}{d\sigma_p(\boldsymbol{x}|\mathbf{0},\mathbf{0})}. \tag{59}$$

The factor

$$\hat{R}_p(\boldsymbol{x}|\boldsymbol{\theta}) \simeq \frac{d\sigma_p(\boldsymbol{x}|\boldsymbol{\theta},\mathbf{0})}{d\sigma_p(\boldsymbol{x}|\text{SM})} \tag{60}$$

approximates the SMEFT variations and is a polynomial in $\boldsymbol{\theta}$. It can be obtained from one of the techniques in Refs. [8–17]. Systematic effects are parametrized by

$$\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu}) \simeq \frac{d\sigma_p(\boldsymbol{x}|\mathbf{0},\boldsymbol{\nu})}{d\sigma_p(\boldsymbol{x}|\text{SM})}. \tag{61}$$

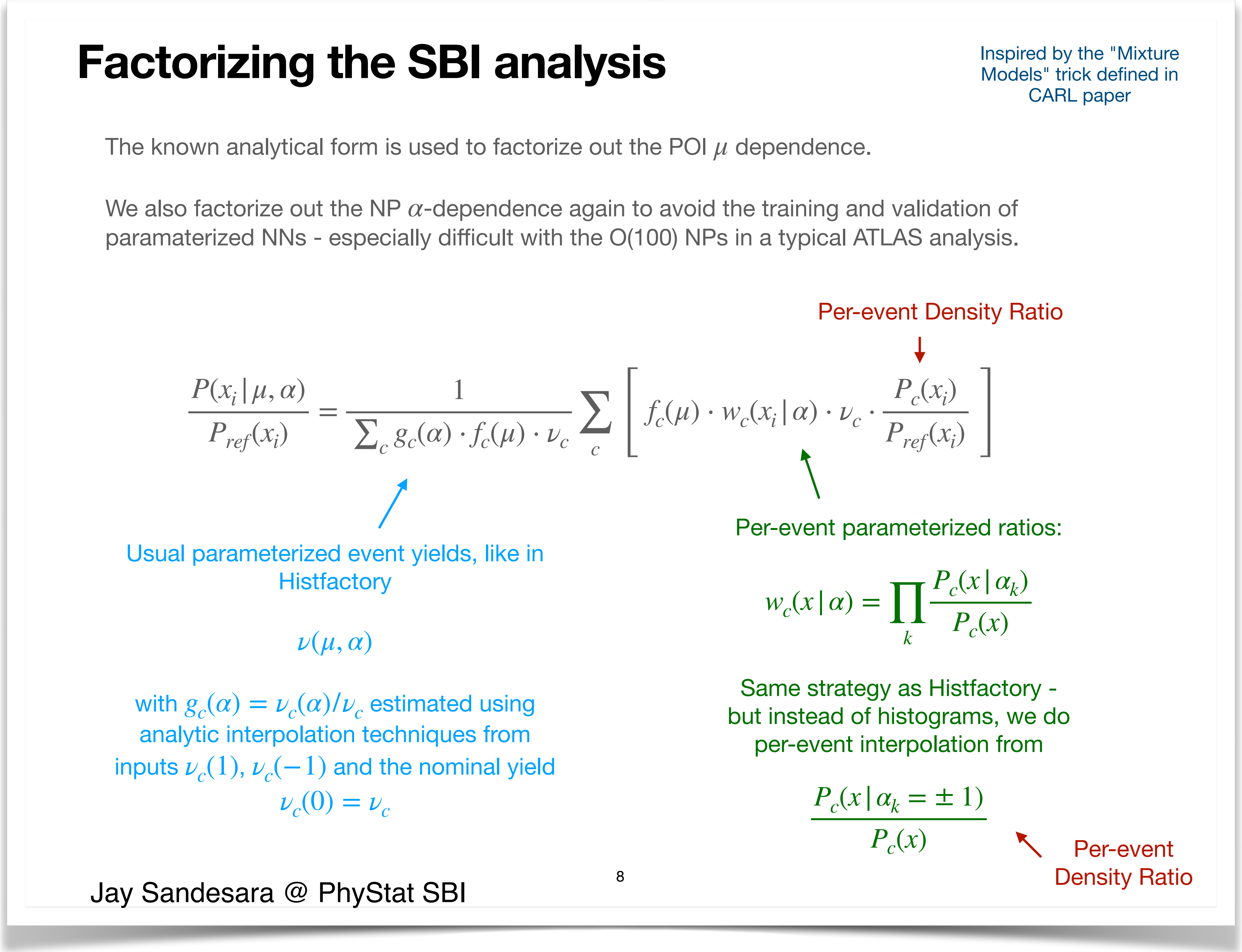


holds to good accuracy. We estimate $\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})$ with Eq. 53 for each process. Following the same steps, we can furthermore factorize $\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu})$ into mutually uncorrelated groups of systematic uncertainties and train each factor individually. For example, uncorrelated one-parameter systematic uncertainties at quadratic accuracy reduce the surrogate to

$$\hat{S}_p(\boldsymbol{x}|\boldsymbol{\nu}) = \prod_{k=1}^{K} \exp\left(\nu_k\hat{\Delta}_{p,k,1}(\boldsymbol{x}) + \nu_k^2\hat{\Delta}_{p,k,2}(\boldsymbol{x})\right) \tag{63}$$

with $2K$ real-valued functions $\hat{\Delta}_{p,k,1}(\boldsymbol{x})$ and $\hat{\Delta}_{p,k,2}(\boldsymbol{x})$ for each p. In most cases, first or second-degree polynomials provide excellent approximations although this is not a limitation of our methodology.

https://arxiv.org/abs/2406.19076



Factorizing the SBI analysis

Inspired by the "Mixture Models" trick defined in CARL paper

The known analytical form is used to factorize out the POI μ dependence.

We also factorize out the NP α-dependence again to avoid the training and validation of paramaterized NNs - especially difficult with the O(100) NPs in a typical ATLAS analysis.

Per-event Density Ratio

$$\frac{P(x_i|\mu,\alpha)}{P_{ref}(x_i)} = \frac{1}{\sum_c g_c(\alpha)\cdot f_c(\mu)\cdot \nu_c}\sum_c\left[f_c(\mu)\cdot w_c(x_i|\alpha)\cdot \nu_c \cdot \frac{P_c(x_i)}{P_{ref}(x_i)}\right]$$

Usual parameterized event yields, like in Histfactory

$\nu(\mu,\alpha)$

with $g_c(\alpha) = \nu_c(\alpha)/\nu_c$ estimated using analytic interpolation techniques from inputs $\nu_c(1)$, $\nu_c(-1)$ and the nominal yield $\nu_c(0) = \nu_c$

Per-event parameterized ratios:

$$w_c(x|\alpha) = \prod_k \frac{P_c(x|\alpha_k)}{P_c(x)}$$

Same strategy as Histfactory - but instead of histograms, we do per-event interpolation from

$$\frac{P_c(x|\alpha_k = \pm 1)}{P_c(x)}$$

Per-event Density Ratio

Jay Sandesara @ PhyStat SBI

8

Population-level / Experiment-wide Neural SBI

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Learning to profile

https://arxiv.org/abs/2203.13079
https://arxiv.org/abs/2306.12584

In the fully-parametrized neural SBI approach, one must learn the dependence of the [likelihood, likelihood ratio, posterior] as a function of the parameters of interest and the nuisance parameters

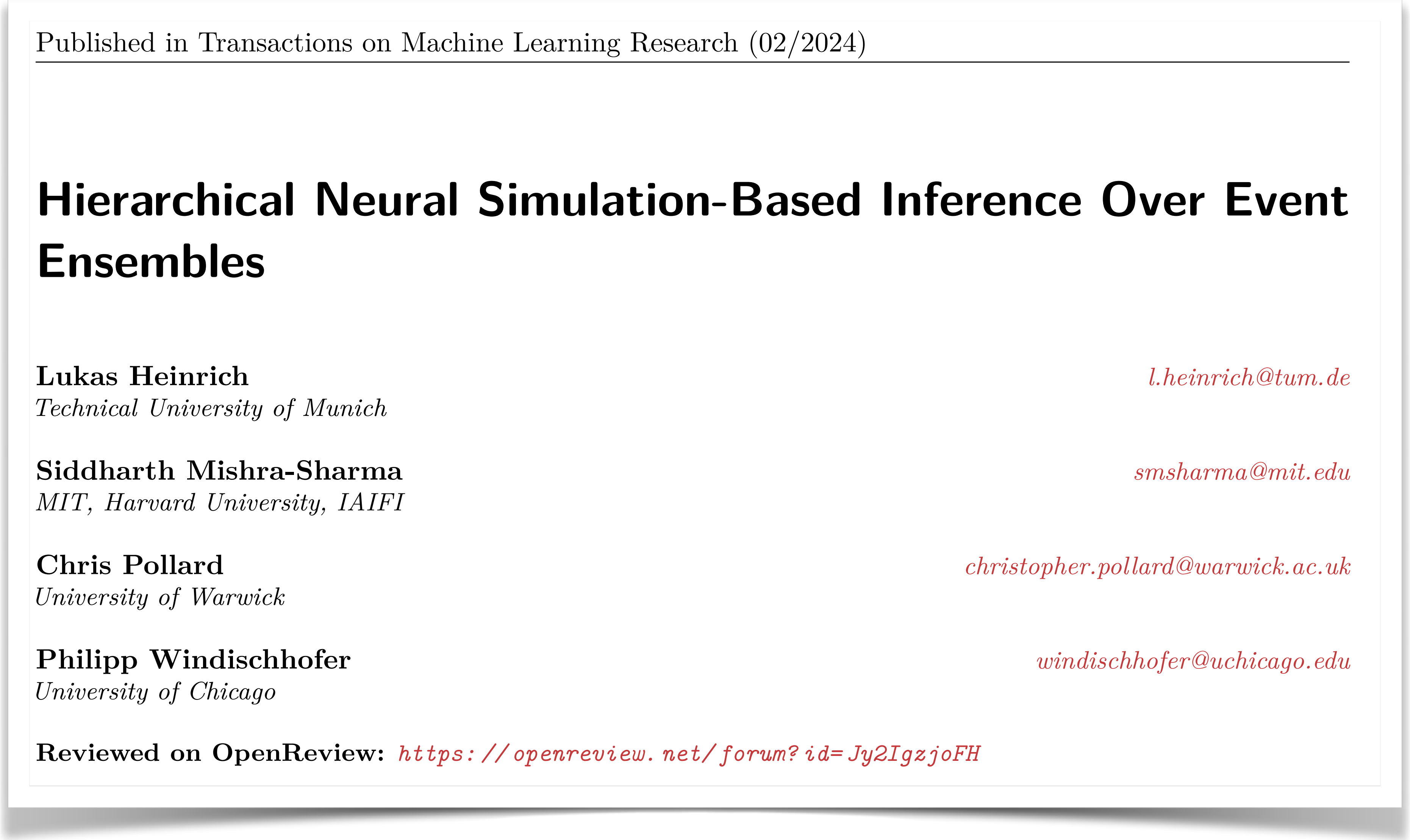
- Then one **eliminates dependence on nuisance parameters** through profiling or marginalization in the standard way
- This is conceptually clean, but computationally difficult with many nuisance parameters

Ultimately, profiled value of nuisance parameters is a function of POI and the full data set - viz. $\hat{\hat{\nu}}(\{x_i\}, \mu)$

- Can we "learn to profile" and just learn the profile likelihood ratio directly?
- Is this easier? Function now depends on the entire dataset

Learning Optimal Test Statistics in the Presence of Nuisance Parameters

Lukas Heinrich
Technical University of Munich



Published in Transactions on Machine Learning Research (02/2024)

Hierarchical Neural Simulation-Based Inference Over Event Ensembles

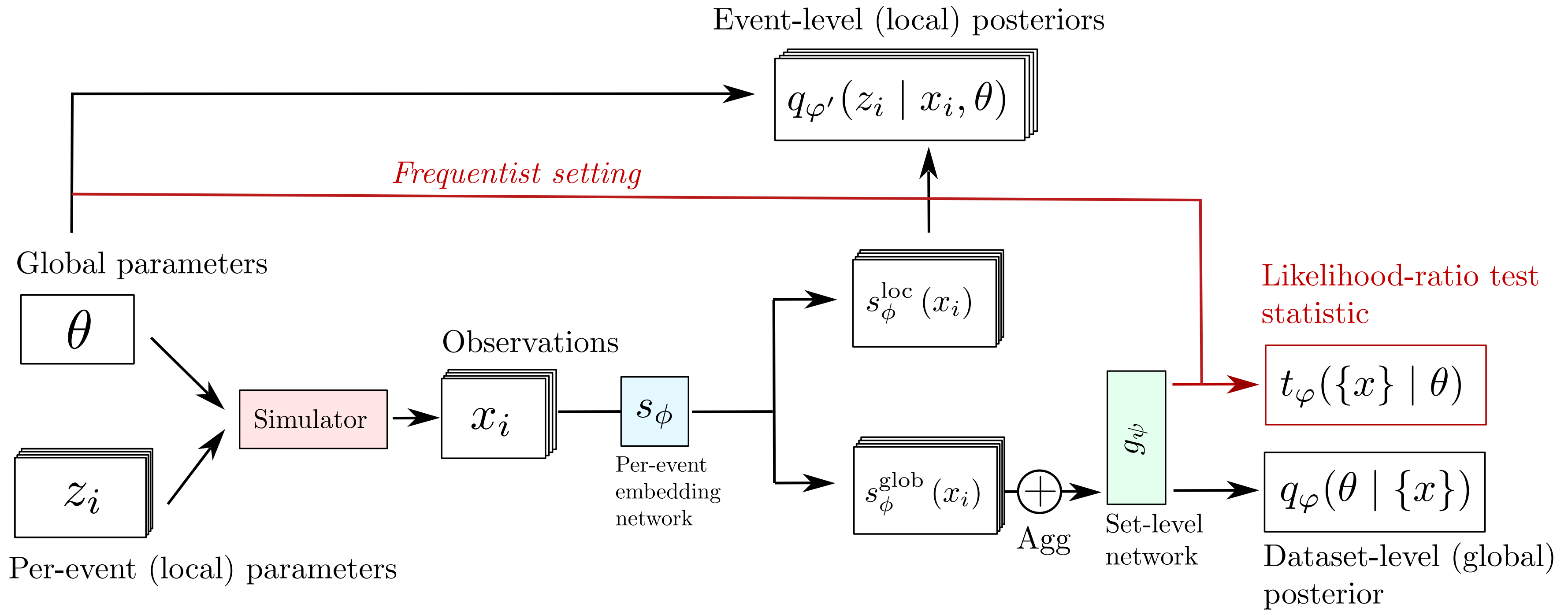
Lukas Heinrich l.heinrich@tum.de
Technical University of Munich

Siddharth Mishra-Sharma smsharma@mit.edu
MIT, Harvard University, IAIFI

Chris Pollard christopher.pollard@warwick.ac.uk
University of Warwick

Philipp Windischhofer windischhofer@uchicago.edu
University of Chicago

Reviewed on OpenReview: https://openreview.net/forum?id=Jy2IgzjoFH


Event-level (local) posteriors
$q_{\varphi'}(z_i \mid x_i, \theta)$
Frequentist setting
Global parameters
θ
z_i
Per-event (local) parameters
Simulator
Observations
x_i
s_ϕ
Per-event embedding network
$s_\phi^{\mathrm{loc}}(x_i)$
$s_\phi^{\mathrm{glob}}(x_i)$
Agg
g_ψ
Set-level network
Likelihood-ratio test statistic
$t_\varphi(\{x\} \mid \theta)$
$q_\varphi(\theta \mid \{x\})$
Dataset-level (global) posterior

