ADRIAN BEVAN

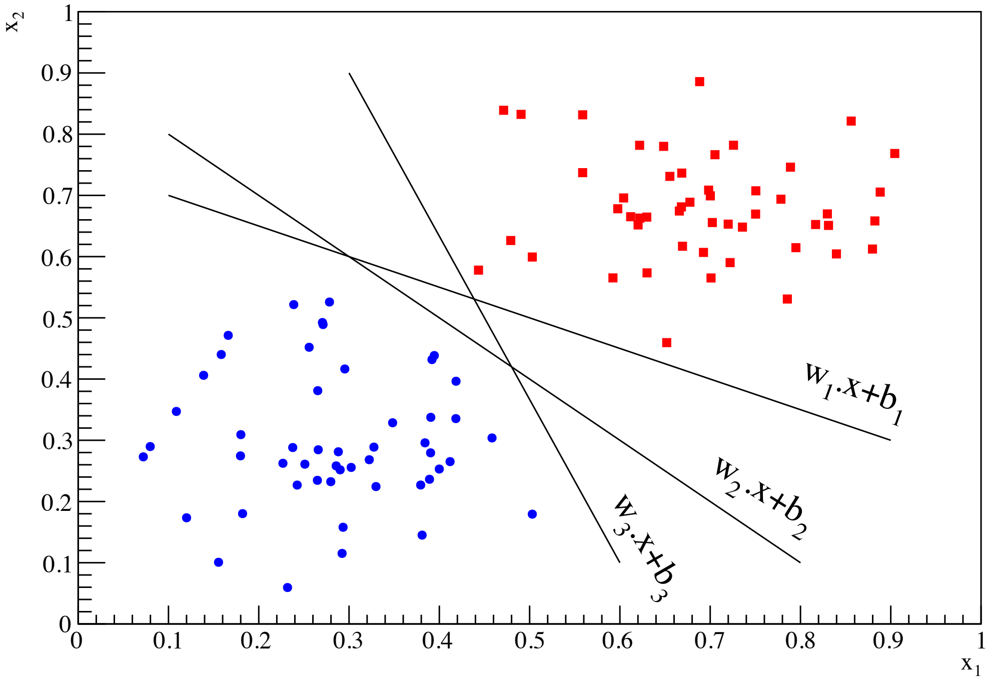
# INTRODUCTION TO MACHINE LEARNING
# PART 3

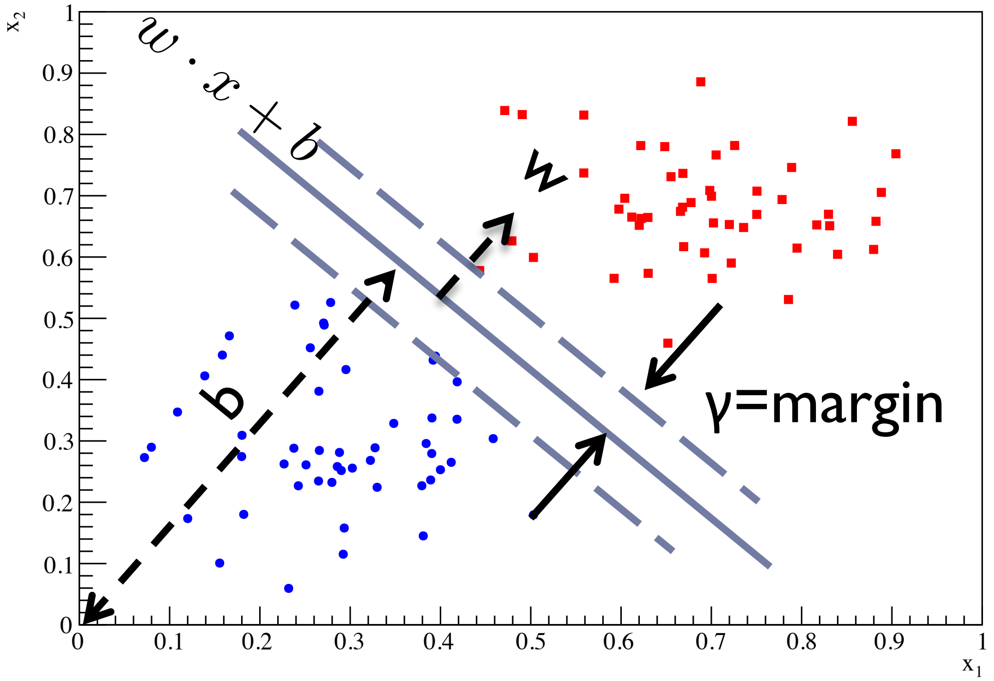
LECTURES GIVEN IN THE PARTICLE PHYSICS DIVISION AT THE RUTHERFORD APPLETON LABORATORY, MAY/JUNE 2020

# SUPPORT VECTOR MACHINES

# HARD MARGIN SVM

- Identify the support vectors (SVs): these are the points nearest the decision boundary.
- Use these to define the hyperplane that maximises the margin (distance) between the optimal plane and the SVs.





- If we can do this with a SVM – we would simply cut on the data to separate classes of event.

# HARD MARGIN SVM: PRIMAL FORM

- Optimise the parameters for the maximal margin hyperplane with:

$$\arg\min_{w,b} \frac{1}{2}||w||^2$$

- such that $y_i(w \cdot x_i - b) \geq 1$ ($y_i$ is called the functional margin)

- Equivalent to solving the following optimisation problem:

