Using an Optical Processing Unit for tracking and calorimetry at the LHC

Laurent Basara (LRI-Orsay)
Biswajit Biswas, Aishik Ghosh, David Rousseau (IJCLab-Orsay)

Thanks to the LightOn team in particular Laurent Daudet, Iacopo Poli for access to LightOn OPU https://www.lighton.ai/

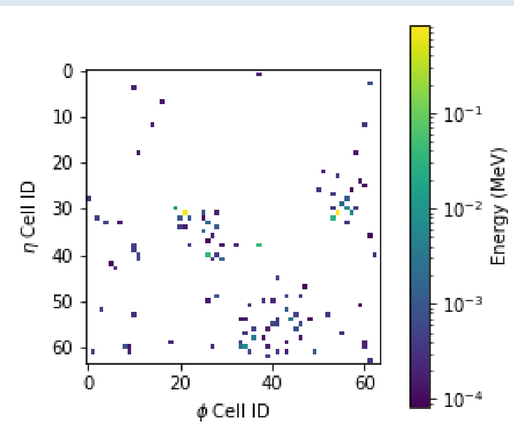
And to Steve Farrell, Wahid Bhimji for access to the dataset and useful discussions

Using an Optical Processing Unit for tracking and calorimetry at the LHC

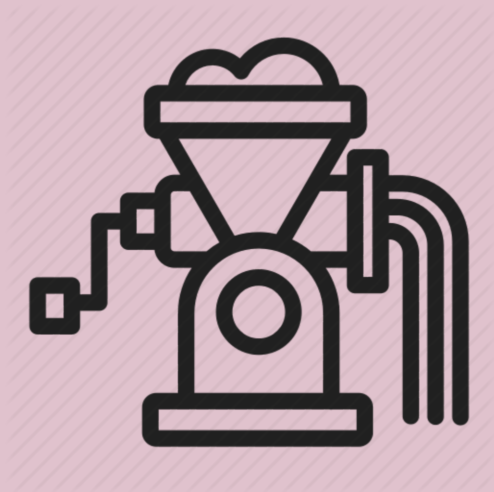
- Optical Processing Units
- OPU for Tracking
- OPU for Calorimetry

Supervised ML

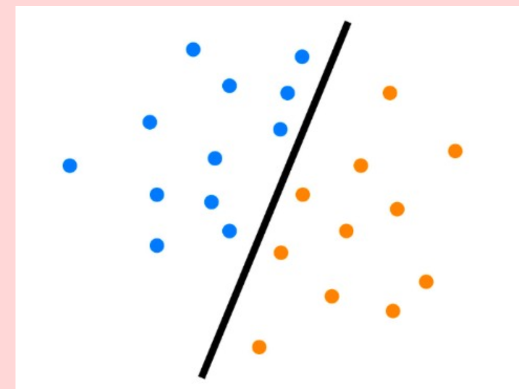
Data representation X
+ ground truth y



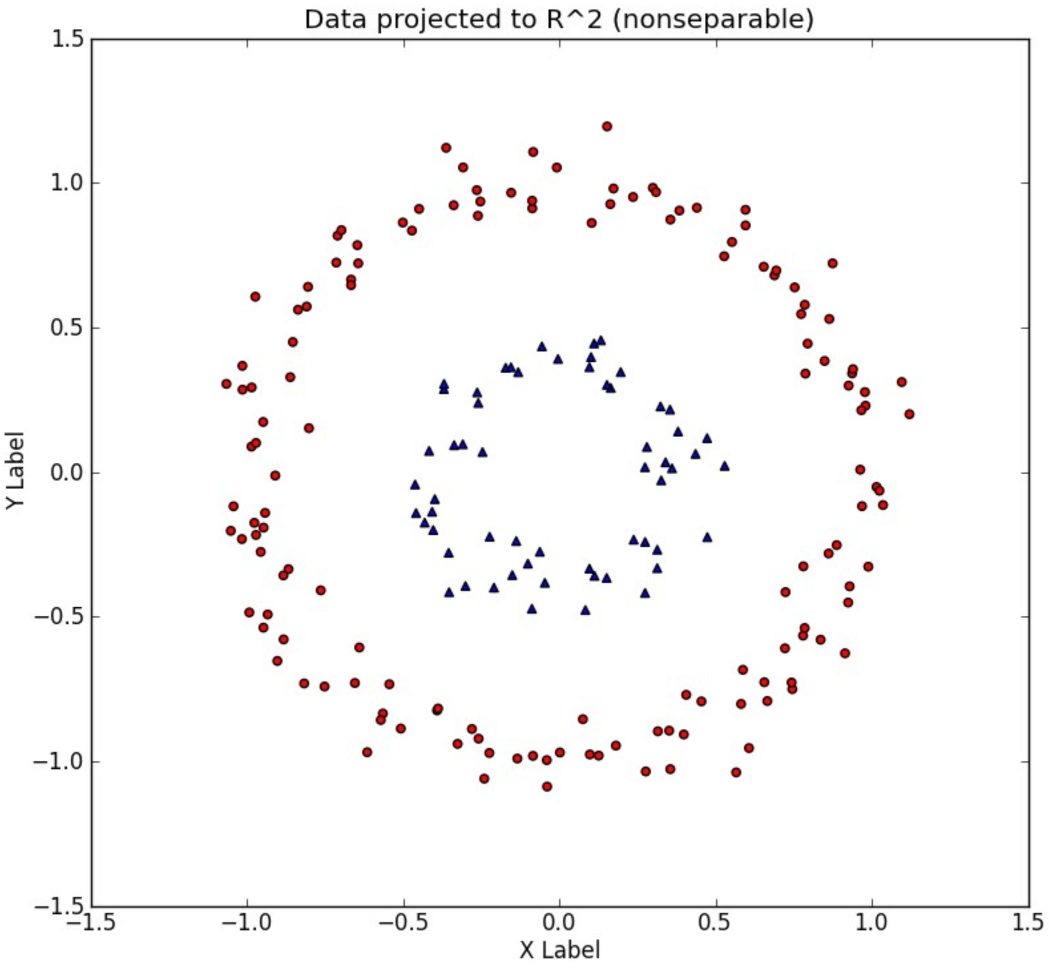
Supervised
ML algorithm

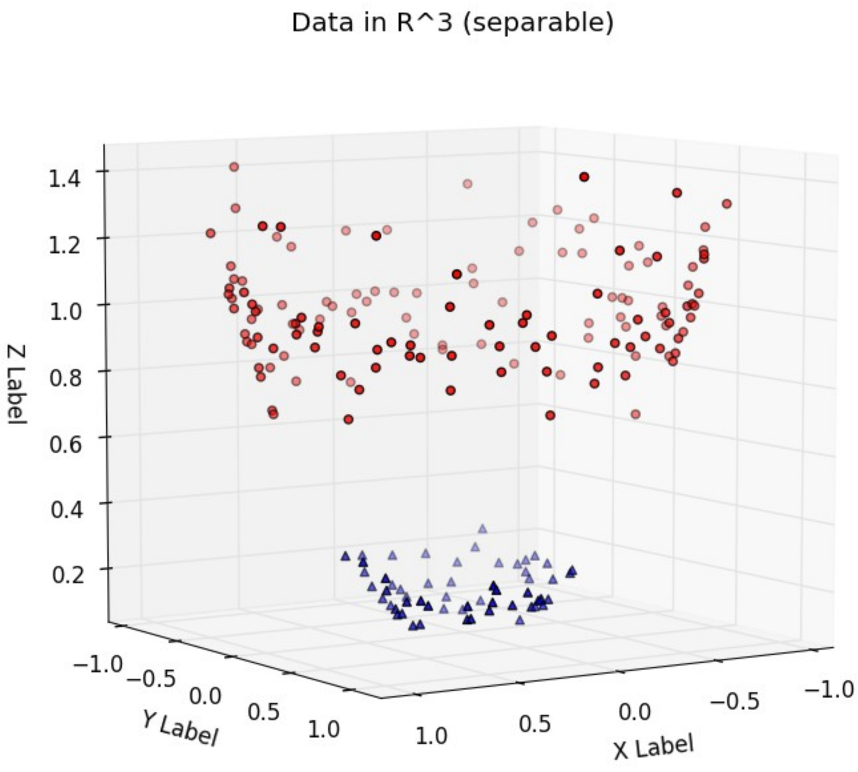


Signal / background
separation



Non-linear problems, SVM and the kernel trick





Kitchen Sinks

Random Features for Large-Scale Kernel Machines

Ali Rahimi and Ben Recht

Abstract

To accelerate the training of kernel machines, we propose to map the input data to a randomized low-dimensional feature space and then apply existing fast linear methods. Our randomized features are designed so that the inner products of the transformed data are approximately equal to those in the feature space of a user specified shift-invariant kernel. We explore two sets of random features, provide convergence bounds on their ability to approximate various radial basis kernels, and show that in large-scale classification and regression tasks linear machine learning algorithms that use these features outperform state-of-the-art large-scale kernel machines.

Everything about the kitchen sink

To fit a kernel SVM, you normally fit a weighted sum of Radial Basis Functions to data:

$$f(x;\alpha) = \sum_{i=1}^{N} \alpha_i k(x, x_i)$$

We showed how to approximate each of these basis functions in turn as a sum of some random functions that did not depend on the data:

$$k(x, x') \approx \sum_{j=1}^{D} z(x;\omega_j) z(x';\omega_j)$$

A linear combination of a linear combination is another linear combination, but with this new linear combination has many fewer (D) parameters:

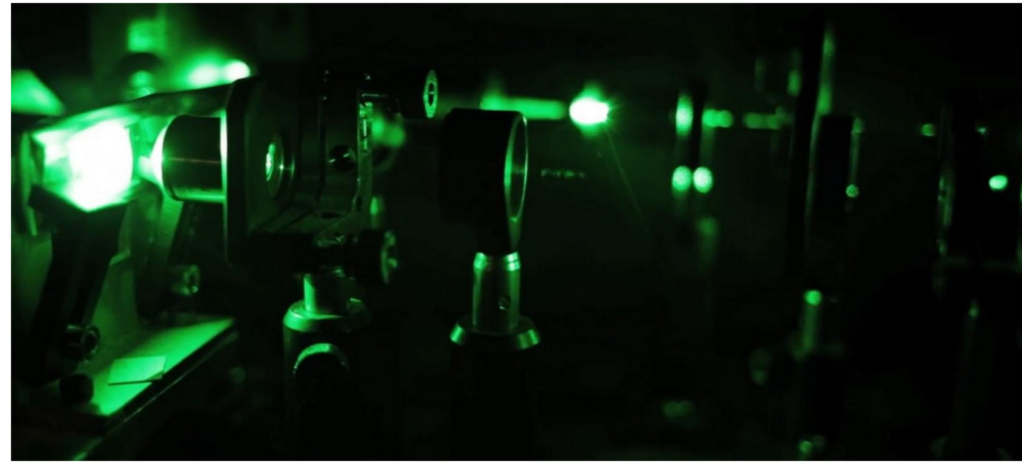
$$f(x;\alpha) \approx \sum_{j=1}^{D} \beta_i z(x;\omega_j)$$

We showed how to approximate a variety of radial basis functions and gave bounds for how many random functions you need to approximate them each of them well.

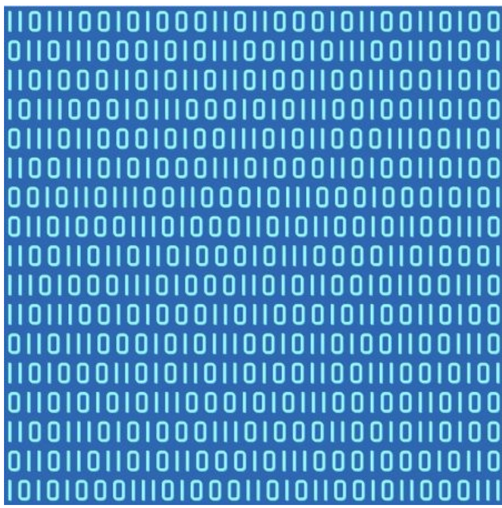
Original paper: https://people.eecs.berkeley.edu/~brecht/papers/07.rah.rec.nips.pdf

Popularization: http://www.argmin.net/2017/12/05/kitchen-sinks/

Optical Processing Unit

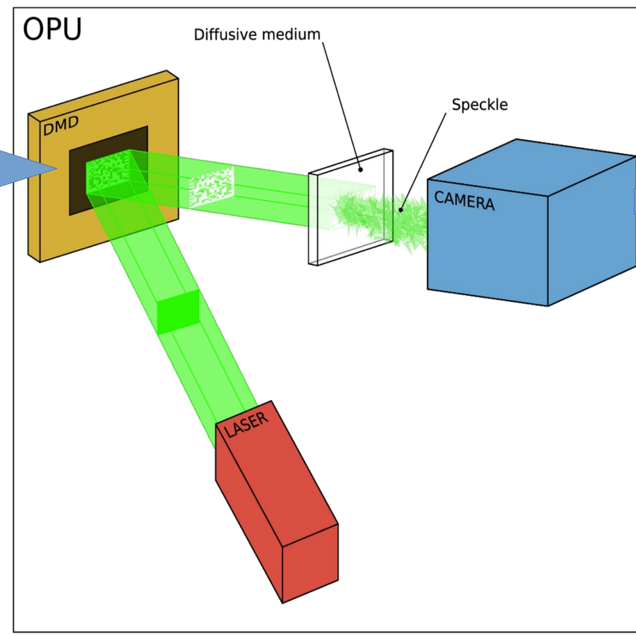


https://docs.lighton.ai/notes/opu.html

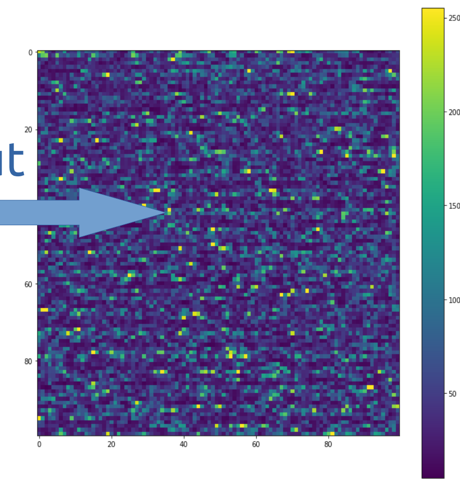


OPU in

1 M bits vector X



OPU out

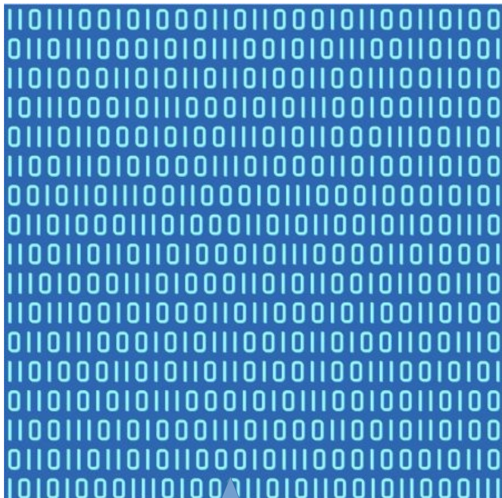


In practice 10K
to 100K bytes
(random features)

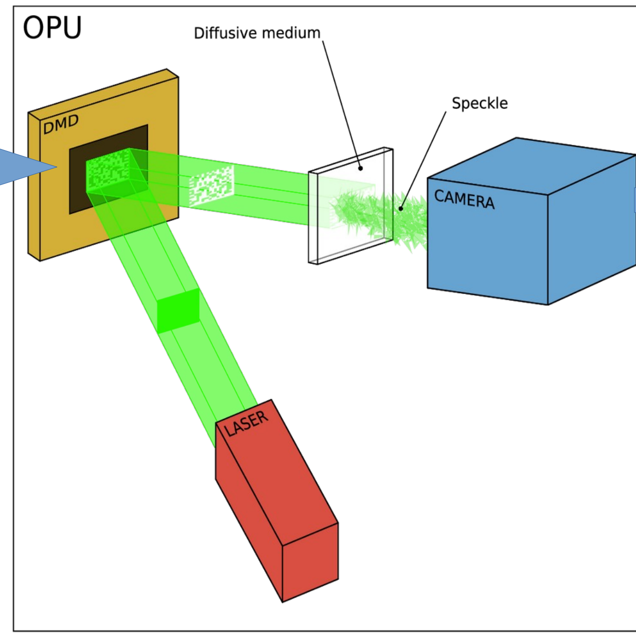
Random matrix multiplication
$y = | Hx |^2$
Size 10^{12} pixels x Rate 2 kHz
~ 10^{15} operations / s for a few Watts

Optical Processing Unit ML workflow

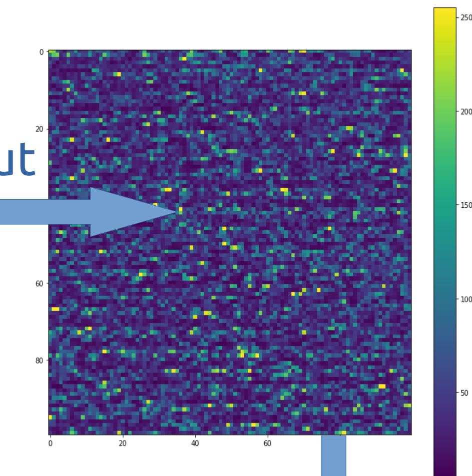
https://docs.lighton.ai/notes/opu.html



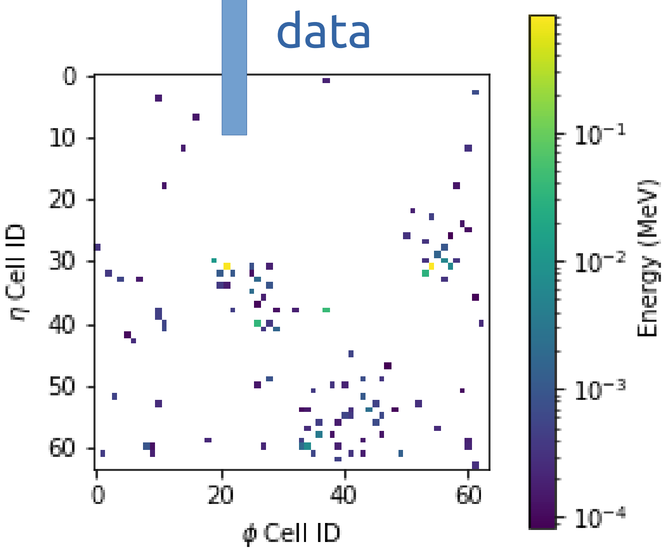
OPU in



OPU out

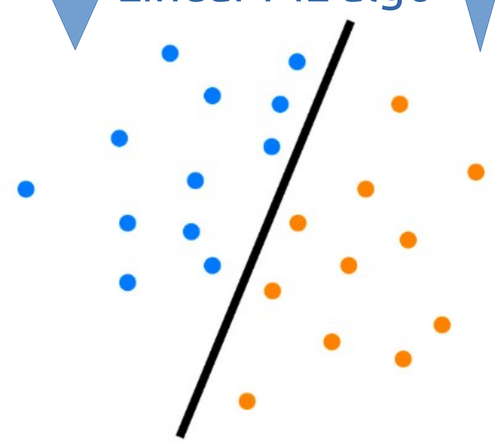


Binarize data



Ground truth

Linear ML algo



Using an Optical Processing Unit for tracking and calorimetry at the LHC

- Optical Processing Units
- OPU for Tracking
  - Single track parameter estimation
  - Multi-track parameter estimation
- OPU for Calorimetry

TrackML challenge numbers

- HL-LHC conditions
- Pile-up: 50 → 200
- 10 K particles / collision
- 100 K 3D points / collision
- 3-20 hits / particle
- Innovative data analysis solutions → OPU ?