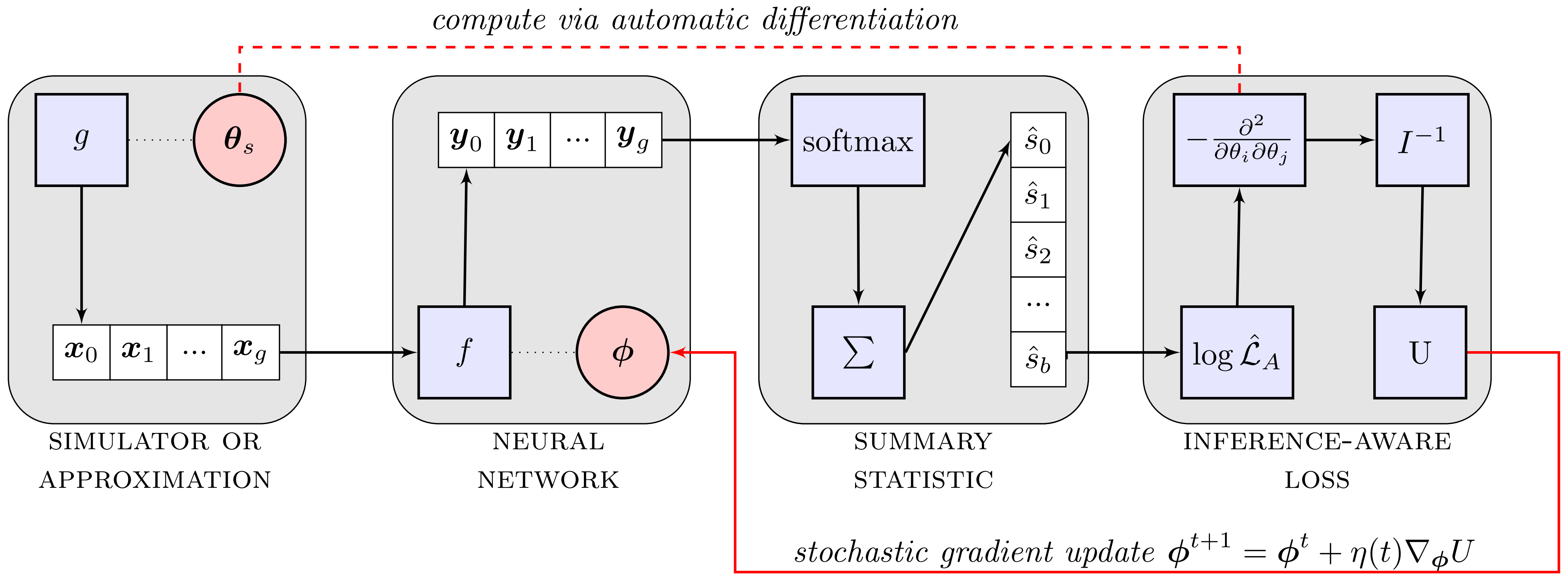
Learning Summary Statistics In the Presence of Systematics

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event	Experiment-wide

Tradition meets differentiable programming

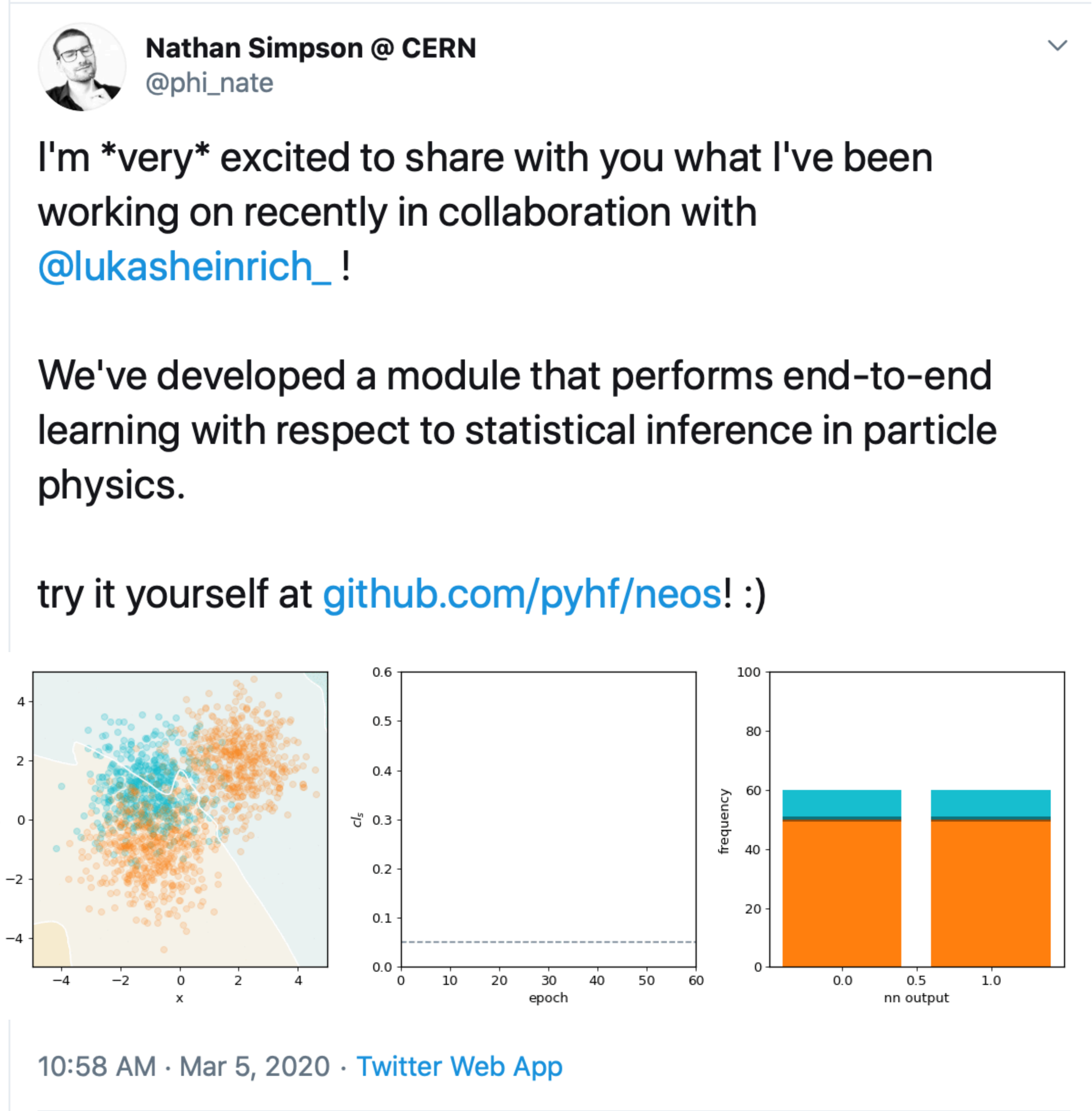
Recent efforts in particle physics to maintain traditional approaches to likelihood estimation with summaries, but optimize summary statistics with automatic differentiation

- Connects to differentiable programming paradigm
- Optimization objective is power of full statistical analysis, which involves backproping through statistical procedure
- Does not exploit i.i.d. property, optimization is "global"



compute via automatic differentiation
g
θ_s
x_0 x_1 ... x_g
y_0 y_1 ... y_g
f
ϕ
softmax
$\sum$
$\hat{s}_0$ $\hat{s}_1$ $\hat{s}_2$ ... $\hat{s}_b$
$-\frac{\partial^2}{\partial\theta_i\partial\theta_j}$
I^{-1}
$\log \hat{\mathcal{L}}_A$
U
SIMULATOR OR APPROXIMATION
NEURAL NETWORK
SUMMARY STATISTIC
INFERENCE-AWARE LOSS
stochastic gradient update $\phi^{t+1} = \phi^t + \eta(t)\nabla_\phi U$

INFERNO: de Castro & Dorigo, [arXiv:1806.04743]



Nathan Simpson @ CERN
@phi_nate

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich_ !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)

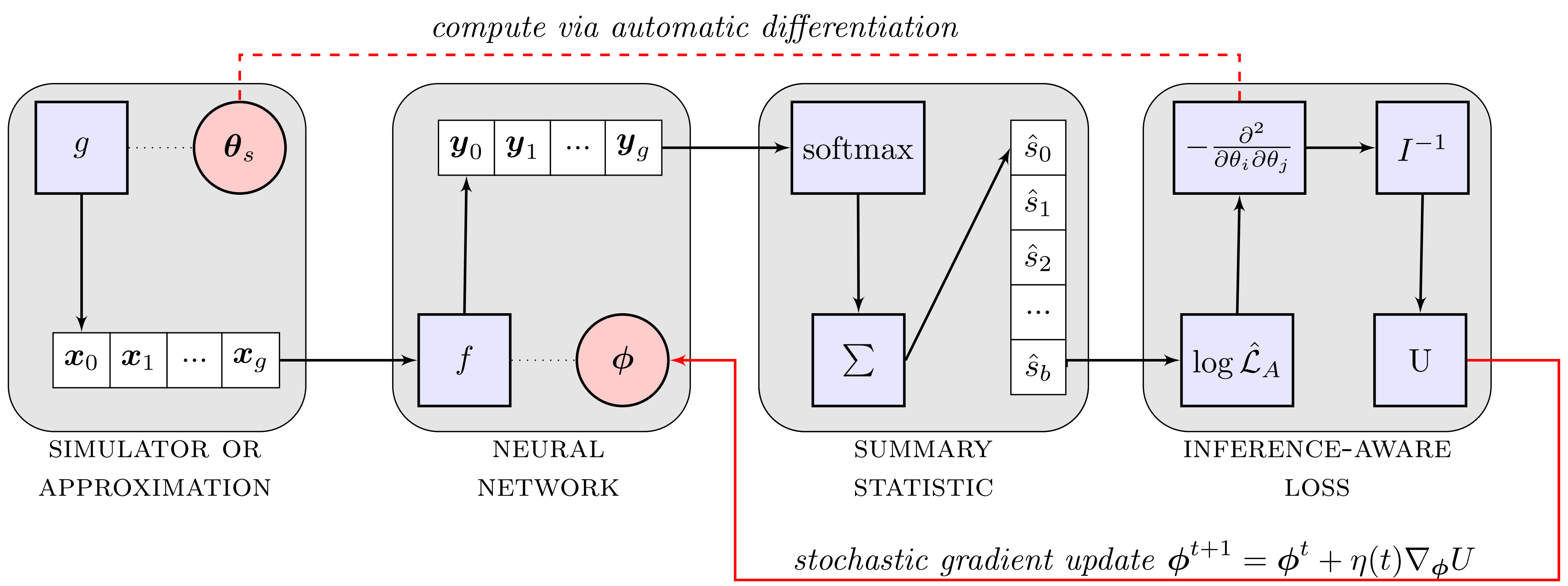
10:58 AM · Mar 5, 2020 · Twitter Web App

https://github.com/pyhf/neos

Tradition meets differentiable programming

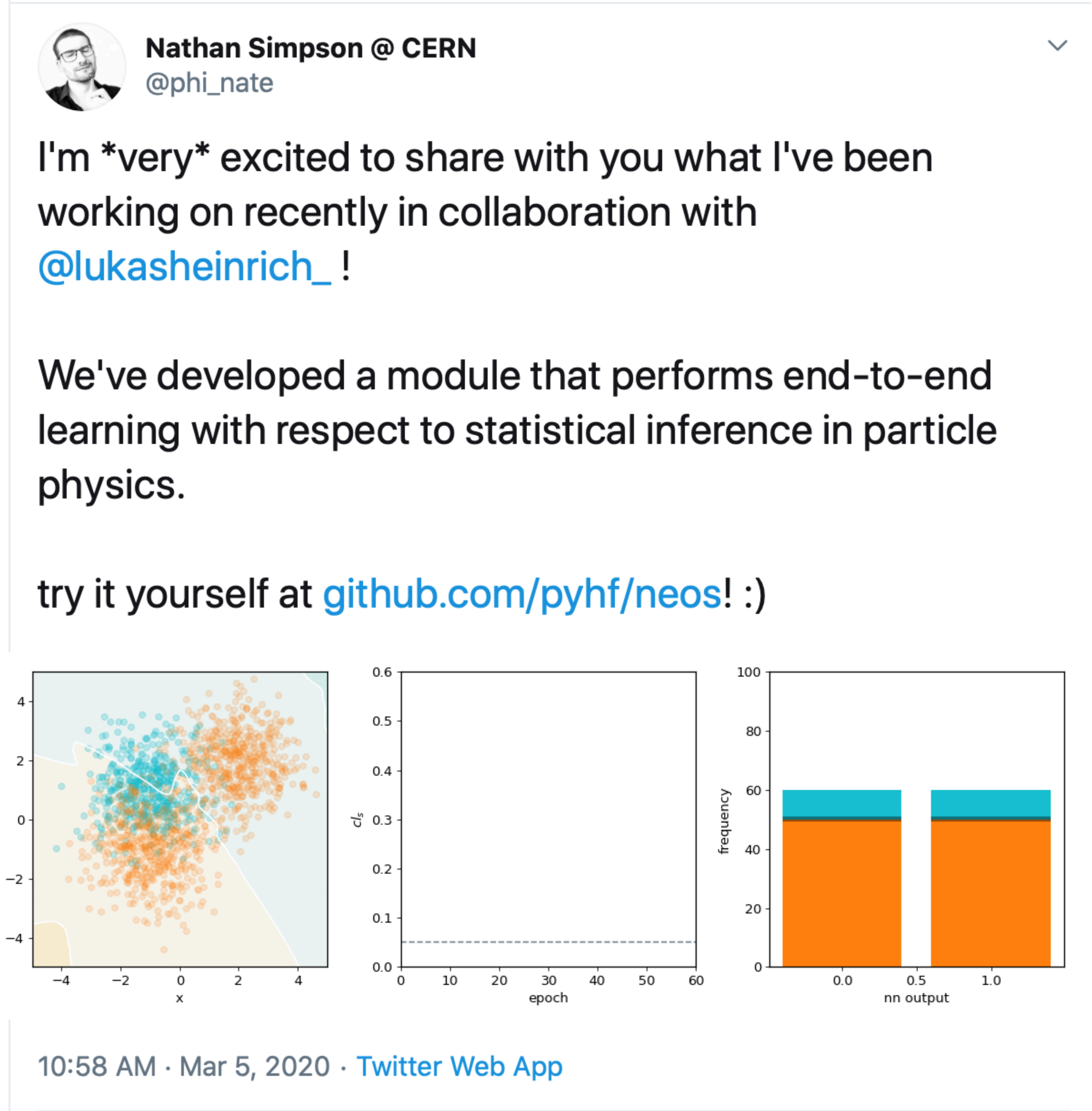
Recent efforts in particle physics to maintain traditional approaches to likelihood estimation with summaries, but optimize summary statistics with automatic differentiation