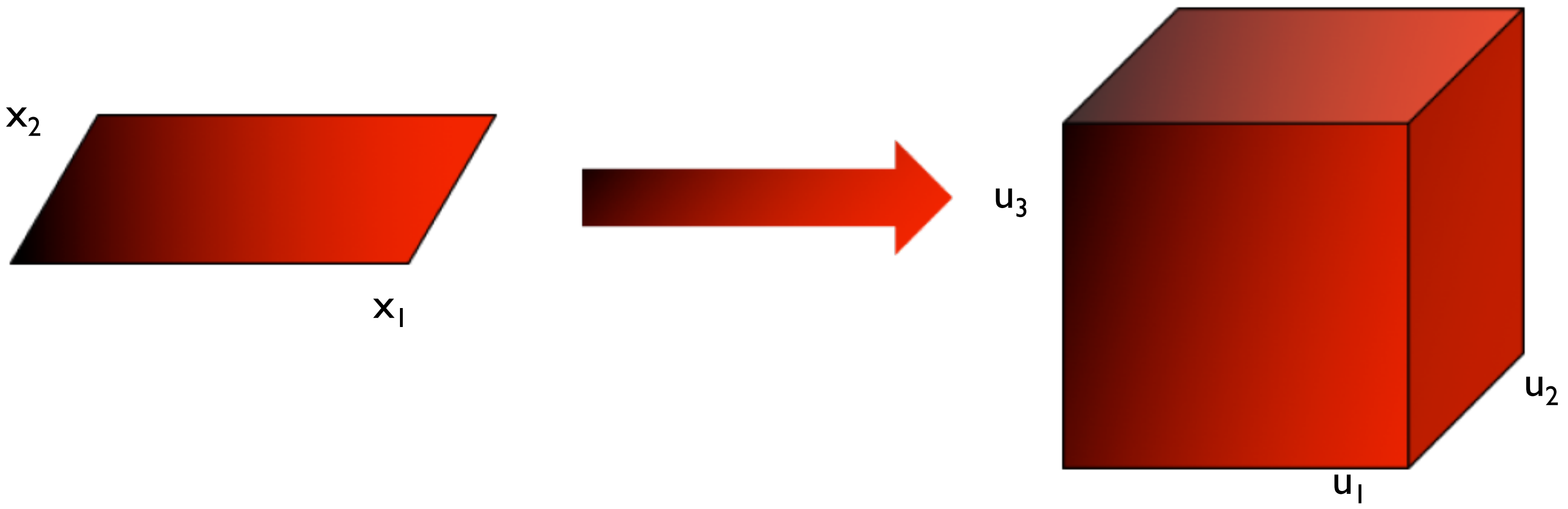
$$\arg\min_{w,b} \max_{\alpha \geq 0} \left[ \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w \cdot x_i - b) - 1] \right]$$

- Where: $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ and $b = \frac{1}{N_{SV}} \sum_{i=1}^{n} (w \cdot x_i - y_i)$

# HARD MARGIN SVM: KERNEL FUNCTIONS

- We can introduce the use of a Kernel Function (KF) to implicitly map from our input feature space X to some potentially higher dimensional dual feature space F.
- Define the function: $K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$

$X \in \{x_1, x_2\}$



$F \in \{u_1, u_2, u_3\}$

- We don't need to know the details of the mapping; this is the "kernel trick”. B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, 2002.*

# HARD MARGIN SVM: KERNEL FUNCTIONS

- We can introduce the use of a Kernel Function (KF) to implicitly map from our input feature space X to some potentially higher dimensional dual feature space F.
- Define the function: $K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$

e.g.

$$x \in \mathbb{R}^n \longrightarrow F \in \{\phi(x)|x \in X\}$$

- We don't need to know the details of the mapping; this is the "kernel trick". B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, 2002.*

# HARD MARGIN SVM: DUAL FORM

- The problem can be solved in the dual space by minimising the Lagrangian for the Lagrange multipliers $\alpha_i$ :

$$\widetilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

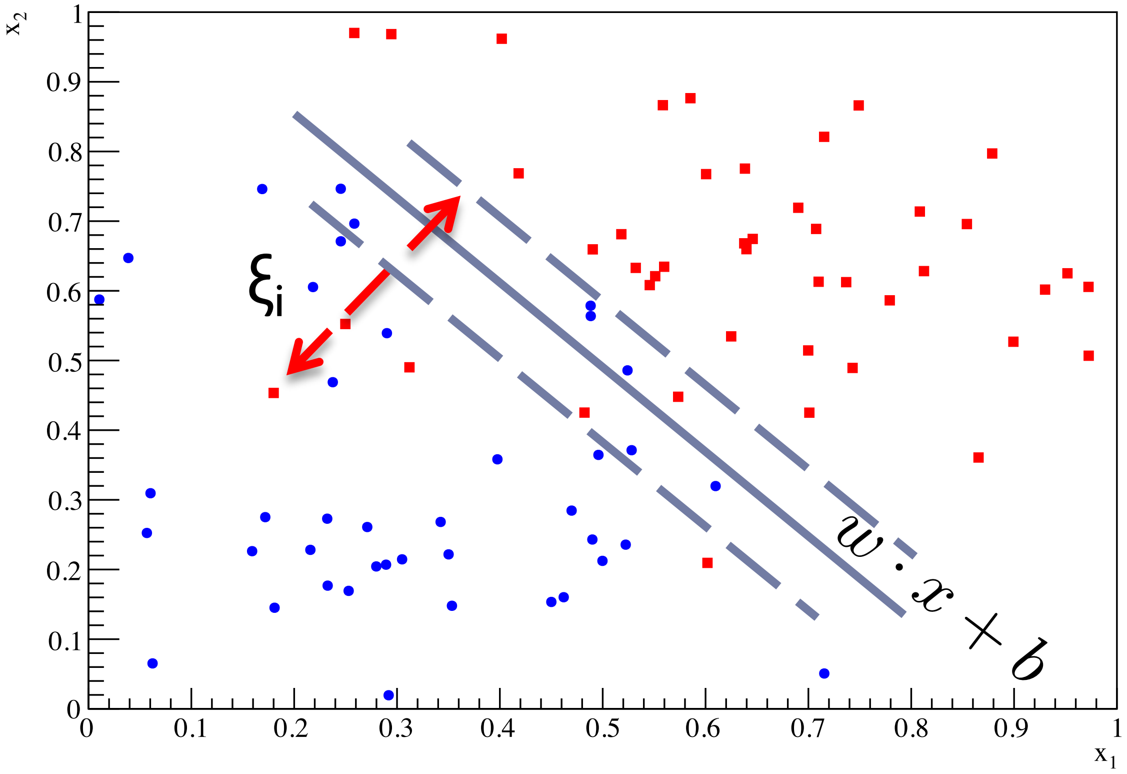
$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j).$$

Dot product KF

- Such that: $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$ .
- $\alpha_i$ are non-zero for SVs only.
- The sum provides a constraint equation for optimisation.

# SOFT MARGIN SVM

- Relax the hard margin constraint by introducing mis-classification:
  - Describe by slack (ξi) and cost (C) parameters.
  - Alternatively describe mis-classification in terms of loss functions.
  - These are iust wavs to describe the error rate.



ξi = distance between the hyper-plane defined by the margin and the ith SV (i.e. now this is a mis-classified event).

Cost (C) multiplies the sum of slack parameters in optimisation.

MVA architecture complexity is encoded by the KF.

- These are much more useful!

# SOFT MARGIN SVM

- The Lagrangian to optimise simplifies when we introduce the slack parameters:

$$\widetilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

- Where $0 \leq \alpha_i \leq C$

- and as before we constrain:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

The optimisation problem in dual space is essentially the same for the hard and soft margin SVMs.

- The algorithm is designed to focus on reducing the impact of misclassified events; again using those closest to the decision boundary to determine that boundary.

# KERNEL FUNCTIONS

- The KF, K(x,y), extends the use of inner products on data in a vector space to a transformed space where

$$K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$$

- The book by
  - *Nello Cristianini and John Shawe-Taylor, called Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000 (and references therein)*
- *discusses a number of KFs and the conditions required for these to be valid in the geometrical representation that SVMs are constructed from.*
- Here I'll focus on the main points and give a few examples of KFs (ones that are implemented in TMVA).

