Learning the Beyond the Standard Model landscape

Sascha Caron (Nikhef and Radboud University)

CERN 15 May 2018

SPOT and idarksurvey: Faruk Diblen, Jisk Attema, Rena Bakshi, Luc Hendriks
BSM-AI: *Sascha Caron Jong Soo Kim, Krzysztof Rolbiecki, Roberto Ruiz de Austri, Bob Stienen, first result was SUSY-AI:* [1605.02797]
Darkmachines: *various*

BSM landscape:

BSM models and their parameters (example MSSM + 19 parameters)

Lets start with a provocative statement

Many many simplified models

are <u>not</u> equal to

a full model

A few steps back – What do we want to know about BSM models ?

I like to know the probability of the **model + parameter set** given the experimental data (or maybe simpler – if a model is excluded or not) for a ***any*** (interesting) model on the arxiv and ***any*** possible set of parameters

Is this possible ?

How ?

Why is this useful ?

- Current practice: Publish experimental model constraints on a 2-dimensional piece of paper

  Drawbacks:

  - either a projection or a simplification of the full parameter space

  - If simplification → see 1st slide.

  - If projection → What if other projection needed ?

- I am convinced that Machine Learning **is <u>the</u> <u>way</u>** to store/encode our BSM results in the 21th century.

Why use SUSY-AI / BSM –AI ?

- Fast statistical results based on earlier analyses
- High accuracy by learning hard-to-see relations in data
- Works also in submodels of the learned model (e.g. mSUGRA)
- Providing confidence levels on prediction

- New way to publish and recast multivariate data
- Creating plots not present in paper
- Re-usability and persistence of analysis and results

How could this work ? An example

- Need to have data points to train on (ATLAS, CMS, recasting tools which run MC simulations, e.g. checkmate)
- Train a Machine Learning regression/classification tool to interpolate between the data points → generalize the result

SPOT and idarksurvey

Aim 1: Quick recast of plots /figures for pheno models (and other high-parameteric models)

Aim 2: Collect model preditions/evaluations or “training data for BSM-AI”

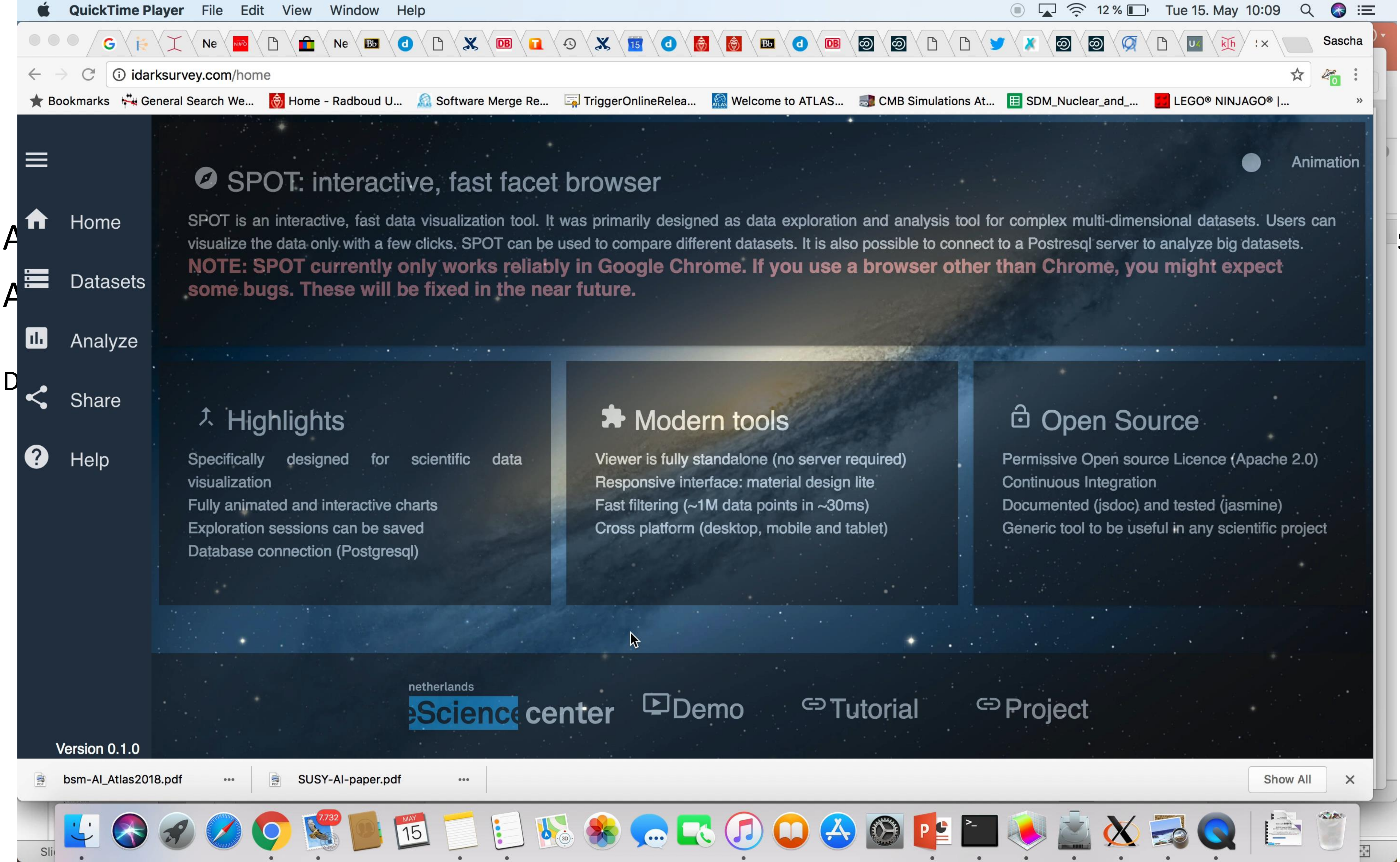
Demo on www.idarksurvey.com

A

A

D

s)



QuickTime Player File Edit View Window Help
Tue 15. May 10:09
idarksurvey.com/home
Home
Datasets
Analyze
Share
Help
Version 0.1.0
Animation
SPOT: interactive, fast facet browser
SPOT is an interactive, fast data visualization tool. It was primarily designed as data exploration and analysis tool for complex multi-dimensional datasets. Users can visualize the data only with a few clicks. SPOT can be used to compare different datasets. It is also possible to connect to a Postresql server to analyze big datasets.
NOTE: SPOT currently only works reliably in Google Chrome. If you use a browser other than Chrome, you might expect some bugs. These will be fixed in the near future.
Highlights
Specifically designed for scientific data visualization
Fully animated and interactive charts
Exploration sessions can be saved
Database connection (Postgresql)
Modern tools
Viewer is fully standalone (no server required)
Responsive interface: material design lite
Fast filtering (~1M data points in ~30ms)
Cross platform (desktop, mobile and tablet)
Open Source
Permissive Open source Licence (Apache 2.0)
Continuous Integration
Documented (jsdoc) and tested (jasmine)
Generic tool to be useful in any scientific project
netherlands eScience center
Demo
Tutorial
Project
bsm-AI_Atlas2018.pdf
SUSY-AI-paper.pdf
Show All

