Levelset Estimation by Bayesian Optimization

K Cranmer, **L Heinrich**, G Louppe
4th Reinterpretation Workshop

May 15th 2018

NEW YORK UNIVERSITY

/lukasheinrich

Searching for BSM:

fundamentally about finding the boundary between models that are consistent with the data and those that are not.

1D: intervals
2D: contours
ND: (hyper-)surfaces

Boundary is usually defined by iso-surfaces of a test statistic (e.g. CLs) at certain values.

Problem:

assessing models is computationally expensive.

This Talk:

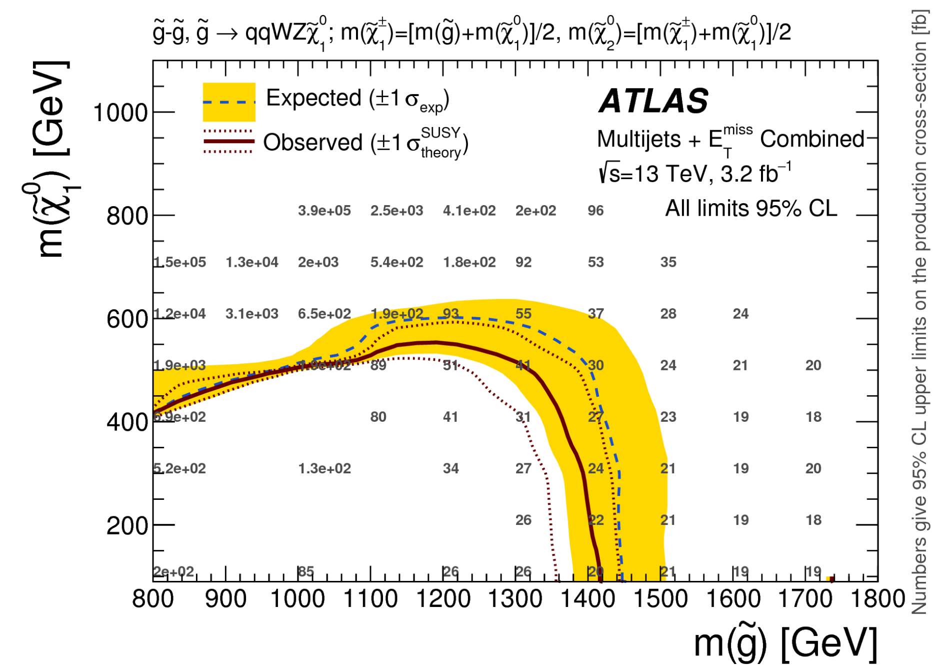
how to find excursion sets / iso-surfaces of generic R^n functions in an efficient way.

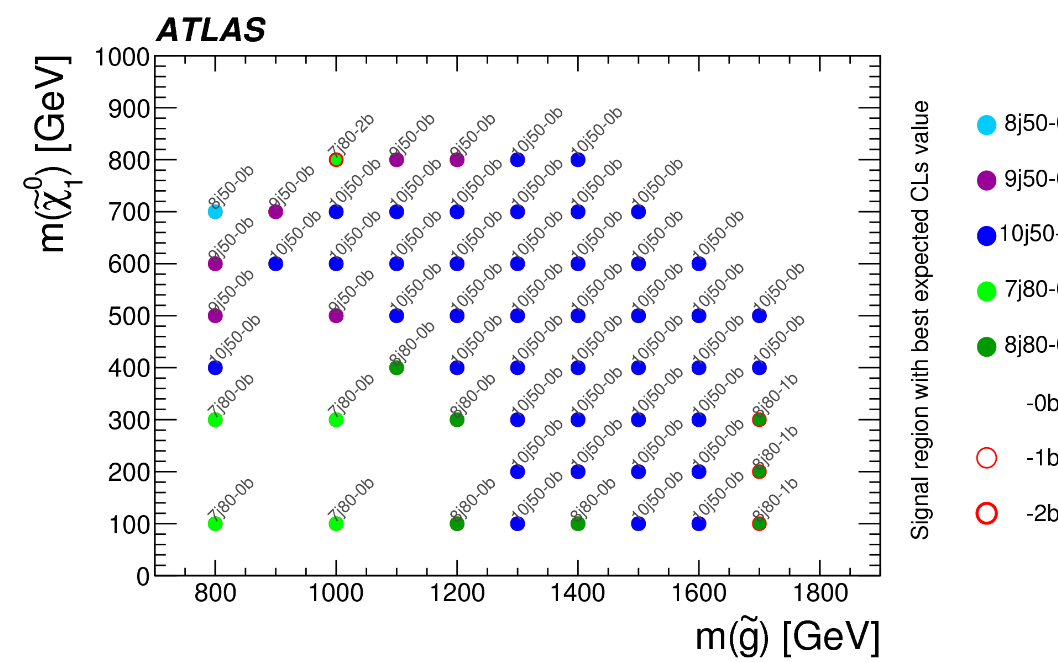
Current Approach (mostly):

- Choose ~regular grid of points ahead of time
- Simulate sufficient number of events per point
- Run sample through analysis + stats
- Interpolate between point results to estimate contour

But:

- many points are not even close to boundary
- need dense grid to resolve contour details
- regular grids to not scale for higher dimension
- details of grid (corner points, spacing, orientation wrt contour all arbitrary but affect the contour)



