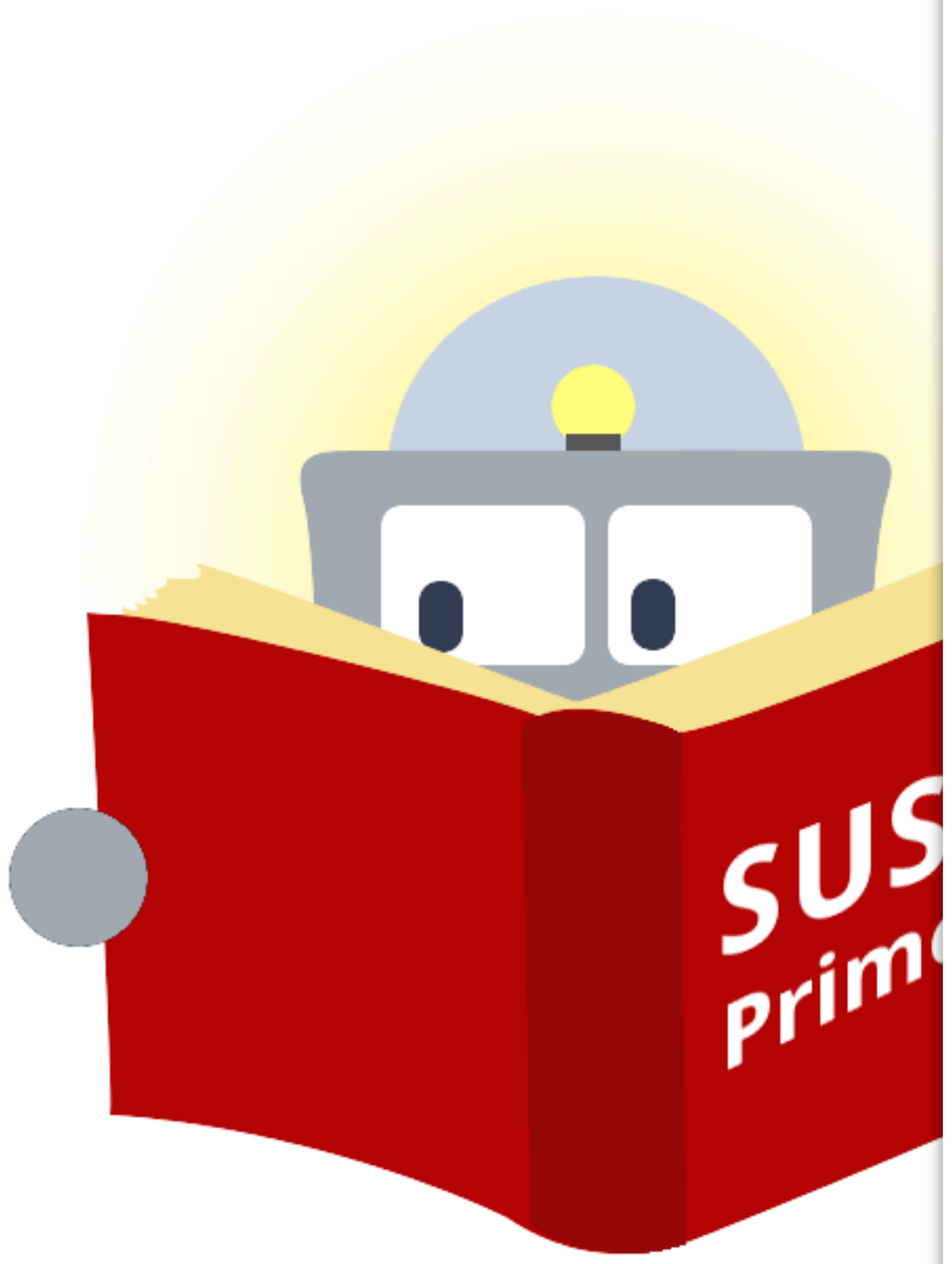
[arXiv:1605.02797]



The BSM-AI project

SUSY-AI: Reinterpreting SUSY LHC Limits with Machine Learning

Sascha Caron, Jong Soo Kim, Krzysztof Rolbiecki,
Roberto Ruiz de Austri, Bob Stienen

netherlands eScience center

Radboud University

Supersymmetry (SUSY)

- Theoretical model of new physics, introducing a symmetry between fermions and bosons

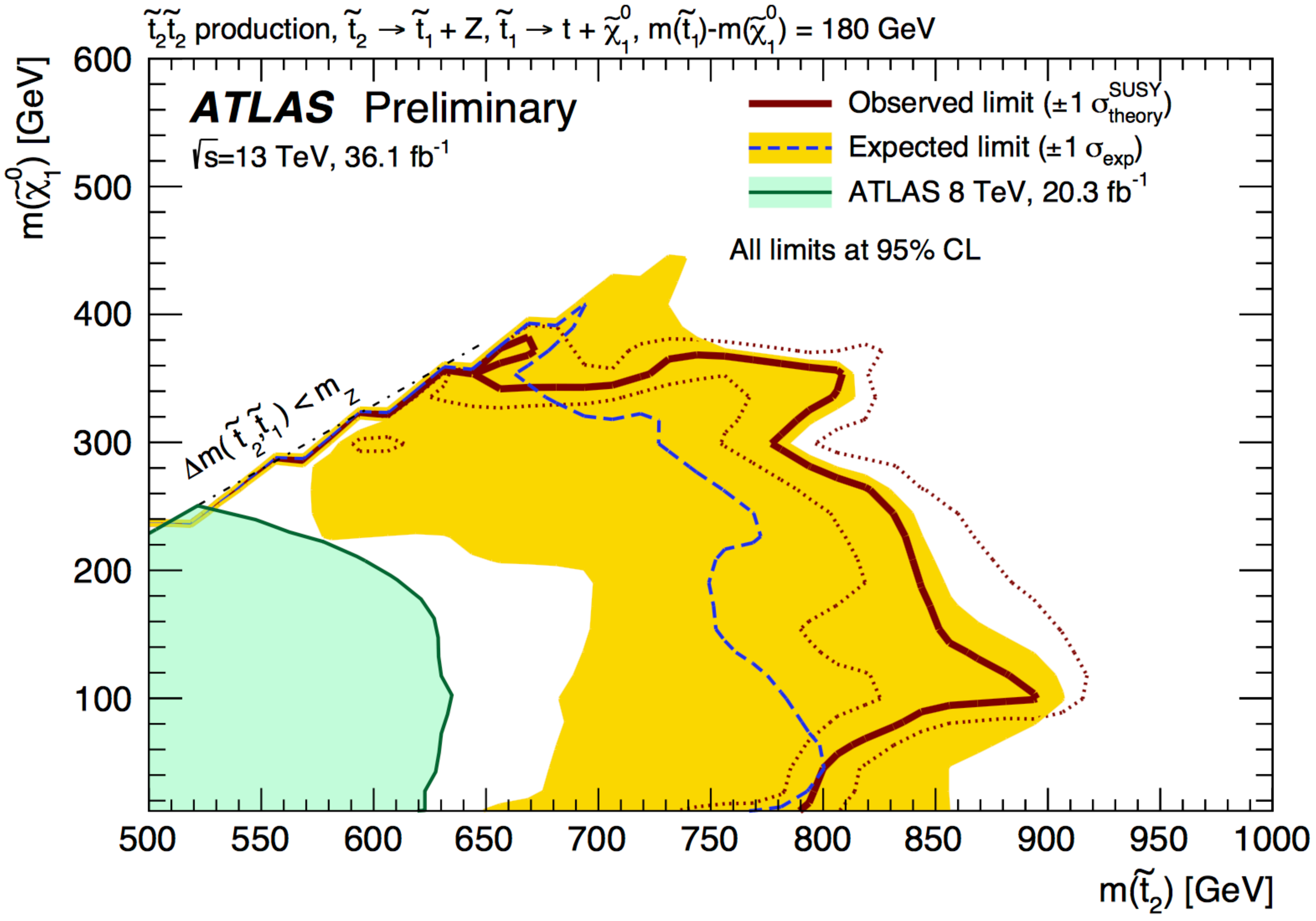
- Minimal version (MSSM) adds ~O(100) free parameters

- ~19 parameters if only looking at the phenomenologically relevant ones (pMSSM)

- No superpartners are discovered yet…

The Plot Problem

atlas_conf_2017_019



The Analysis Problem

- Often analyses are only done on simplified models
- Detector simulations only available within the experimental collaboration
- Determination of exclusion of a single model point is resource intensive (~O(CPU hours))

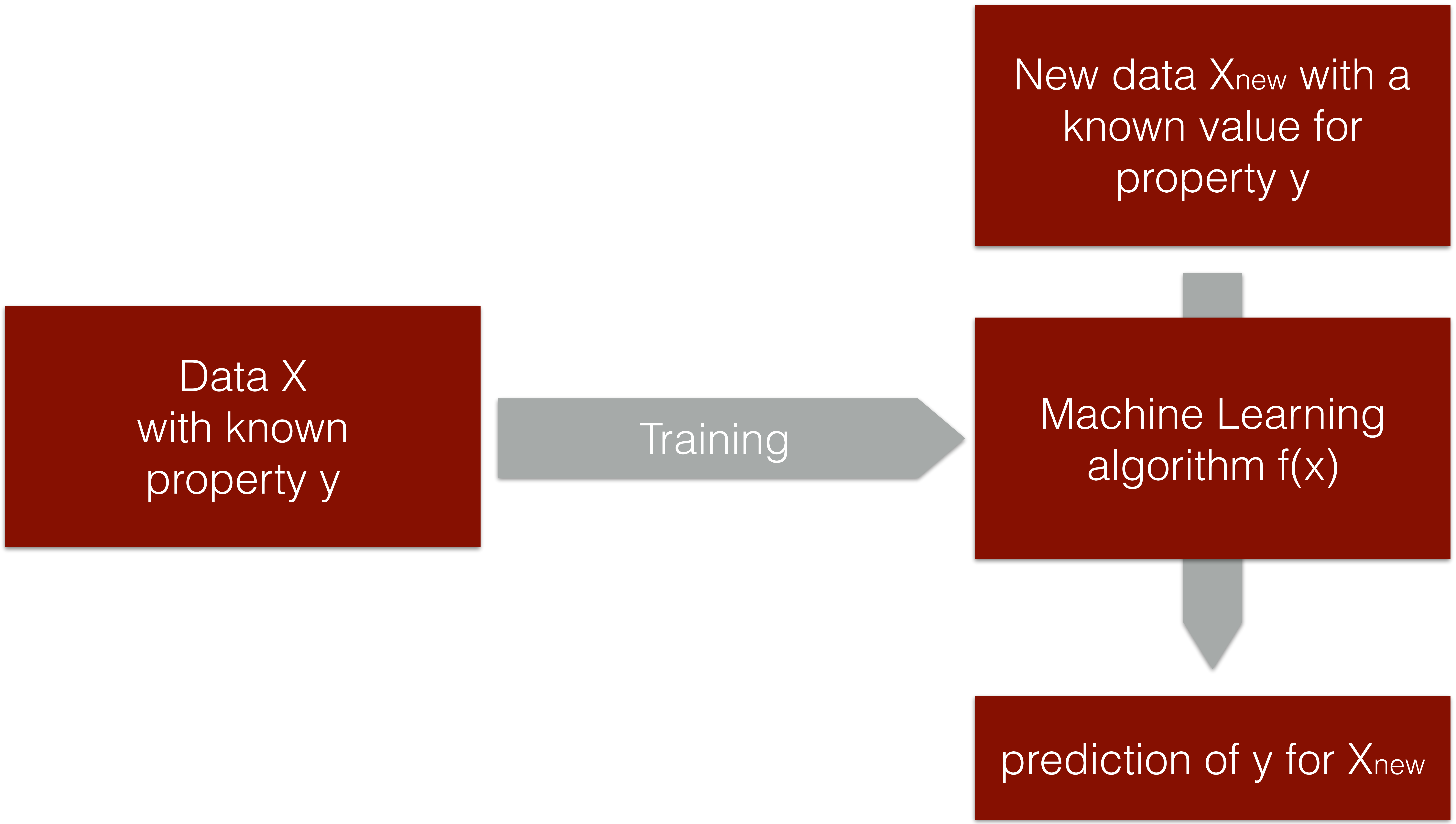
Machine Learning

- Statistics of big data
  - Prediction of data properties based on example (training) data via smart interpolation
- Wide range of algorithms…
  (e.g. boosted decision trees, k-nearest neighbours, neural networks)
- … and applications
  (e.g. health care, advertising, finance, spam detection, car auto pilots)