# KERNEL FUNCTIONS: RADIAL BASIS FUNCTION (RBF)

- Commonly used KF that maps the data from X to F.

- Distance between two support vectors is computed and used as an input to a Gaussian KF.

- For two data x and y in X space we can compute K(x, y) as

$$\boldsymbol{K(x,y) = e^{-||x-y||^2/\sigma^2}}$$

- One tuneable parameter in mapping from X to F; given by $\Gamma = 1/\sigma^2$.

# KERNEL FUNCTIONS: MULTI GAUSSIAN KERNEL

- Extend the RBF function to recognise that the bandwidth of data in problem space can differ for each input dimension; i.e. the norm of the distance between two support vectors can result in loss of information.

- Overcome this by introducing a $\Gamma_i=1/\sigma_i$ for each dimension:

$$K(x, y) = \prod_{i=1}^{\dim(X)} e^{-||x_i - y_i||^2/\sigma_i^2}$$

- Down side … we increase the number of parameters that need to be optimally determined for the map from X to F.

# KERNEL FUNCTIONS: MULTI GAUSSIAN KERNEL

- The multi-gaussian kernel does not include off-diagonal terms that would allow for accommodation of correlations between parameters.
  - De-correlate the input feature space to overcome this deficiency, or alternatively one could implement a variant of this kernel function using:

$$K(x,y) = e^{-(x-y)^T \Sigma^{-1} (x-y)}$$

  - Here Σ is an n x n matrix corresponding to the covariance matrix for the problem.
  - However this would be very computationally expensive to optimise (and is **not** implemented in TMVA).

# KERNEL FUNCTIONS: POLYNOMIAL

- There are many different types of polynomial kernel functions that one can study.
- A common variant is of the form:

$$K(x, z) = (\langle x \cdot z \rangle + c)^d = \left( \sum_{i=1}^{\ell} x_i z_i + c \right)^d$$

- c and d are tuneable parameters.
- The sum is over support vectors (i.e. events in the data set for a soft margin SVM).

# KERNEL FUNCTIONS: PRODUCTS AND SUMS

- Valid (Mercer) kernels satisfy Mercer's conditions(*). This allows us to construct new kernels from known Mercer kernels that are products and sums.
  - The sum of Mercer KFs is a valid KF.
  - The product of Mercer KFs is a valid KF.

* Mercer's conditions require that the Gramm matrix formed from SVs is positive semi-definite. This is a consequence of the geometric interpretation of SVMs given x is real. Modern extensions of the SVM construct allow for complex input spaces, and for example can be based on Clifford algebra to accommodate this extension.

Complex input spaces are of interest for electronic engineering problems.

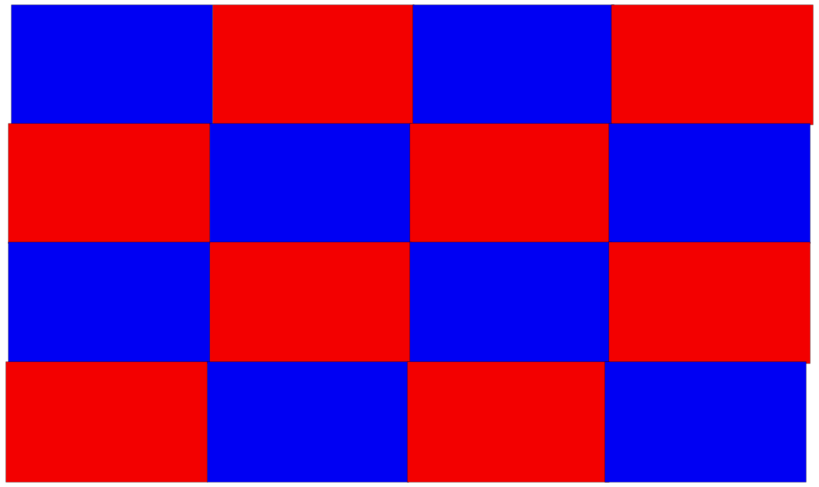
N.B. It is conceivable that one could be interested in using these if an amplitude analysis were to be written using SVMs to directly extract phase and magnitudes… but that could also be incorporated by mapping the complex feature space element into a doublet of reals.

J. Mercer. Phil.Trans.Roy.Soc.Lond., A209:415, 1909.

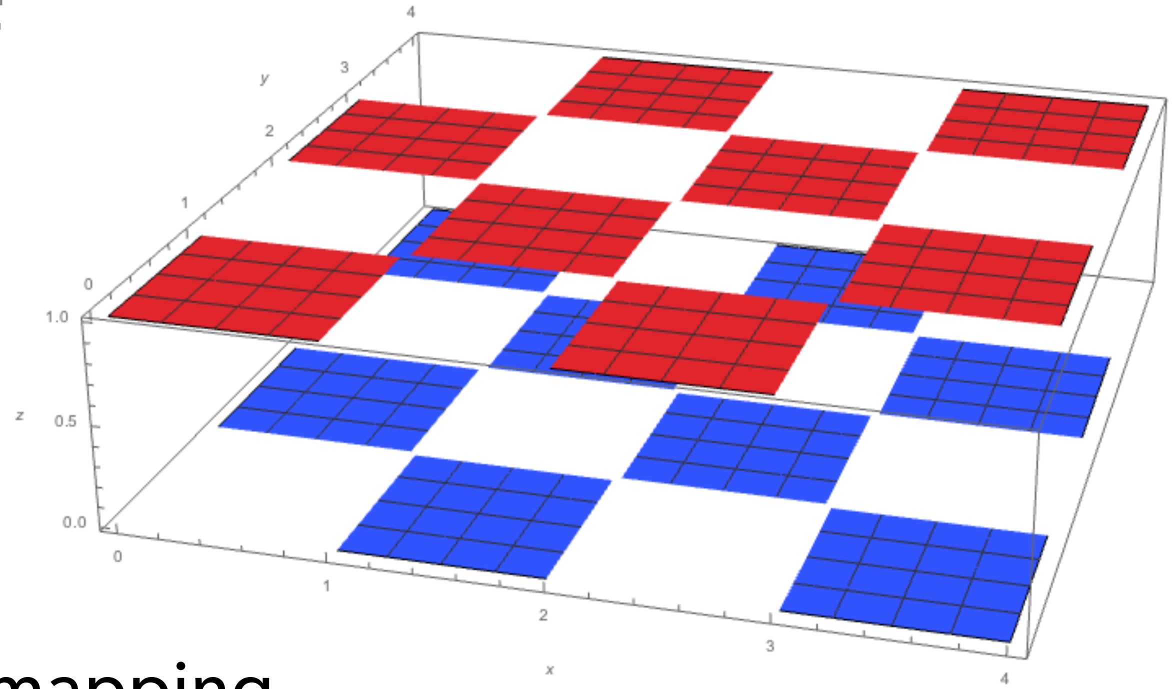
# EXAMPLES: CHECKER BOARD

- Generate squares of different colour.
- Use SVM to classify the pattern into +1 and −1 targets.
- Hard margin SVM problem; but can solved for using soft margin SVM.
- Not easy to solve in 2D (x, y) with a linear discriminant, but e.g. a 3D space of (x, y, colour) allows us to separate the squares.