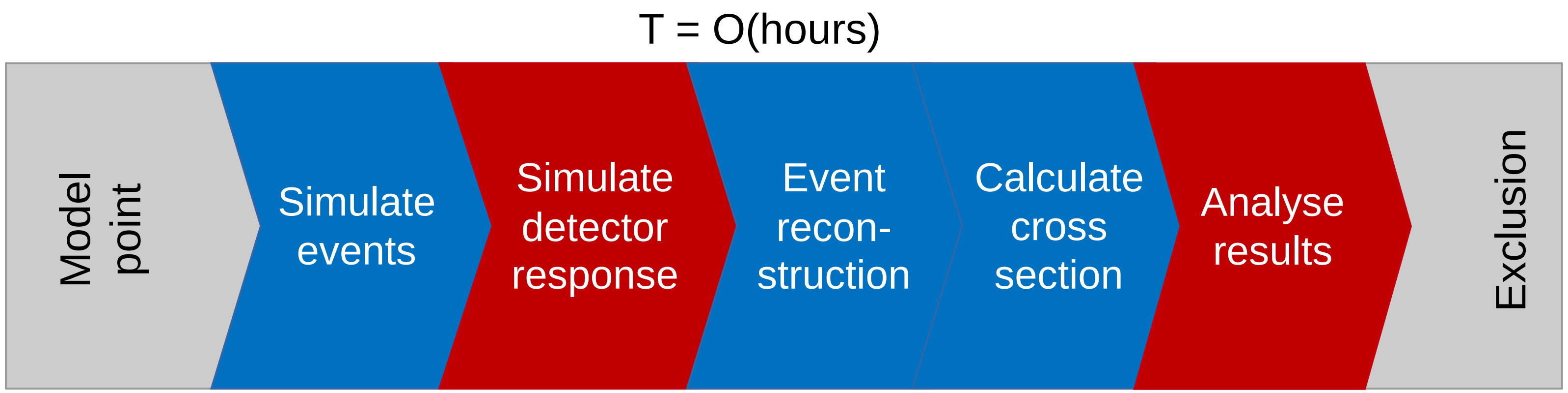
idark: intelligent (dark) model survey

SUSY-AI is a part of BSM-AI and BSM-AI is a part of the idark project

SPOT: Faruk Diblen, Jisk Attema, Rena Bakshi

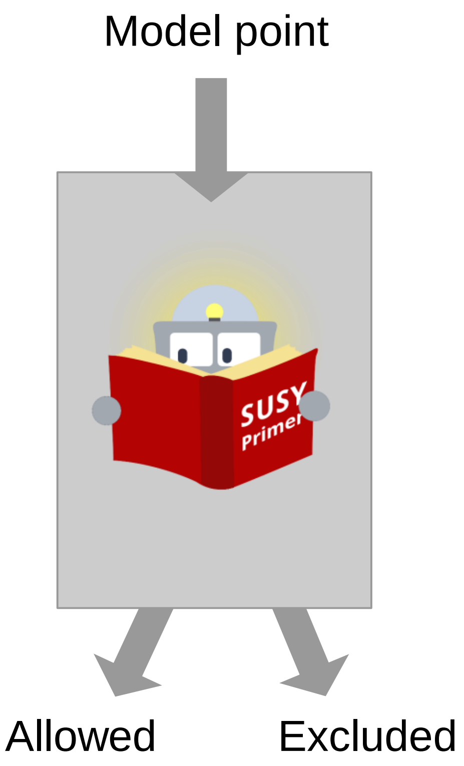
BSM-AI: *Sascha Caron Jong Soo Kim, Krzysztof Rolbiecki, Roberto Ruiz de Austri, Bob Stienen, first result was SUSY-AI:* [1605.02797]

ATLAS analysis chain for each of the 300000 model points



Machine Learning Applied

- Training data: model points in supersymmetric model with only phenomenologically relevant parameters (pMSSM)
  source: ATLAS [1508.06608]
- Testing data: independent (unseen) data



Is currently a classification algorithm within scikit-learn (a Random Forest)

For 13 TeV limits we use also:
Alan Barr and Jesse Liu,
[arXiv:1605.09502]

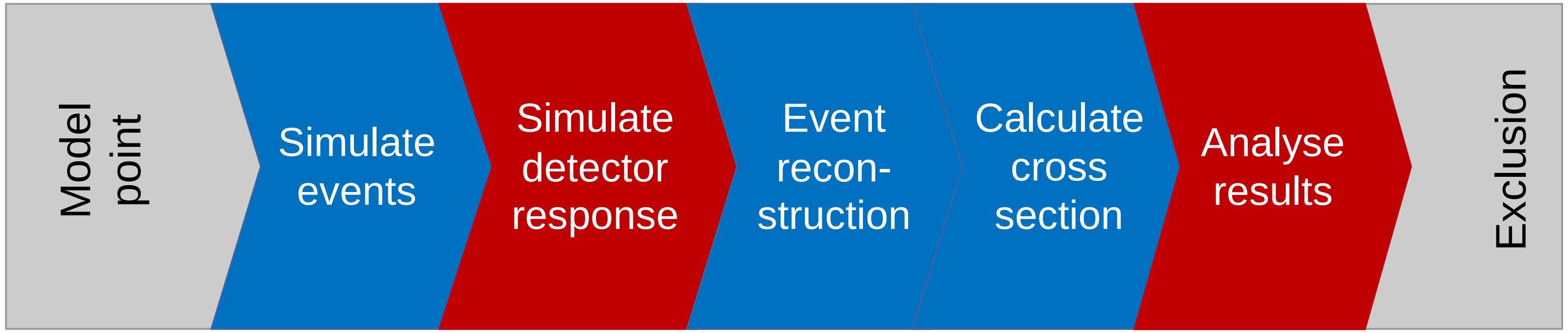
With Machine Learning we
use smart “generalization” to
go from discrete data to a continuous function