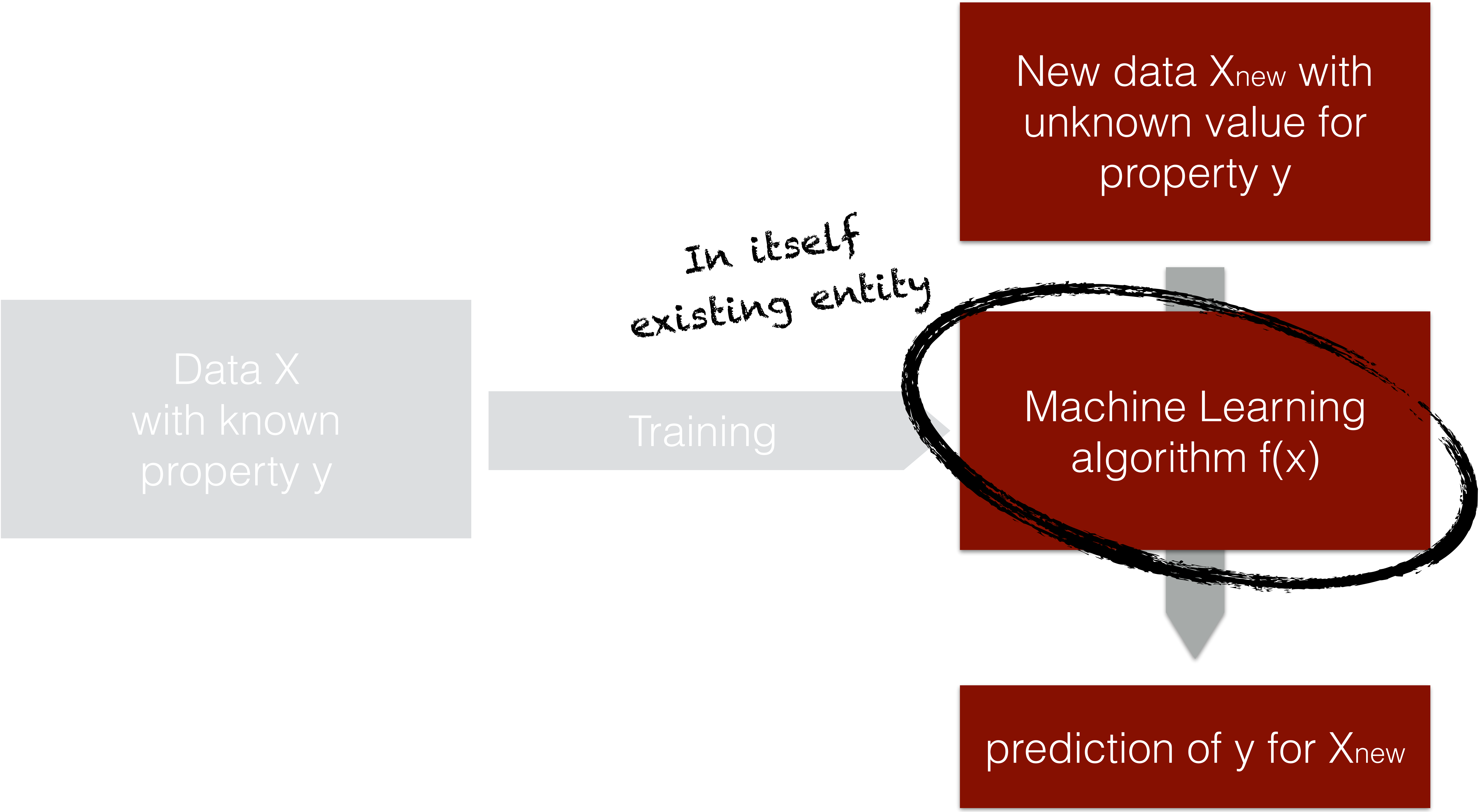
Machine Learning

f(x) predicts y

Machine Learning



Machine Learning



The idea

Machine Learning as a tool to reinterpret experimental results and to determine the exclusion of model points

Training data

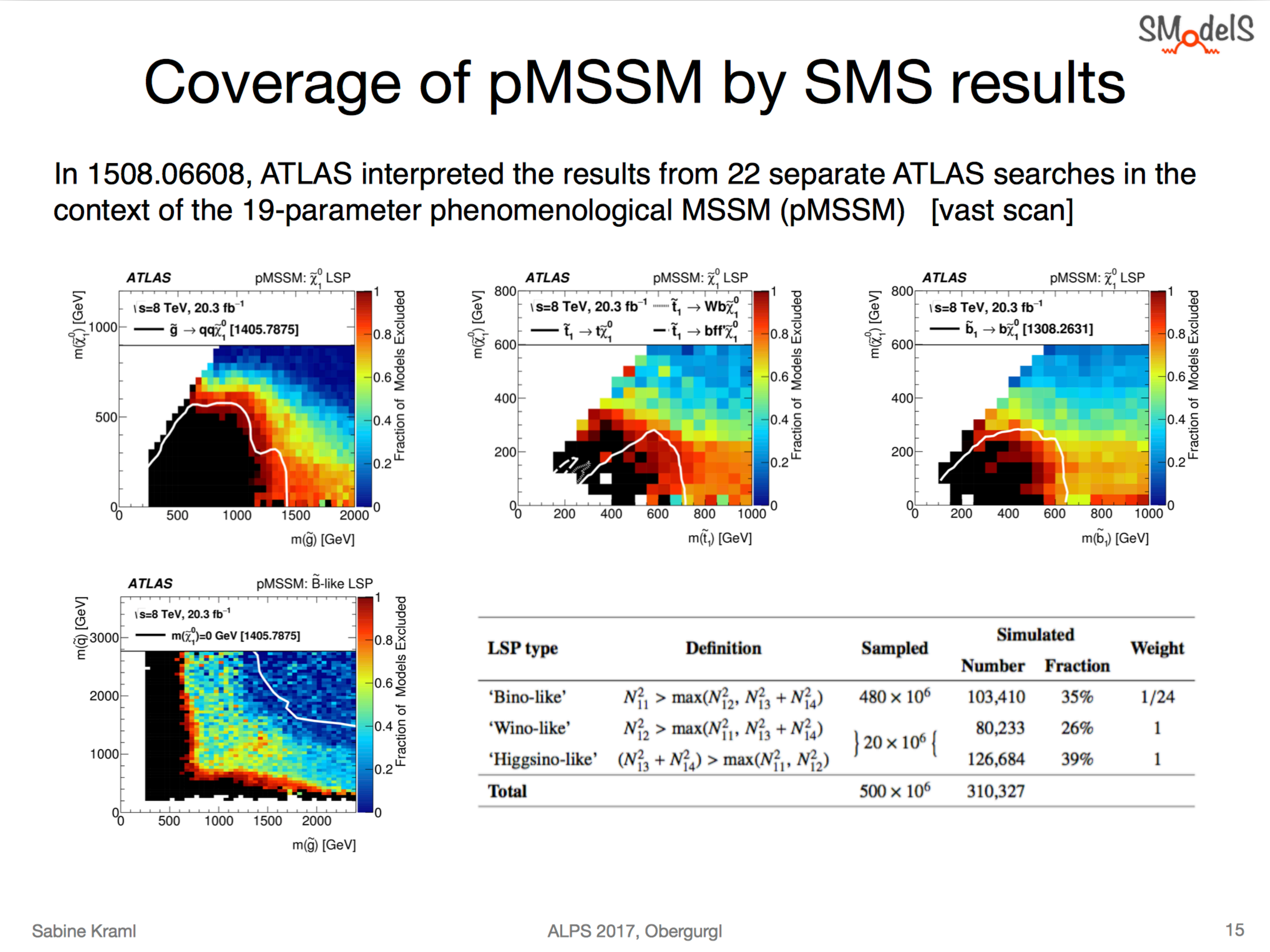
>300,000 model points with exclusion as determined by:

- ATLAS at 8TeV [arXiv: 1508.06608]
- Barr & Liu at 13TeV [arXiv: 1605.09502]

All data has correct Higgs mass and relic density, and is not excluded by precision experiments (LHCb, e.g. B_s decay) or by LUX or Xenon100

Algorithm

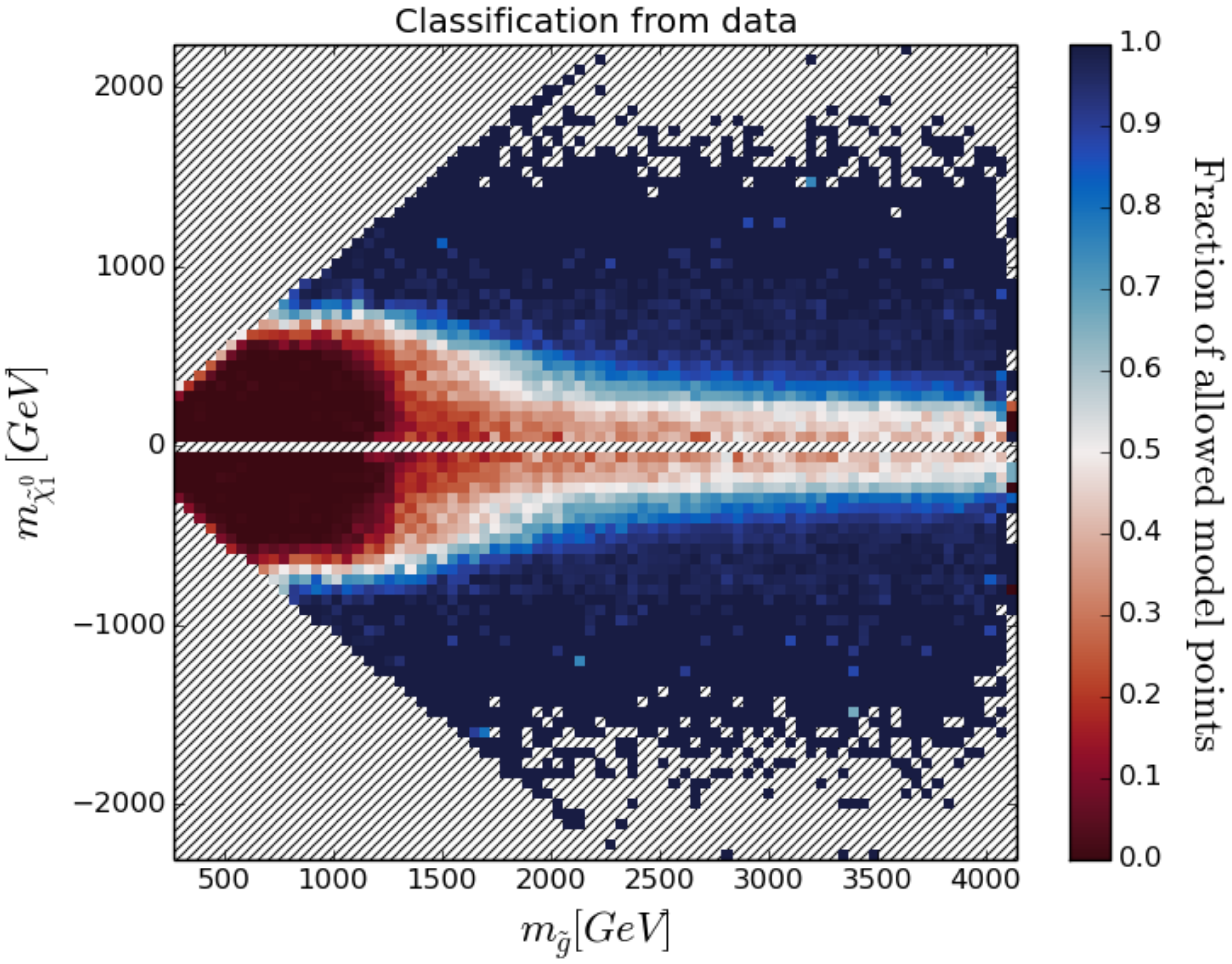
Random Forest (a smartly constructed set of decision trees) in scikit-learn Python package



LSP type	Definition	Sampled	Simulated Number	Simulated Fraction	Weight
'Bino-like'	$N_{11}^2 > \max(N_{12}^2, N_{13}^2 + N_{14}^2)$	480×10^6	103,410	35%	1/24
'Wino-like'	$N_{12}^2 > \max(N_{11}^2, N_{13}^2 + N_{14}^2)$	20×10^6	80,233	26%	1
'Higgsino-like'	$(N_{13}^2 + N_{14}^2) > \max(N_{11}^2, N_{12}^2)$		126,684	39%	1
Total		500×10^6	310,327		