$X \mapsto F$



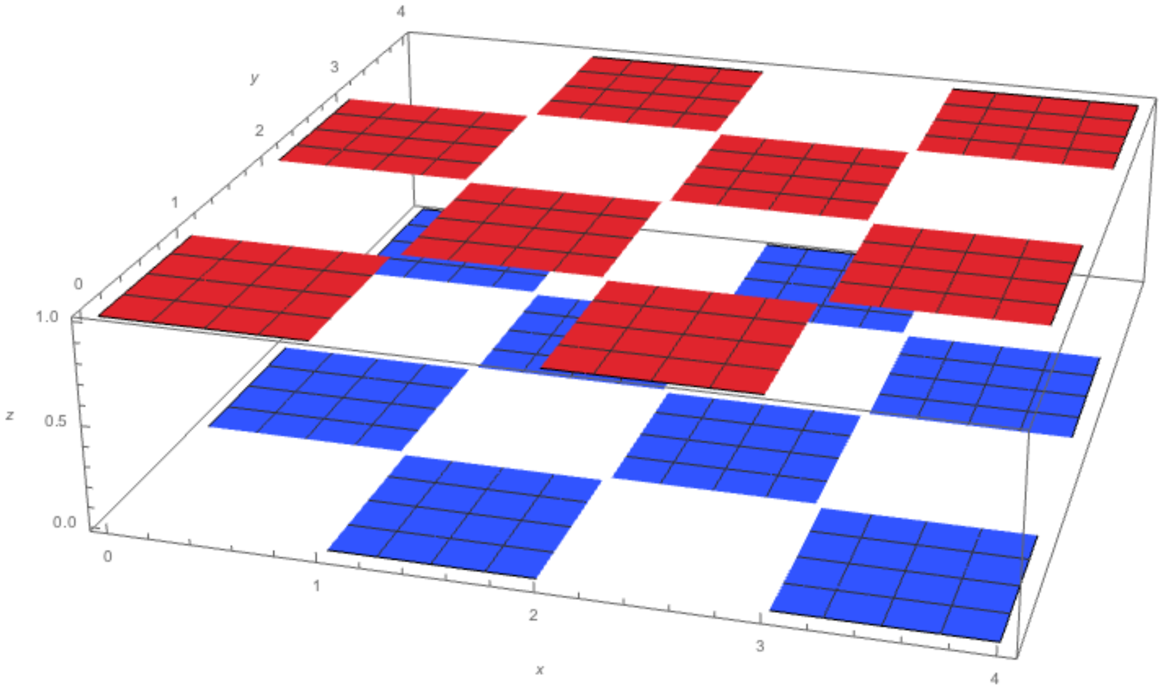
$\mapsto$



- Want to find a KF that approximates this mapping.

# EXAMPLES: CHECKER BOARD

- Generate 1000 events in the blue and red squares and give each event x and y values.



This is the ideal feature space that we would like to implicitly map into.

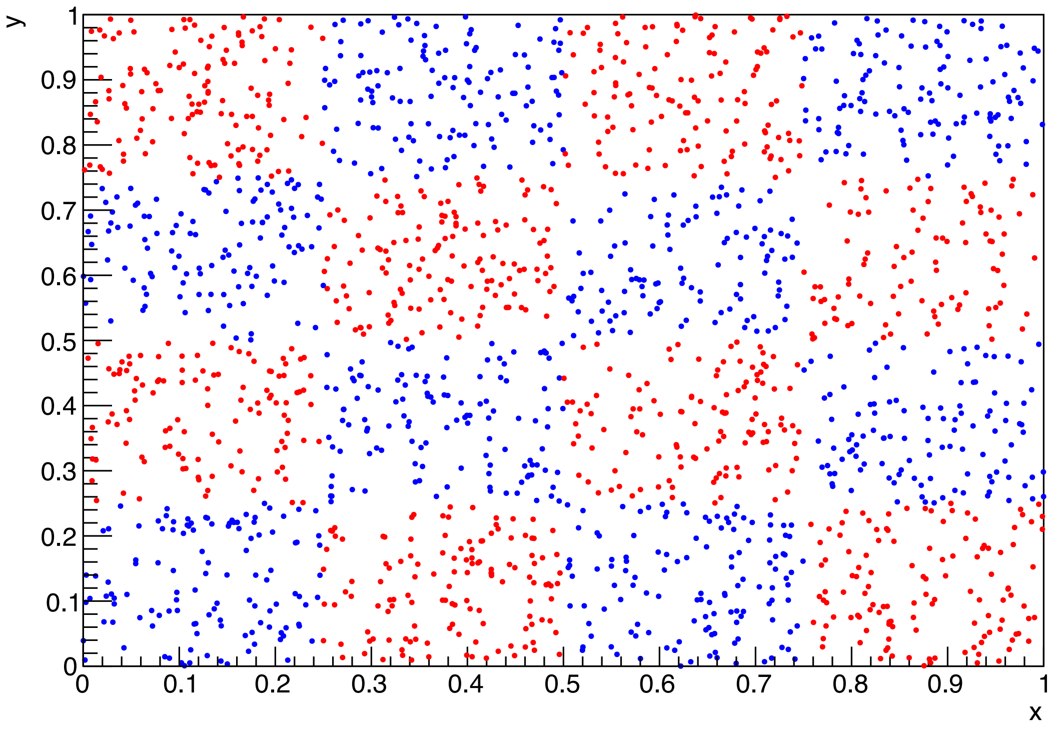
Because we implicitly do the mapping via choice of KF, in practice we don't explicitly map into this space; but we implicitly map into another space that we hope will be approximately topologically equivalent.