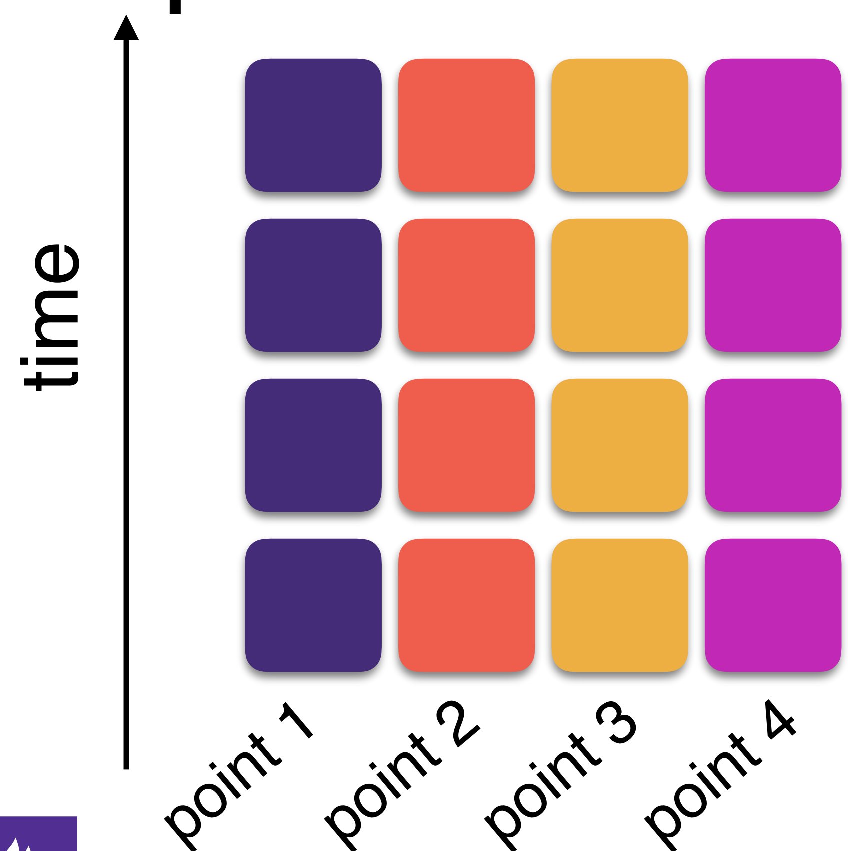


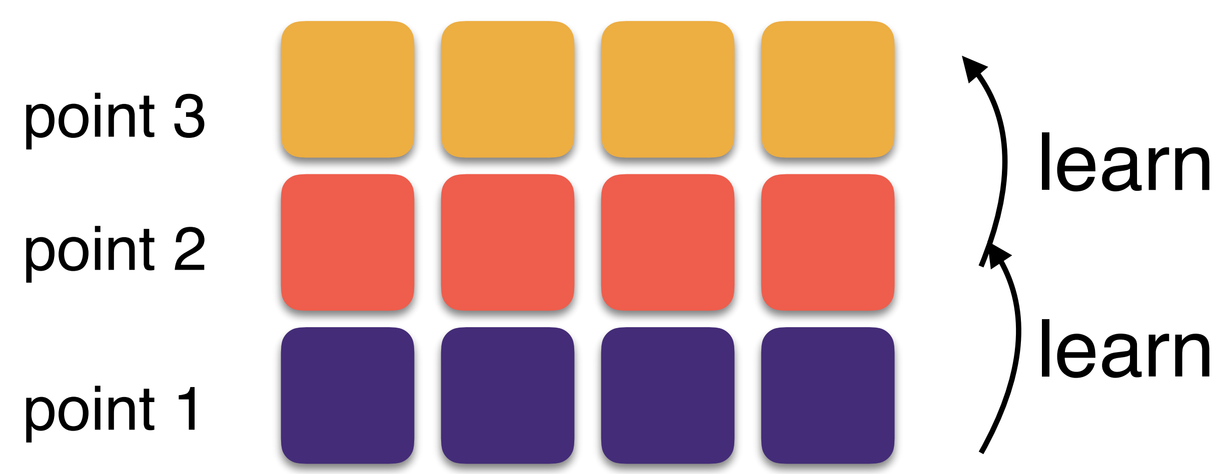
ATLAS-SUSY-2015-07

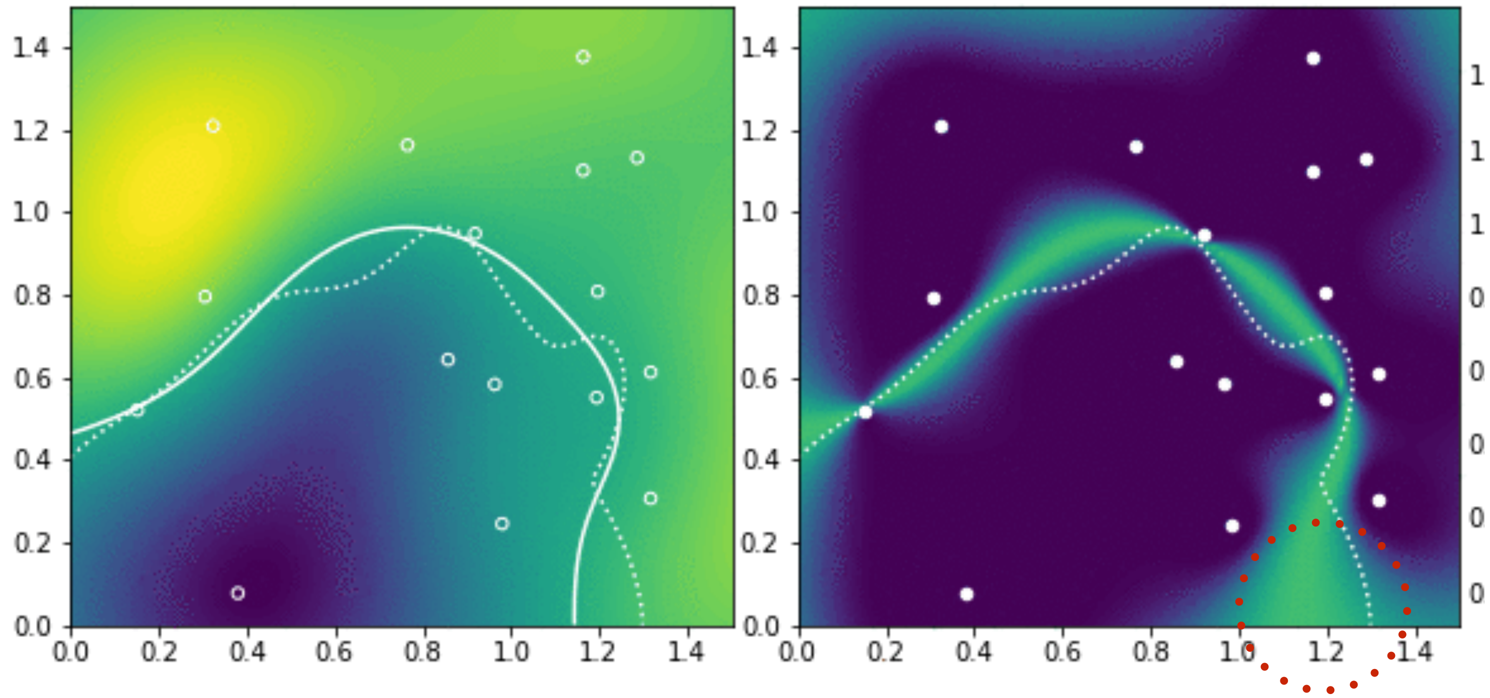
Instead of choosing all points at once, irrespective of what the eventual contour looks like …

… can we construct an smart algorithm that helps us find the points that actually make sense to generate, by iteratively working in what we learn from already generated points.
…perhaps we can do with much fewer points / only generate points close to contour