Sabine Kraml's talk

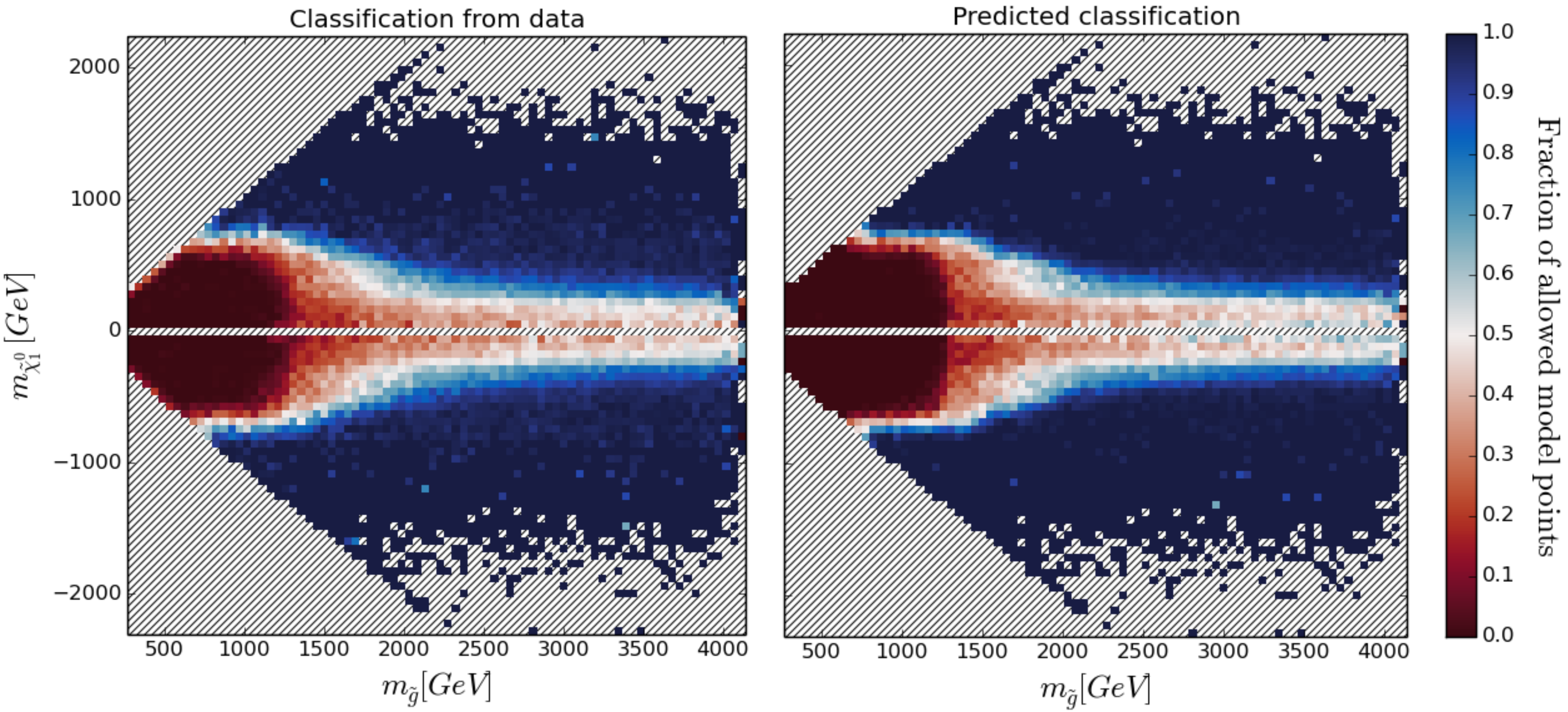
Performance gluino vs neutralino1



Performance gluino vs neutralino1

93.2% accuracy @ 8TeV

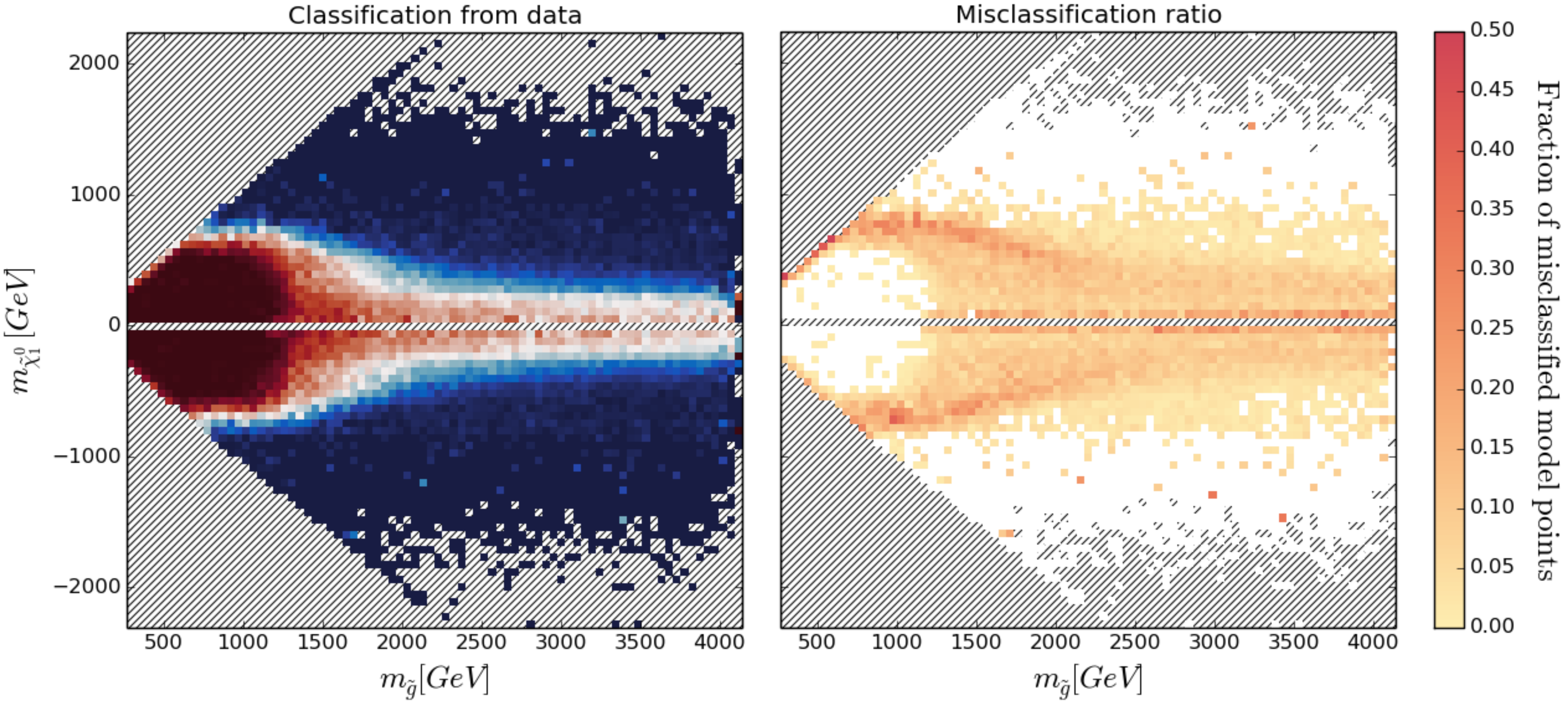
92.7% accuracy @ 13 TeV



Performance gluino vs neutralino1

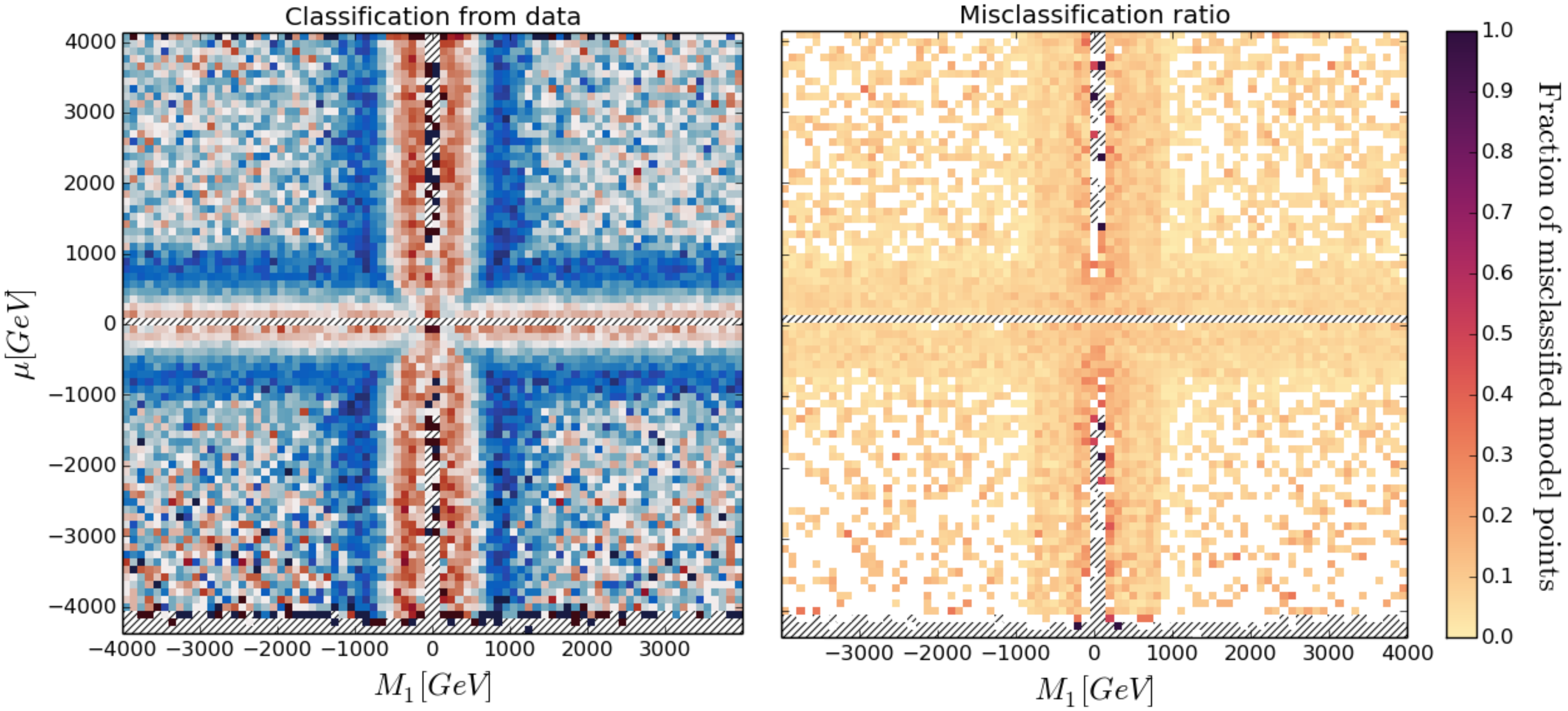
93.2% accuracy @ 8TeV

92.7% accuracy @ 13 TeV



Performance M1 vs mu

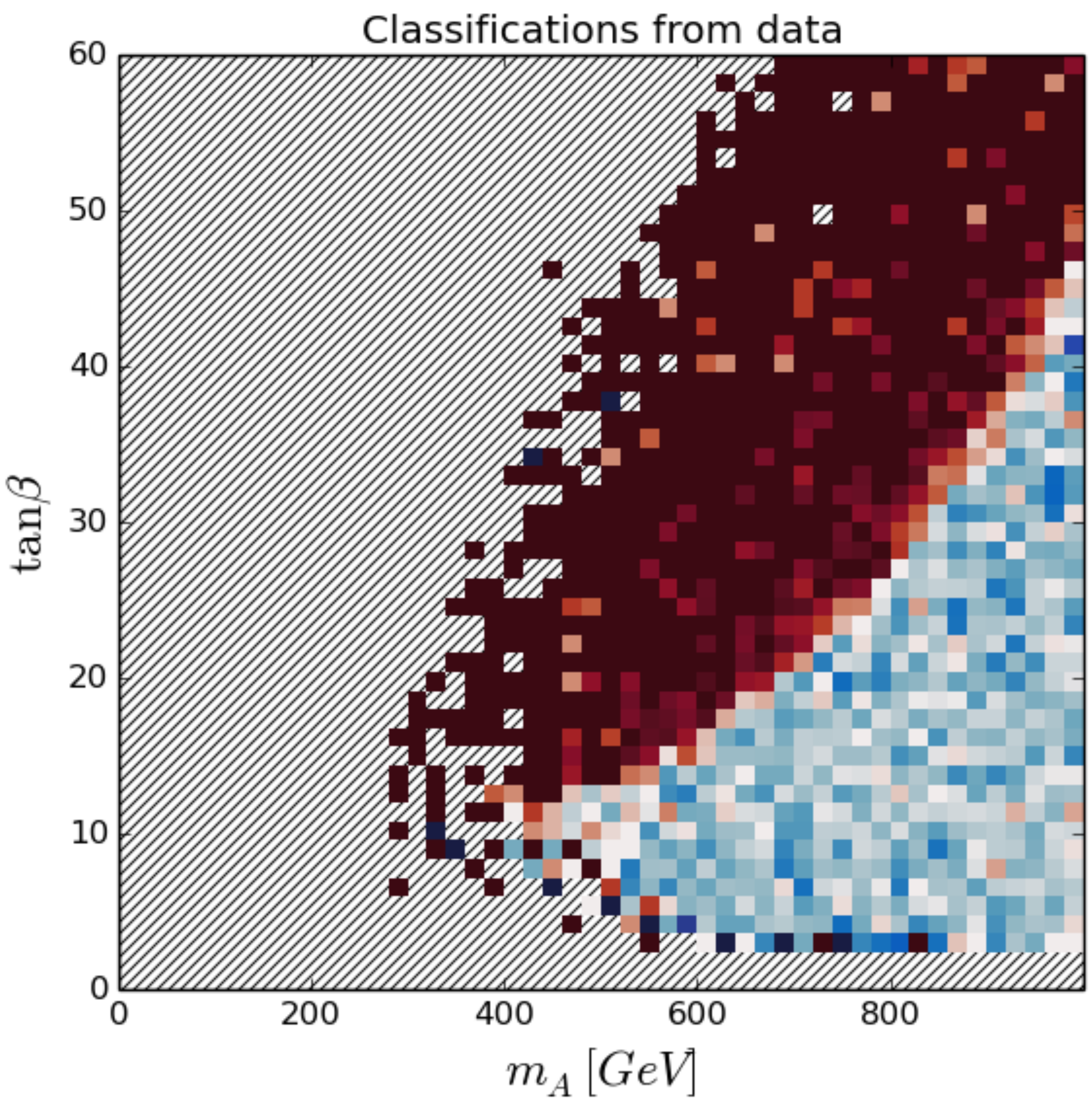
93.2% accuracy @ 8TeV 92.7% accuracy @ 13 TeV