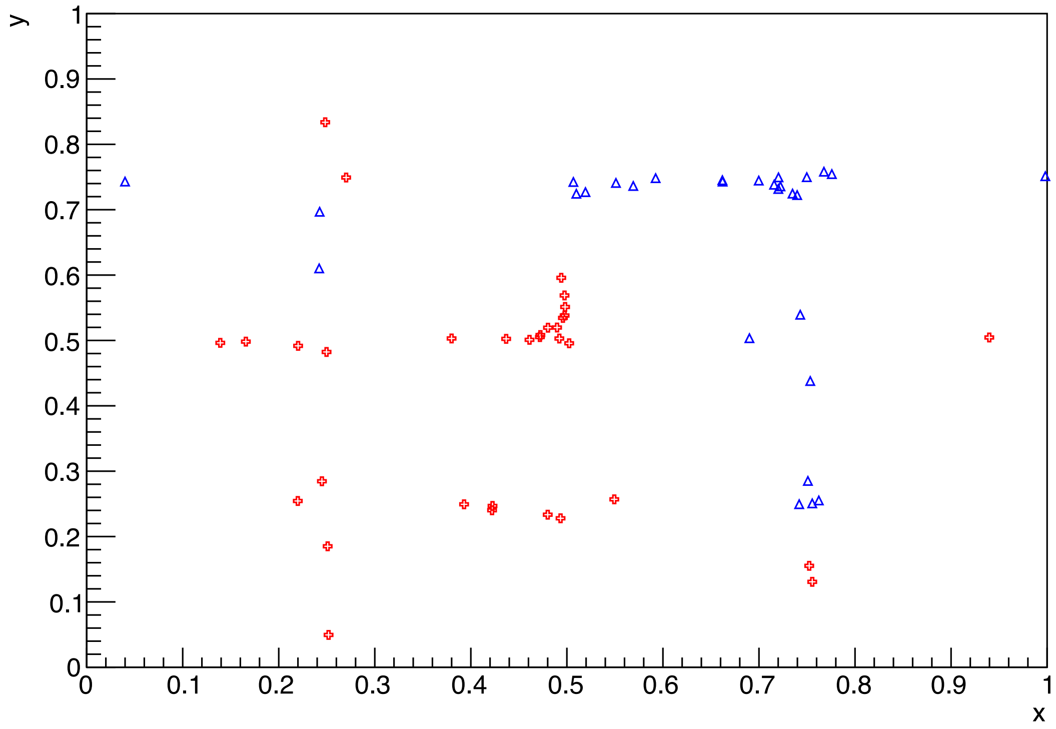
- e.g. Use a multi-Gaussian kernel function with $\Gamma_1=1$, $\Gamma_2=2$ and cost of $10^4$ (not optimised) to see what separation we can obtain.

# EXAMPLES: CHECKER BOARD

- Correctly classified events

Incorrectly classified events



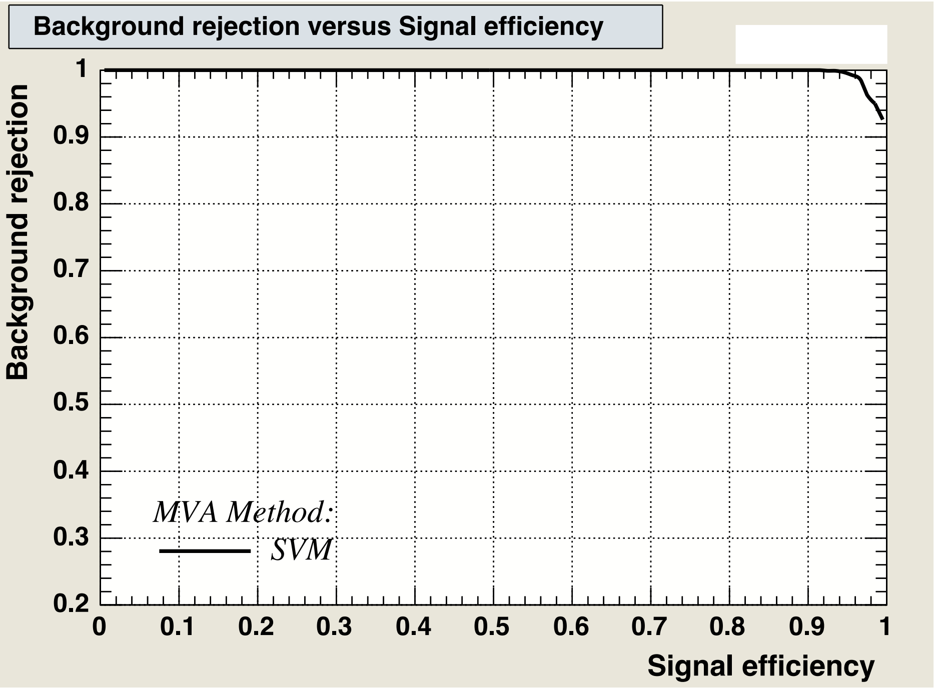


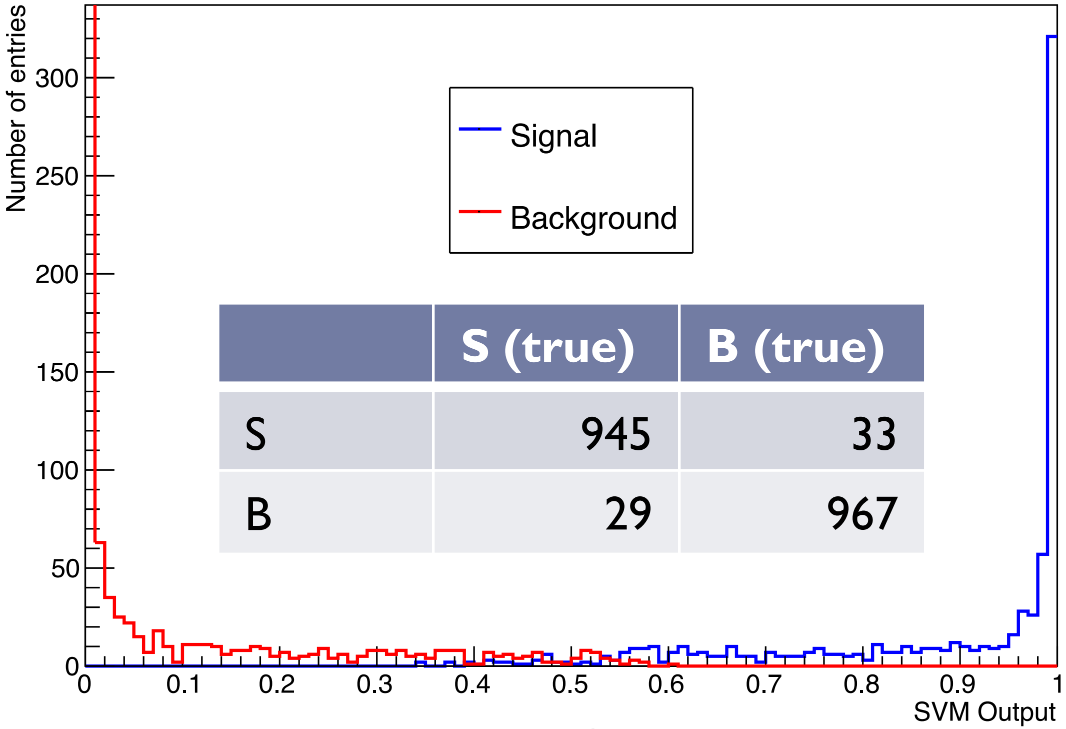
- Signal mis-classification rate ~3.3%.

