Learning (from) High-dimensional Models

Jisk Attema
Tom Heskes
Roberto Ruiz de Austri

Sascha Caron
Sydney Otten
Jong Soo Kim

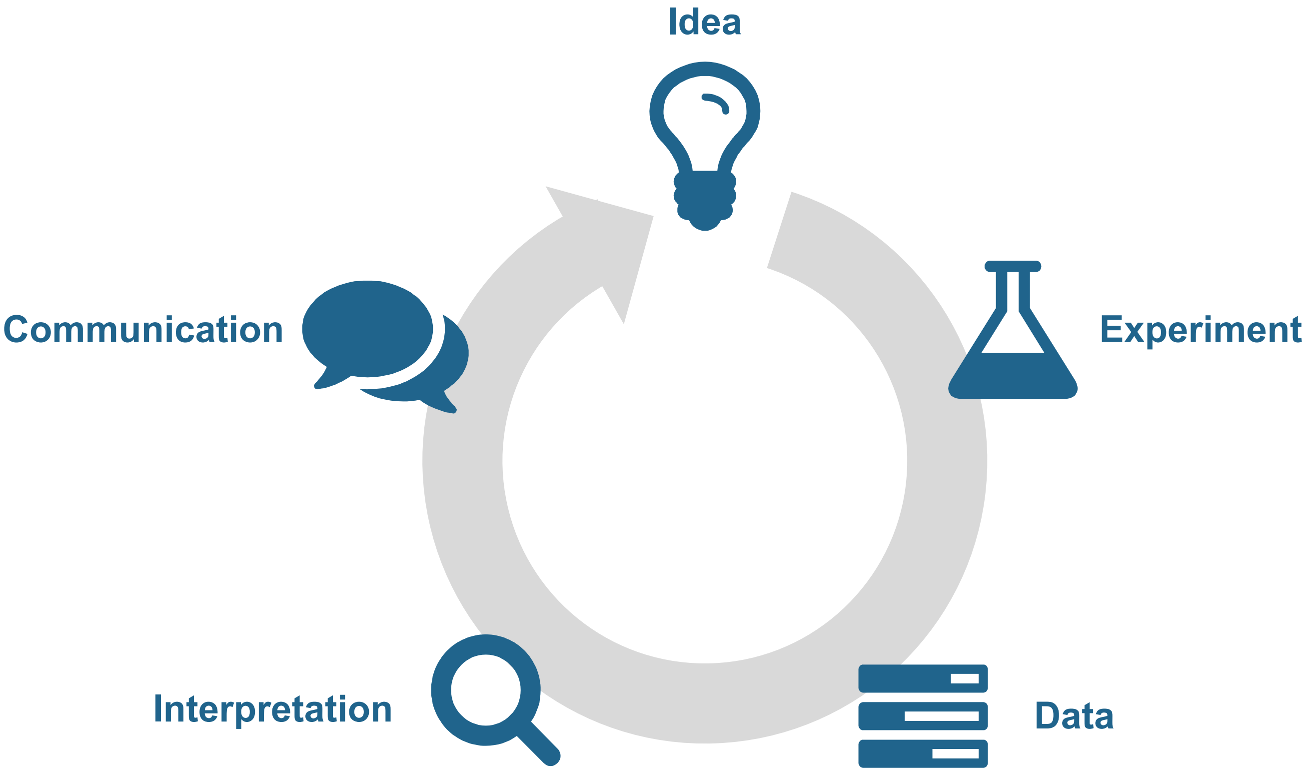
Faruk Diblen
Krzysztof Rolbiecki
<u>Bob Stienen</u>

netherlands eScience center

Radboud University

How do we do (particle) physics?

The Circle of Physics

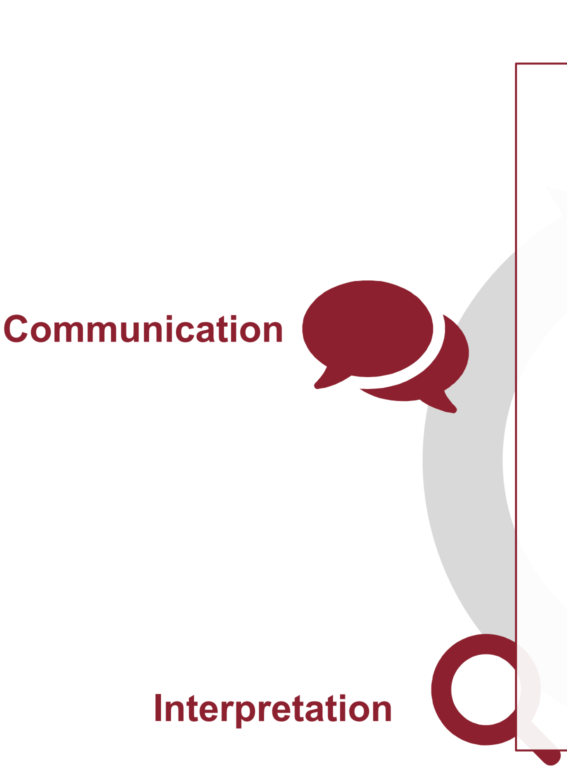


The Circle of Physics

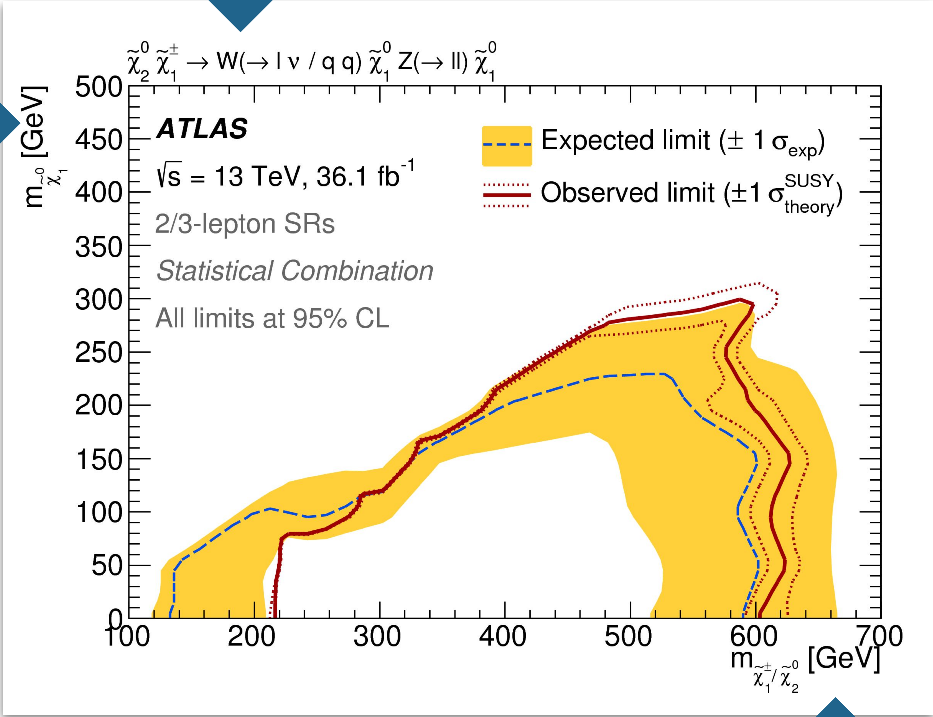
- Inherently model dependent
  → different model = different interpretation

- Interpretation of the results in the context of a single model point is computationally very expensive
  → Simplified models are often used, but

The Circle of Physics



- Images in papers are inherently 2-dimensional
  → displaying more than 4 dimensions in a plot is difficult
- Simplified models are often used, but at the cost of information loss
- Raw data can be published (e.g. model points + evaluations)
  → Individual results are not extremely useful