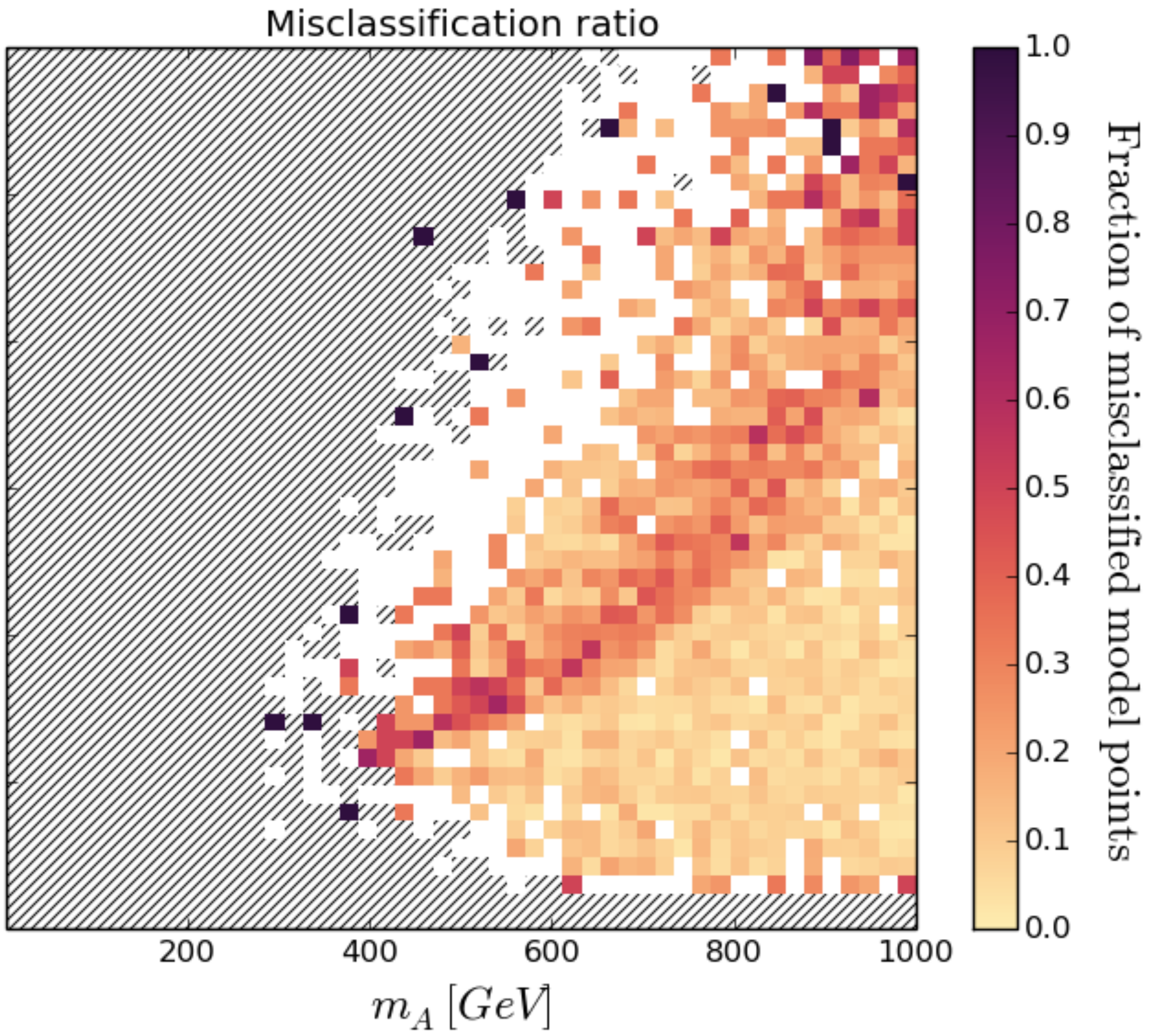


Performance mA vs tan(beta)

93.2% accuracy @ 8TeV

92.7% accuracy @ 13 TeV





Confidence

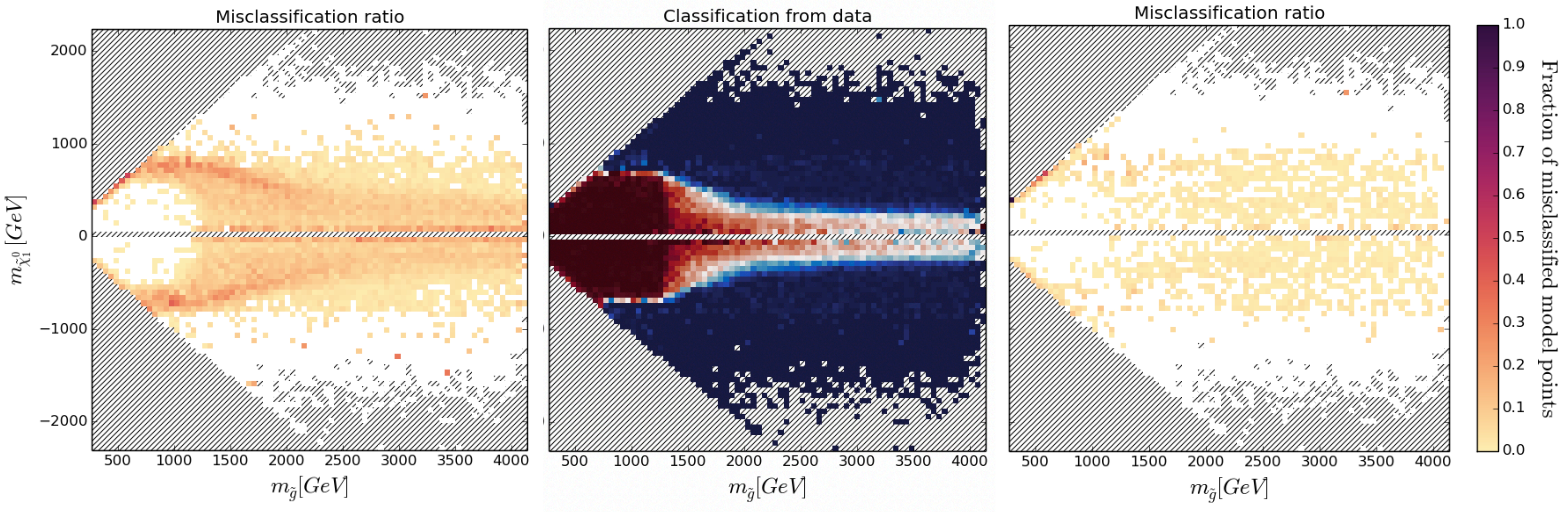
- Predicted exclusion is continuous (value between 0 and 1), can be associated with degree of confidence on prediction

- Allows for requiring minimum degree of confidence

Confidence (>95%) gluino vs neutralino1

99.1% accuracy on 70.6% of total data @ 8TeV

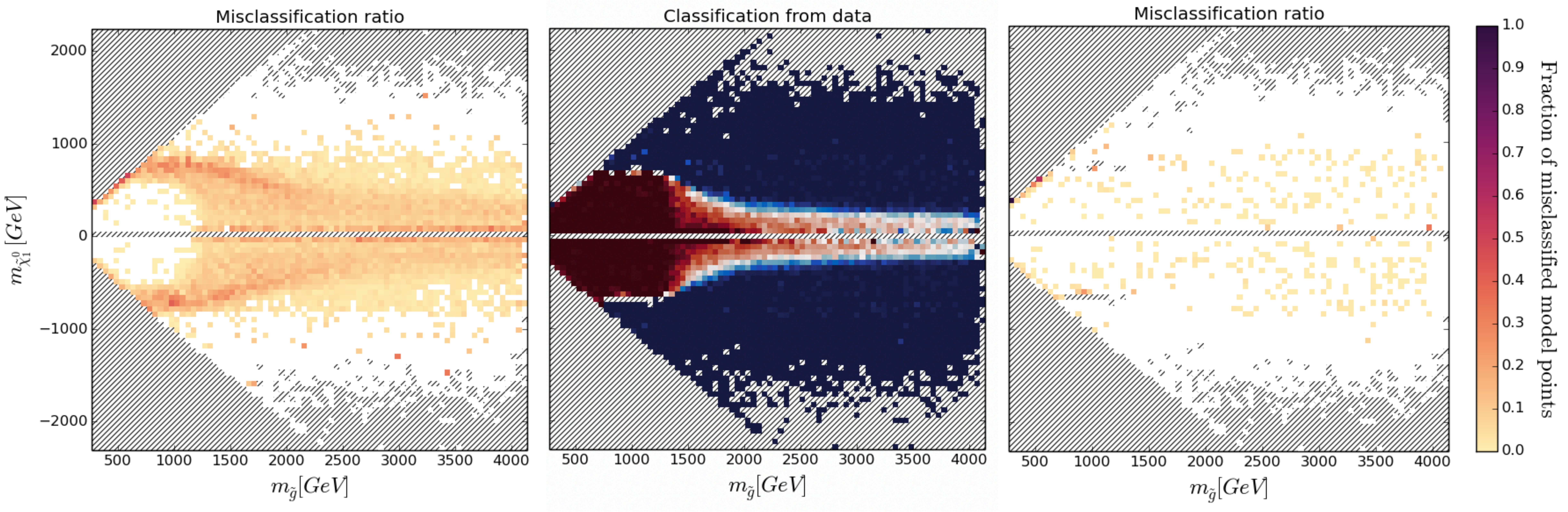
99.0% accuracy on 68.0% of total data @ 13 TeV



Confidence (>99%) gluino vs neutralino1

99.7% accuracy on 51.6% of total data @ 8TeV

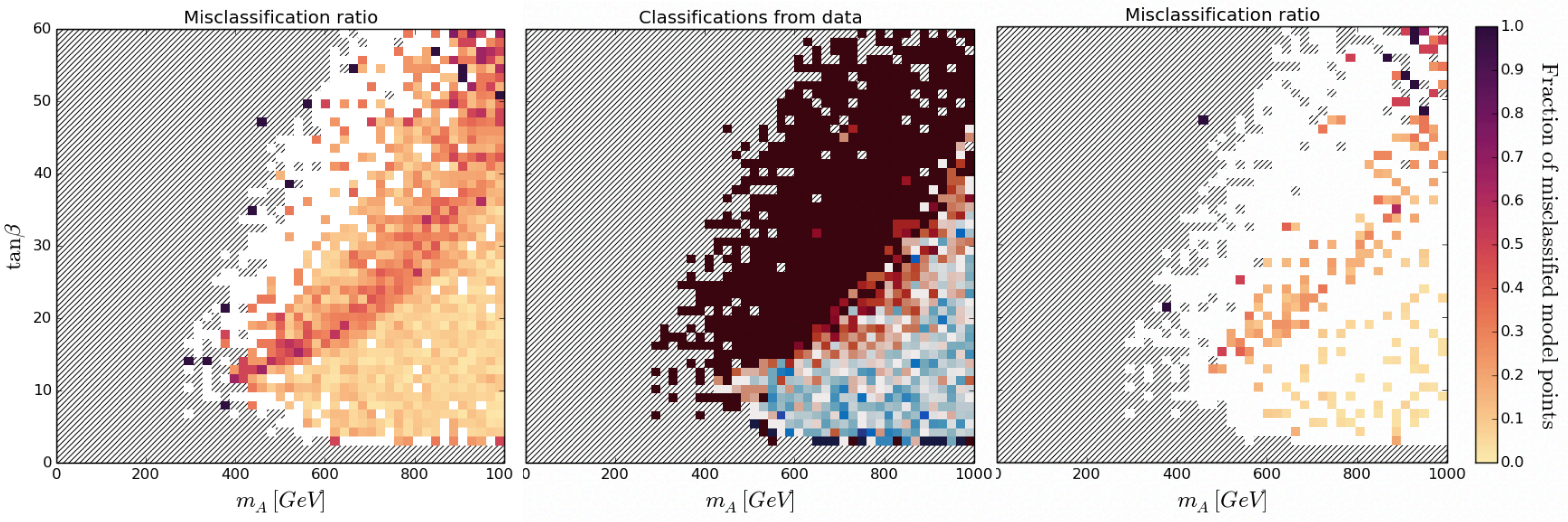
99.7% accuracy on 47.6% of total data @ 13 TeV