HEP largely easily parallellizable — reorder the loops and save computations

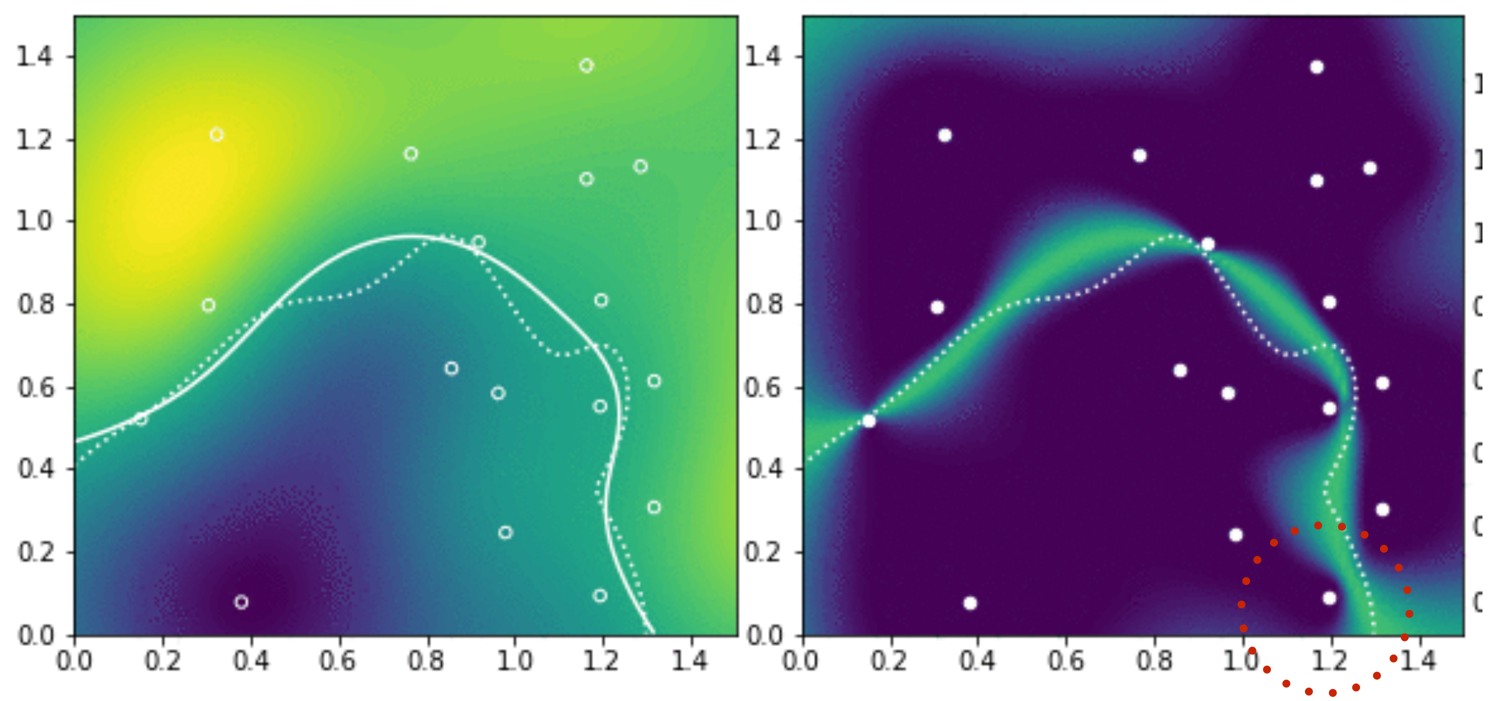






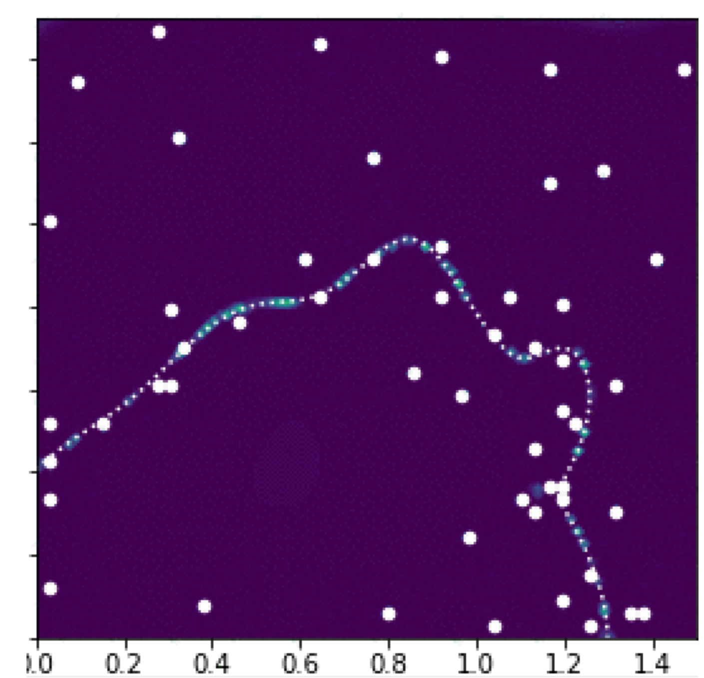
1) observe contour
2) decide next point
3) improve contour