- Connects to differentiable programming paradigm
- Optimization objective is power of full statistical analysis, which involves backproping through statistical procedure
- Does not exploit i.i.d. property, optimization is “global”



compute via automatic differentiation
g
θ_s
x_0 x_1 … x_g
y_0 y_1 … y_g
f
ϕ
softmax
Σ
$\hat{s}_0$ $\hat{s}_1$ $\hat{s}_2$ … $\hat{s}_b$
$-\frac{\partial^2}{\partial\theta_i\partial\theta_j}$
I^{-1}
$\log\hat{\mathcal{L}}_A$
U
SIMULATOR OR APPROXIMATION
NEURAL NETWORK
SUMMARY STATISTIC
INFERENCE-AWARE LOSS
stochastic gradient update $\phi^{t+1} = \phi^t + \eta(t)\nabla_\phi U$

INFERNO: de Castro & Dorigo, [arXiv:1806.04743]



Nathan Simpson @ CERN
@phi_nate

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich_ !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)

10:58 AM · Mar 5, 2020 · Twitter Web App

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

neos

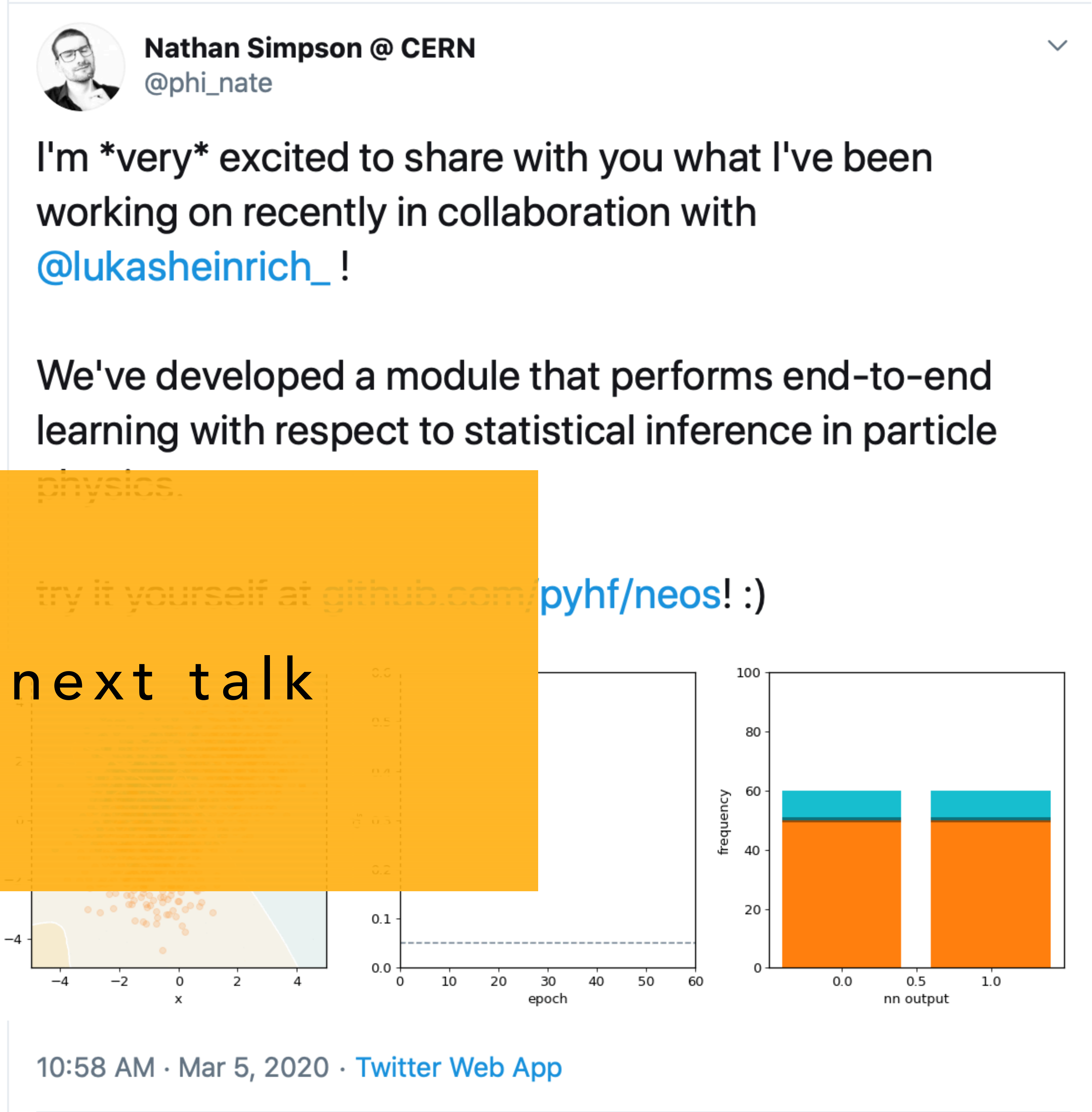
https://github.com/pyhf/neos

Tradition meets differentiable programming

Recent efforts in particle physics to maintain traditional approaches to likelihood estimation with summaries, but optimize summary statistics with automatic differentiation

- Connects to differentiable programming paradigm
- Optimization objective is power of full statistical analysis, which involves backproping through statistical procedure
- Does not exploit i.i.d. property, optimization is "global"

Similar point by Artur Mensch with SANNT in next talk



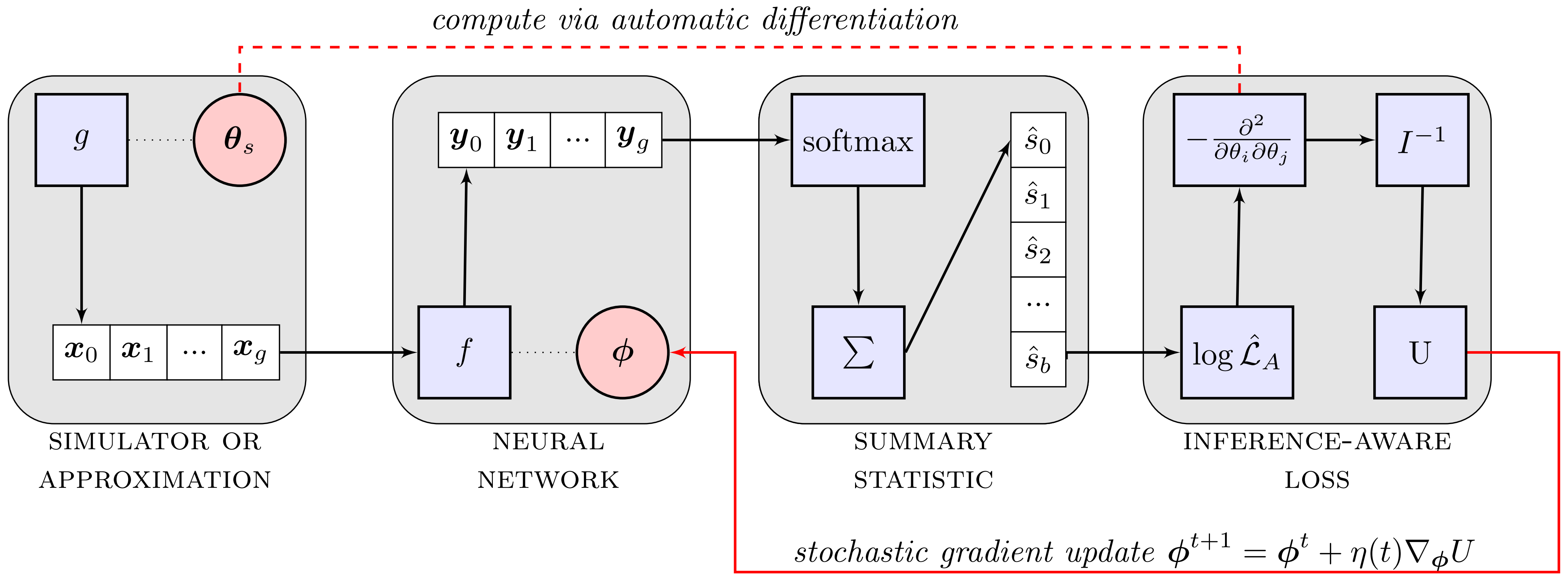
Nathan Simpson @ CERN
@phi_nate

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich_ !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)

10:58 AM · Mar 5, 2020 · Twitter Web App

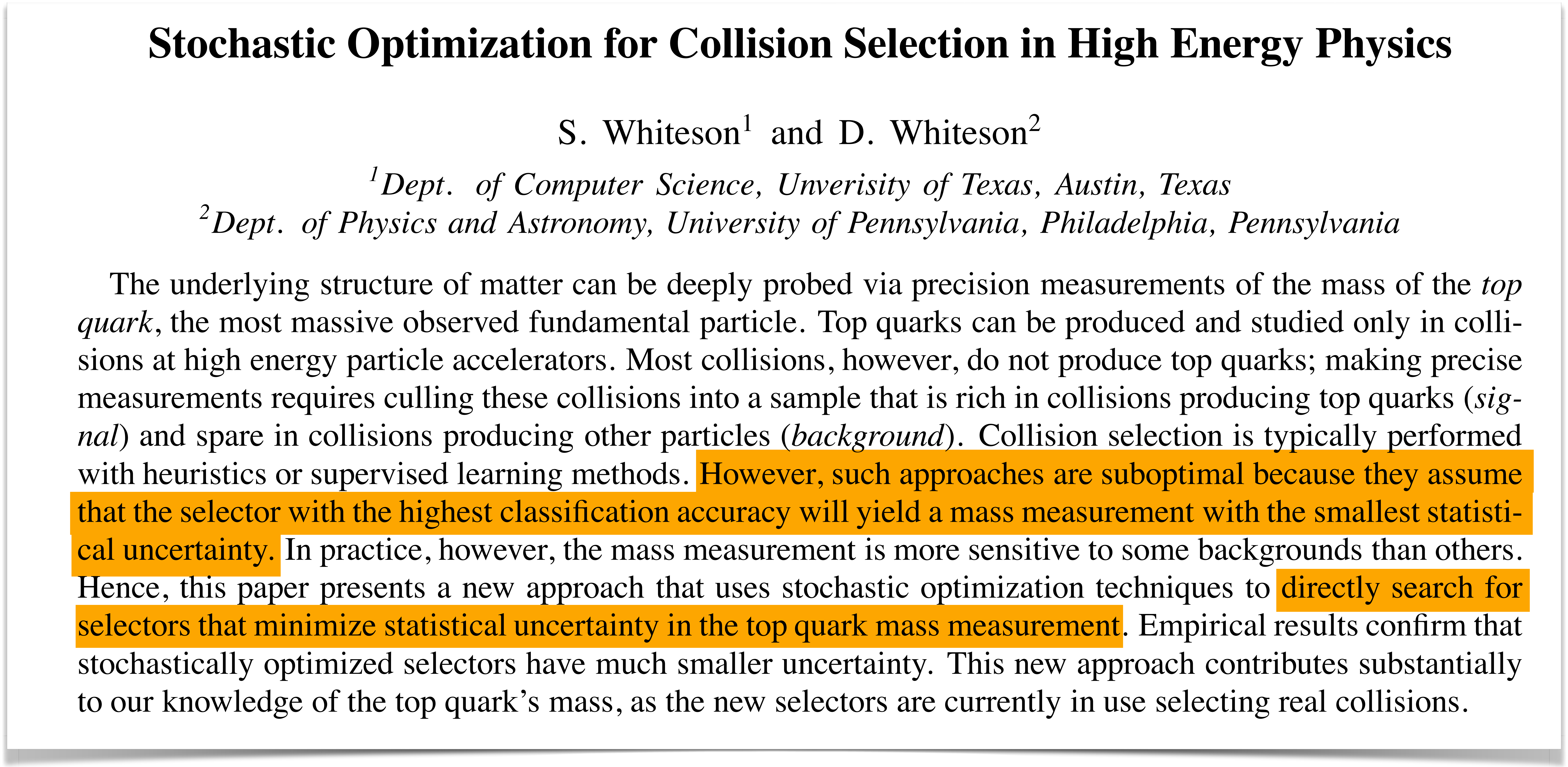


compute via automatic differentiation
g
θ_s
x_0 x_1 … x_g
y_0 y_1 … y_g
f
ϕ
softmax
Σ
$\hat{s}_0$ $\hat{s}_1$ $\hat{s}_2$ … $\hat{s}_b$
$-\frac{\partial^2}{\partial\theta_i\partial\theta_j}$
I^{-1}
$\log \hat{\mathcal{L}}_A$
U
SIMULATOR OR APPROXIMATION
NEURAL NETWORK
SUMMARY STATISTIC
INFERENCE-AWARE LOSS
stochastic gradient update $\phi^{t+1} = \phi^t + \eta(t)\nabla_\phi U$

INFERNO: de Castro & Dorigo, [arXiv:1806.04743]

https://github.com/pyhf/neos

A similar point (2006)



Stochastic Optimization for Collision Selection in High Energy Physics

S. Whiteson[1] and D. Whiteson[2]

[1]*Dept. of Computer Science, Unverisity of Texas, Austin, Texas*
[2]*Dept. of Physics and Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania*

The underlying structure of matter can be deeply probed via precision measurements of the mass of the *top quark*, the most massive observed fundamental particle. Top quarks can be produced and studied only in collisions at high energy particle accelerators. Most collisions, however, do not produce top quarks; making precise measurements requires culling these collisions into a sample that is rich in collisions producing top quarks (*signal*) and spare in collisions producing other particles (*background*). Collision selection is typically performed with heuristics or supervised learning methods. However, such approaches are suboptimal because they assume that the selector with the highest classification accuracy will yield a mass measurement with the smallest statistical uncertainty. In practice, however, the mass measurement is more sensitive to some backgrounds than others. Hence, this paper presents a new approach that uses stochastic optimization techniques to directly search for selectors that minimize statistical uncertainty in the top quark mass measurement. Empirical results confirm that stochastically optimized selectors have much smaller uncertainty. This new approach contributes substantially to our knowledge of the top quark's mass, as the new selectors are currently in use selecting real collisions.