Confidence (>95%) mA vs tan(beta)

99.1% accuracy on 70.6% of total data @ 8TeV

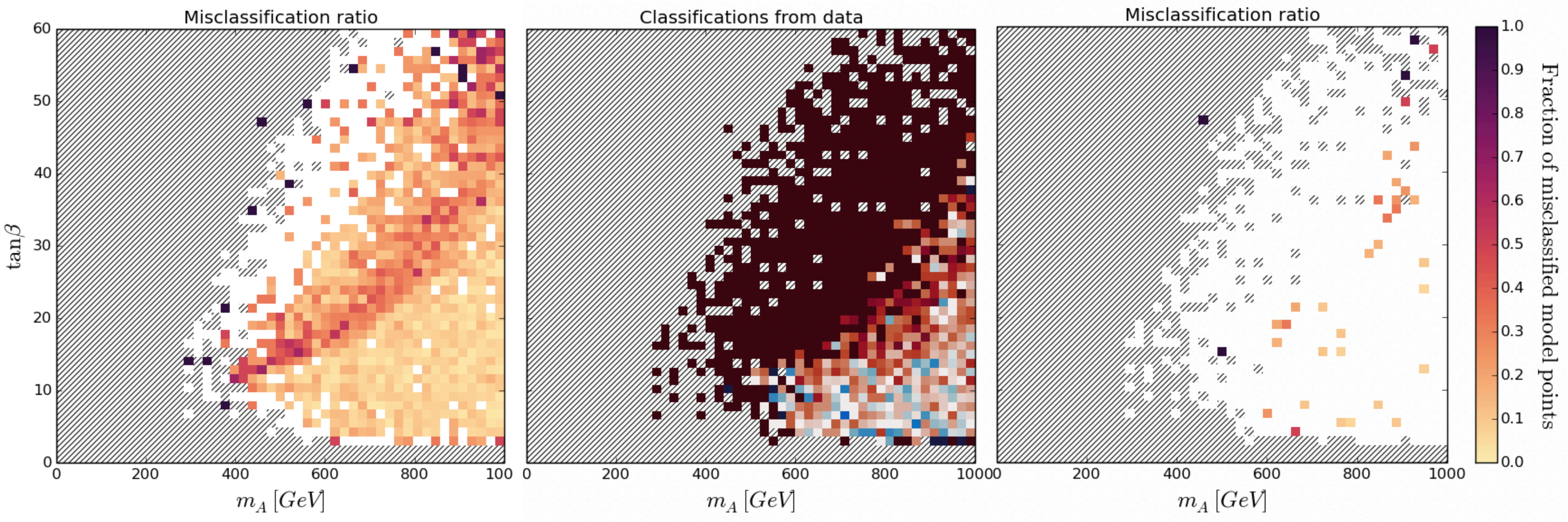
99.0% accuracy on 68.0% of total data @ 13 TeV



Confidence (>99%) mA vs tan(beta)

99.7% accuracy on 51.6% of total data @ 8TeV

99.7% accuracy on 47.6% of total data @ 13 TeV



SUSY-AI

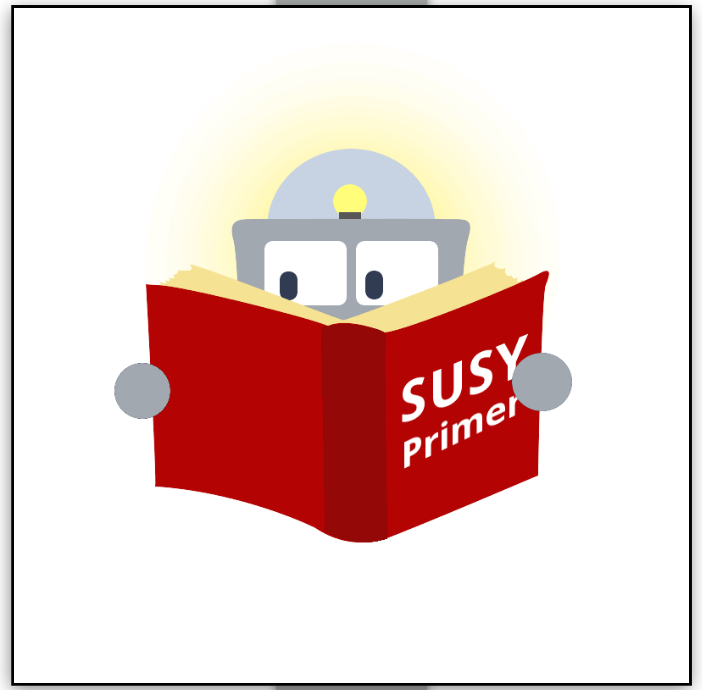
- Algorithms (both 8TeV and 13TeV) are publicly available at http://susyai.hepforge.org

```python
from susyai import susyai
import numpy as np

sa = susyai("susyai_classifier_python_v3.pkl")
data = np.array([[30, 4.0276e2, 7.3196e2, 2.1862e3, 1.0,
          4.0713e3, 4.4890e3, 4.4752e3, 4.4743e3, 2.8806e3,
          3.7855e3, 1.3240e3, 2.9076e3, 4.2226e3, 4.2056e3,
          3.4290e3, 3.8608e3, -4.3154e3, -8.1538e3, -7.3680e3]])
clas, pred, cert = sa.predict(data)
```

- Up to 5,000 model point predictions per second / CPU

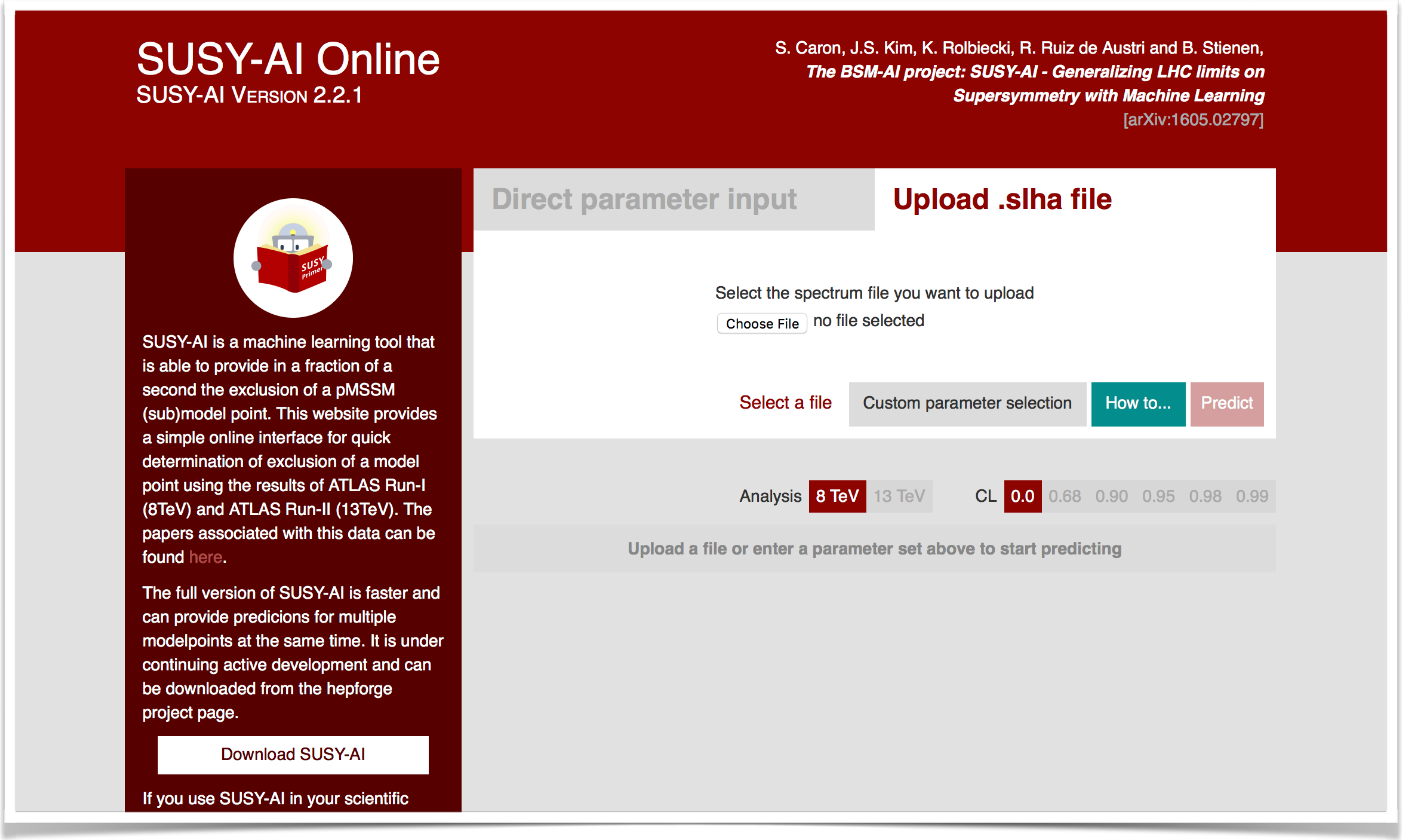
Modelpoint

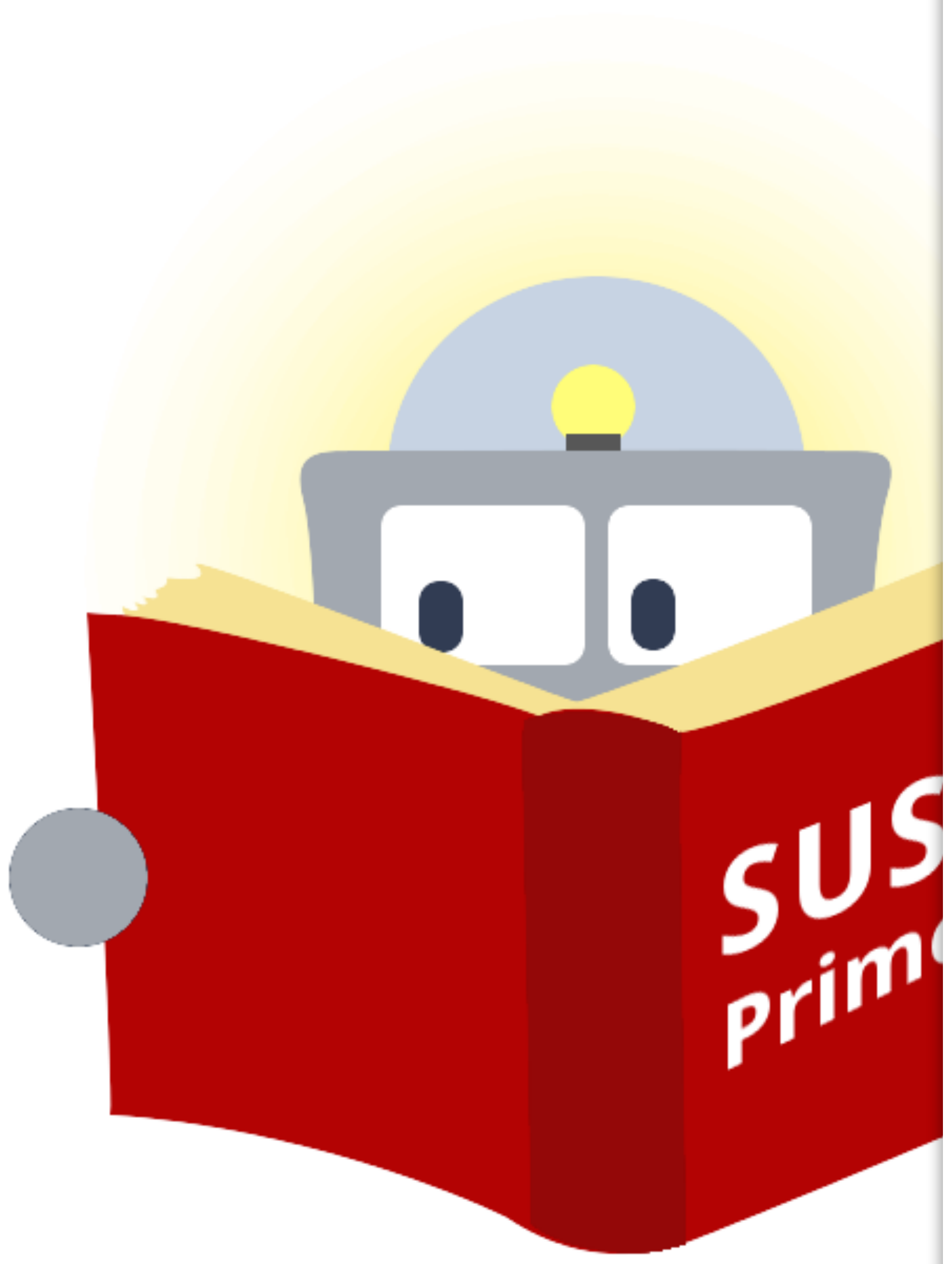


excluded / allowed

SUSY-AI online

http://www.susy-ai.org/





Conclusion

[arXiv:1605.02797]

- We created a Machine Learning algorithm that can predict model point exclusion in a fraction of a second

- It works within the general pMSSM, but method is not limited to this parameter space (let me know if you have data!)