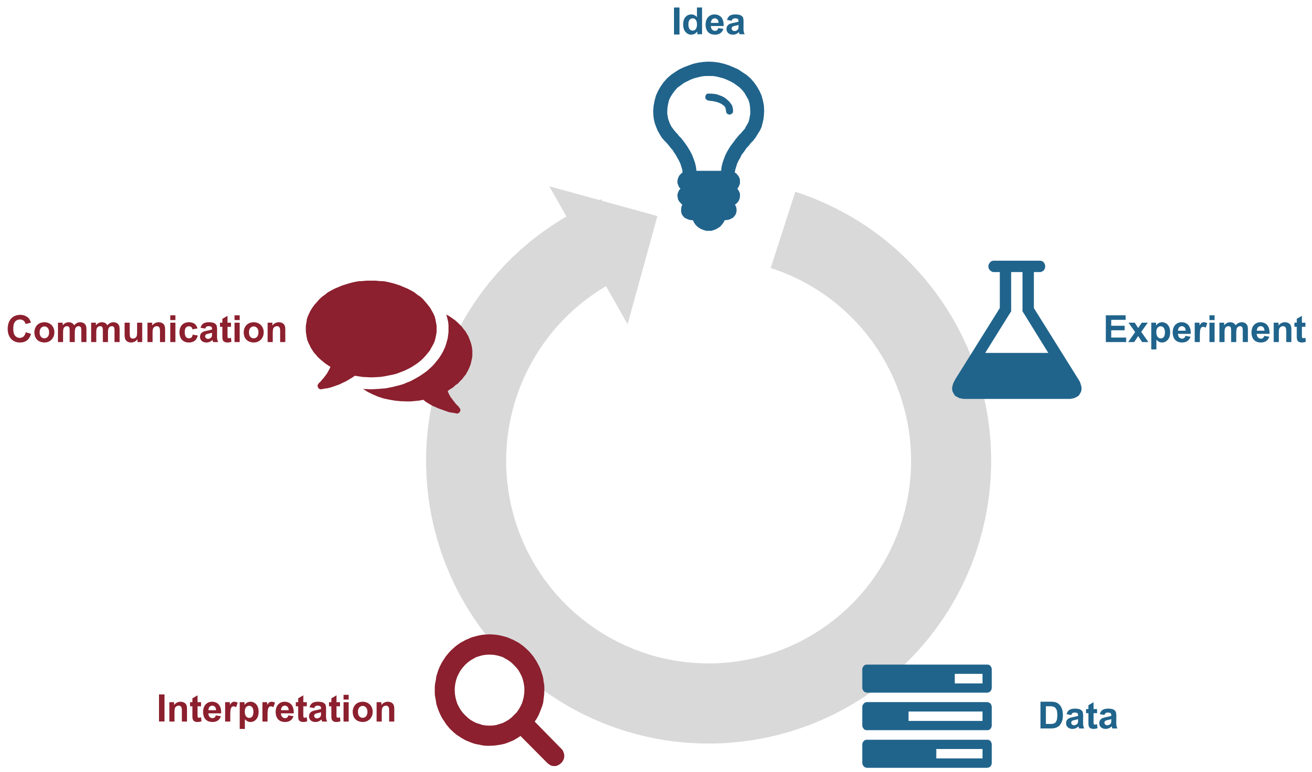


What if...

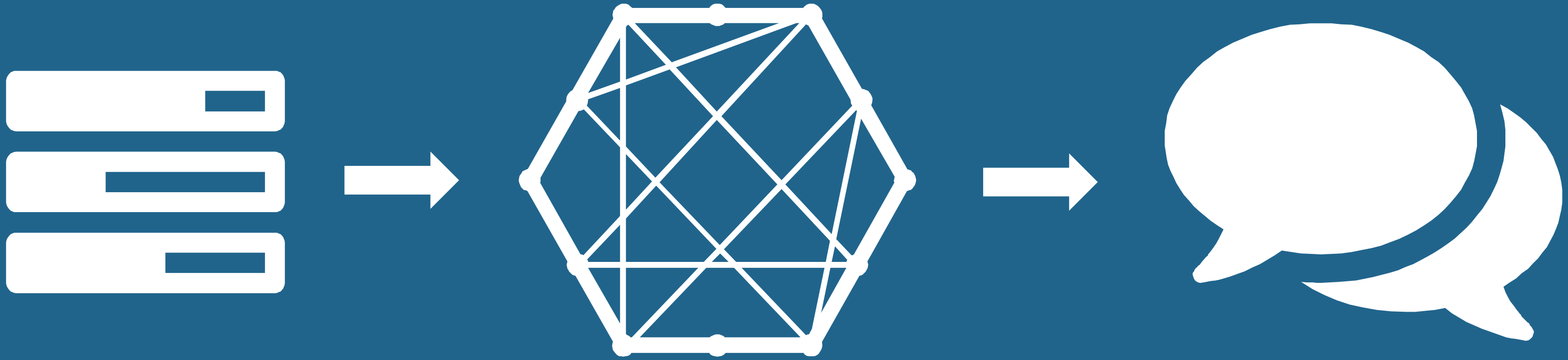
- i don't have a 100% BR to the specified final state?
- i want to know the exclusion in another projection?
- i have the other free parameters set differently?

Core of the problem:
Plotting N>2 dimensions is hard

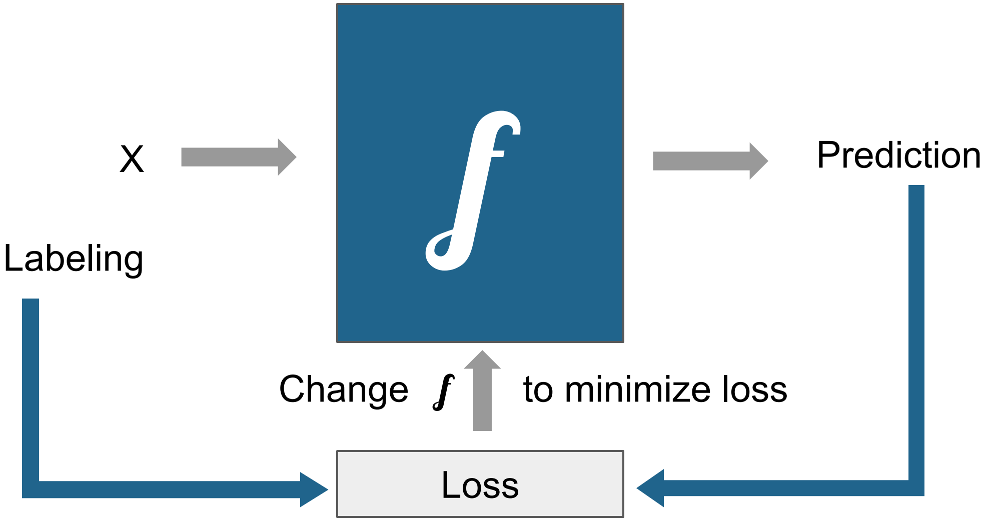
The Circle of Physics



How to manage our information to retain most of it?



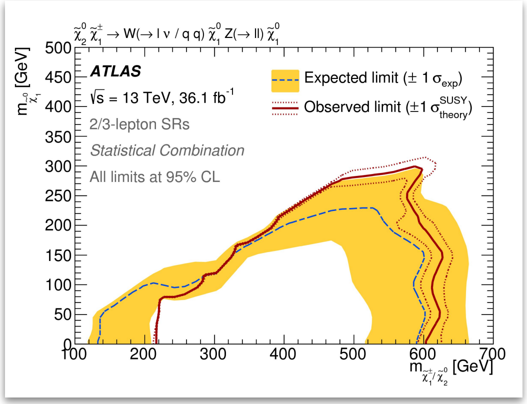
Machine Learning as a solution



Machine Learning as a solution

Example

m_chargino
m_neutralino

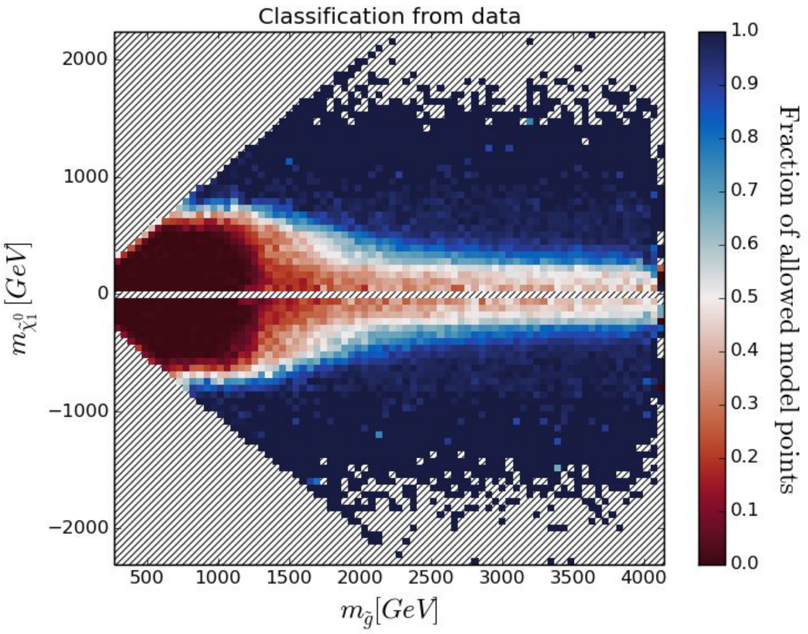


Encodes our model and entire analysis workflow

But... can be N>2...