* Shimon Whiteson (Daniel Whiteson's brother) is now Professor of Computer Science at Oxford and the Head of Research at Waymo UK.

https://arxiv.org/abs/hep-ex/0607012

PhyStat 2003 & PhysicsGP

https://arxiv.org/abs/physics/0310110
https://arxiv.org/abs/physics/0402030

Discussed high-level strategies:

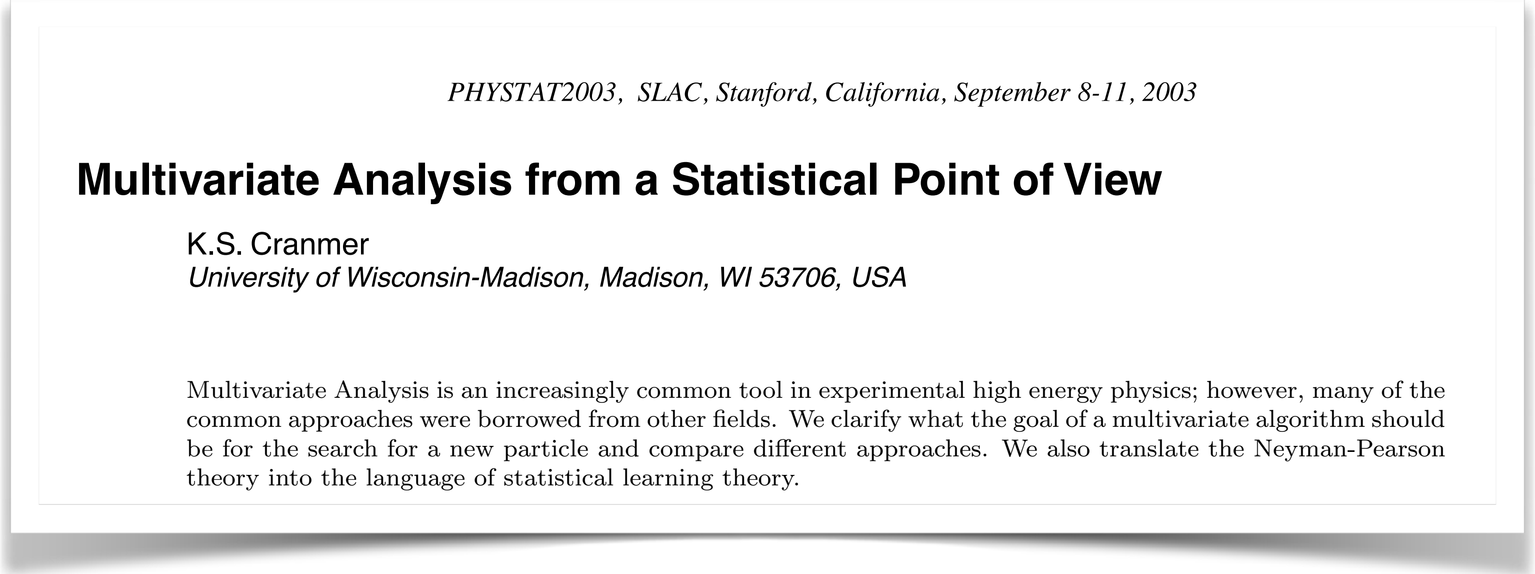
- **Indirect:** Optimize an objective that yields a well-motivated function (e.g. approximate likelihood ratio)
  - Argue that the resulting classifier should be close to optimal
- **Direct:** optimize expected discovery significance
  - **Objective includes systematic uncertainty!**

Genetic Programming / Symbolic Regression:

- Search through space of cuts / summary statistics expressed symbolically using genetic programming
- Interpretable. Less prone to overfitting (low VC dimension)

Hypothesis Testing & Statistical Learning Theory:

- Expressed Neyman-Pearson in terms of Risk
- Discussion of VC dimension for NNs, SVMs, symbolic cuts

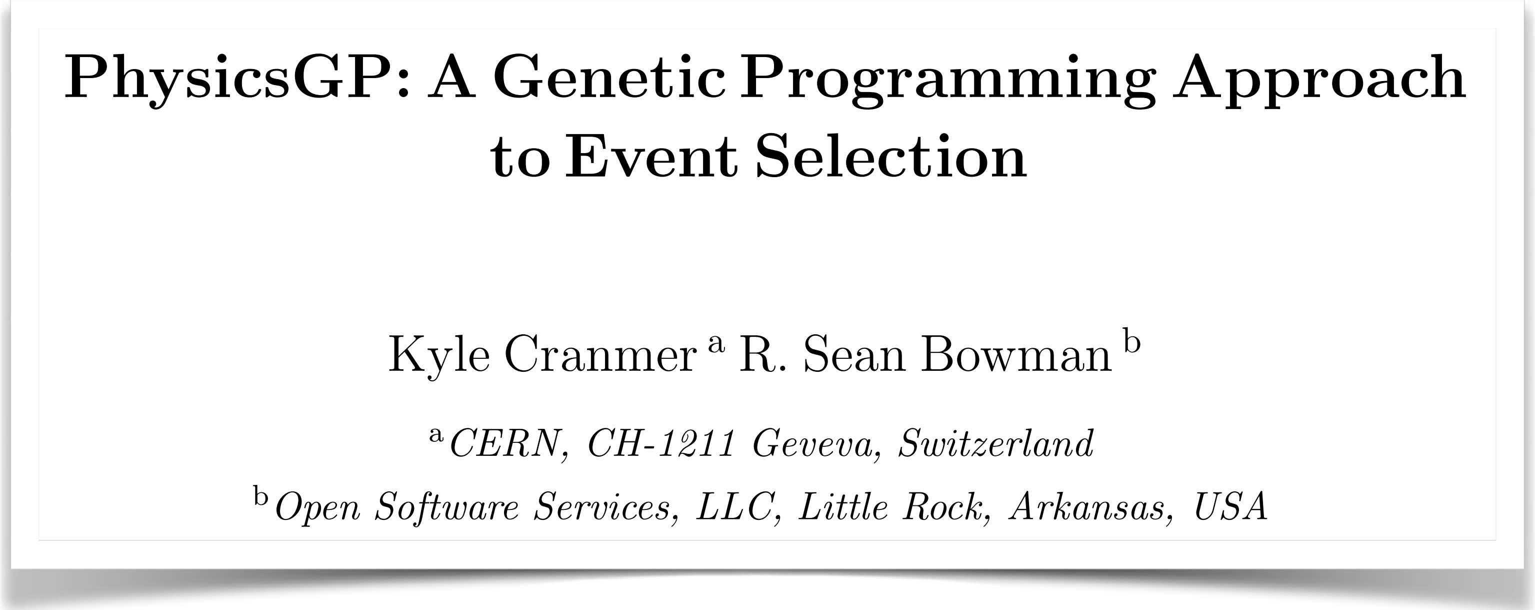


PHYSTAT2003, SLAC, Stanford, California, September 8-11, 2003

Multivariate Analysis from a Statistical Point of View

K.S. Cranmer
University of Wisconsin-Madison, Madison, WI 53706, USA

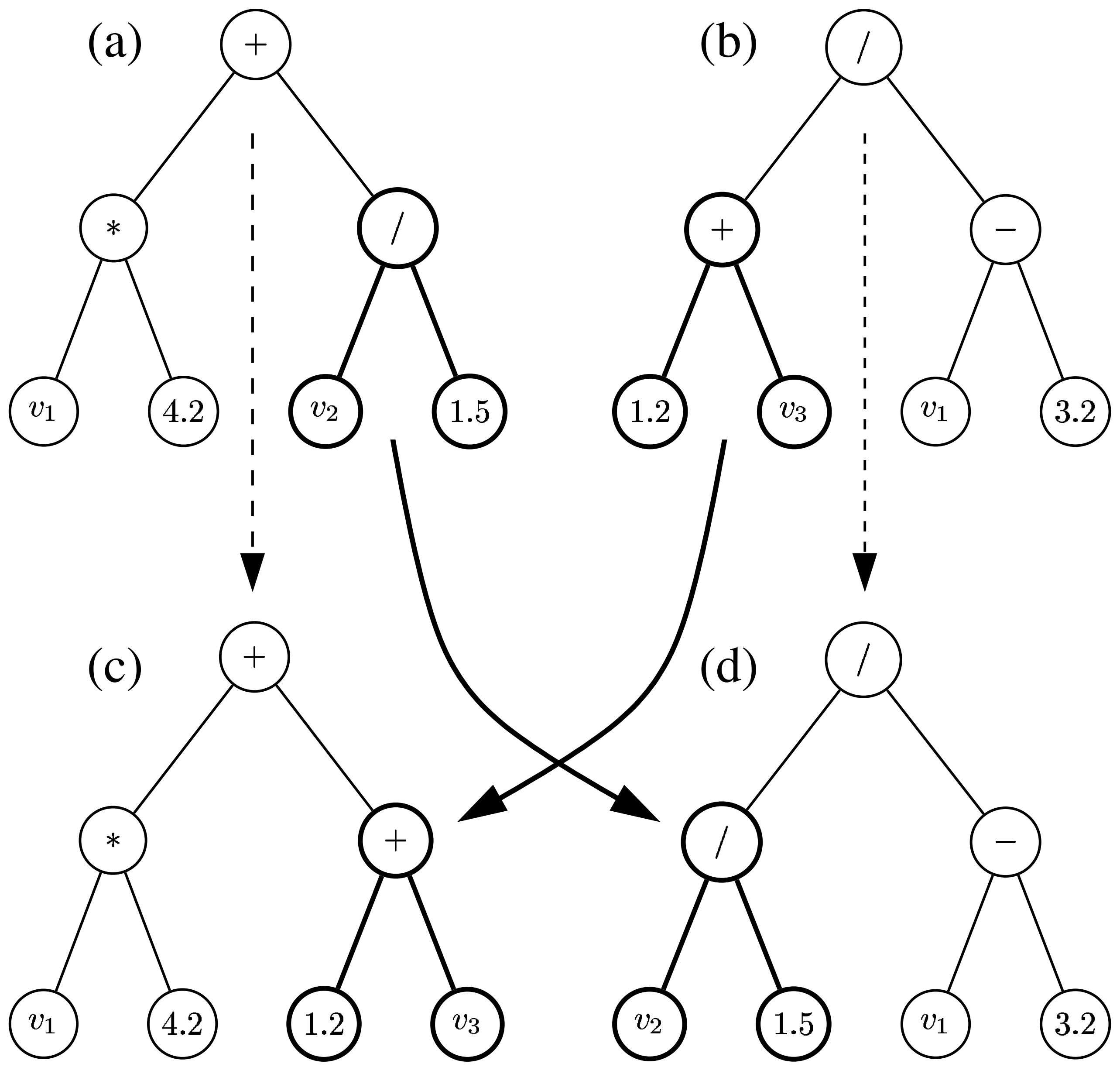
Multivariate Analysis is an increasingly common tool in experimental high energy physics; however, many of the common approaches were borrowed from other fields. We clarify what the goal of a multivariate algorithm should be for the search for a new particle and compare different approaches. We also translate the Neyman-Pearson theory into the language of statistical learning theory.



PhysicsGP: A Genetic Programming Approach to Event Selection

Kyle Cranmer [a] R. Sean Bowman [b]

[a] *CERN, CH-1211 Geveva, Switzerland*
[b] *Open Software Services, LLC, Little Rock, Arkansas, USA*



Comment

Learning a function of experiment-level data $\{x_1, \ldots, x_n\}$ objective is much difficult that learning a function of for an individual event x_i

- Function is more complicated & the optimization itself is more expensive than for an event-level objective

Intuition and confusion:

- Final sensitivity (including systematics) is a function of all the events in the dataset
- Nuisance parameters affect all the events, introduces "correlation"
- Profiling / ability to constrain nuisance parameters is a function of all of the events
- Makes it seem like this event-level optimization is required to be "optimal"

Resolution:

- The data is assumed to be i.i.d., so event-level modeling should be sufficient
- If you can learn the per-event likelihood function $p(x_i | \theta, \nu)$, then it is possible to profile or marginalize the likelihood for the full dataset, which is effectively "optimal".
- So event-level optimization isn't required conceptually. Practical question, which approach is easier.

Summary

Four approaches to incorporating systematics

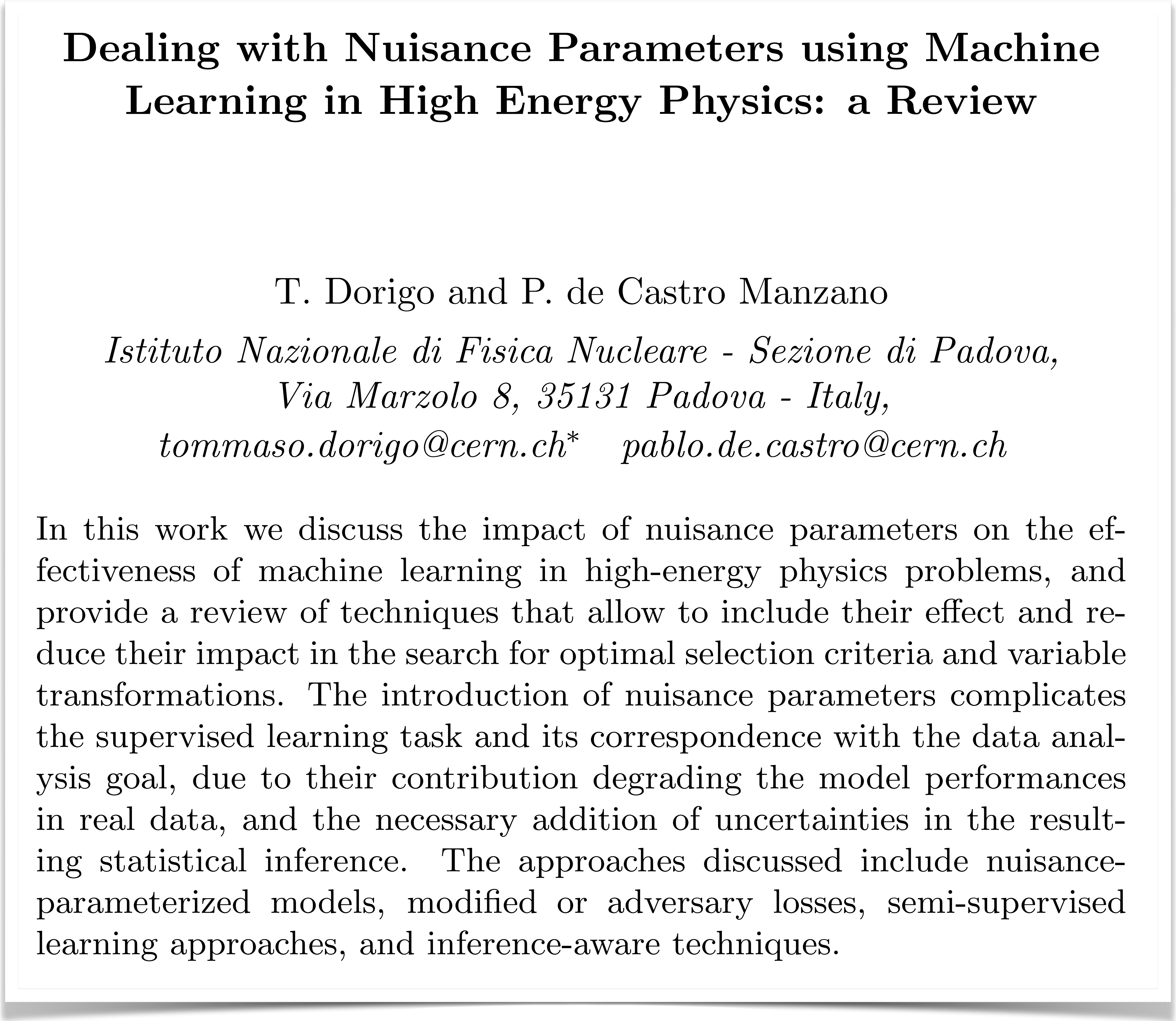
propagation of errors: one works with a model $f(x)$ and simply characterizes how uncertainty in the data distribution propagate through the function to the down-stream task irrespective of how it was trained.

data augmentation: one trains a model $f(x)$ in the usual way using training data from multiple domains by sampling from some distribution over ν.

domain adaptation: one incorporates knowledge of the distribution for domains (or the parameterized family of distributions $p(x \mid y, \nu)$) into the training procedure so that the performance of $f(x)$ for the down-stream task is robust or insensitive to the uncertainty in ν.

parameterized models: instead of learning a single function of the data $f(x)$, one learns a family of functions $f(x; \nu)$ that is explicitly parameterized in terms of nuisance parameters and then accounts for the dependence on the nuisance parameters in the down-stream task.