- Background mis-classification rate ~3.7%.

# EXAMPLES: CHECKER BOARD

- The confusion matrix ([in-]correctly classified events) for this example shows a high level of correct classification:





| | S (true) | B (true) |
|---|---|---|
| S | 945 | 33 |
| B | 29 | 967 |

- This SVM does a good job of separating signal from background.
- An optimised output would provide a better solution.
- BDTs and NNs work well with this kind of problem as well.

# EXAMPLES: CHECKER BOARD

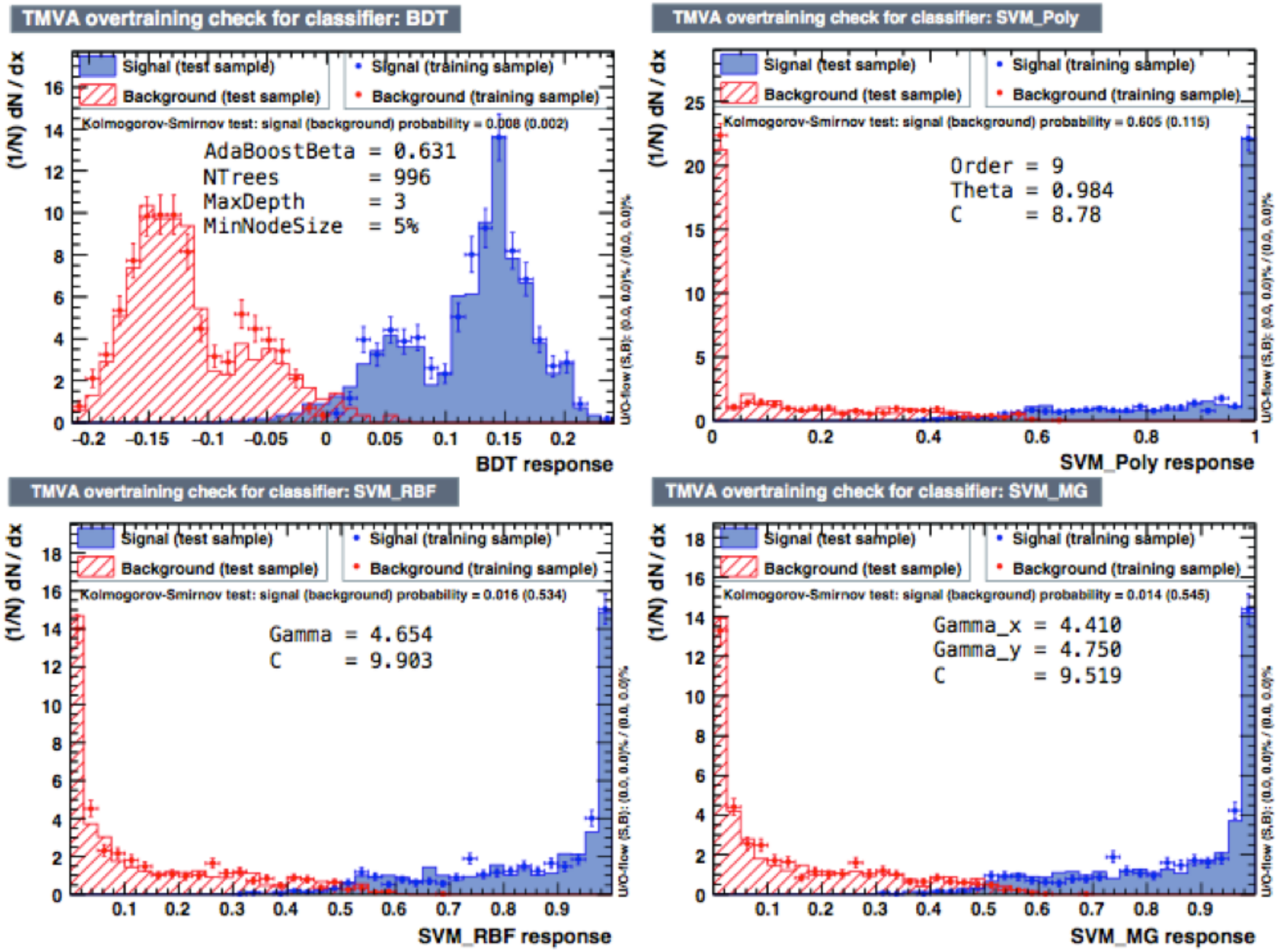
- Optimised results for comparison: Very similar responses.

BDT

SVM Polynomial (1)

SVM RBF (2)

SVM Multi-Gaussian (3)



Trained using the hold out method of cross validation (what is normally done in TMVA), with optimised hyper-parameters.