SUSY-AI as proof-of-principle

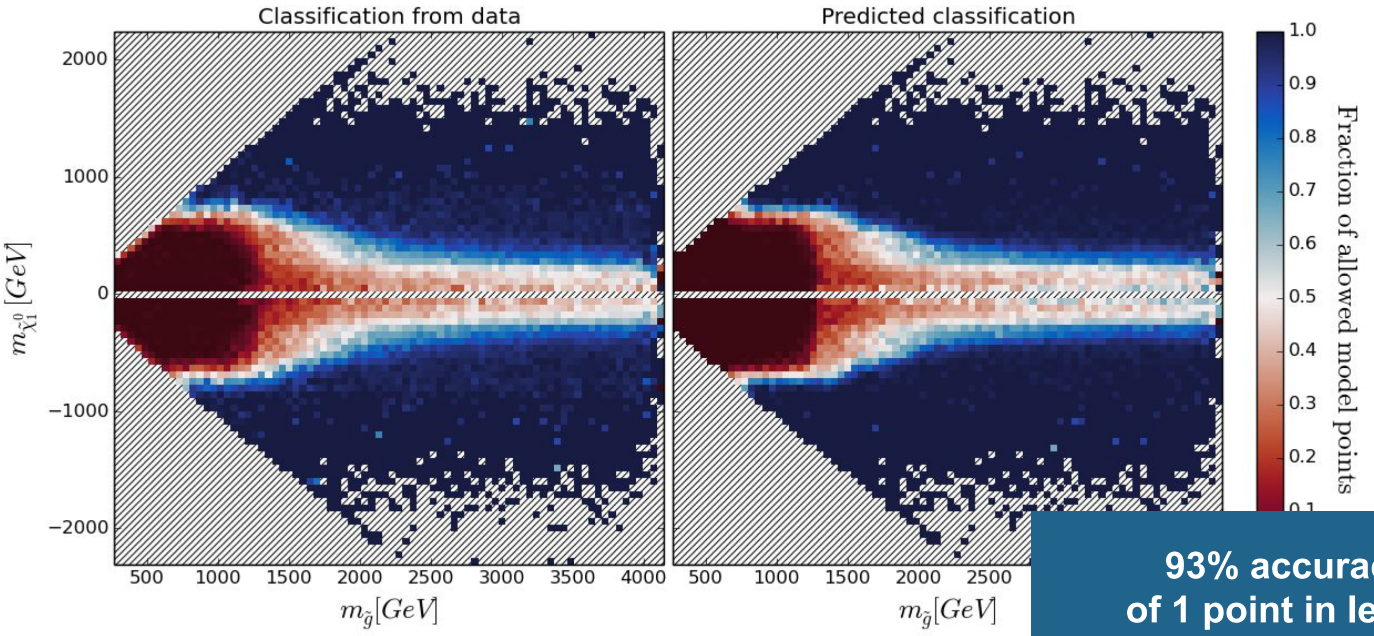
DOI: 10.1140/epjc/s10052-017-4814-9



- pMSSM19
- 300,000 training points
  10.1007/JHEP10(2015)134
- Exclusion determined by 22 different analyses
- RandomForest (for the *connaisseurs*)

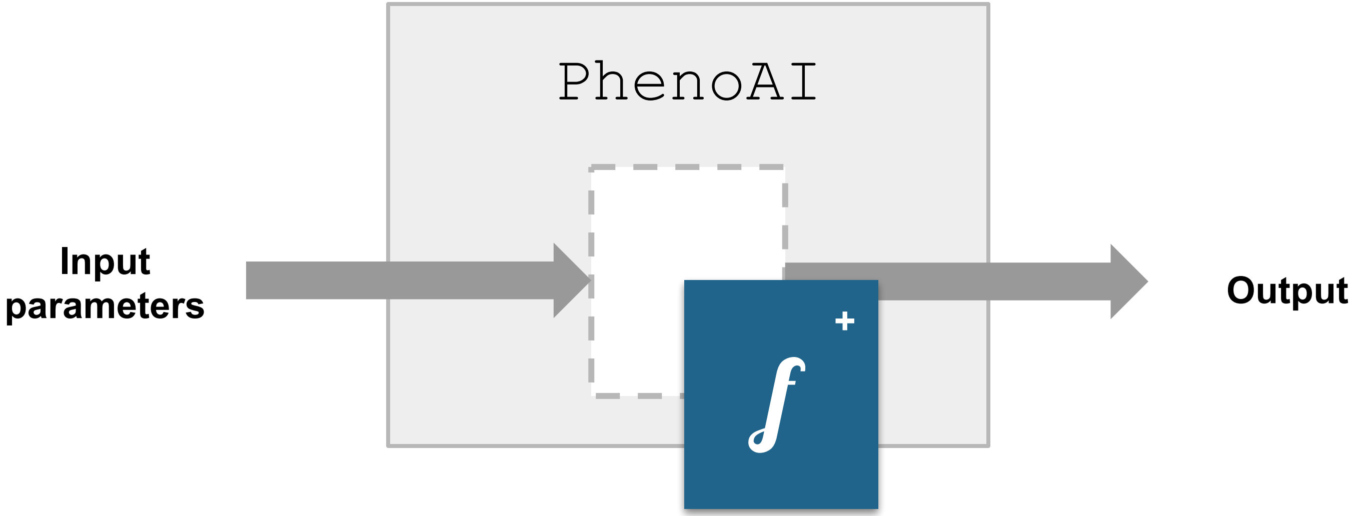
SUSY-AI as proof-of-principle

DOI: 10.1140/epjc/s10052-017-4814-9



93% accuracy at a rate of 1 point in less than a ms in a full 19-dimensional model

PhenoAI as natural evolution



Machine Learning is abstracted away:
anyone with Python knowledge can use the trained models

Communication of high-dimensional results becomes possible:
publish a trained algorithm

PhenoAInalyses

- Trained algorithms (**AInalyses**) still need to be made. You can do this yourself, or…

- … download one from the AInalysis library on the PhenoAI website

- Currently working on AInalyses for:
  - Cross Sections
  - Electroweakino
  - Likelihoods from Gambit

Supported ML libraries

All estimators and models created with Keras/tensorflow and scikit-learn are supported within PhenoAI. We are in the process of adding support for ROOT TMVA models as well.