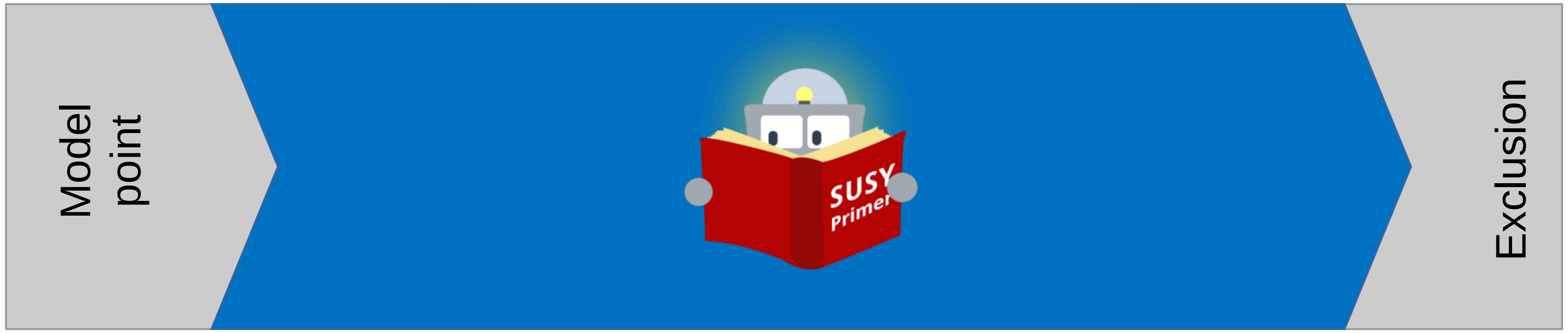
→ We get information on the parameter space **“in between” the points and information on the quality of our sampling**

→ We can “learn” the exclusion boundary (or even the likelihood or confidence level) in the full parameter space of the **MSSM19**

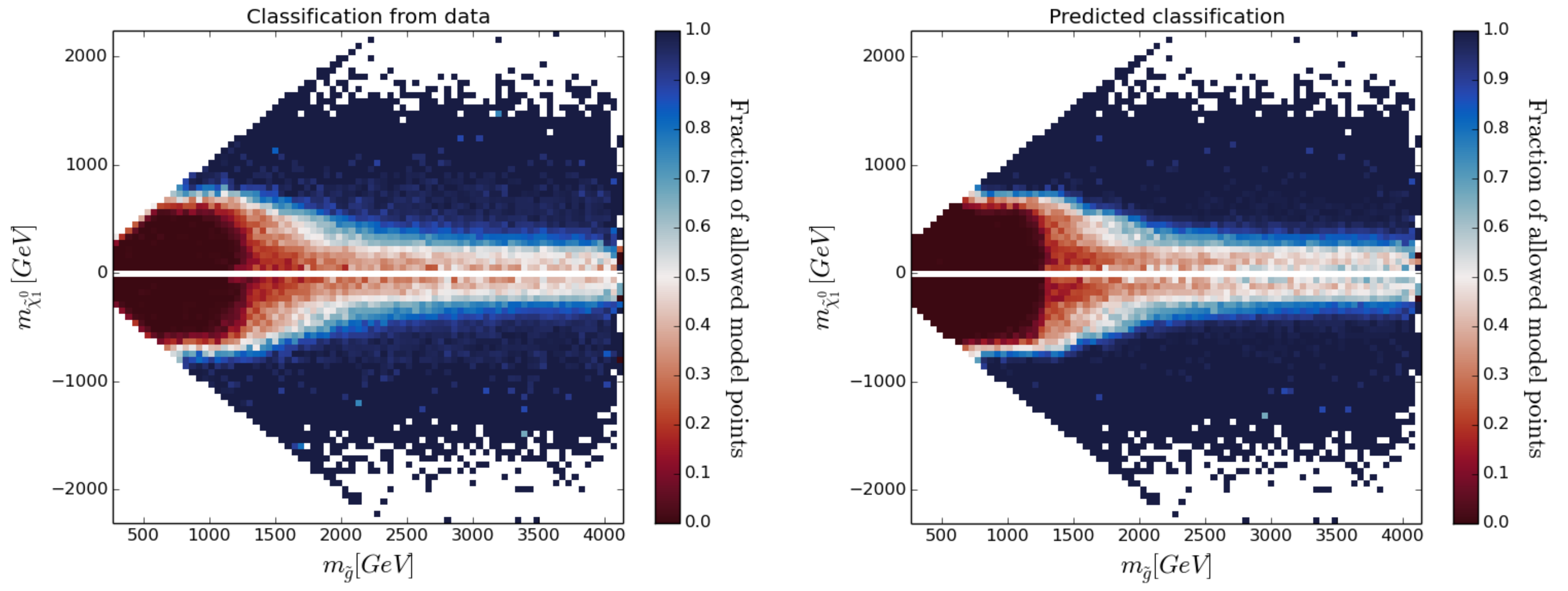
Exclusion analysis

T = O(hours)

