SCHOOL OF DATA ANALYSIS

Learning Machine Learning through High Energy Physics

DESY, Zeuthen 2017

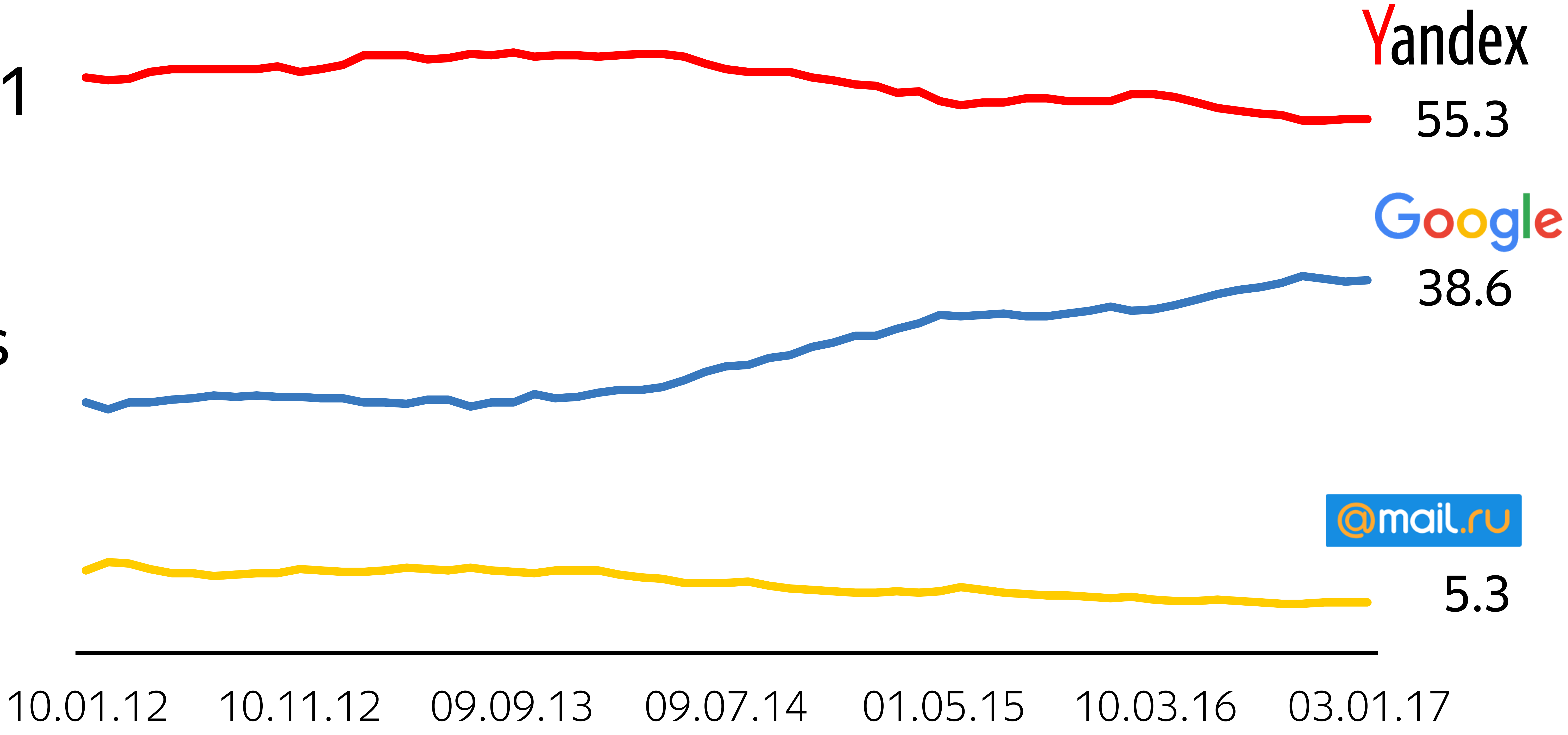
Andrey Ustyuzhanin
Yandex School of Data Analysis, Higher School of Economics

What is Yandex

Public company since 2011

In Q3 2016 search share across different platforms were approximately[1]:

64% on desktop
38% on Android
42% on iOS

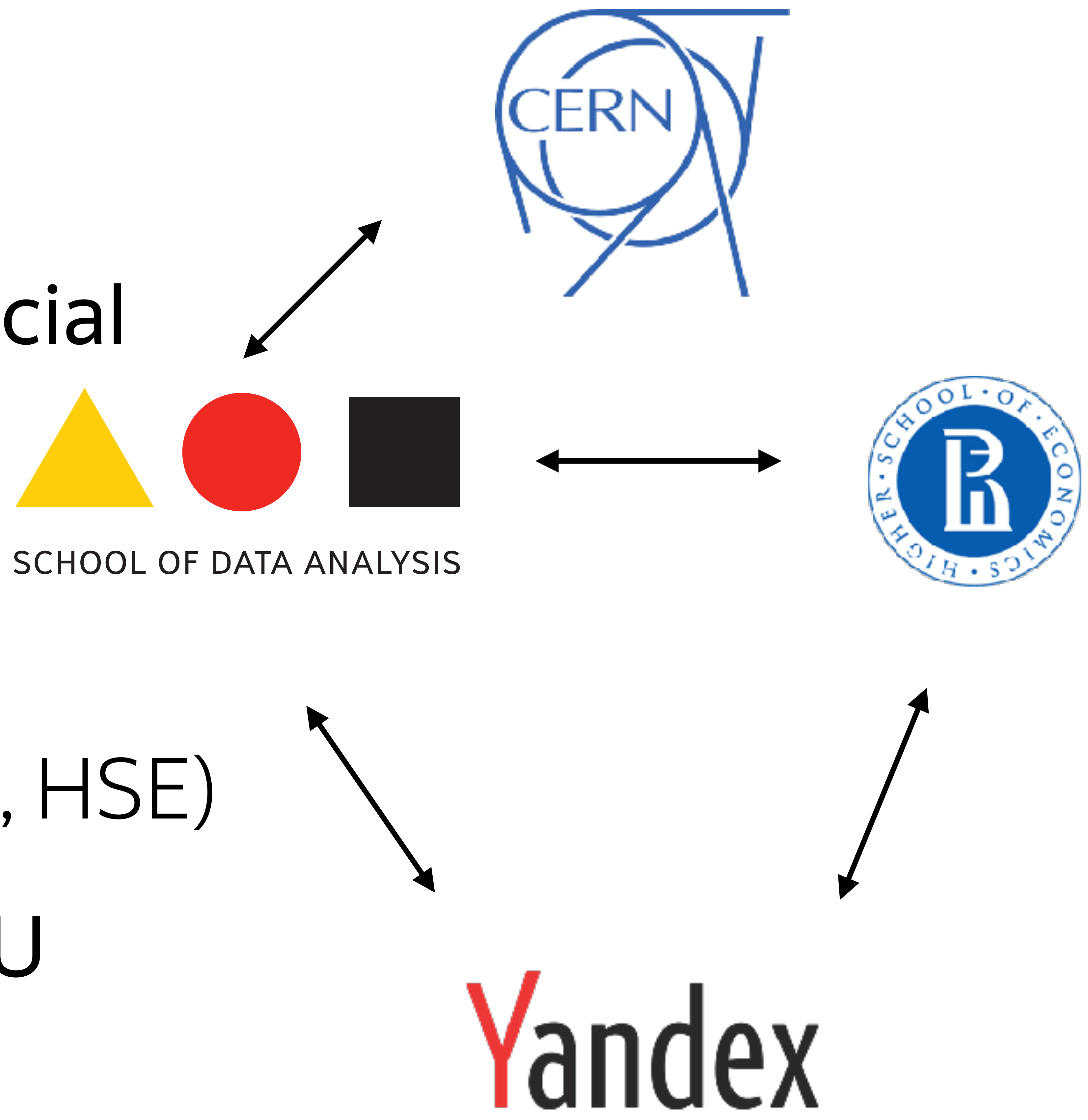


Source: Liveinternet.ru 2012-December, 2016; includes desktop and mobile

[1] Based on company estimates, as provided on Q3 2016 earnings call

Our group

- Yandex School of Data Analysis (YSDA) - non-commercial educational organisation;
- Research group at Yandex School of Data Analysis
  - 2 physicists (PhD), 8 data scientists (6 of them are graduate/undergraduate students from MIPT, MSU, HSE)
- Laboratory of Methods for Big Data Analysis, HSE NRU
- YSDA is member of HEP collaborations:
  - CERN: LHCb (since 2015), SHiP (since 2014)
  - CRAYFIS (since 2015), OPERA



Working group directions

Research & Development

- Solving scientific and technical HEP challenges from LHCb, SHiP, CRAYFIS by means of Machine Learning

Education

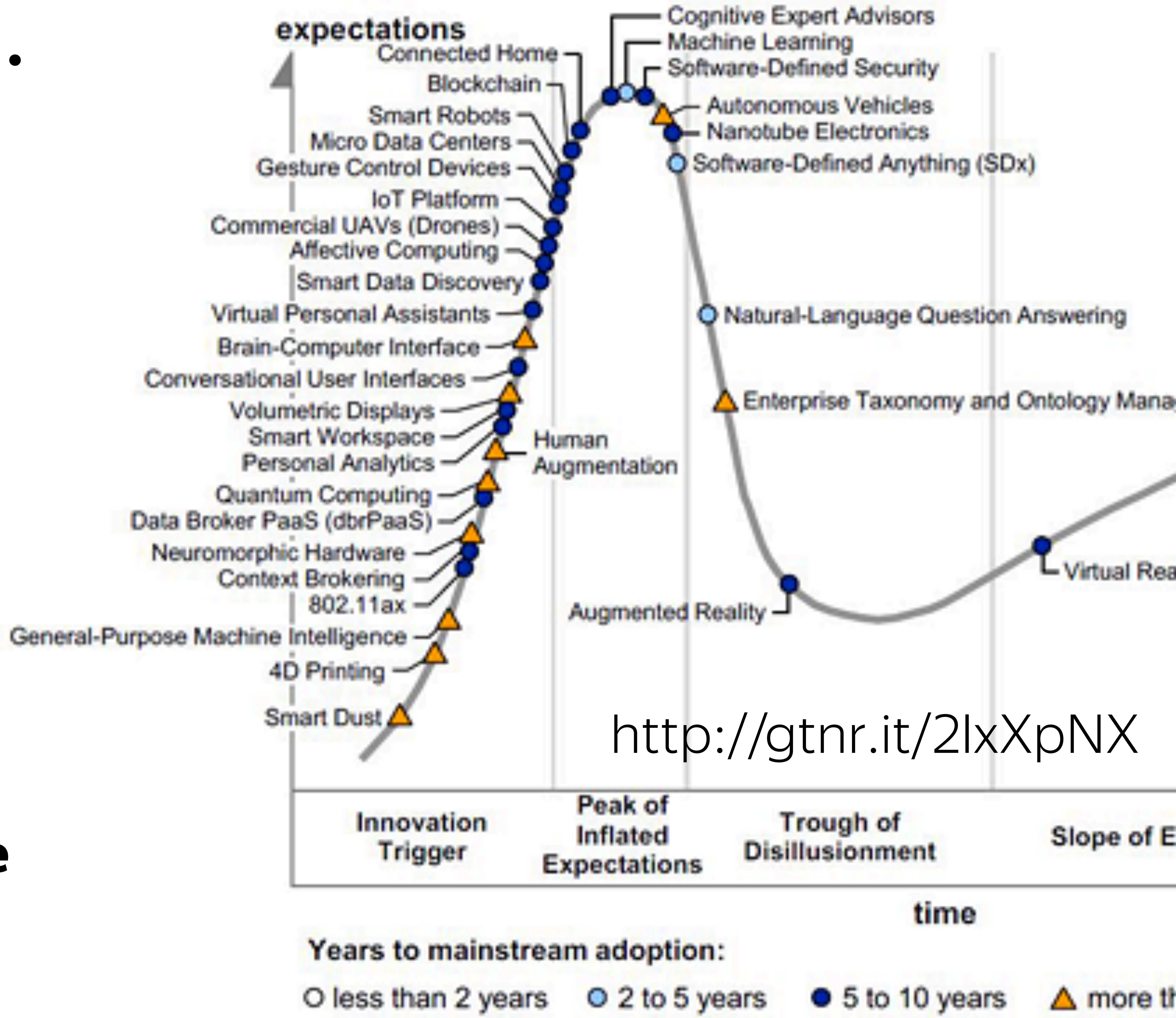
- Machine Learning Courses (YSDA, Imperial College London, Helsinki CSC)
- Summer Schools on Machine Learning (bit.ly/mlhep2016, bit.ly/mlhep2015, bit.ly/mlhep2017)

Outreach

- Masterclasses, Data&Science (https://events.yandex.ru/events/ds/),
- Hackatnones on data science, Machine Learning (http://bit.ly/2lxUWCO)

Machine Learning is …

- Pandemia
- Technomagic
- Panacea
- Answer to Big Data Challenge
- King of the hill (right)
- Disciple of statistics and optimisation methods
- AI harbringer
- **Central part of the Data Science**



http://gtnr.it/2lxXpNX

Source: Gartner (July 2016)

Namely

Machine Learning is about learning algorithms $\boldsymbol{A}$ that:

- defined on sample set $\mathscr{X}$ (e.g. R^n) and targets $\mathscr{Y}$ (e.g. {0, 1});
- take a problem (dataset) $\boldsymbol{D} = (\boldsymbol{X}, \boldsymbol{y}) \subseteq \mathscr{X} \times \mathscr{Y}$;
- learn relation between $\mathscr{X}$ and $\mathscr{Y}$;
- and return prediction function:

$$\boldsymbol{A}(\boldsymbol{D}) = f$$
$$f : \mathscr{X} \rightarrow \mathscr{Y}$$

- that minimises given metrics (loss function $\mathscr{L}$)

«No Free Lunch» Theorem

> No free lunch theorem states that on average by all datasets all learning algorithms are equally bad at learning.

For example:

- crazy algorithm *A*(θ):

$$f(x) = \left\lfloor \left(\left\lceil \sum_i x_i + \theta \right\rceil \mod 17 + 1027 \right)^{\pi} \right\rfloor \mod 2$$

- and SVM

perform equally [bad] **on average** for **all possible datasets.**

So are ML algorithms useless?

No Free Lunch theorem applies to:

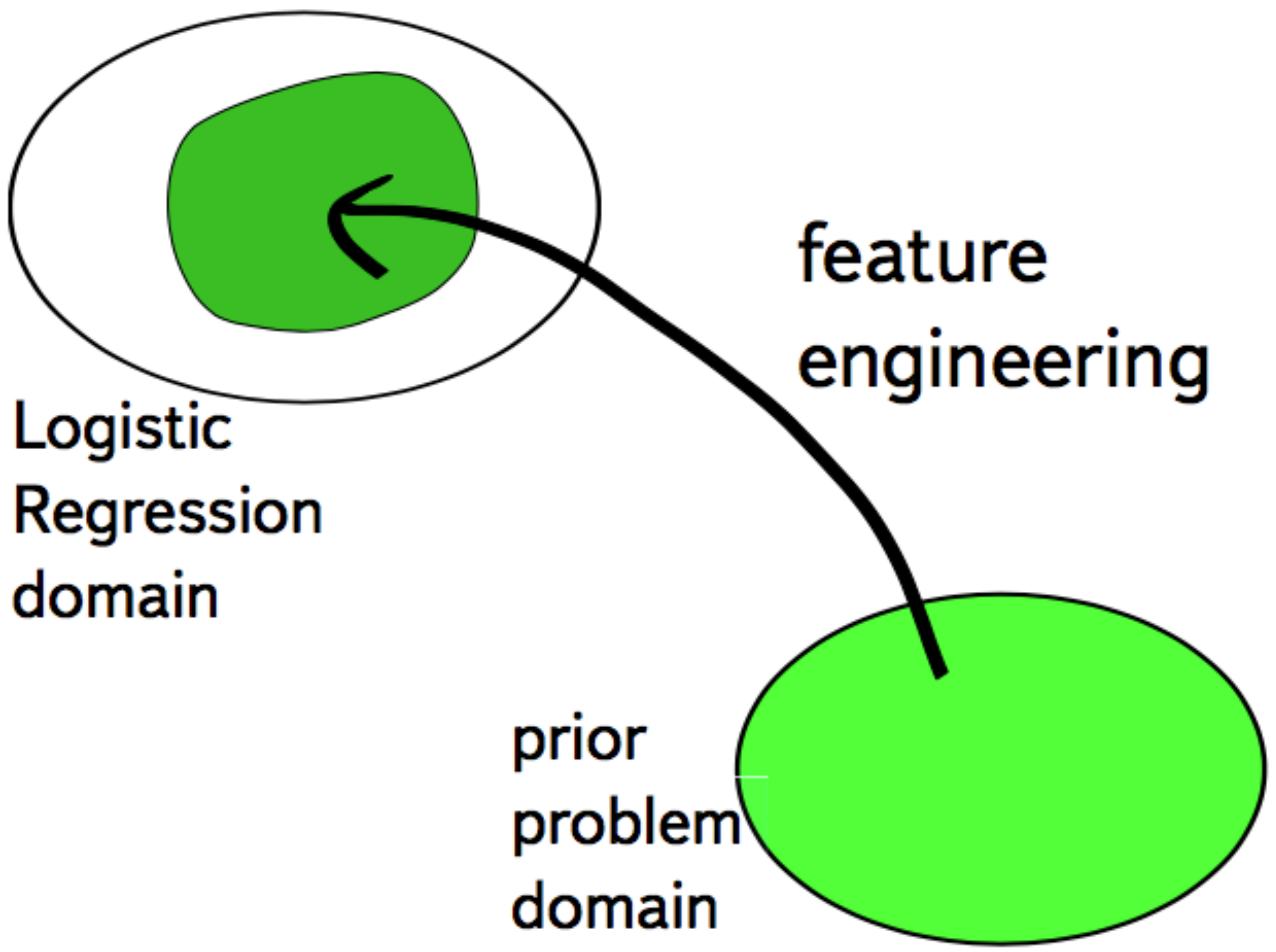
- one learning algorithm;
- against all possible problems.

in real world:

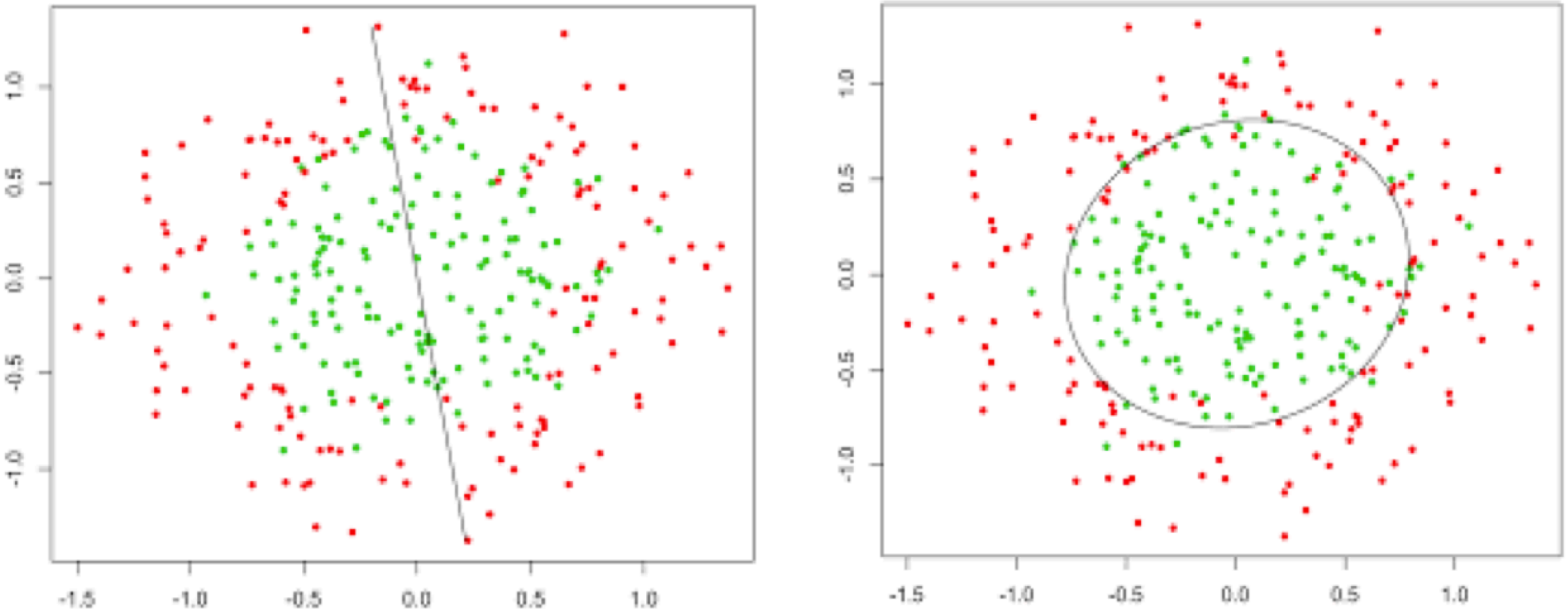
- **data scientist** with prior knowledge of the world;
- problem description;
- data description;
- a set of standard algorithms.

Traditional Machine Learning (simplified)

- analyse a problem and make assumptions;
- pick an algorithm from a toolkit (e.g. logistic regression);
- provide assumptions suitable for the algorithm (feature engineering).



Feature engineering illustration



How can we separate green from red by linear model?


(C) CERN

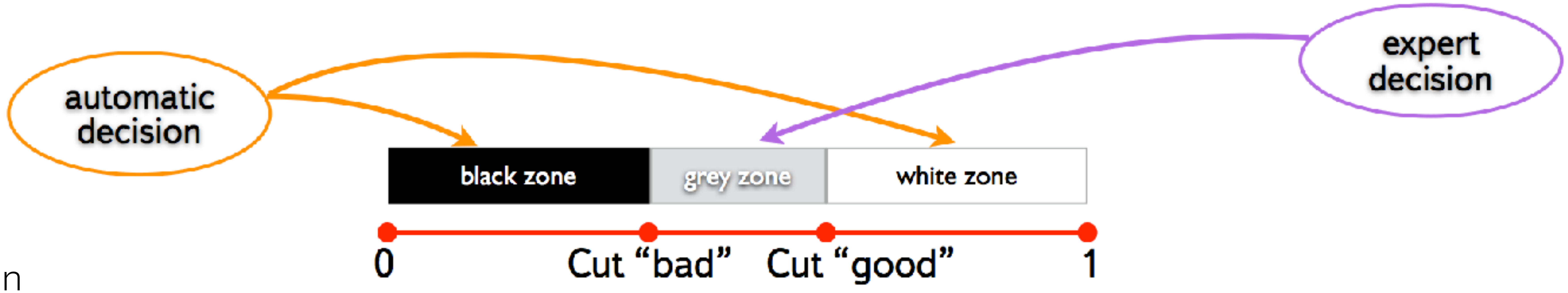
Problem 1: Data Certification (CMS)

- Traditionally, quality of the data at CERN CMS experiment is determined manually which requires considerable amount of human efforts;
- ML can save some of those efforts;
- Data: CMS 2010B run open data;
- Aim: automated classification of Lumisections as “good” or “bad”;
- Features: particle flow jets, Calorimeter Jets, Photons, Muons;
- The dataset was flagged by experts (3 FTE).