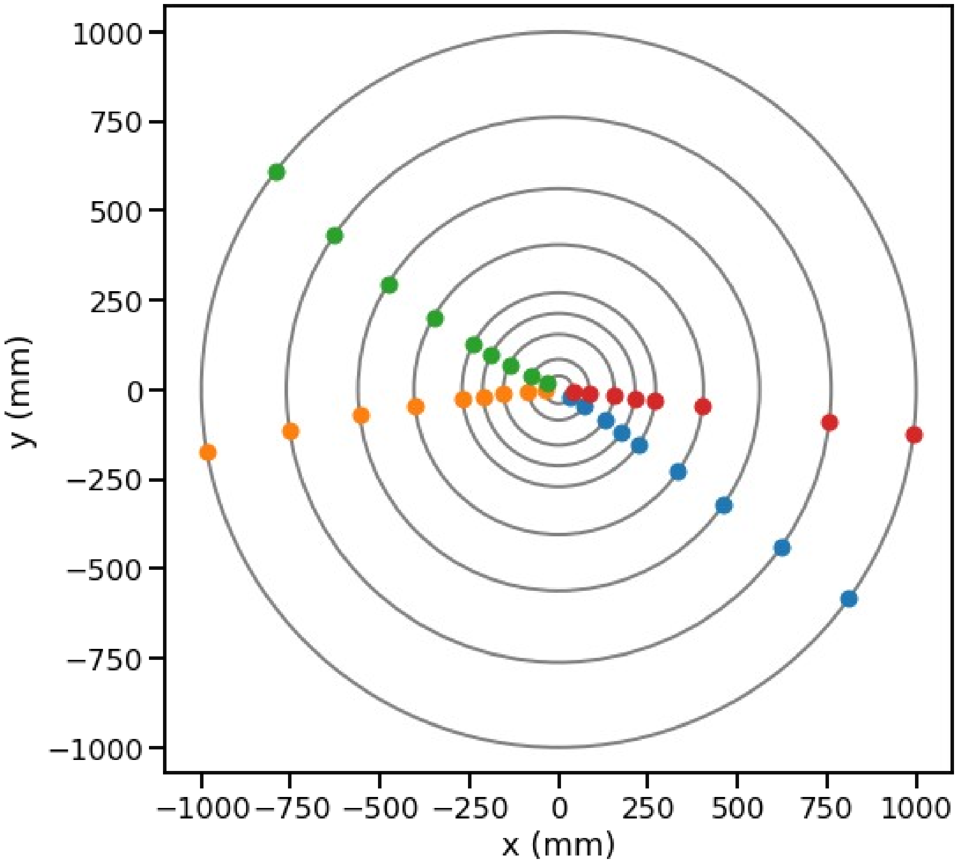
How to proceed?

- Track following? No simple geometry of successive layers
- Compress the hits seen in electronics?
  - 2B electronic channels (!) → 1M OPU bits
  - Test with layered tSVD, autoencoders… did not give anything interesting
- Use a more manageable dataset
  - 2D dataset from RAMP track challenge

RAMP 2D dataset

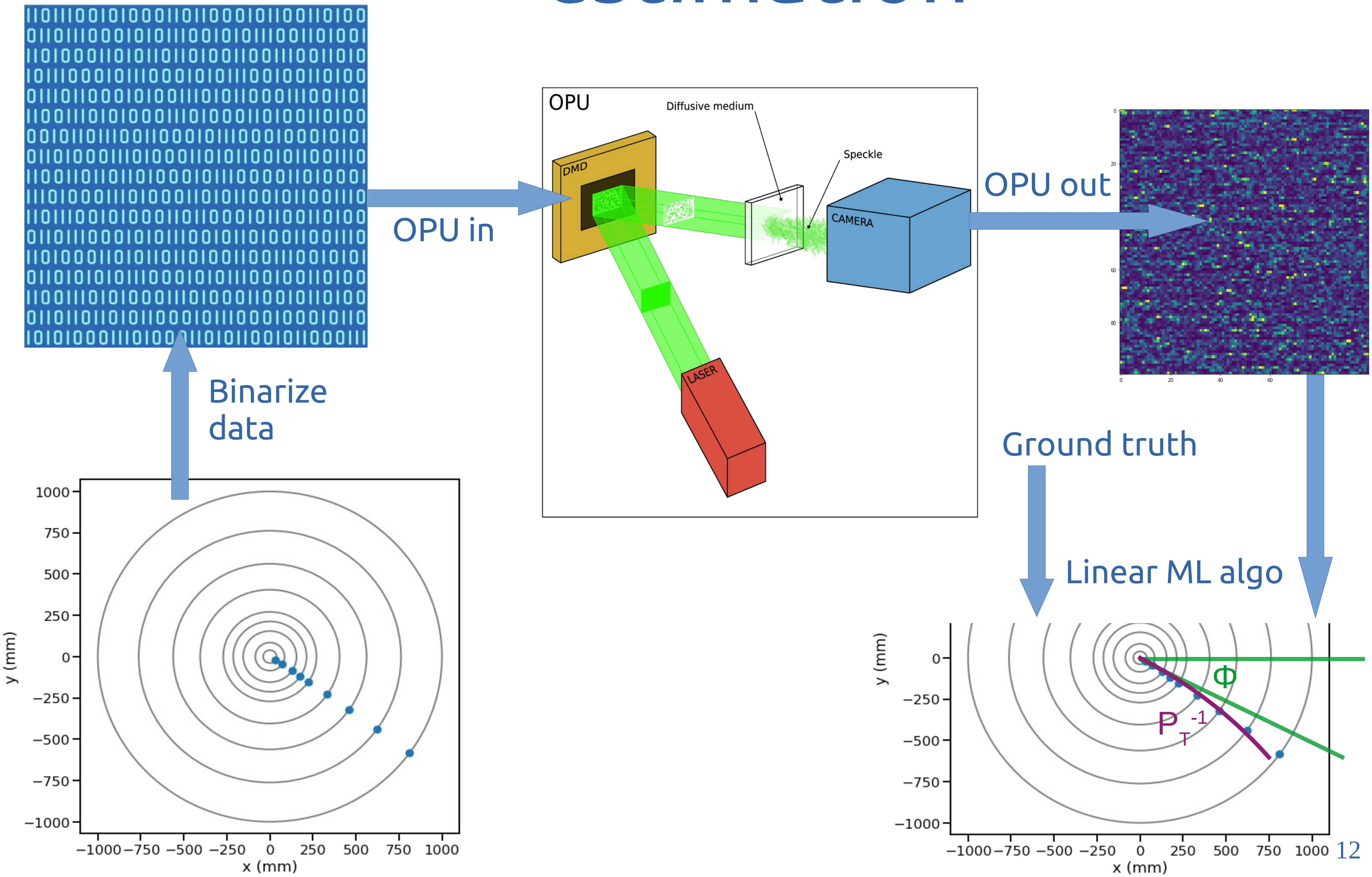
Track reconstruction at LHC as a collaborative data challenge use case with RAMP

Sabrina Amrouche , Nils Braun , Paolo Calafiura , Steven Farrell , Jochen Gemmler , Cécile Germain [1, 2] , Vladimir Vava Gligorov [3] , Tobias Golling , Heather Gray , Isabelle Guyon [2] , Mikhail Hushchyn , Vincenzo Innocente , Balázs Kégl [1, 4] , Sara Neuhaus , David Rousseau [1] , Andreas Salzburger , Andrei Ustyuzhanin , Jean-Roch Vlimant , Christian Wessel , Yetkin Yilmaz [1, 4] Détails

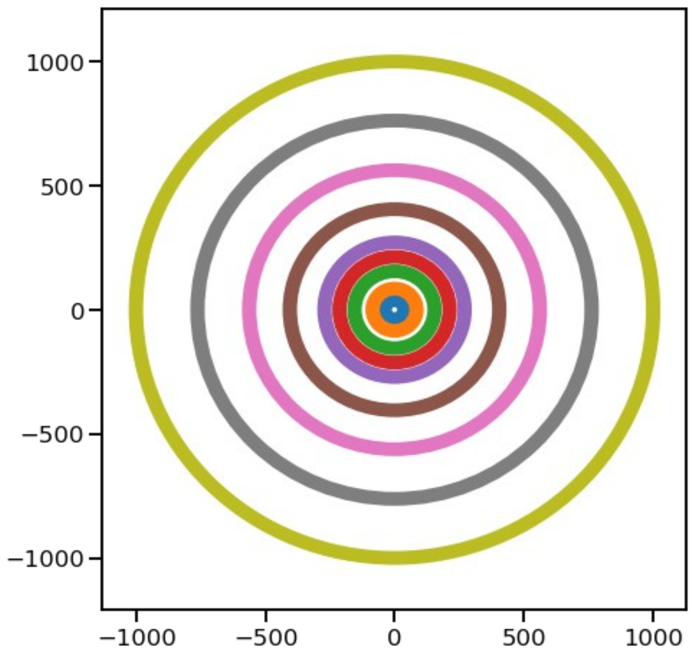


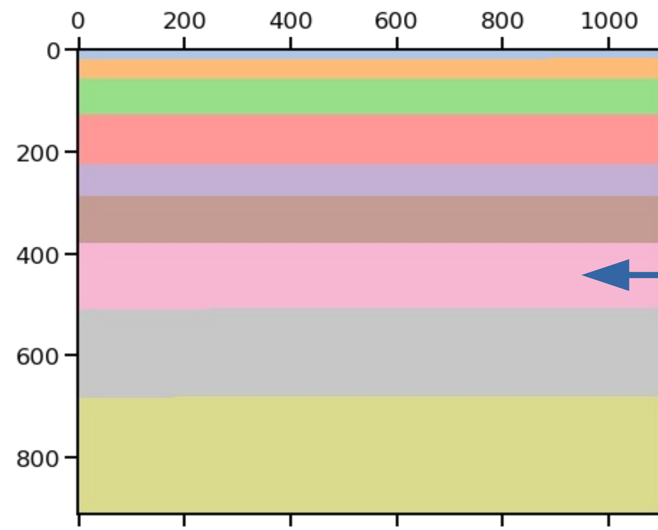
- Original paper : http://inspirehep.net/record/1616034
- Original library : https://github.com/yetkinyilmaz/tracking
- Python3 port : https://github.com/LAL/tracking2Dsim
- 60 K events
- 9 layers, 530 K pixels total

Single track parameter estimation

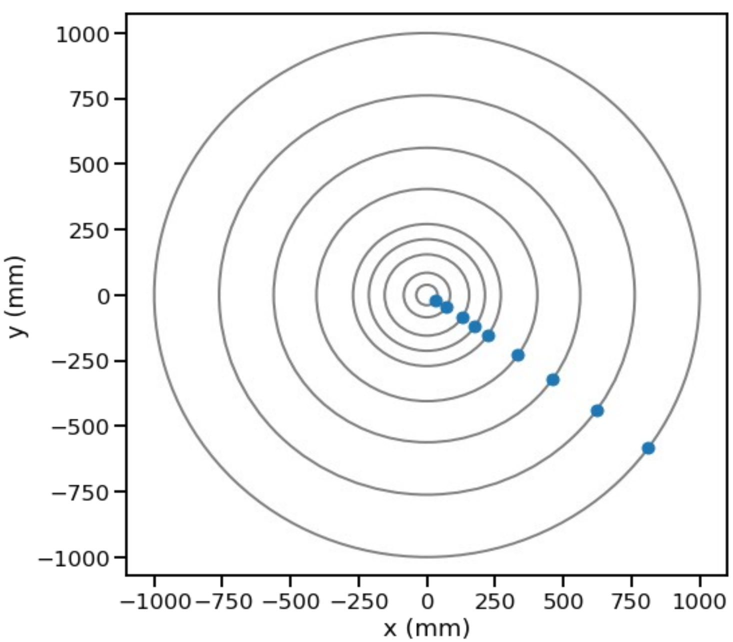


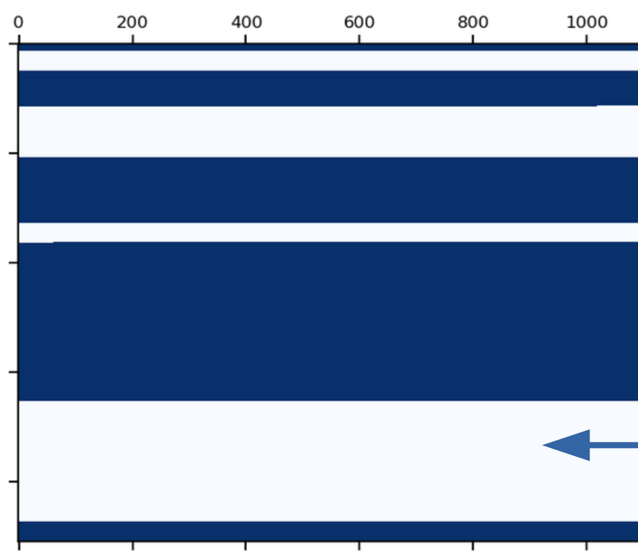
Single track: Binary encoding





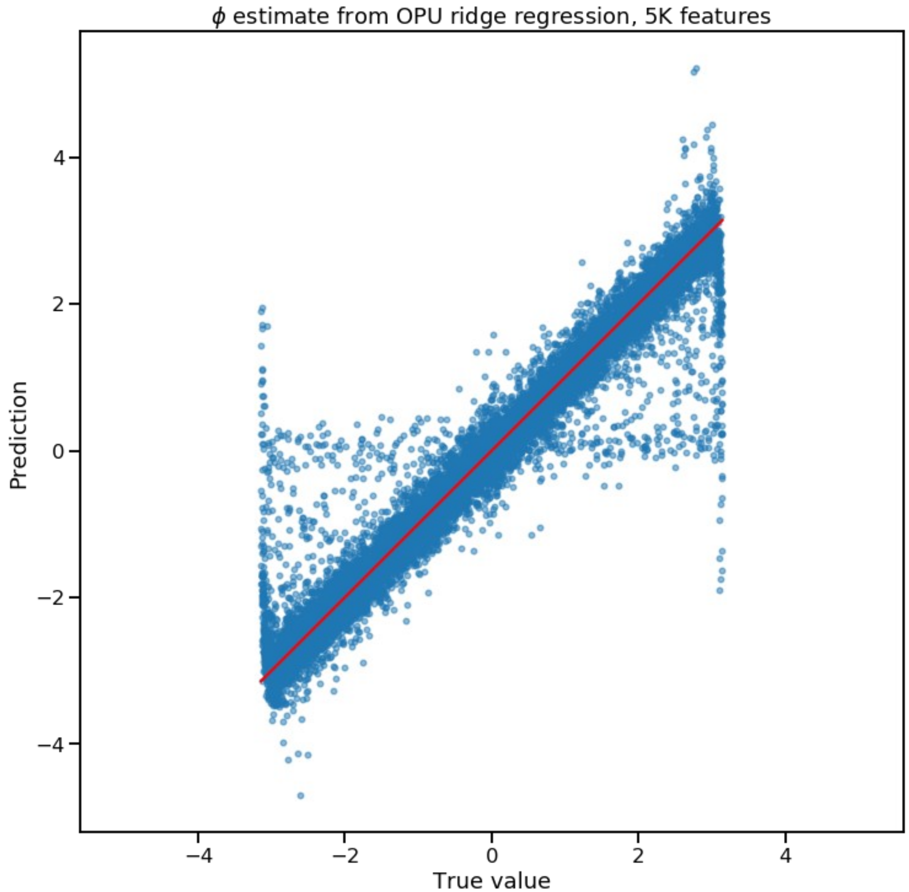
~ 2 times less tracker pixels than DMD pixels
→ all pixels represented on DMD at once