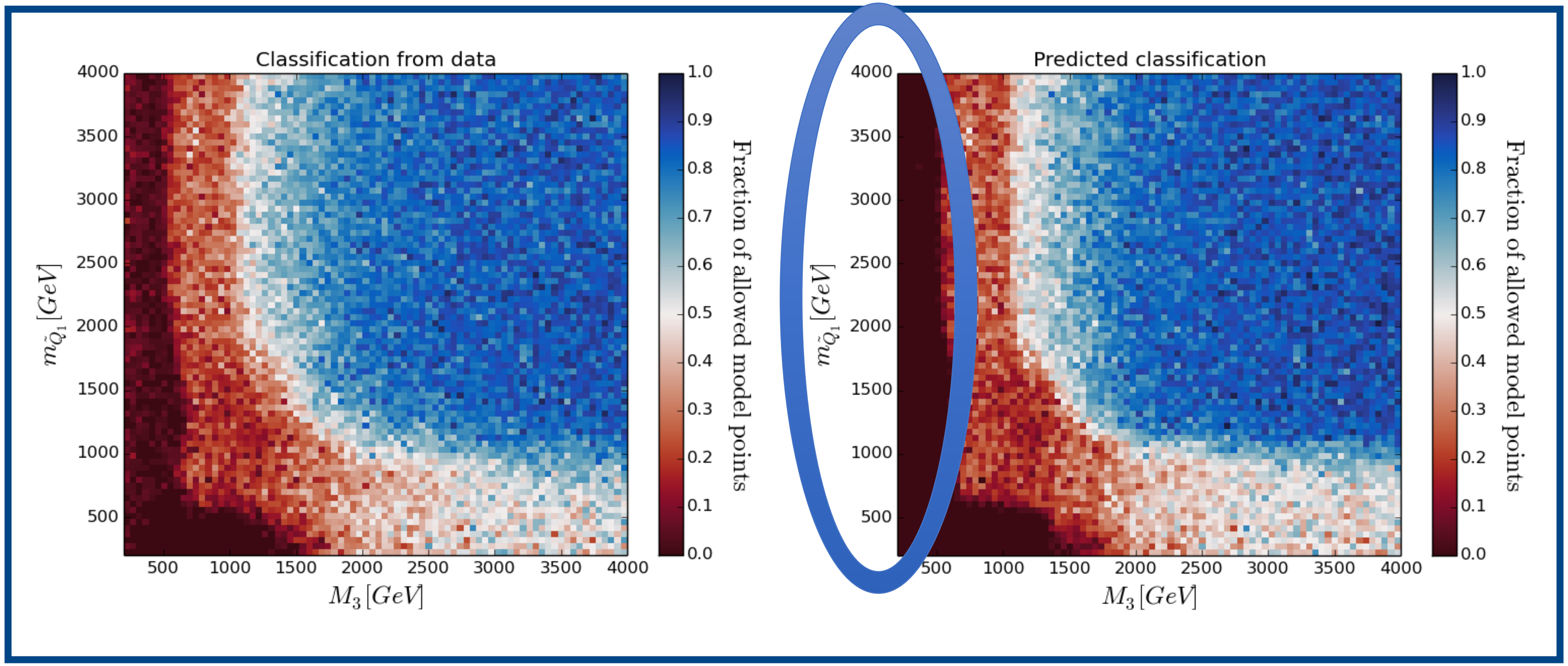




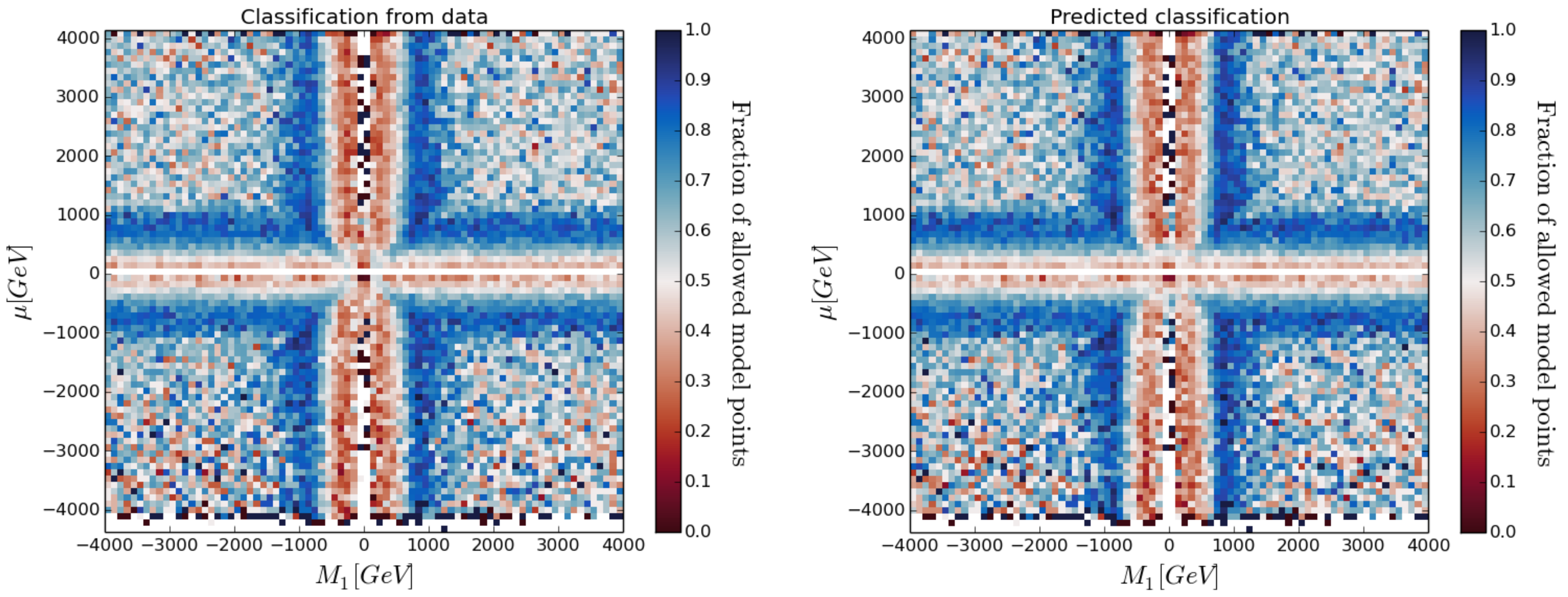
T = < O(1 ms)



All plots here by Bob Stienen !



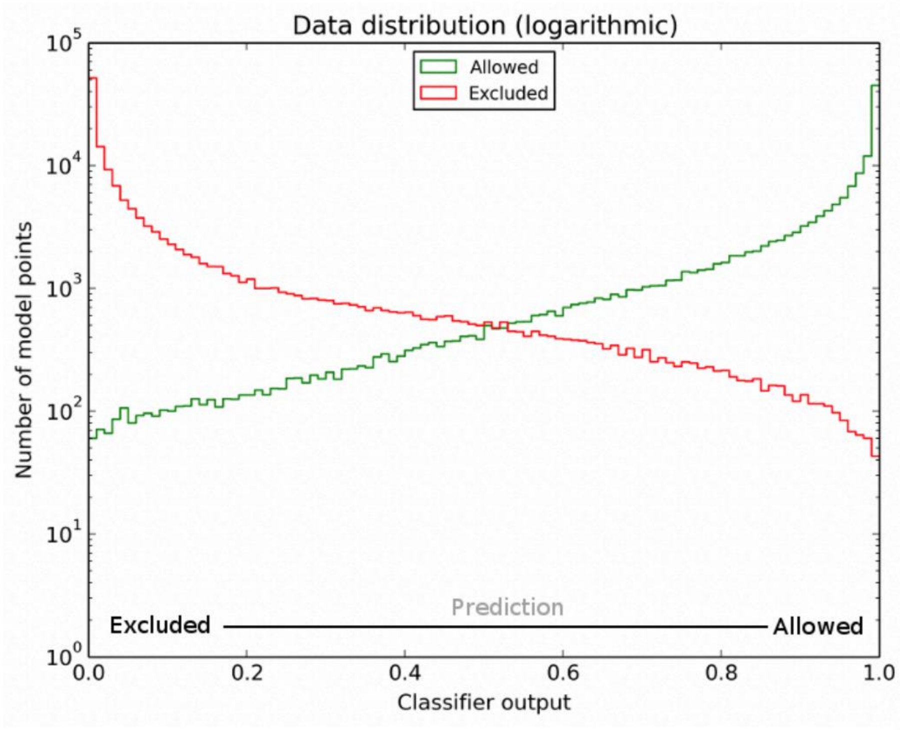
Here SUSY-AI is not perfect
but we have more information
than just excluded or not.
→ SUSY-AI output variable