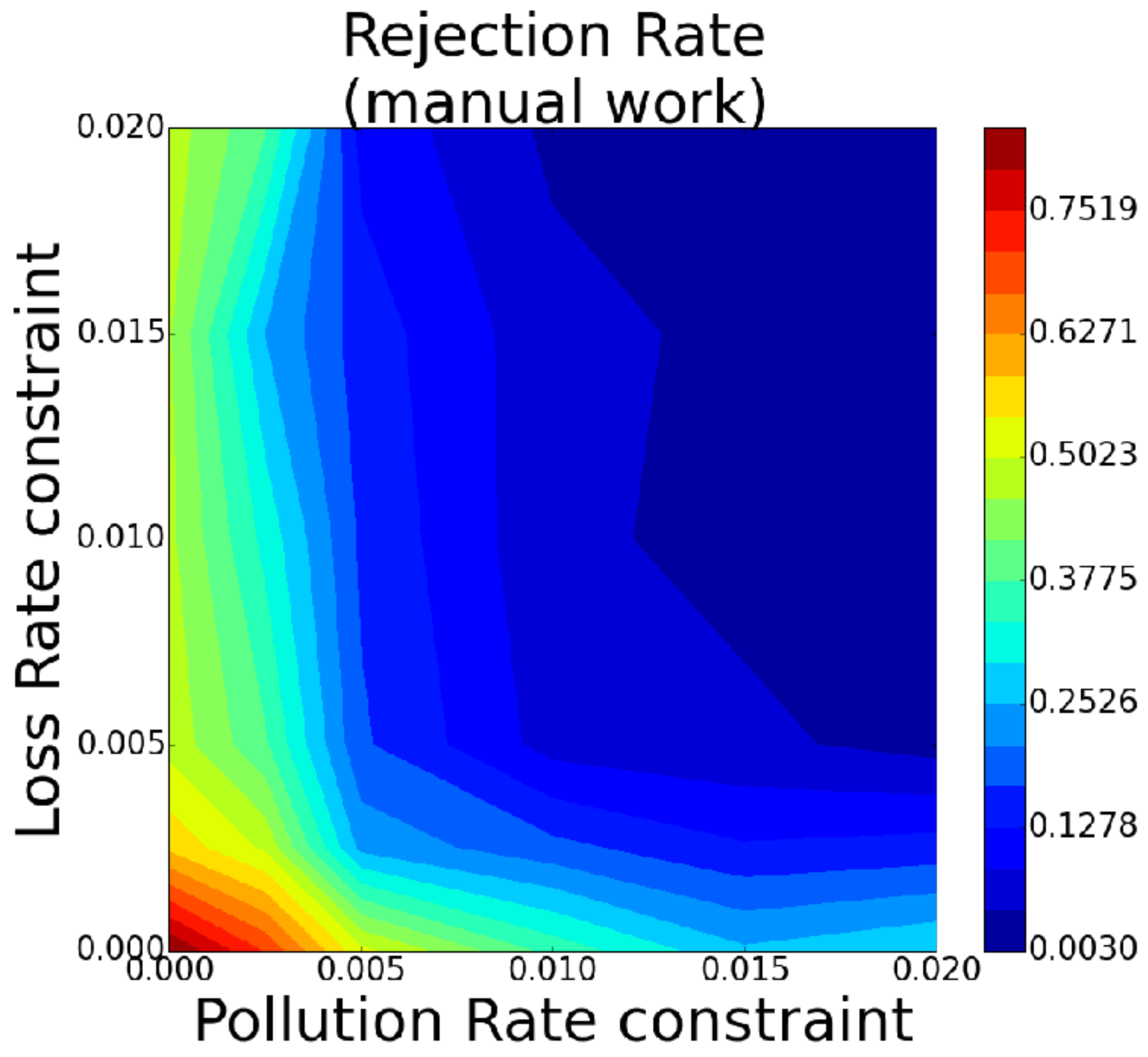
$$\text{Rejection Rate} = \frac{\text{Rejected}}{\text{Total quantity of samples}} \to \min;$$

$$\text{Pollution Rate} = \frac{\text{False Positive}}{\text{True Positive} + \text{False Positive}} \leq \text{const};$$

$$\text{Loss Rate} = \frac{\text{False Negative}}{\text{True Positive} + \text{False Negative}} \leq \text{const}.$$



Results



- The aim is to minimise the Manual work with low Loss Rate ("good" classified as "bad") and Pollution Rate ("bad" classified as "good");

- ~80% saving on manual work is feasible for Pollution & Loss rate of 0.5%.
- Next steps: adopt technique for 2016 data & run in production

http://bit.ly/2l0MLiN

Machine Learning Challenges

Complications

Lumisection representation
Feature engineering
Continuous quality update

Algorithms:

Supervised learning, binary classification:

- Neural Networks, Gradient Boosting

Active Learning

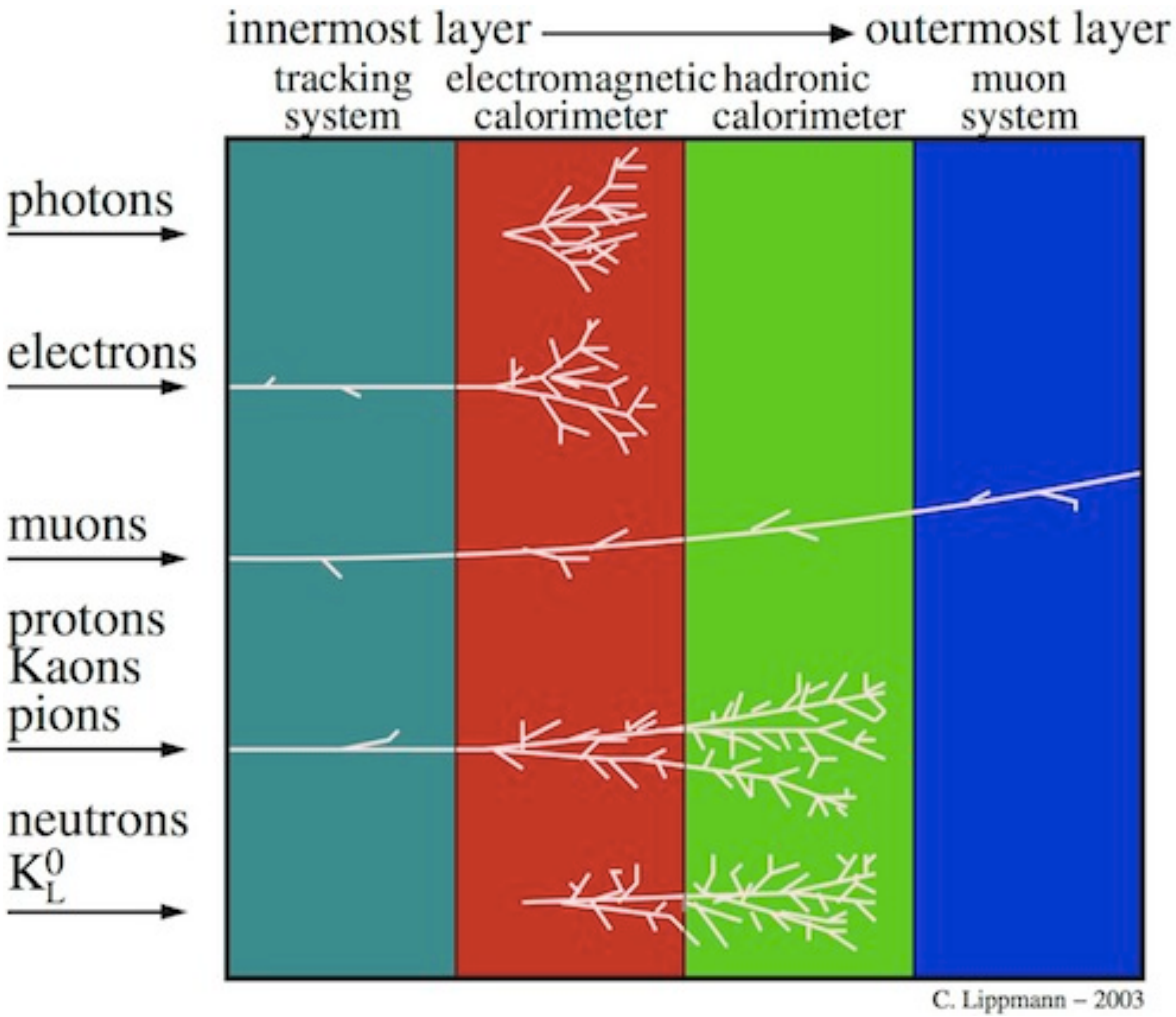
Problem 2: LHCb Particle Identification (PID)

- Problem: identify charged particle associated with a track (multiclass classification problem)

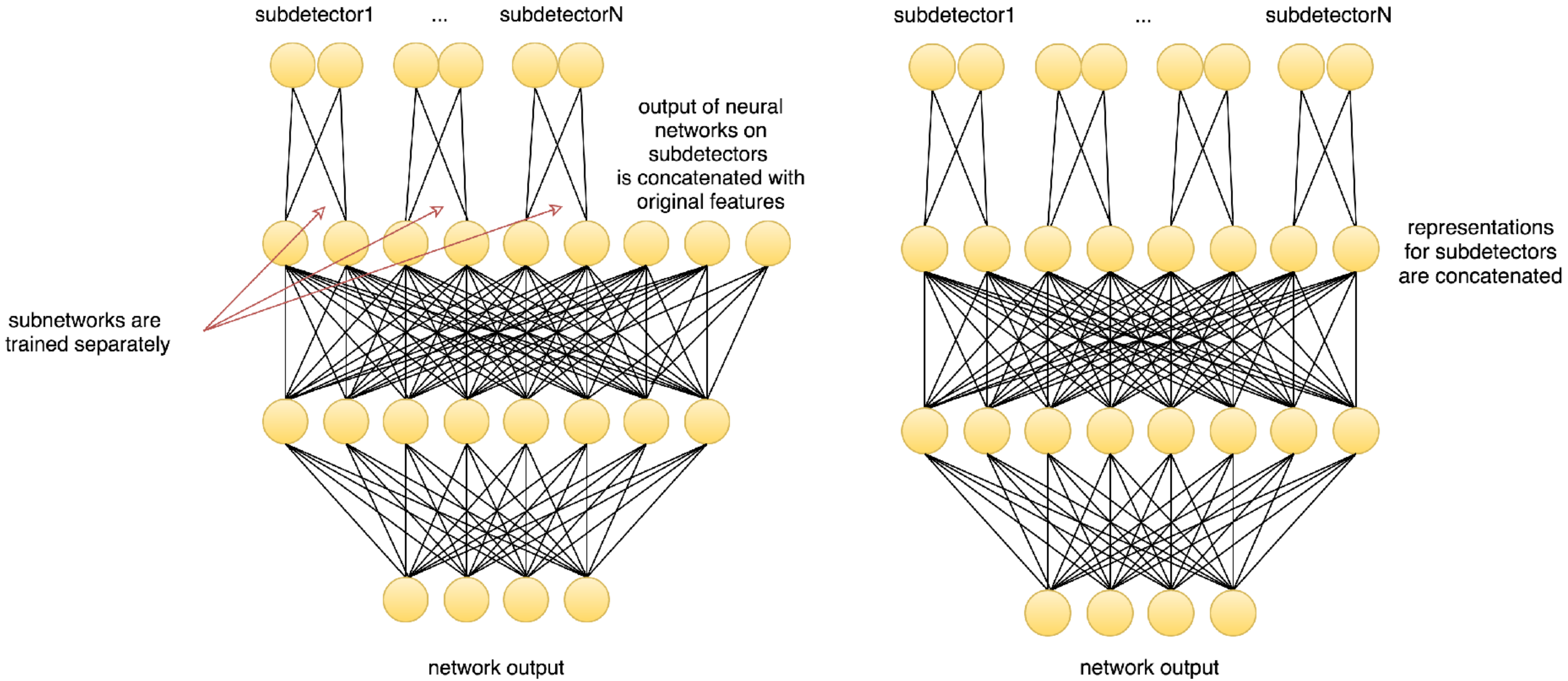
  particle types: Electron, Muon, Pion, Kaon, Proton and Ghost;

- LHCb detector provides diverse plentiful information, collected by subdetectors: **CALO**, **RICH**, **Muon** and **Track** observables, his information should be **combined;**

- Monte Carlo-simulated samples.



Neural Networks: Stacking and Special

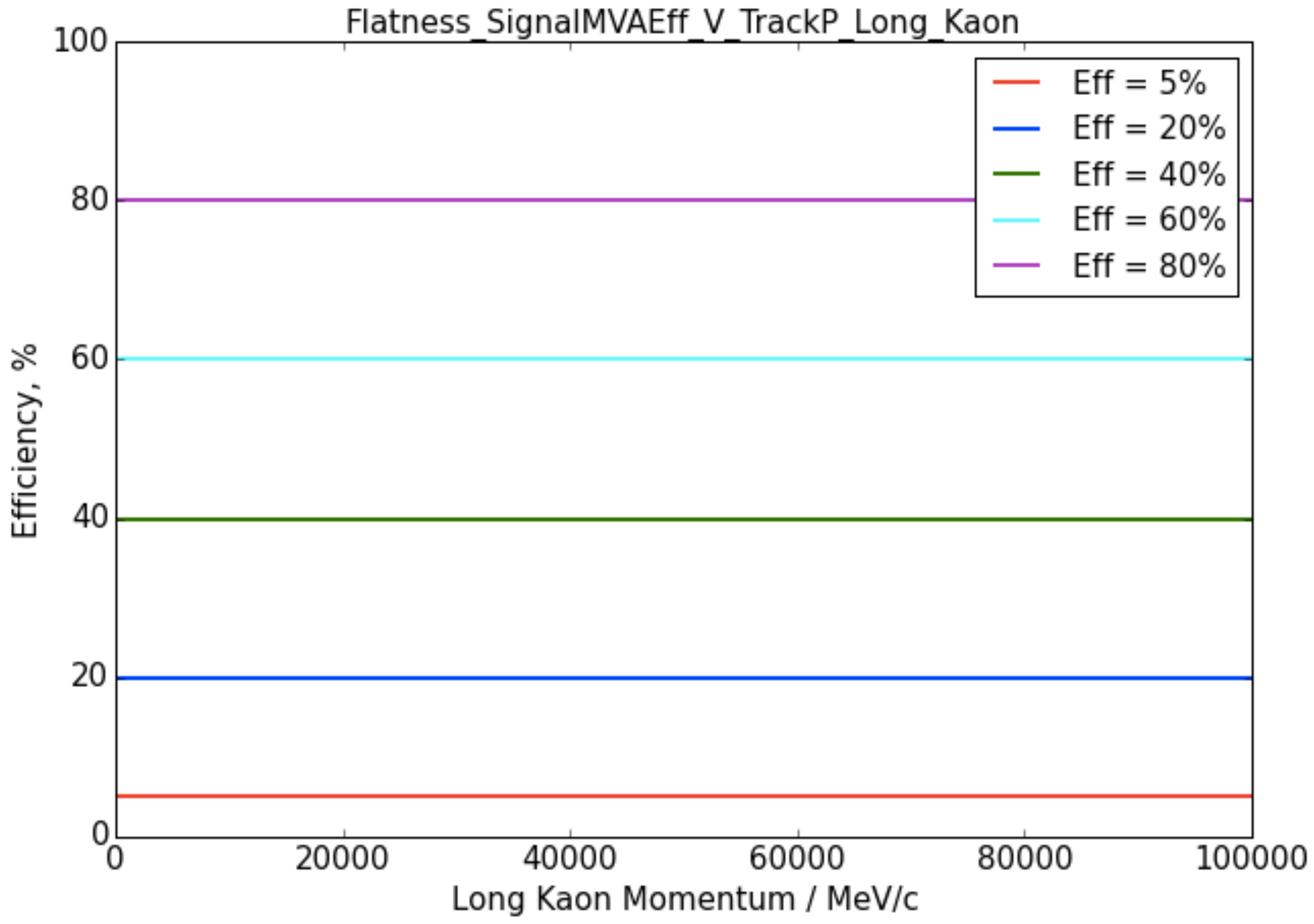


Models AUCs

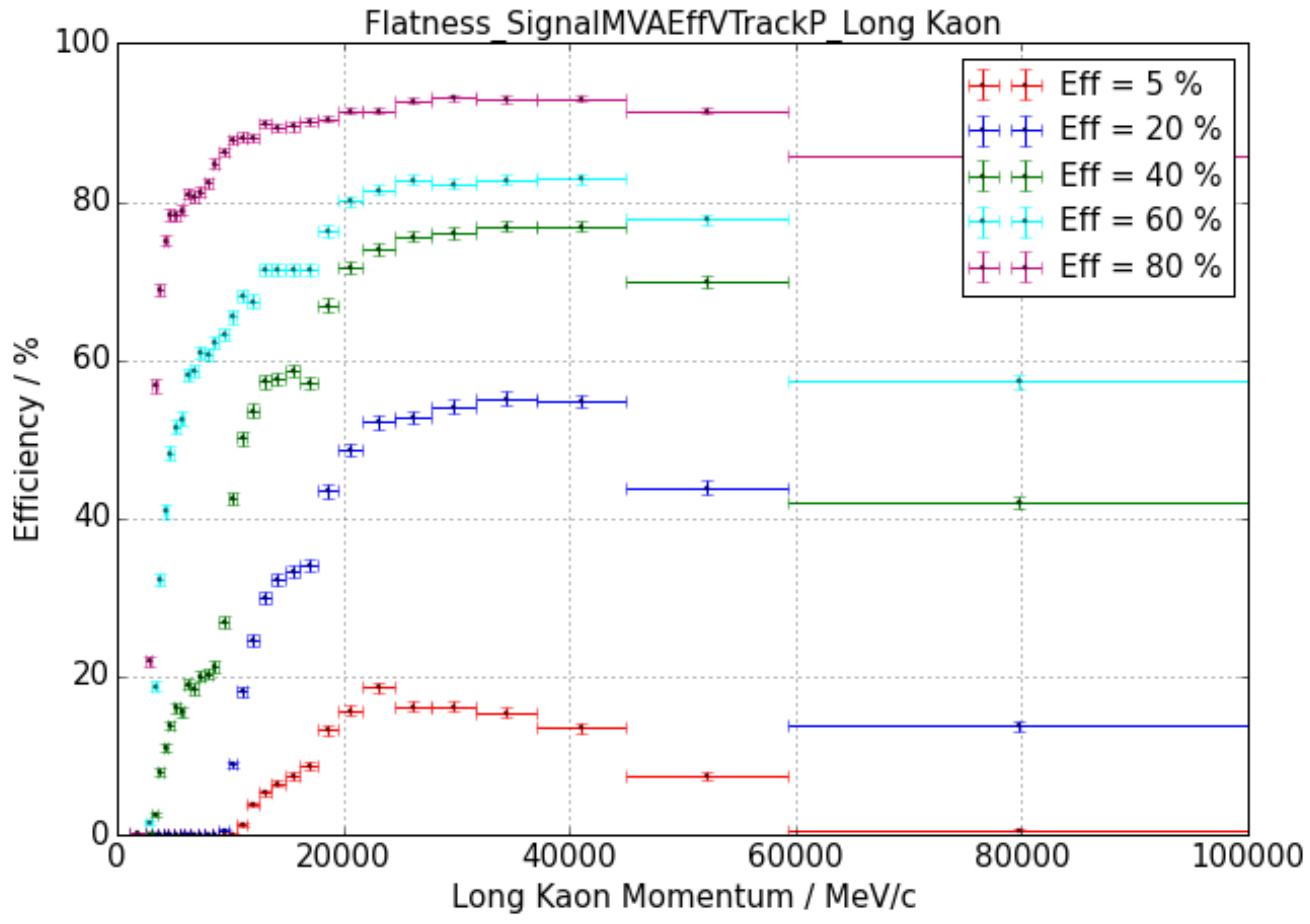
	Ghost	Electron	Muon	Pion	Kaon	Proton
baseline	0.9484	0.9854	0.9844	0.9345	0.9147	0.9178
keras DL	0.9632	0.9914	0.9925	0.9587	0.9319	0.9320
XGBoost	0.9609	0.9908	0.9922	0.9568	0.9303	0.9302
special BDT	0.9636	0.9913	0.9926	0.9576	0.9309	0.9310

- ROC AUC - a generic ML quality metric, deviation is ~10^{-4}, due to large training/testing sample
- BDT has similar quality to keras DL
- Training procedure and prediction time for BDT grows up linearly depending on number of classes

Improving PID with flat models



Ideal world



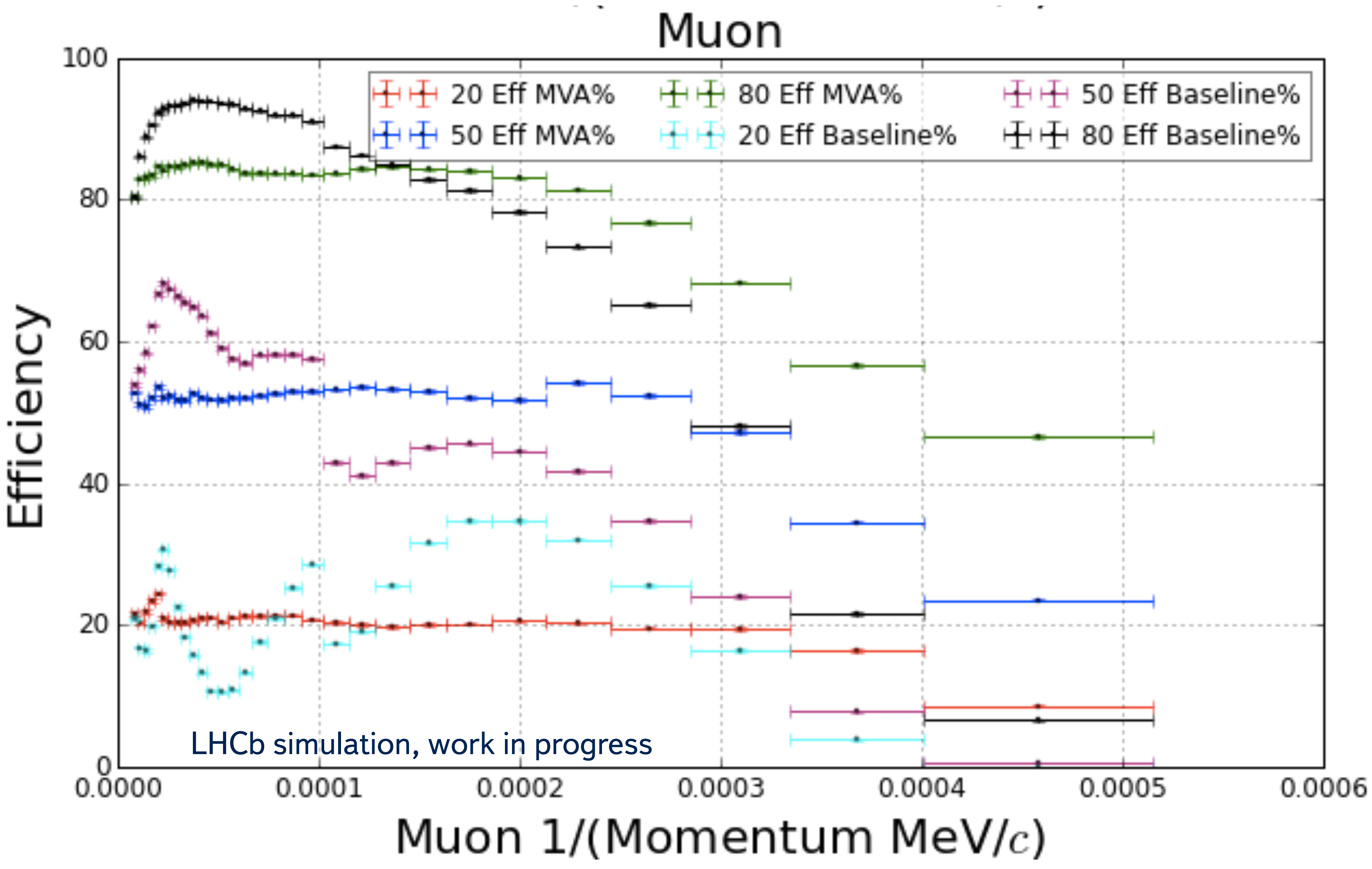
Real world

Information from subdetectors strongly depends on particle momentum (energy), that leads to strong dependency between PID efficiency and momentum. Undesirable for physics analyses.