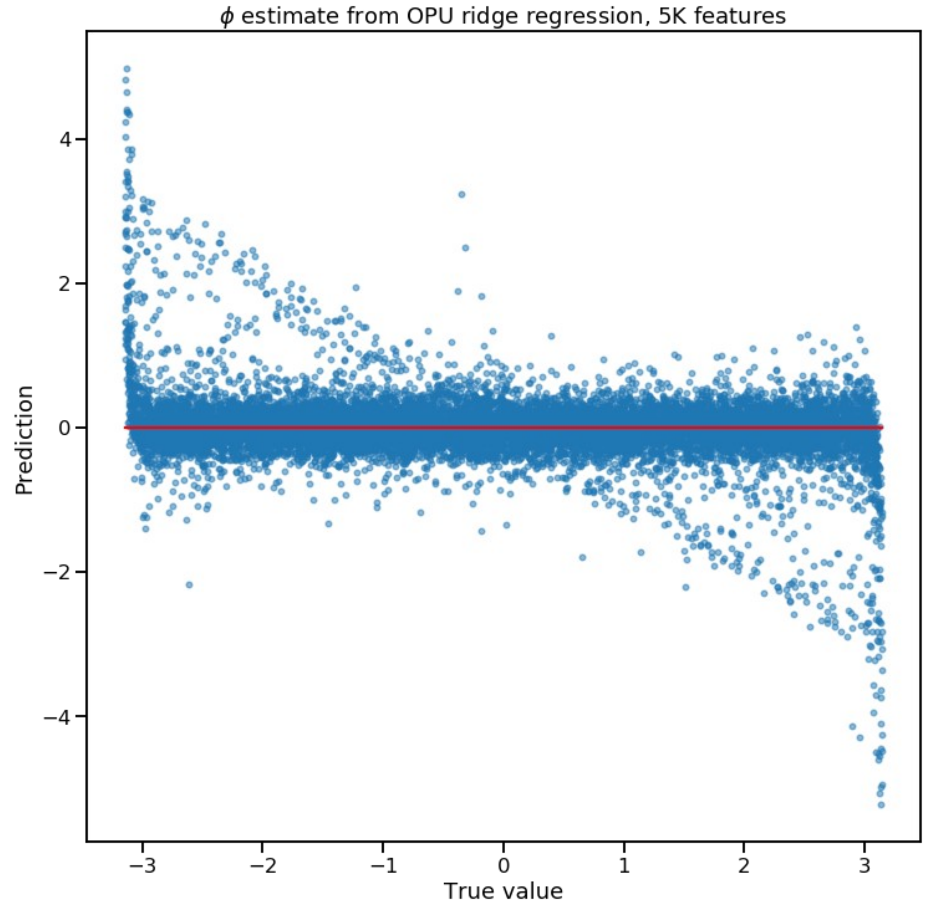




Each layer at most one hit we change bool value at each hit seen
→ ~ 50 % of DMD lit

Initial angle estimation

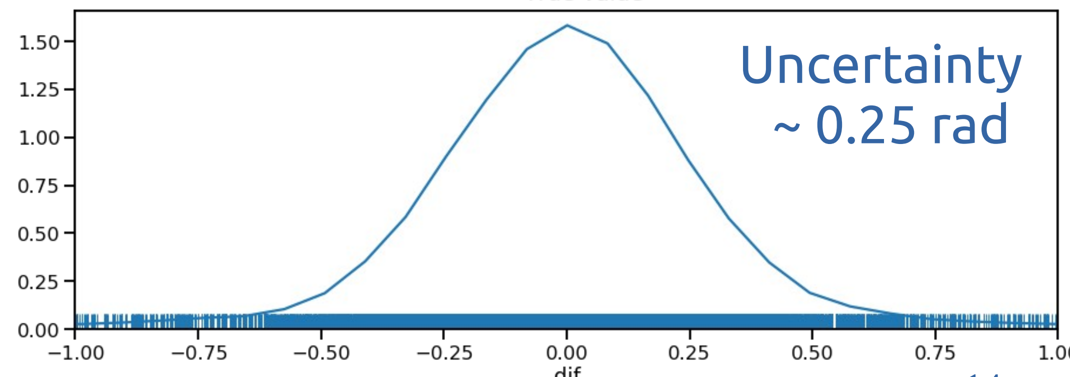




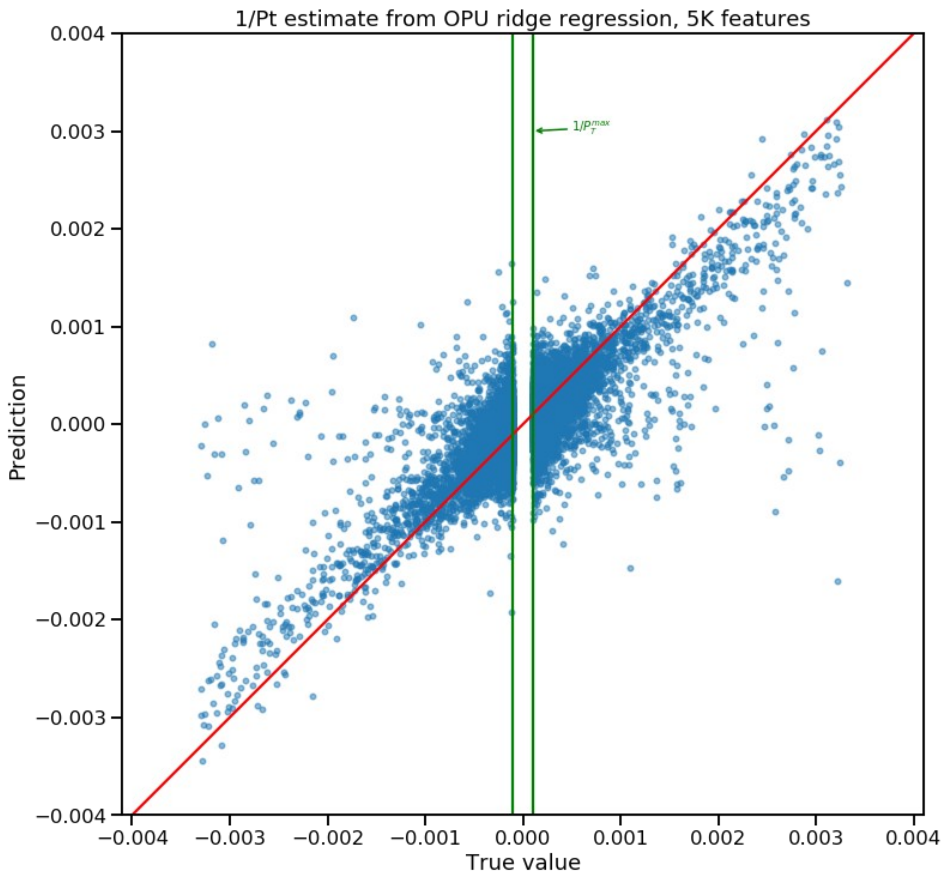
$$\boxed{\min_{\beta\in R^{m\times n}} ||X\beta - y|| + ||\gamma\beta||}$$

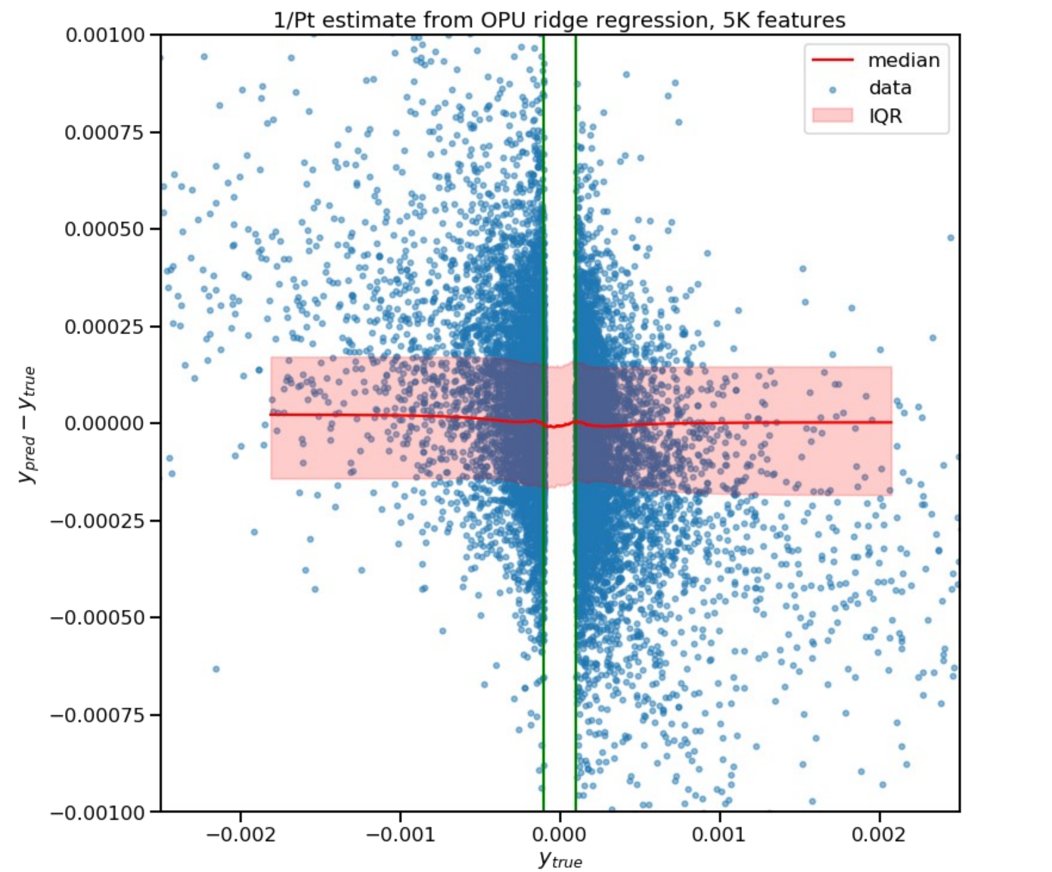
(5K) random features

Ground truth angle



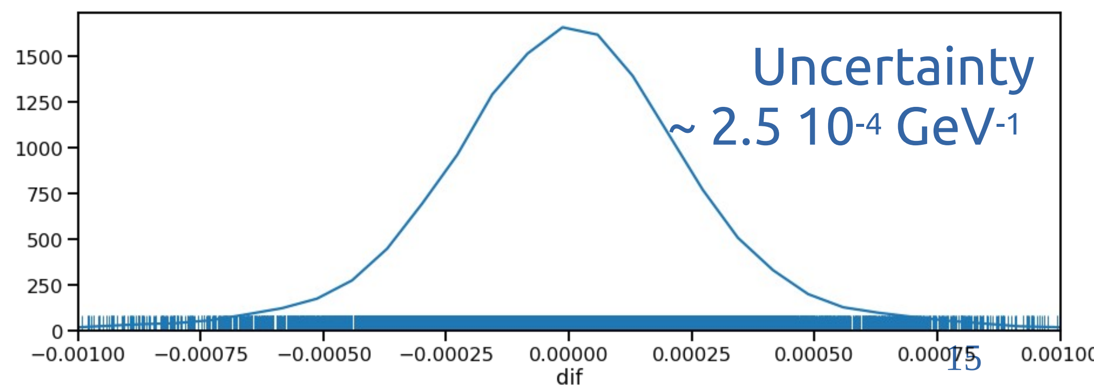
(inverse) momentum estimation



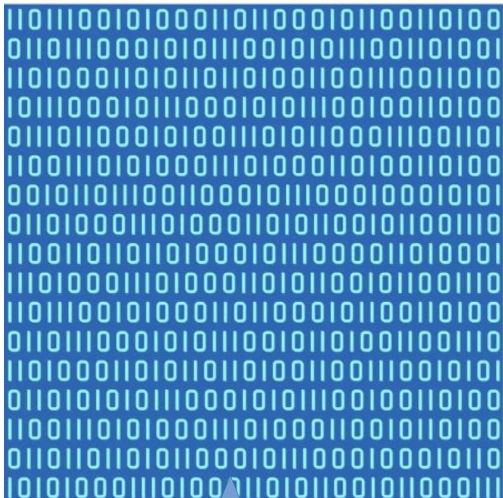


$$\boxed{\min_{\beta \in R^{m\times n}} ||X\beta - y|| + ||\gamma\beta||}$$

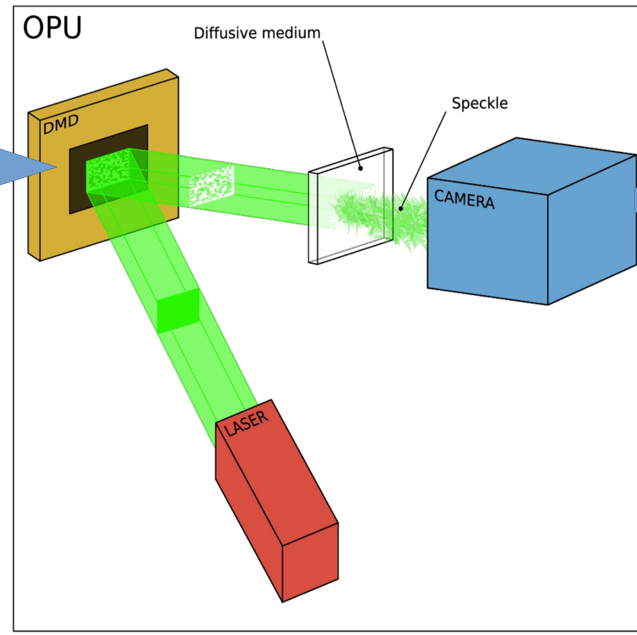
Inverse momentum (~curvature)