A review with other approaches



Dealing with Nuisance Parameters using Machine Learning in High Energy Physics: a Review

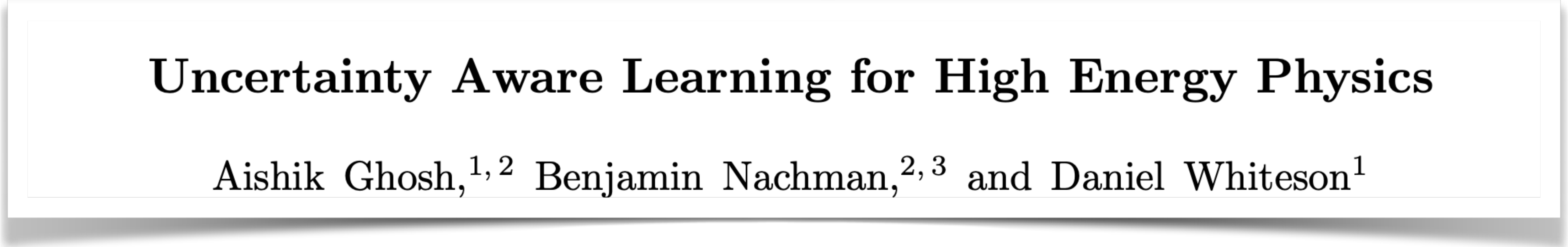
T. Dorigo and P. de Castro Manzano

Istituto Nazionale di Fisica Nucleare - Sezione di Padova,
Via Marzolo 8, 35131 Padova - Italy,
*tommaso.dorigo@cern.ch** *pablo.de.castro@cern.ch*

In this work we discuss the impact of nuisance parameters on the effectiveness of machine learning in high-energy physics problems, and provide a review of techniques that allow to include their effect and reduce their impact in the search for optimal selection criteria and variable transformations. The introduction of nuisance parameters complicates the supervised learning task and its correspondence with the data analysis goal, due to their contribution degrading the model performances in real data, and the necessary addition of uncertainties in the resulting statistical inference. The approaches discussed include nuisance-parameterized models, modified or adversary losses, semi-supervised learning approaches, and inference-aware techniques.

To appear in "Artificial Intelligence for Particle Physics", World Scientific Publishing Co
https://inspirehep.net/literature/1807719

See also this paper that compares the approaches I mentioned and advocates parameterized approach



Uncertainty Aware Learning for High Energy Physics

Aishik Ghosh,[1,2] Benjamin Nachman,[2,3] and Daniel Whiteson[1]

https://arxiv.org/abs/2105.08742

Conclusion

Systematic uncertainties usually have a negative connotation since they reduce the sensitivity of an experiment.

However, the practical and conceptual challenges posed by various types of systematic uncertainty also have a long track record of motivating new ideas.

Thank you / Questions?

Learning to Pivot

x Choice (Summary Stat)	Low-dim summary stat designed by expert	Low-Dim summary stat learned / optimized	Low-level x, no explicit summary stat (learned implicitly)
Model target	Density / Likelihood	Likelihood Ratio	
x-dependence	Low-dim x Histogram, Kernel	NN (or Tree)	
θ-dependence	Fixed Parametrization / Interpolation / Morphing	Agnostic / "non-parametric" (e.g. NN, GP)	
Scope of optimization objective	N/A (constructive)	Per-Event (Classifier)	Experiment-wide (Adversary & Hyper parameter opt.)

Learning to pivot with adversarial networks

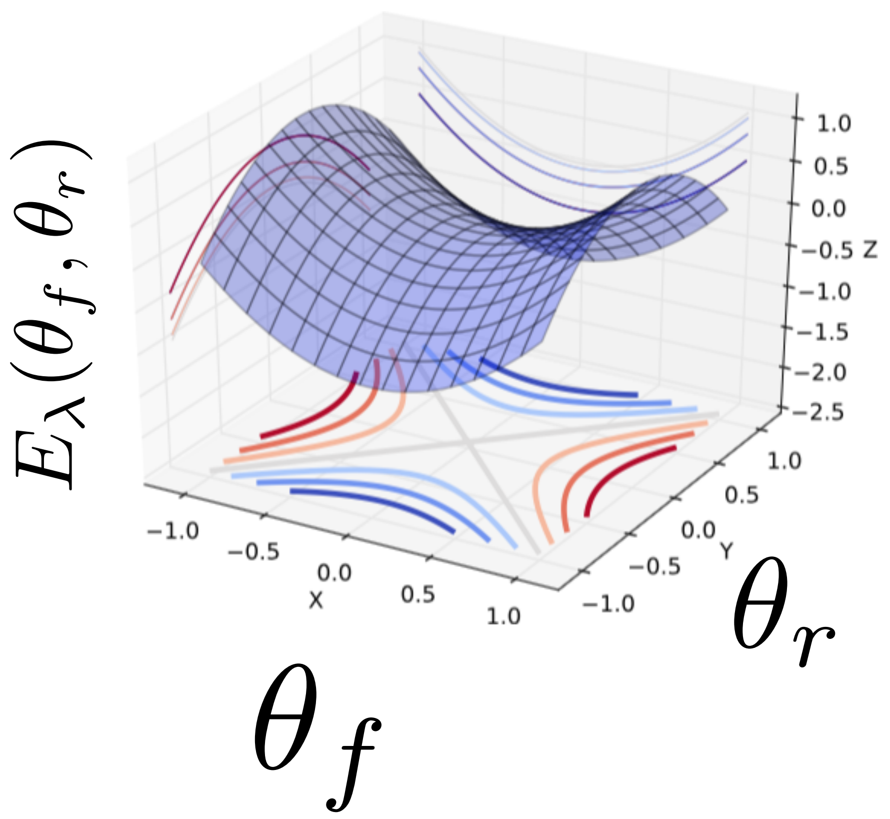
G. Louppe, M. Kagan, K. Cranmer,
arXiv:1611.01046

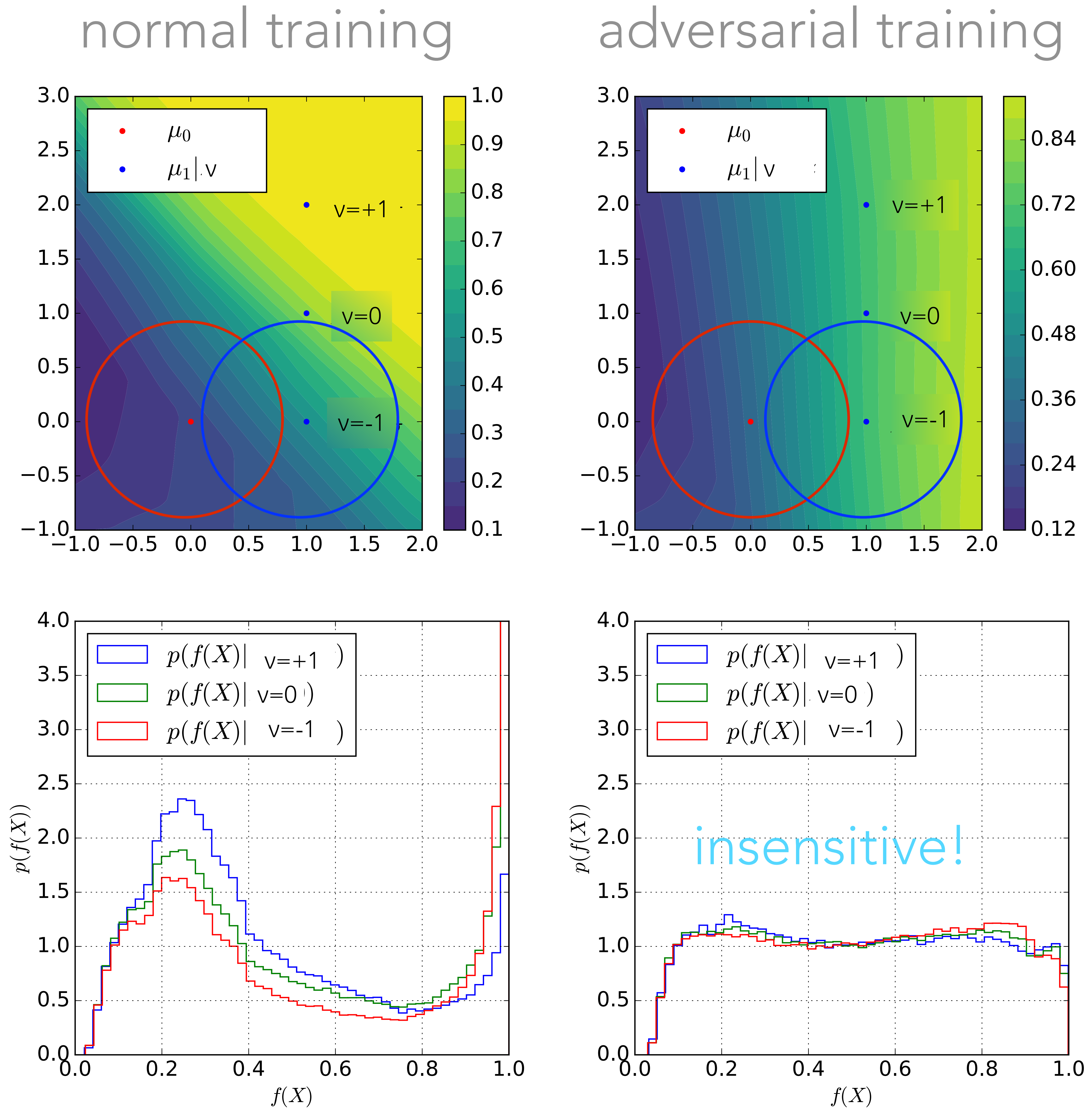
Typically classifier $f(x)$ trained to minimize loss $\mathbf{L_f}$.

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

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

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





An example of learning to pivot

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

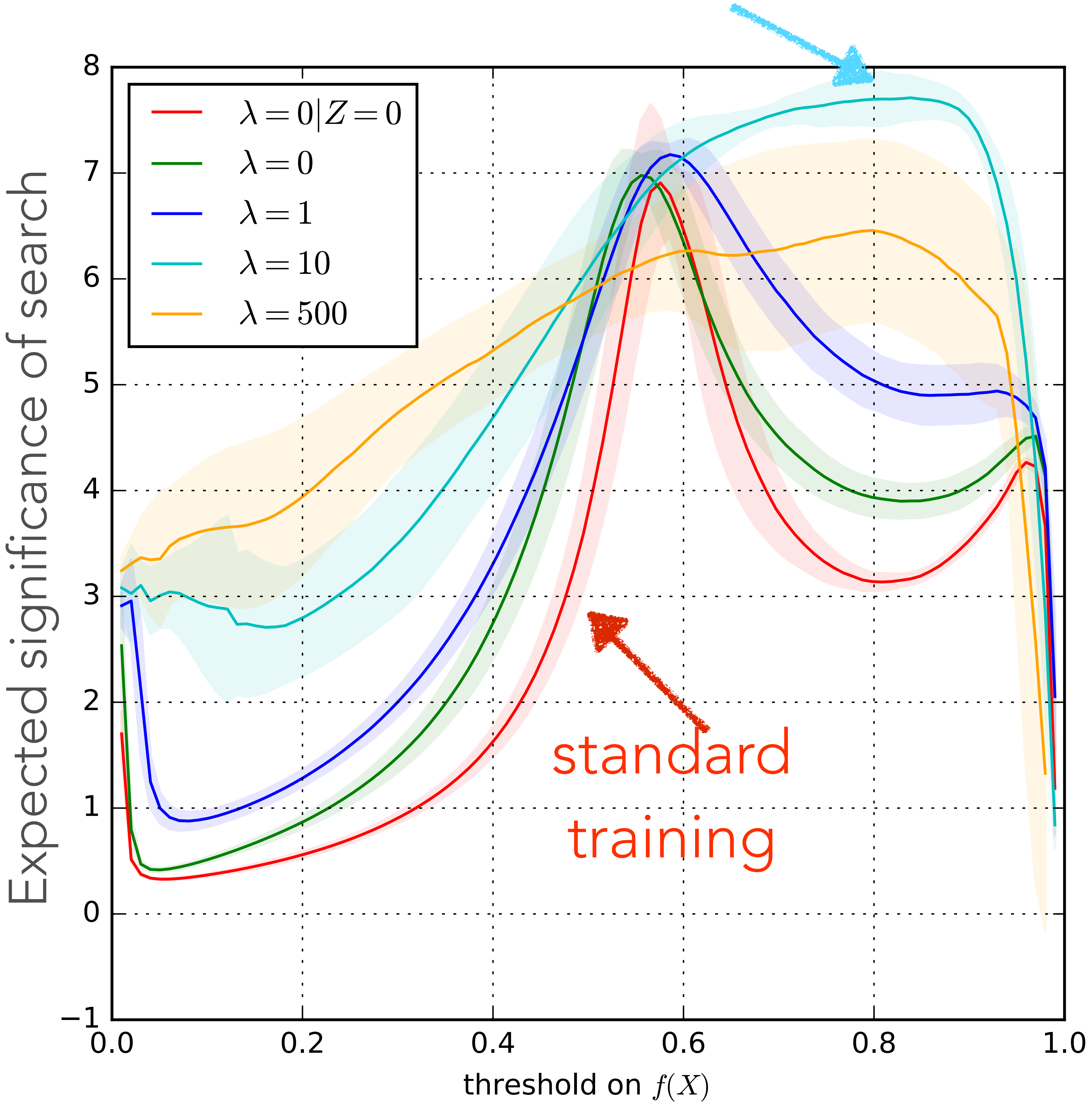
An example:
background: 1000 QCD jets
signal: 100 boosted W's

Train W vs. QCD classifier

Pileup as source of uncertainty

Simple cut-and-count analysis with background uncertainty.