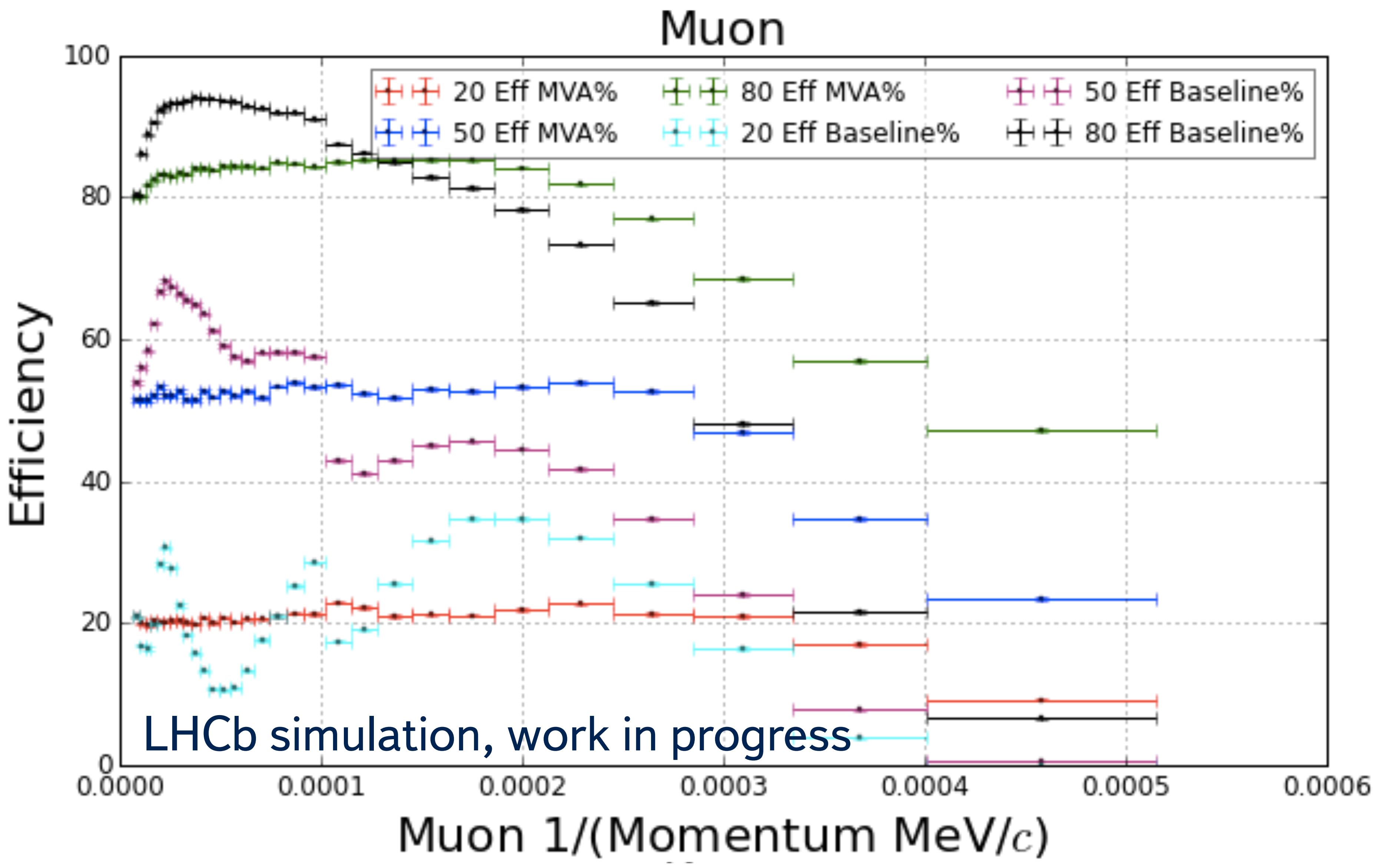
Flat Model vs Baseline

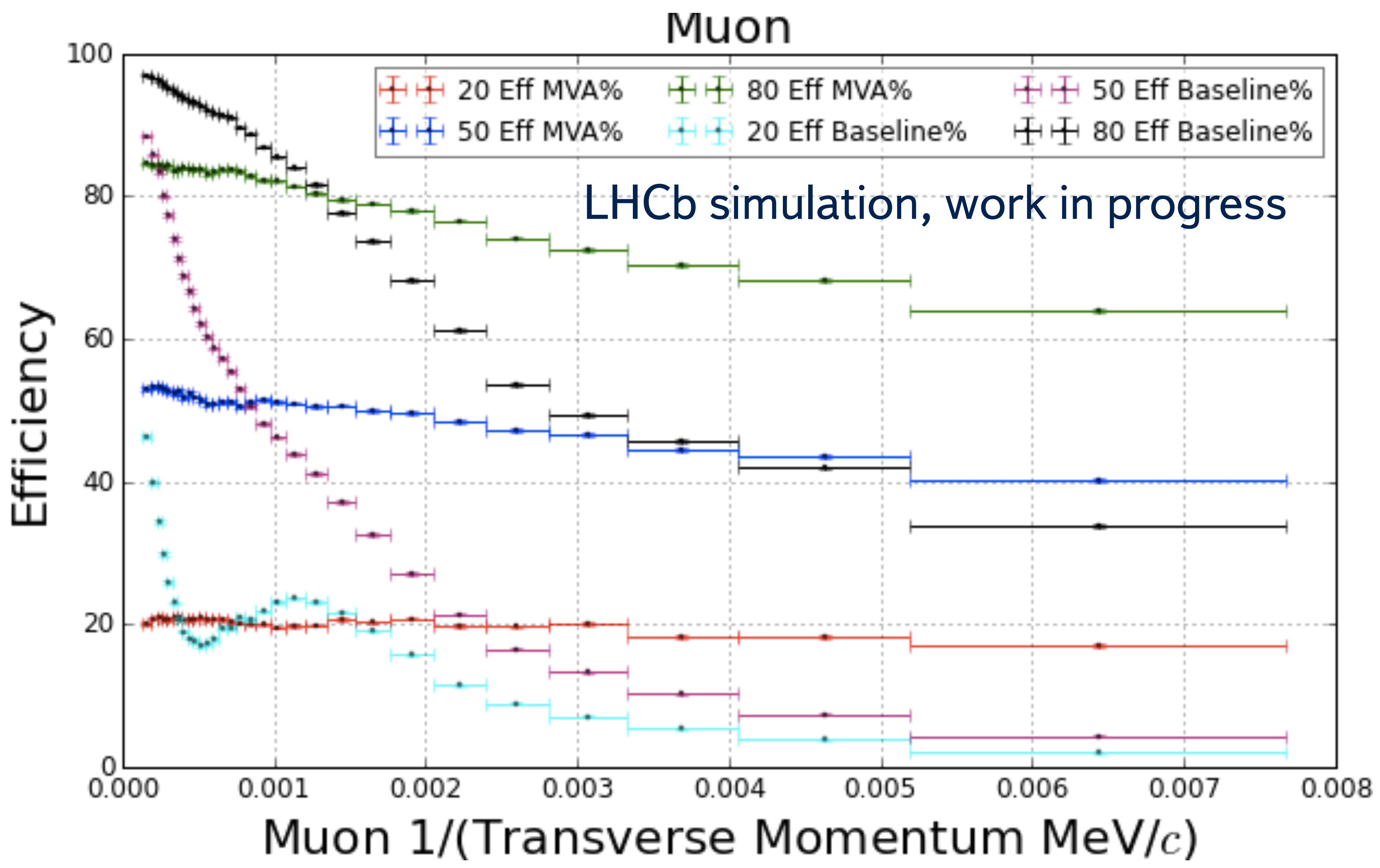
Uniform boosting suppresses this dependency:

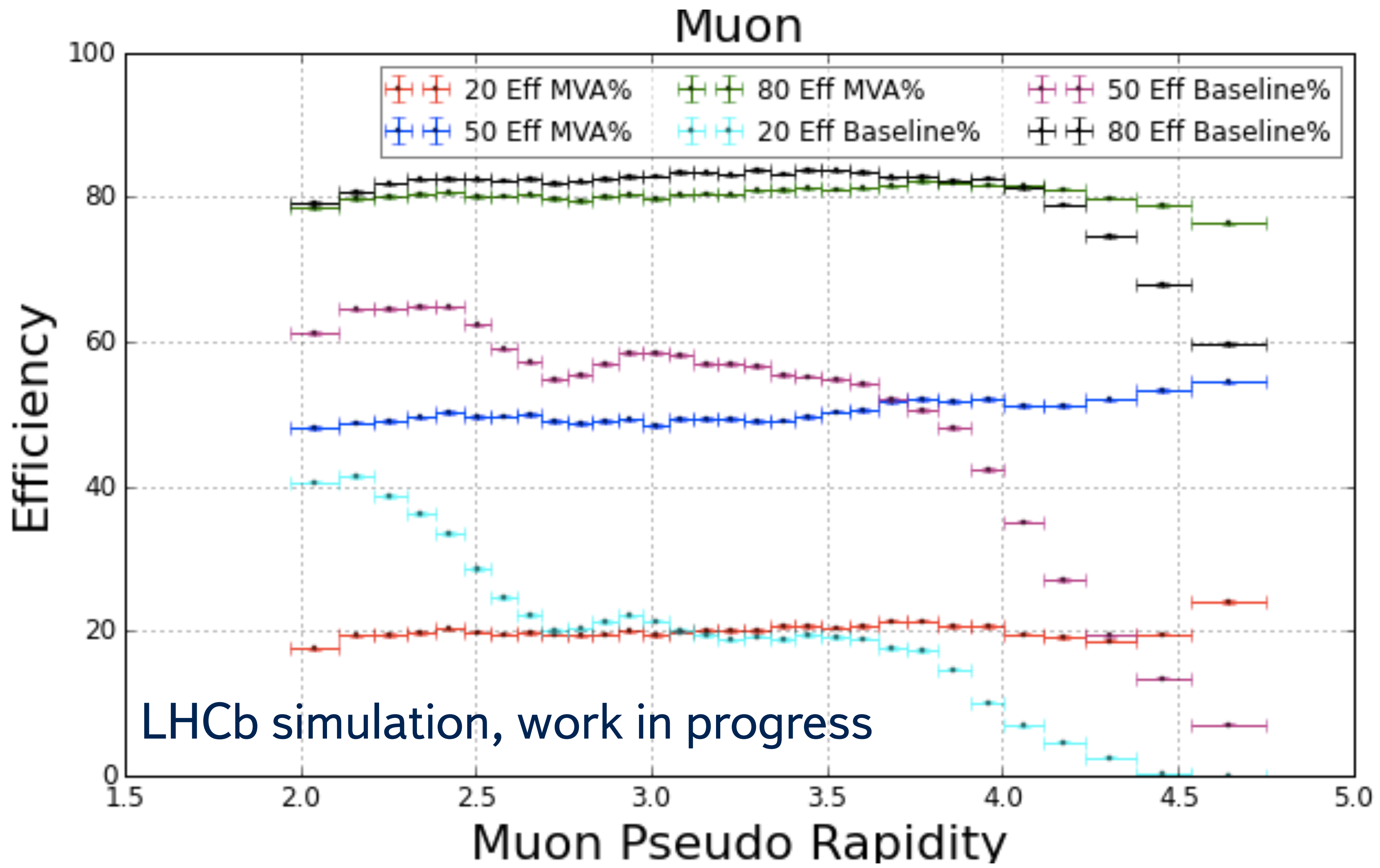
- based on gradient boosting approach
- modified loss-function to have «unflatness» that penalises for «bumps»
- https://arxiv.org/pdf/1410.4140v1.pdf

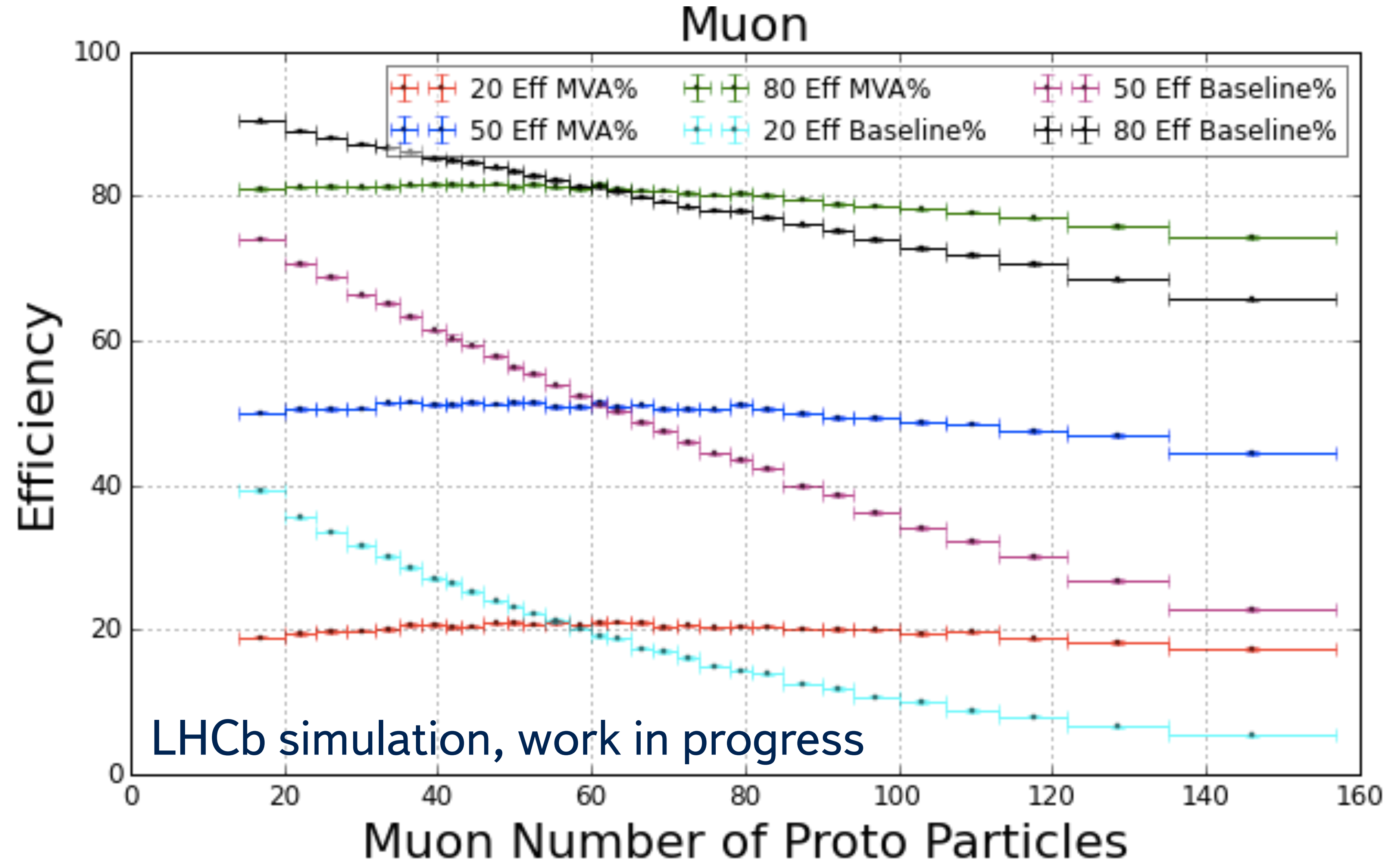


Uniform boosting provides flatness along 4 variables at once









Machine Learning Challenges & Methods

Data representation (particle traces)
Model blending/ensembling from different sub-detectors
Metric Selection

- Multiclassification? One-vs-One? One-vs-all? Accuracy? Log-loss? ROC-AUC?

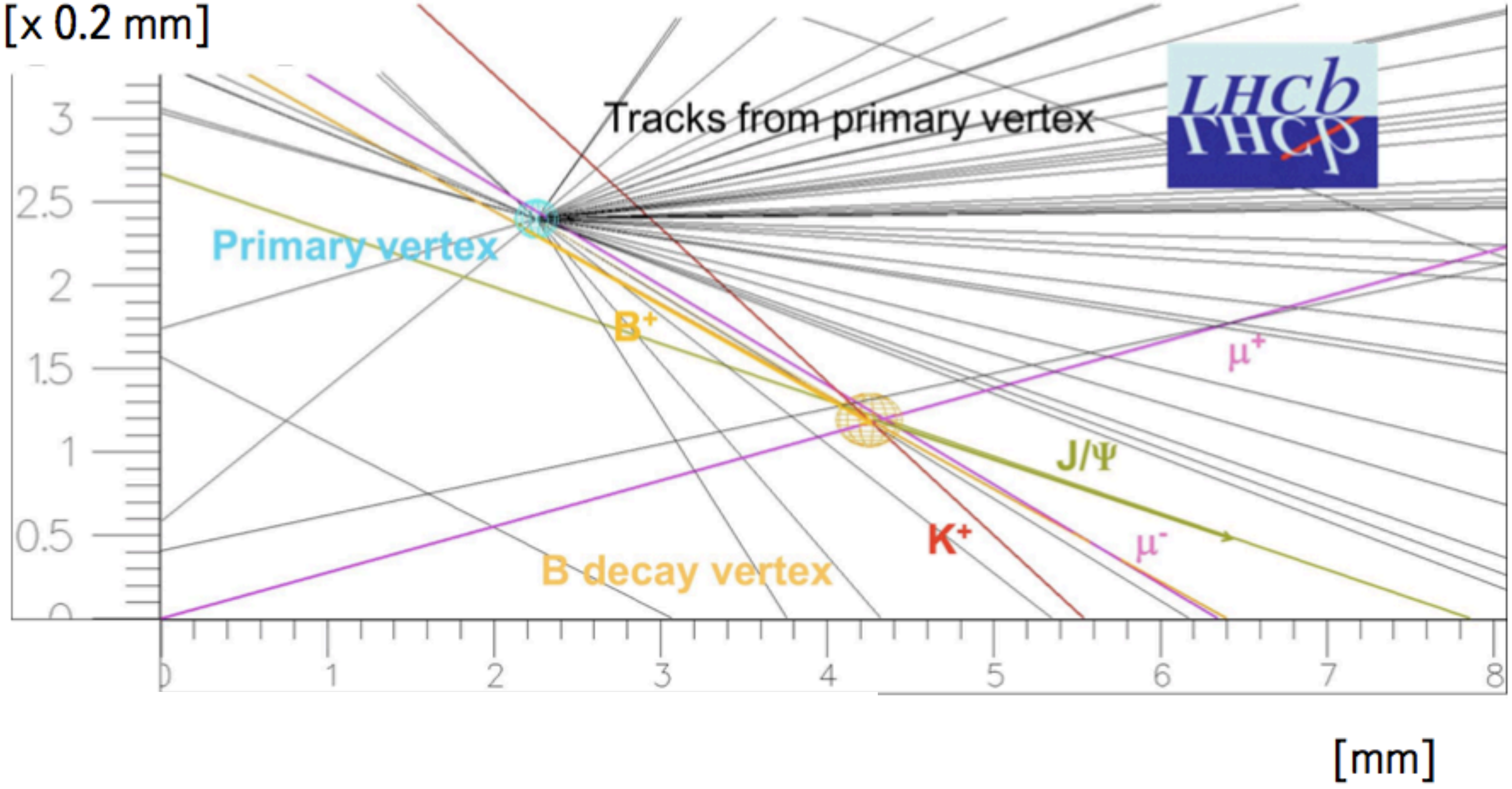
Reduce model output dependence on momentum (flatness)

Methods:

Multi-class classification
Deep NN
Advanced Boosting (altering loss function)

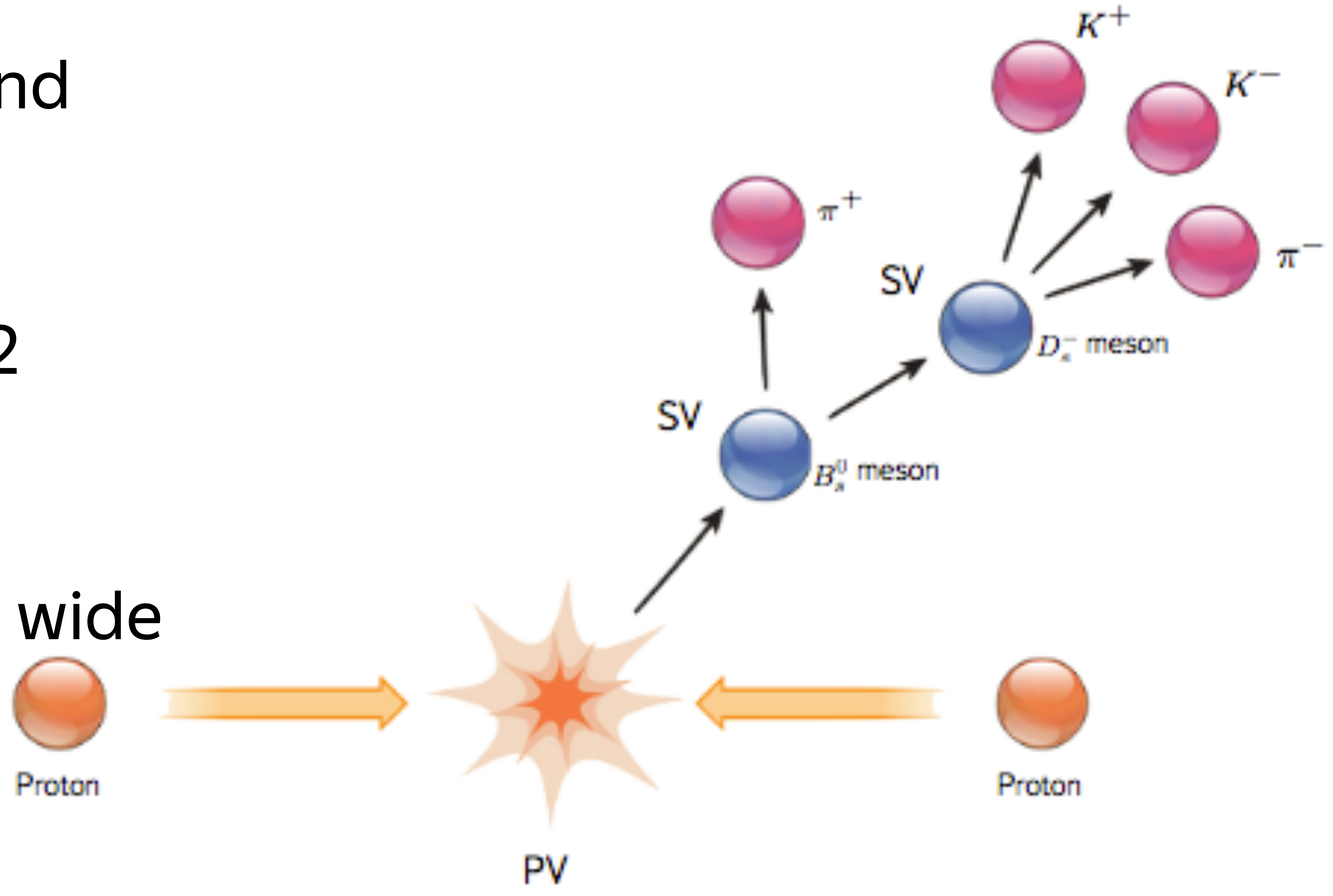
http://bit.ly/2l0yvXc

Problem 3: LHCb Topological Trigger



LHCb Topological Trigger

- Generic trigger for decays of beauty and charm hadrons;
- Part of Software trigger;
- Inclusive for any B decay with at least 2 charged daughters including missing particles;
- Look for 2, 3, 4 track combinations in a wide mass range.