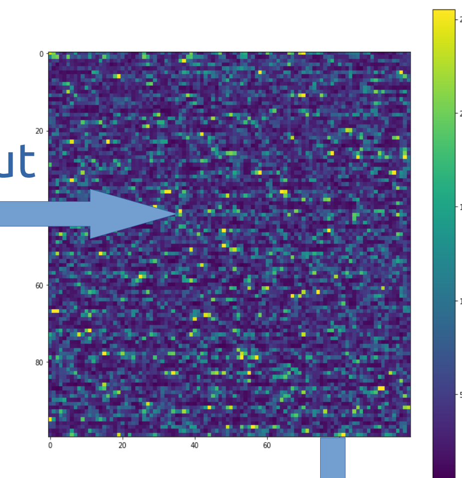
Multi-track parameter estimation



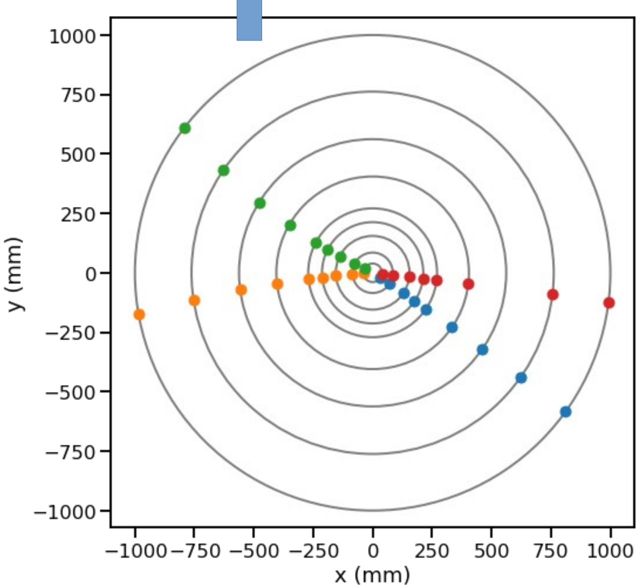
OPU in



OPU out



Binarize data

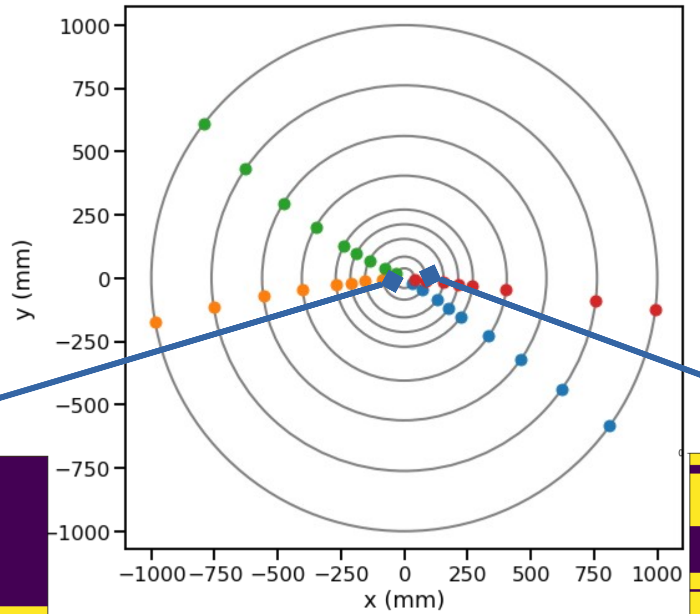


Ground truth

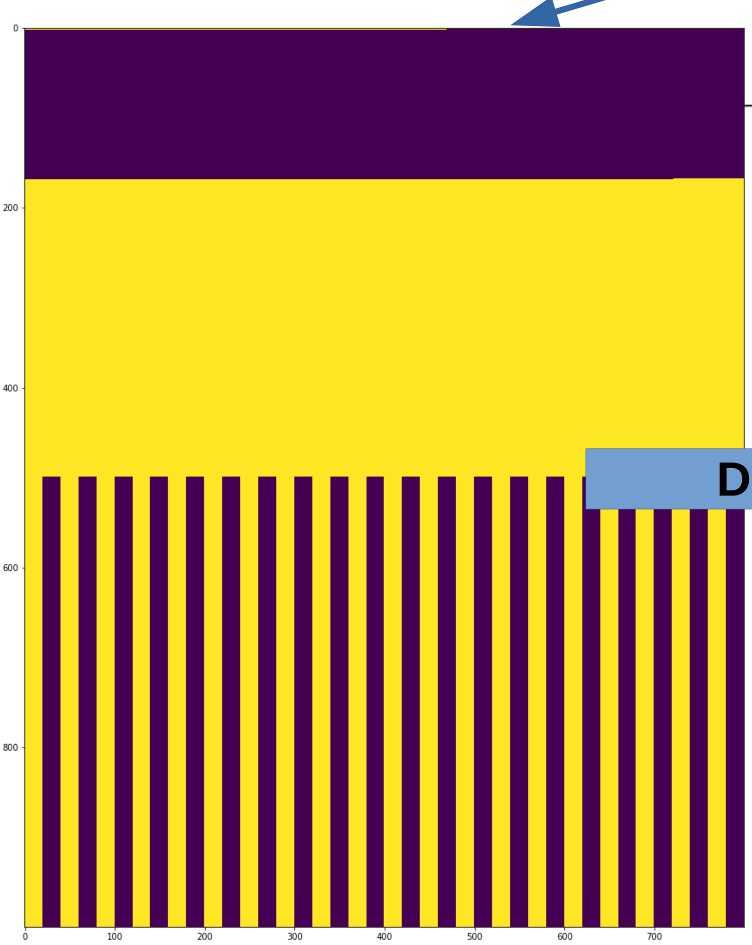
Linear ML algo

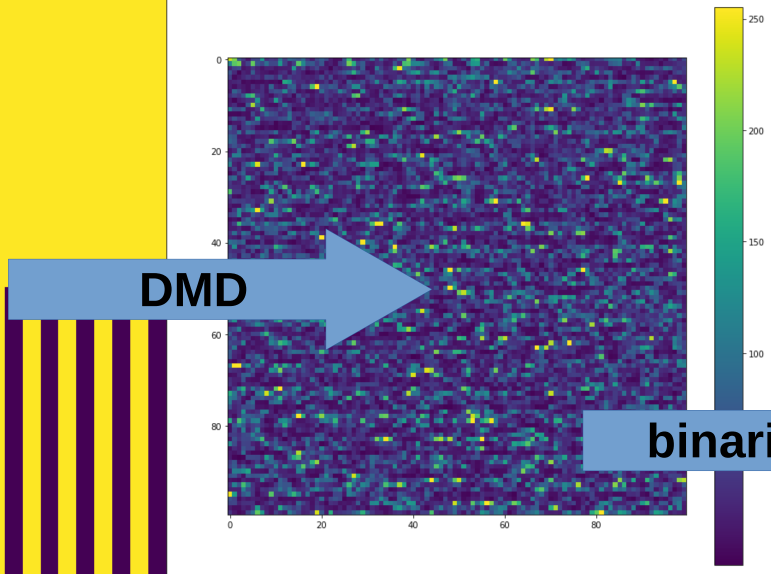
N particles ?
Angles ?
Momenta ?

Binary encoding

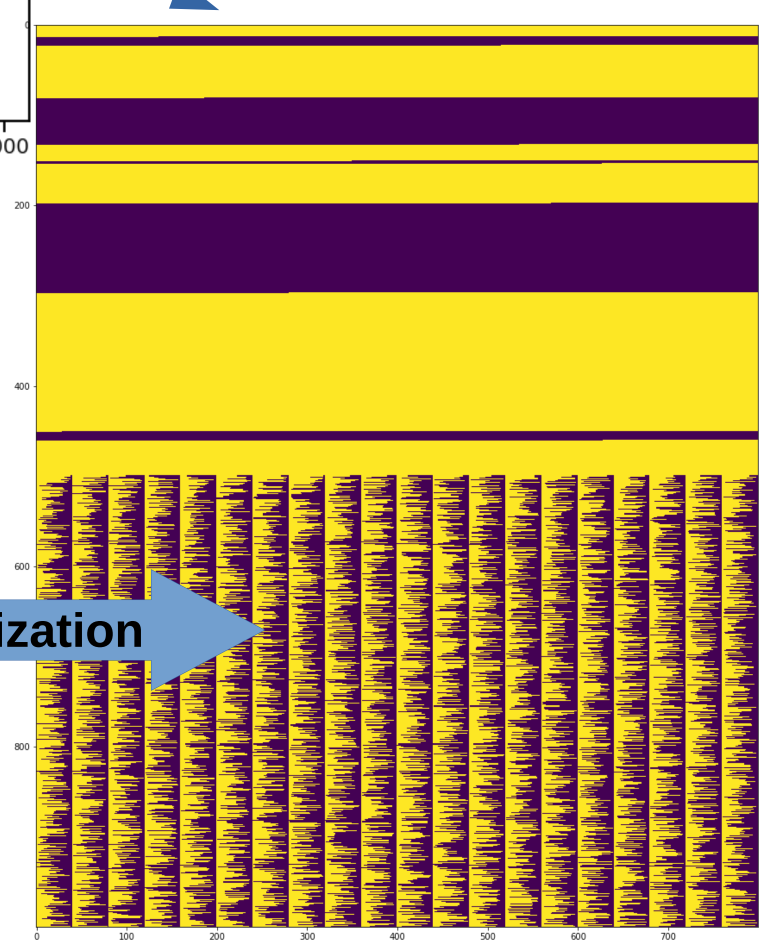


First layer

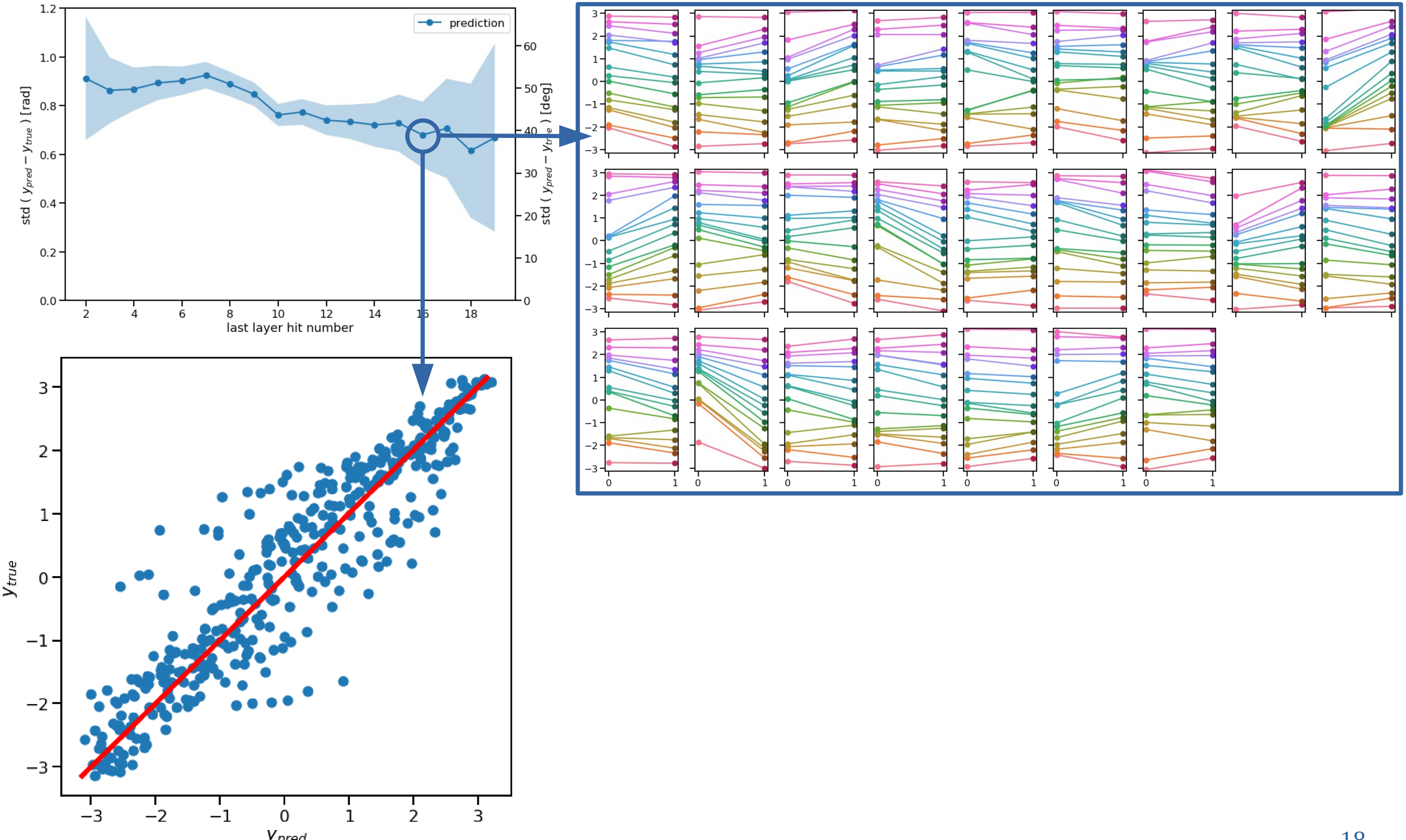




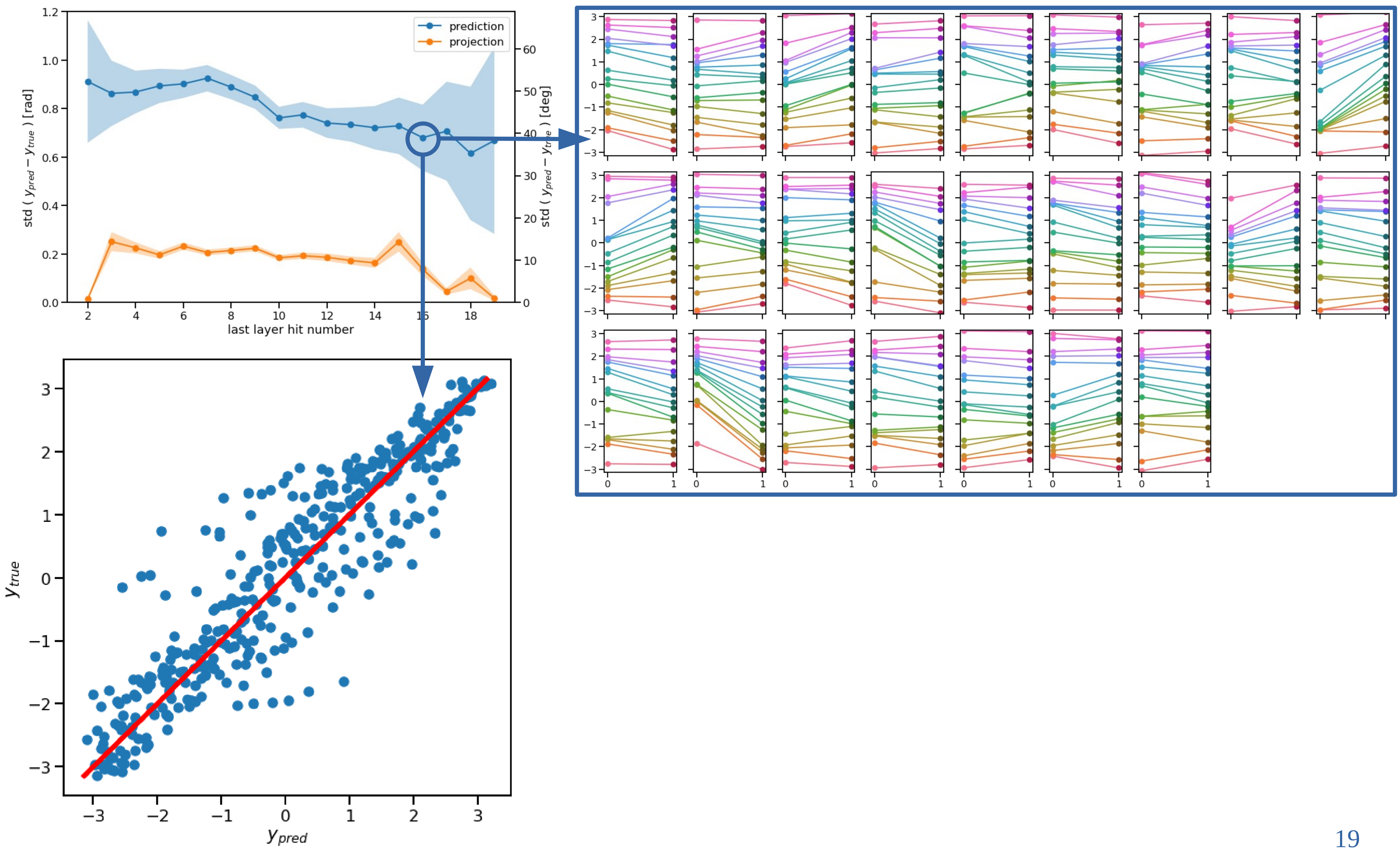
Second layer



Standard deviation wrt hit number



Standard deviation wrt hit number



Conclusions on Tracking

- OPU provides physical device to reduce dimensionality / training time
- Casting a Tracking problem for OPU is hard ; nonetheless estimations of
  - Single particle parameters (angle, inverse momentum)
  - Number of particles, position projected on next layer
  - OPU « makes sense » without matching traditional methods
- More suited to calorimetry?

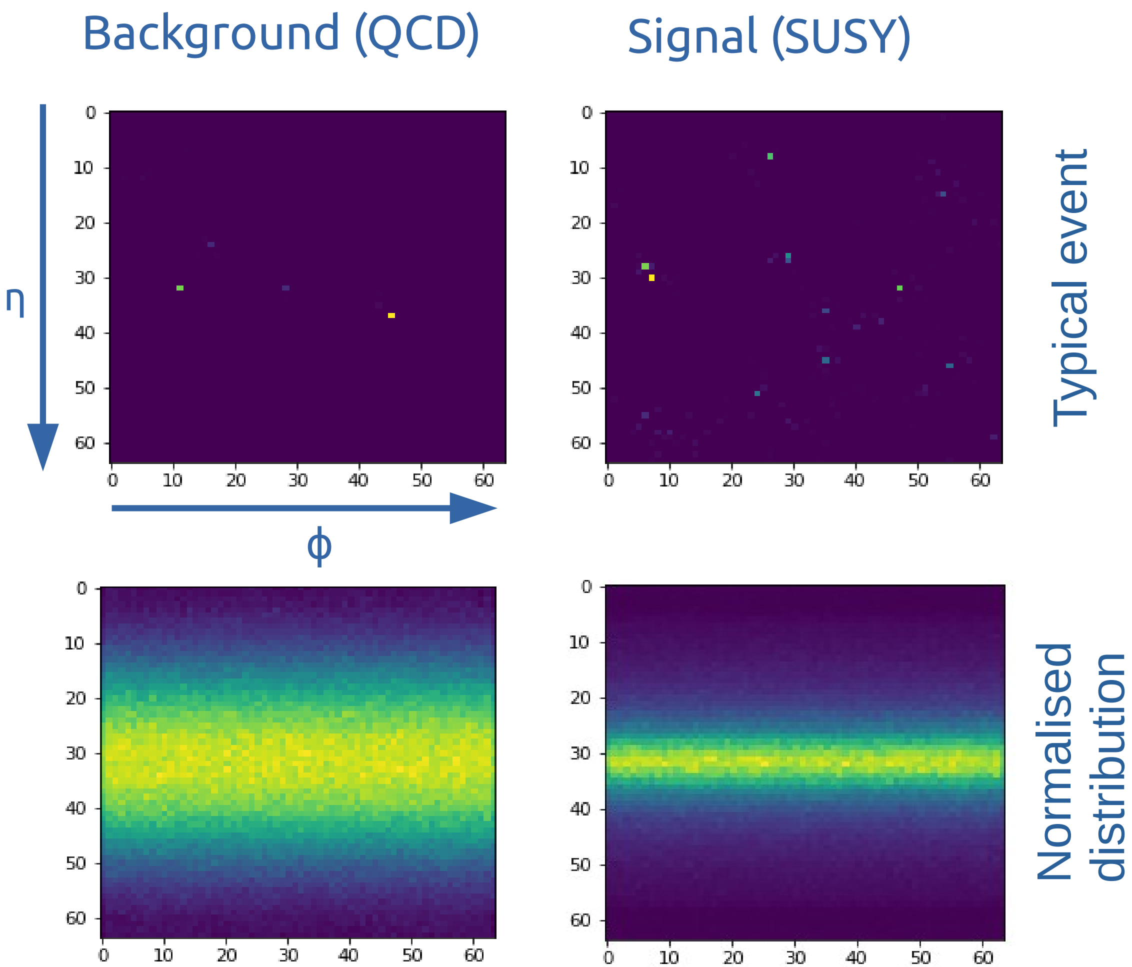
Using an Optical Processing Unit for tracking and calorimetry at the LHC

- Optical Processing Units
- OPU for Tracking
- OPU for Calorimetry
  - SUSY vs QCD
  - W / ttbar vs QCD

Study 1: following arXiv:1711.03573

Deep Neural Networks for Physics Analysis on low-level whole-detector data at the LHC

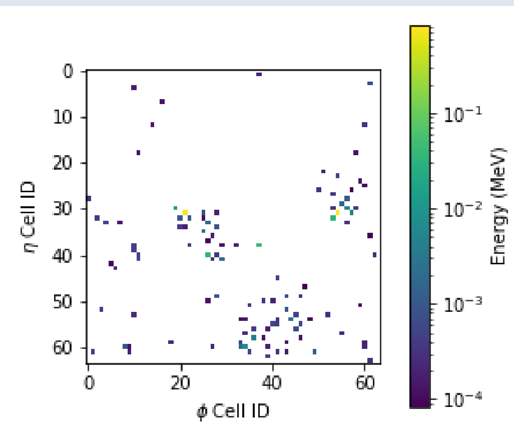
Wahid Bhimji[1], Steven Andrew Farrell[1], Thorsten Kurth[1], Michela Paganini[1,2], Prabhat[1], Evan Racah[1]



- Signal : RPV-Susy, gluino-cascade decays, gluino and neutralino masses of 1400 GeV and 850 GeV
- Background : QCD
- Training sample: 400k events
- Uniform 64x64 bins correspond 0.1x0.1(ηxφ) ATLAS HCAL resolution ; intensity = energy deposited
- Images cover entire detector, whole events classification

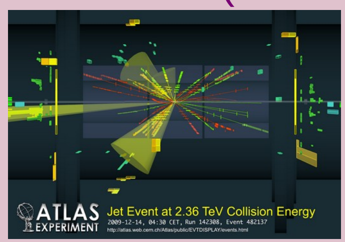
OPU competitive with CNN ?