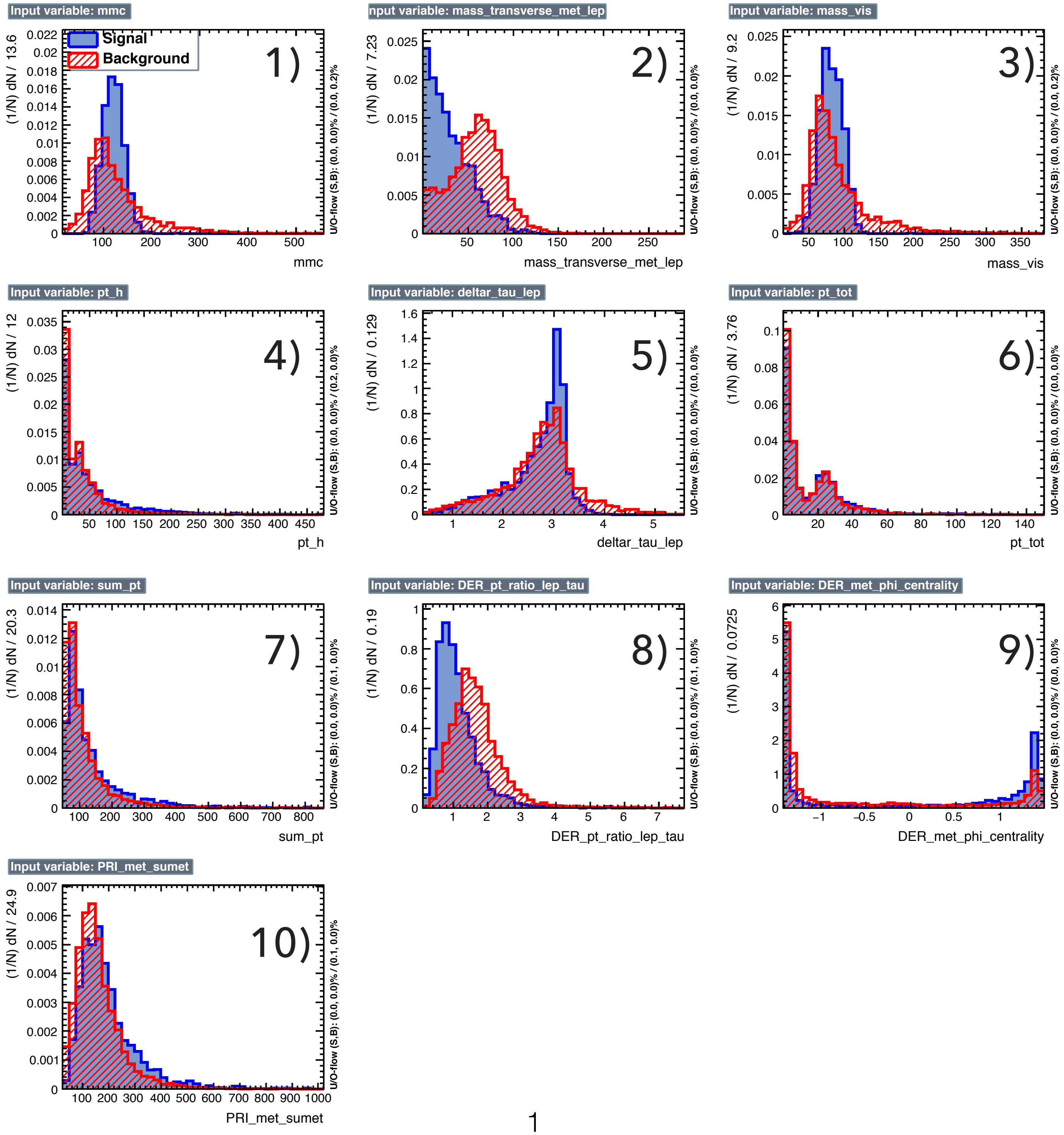
# EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- Use the Kaggle data challenge sample of signal and background events. LHC data (from ATLAS).
- Packaged up in a convenient format (CSV file).
- Sufficient description of variables provided for non-HEP users to apply machine learning (ML) techniques to HEP data.
- Real application to compare performance for different KFs and different MVAs.

https://www.kaggle.com/c/higgs-boson

# EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- Use 10 variables as inputs; 20K events.



1) MMC
2) transverse mass between MET and lep
3) Visible invariant mass of H
4)p $_T$(H)
5) R between $\tau_{had}$ and lepton
6) $p_T$(tot)
7) $\Sigma p_T$
8) $p_T$(lepton)/$p_T$(had $\tau$)
9) MET $\phi$ centrality
10) $ET_{total}$

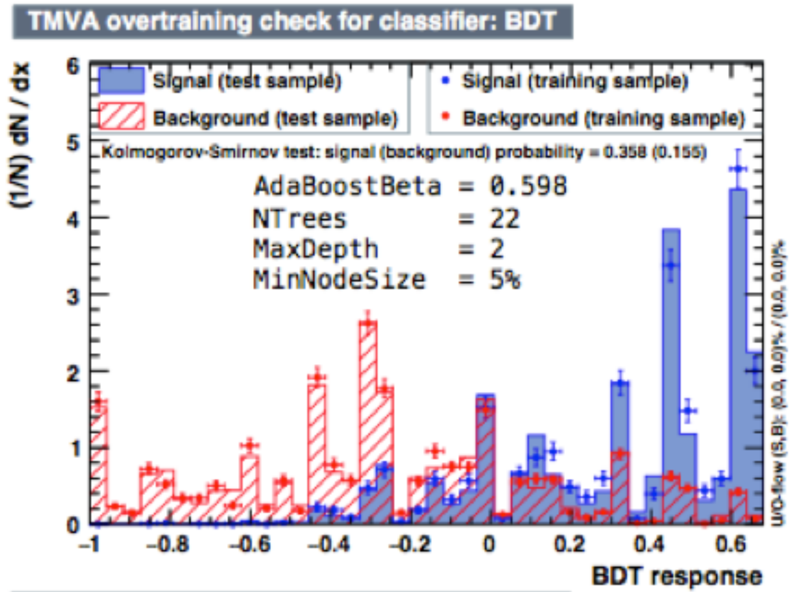
This selection of variables is not optimised, and is selected in order to show a physics example for illustrative purposes.

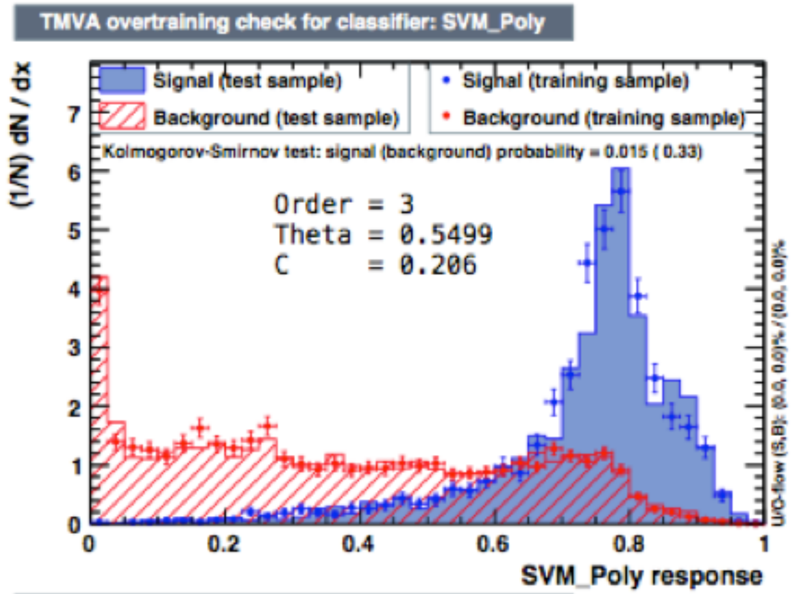
# EXAMPLES: H → $\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- NOTE: this is an illustrative example – not a fully optimised analysis of the sample; hyper-parameters are optimised.