lots of uncertainty
in contour here

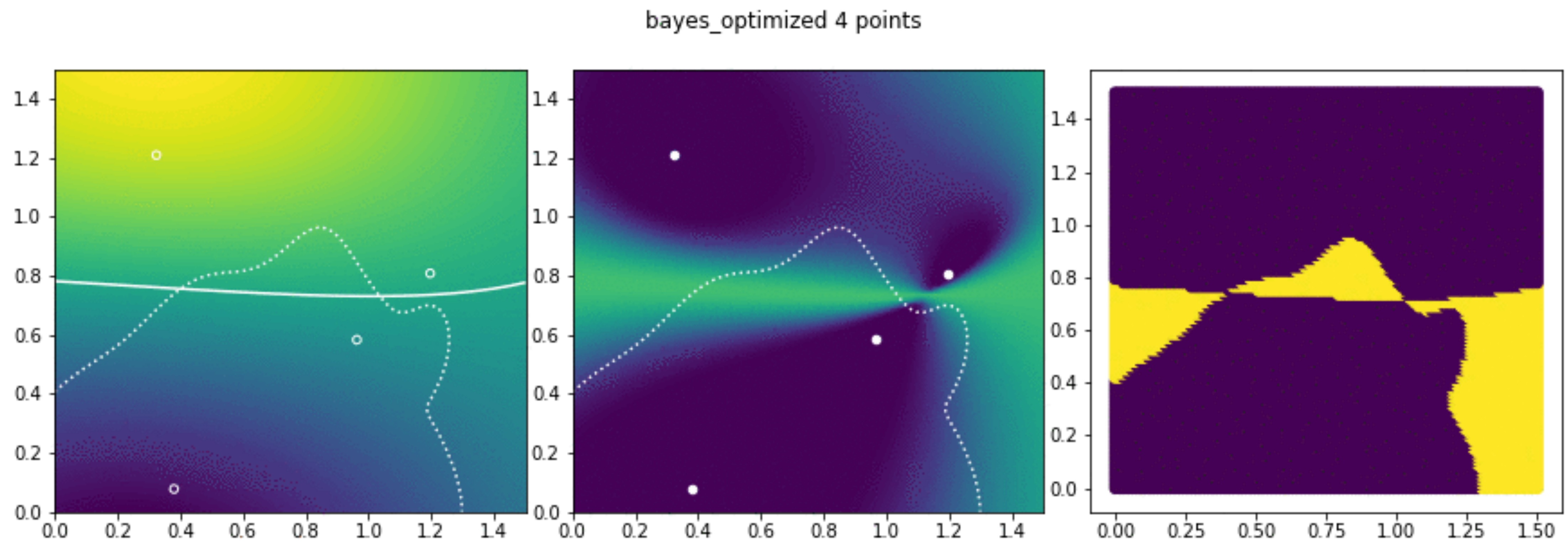


high-value point
close to contour

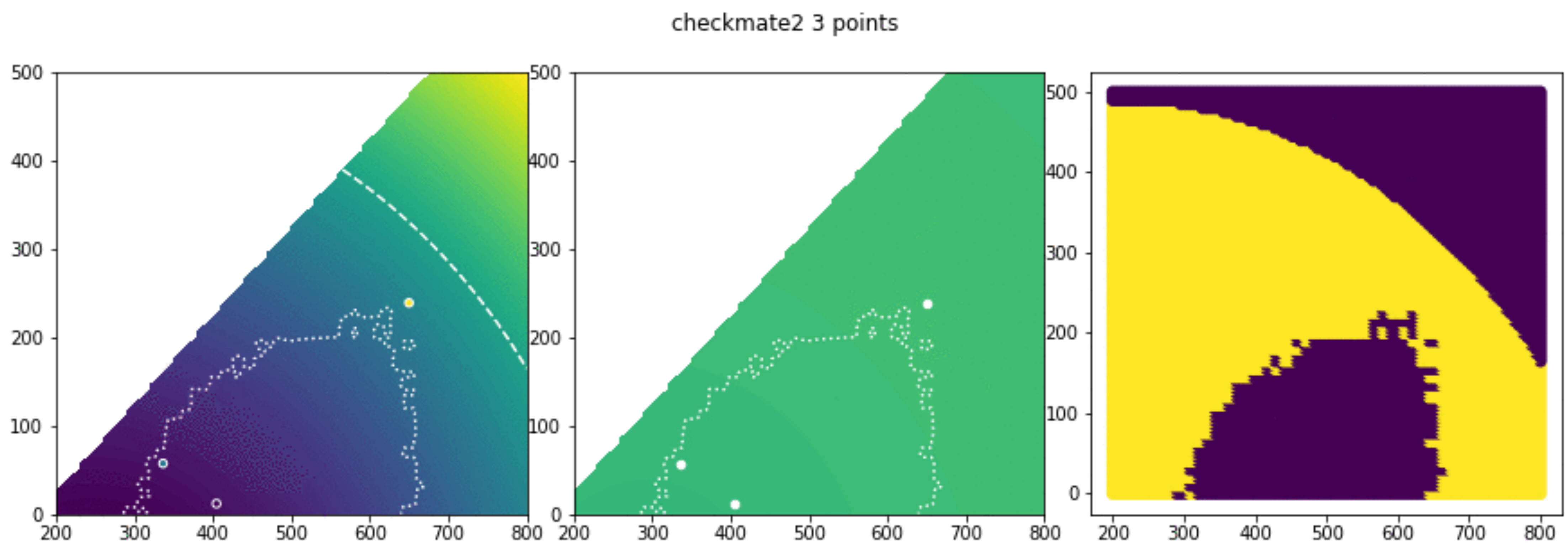
result: points
where they matter



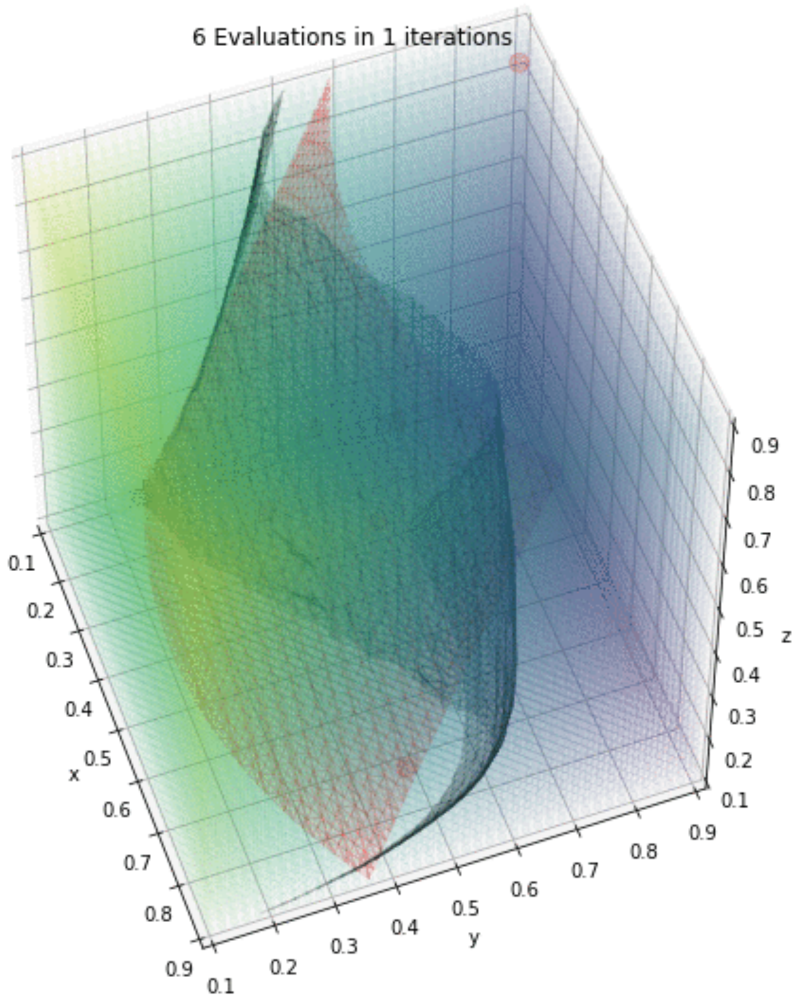
http://lheinric.web.cern.ch/lheinric/contour/gifs/demo_gif.gif