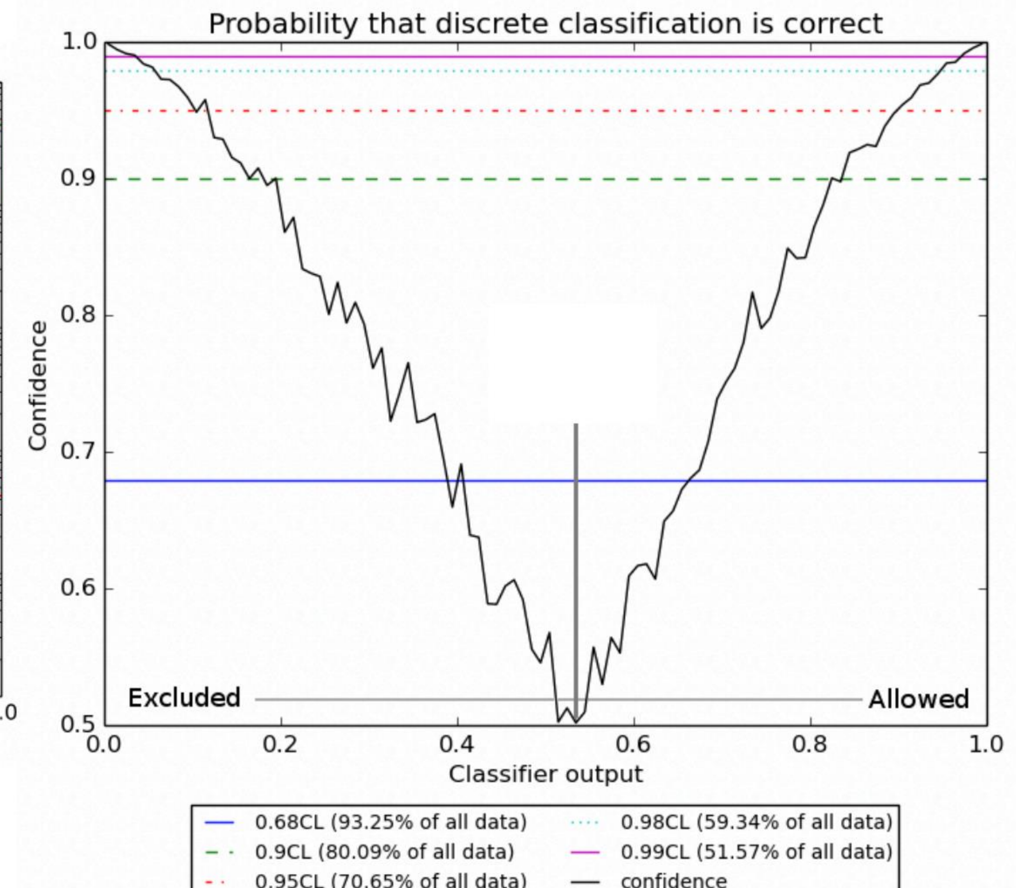

Classification from data
Predicted classification
Fraction of allowed model points
$\mu[GeV]$
$M_1\,[GeV]$
4000
3000
2000
1000
0
−1000
−2000
−3000
−4000
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Used training data to learn classification

It determines a **confidence** level of
its **classification** using the training data.

Ratio of majority class per bin

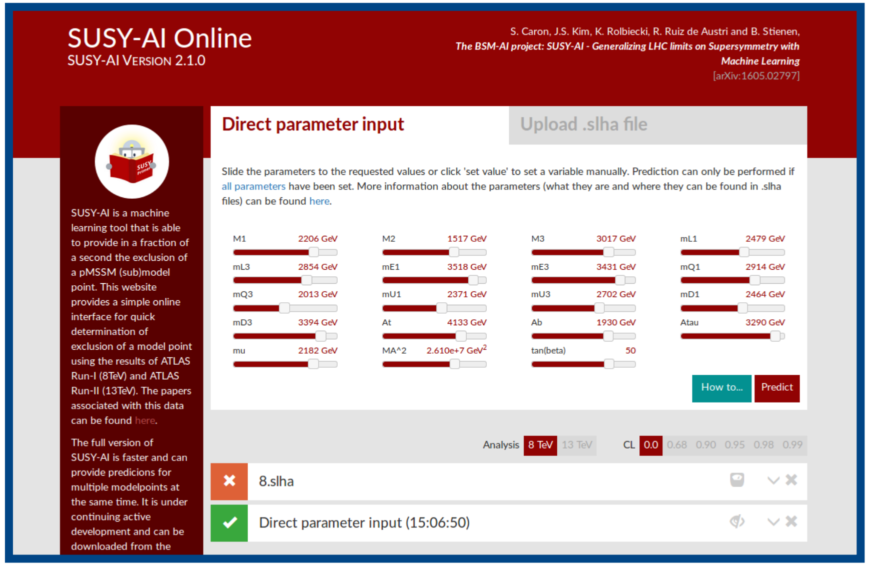




SUSY-AI (Online)

- Tool has been published
  https://susyai.hepforge.org
  - Python interface to classifier
  - Scikit-learn package for ML implementation

- Online interface
  http://susy-ai.org/
  - All functionalities except batch predictions
  - Predictions in < 2 seconds



Steering the LHC analyses:
Where does SUSY-AI like to have more points ?

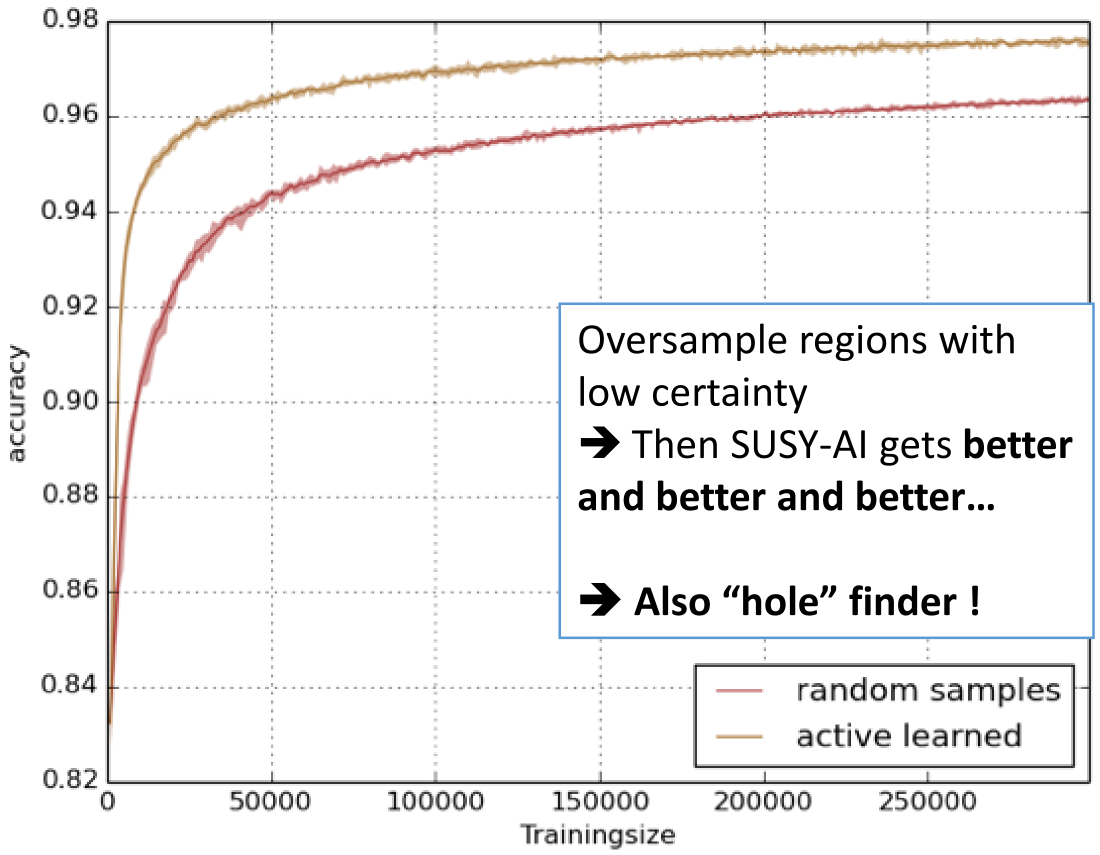
A) We like have more points where SUSY-AI is less certain
→ Sample regions with low SUSY-AI "certainty"