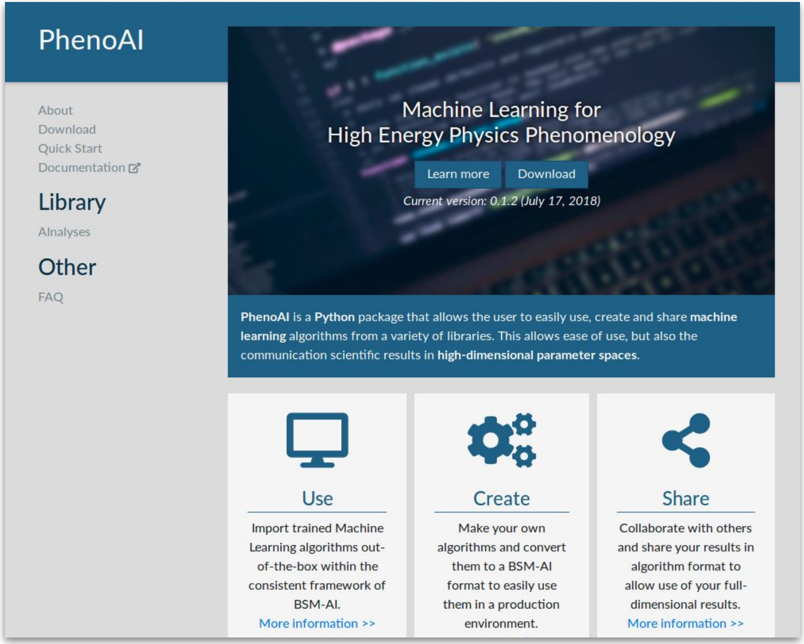
TensorFlow

Keras

scikits learn

PhenoAI *"Pheno for the masses"*

- Stable beta PhenoAI is available via pip3 (`phenoai`) and via the website http://hef.ru.nl/~bstienen/phenoai
- Extensive documentation available
- Started to collect algorithms for AInalysis library



But what about data?

Data publishing

- Individual data points (e.g. model points) are not really informative on their own

- Data can be published on HEPData, but...
  - … lacks an easy interface to navigate and explore the data
  - … data sets can not be easily compared

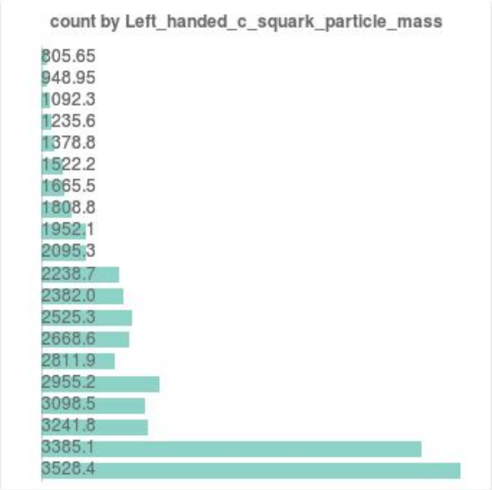
Result: Publishing information like model point evaluations is still not extremely common in our field.

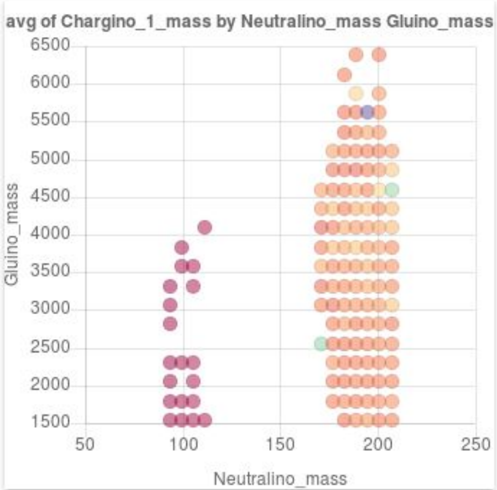
iDarkSurvey for Data Publishing

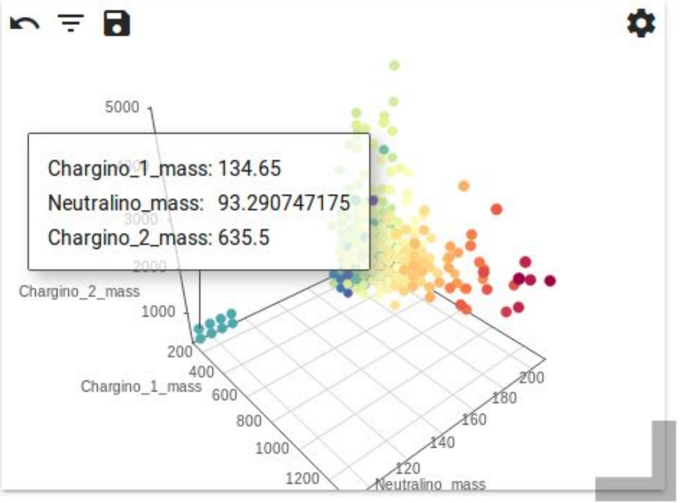
netherlands eScience center

- iDarkSurvey is an instance of SPOT, a plotting and data collection tool
- Online data storage for high energy physics data
- Has online plotting interface to explore data
- Multiple data sets can easily be compared within the same plots
- Own data can be viewed alongside the data in the database
- Online demo at http://www.idarksurvey.org/

iDarkSurvey for Data Publishing





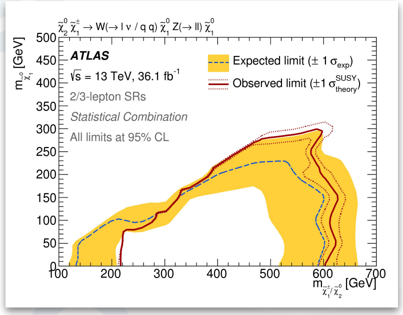


http://www.idarksurvey.org/

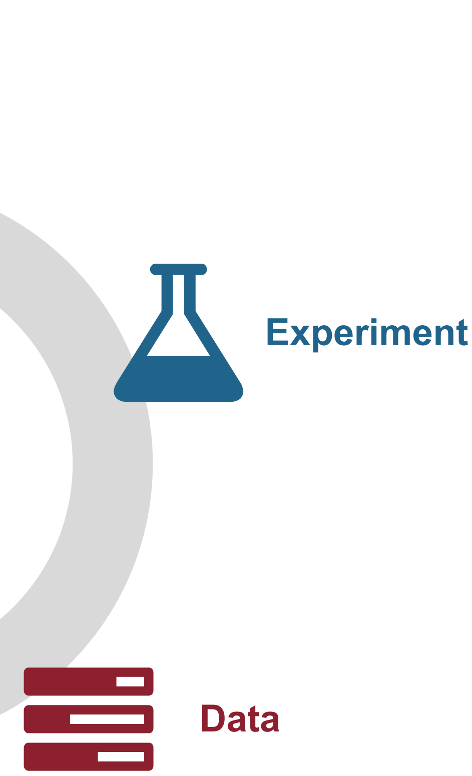
The Circle of Physics

- Sampling in particle physics is most commonly grid sampling, which is intractable for high-dimensional spaces

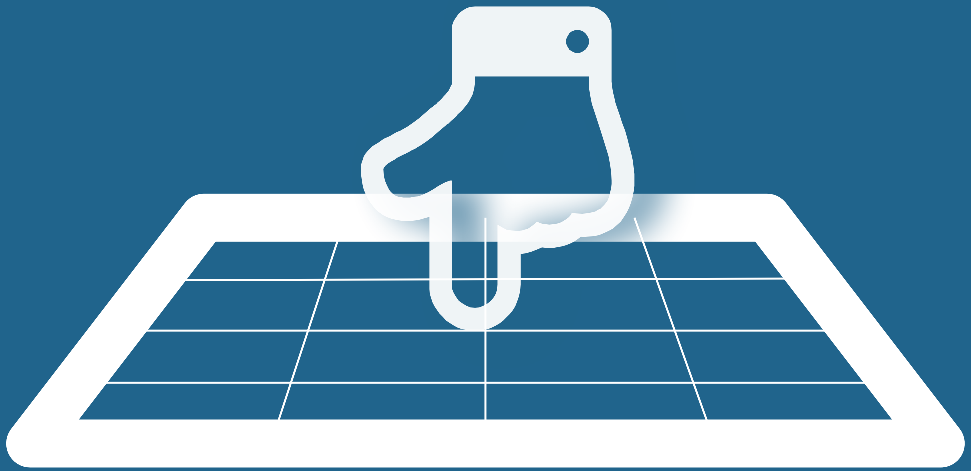
- Evaluation of truth label can take O(hour)