Calorimeter image
+ ground truth



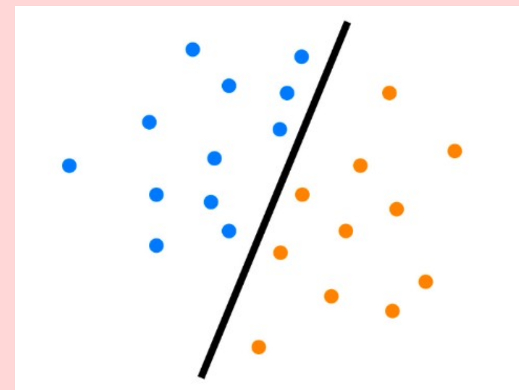
Supervised
ML algorithm

Feature engineering
Classical ML (BDT…)



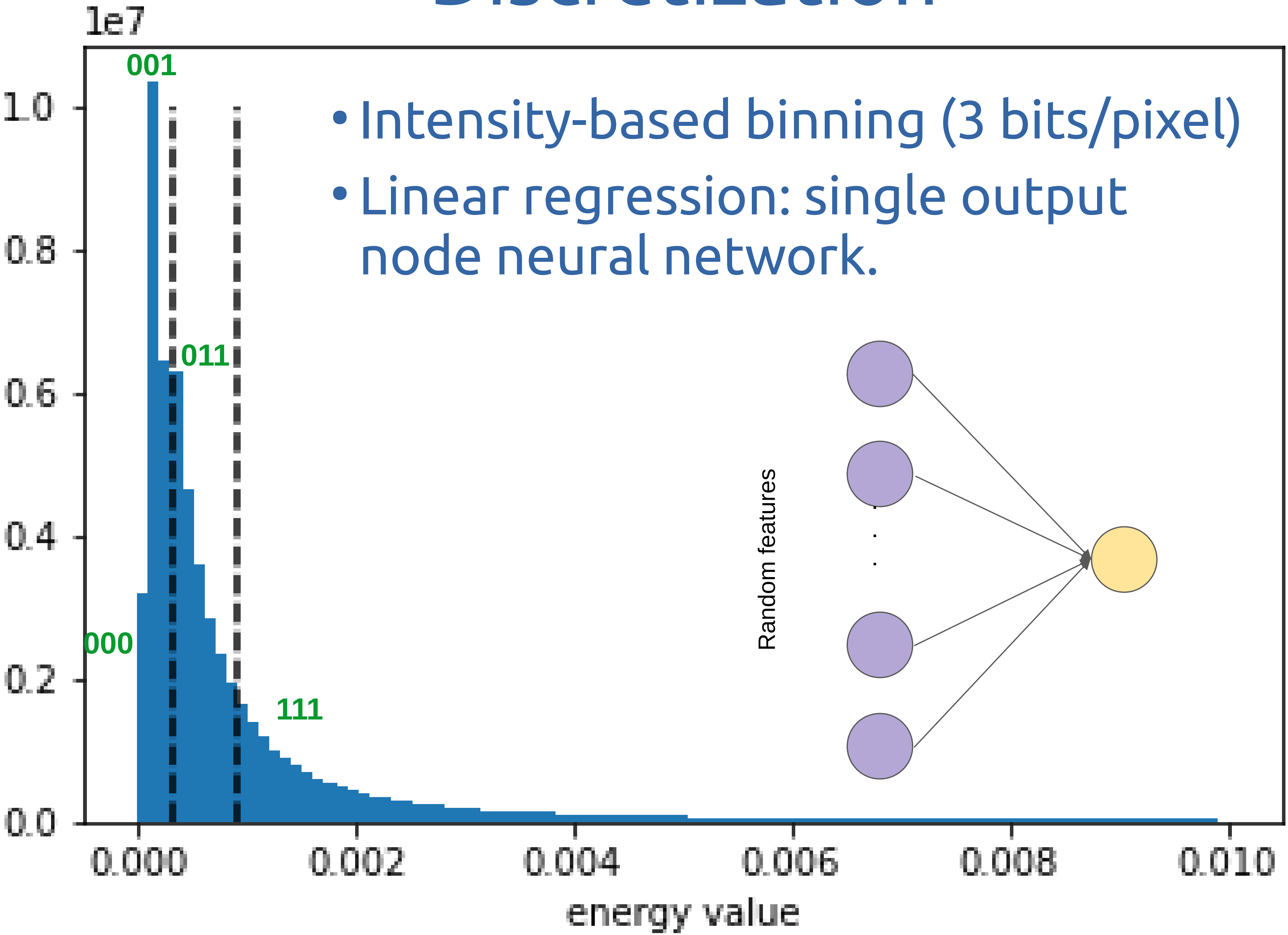
Raw features
Modern ML :
- CNN
- OPU ?

Signal / background
separation

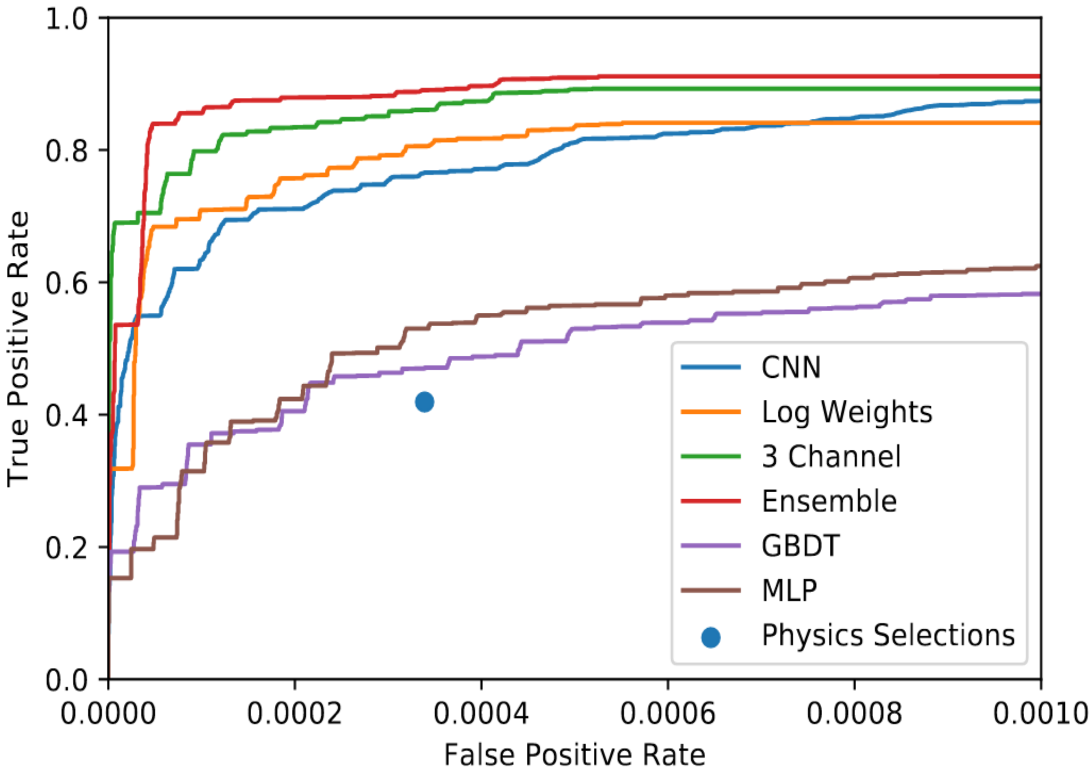


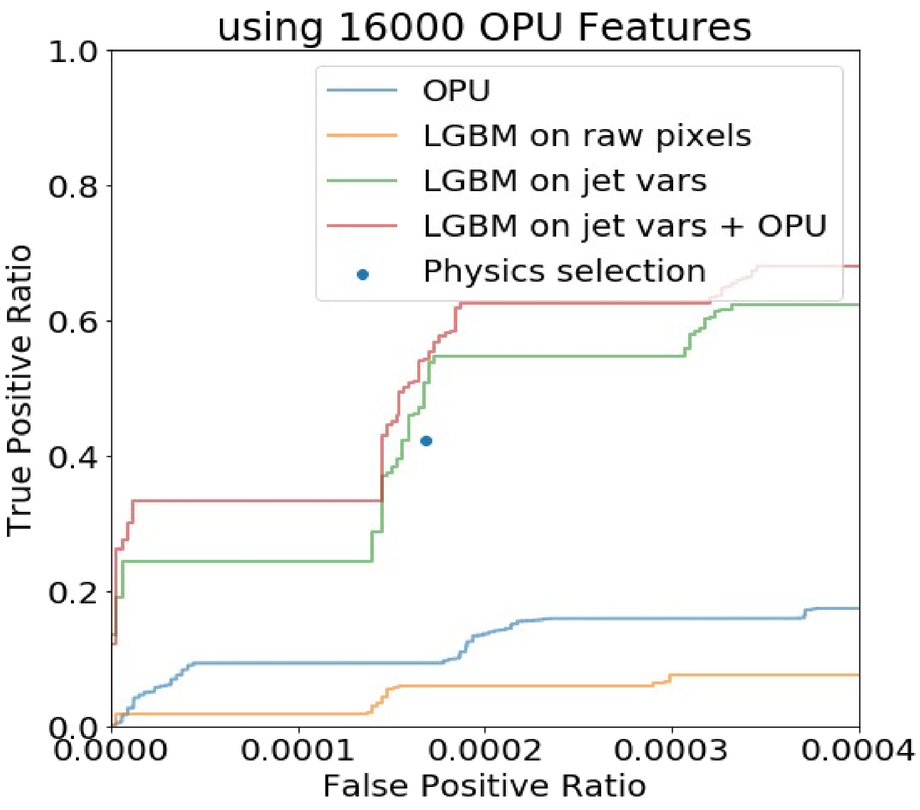
Discretization

- Intensity-based binning (3 bits/pixel)
- Linear regression: single output node neural network.



Results

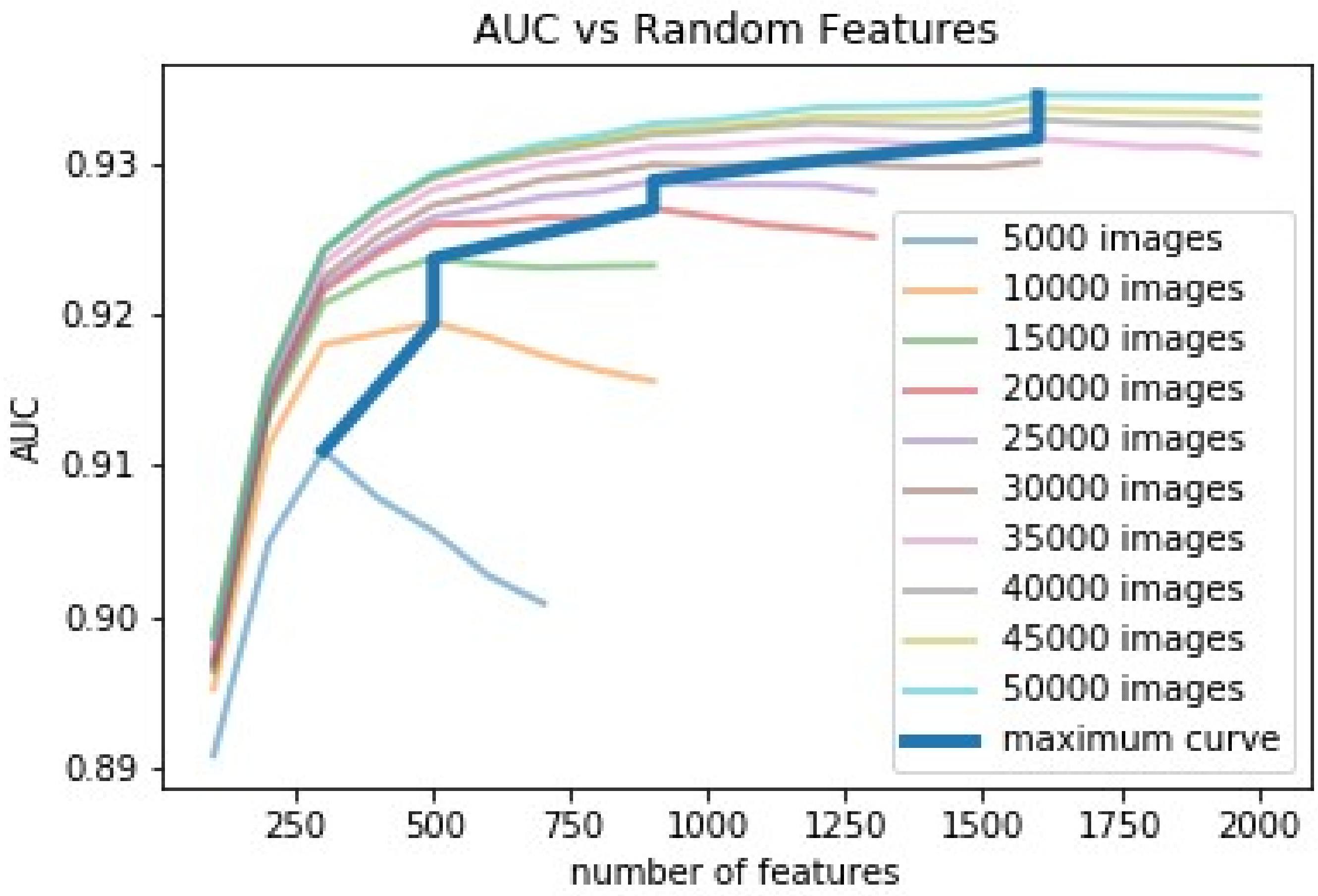




CNN results from original paper

This study

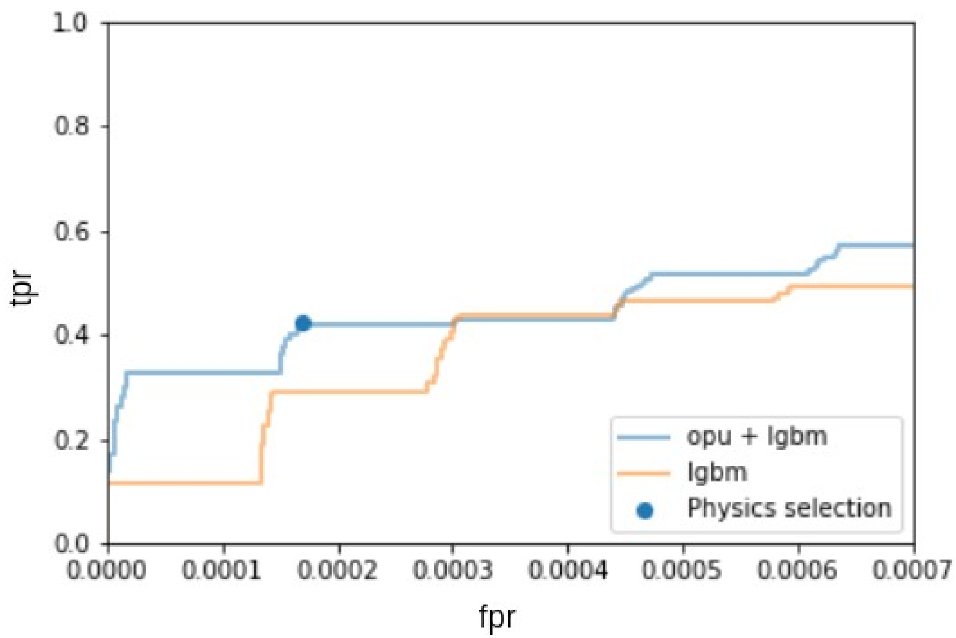
Performance vs number of images and features