B) We also want have points as "targets" in holes
→ Sample islands with "non-excluded" with high "Certainty"

Timeline: Provide a list of targets
make ATLAS-internal version of SUSY-AI using 13 TeV scans
Provide again a list of targets

→ **Until we have learned and excluded the low mass MSSM19 with >99 % accuracy**

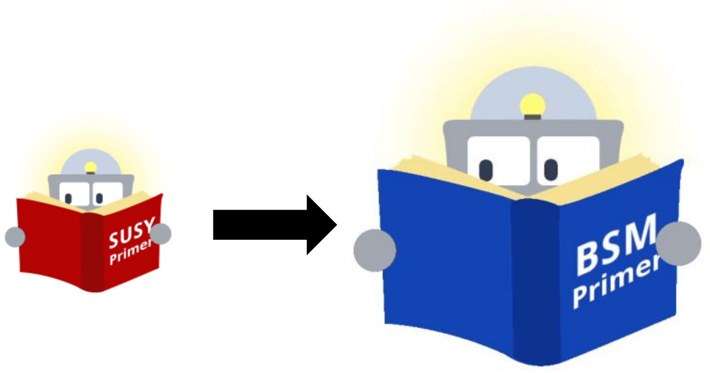
How to improve ? Active Learning



Oversample regions with low certainty
➔ Then SUSY-AI gets **better and better and better…**

➔ **Also "hole" finder !**

BSM-AI

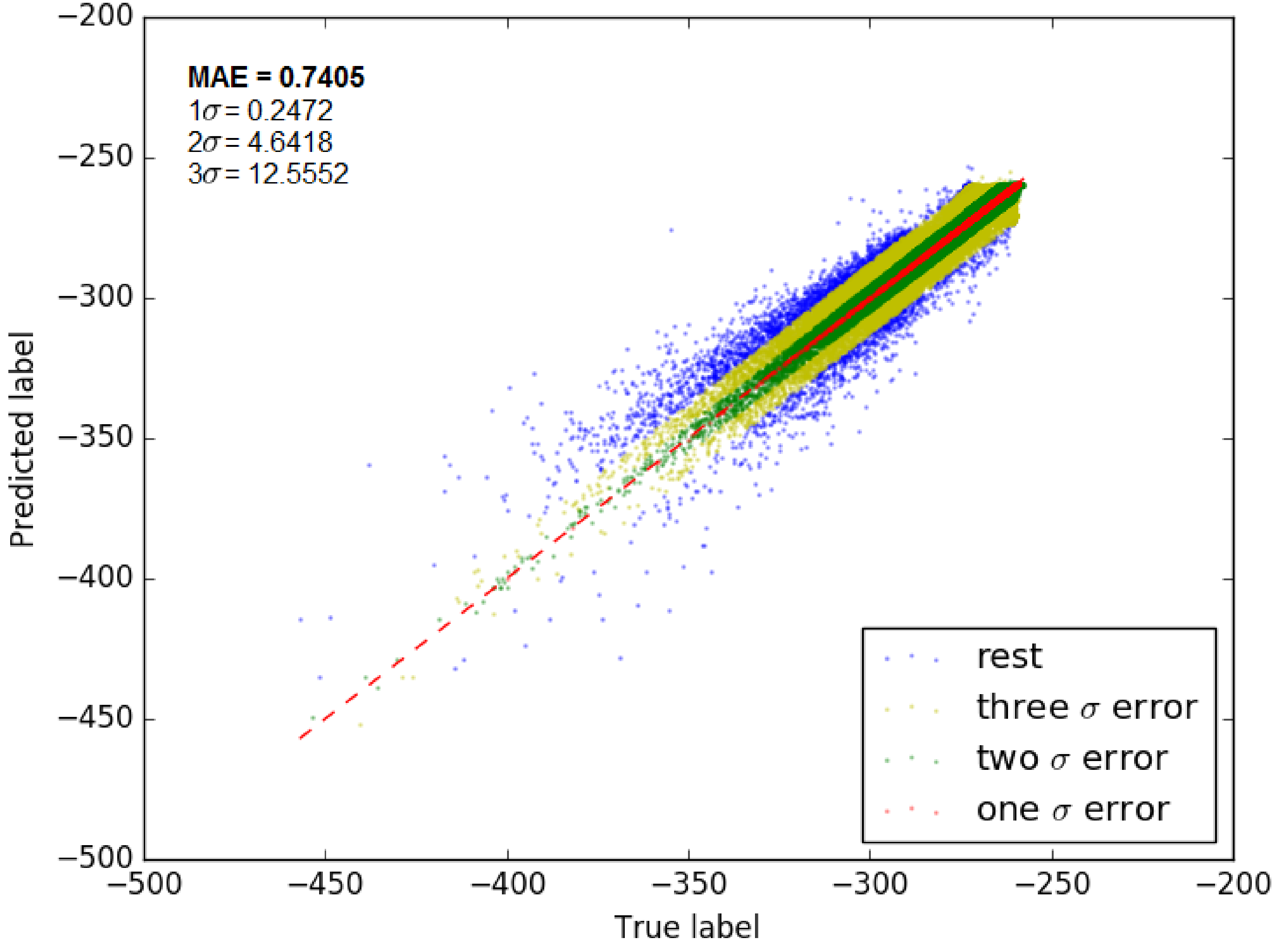


- Python package
- **Framework** allowing generalization to ***any* model** stored as a ML algorithm constructed by
  - scikit-learn
  - keras + tensorflow

- Online library of trained algorithms
- Allows remote querying via server-client structure
  only 1 instance needed for entire parallelized pipeline

- Currently finalizing documentation
  looking for enthusiastic testers!