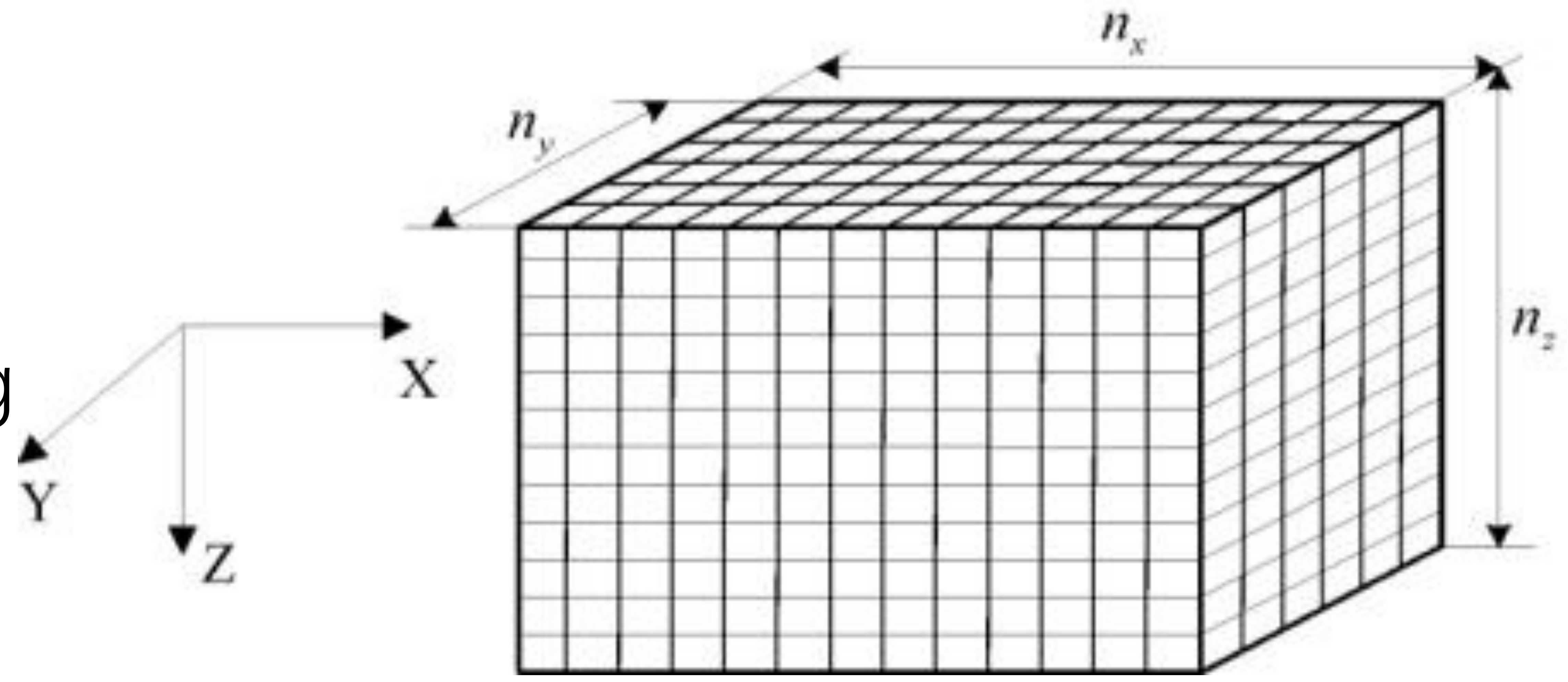
Machine Learning Challenges & Methods

- Definition of event (variable number of particles)
- Training subsample selection
- Training scheme (different decays)
- Metric selection
- Real-time demand, quality-speed trade-off

Methods

- Binary classification
- Model blending
- Feature selection
- Model speed-up

Online part using Bonsai BDT

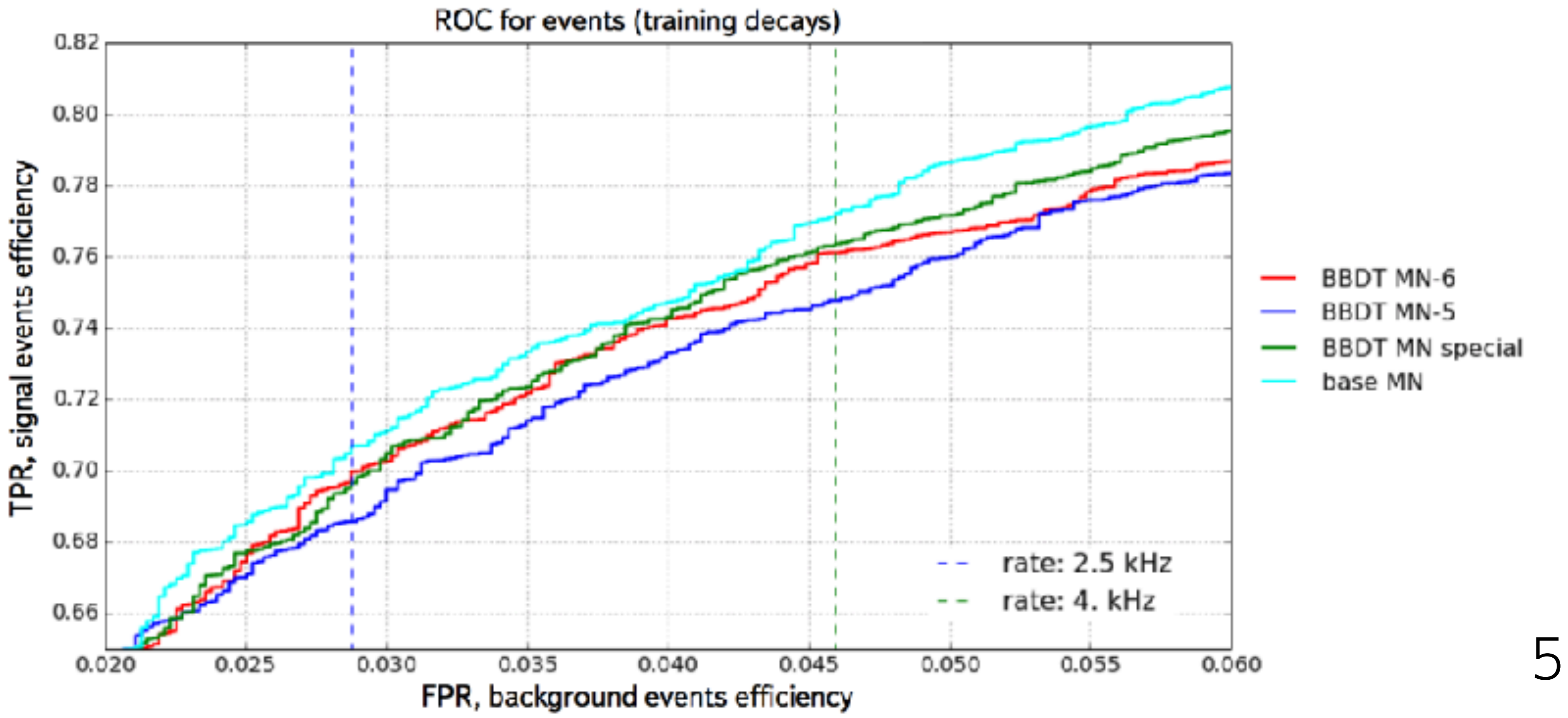


Features hashing using bins before training

Converting decision trees to
n-dimensional table (lookup table)

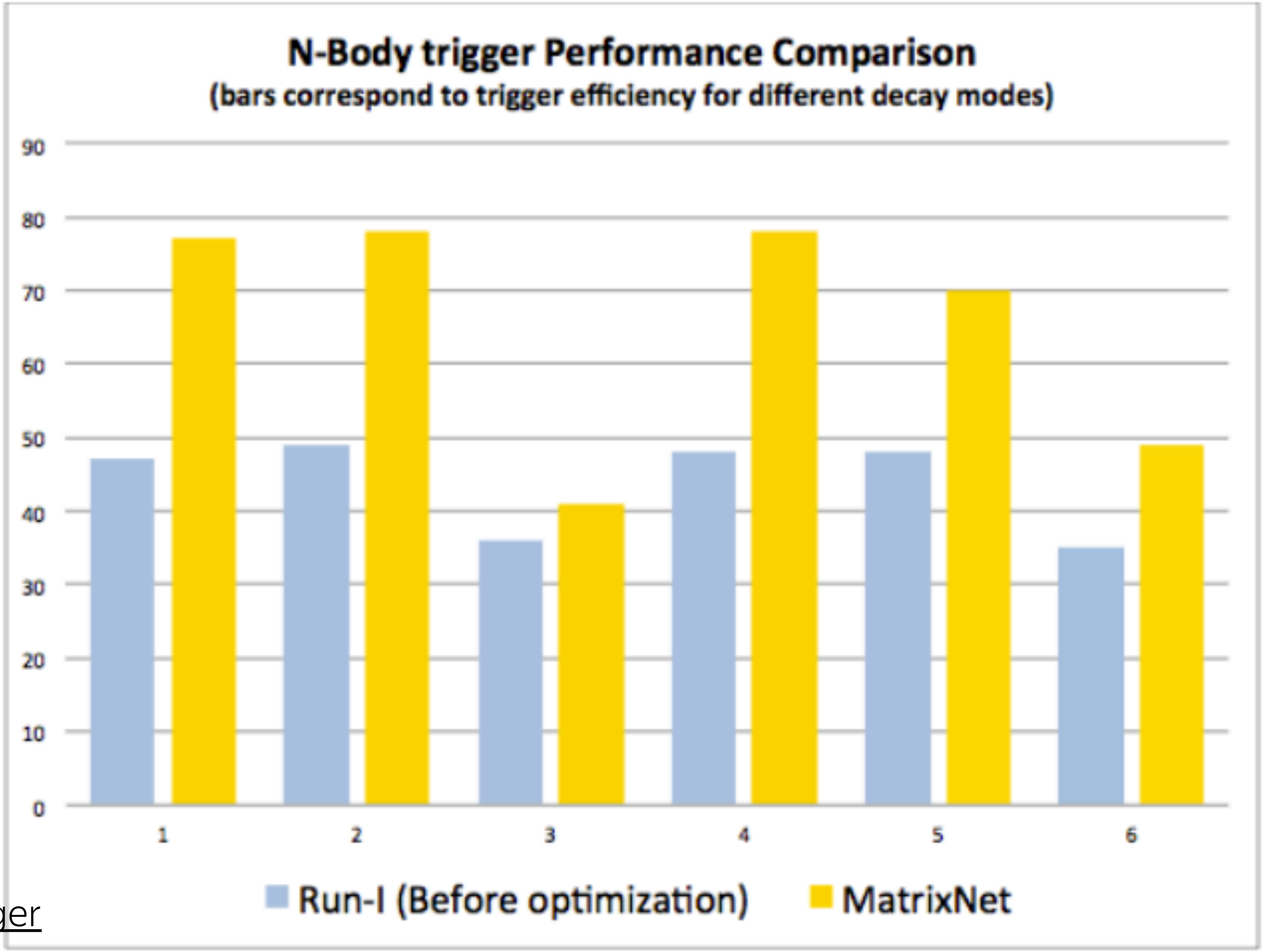
Table size is limited in RAM (1Gb), thus count of bins for each
features should be small (5 bins for each of 12 features)

Discretisation reduces the quality by ~1%



Trigger optimisation results

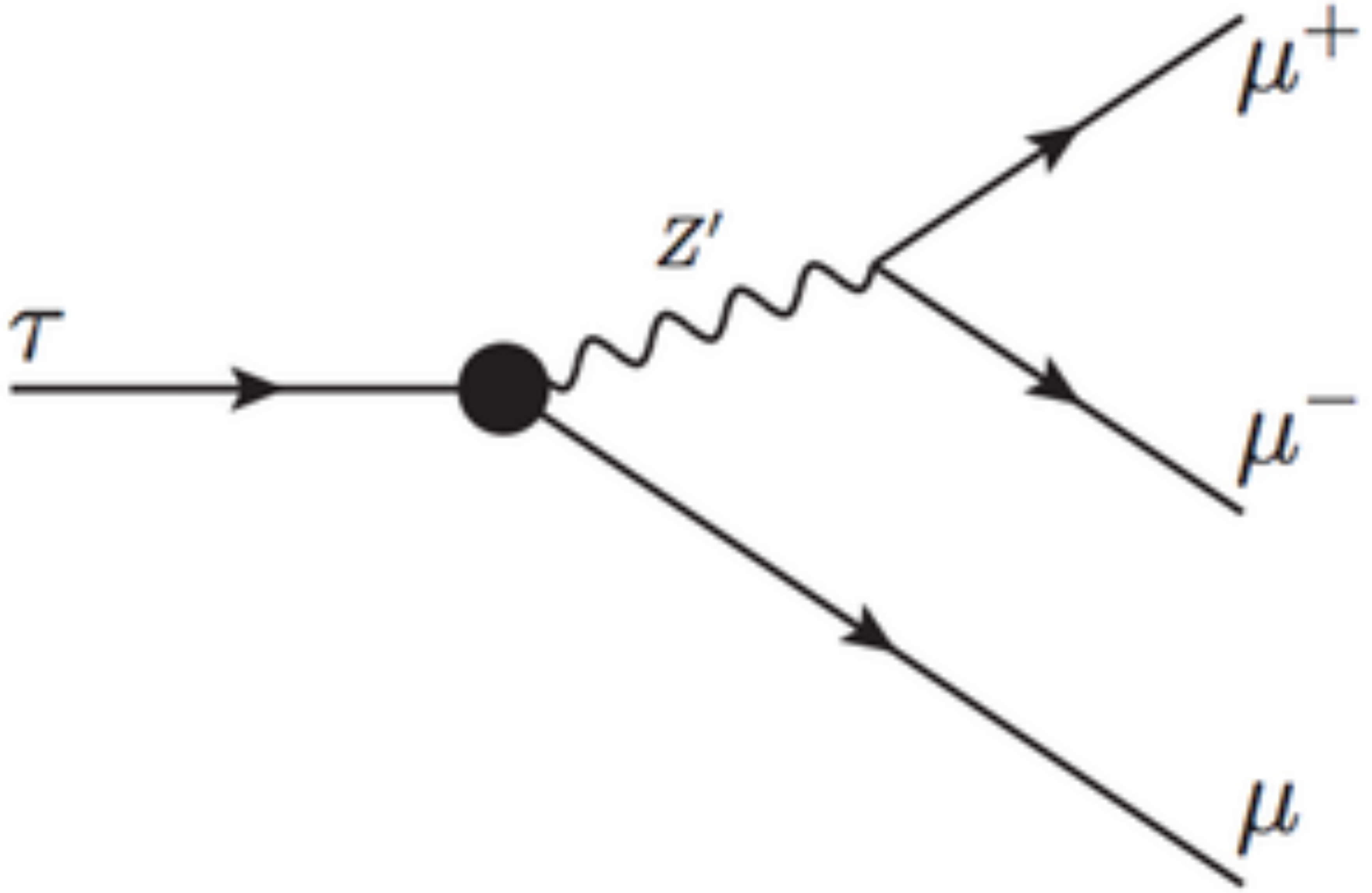
1. $B^0 \to K^*[K^+\pi^-]\mu^+\mu^-$
2. $B^+ \to \pi^+K^-K^+$
3. $B_s^0 \to D_s^-[K^+K^-\pi^-]\mu^+\nu_\mu$
4. $B_s^0 \to \psi(1S)[\mu^+\mu^-]K^+K^-\pi^+\pi^-$
5. $B_s^0 \to D_s^-[K^+K^-\pi^-]\pi^+$
6. $B^0 \to D^+[K^-\pi^+\pi^+]D^-[K^+\pi^-\pi^-]$



http://arxiv.org/abs/1510.00572

https://github.com/yandexdataschool/LHCb-topo-trigger

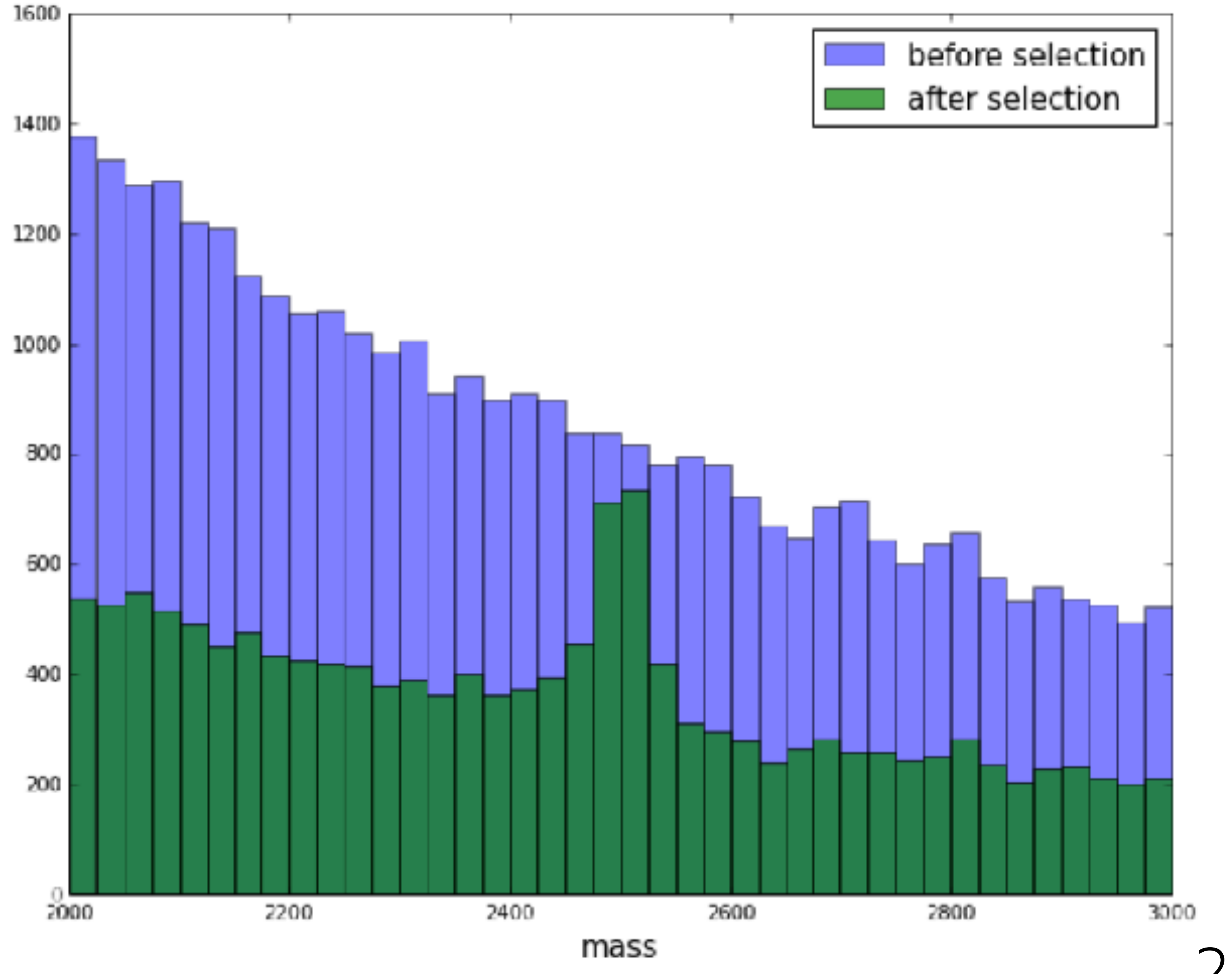
Problem 4: $\tau^- \rightarrow \mu^- \mu^+ \mu^-$



- Search for very-very rare decay (10^{-40} according to standard model);
- Current sensitivity of LHCb is about 10^{-10};
- Data sample is selected from what has been collected by triggers;
- Sits on the top of data analysis chain (after tracking, triggers, preselections, etc), so data and results should be treated under certain assumptions.

ML-flavoured sub-problem

- Every decay candidate described by set of high-level features;
- Classification: differentiate decays containing signal from others (background);
- **Simulated** sample of signal, **real** sample for background but , model should not pick simulation-specific information;
- Trained model output should not correlate with mass of mother particle.



Results:

- http://arxiv.org/abs/1409.8548
- https://www.kaggle.com/c/flavours-of-physics
- Data Doping: http://bit.ly/2lJSEzU
- https://github.com/yandexdataschool/hep_ml/

ML Challenges

Constrained classification:

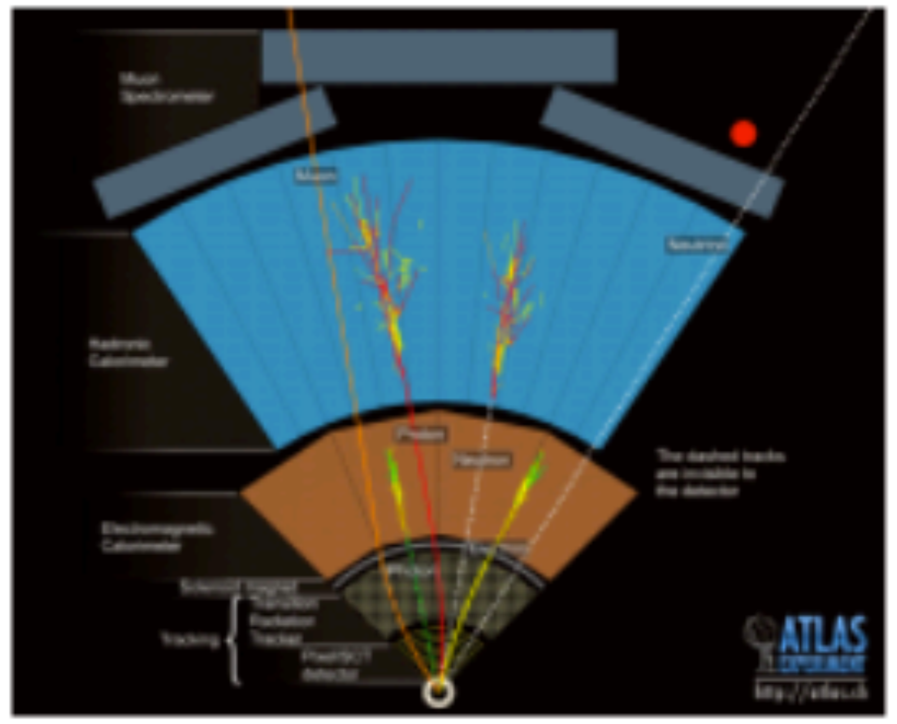
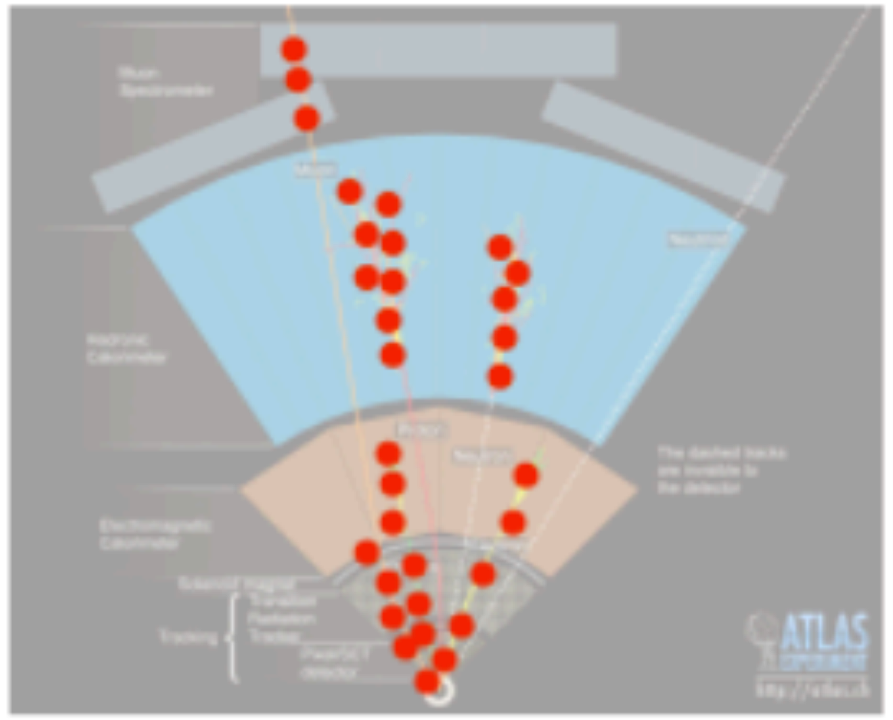
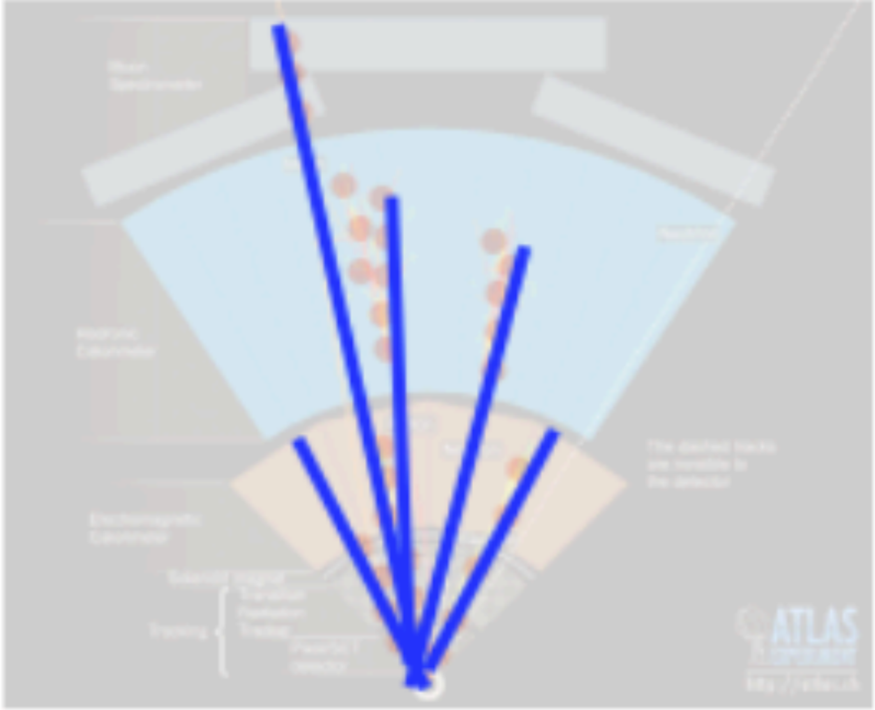
- flatness;
- signal/background vs MC/real-data check.