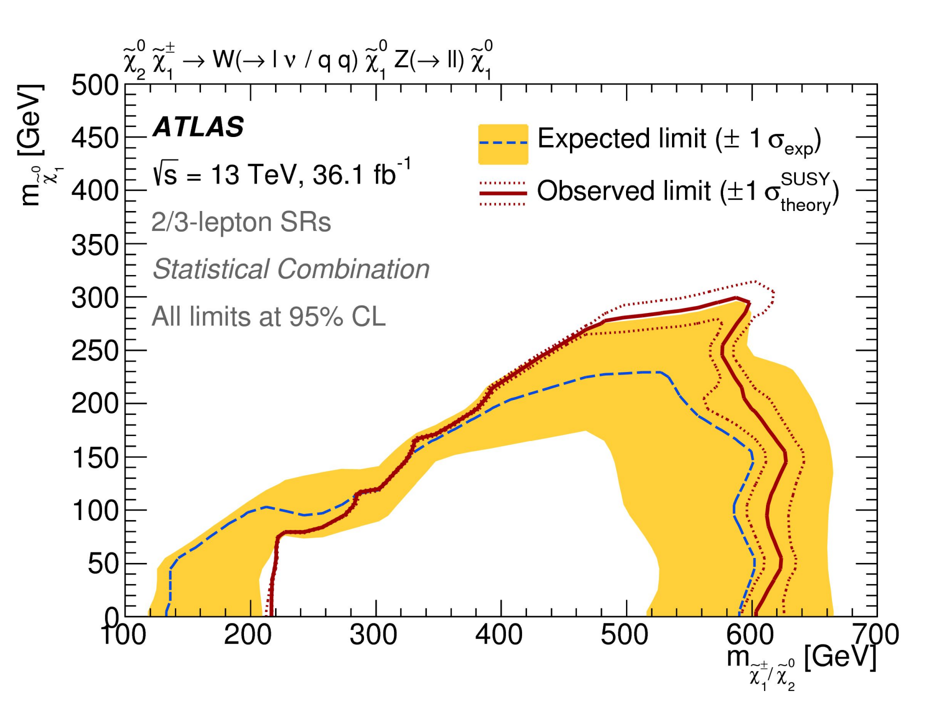
- In ideal world we want "most bang for our buck": get most informative points only



Can we aim our sampling?



Where to aim?

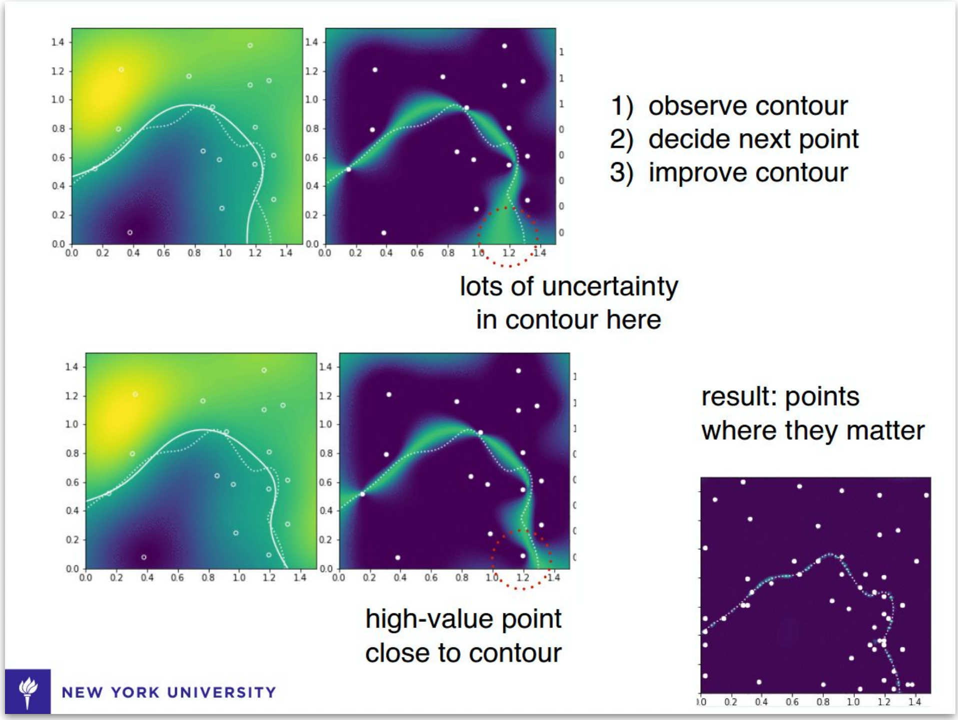


Depends on case, e.g.

- **Binary exclusion**
  Around decision boundary

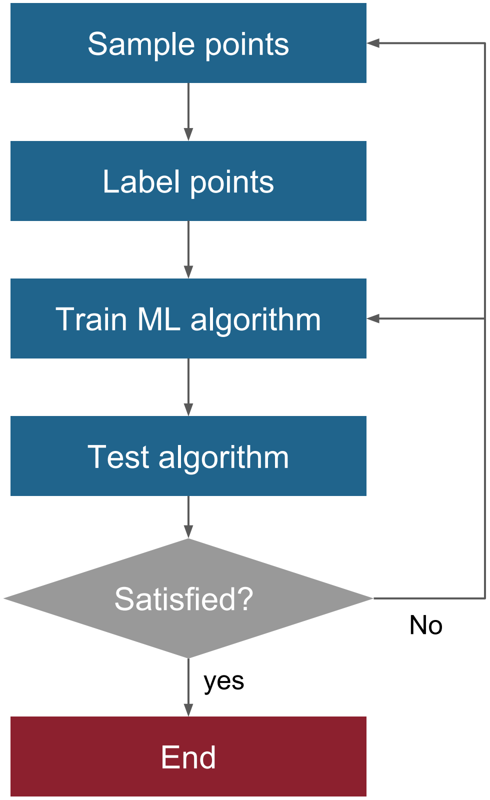
- **Global regression**
  Regions with highest 'uncertainty', could basically be anywhere in the parameter space