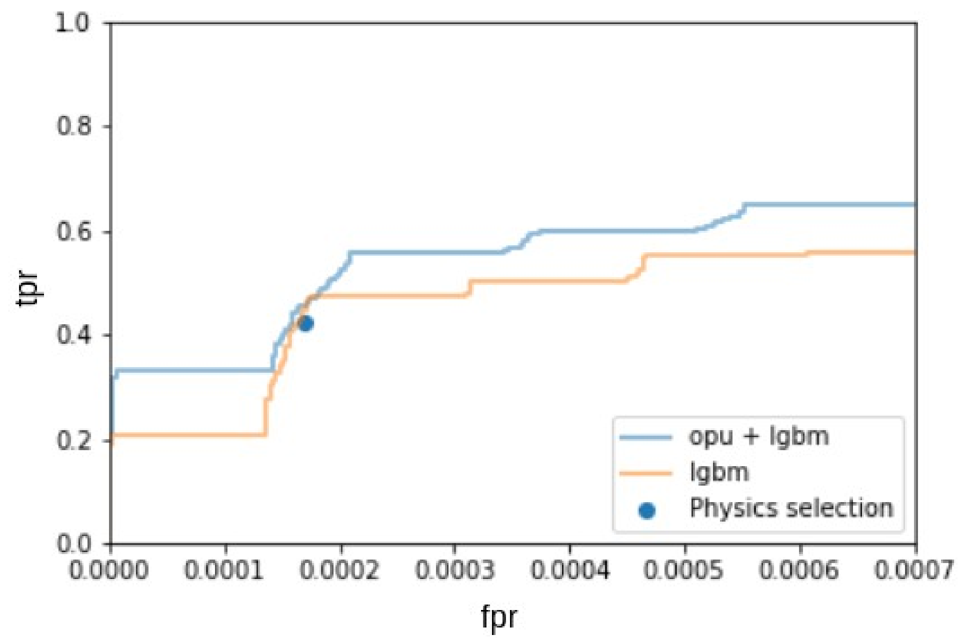
- Optimal number of features increases with number of training images
- Even low number of images allows high accuracy

Study 1 : conclusions

- OPU achieves better results on calorimetry datasets than for tracking
- Not on par with CNN and state of the art studies
- But… NN require a large amount of training data
- OPU + BDTs scalable even when $N_{events} \simeq N_{pixels}$



(a)4096 Training images



(b)8192 Training images

Study 2: following arXiv:1807.00083

Topology classification with deep learning to improve real-time event selection at the LHC

Thong Q. Nguyen, Daniel Weitekamp III, Dustin Anderson, Roberto Castello, Olmo Cerri, Maurizio Pierini, Maria Spiropulu, Jean-Roch Vlimant

- Synthetic data corresponding to W, ttbar and QCD (10 K events each), loosely inspired by the LHC running configuration in 2015-2016
- List of reconstructed PF candidates associated to charged particles, photons and neutral hadrons
- Binned in 2D arrays consisting of :
  - two barrel region (|η|<1.5 ; bin size 0.0187×0.0187)
  - two end-cap regions (1.5≤ |η| <3.0 ; bin size 0.0187×0.0187)
  - two forward regions (3.0≤ |η| <5.0 ; bin size : 0.175 in η, 0.175 to 0.35 in φ)
  - value : scalar sum of the pT of the particles in that cell.

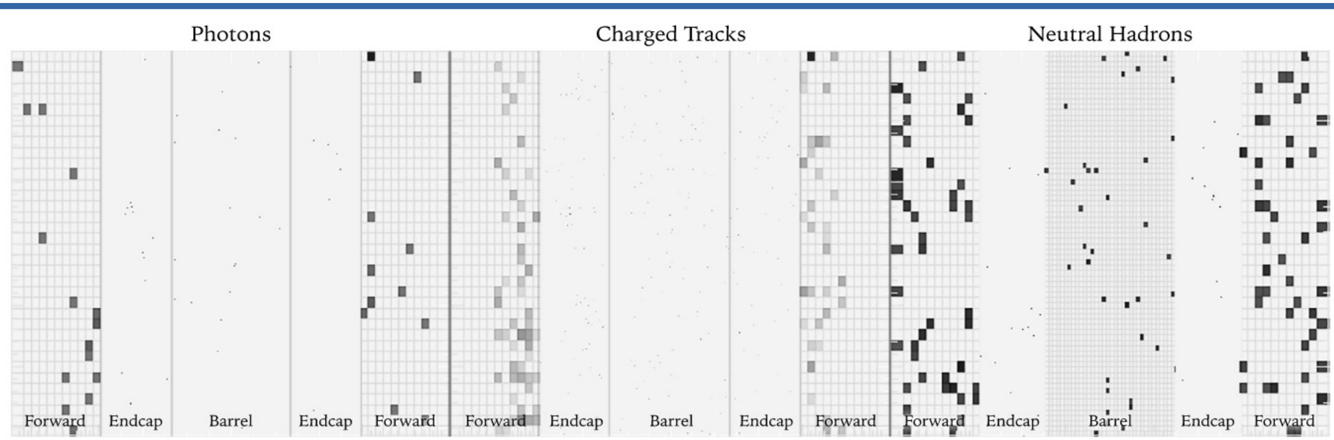


Figure 2: An example of a $t\bar{t}$ event as the input of the raw-image classifier. Vertical and horizontal axess are the ϕ and η coordinates, respectively, of the sub-detectors.

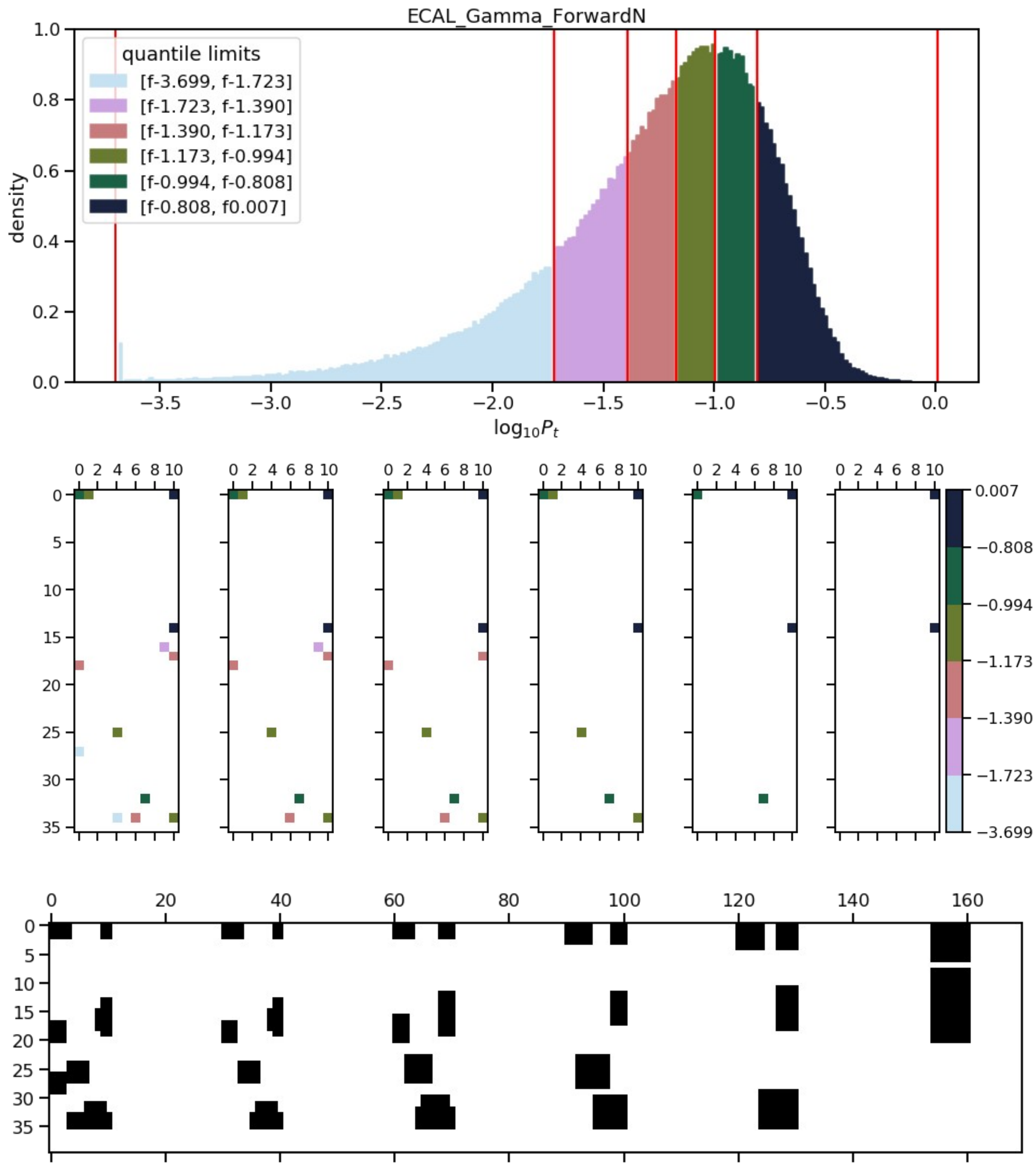
Discretization

Example of ECAL Gamma ForwardN

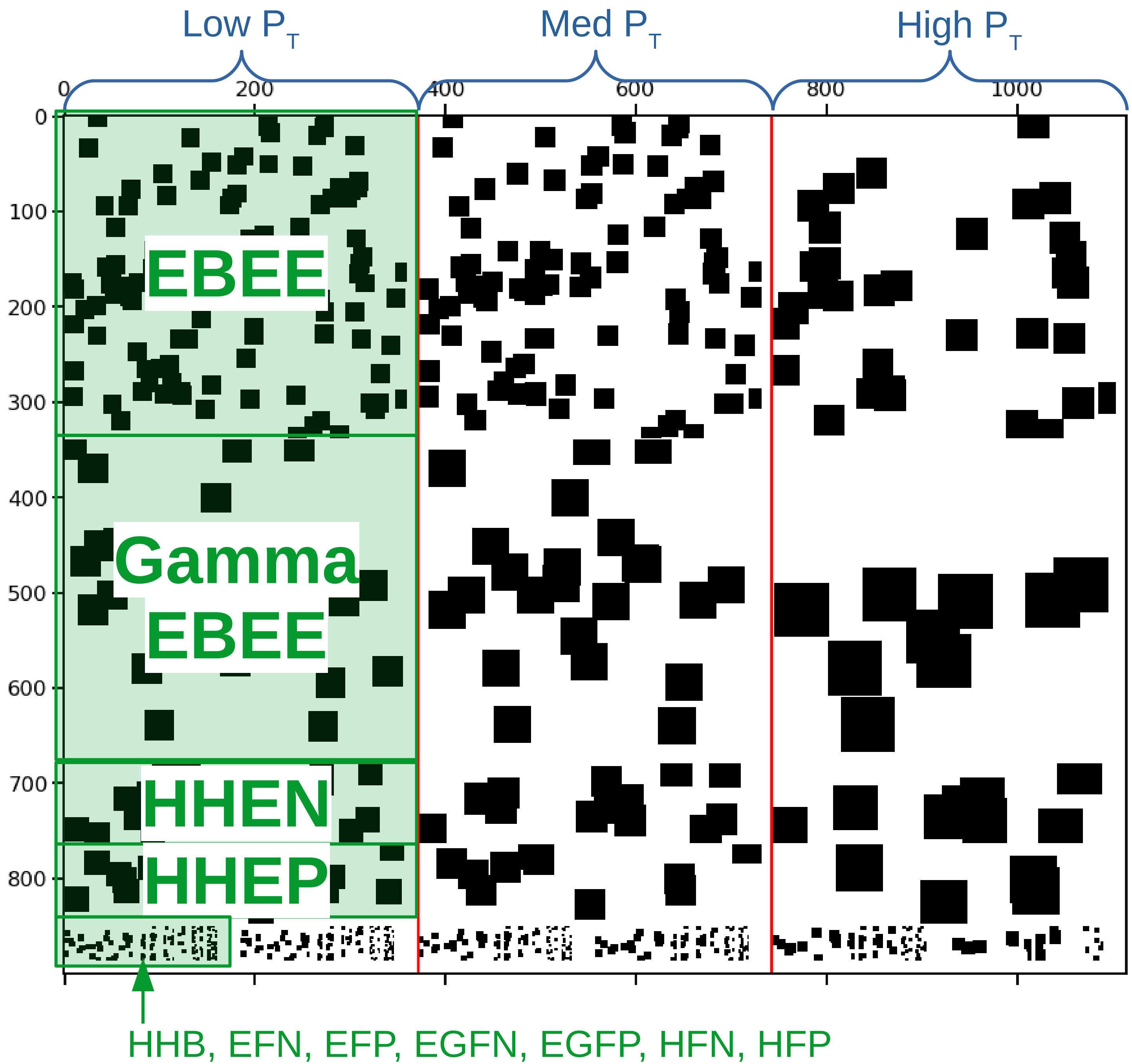
Pt distribution binned in 6 quantiles

Element in quantile n duplicated n times (OPU pixels uncorrelated)

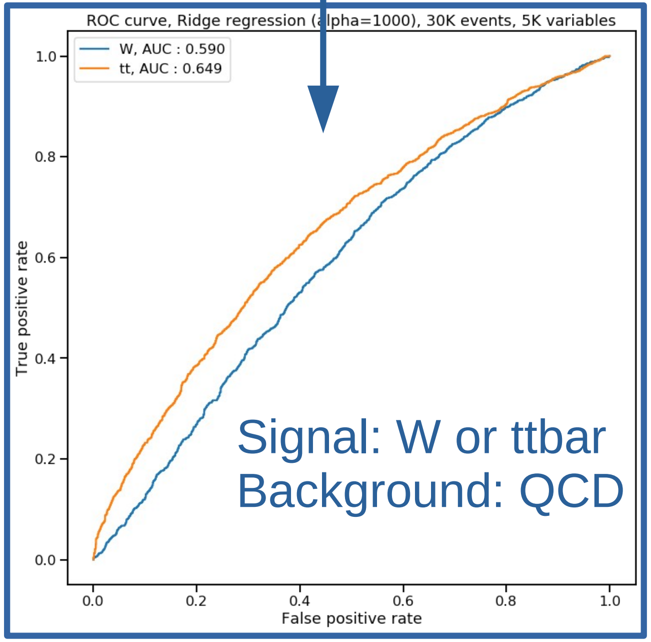
OPU performs best around 25 % hits on : custom-size filter