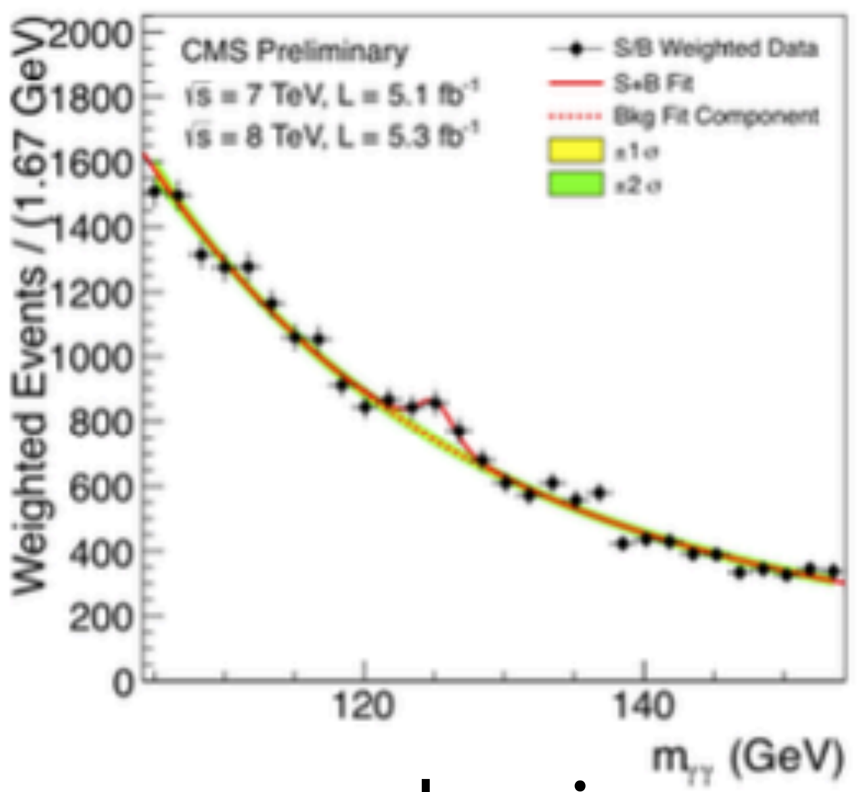
Metric?

- Prefer classifier with higher number of true positive with lowest possible false positive number;
- Chosen metric: constrains + weighted ROC AUC.

Problem, HEP	Experiment	ML methods
Particle Identification	LHCb	DNN, classification, advanced Boosting
MC generation optimization	SHiP	GP, model calibration, non-convex optimisation
Tracking	LHCb, SHiP, COMET	Tracking, Clustering, real-time
Jet identification	LHCb	CNN, multi classification
Triggers	CRAYFIS	Enhanced Convolutional Neural Nets (CNN)
Data modelling	CRAYFIS	Generative Adversarial Nets (GAN)
Anomaly Detection, data certification	LHCb	Time Series, Binary classification
Triggers	LHCb	Classification, real-time
Detector optimisation	SHiP	Surrogate modelling

HEP Feature Engineering down to discovery



hits | clustering | tracking | selection | analysis

~10^7 → ~10^4 → ~50 → ~10 → ~1

10 ms | 100 ms | 1 week-1 month | 10 month

Could it be automated a bit more?

Going deeper

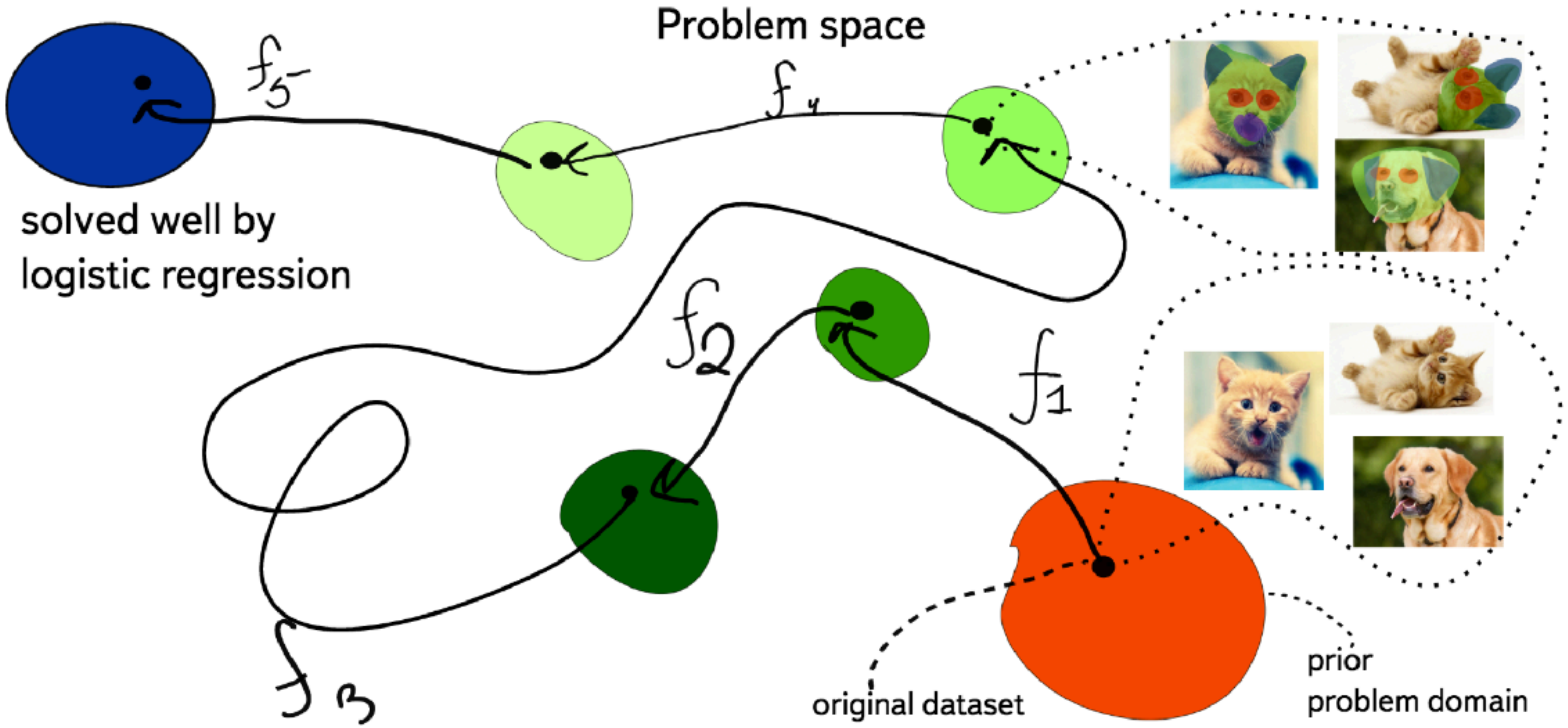
How to train machine to recognise a kitten?

```
[[ 22  25  28  32  29 ...,  58  36  35  34  34]
 [ 26  29  30  31  36 ...,  65  38  42  41  42]
 [ 27  28  31  30  40 ...,  84  58  51  52  44]
 [ 27  26  27  29  43 ...,  90  70  60  57  43]
 [ 20  26  28  28  31 ...,  83  73  62  52  45]
 ...,
 [173 187 180 183 184 ..., 170 227 244 219 199]
 [193 199 194 188 185 ..., 181 197 201 209 187]
 [175 177 156 166 171 ..., 226 215 194 185 182]
 [161 159 160 187 178 ..., 216 193 220 211 200]
 [178 180 177 185 164 ..., 190 184 212 216 189]]
```

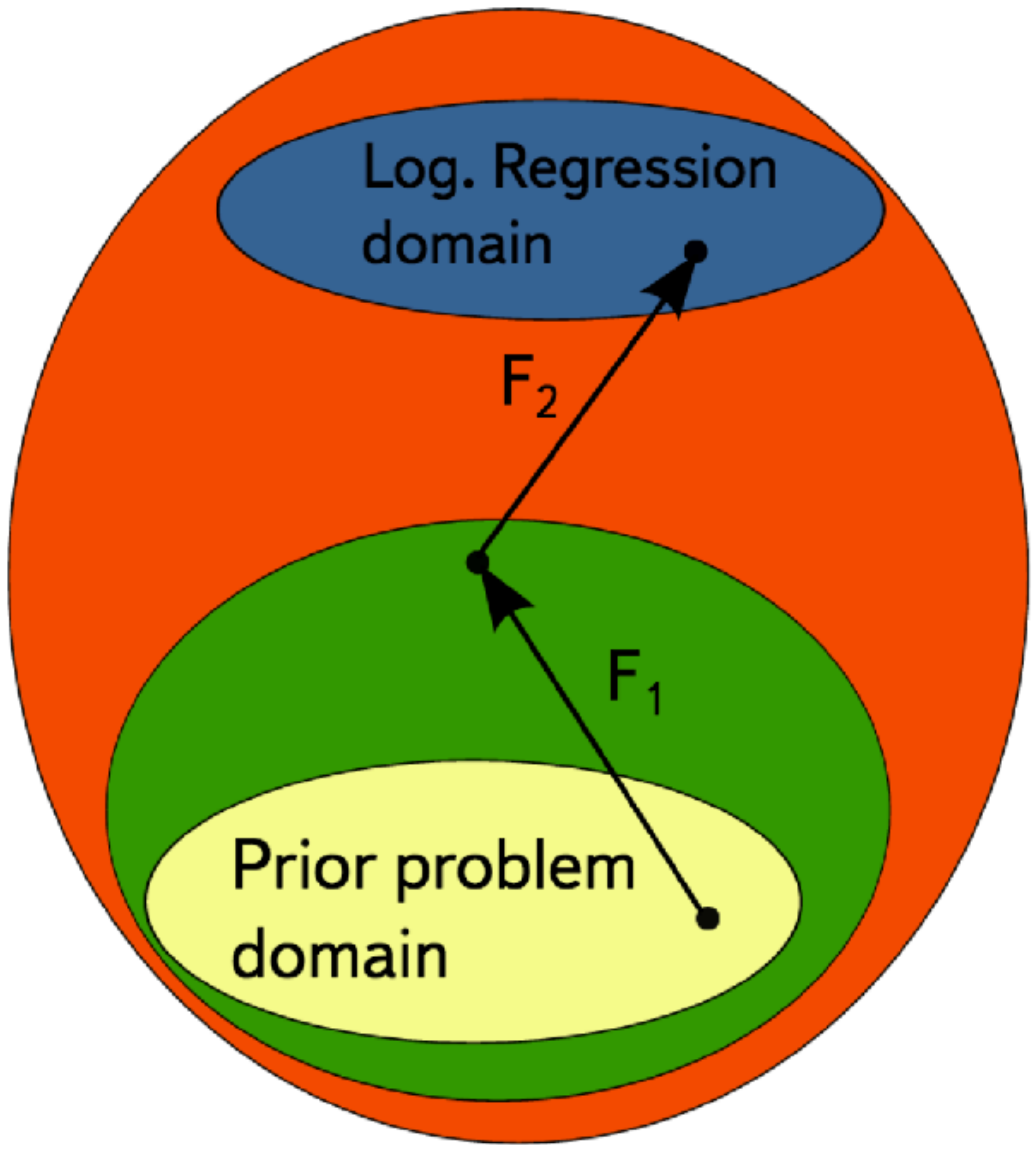
Hand-made feature engineering

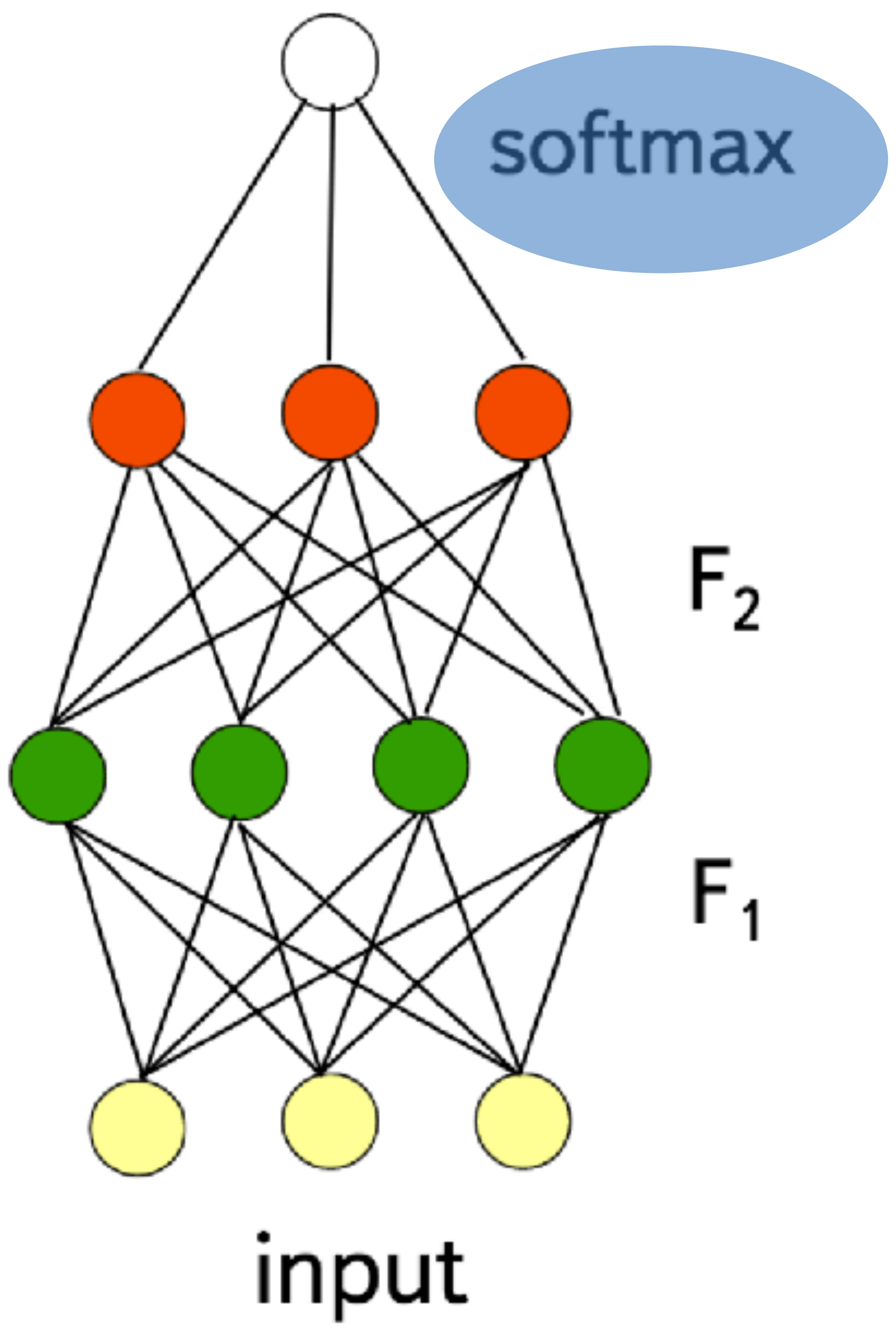
Traditional approach:

- edge detection;
- image segmentation;
- fit nose, ears, eyes;
- average, standard deviation of segment color;
- fluffiness model;
- kitten's face model;
- logistic regression.



Deep learning learns it from the data





Approach comparison

Hand-made:

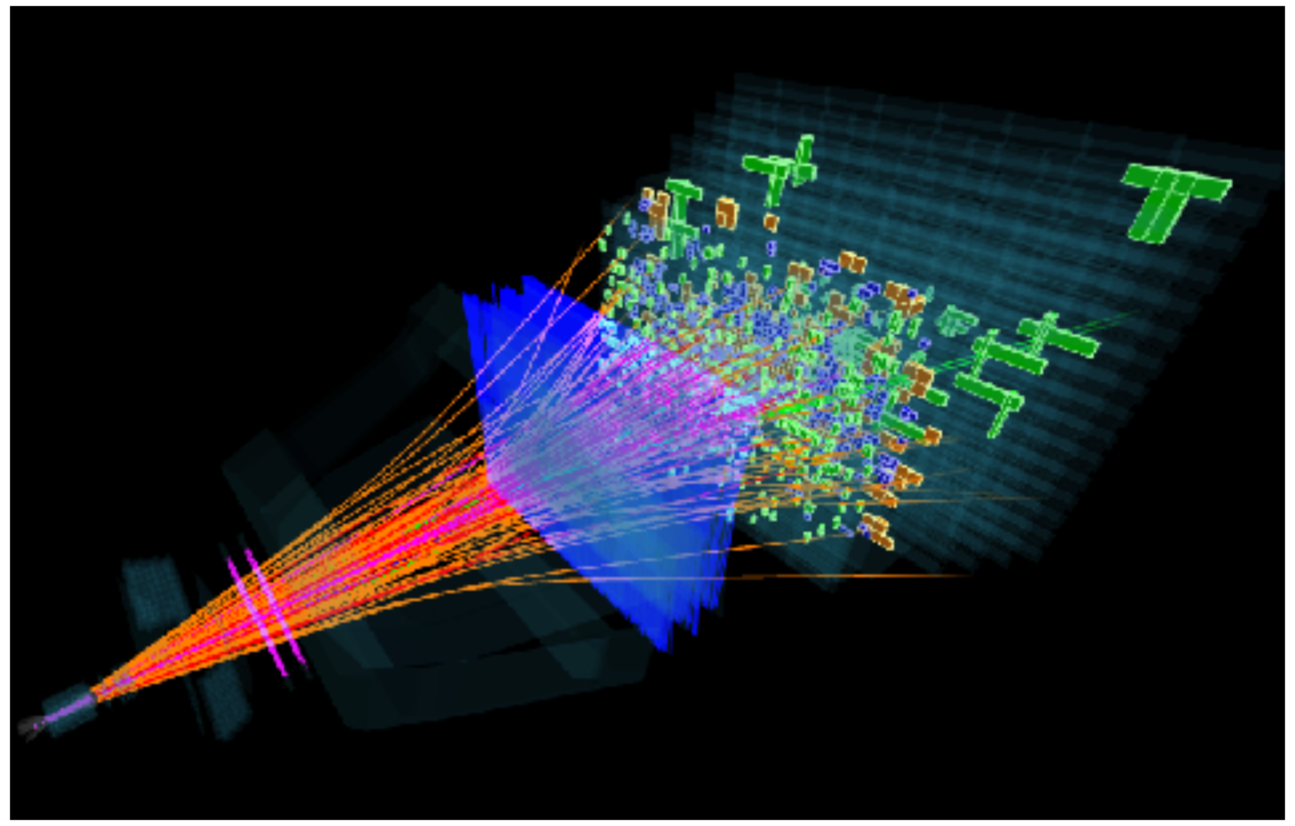
- edge detection;
- image segmentation;
- fit nose, ears, eyes;
- average,
  standard deviation
  of segment color;
- fluffiness model;
- kitten's face model;
- logistic regression.

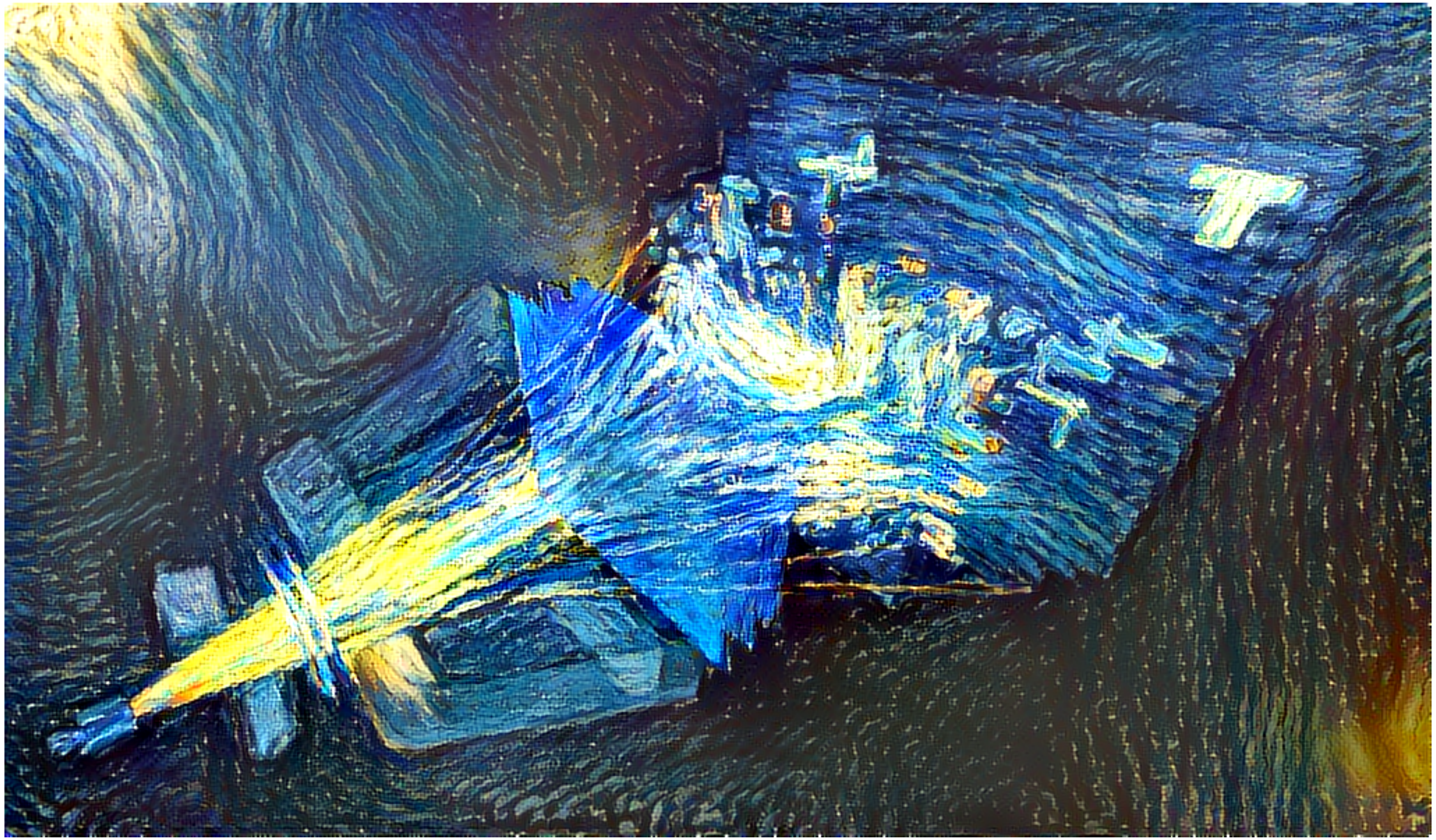
Deep Learning-way:

- non-linear transformation;
- another non-linear transformation;
- non-linear transformation, again;
- non-linear transformation, and again;
- non-linear transformation (why not?);
- logistic regression

Allows for exchanging excess of data to more generic way of feature/ transformations description and in turn helps dealing with much harder stuff.

How Deep Learning can be applied?





https://commons.wikimedia.org/wiki/File:Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg

https://github.com/jcjohnson/neural-style

Deep Learning application examples for HEP

Jet flavour identification:

- https://arxiv.org/abs/1407.5675 - CNN, Josh Cogan et al;
- https://arxiv.org/abs/1603.09349 - DNN for jets, Pierre Baldi et al;
- https://arxiv.org/abs/1701.05927 - GAN for jets, Luke de Oliveira et al;
- https://arxiv.org/abs/1702.00748 - RNN for jets, Gilles Louppe et al;

Ultimate application:

- Design detector/experiment D for X (Dark Matter, Sterile Neutrino, etc), so $P(X|D) \rightarrow max$.

In more details…

- Challenges on Kaggle:
  «HEP triggers», https://inclass.kaggle.com/c/data-science-hep-triggers
  «Higgs Boson», https://www.kaggle.com/c/Higgs-boson
  «Flavours of Physics», https://www.kaggle.com/c/flavours-of-physics;
- YSDA Course «Machine Learning for High Energy Physics»;
  Coursera «Advanced Machine Learning» Specialisation
  to be launched in 2017;
- Summer Schools:
  MLHEP 2015, 2016, http://bit.ly/mlhep2015, http://bit.ly/mlhep2016,
  MLHEP 2017 - http://bit.ly/mlhep2017 , Reading UK, 17-23 Jul.

Conclusion

Machine Learning is a great tool for exceeding expectations:

- rooted in Math (statistics, numerical optimisation, computer science);
- lots of tools and approaches with various advantages and limitations;
- to great extent is an art (metric selection, expressing problem assumptions in features/transformations, data handling, uncertainties evaluation);
- can be mastered through practice.

LHC was designed as international physics laboratory. We see it is as rich source of interesting challenges that can be addressed by Machine Learning.