optimal tradeoff of classification vs. & robustness

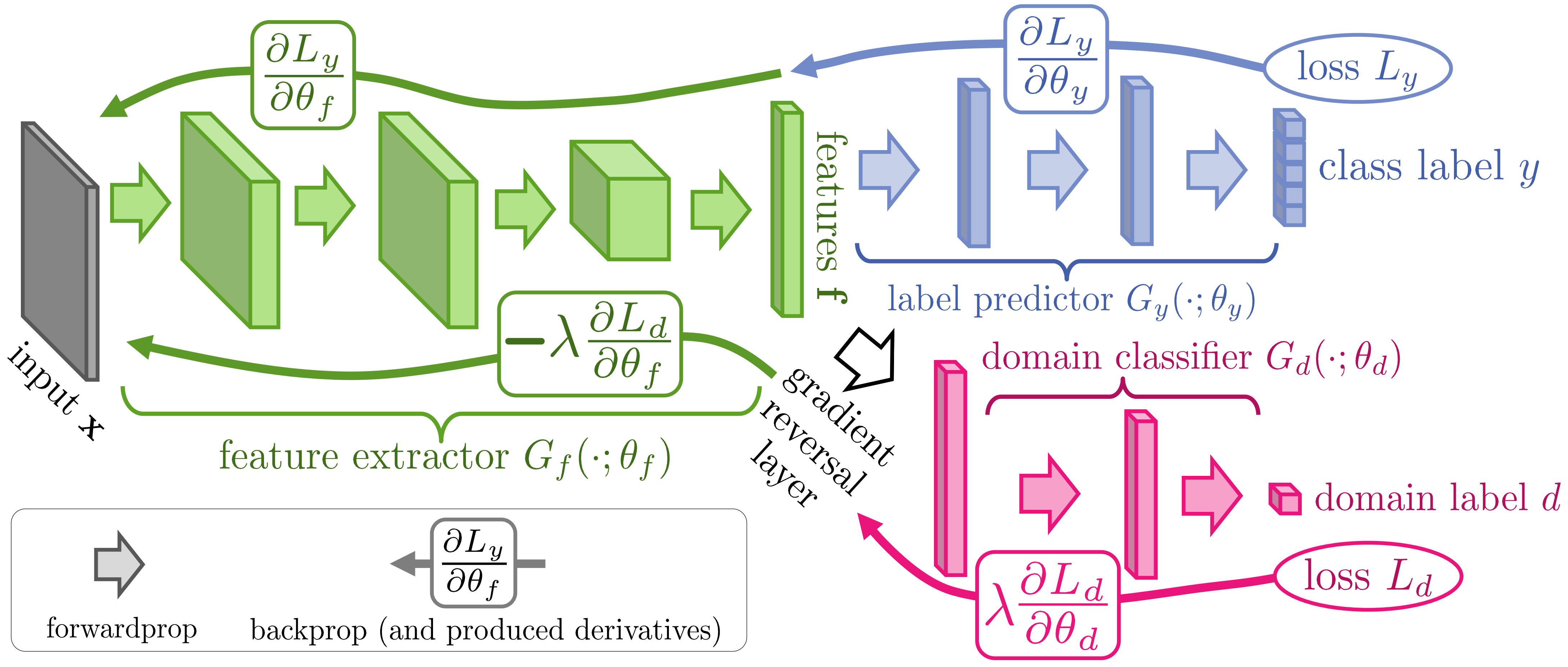


standard training

Domain adaptation

https://arxiv.org/abs/1505.07818

GANIN, USTINOVA, AJAKAN, GERMAIN, LAROCHELLE, LAVIOLETTE, MARCHAND AND LEMPITSKY



$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
input $\mathbf{x}$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
loss L_d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
forwardprop
$\frac{\partial L_y}{\partial \theta_f}$
backprop (and produced derivatives)

In machine learning literature, the setting where training data doesn't match real world data is referred to as "domain shift" and techniques to mitigate the loss in performance are called "domain adaptation"

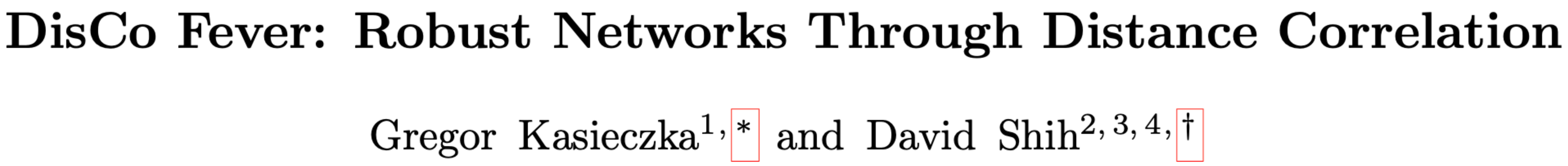
A similar adversarial technique was introduced in arxiv:1505.07818 where adversary tries to get distribution of hidden state features to be invariant. This works for discrete domains, but doesn't generalize well to continuous nuisance parameters.

- adversary works on some low-level features (not just the class prediction)

Learned adversary → explicit regularization

One way of interpreting the mini-max game $\hat{\theta}_f, \hat{\theta}_r = \arg \min_{\theta_f} \max_{\theta_r} E(\theta_f, \theta_r).$ is to minimize a **regularized** loss term $\tilde{L}(\theta_f) = \arg\max_{\theta_r} E_\lambda(\theta_f, \theta_r)$ where the optimization with respect to θ_r is not exposed

This motivates another approach in which the regularization is not achieved through a learned adversary, but some other measure of discrepancy



DisCo Fever: Robust Networks Through Distance Correlation

Gregor Kasieczka[1,*] and David Shih[2,3,4,†]

$$L = L_{classifier}(\vec{y}, \vec{y}_{true}) + \lambda\, \mathrm{dCorr}^2_{y_{true}=0}(\vec{m}, \vec{y})$$

$$\begin{aligned}\mathrm{dCov}^2(X,Y) = &\langle |X - X'||Y - Y'|\rangle \\ &+ \langle |X - X'|\rangle\langle |Y - Y'|\rangle \\ &- 2\langle |X - X'||Y - Y''|\rangle\end{aligned}$$

Phys.Rev.Lett. 125 (2020) 12, 122001

Ancient History

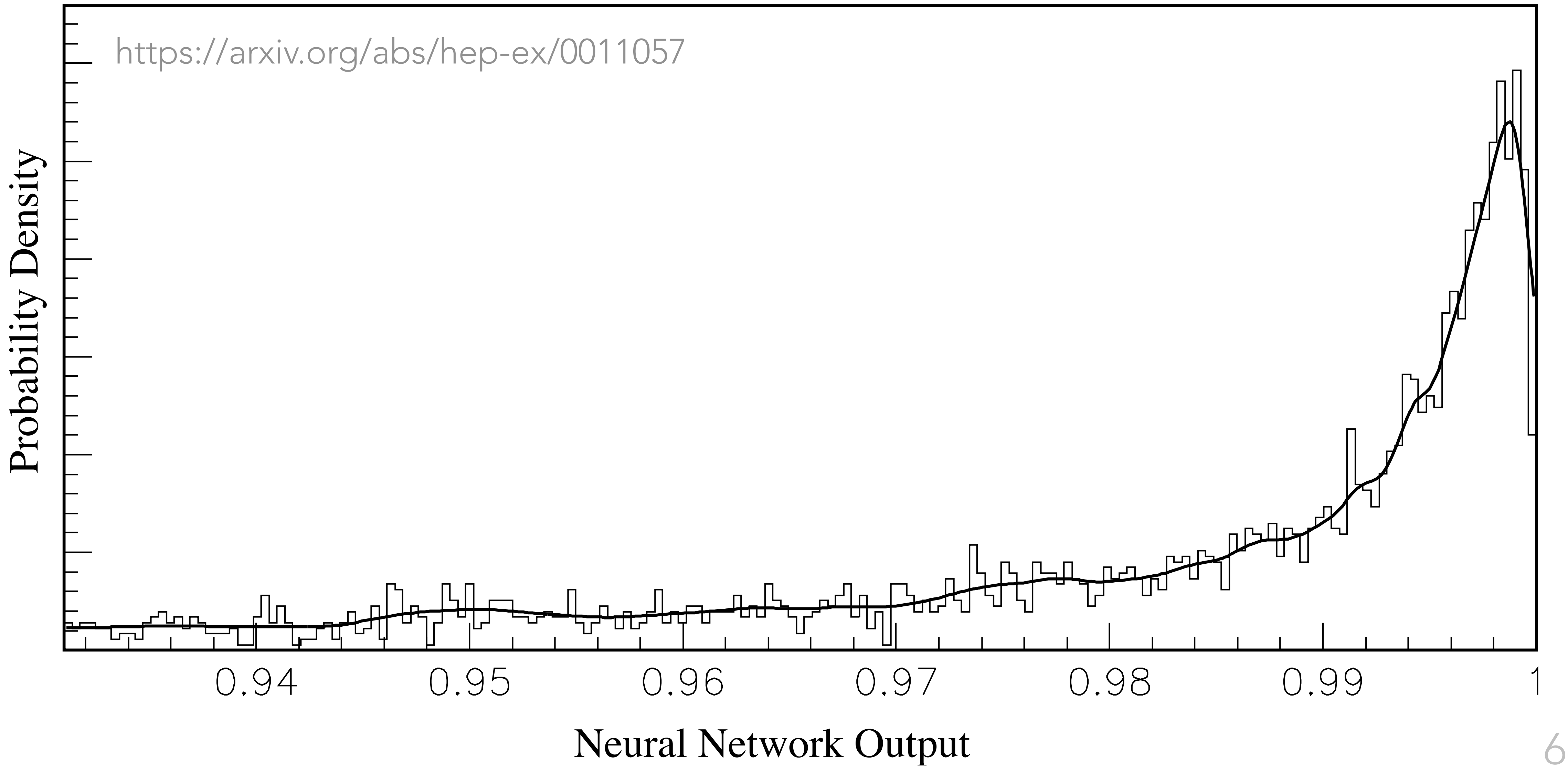
MC Stat Uncertainties ⇒ Kernel Density Estimation

Back in ~1999, the four experiments LEP experiments were performing the first likelihood-based combinations

- Input to the combinations were histograms, **but** limited Monte Carlo sample size led to unphysical fluctuations
- Now we explicitly treat these as **MC stat uncertainty** with nuisance parameters, but at the time the desire was to **smooth** the distributions

My first paper was to introduce kernel density estimation for this (KEYS)

- An example of density estimation
- Non-parametric. ML-adjacent
- Regularization is important



https://arxiv.org/abs/hep-ex/0011057

Nice work, but be careful about specializing in statistics and software.

Fred James

Author of MINUIT / MINOS

Editor of Computer Physics Communications for many years

The first PhyStat

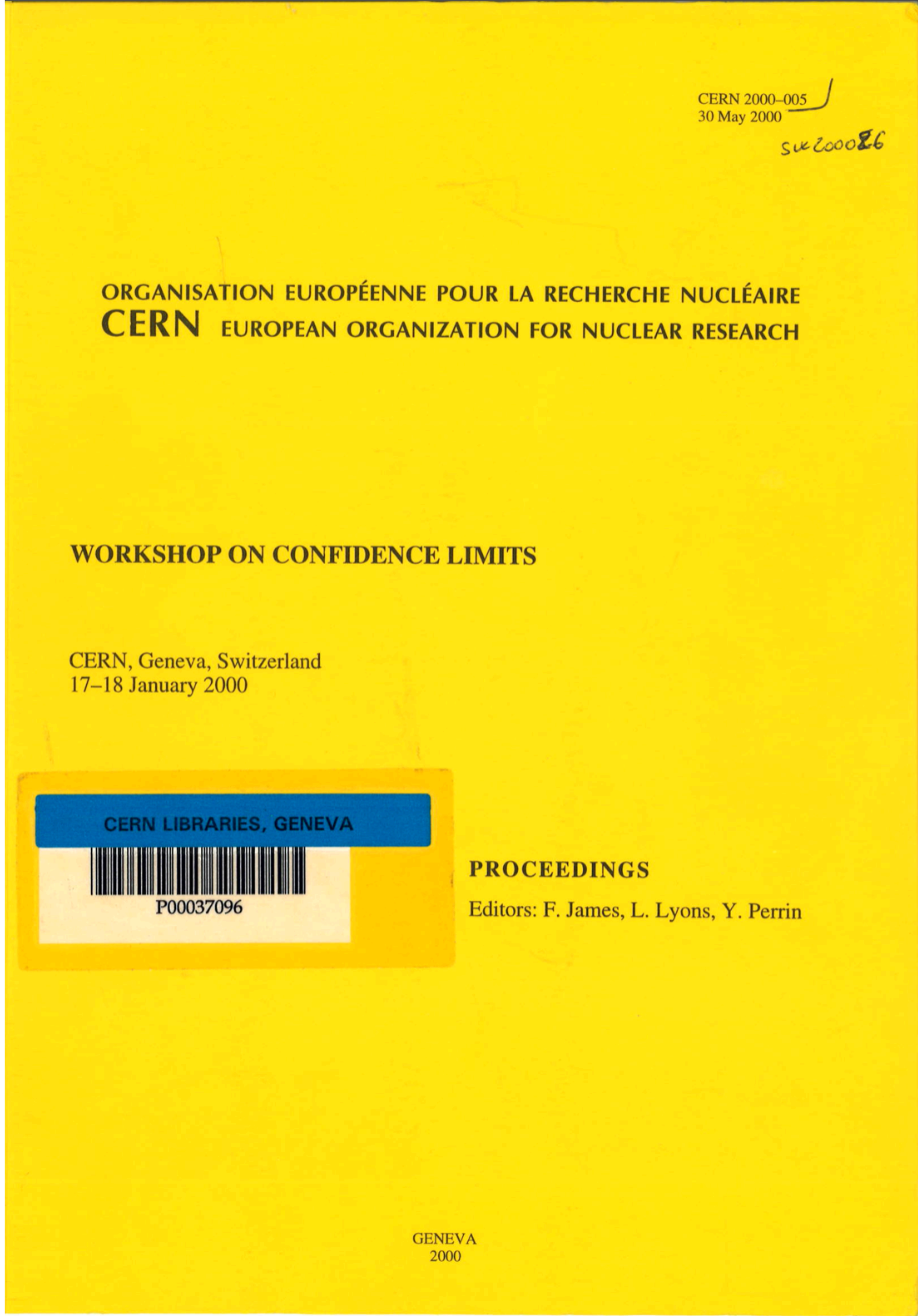
It was 24 years ago!