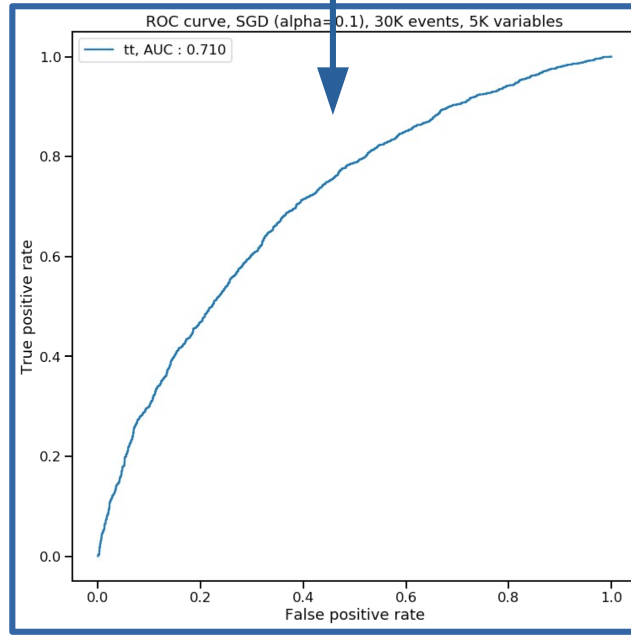


Done for all subdetectors



Ridge regression and SGD results





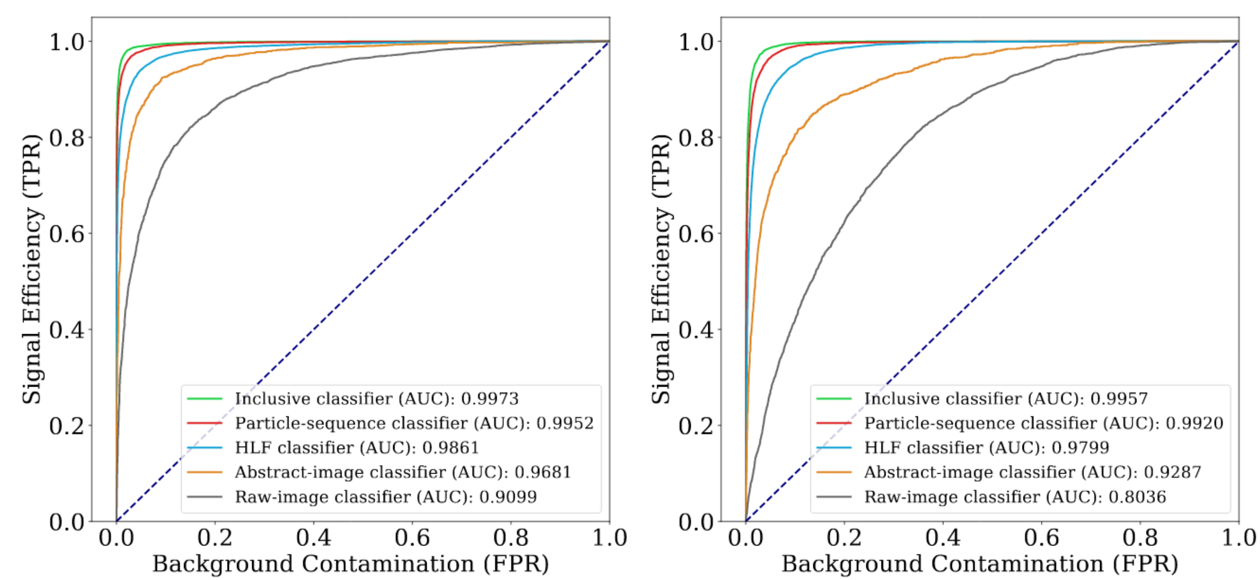


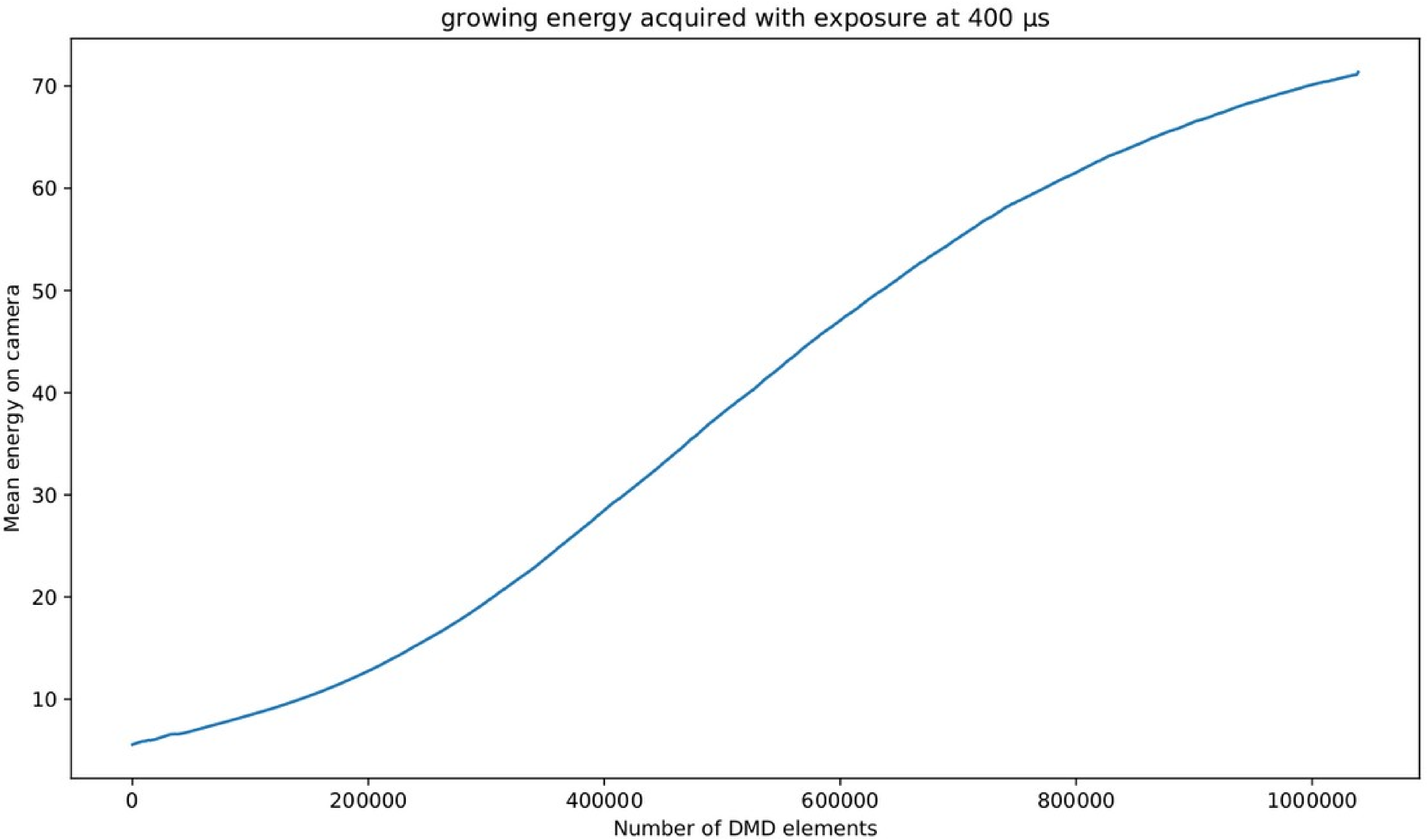
Figure 5: ROC curves for the $t\bar{t}$ (left) and W (right) selectors described in the paper.

Summary

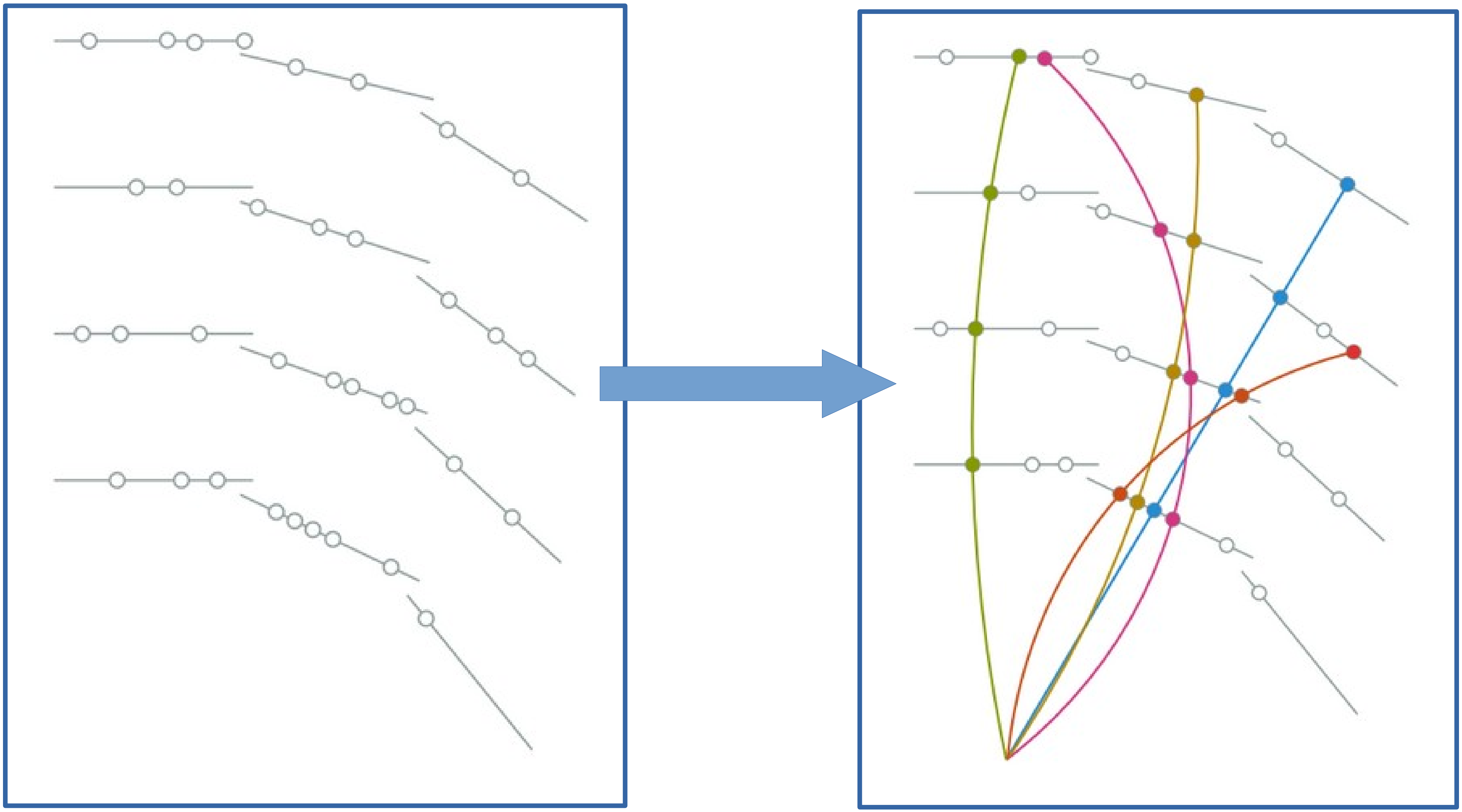
- OPU provides a device to reduce classification dimensionality / training time through physical random matrixes
- Casting a Tracking problem for OPU is hard
  - Estimations of various parameters
  - OPU « makes sense » without matching traditional methods
- Calorimetry
  - Faster training than CNNs, far less training data ($N_{features} \simeq N_{pixels}$), more robust
  - Study 2 : not as good as dedicated classifier, but decent discrimination
  - Many other possibilities for casting images on the OPU could be tried out
- In conclusion, maybe OPU has a niche application
  for frequent tries on few events,
  but casting it for HEP problems remains an open question

Backup

OPU response curve

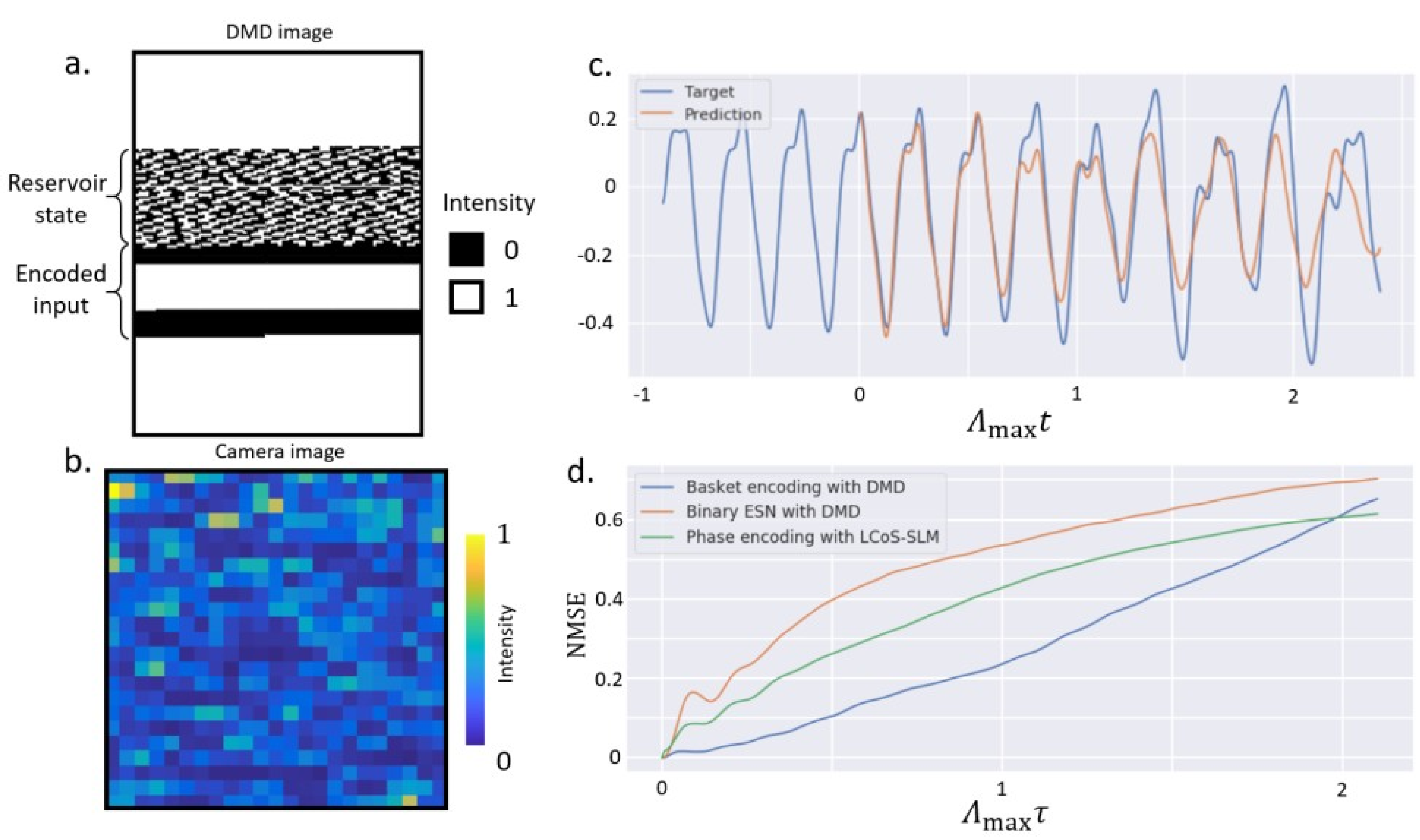


The TrackML challenge: connect the dots

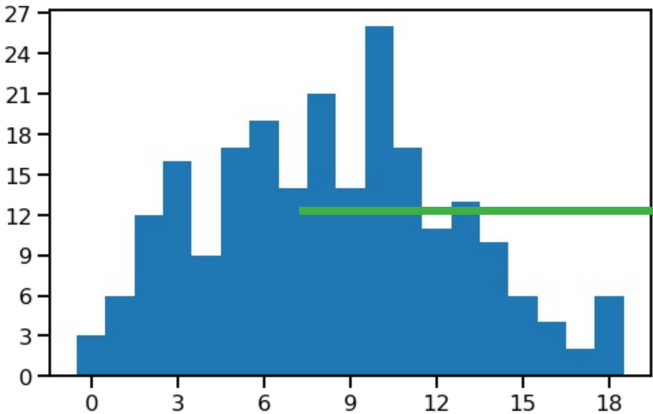


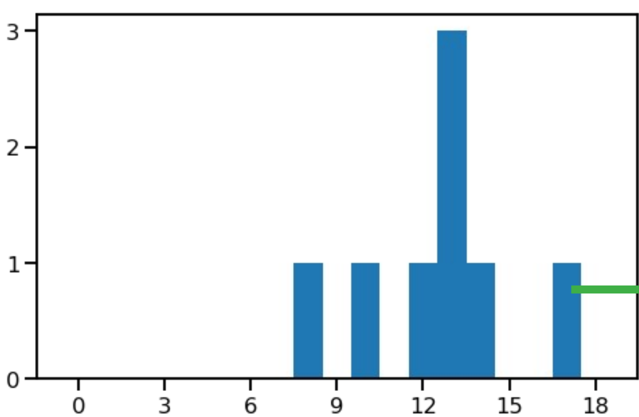
Reservoir computing

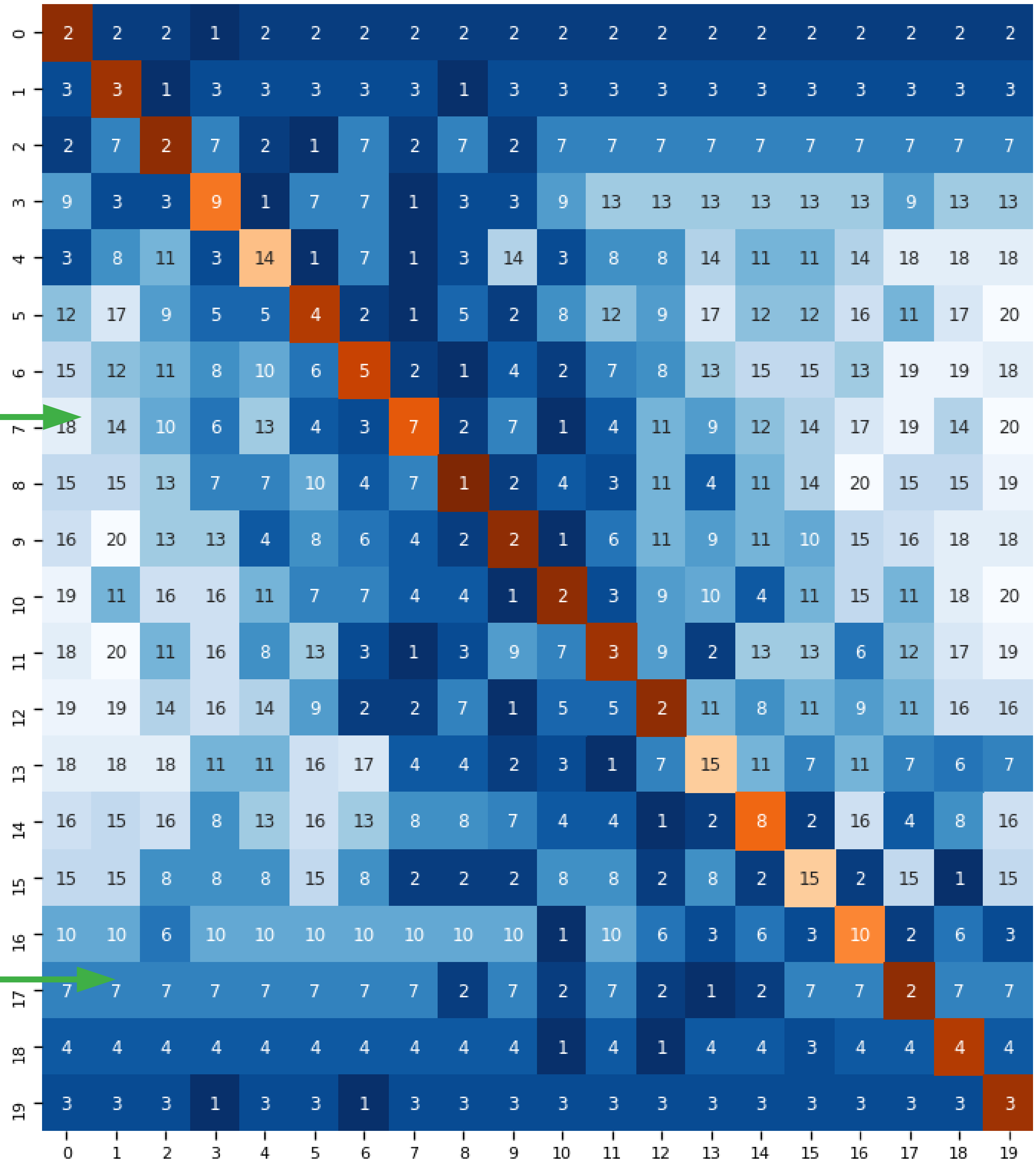
Inspired by arxiv:1907.00657, J. Dong and al. :
« Optical Reservoir Computing using multiple light scattering for chaotic systems prediction »