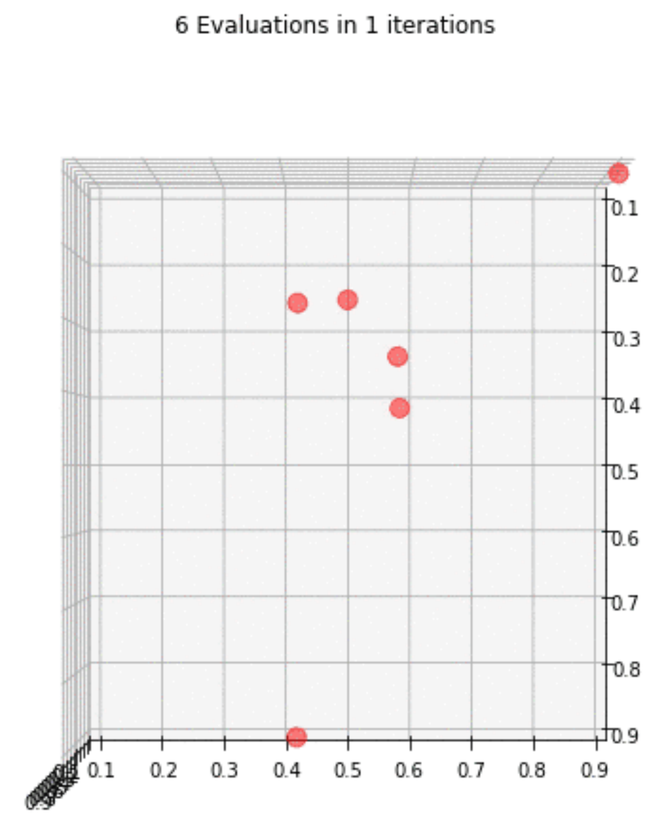


http://lheinric.web.cern.ch/lheinric/contour/gifs/methods_checkmate.gif

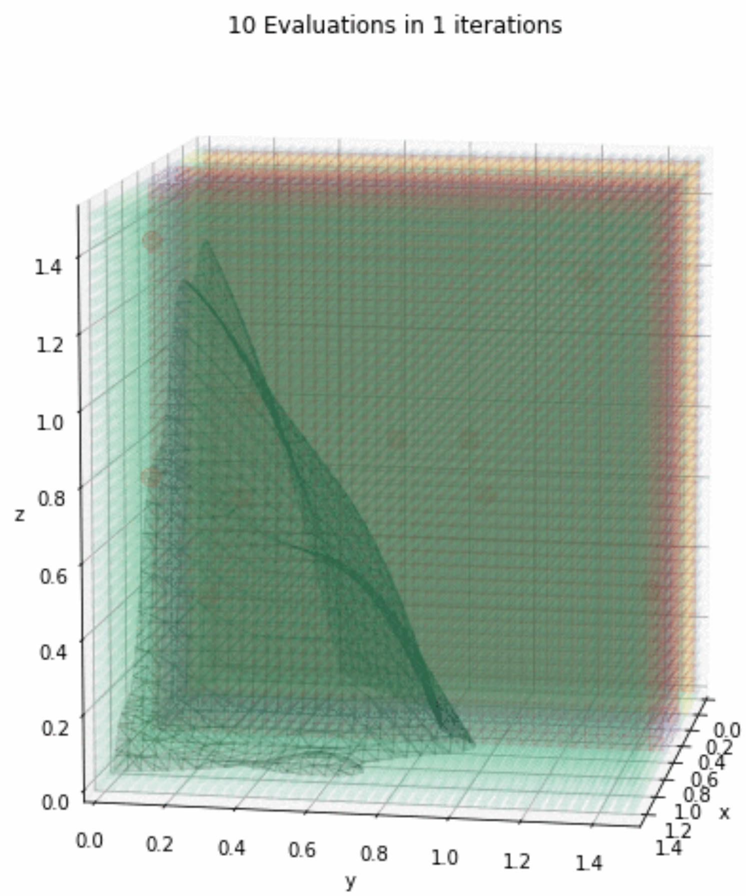


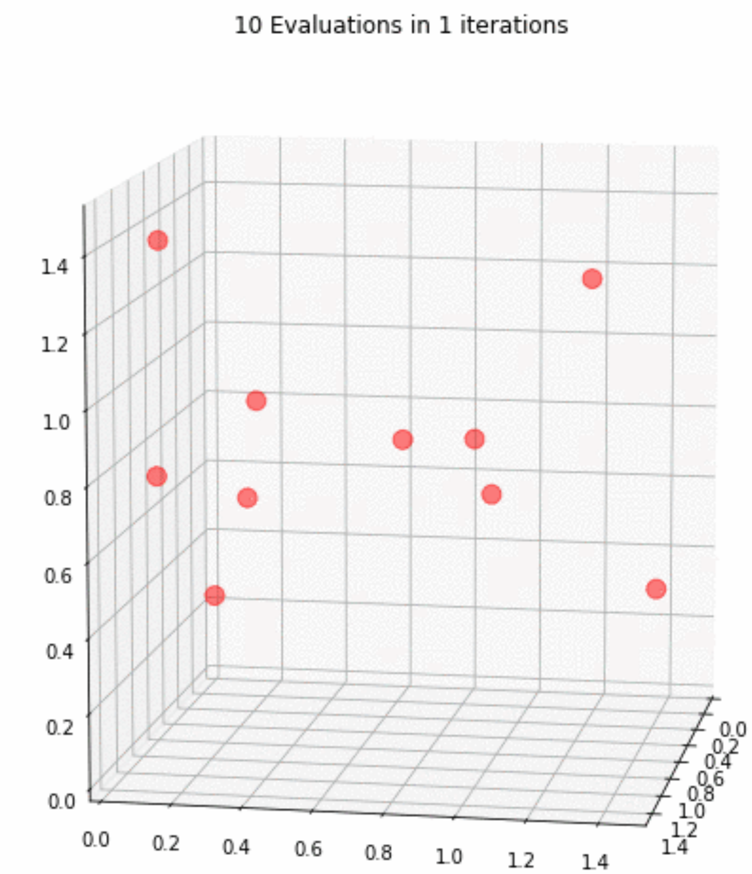
http://lheinric.web.cern.ch/lheinric/contour/animation3ddarkhiggs.gif





http://lheinric.web.cern.ch/lheinric/contour/animation3dtoy.gif





Gaussian Processes:

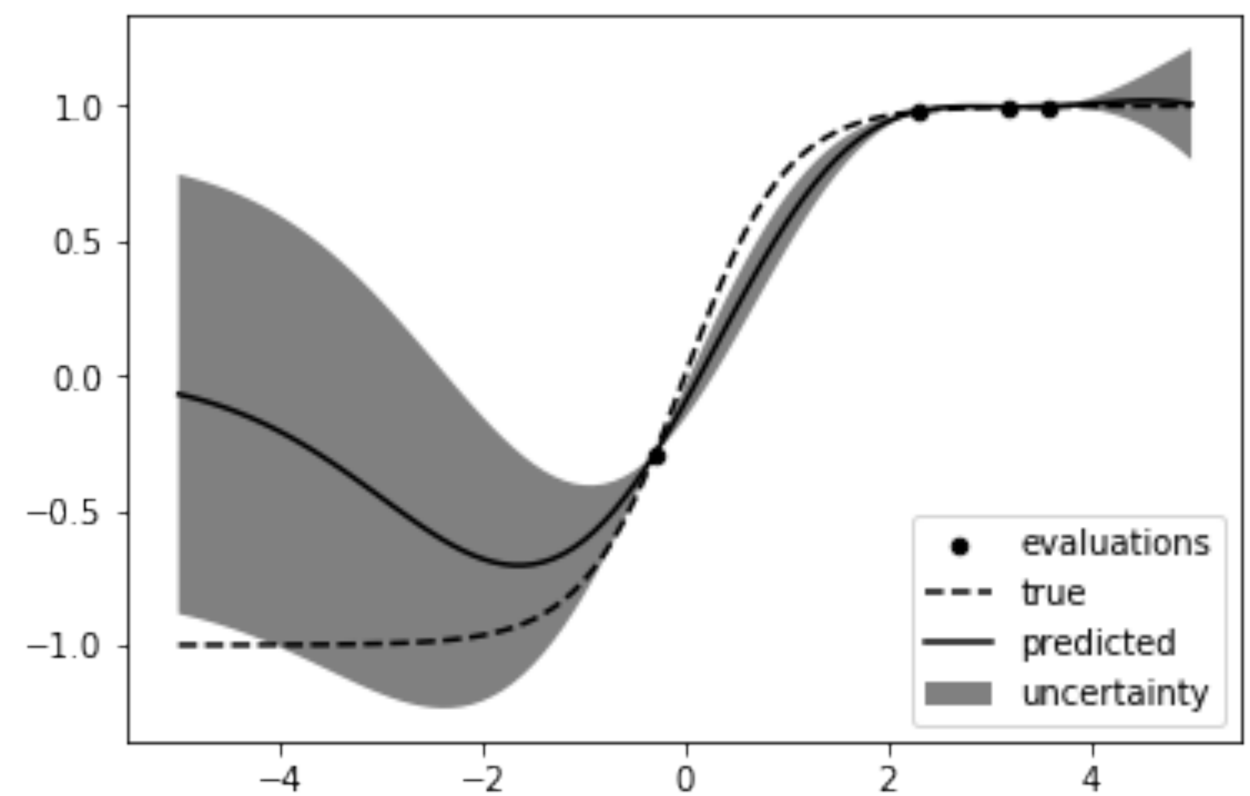
A generalization of multivariate normal distribution to *stochastic fields,* such that for any vector of points,