Thank you for attention!

anaderi@yandex-team.ru

Special Thanks to

Tatiana Likhomanenko
Fedor Ratnikov
Denis Derkach
Maxim Borisyak
Mikhail Hushchyn
and all YSDA research team for helping crafting these slides

References

«Pattern Recognition and Event Reconstruction in Particle Physics Experiments» http://arxiv.org/abs/physics/0402039

«Pattern recognition» http://bit.ly/28KWfct

«Pattern recognition in HEP» http://bit.ly/28LUPSy

«Современные методы обработки данных в физике высоких энергий» http:// www1.jinr.ru/Pepan/v-33-3/v-33-3-11.pdf

«Performance Evaluation of RANSAC Family» http://www.bmva.org/bmvc/2009/Papers/Paper355/Paper355.pdf

https://en.wikipedia.org/wiki/Kalman_filter

Schaffer, Cullen. "A conservation law for generalization performance." Proceedings of the 11th international conference on machine learning. 1994.

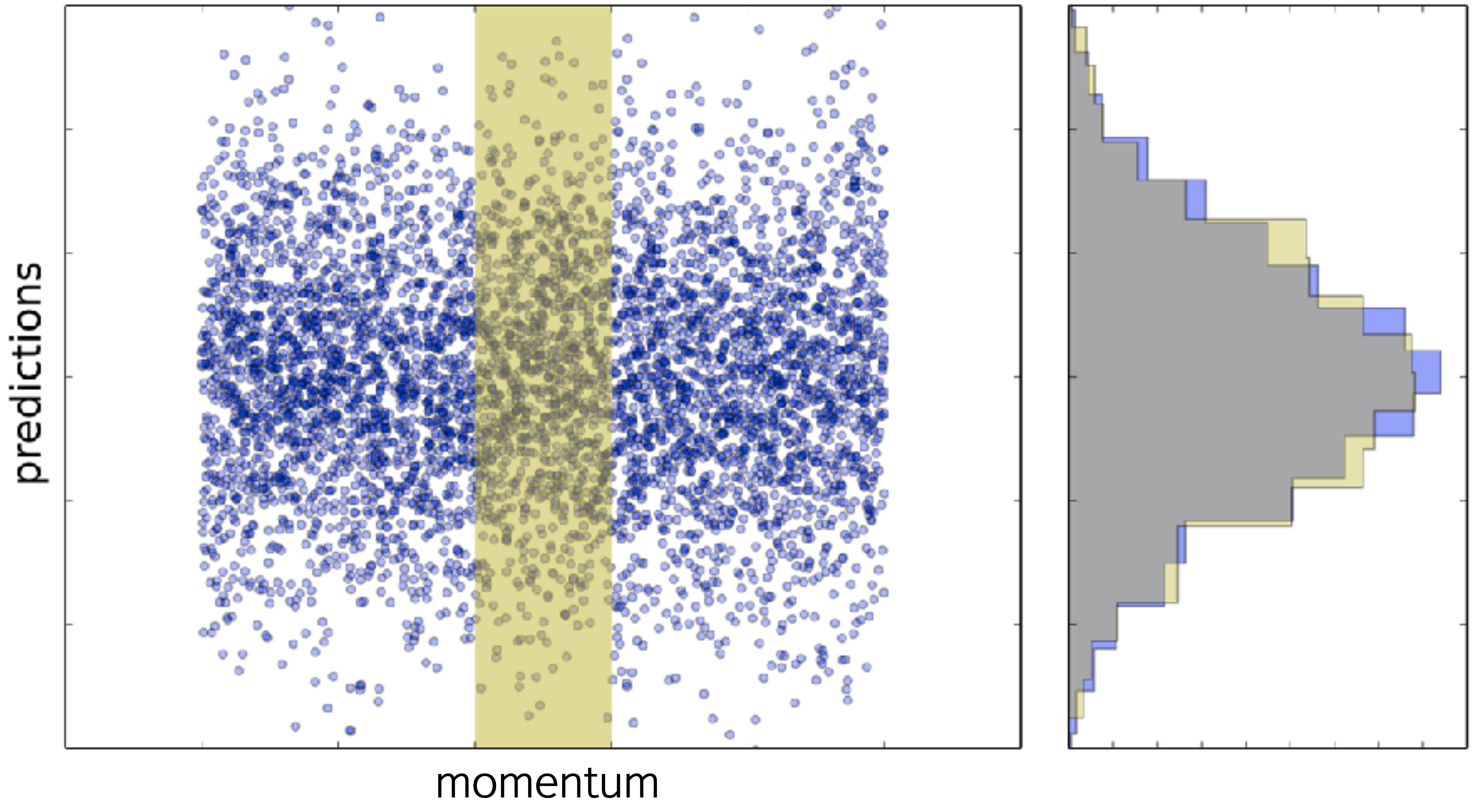
Wolpert, David H. "The supervised learning no-free-lunch theorems." Soft computing and industry. Springer London, 2002. 25-

Wolpert, David H., and William G. Macready. "No free lunch theorems for optimisation." IEEE transactions on evolutionary computation 1.1 (1997): 67-82.

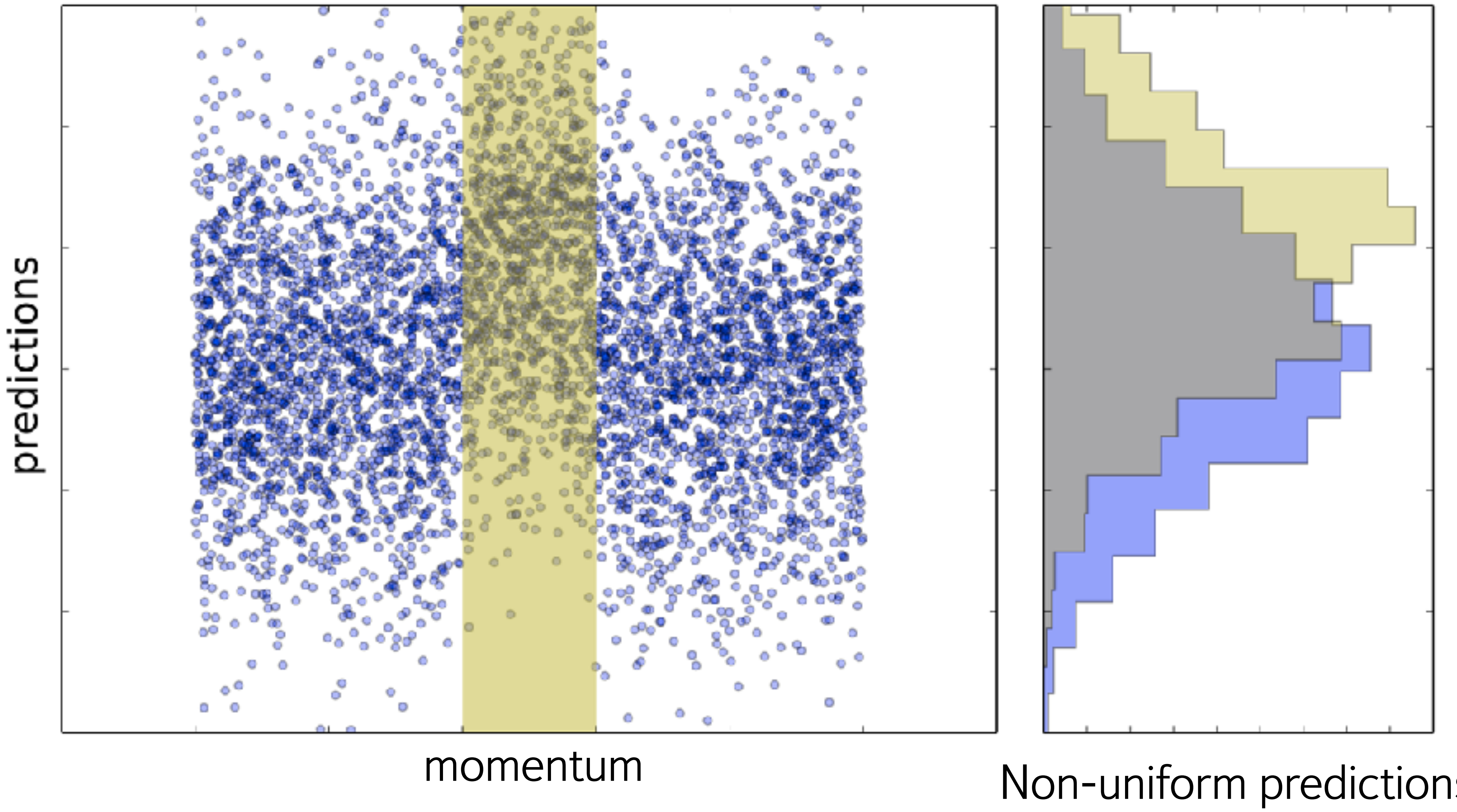
Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117

Backup Slides

Non-uniformity measure



Uniform predictions



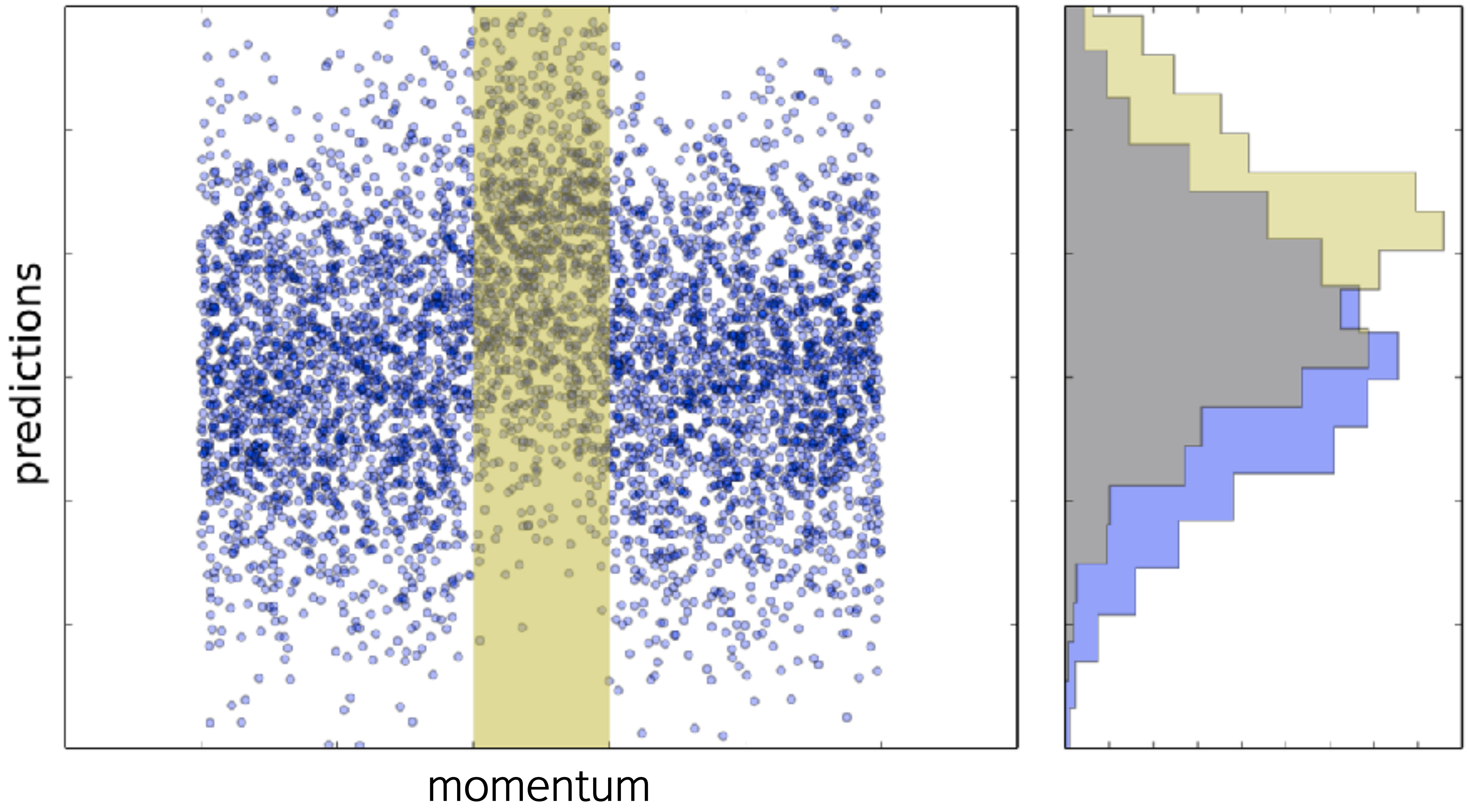
Non-uniform predictions (peak in highlighted region)

- difference in the efficiency can be detected by analyzing distributions
- uniformity = no statistical dependence between the momentum and predictions

Non-uniformity measure

Average contributions (difference between global and local distributions) from different regions in the momentum: use for this Cramer-von Mises measure (integral characteristic)

$$\text{CvM} = \sum_{\text{region}} \int |F_{\text{region}}(s) - F_{\text{global}}(s)|^2 \, dF_{\text{global}}(s)$$



Flat model construction

- Classifier optimizes a loss function during training
- Idea is to use additional loss term in the optimization problem (FL is flatness loss):

$$loss = AdaLoss + \alpha FL$$

The ***AdaLoss*** term corresponds to the classification quality, the ***FL*** term - to the flatness, α is a parameter to control the trade-off

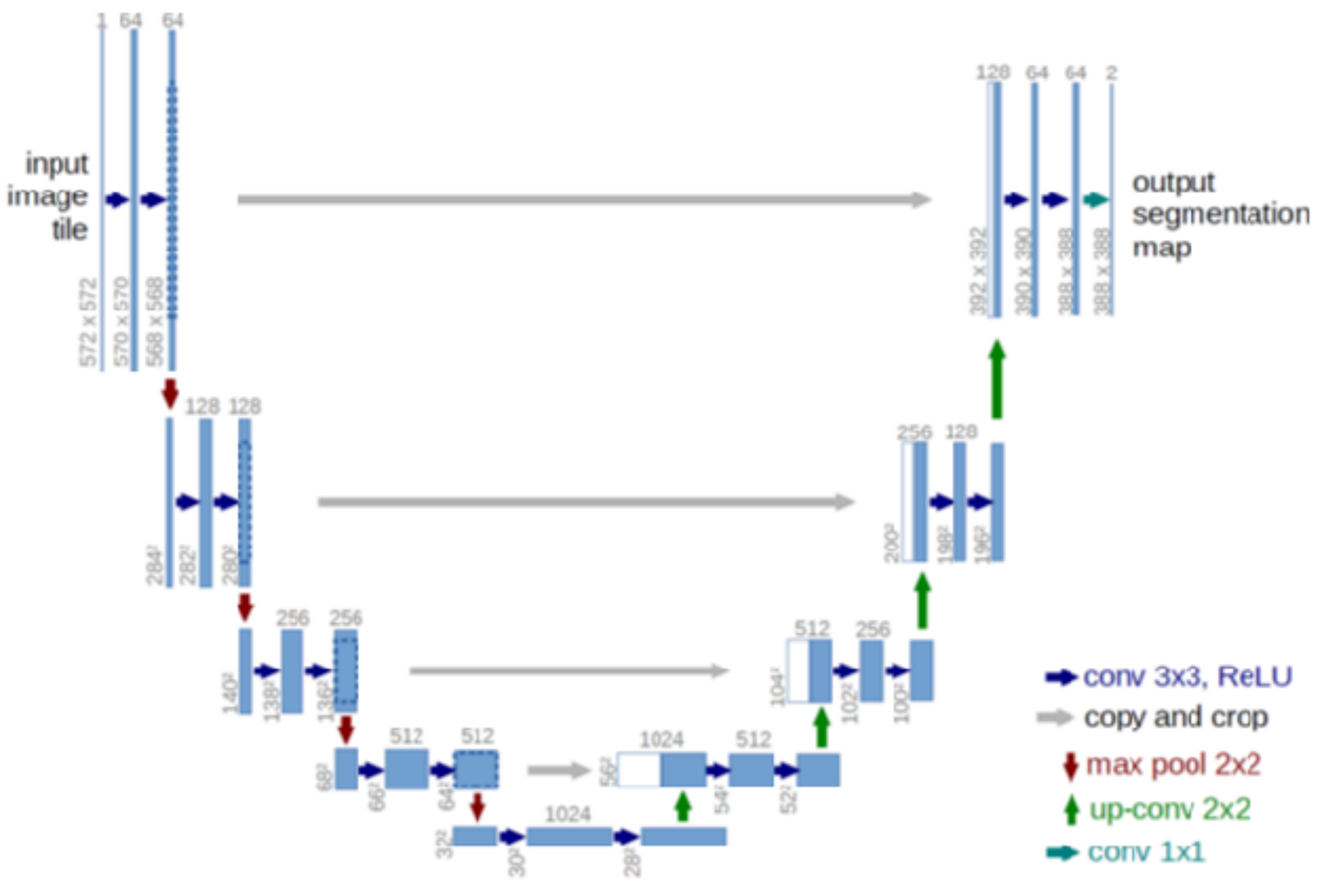
- Optimization methods use gradient of the loss
- Cramer-von Mises metric is not differentiable
- Flatness loss is similar to the Cramer-von Mises metric, but it is differentiable

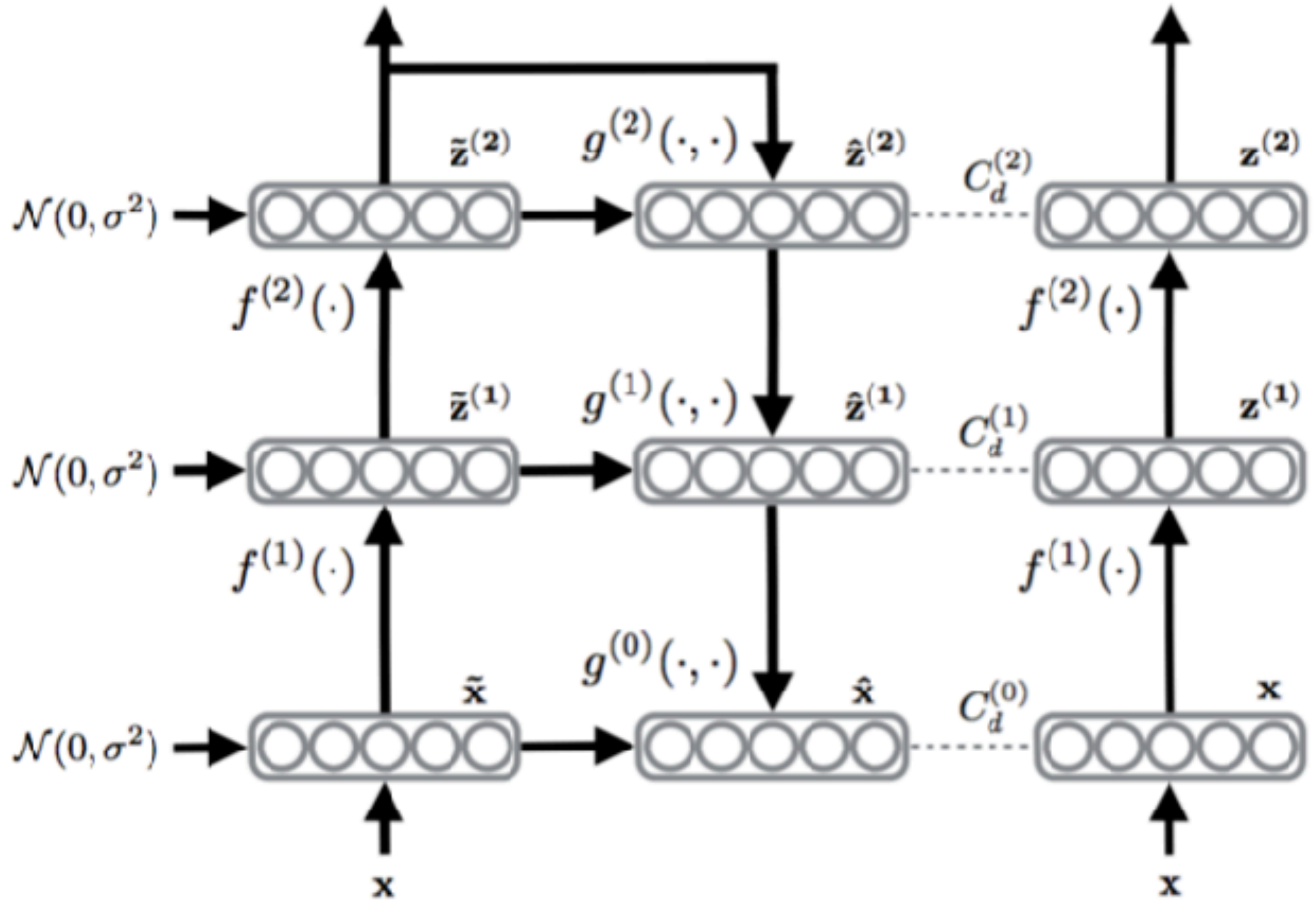
Deep Learning: hacking model

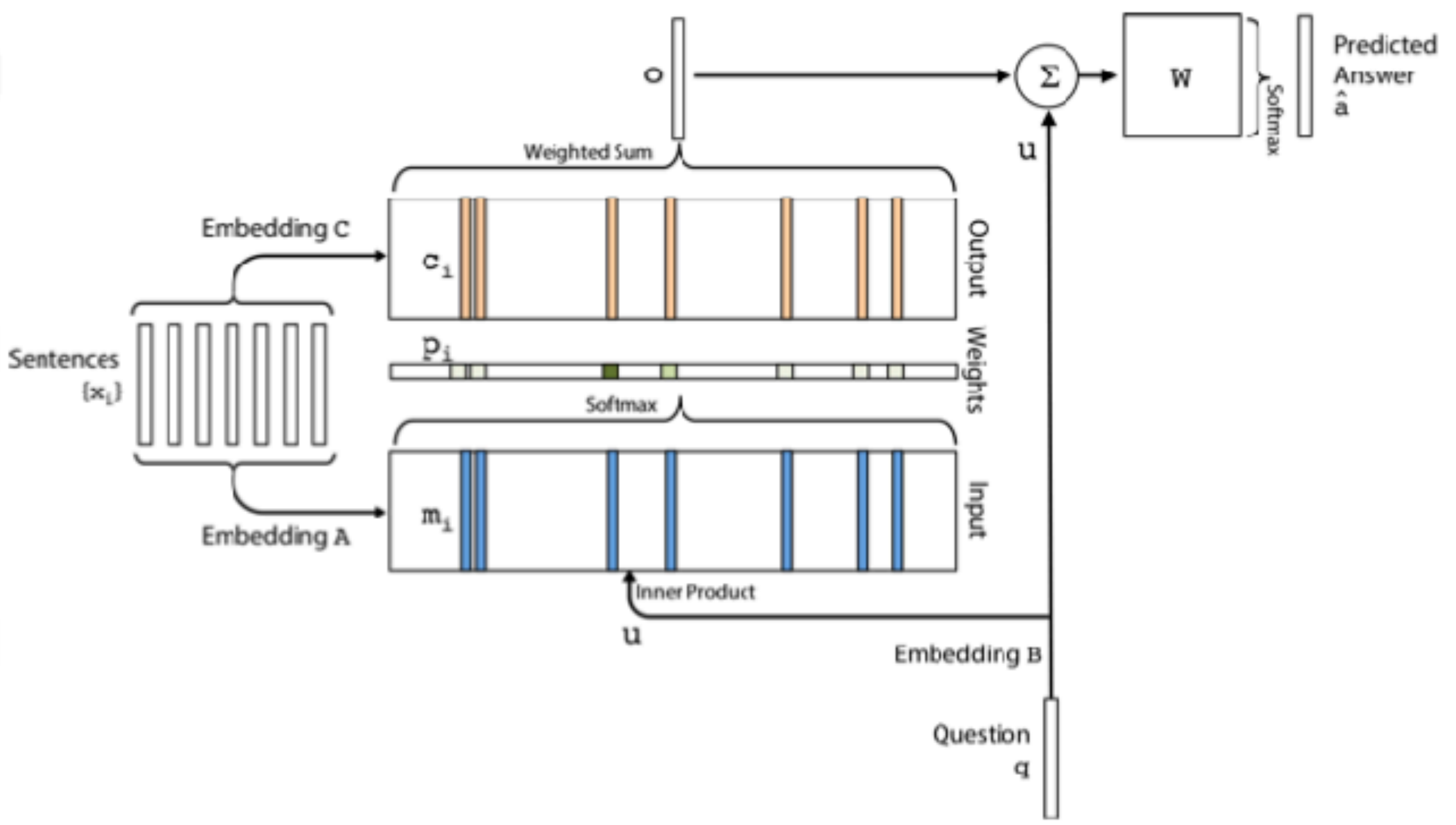
hacking layers:

> restrictions on weights: convolutions, ...;
> new operations: pooling, kernels, ...;
> specific unit behaviour: GRU, LSTM units;

combining layers, architecture of network (U-net, ladder net, end-to-end memory network):







Deep Learning: hacking model

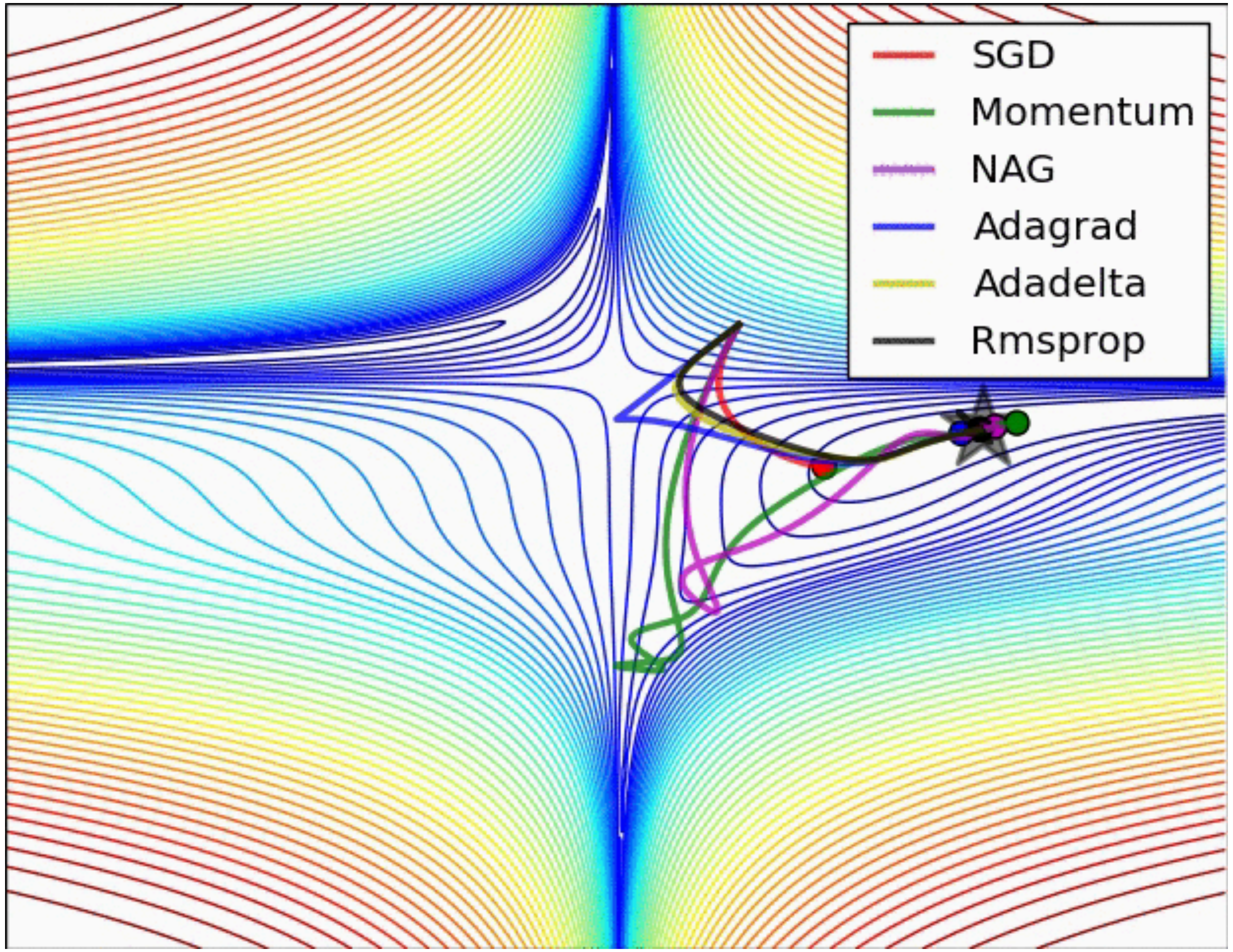
restrictions on search space:

- regularisation, e.g.:
  - $$\mathcal{L} = \mathcal{L}_{cross\text{-}entropy} + \alpha \|W\|_2^2$$
- regularisation with respect to solution W_0 of a similar problem:

$$\mathcal{L} = \mathcal{L}_{cross\text{-}entropy} + \alpha \|W - W_0\|_2^2$$

Deep Learning: Hacking Search Procedure

- SGD-like methods:
  - adam, adadelta, adamax,
  - rmsprop;
  - nesterov momentum;
- quasi-Newton methods



Deep Learning: Hacking search procedure

data augmentation:

shifts, rotations, ...:

- searching for a network that labels shifted, rotated, ... samples the same way as original ones;

random noise:

- pushing separation surface farther from samples;

interference with network:

drop-out, drop-connect:

- searching for a robust network.

Deep Learning: Hacking search procedure

hacking objectives:

introducing loss for each layer:

$$\mathscr{L} = \mathscr{L}_n + \sum_{i=1}^{n-1} C_i \, \mathscr{L}_i$$

where:

› $\mathscr{L}_i$ - loss on i-th layer.

Deeply Supervised Networks:

- searches for network that obtains good intermediate results.

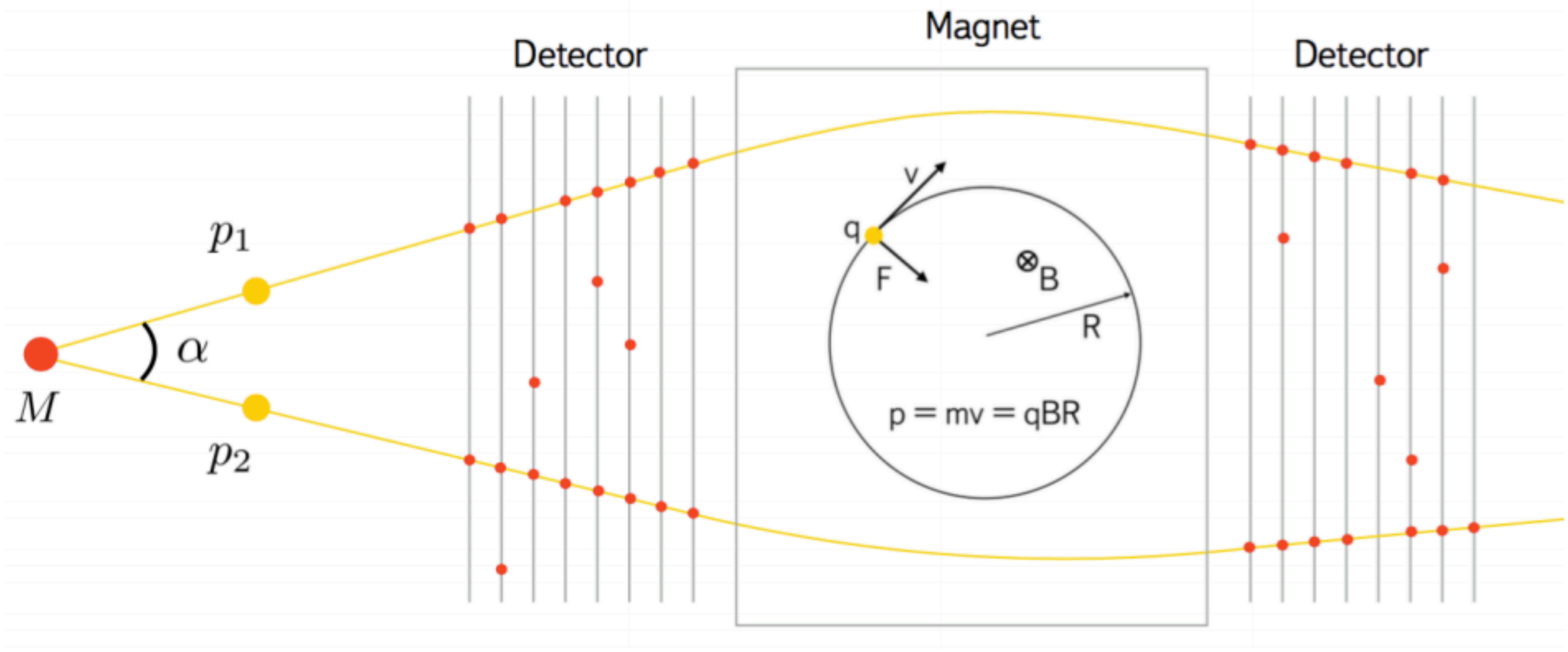
Deep Learning: Hacking initial guess

solution for a similar problem as initial guess for search;

pretraining on a similar dataset:

- unsupervised pretraining on unlabeled samples;
- supervised pretraining.

Problem X: Tracking



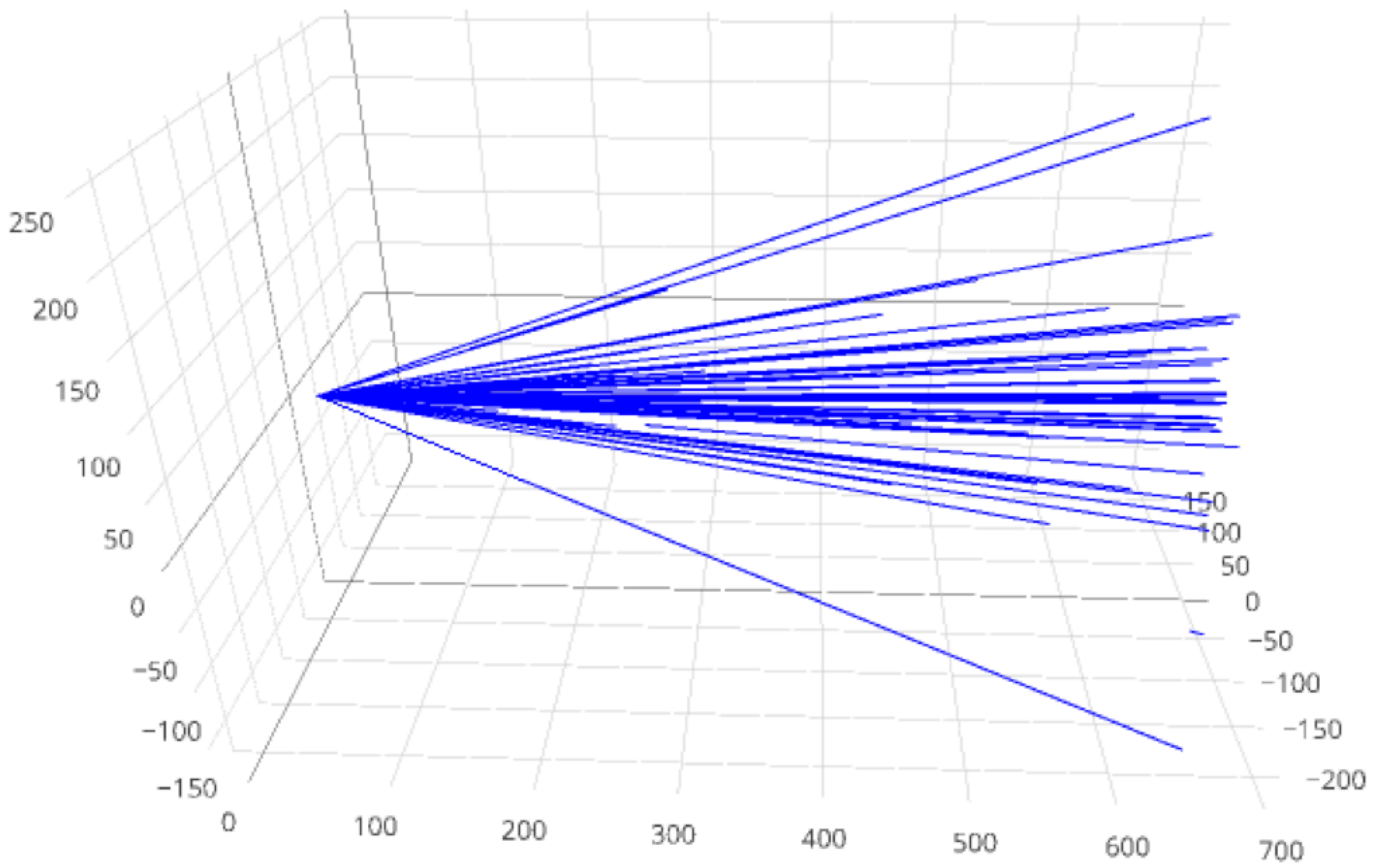
1. Make particles **tracks** from **hits** and reconstruct its **parameters**.

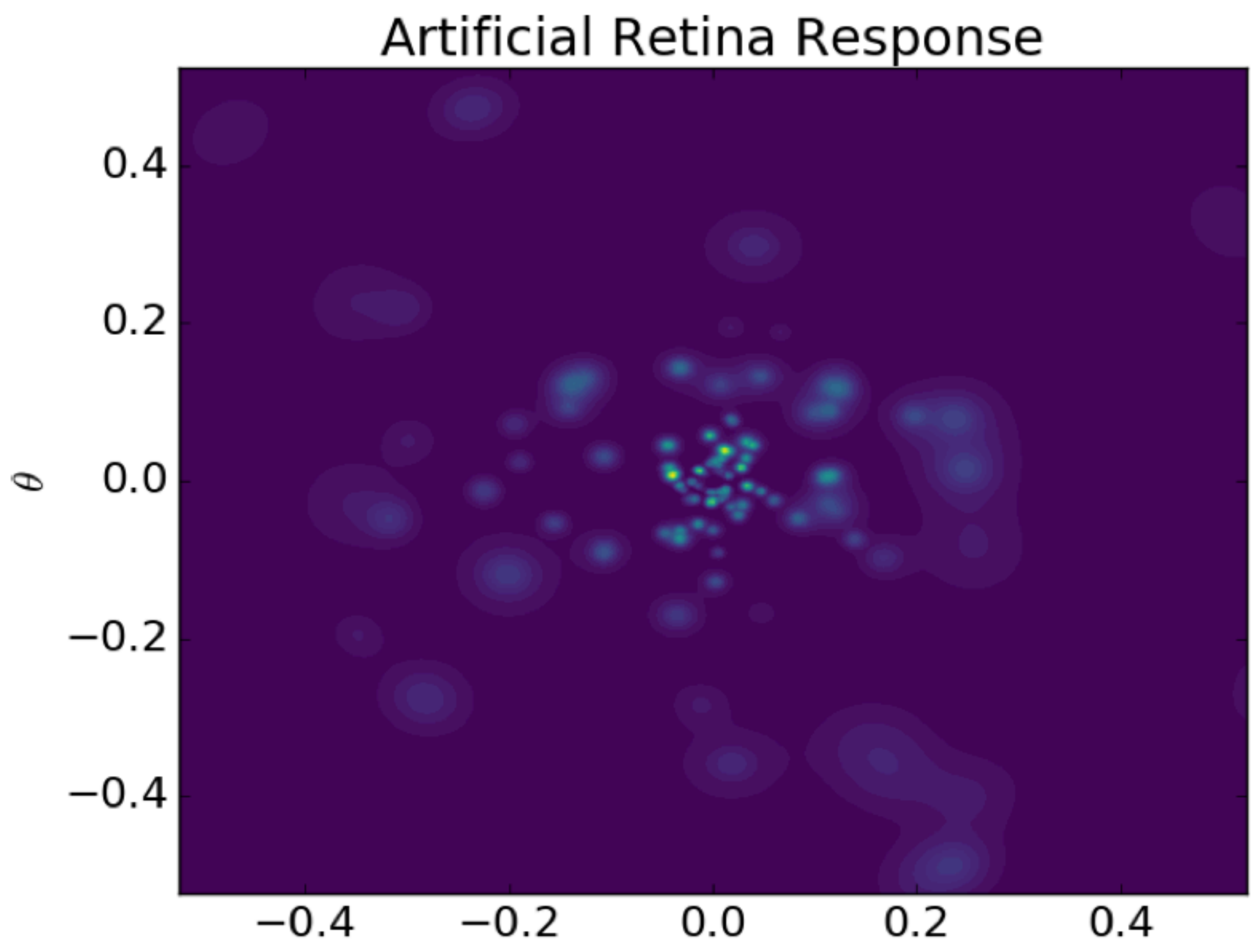
2. Combine the tracks before and after the magnet. Reconstruct full tracks. Calculate particles properties (angles, momenta, vertices, etc).

Variety of Metrics

- Track finding efficiency
- Event reconstruction efficiency
- Ghost Rate
- Clone Rate

LHCb VELO artificial retina





Example of 3D event and Artificial Retina response

$$R(\theta) = \sum_{i=1}^{N} \exp\left(-\frac{\rho^2(\theta, \mathbf{x}_i)}{\sigma^2}\right)$$

Artificial Retina

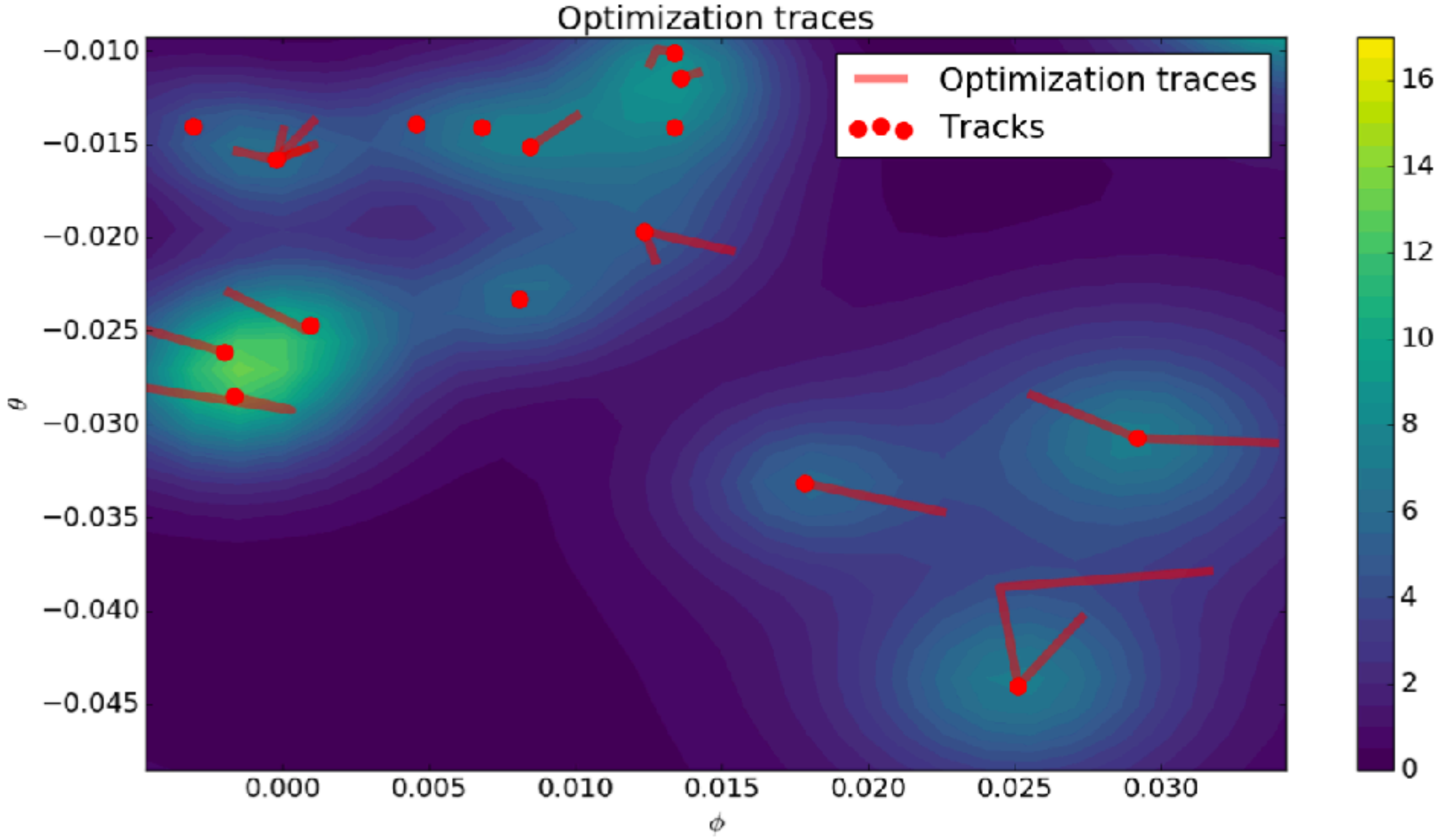
Pros:

- gradient surface
- suitable for parallelisation
- works under high track occupancy conditions
- suitable for gradient-based optimisation algorithms for finding maxima (see →)
- comparable performance (efficiency, ghost rates) to LHCb-upgrade tracking(VELOUT)

https://doi.org/10.1088/1748-0221/10/03/C03008

Cons:

- need to scan big volume of parameter space to find all local maxima



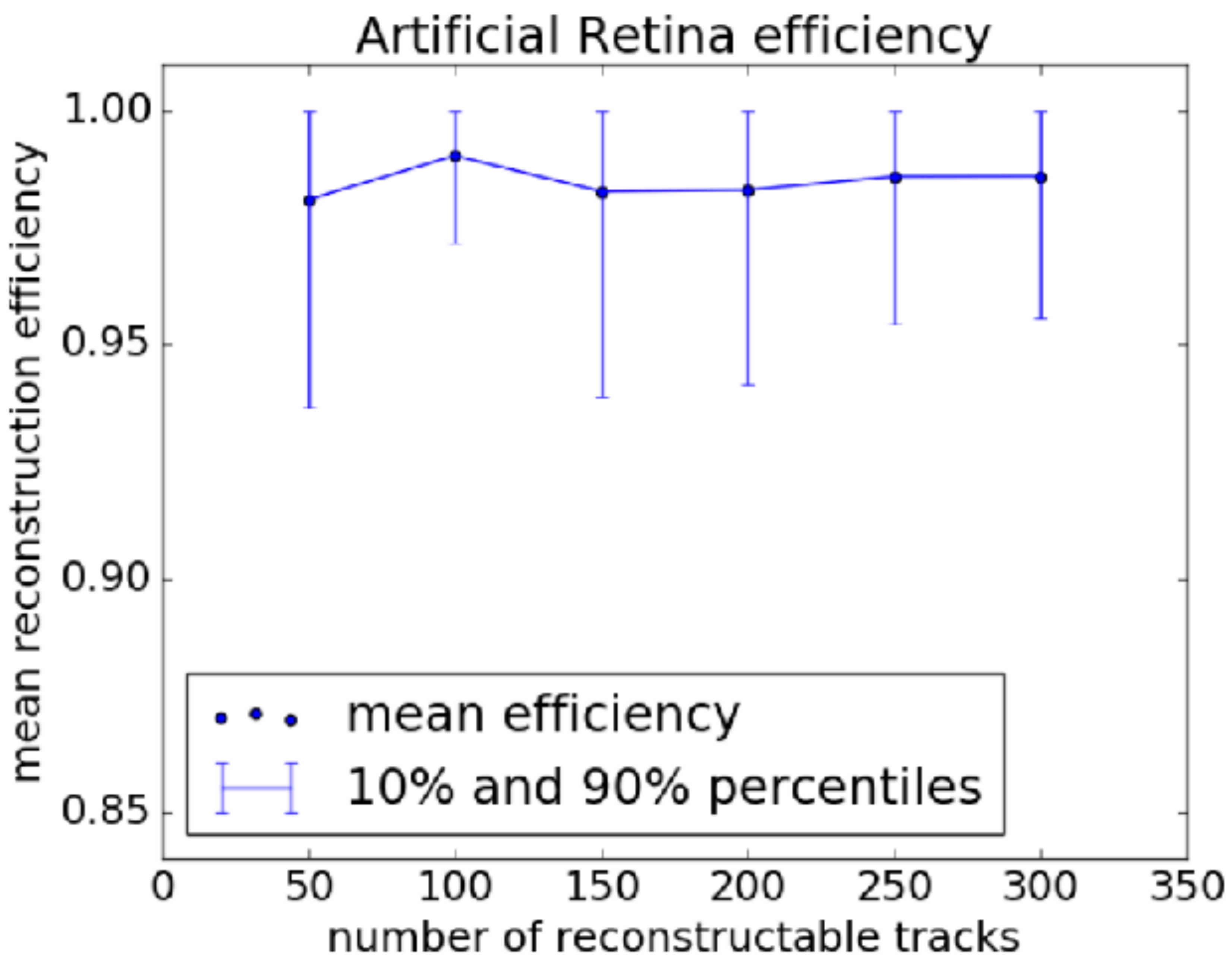
Results

simplified track model - straight line parametrised by 2 angles
detector geometry taken from LHCb upgrade TDR (CERN-LHCC-2013-021)