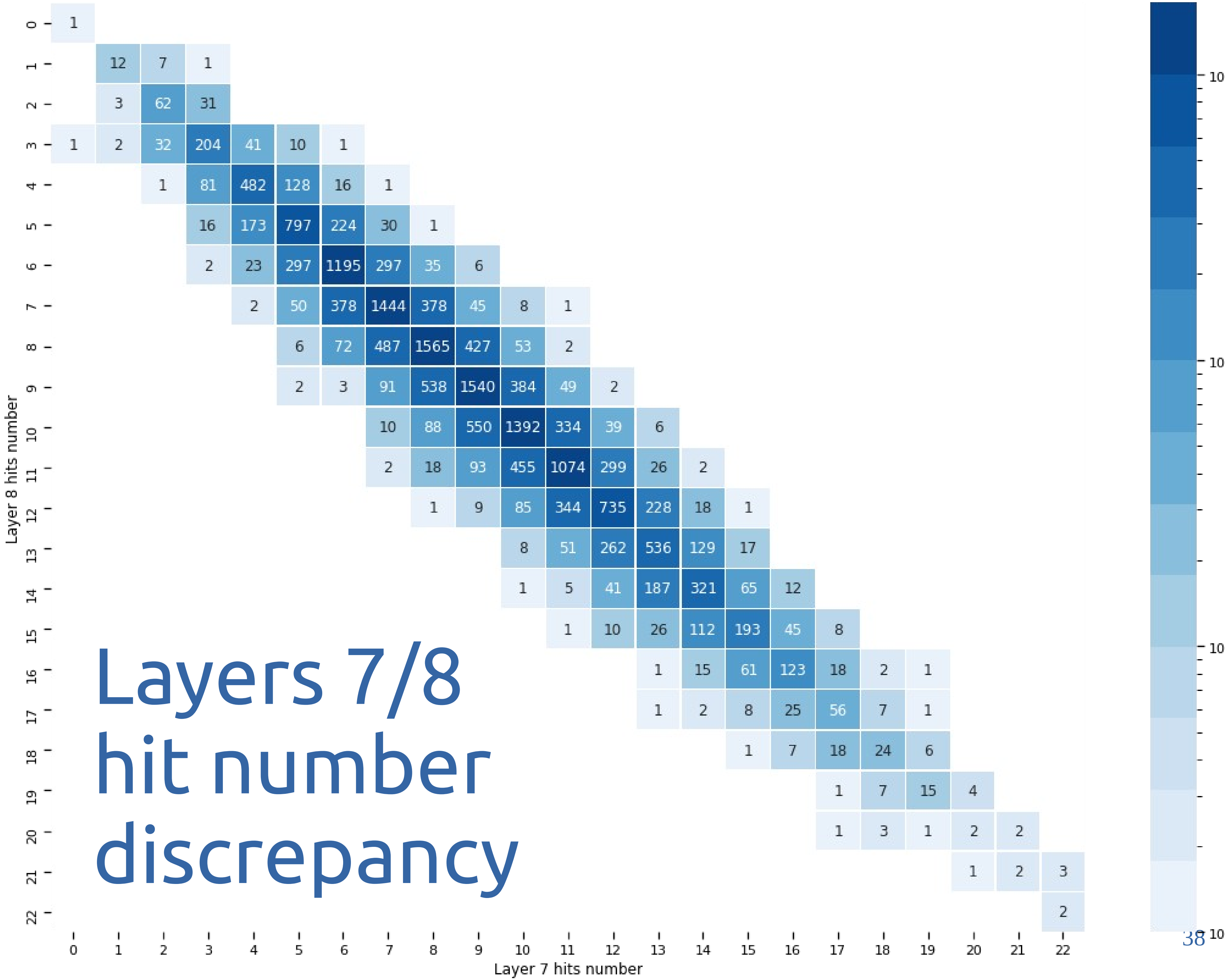
How bad is it ?


27
24
21
18
15
12
9
6
3
0
0
3
6
9
12
15
18

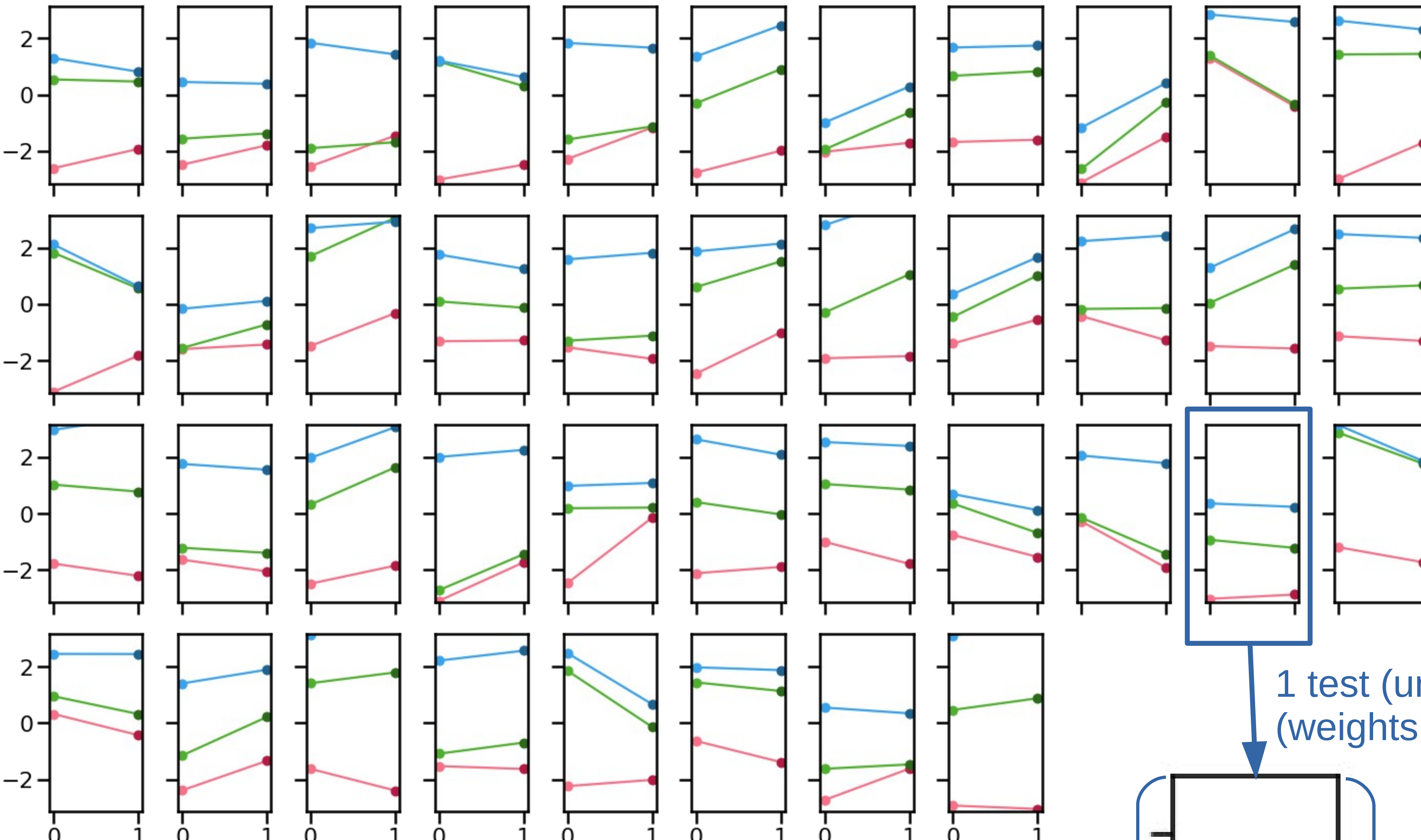

3
2
1
0
0
3
6
9
12
15
18


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
0: 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1: 3 3 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3
2: 2 7 2 7 2 1 7 2 7 2 7 7 7 7 7 7 7 7 7 7
3: 9 3 3 9 1 7 7 1 3 3 9 13 13 13 13 13 13 9 13 13
4: 3 8 11 3 14 1 7 1 3 14 3 8 8 14 11 11 14 18 18 18
5: 12 17 9 5 5 4 2 1 5 2 8 12 9 17 12 12 16 11 17 20
6: 15 12 11 8 10 6 5 2 1 4 2 7 8 13 15 15 13 19 19 18
7: 18 14 10 6 13 4 3 7 2 7 1 4 11 9 12 14 17 19 14 20
8: 15 15 13 7 7 10 4 7 1 2 4 3 11 4 11 14 20 15 15 19
9: 16 20 13 13 4 8 6 4 2 2 1 6 11 9 11 10 15 16 18 18
10: 19 11 16 16 11 7 7 4 4 1 2 3 9 10 4 11 15 11 18 20
11: 18 20 11 16 8 13 3 1 3 9 7 3 9 2 13 13 6 12 17 19
12: 19 19 14 16 14 9 2 2 7 1 5 5 2 11 8 11 9 11 16 16
13: 18 18 18 11 11 16 17 4 4 2 3 1 7 15 11 7 11 7 6 7
14: 16 15 16 8 13 16 13 8 8 7 4 4 1 2 8 2 16 4 8 16
15: 15 15 8 8 8 15 8 2 2 2 8 8 2 8 2 15 2 15 1 15
16: 10 10 6 10 10 10 10 10 10 10 1 10 6 3 6 3 10 2 6 3
17: 7 7 7 7 7 7 7 7 2 7 2 7 2 1 2 7 7 2 7 7
18: 4 4 4 4 4 4 4 4 4 4 1 4 1 4 4 3 4 4 4 4
19: 3 3 3 1 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3

Layers 7/8 hit number discrepancy

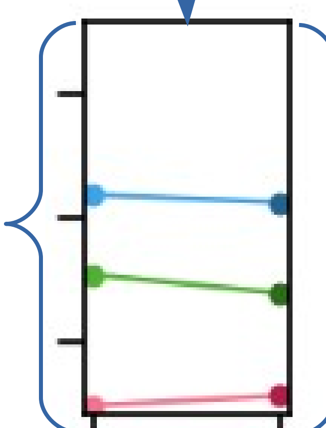


Oracle on number of hits (3)



1 test (unseen) event
(weights from training)

True L8
angle

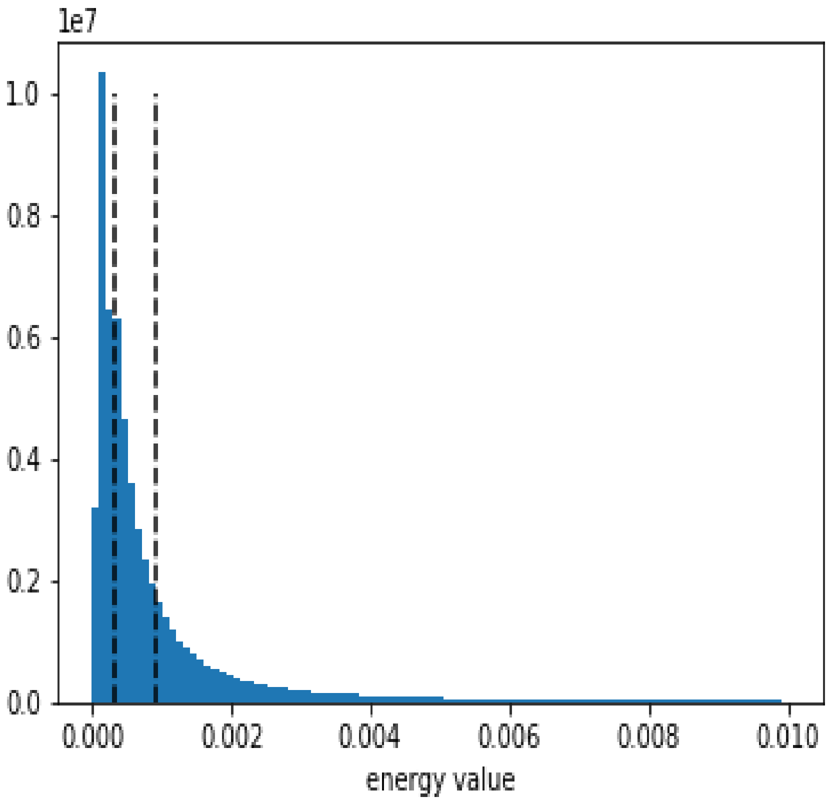


Predicted
L8 angle

$$\min_{\beta\in R^{m\times n}} ||X\beta - y|| + ||\gamma\beta||$$

(3) last layer hit position
X (20K) events

Encoding Scheme

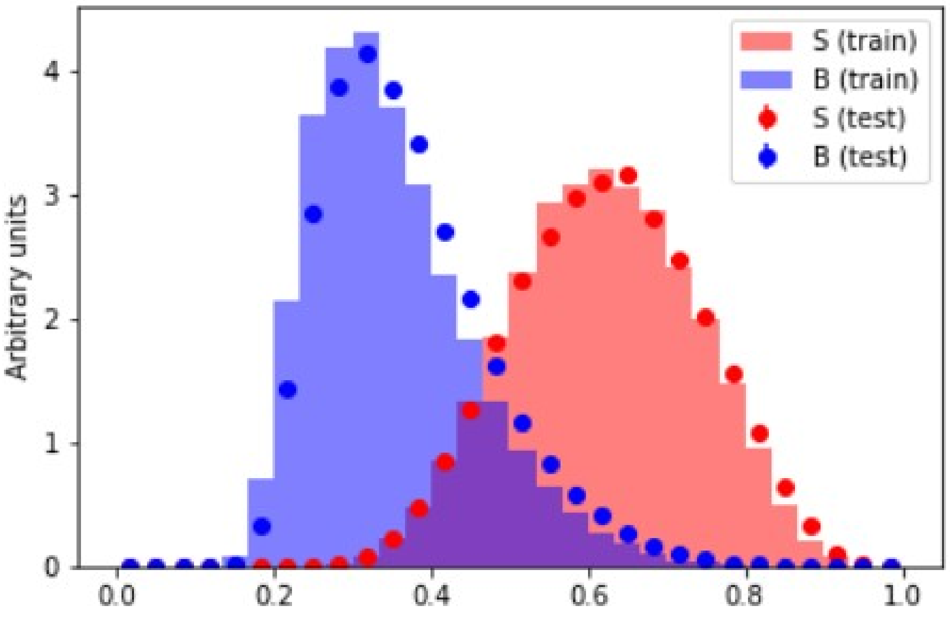


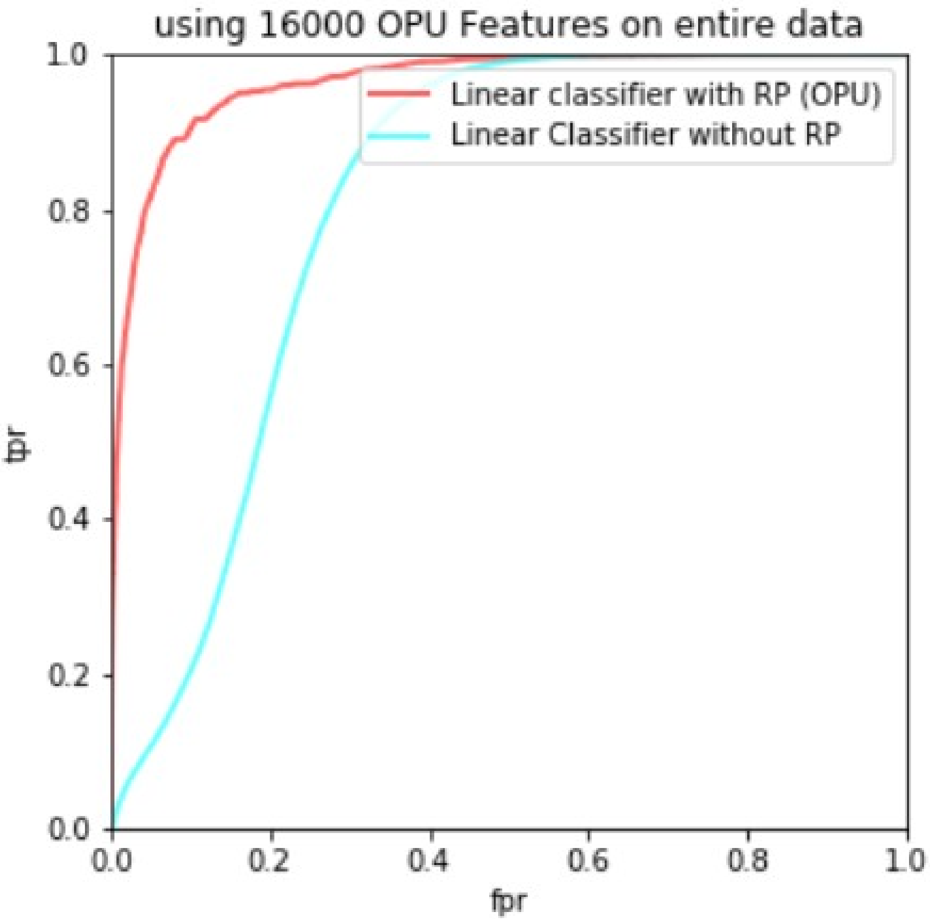
Distribution of energy (Excluding zeros)

Pixel (energy) value	encoding
x = 0	000
x>0 and x<=.00031528	001
$x>3.1528*10^{-4}$ and $x<=9.1565*10^{-4}$	011
$x>9.1565*10^{-4}$	111

The intensity based binning performed much better then auto-encoders

Predictions using OPU





Estimate next layer hits number

$$\min_{\beta \in R^{m \times n}} ||X\beta - y|| + ||\gamma\beta||$$

(10K) random features
X (10K) events

(1) last layer hit number
X (10K) events