- Algorithm can be stored: method can be used to communicate multivariate results

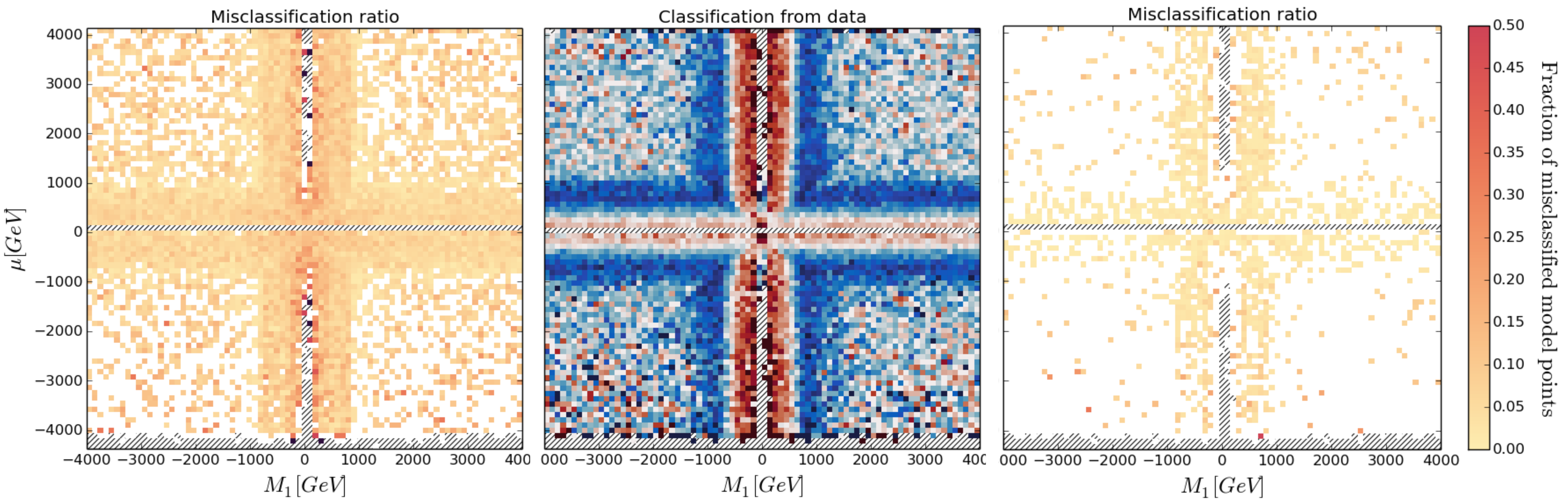
- Website is online and algorithm is publicly available (you can start applying LHC limits to your data right away!)

Back-up slides

Confidence (>95%) M1 vs mu

99.1% accuracy on 70.6% of total data @ 8TeV

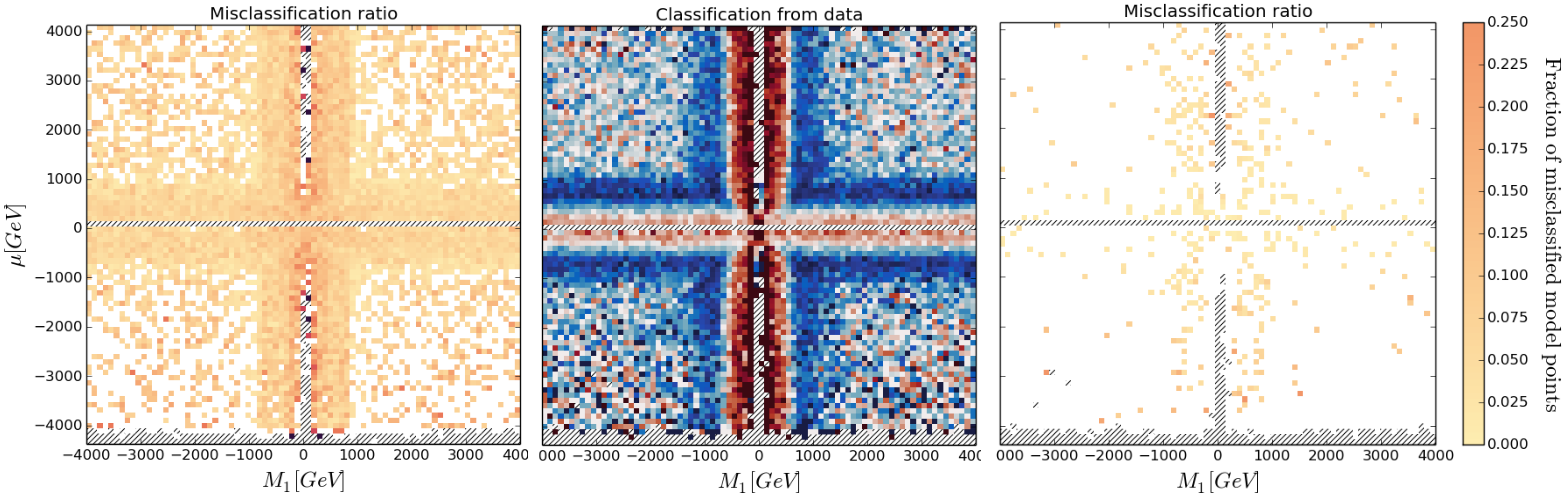
99.0% accuracy on 68.0% of total data @ 13 TeV