- I was just starting as a graduate student

Louis suggested I think about frequentist statistical procedures with systematics

Louis Lyons of Oxford, co-convenor of the workshop on confidence limits.

https://cds.cern.ch/record/411537?ln=en



CERN 2000–005
30 May 2000

SUC200026

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

WORKSHOP ON CONFIDENCE LIMITS

CERN, Geneva, Switzerland
17–18 January 2000

PROCEEDINGS

Editors: F. James, L. Lyons, Y. Perrin

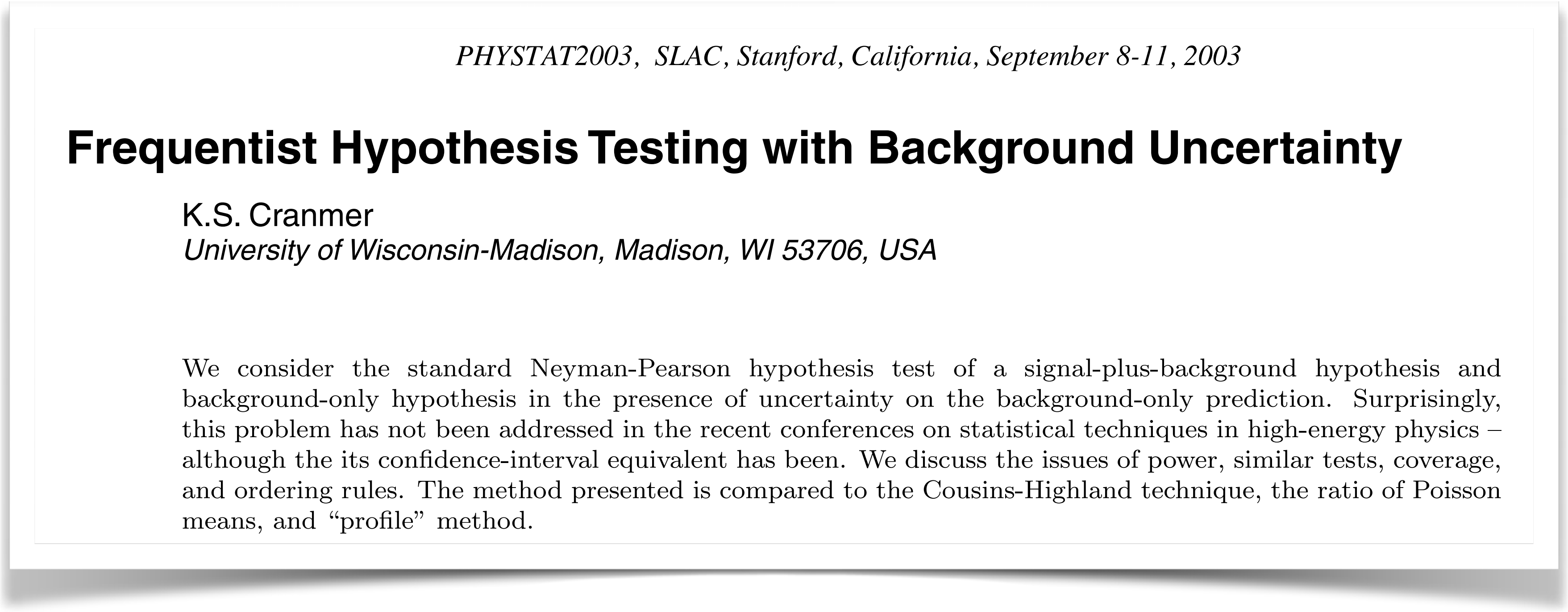
GENEVA
2000

https://arxiv.org/abs/physics/0310108

Neyman construction with systematics & profiling

At PhyStat 2003, I presented my first work on frequents hypothesis testing & Neyman Construction with nuisance parameters

- Mainly translating Kendall and Stuart
- Early days of HEP understanding the profile likelihood ratio

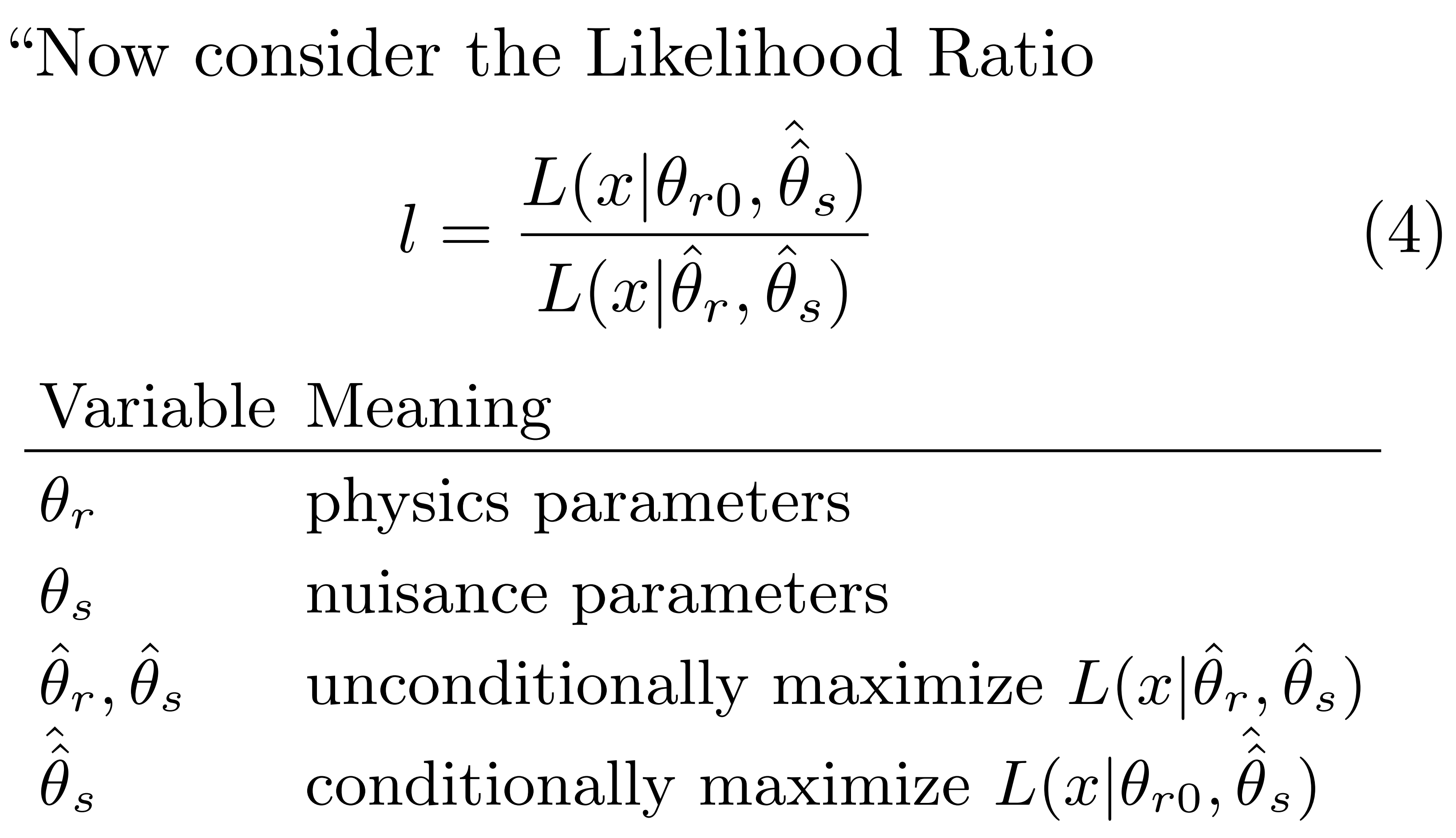


PHYSTAT2003, SLAC, Stanford, California, September 8-11, 2003

Frequentist Hypothesis Testing with Background Uncertainty

K.S. Cranmer
University of Wisconsin-Madison, Madison, WI 53706, USA

We consider the standard Neyman-Pearson hypothesis test of a signal-plus-background hypothesis and background-only hypothesis in the presence of uncertainty on the background-only prediction. Surprisingly, this problem has not been addressed in the recent conferences on statistical techniques in high-energy physics – although the its confidence-interval equivalent has been. We discuss the issues of power, similar tests, coverage, and ordering rules. The method presented is compared to the Cousins-Highland technique, the ratio of Poisson means, and "profile" method.



"Now consider the Likelihood Ratio

$$l = \frac{L(x|\theta_{r0}, \hat{\hat{\theta}}_s)}{L(x|\hat{\theta}_r, \hat{\theta}_s)} \tag{4}$$

Variable	Meaning
θ_r	physics parameters
θ_s	nuisance parameters
$\hat{\theta}_r, \hat{\theta}_s$	unconditionally maximize $L(x\|\hat{\theta}_r, \hat{\theta}_s)$
$\hat{\hat{\theta}}_s$	conditionally maximize $L(x\|\theta_{r0}, \hat{\hat{\theta}}_s)$

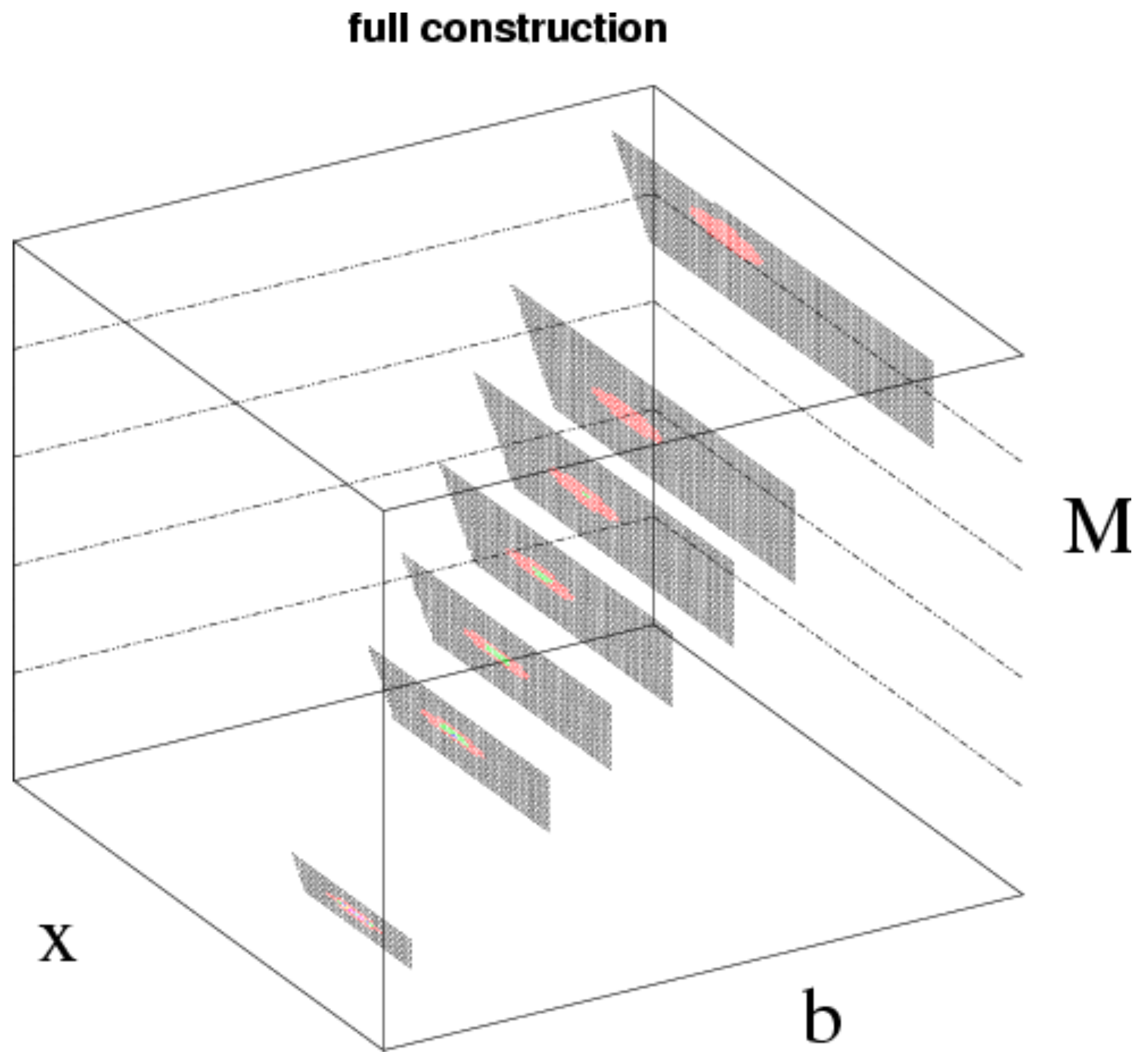
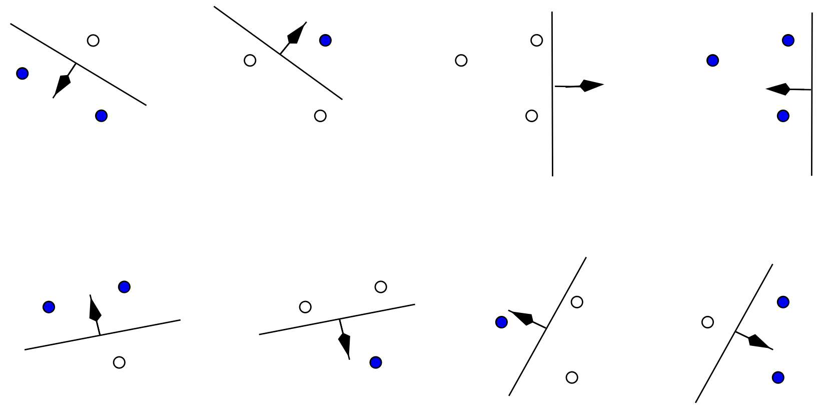


Figure 1: The Neyman construction for a test statistic x, an auxiliary measurement M, and a nuisance parameter b. Vertical planes represent acceptance regions W_b for H_0 given b. The condition for discovery corresponds to data (x_0, M_0) that do not intersect any acceptance region. The contours of $L(x, M|H_0, b)$ are in color.

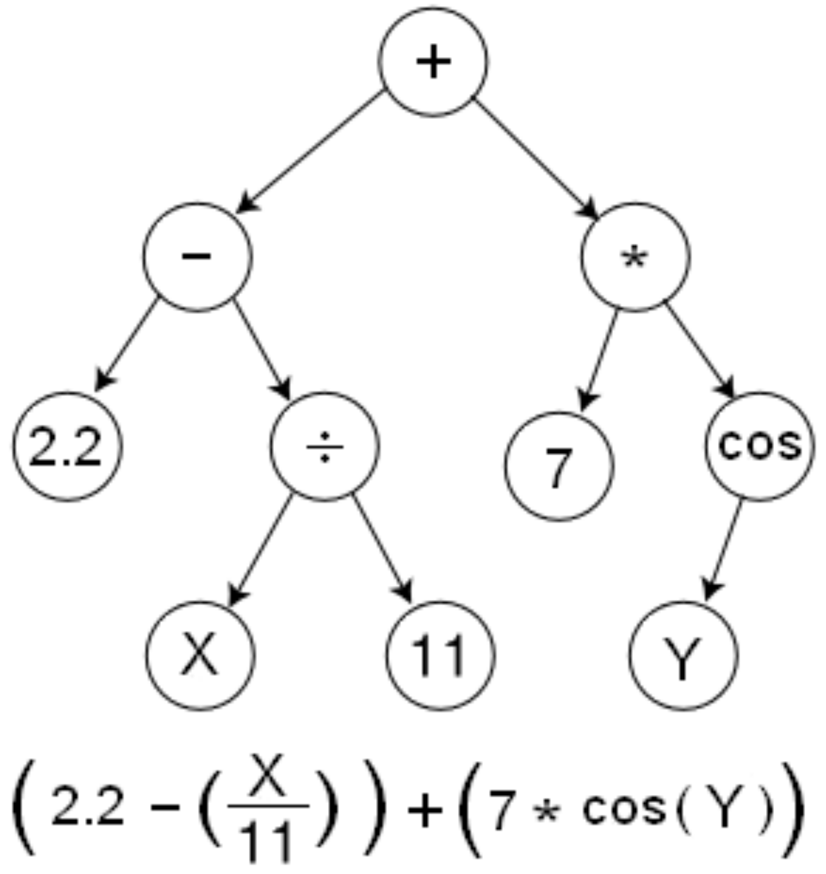
Historical Context

In late 1990s and early 2000s, HEP was using neural networks (mainly shallow MLPs).