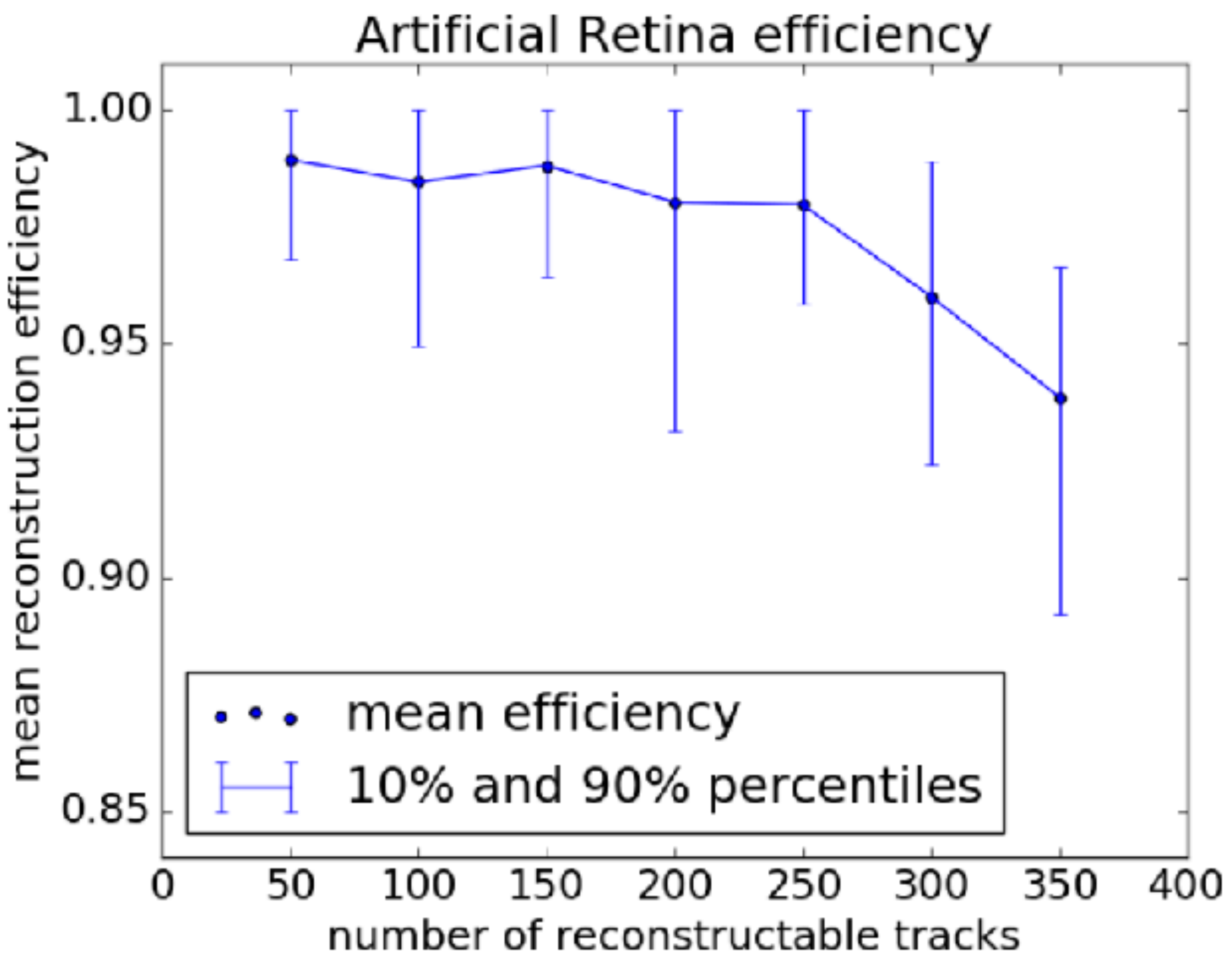
α - speedup factor wrt to grid search;

Ghost rate is strictly zero in all cases. Multiple results for the same track are merged within ϵ-radius (10^{-3} rad).

$\alpha = 1/3$



$\alpha = 1/10$



http://bit.ly/2kk7Uaq

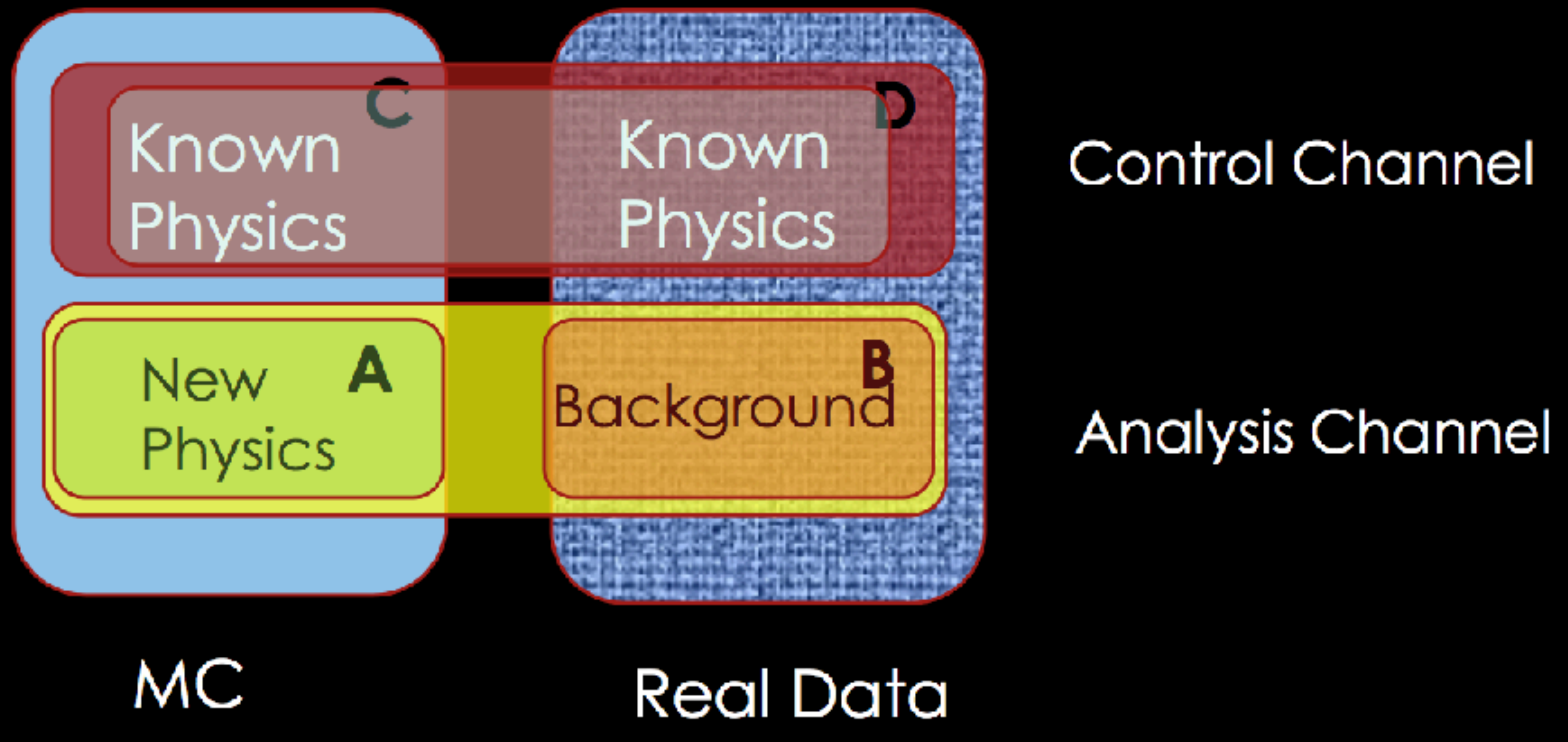
«Machine Learning» Challenges & Methods

- Metric selection:
  - trade-off between efficiency, ghost rate, clone rate
  - Hardware-imposed assumptions (straw tube, fiber scintillator, etc)
- Ideally (hyper) parameters of tracking algorithm should maximise probability of finding effect/events we are interested in;
- Implementation challenges:
  - Speed-accuracy trade-off
  - Parallelization
- Clustering (Unsupervised), RANSAC, Hough Transform, Deformable Templates, Hopfield NN, Track Following, Kalman Filter, …

BREAKING THE RULES: DATA DOPING

- Recipe to build a physically sound classifier:
  1. Not to use reconstructed mass, nor features allowing easy mass reconstruction
  2. Try to not use variable regions for which the Monte Carlo simulation doesn't agree with real data

In order to fullfill 2 we have to break the rules and take a look to the control channel



Goal: train a classifier able to separate A from B, but not C from D

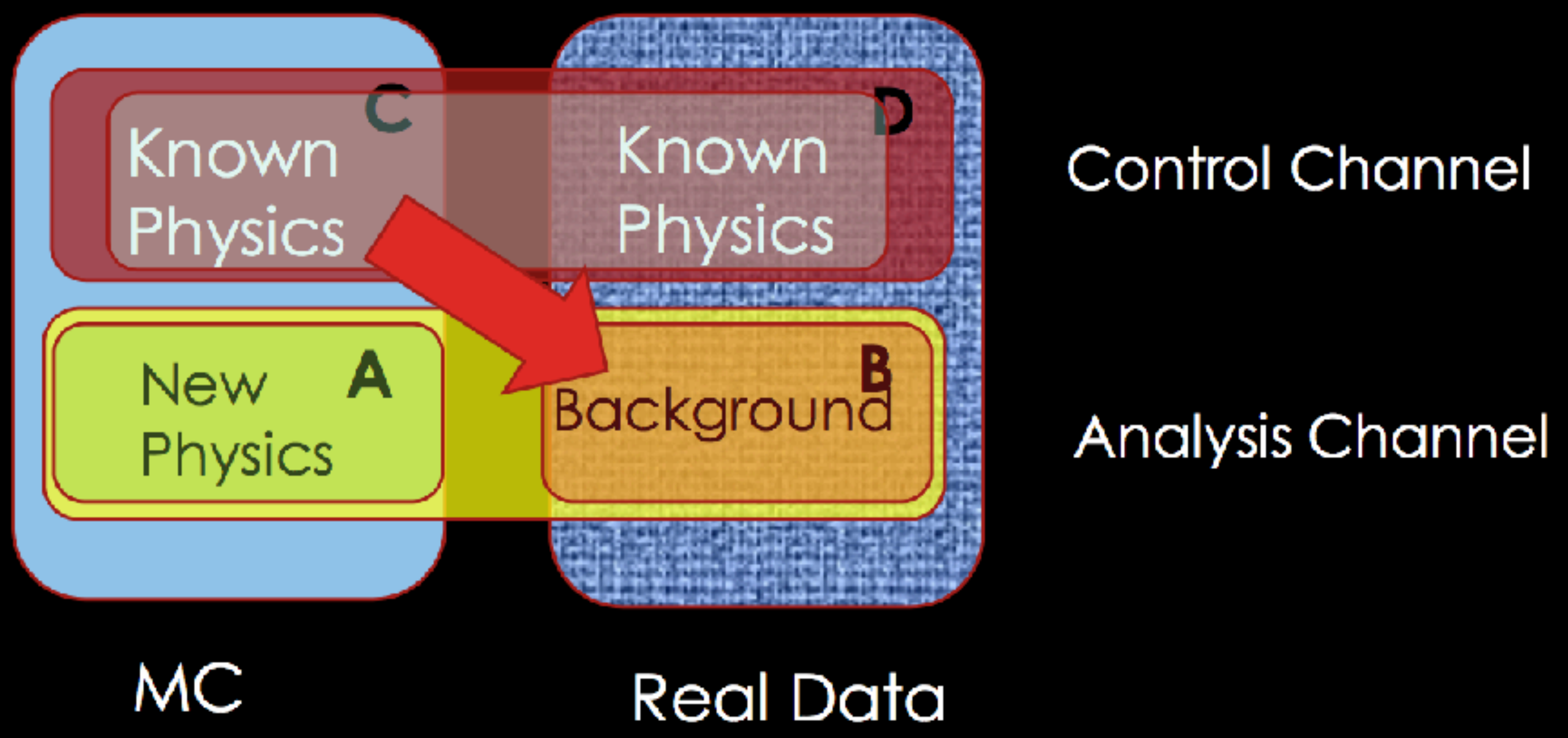
Max(wAUC(A,B)) with KS(C,D)<epsilon

Hypothesis: Control Channel & Analysis channel share the same MC "defects"

BREAKING THE RULES: DATA DOPING

- The idea is to "dope" (in the semiconductor meaning) the training set with a **small number of Monte Carlo events from the control channel , but labeled as background**.

  This disallow the classifier to pick features discriminating data and Monte Carlo.

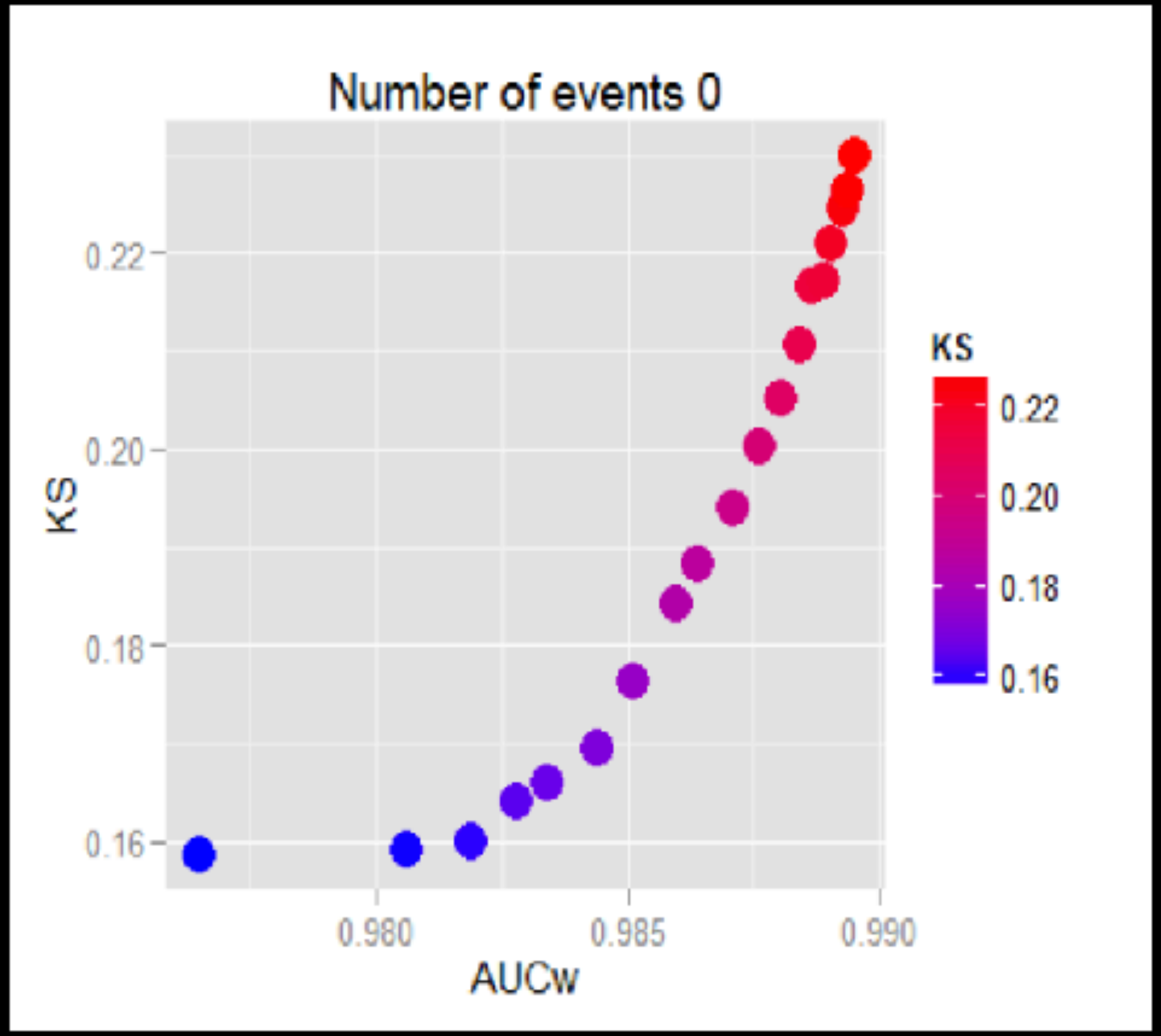


There are two parameters that regularize the learning:

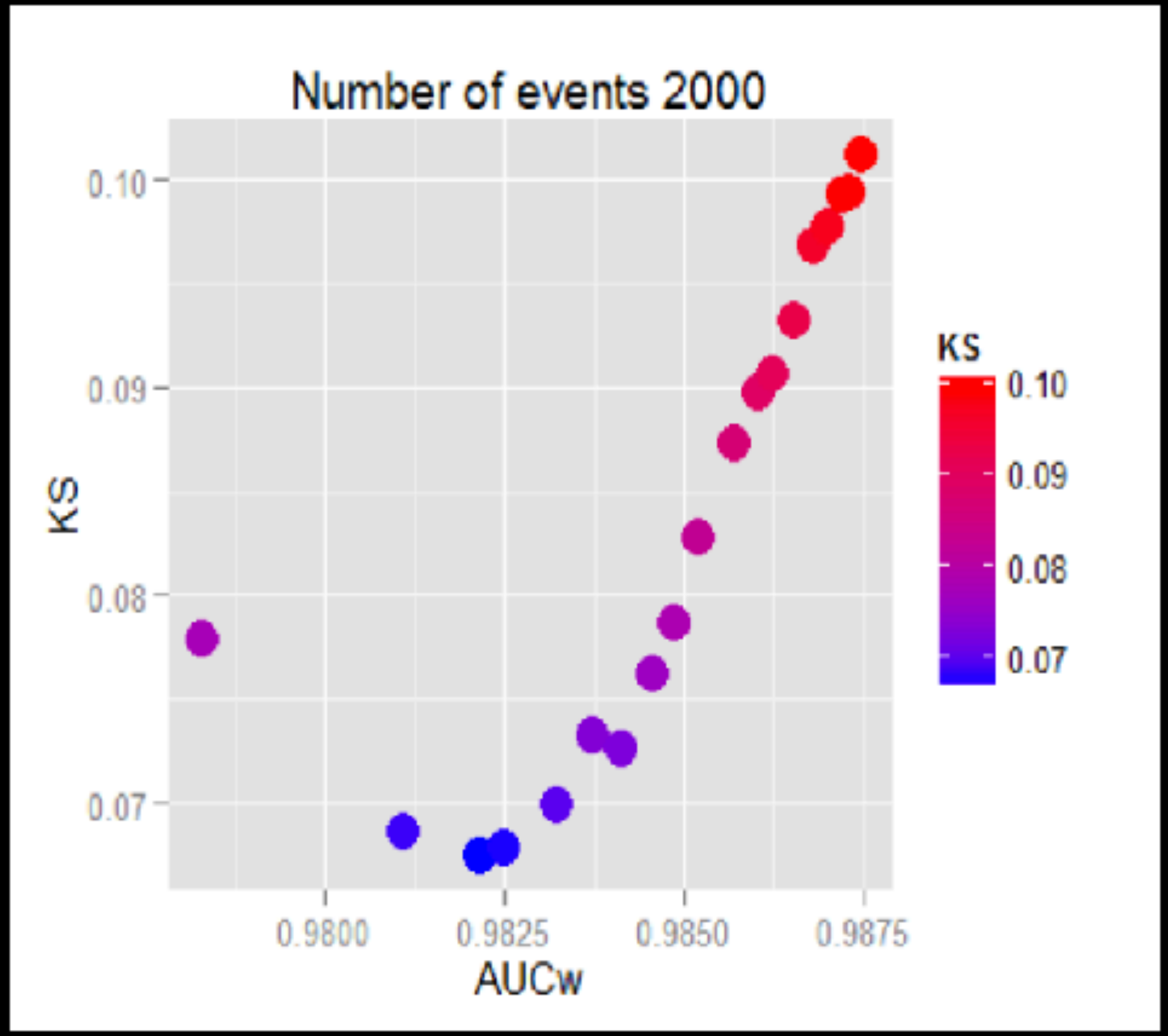
- The number of "doping" events
- the complexity of the classifier (for instance number of trees)

BREAKING THE RULES: DATA DOPING

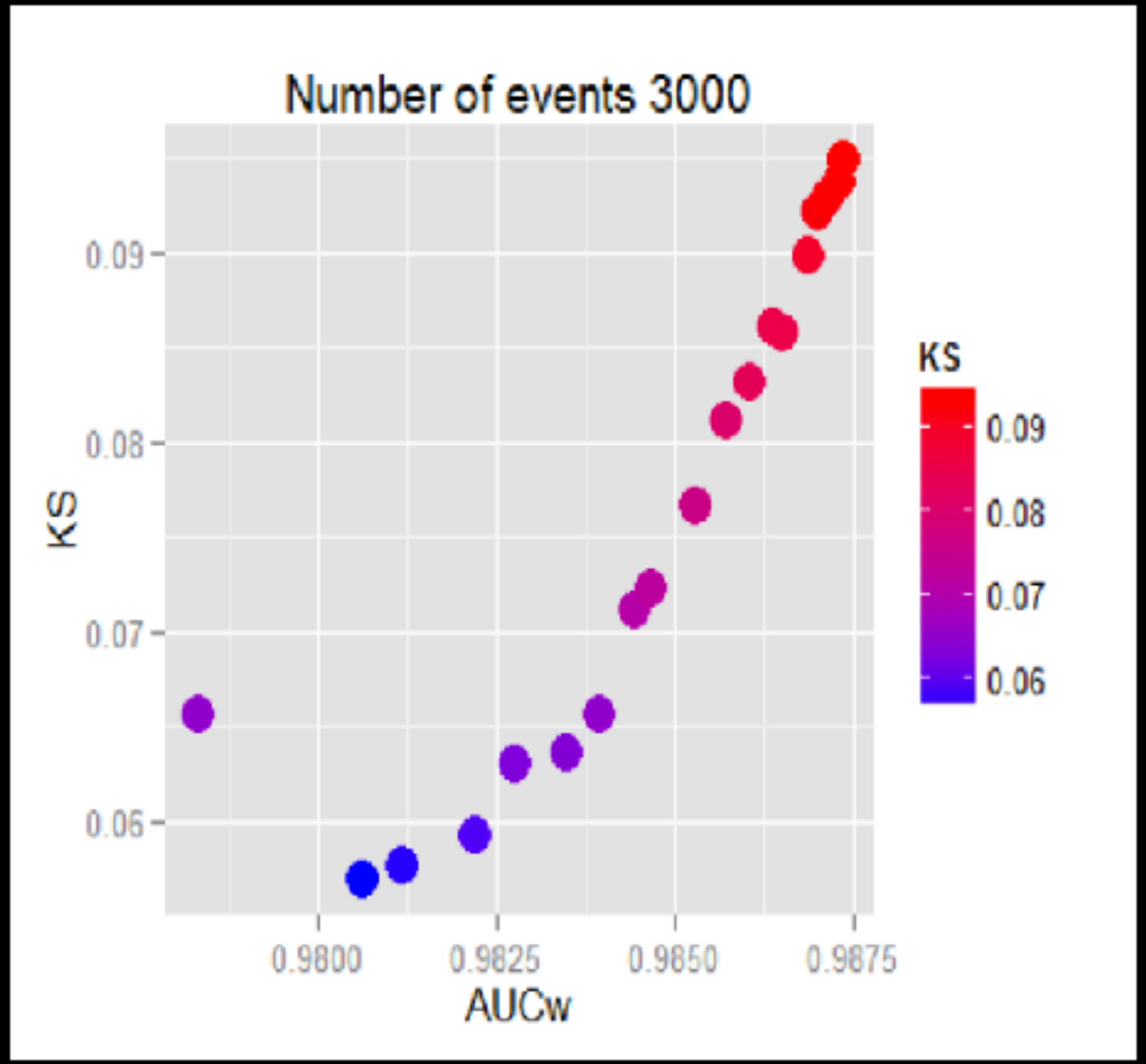
Grid search over Classifier complexity (n_ trees) and Number (weight) of doping events
Dammit! A new hyperparameter….



Free classifier



Doping events: 2000



Doping events: 3000