Gaussian Processes

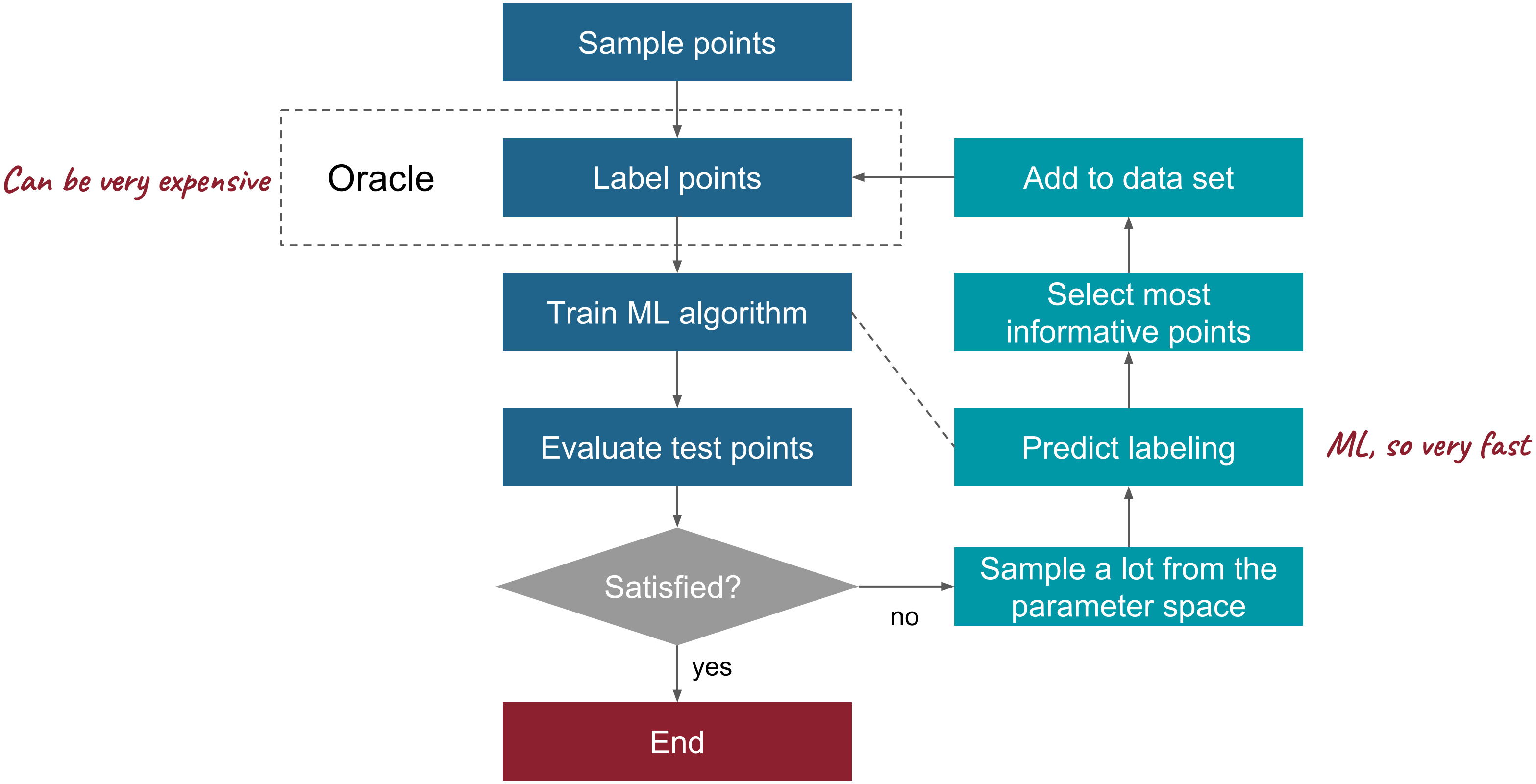


Levelset Estimation by Bayesian Optimization
K. Cranmer, L. Heinrich, G. Louppe
https://indico.cern.ch/event/702612/timetable/

Optimization of Machine Learning algorithm



Active Learning



Active Learning

Select most informative points

Uncertainty sampling

Use output of algorithm as probability:

- Softmax output layer
- Platt scaling
- Other calibration methods

Select points with lowest associated probability.

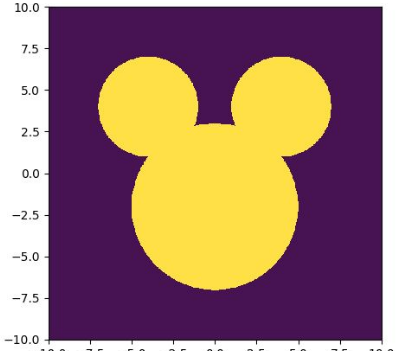
Query by Committee

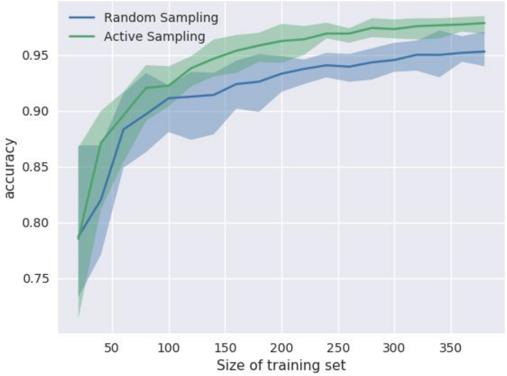
Train multiple algorithms on same data with natural variation:

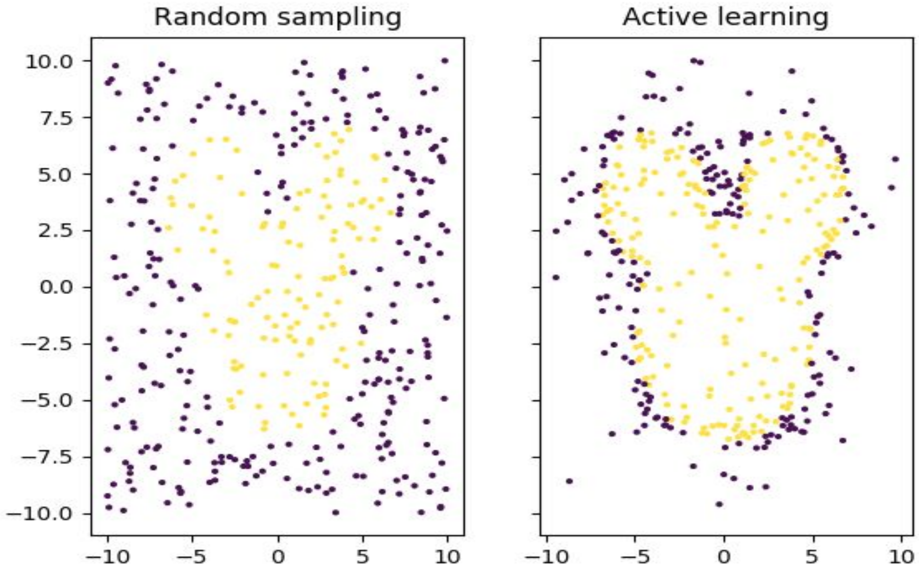
- Bagging
- Vary the algorithms themselves (e.g. different NN architectures)

Let all algorithms make predictions on points, select those points with largest spread in the prediction.