BSM-AI regression example… Learning GAMBIT likelihoods

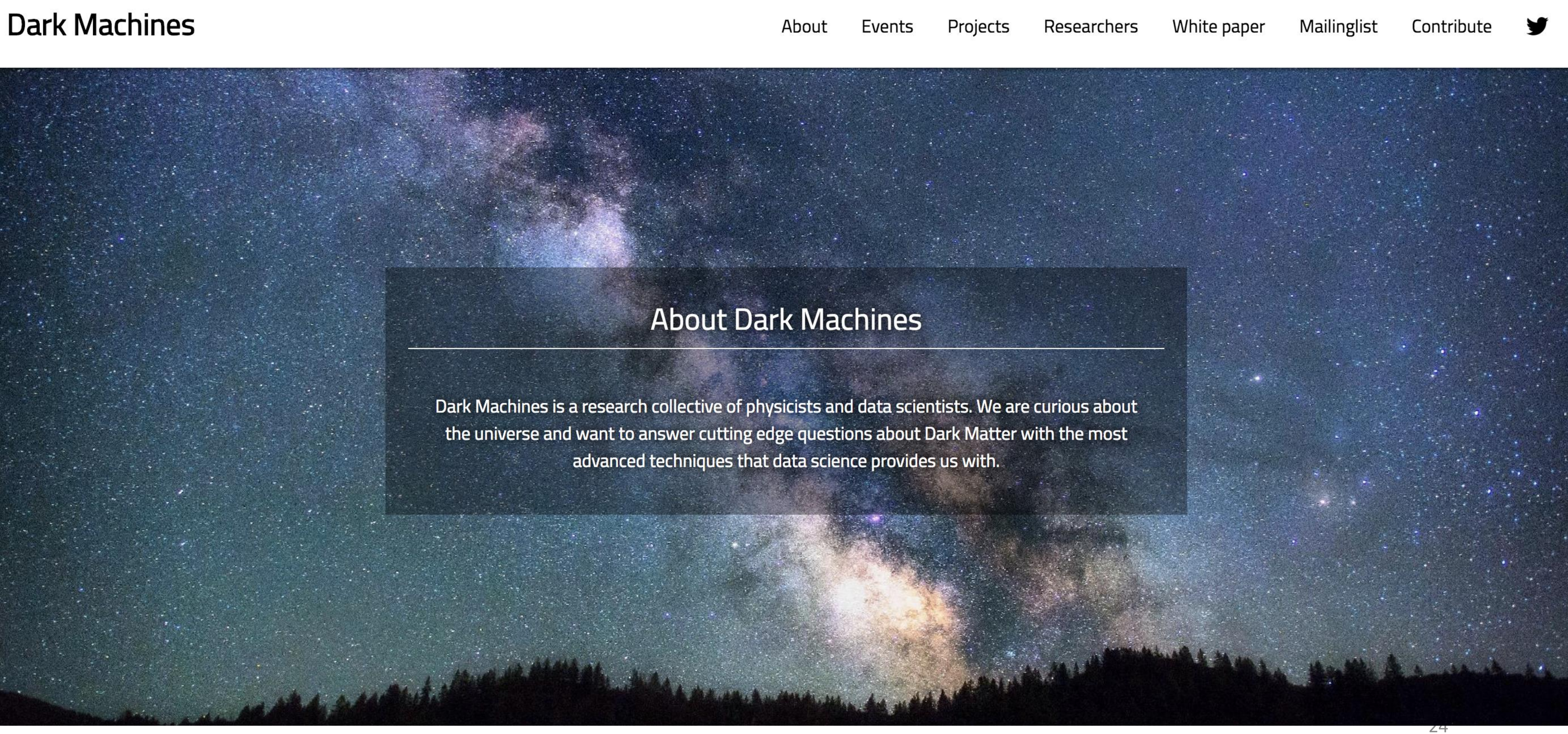


MSSM - 7

https://arxiv.org/abs/1705.07917

Plot by Sydney Otten

Join www.darkmachines.org



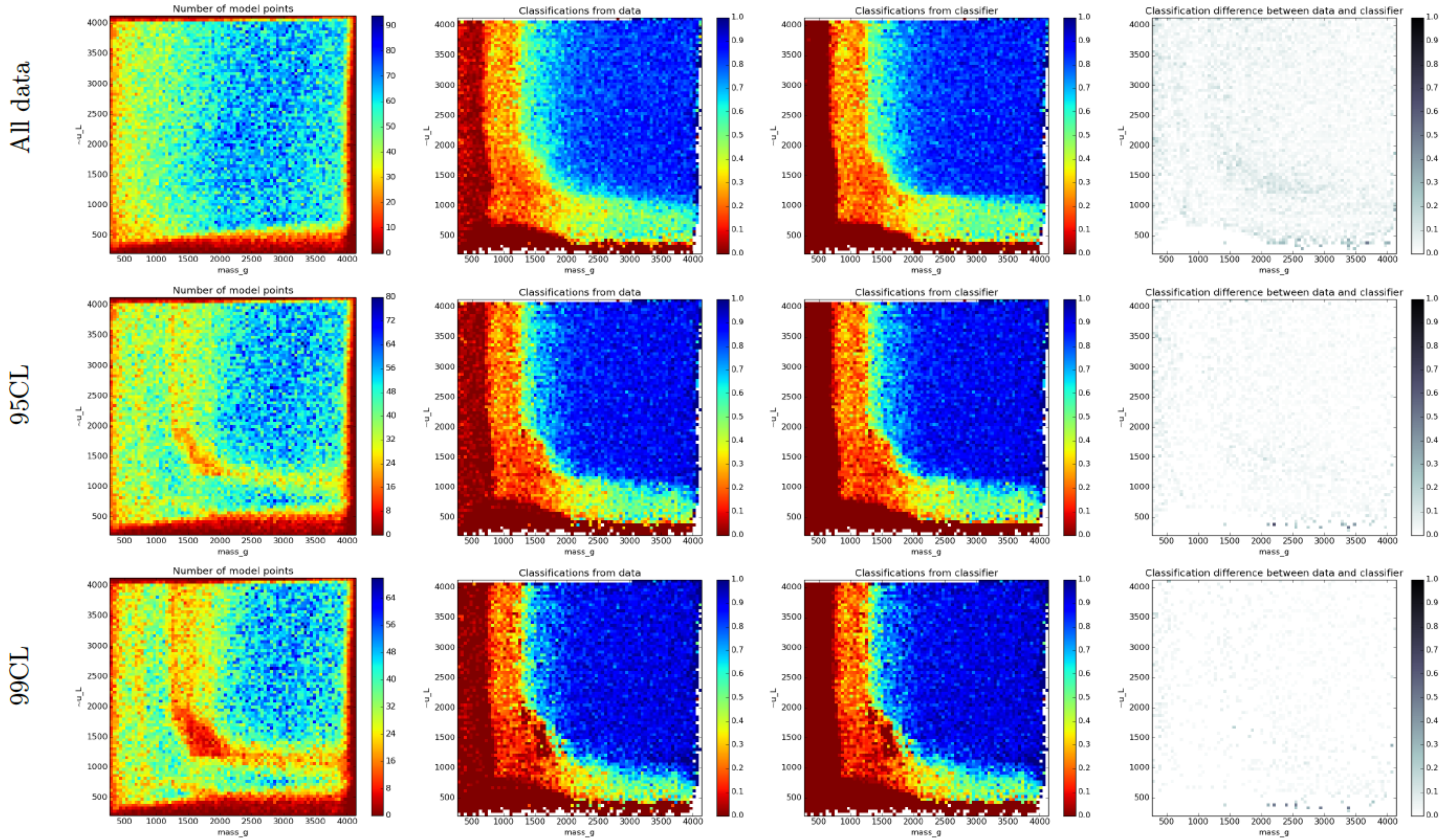
Summary

- We propose to explore models in full parameter space
- Store solutions (at idarksurvey.com)
- Train Machine Learning on the model information