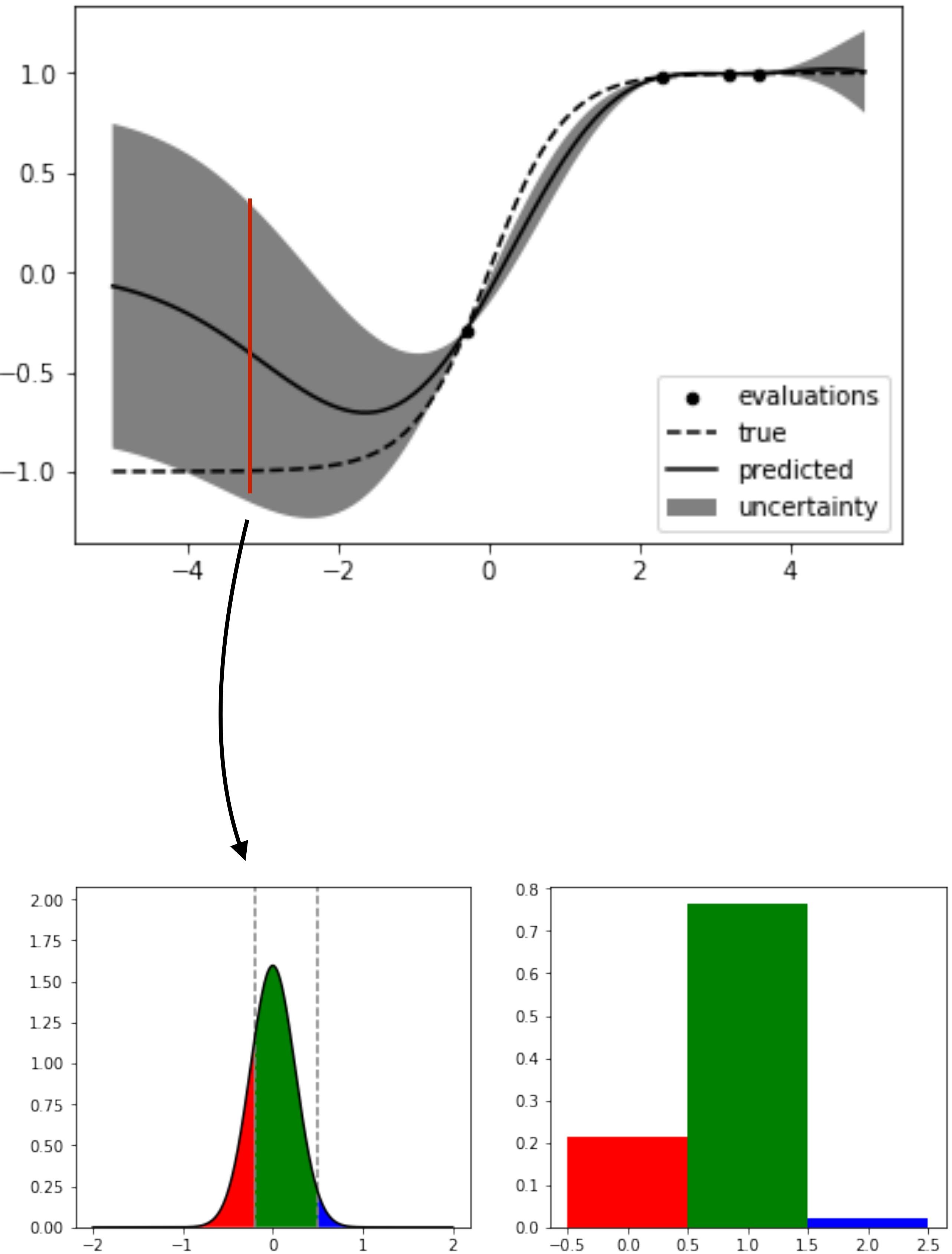
$$Y(\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$$

GP is specified by a Kernel function and its hyperparameters.

Given a limited set of data points (i.e. function evaluations) the hyper-parameters can be fitted and the GP be used to predict function values across the entire domain.

Prediction includes mean value but also uncertainty.





for each point, the value of the function modeled by the GP is described by a normal distribution.

Given a set if thresholds, a discrete pdf S(x) describes probability of x being a member of a given excursion set

Entropy of S(x) gives measure of uncertainty of classification.

<S> gives global assessment of current contour uncertainty.

$$H[S] = S_i(x) \log S_i(x)$$

$$\langle H[S] \rangle = \int \mathrm{d}x' \; H[S(x')]$$

Bayesian Optimization:

With given dataset **D**, we can construct levelset estimates via the GP as well as assess the (average) (un-)certainty of those estimates

Bayesian Optimization: optimize a given objective function through sequential design, i.e. choose new, optimal, points to evaluate functions to improve the model based on prior information.

balance **exporation** of unknown space vs **exploitation** of already acquired data towards the objective.

answer to the question:

Which point(s) should we evaluate next to improve quality of contours / excursion sets.

Strategy: based on the current model, build an *acquisition function* that indicates quality / helpfulness of new points to reach the *objective* (low uncertainty about excursion sets)

Bayesian Optimization:

For each candidate point x, GP gives us a p.d.f of possible evaluations Y(x). Use this to compute the **expected improvement** in the global quality assessment:

$$\mathrm{acq}(x) = \int \mathrm{d}x' \; H[S(x'|\mathcal{D})] - \mathbb{E}_{y\sim Y(x)} \int \mathrm{d}x' \; H[S(x'|\mathcal{D}\cup(x,y))]$$

Integrand H[S(x')] - E[H[S(x')|Y(x)]] is the **mutual information** between S(x') and Y(x).

$$I(S(x'),Y(x')) = H[S(x')] - \mathbb{E}_{y\sim Y(x)} \; H[S(x'|Y(x')] = \\ H[Y(x)] - \mathbb{E}_{S(x')} \; H[S(Y(x)|S(x')]$$

second formulation H[Y] - E[Y|S] is computationally more tractable.