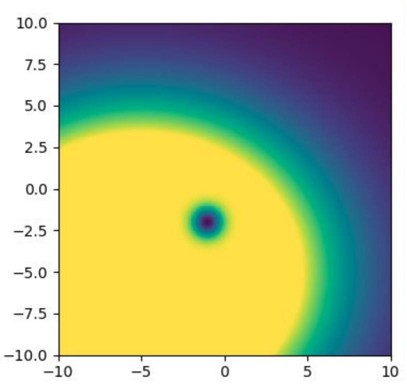
Simplified example I

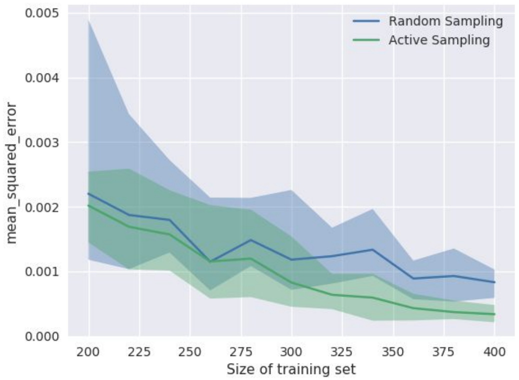


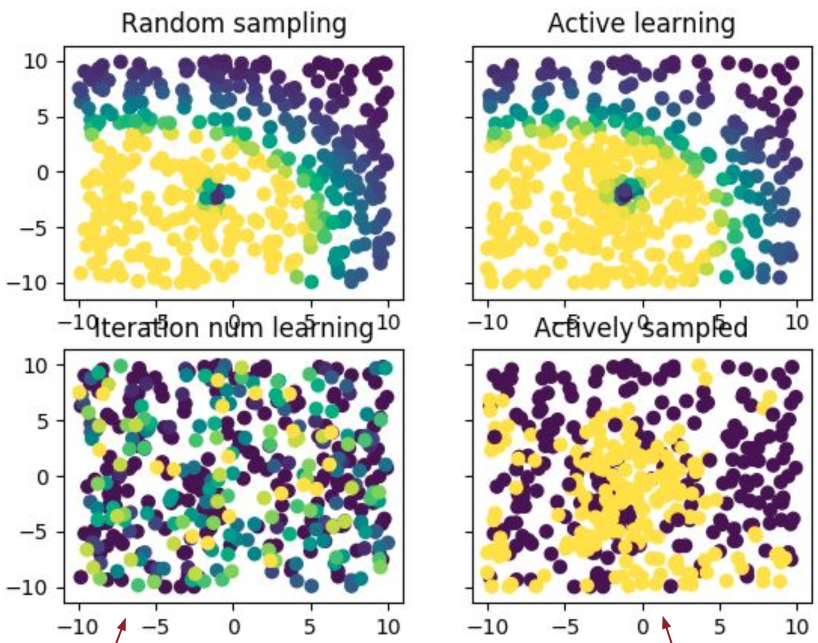




Simplified example II



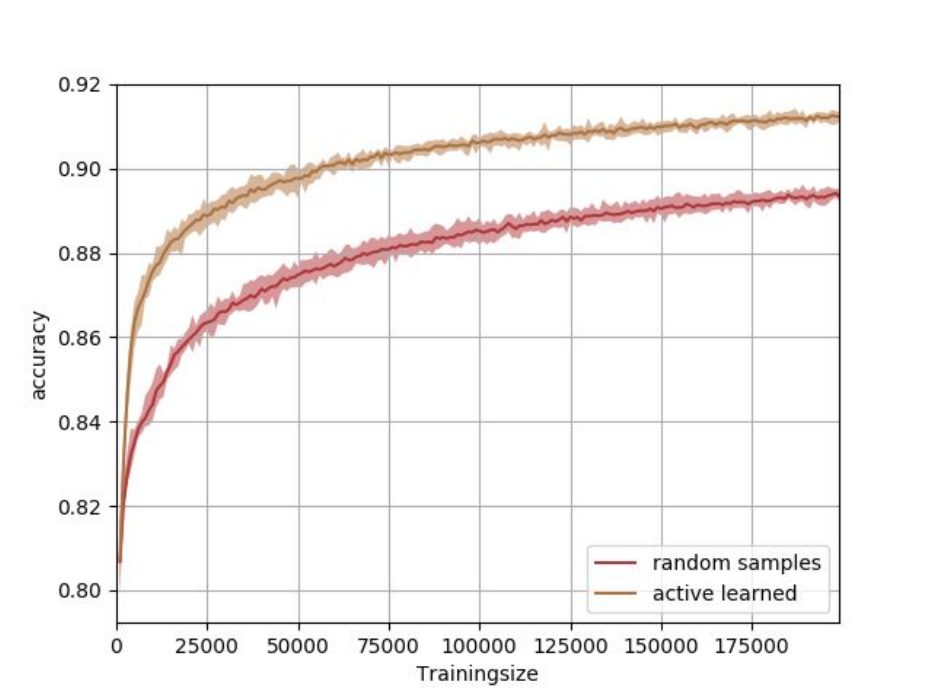




Colour indicates iteration at which the point was selected

Yellow: active learned
Purple: Randomly picked

Real-life example



1. ATLAS pMSSM-19 data (source for SUSY-AI) to train a neural network

2. This NN is the oracle (mimicking the true simulation chain)

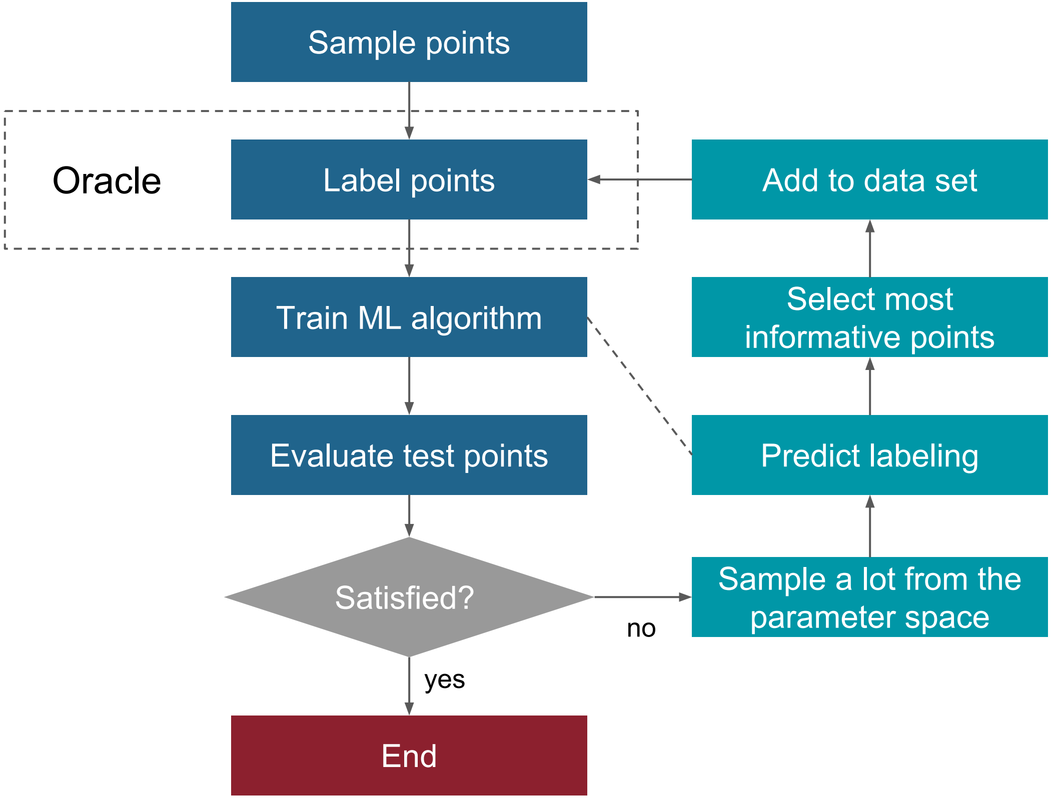
3. Use Active Learning with RandomForests to get accuracy development plot

Active learning

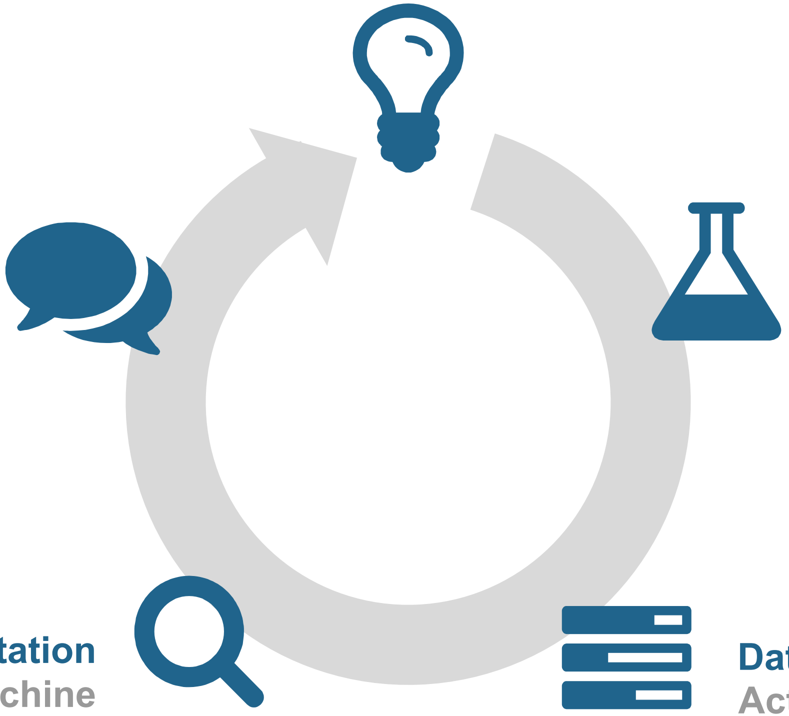
- Works for any dimensionality, as long as ML algorithm is chosen accordingly

- Working on the Gambit MSSM7 data as second real-life example

- Working on first applications



Conclusion



Idea

Experiment

Data
Active learning for more informative samples

Interpretation
with the help of Machine Learning models

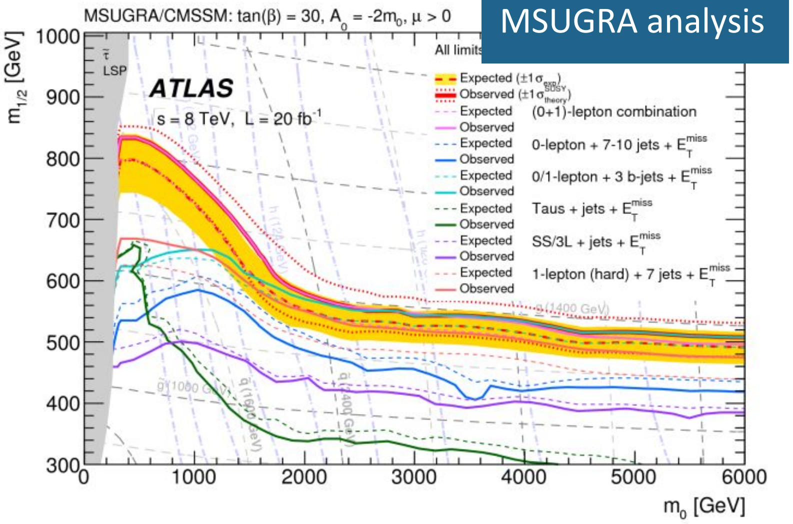
Communication
Publish the Machine learning models themselves and their training data for high-dimensional reinterpretation