- Decision trees were growing in popularity (a topic at PhyStat 2003)
- Support Vector Machines and Vapnik's Statistical Learning Theory were becoming very popular
  - Provided formal guarantees, unique solutions, etc.
  - VC Dimension captured intuitive notion that very flexible models can overfit



- John Koza introduced Genetic Programming
  - Discrete optimization / search over expressions



Comparing these for HEP was one of my first ML projects

- … and we still called it "multivairate analysis" 😆

PhyStat 2003 & PhysicsGP

https://arxiv.org/abs/physics/0310110
https://arxiv.org/abs/physics/0402030

Discussed high-level strategies:

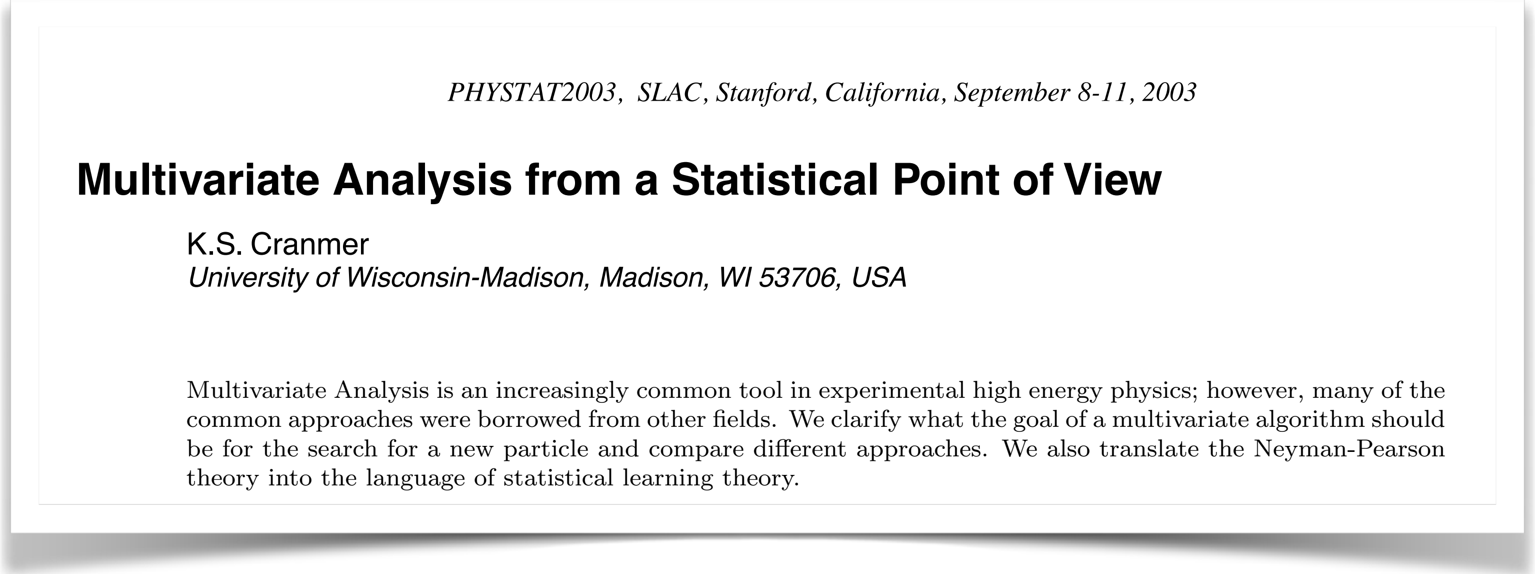
- **Indirect:** Optimize an objective that yields a well-motivated function (e.g. approximate likelihood ratio)
  - Argue that the resulting classifier should be close to optimal
- **Direct:** optimize expected discovery significance
  - **Objective can include systematic uncertainty!**

Genetic Programming / Symbolic Regression:

- Search through space of cuts / summary statistics expressed symbolically using genetic programming
- Interpretable. Less prone to overfitting (low VC dimension)

Hypothesis Testing & Statistical Learning Theory:

- Expressed Neyman-Pearson in terms of Risk
- Discussion of VC dimension for NNs, SVMs, symbolic cuts

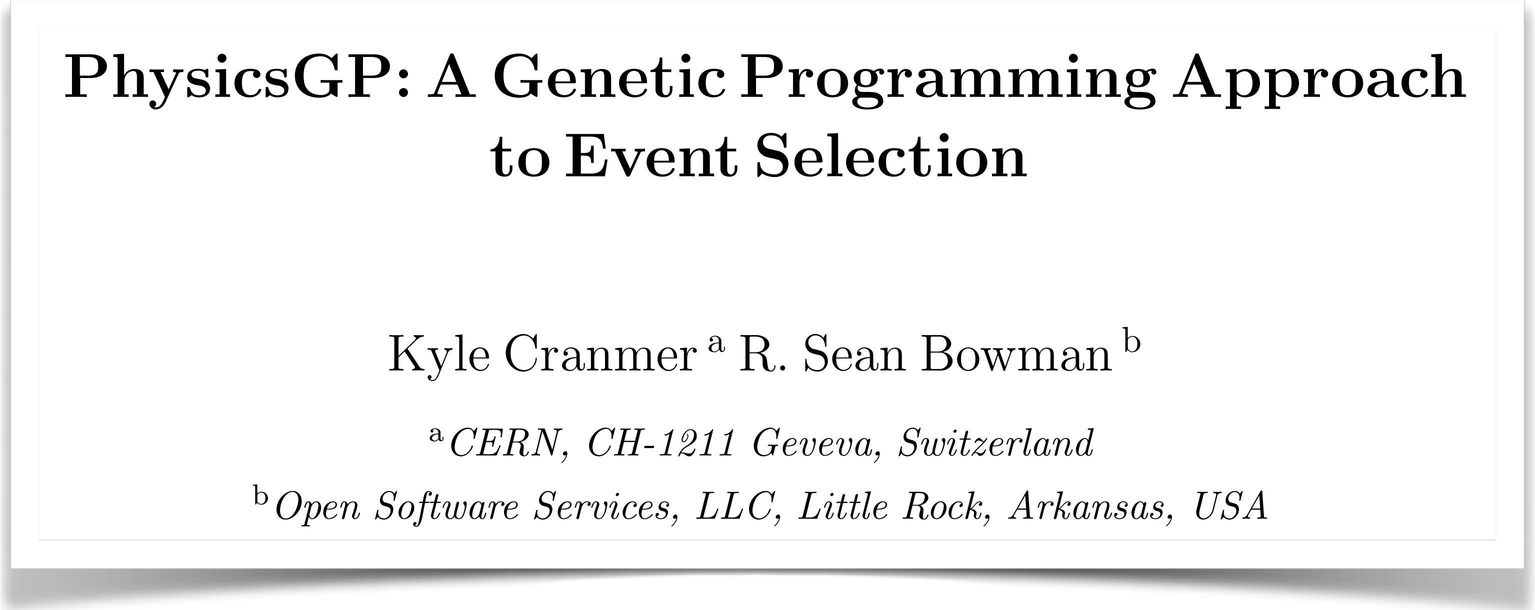


PHYSTAT2003, SLAC, Stanford, California, September 8-11, 2003

Multivariate Analysis from a Statistical Point of View

K.S. Cranmer
University of Wisconsin-Madison, Madison, WI 53706, USA

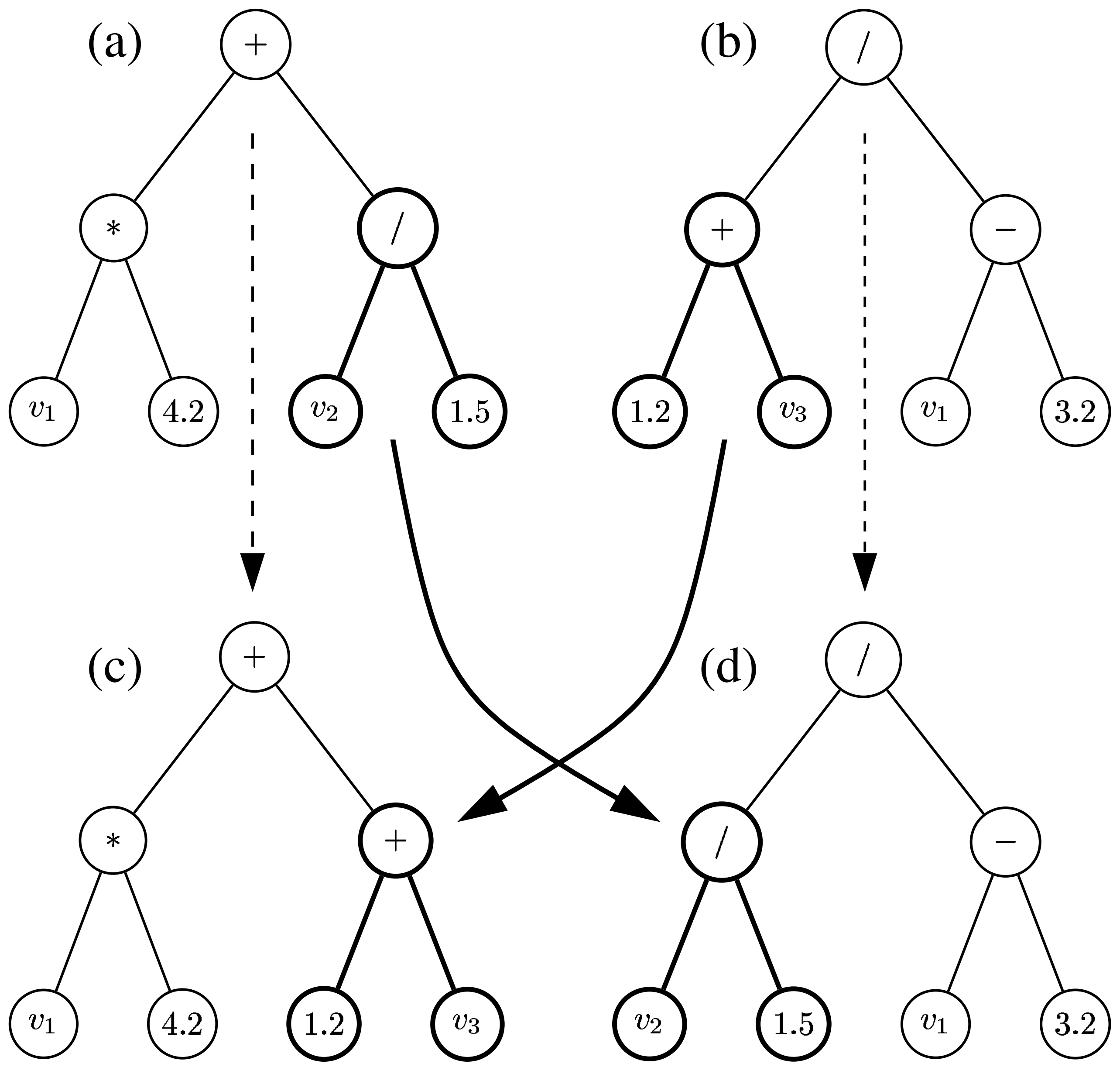
Multivariate Analysis is an increasingly common tool in experimental high energy physics; however, many of the common approaches were borrowed from other fields. We clarify what the goal of a multivariate algorithm should be for the search for a new particle and compare different approaches. We also translate the Neyman-Pearson theory into the language of statistical learning theory.



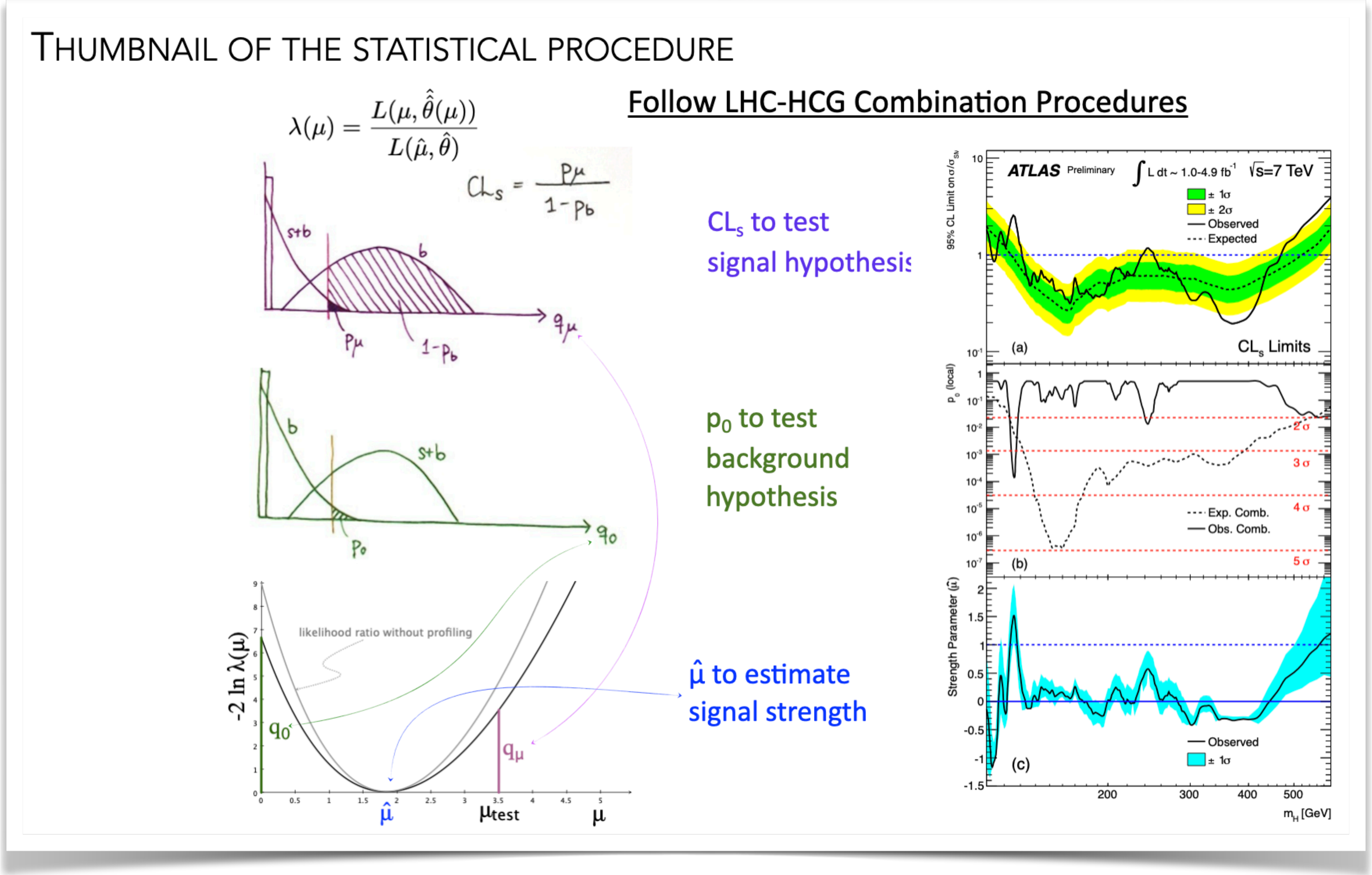
PhysicsGP: A Genetic Programming Approach to Event Selection

Kyle Cranmer [a] R. Sean Bowman [b]

[a] *CERN, CH-1211 Geveva, Switzerland*
[b] *Open Software Services, LLC, Little Rock, Arkansas, USA*



For PhyStat 2005 and 2007, my focus was mainly on statistical procedures and software for the LHC (RooFit, RooStats, profile likelihood ratio, asymptotics, HistFactory, workspaces, etc.)



THUMBNAIL OF THE STATISTICAL PROCEDURE

Follow LHC-HCG Combination Procedures

$$\lambda(\mu) = \frac{L(\mu, \hat{\hat{\theta}}(\mu))}{L(\hat{\mu}, \hat{\theta})}$$

$$CL_s = \frac{p_\mu}{1 - p_b}$$

CL_s to test signal hypothesis

p_0 to test background hypothesis

$\hat{\mu}$ to estimate signal strength