Bayesian Optimization:

For each candidate point x, GP gives us a p.d.f of possible evaluations Y(x). Use this to compute the **expected improvement** in the global quality assessment:

$$\mathrm{acq}(x) = \int \mathrm{d}x' \; H[S(x'|\mathcal{D})] - \mathbb{E}_{y\sim Y(x)} \int \mathrm{d}x' \; H[S(x'|\mathcal{D}\cup(x,y))]$$

Integrand H[S(x')] - E[H[S(x')|Y(x)]] is the **mutual information** between S(x') and Y(x).

$$I(S(x'),Y(x')) = H[S(x')] - \mathbb{E}_{y\sim Y(x)} \; H[S(x'|Y(x')] =$$
$$H[Y(x)] - \mathbb{E}_{S(x')} \; H[S(Y(x)|S(x')]$$

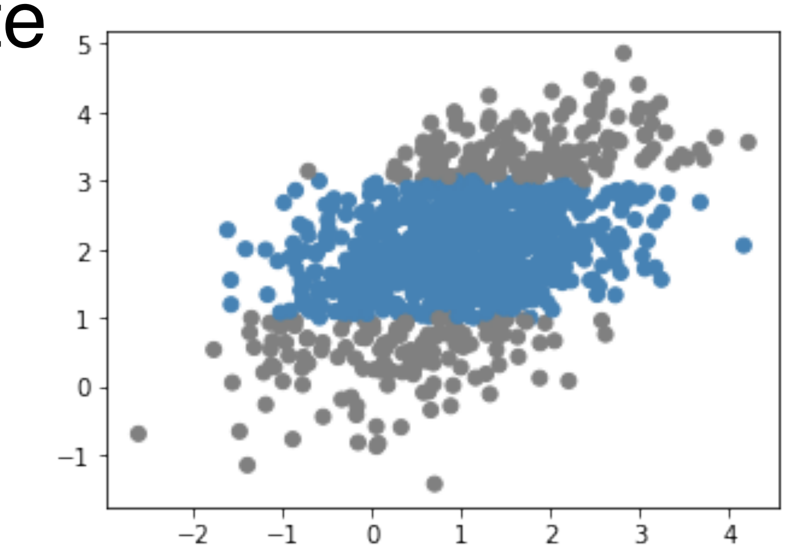
Bayesian Optimization:

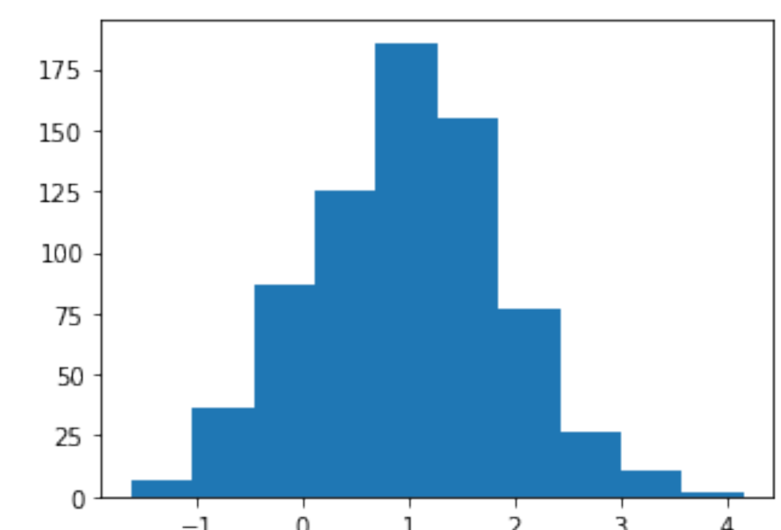
$$I(S(x'), Y(x')) = H[S(x')] - \mathbb{E}_{y \sim Y(x)}\, H[S(x'|Y(x')] =$$
$$H[Y(x)] - \mathbb{E}_{S(x')}\, H[S(Y(x)|S(x')]$$

H[Y]: entropy of a normal distribution with parameters specified by GP

H[Y|S]: entropy of a marginal distribution of bivariate normal distribution with one dimension truncated

moments of this distribution can be derived analytically. Use normal distribution[1] with same moments to approximate entropy H[Y|S]





[1]normal is max entropy distribution, so conservative choice -> more expoloration

Benchmarking

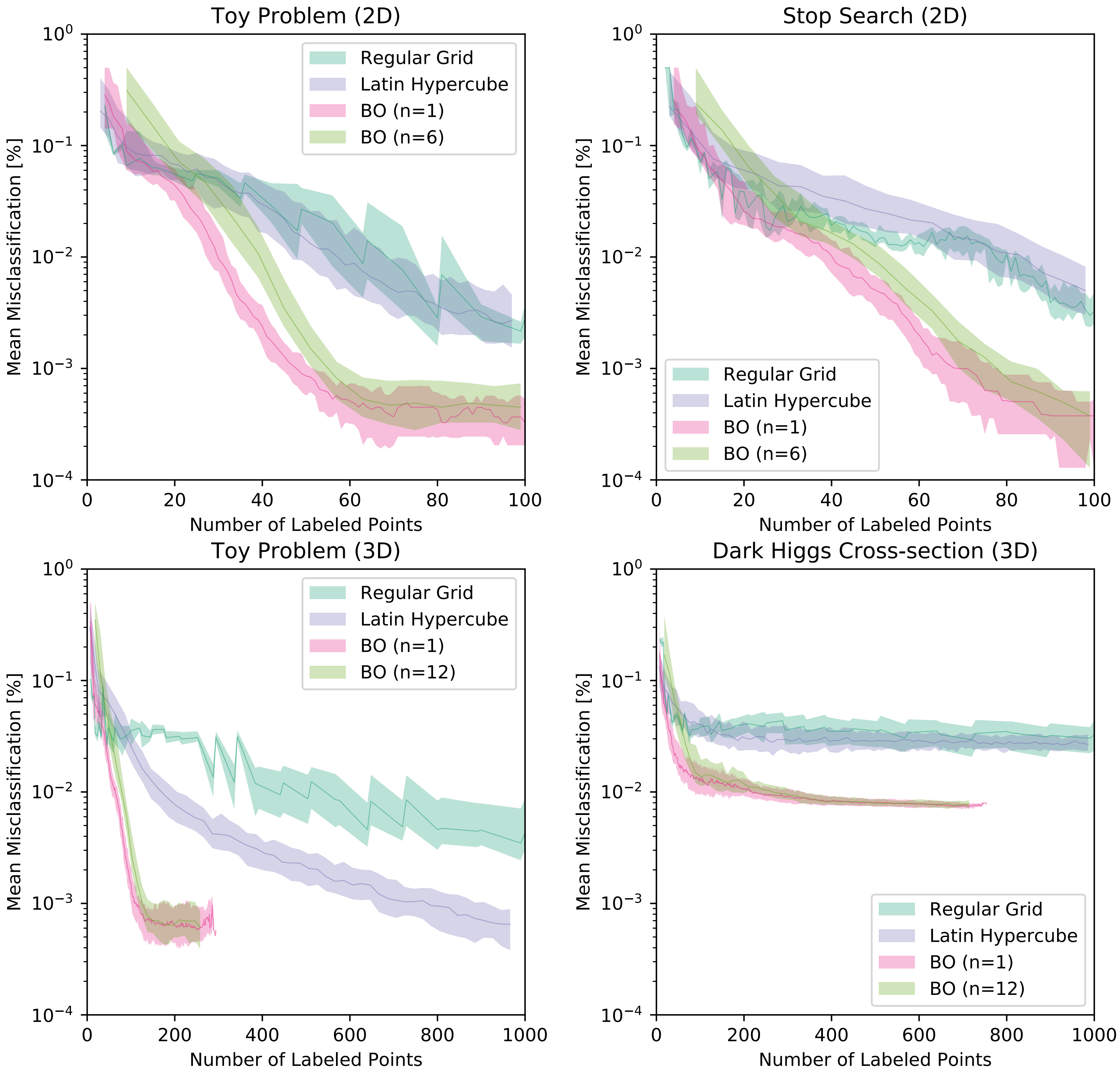
Compared Bayesian Optimization to two strategies for 2D and 3D cases:

1. regular grids with random alignments

2. latin hypercube sampling

BO yields level sets of equal quality with **much** fewer evaluations (e.g. generated samples). E.g. three dimensional parameters scans quite possible < 100 points.

Works reobustly for large batch sizes.



Conclusion:

Regular grids do not scale for high dimensions to determine (iso-)surfaces of scalar functions (such as CLs). Many points irrelevant for determining the surfaces → wasted compute.

Designed Bayesian Optimization algorithm that sequentially incorporates prior information to determine the best points to evaluate next to reach the *objective* (i.e. an accurate contour/(hyper-)surface)

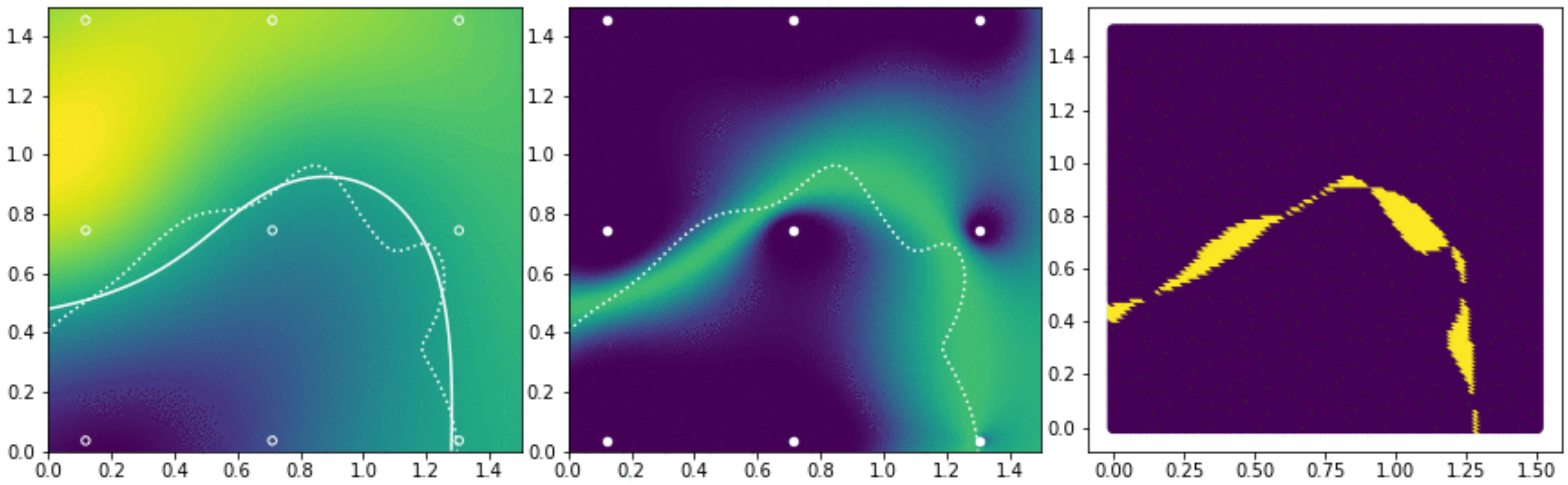
Evaluated on real physics examples (CheckMate, MadGraph) — observerd significant savings potential in computational resources.

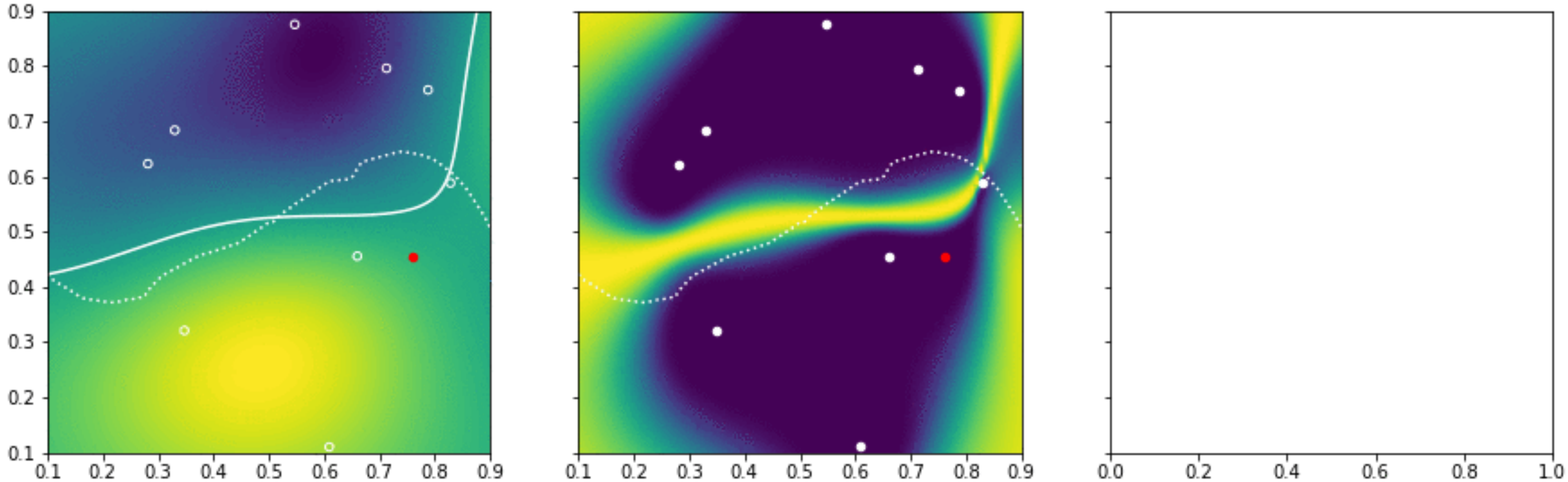
Future Work:

- higher dimensions (pMSSM - 10 / -19 ?) through parallel GP computation
- adaptive batching techniques
- GPUs

Backup

ran_regulargrids 9 points




0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
0.0 0.2 0.4 0.6 0.8 1.0