Confidence (>99%) M1 vs mu

99.7% accuracy on 51.6% of total data @ 8TeV

99.7% accuracy on 47.6% of total data @ 13 TeV



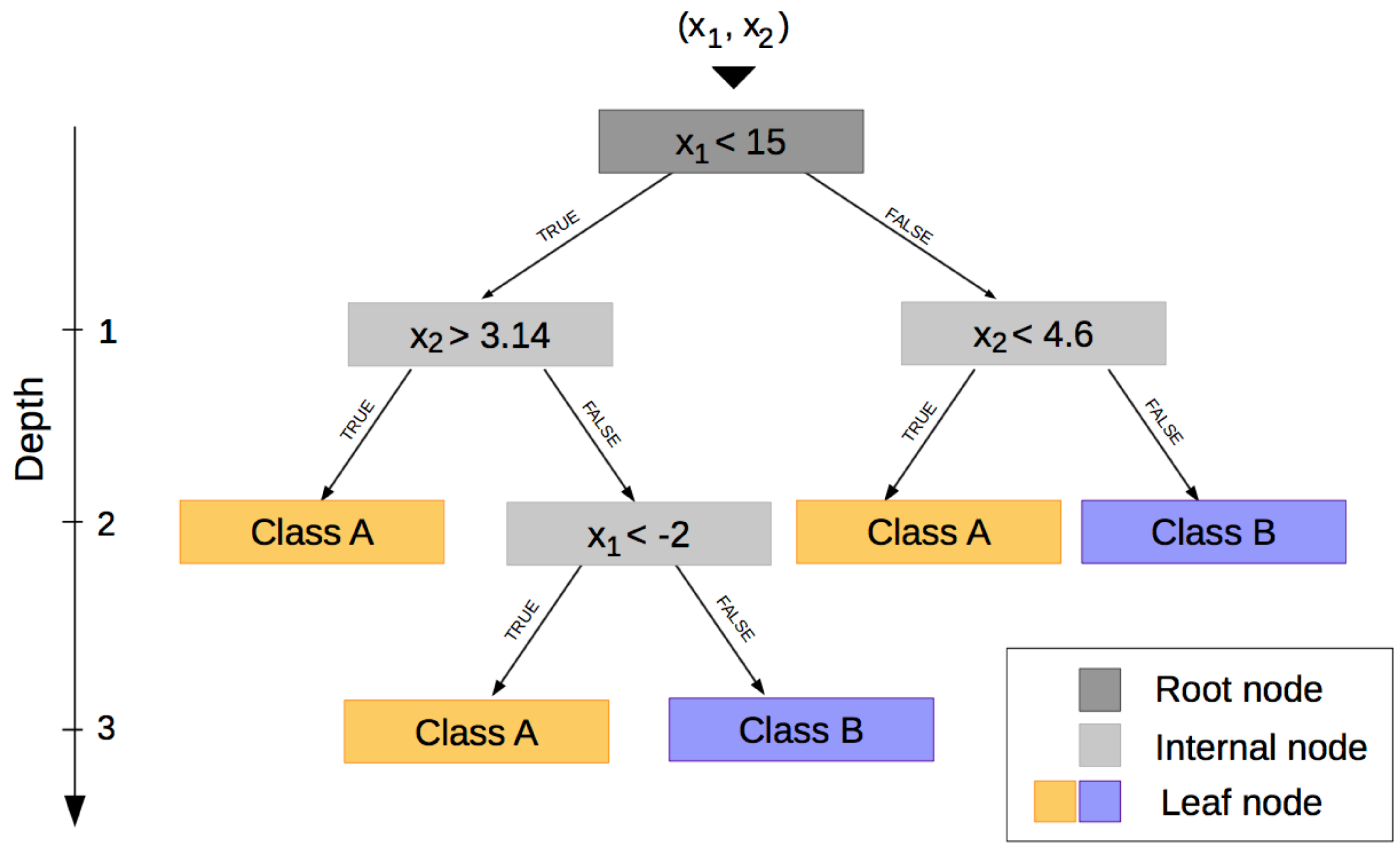
Parameters

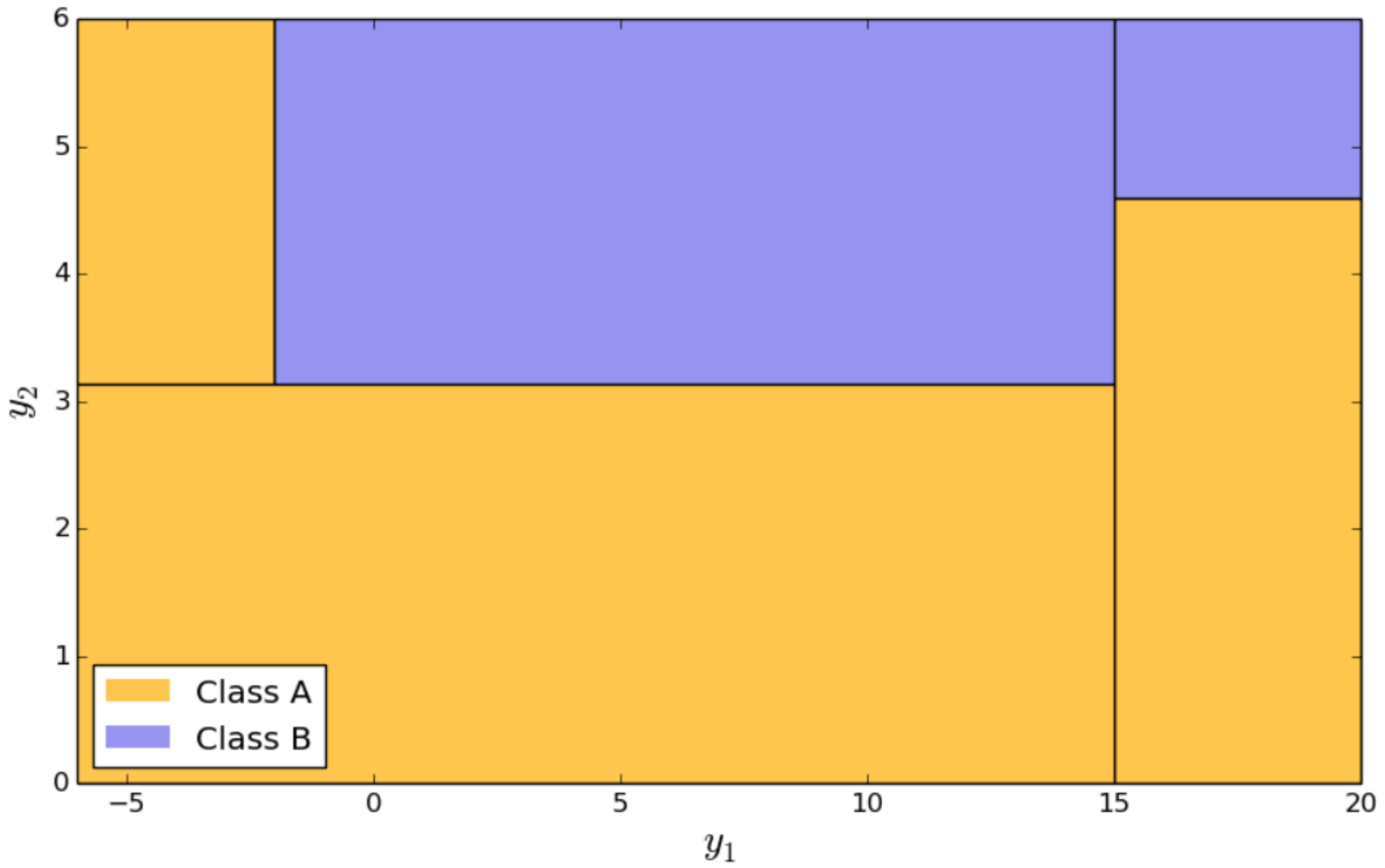
Parameter	Description	Scanned range
$m_{\tilde{L}_1}$	$1^{\mathrm{st}}/2^{\mathrm{nd}}$ gen. $SU(2)$ doublet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{E}_1}$	$1^{\mathrm{st}}/2^{\mathrm{nd}}$ gen. $SU(2)$ singlet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{L}_3}$	3^{rd} gen. $SU(2)$ doublet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{E}_3}$	3^{rd} gen. $SU(2)$ singlet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{Q}_1}$	$1^{\mathrm{st}}/2^{\mathrm{nd}}$ gen. $SU(2)$ doublet soft breaking squark mass	[200 GeV, 4 TeV]
$m_{\tilde{U}_1}$	$1^{\mathrm{st}}/2^{\mathrm{nd}}$ gen. $SU(2)$ singlet soft breaking squark mass	[200 GeV, 4 TeV]
$m_{\tilde{D}_1}$	$1^{\mathrm{st}}/2^{\mathrm{nd}}$ gen. $SU(2)$ singlet soft breaking squark mass	[200 GeV, 4 TeV]
$m_{\tilde{Q}_3}$	3^{rd} gen. $SU(2)$ doublet soft breaking squark mass	[100 GeV, 4 TeV]
$m_{\tilde{U}_3}$	3^{rd} gen. $SU(2)$ singlet soft breaking squark mass	[100 GeV, 4 TeV]
$m_{\tilde{D}_3}$	3^{rd} gen. $SU(2)$ singlet soft breaking squark mass	[100 GeV, 4 TeV]
A_t	Stop trilinear coupling	[−8 TeV, 8 TeV]
A_b	Sbottom trilinear coupling	[−4 TeV, 4 TeV]
A_τ	Stau trilinear coupling	[−4 TeV, 4 TeV]
$\|\mu\|$	Higgsino mass parameter	[80 GeV, 4 TeV]
$\|M_1\|$	Bino mass parameter	[0 TeV, 4 TeV]
$\|M_2\|$	Wino mass parameter	[70 GeV, 4 TeV]
M_3	Gluino mass parameter	[200 GeV, 4 TeV]
M_A	Pseudoscalar Higgs mass	[100 GeV, 4 TeV]
$\tan\beta$	Ratio of vacuum expectation values	[1, 60]

Analyses run by ATLAS

Reference	Final State	Category
[30]	0 lepton + 2–6 jets + $\not{E}_T$	Inclusive
[31]	0 lepton + 7–10 jets + $\not{E}_T$	
[32]	1 lepton + jets + $\not{E}_T$	
[33]	$\tau(\tau/\ell)$ + jets + $\not{E}_T$	
[34]	SS/3 lepton + jets + $\not{E}_T$	
[35]	b-jets + 0/1 lepton + $\not{E}_T$	
[36]	monojet	
[37]	0 lepton stop search	Third generation squarks
[38]	1 lepton stop search	
[39]	2 lepton stop search	
[40]	monojet search	
[41]	stop search with Z in final state	
[42]	2b-jets sbottom search	
[4]	asymmetric stop search	
[43]	1 lepton plus Higgs final state	Electroweak
[44]	dilepton final state	
[45]	2τ final state	
[46]	trilepton final state	
[47]	four-lepton final state	
[48]	disappearing track	
[49, 50]	Long-lived particle search	Other
[51]	$H/A \to \tau\tau$ search	

Decision trees

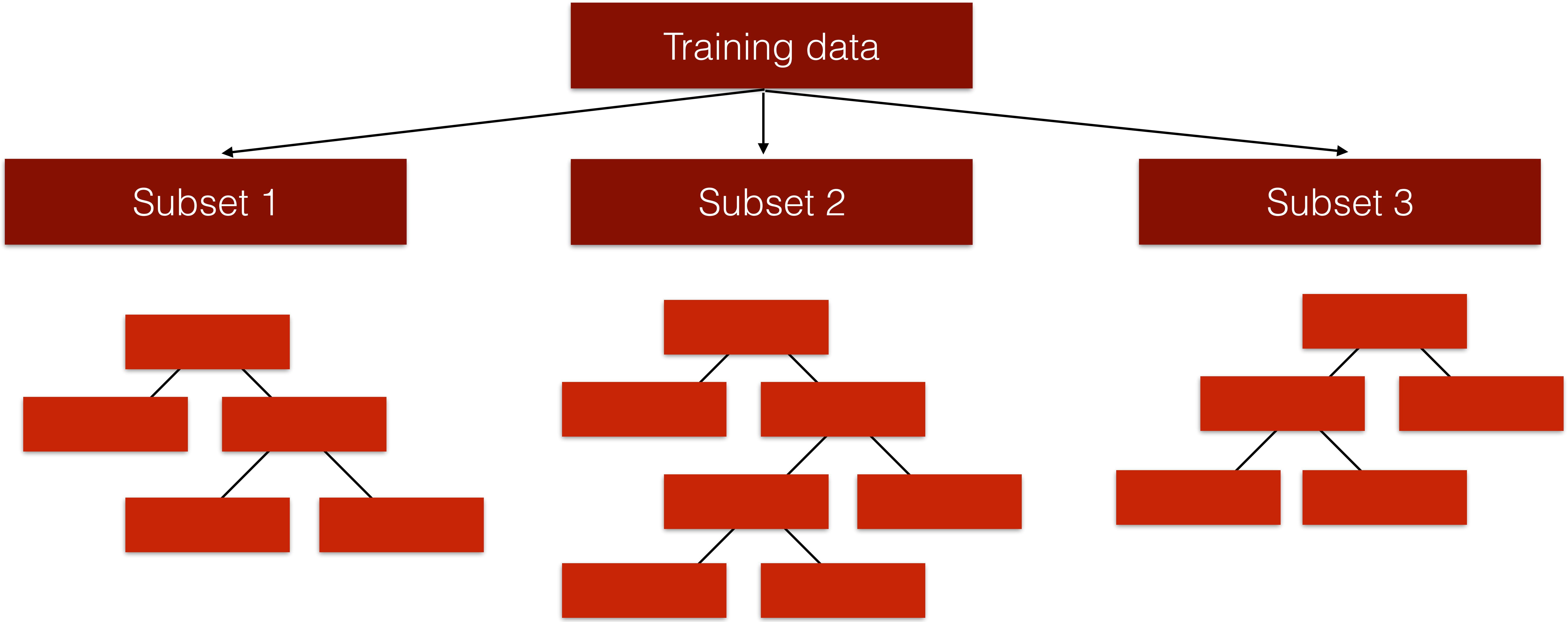




Random Forest (1/2)

- Combination of multiple decision trees, prediction by majority vote

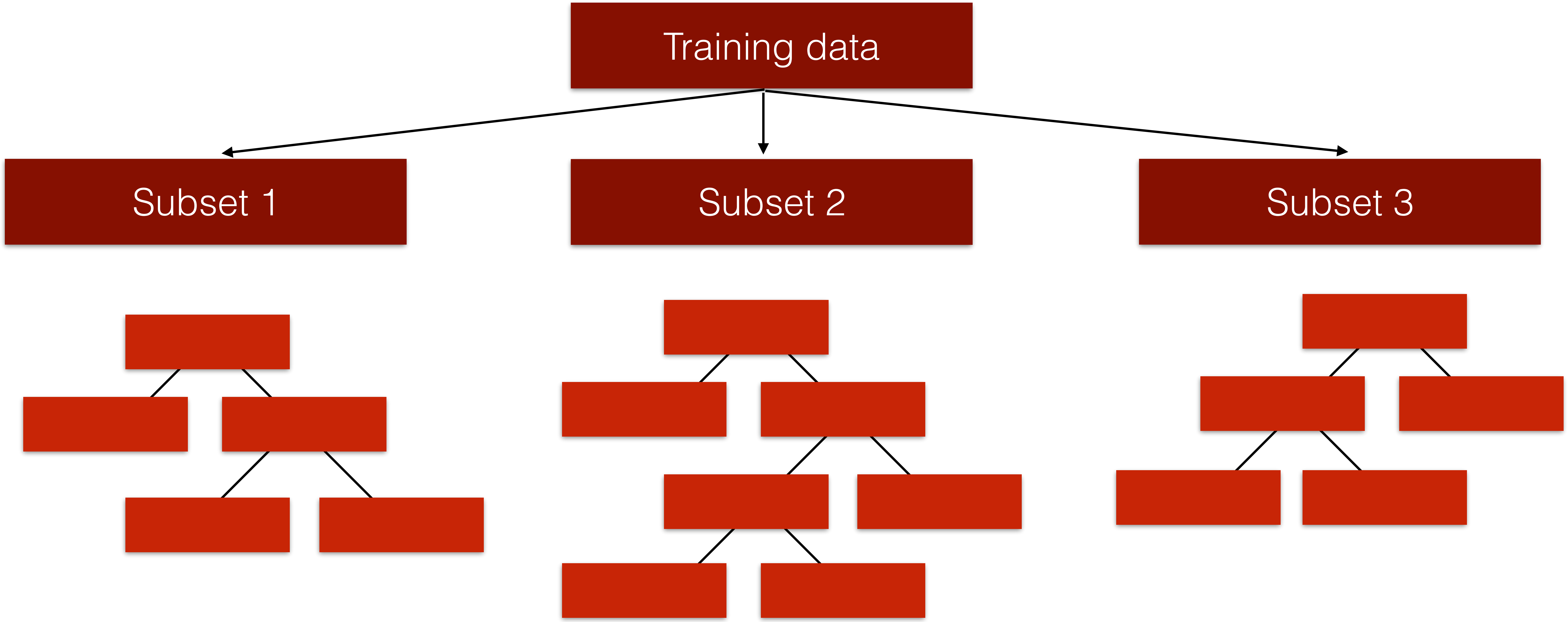
- Introducing the randomness in the forest: trees are constructed with *bagging* (each tree trained on unique subset of training data)



Random Forest (2/2)

- Subsets are of the same size as training data set and data points are selected with replacement —> same datapoint can be selected multiple times
  ~63.2% of model points in subset are unique

- Moreover, only subset of parameters is considered at each node to split on



Out-of-bag estimation

- Only ~63.2% of training data is used in training of a single tree

- Use remaining 37.8% for independent testing

- This can be done for every single tree in the forest

- Combined output is independent prediction by forest on its training data —> useful for testing purposes (no train:test split needed!)