  ===> Prototype is SUSY-AI

  ===> generic catalogue to store all those ML files will be BSM-AI

Number of model points | True classification | Prediction by classifier | Difference between classification and prediction

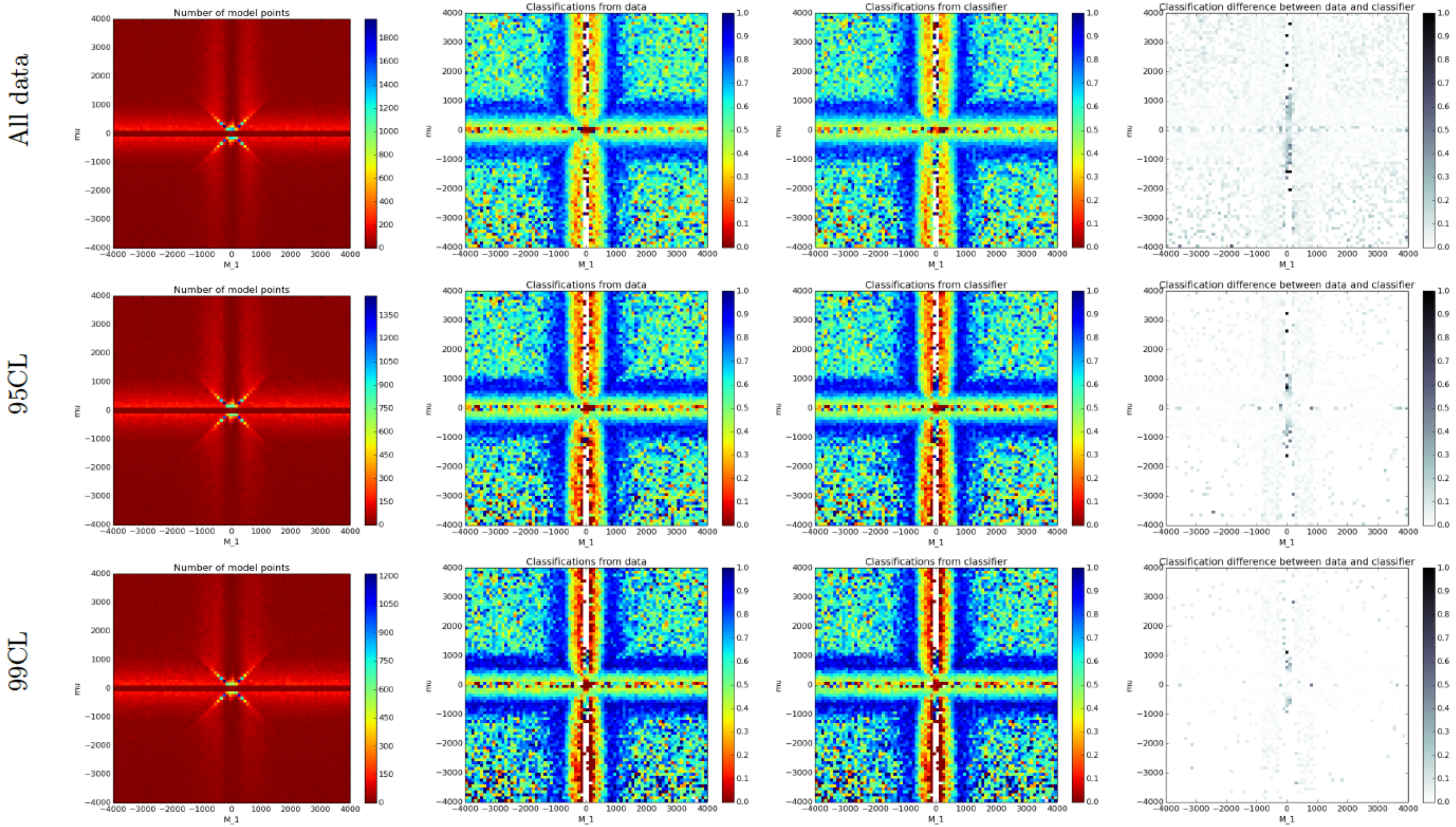


Number of model points

True classification

Prediction by classifier

Difference between classification and prediction

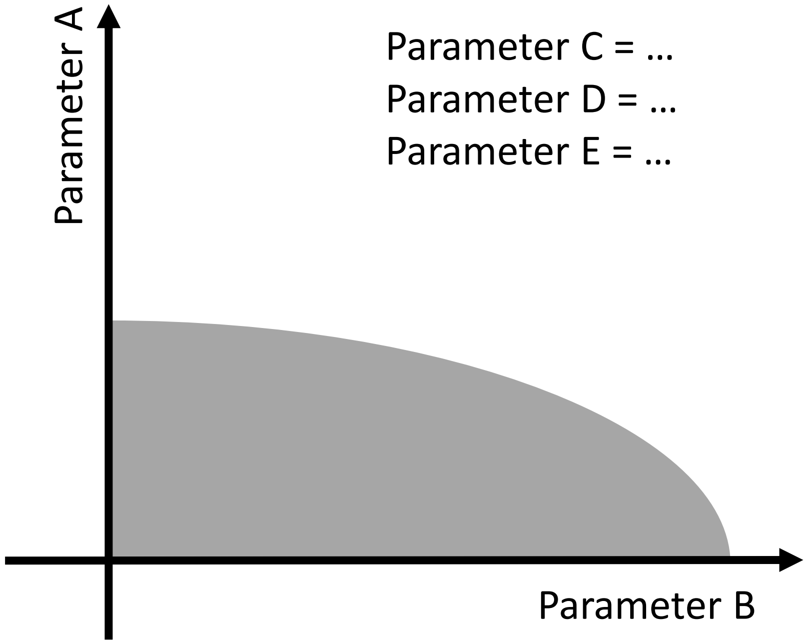


Model exclusion in Particle Physics

We are used to publish on a piece of paper…
i.e. in 2 dimensions.

<u>What we usually do in ATLAS ?</u>

a) Forget about the 6 dimensional model, take a "simplified" model with only A and B
b) Set parameter C= …, D= …, E=… and plot A vs B
c) More sophisticated: Show projections of the model likelihood on parameters A, B as well as C,D etc.



New idea

Publish many model evaluations:
Likelihood (A=12, B=3, C=4, D=5, E=8)
Likelihood (A=5, B=9, C=6, D=2, E=3)