Extra slides

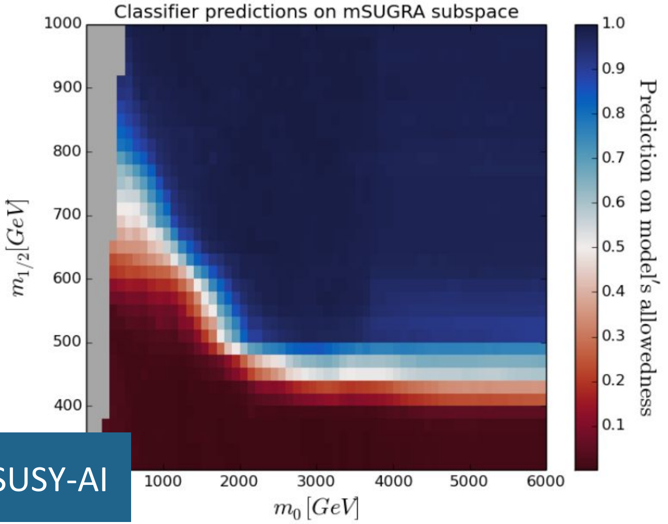
What about my simplified model?

Training on a full model still allows access to submodels. SUSY-AI was trained on the pMSSM19, of which MSUGRA/CMSSM is a submodel.

MSUGRA analysis



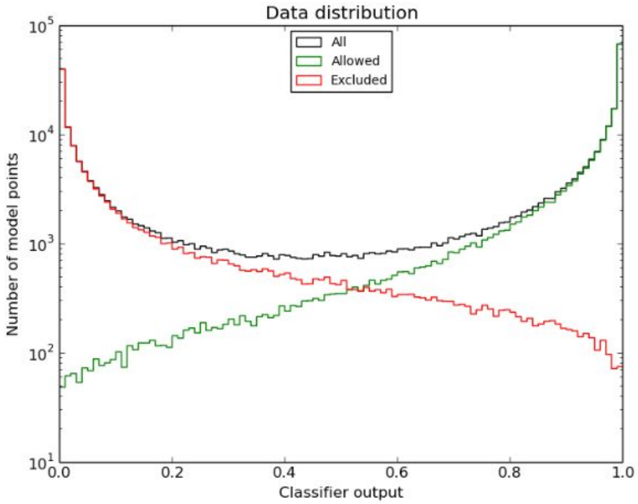
SUSY-AI

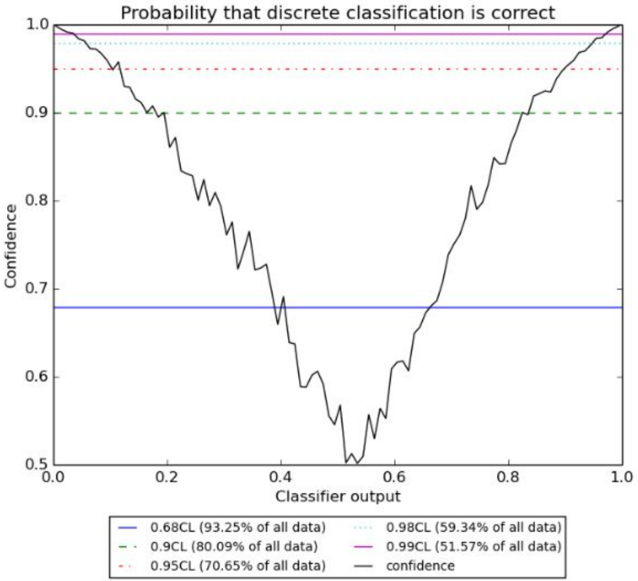


Comparison not entirely fair: the dedicated MSUGRA/CMSSM scan combined signal regions in a smart way, whereas the exclusion of the SUSY-AI dataset uses the simple: “if excluded by any analysis -> excluded”

Confidence construction from SUSY-AI

SUSY-AI is a classifier, but outputs a continuous value between 0 (excluded) and 1 (allowed). It can *not* be interpreted as a probability, but can be transformed into one.





Is PhenoAI really that simple?

```
from phenoai.phenoai import PhenoAI

master = PhenoAI()
master.add("./example_ainalysis", "example")
result = master.run(X)
```

Yes

Learning to use PhenoAI

PhenoAI aims to be as easy to use as possible. To this end we have created:

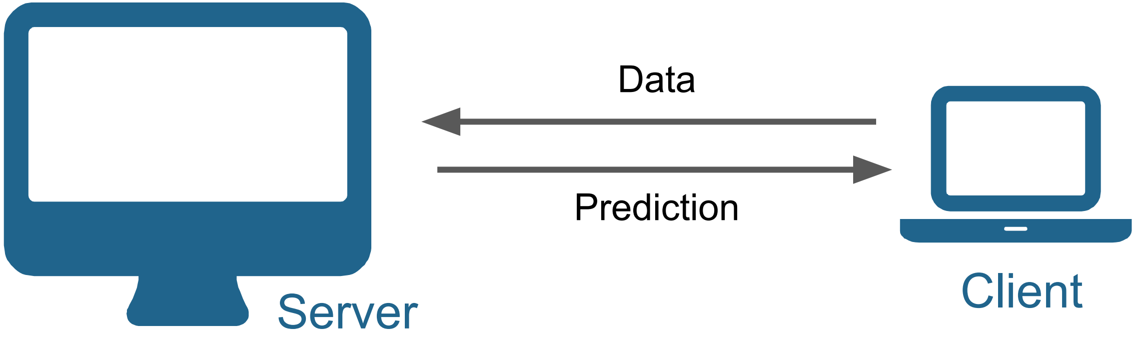
- online documentation
- in-code documentation
- example scripts
- a quick start manual

We are busy optimizing the learning experience of PhenoAI even further, making material as a tutorial and a cheat sheet.

Server-client structure

PhenoAI has a built-in ability to create a server-client structure. The server has the AInalyses loaded, the client can be added to any script and will query the server for prediction on a specific data set. In this way, the loading and configuration overhead are needed only once.

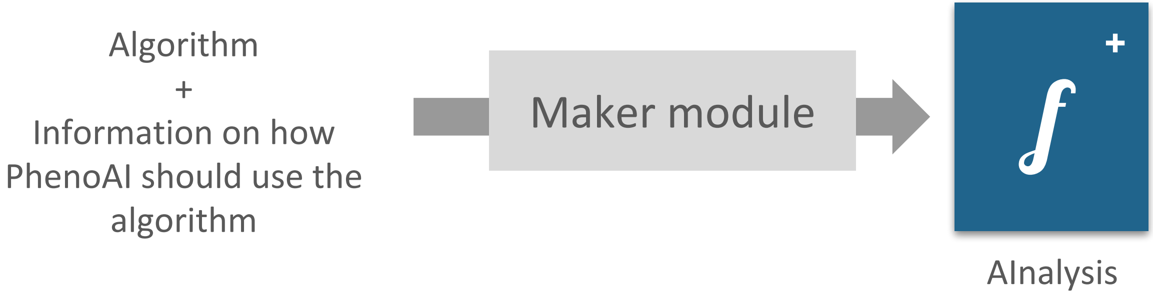
Server and client can of course just be the same machine



Maker module

In order to use a trained algorithm within PhenoAI, it needs to be stored within a folder with a PhenoAI configuration file. This collective as files is called an AInalysis and can, in principle, be made by hand. It is however more convenient to use the `phenoai.maker` module. Which will indicate if errors are made.

Example scripts on how to use the maker module are availble.



DarkMachines

PhenoAI is connected to the DarkMachines initiative as well, a research collective aiming to unravel the mystery that is dark matter with the help of machine learning. See darkmachines.org for more information.