Out-of-bag vs train:test split

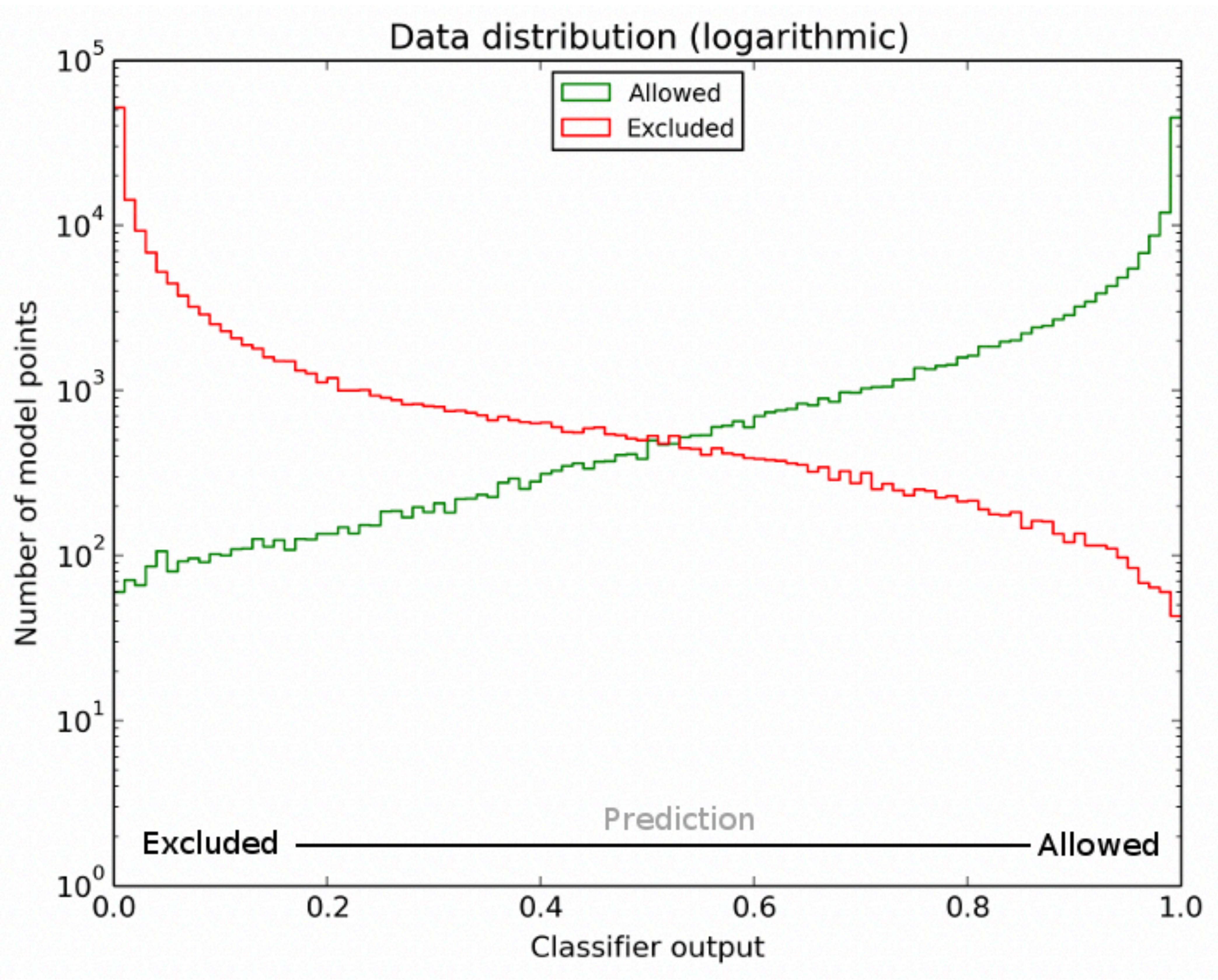
Out-of-bag

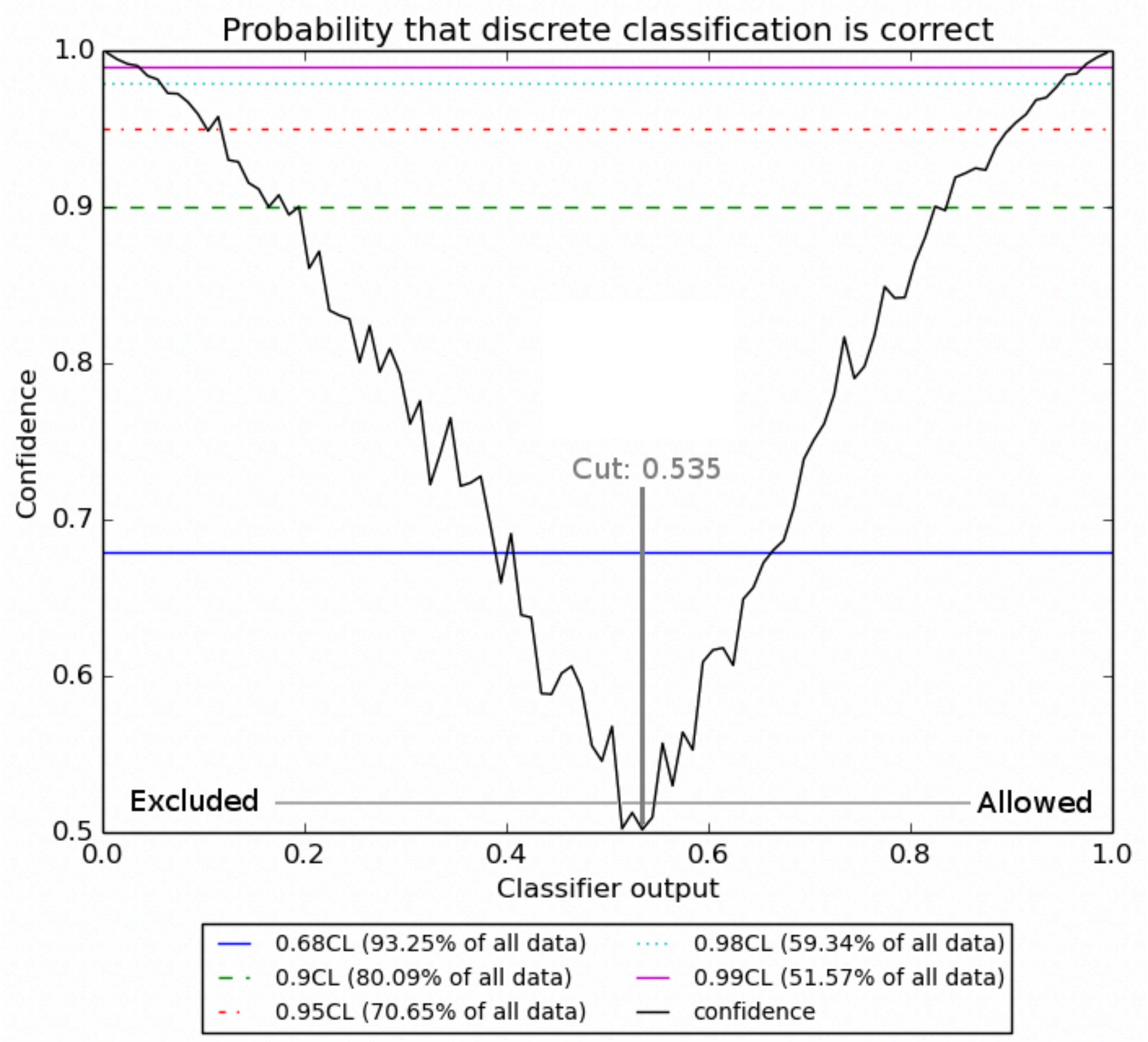
CL	#	# / total	Accuracy	Precision	Sensitivity	NPV	Specificity
0.0	310 324	1.0000	0.93226	0.93951	0.94665	0.92152	0.91133
0.68	289 371	0.93248	0.95735	0.96072	0.96835	0.95222	0.94094
0.95	219 233	0.70646	0.99094	0.99092	0.99426	0.99096	0.98573
0.98	184 230	0.59367	0.99543	0.99573	0.99672	0.99496	0.99346
0.99	160 034	0.51570	0.99708	0.99747	0.99764	0.99649	0.99624

Dataset splitting train:test = 75:25

CL	#	# / total	Accuracy	Precision	Sensitivity	NPV	Specificity
0.0	77 581	1.0000	0.92271	0.91653	0.93049	0.92912	0.91491
0.68	70 375	0.90712	0.9545	0.95516	0.95302	0.95386	0.95595
0.95	48 900	0.63031	0.99022	0.99047	0.9893	0.99	0.99109
0.98	39 815	0.51321	0.99485	0.99559	0.99353	0.99419	0.99604
0.99	34 004	0.43830	0.99644	0.99685	0.99554	0.99608	0.99724

Construction of confidence measure





Random Forest configuration

Optimal configuration was found via a grid search

- Number of trees
  900
- Maximum features considered each split
  12 (out of a total of 19)
- Maximum depth of each individual tree
  30

Random Forest vs Boosted Decision Trees (1/2)

- Both are sets of decisions trees, but constructed in different ways: bagging vs boosting respectively
  - Bagging: introduce randomness in training of the trees and average over them.
  - Boosting: train each tree iteratively to do better on the mistakes of the previous trees (increase weight of misclassified points by previous tree)
- Both bagging and boosting are well understood methods to reduce overtraining.

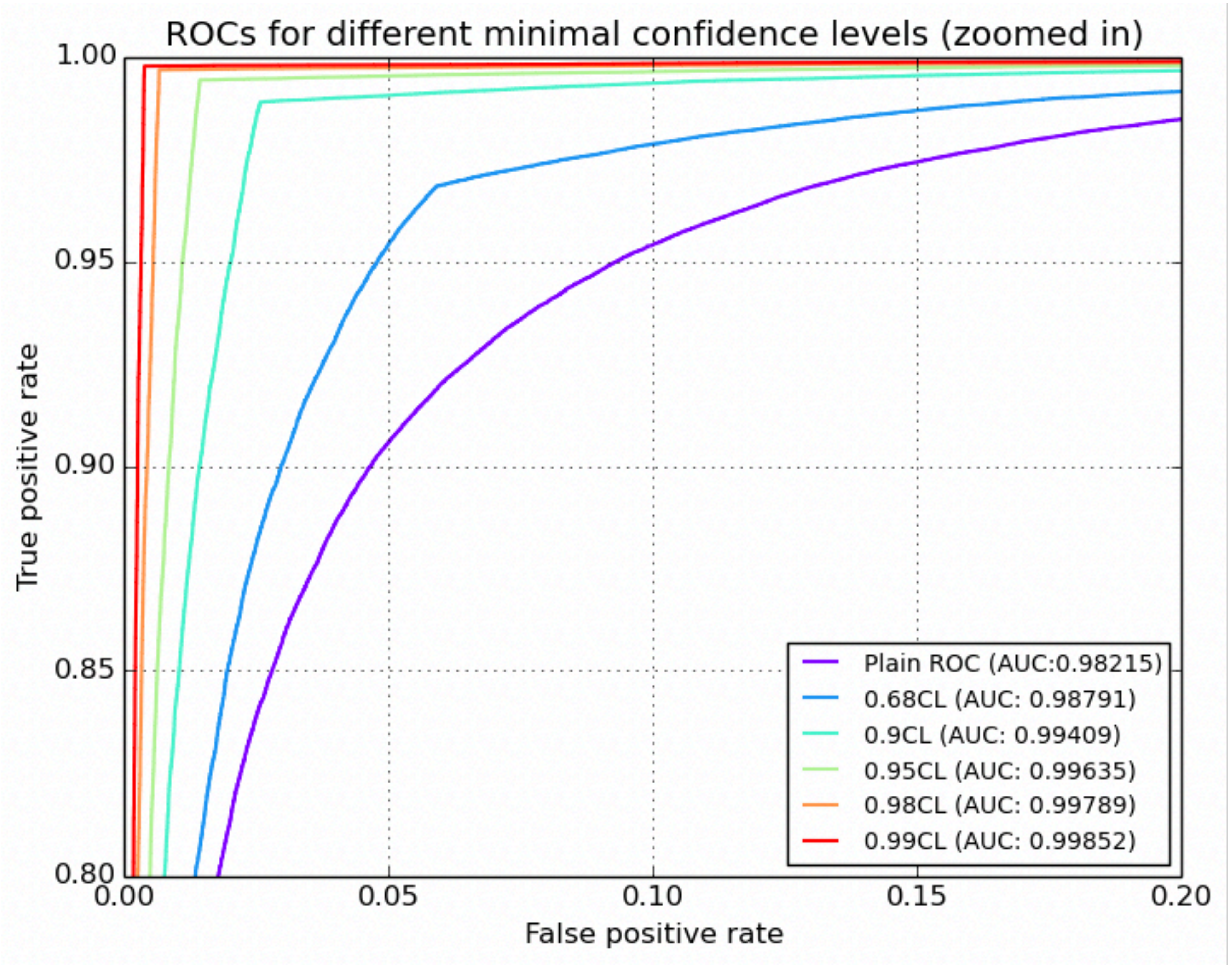
Random Forest vs Boosted Decision Trees (2/2)

- Bagging is less sensitive to outliers (it tries to correctly classify all data points) and its output is more closely linked to prediction confidence

- Boosting reduces in theory both bias and variance, but does tend to overfit sometimes. It uses shorter trees and is faster in training and use.

More pragmatic approach:
Random Forest not in TMVA (though hard to find), Boosted Decision Trees not in scikit-learn

ROC curve



ROCs for different minimal confidence levels (zoomed in)
1.00
0.95
0.90
0.85
0.80
True positive rate
0.00
0.05
0.10
0.15
0.20
False positive rate
Plain ROC (AUC:0.98215)
0.68CL (AUC: 0.98791)
0.9CL (AUC: 0.99409)
0.95CL (AUC: 0.99635)
0.98CL (AUC: 0.99789)
0.99CL (AUC: 0.99852)

Comparison to model for human

- 20 individual decision trees with maximum depth of 5 (=21 cuts in parameter space)

- Markers are placed at value for cut with the highest accuracy