BDT

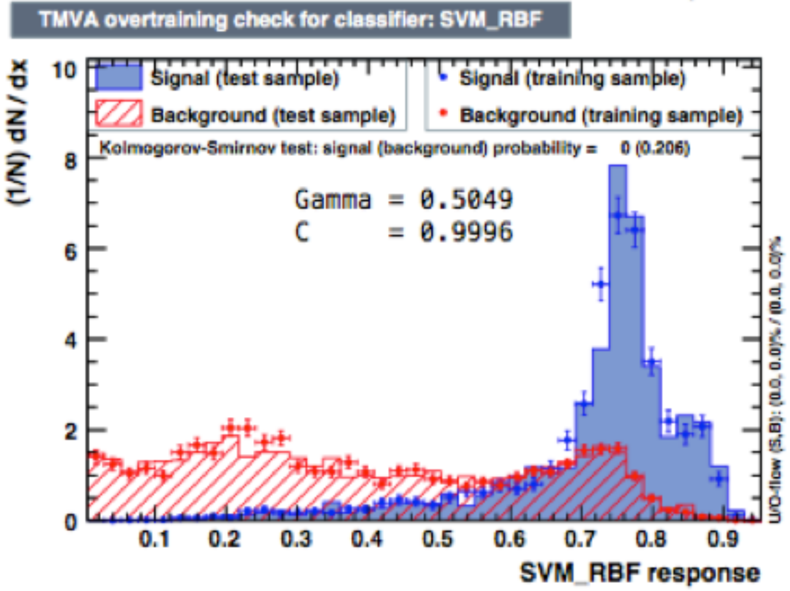
Spiky as optimisation chooses a low number of trees.

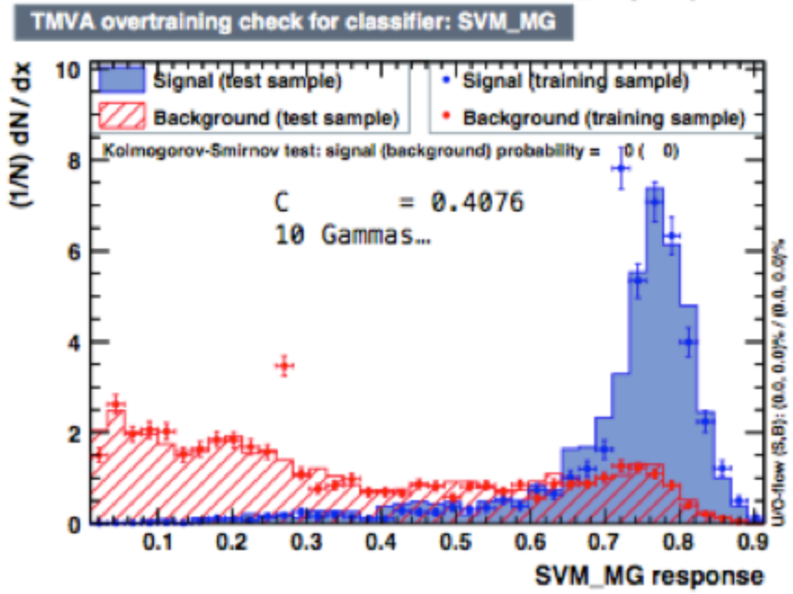




SVM Polynomial (1)

SVM RBF (2)



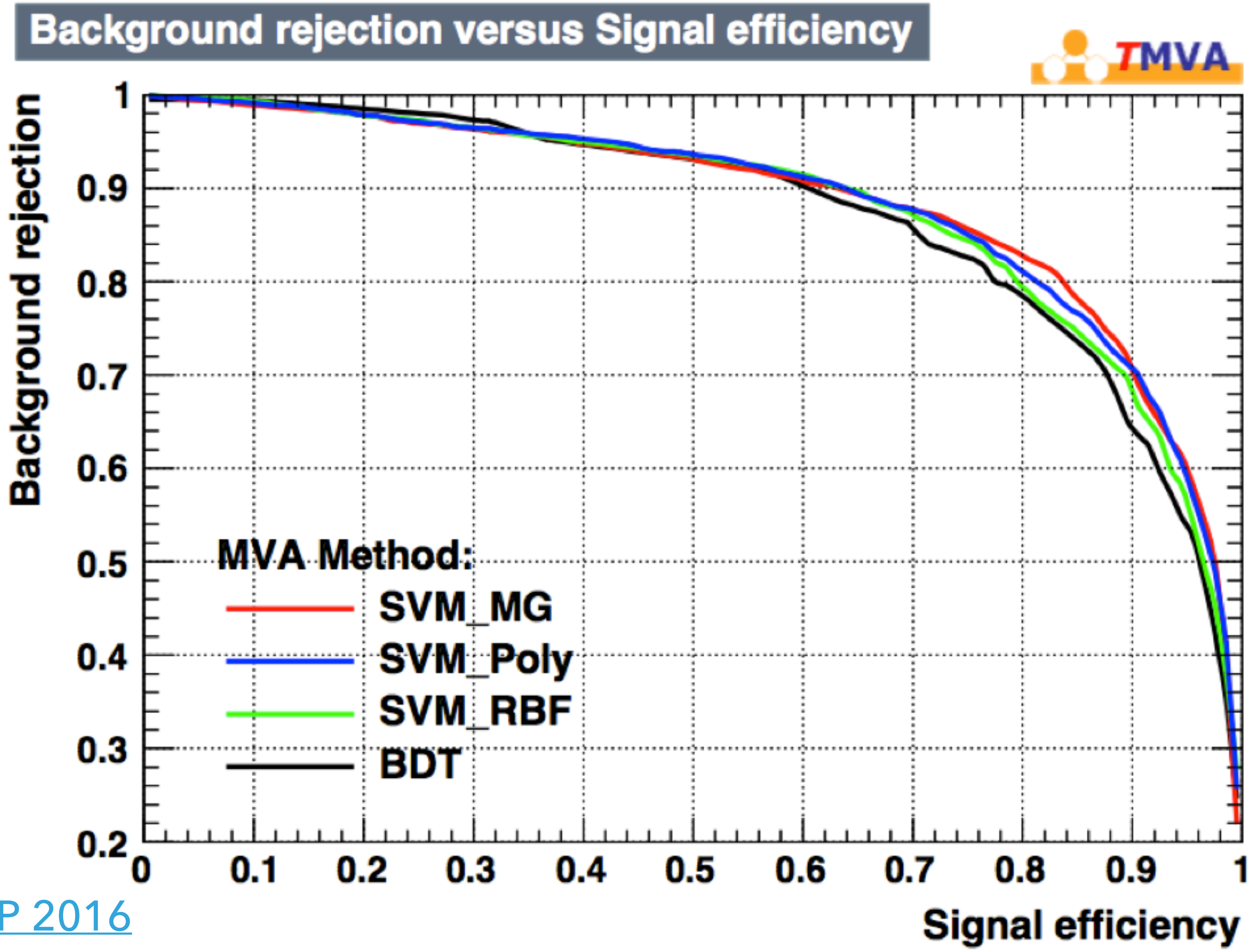


SVM Multi-Gaussian (3)

Trained using the hold out method of cross validation (what is normally done in TMVA), with optimised hyper-parameters.

# EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- SVM provides comparable performance to BDT (and neural networks)*.

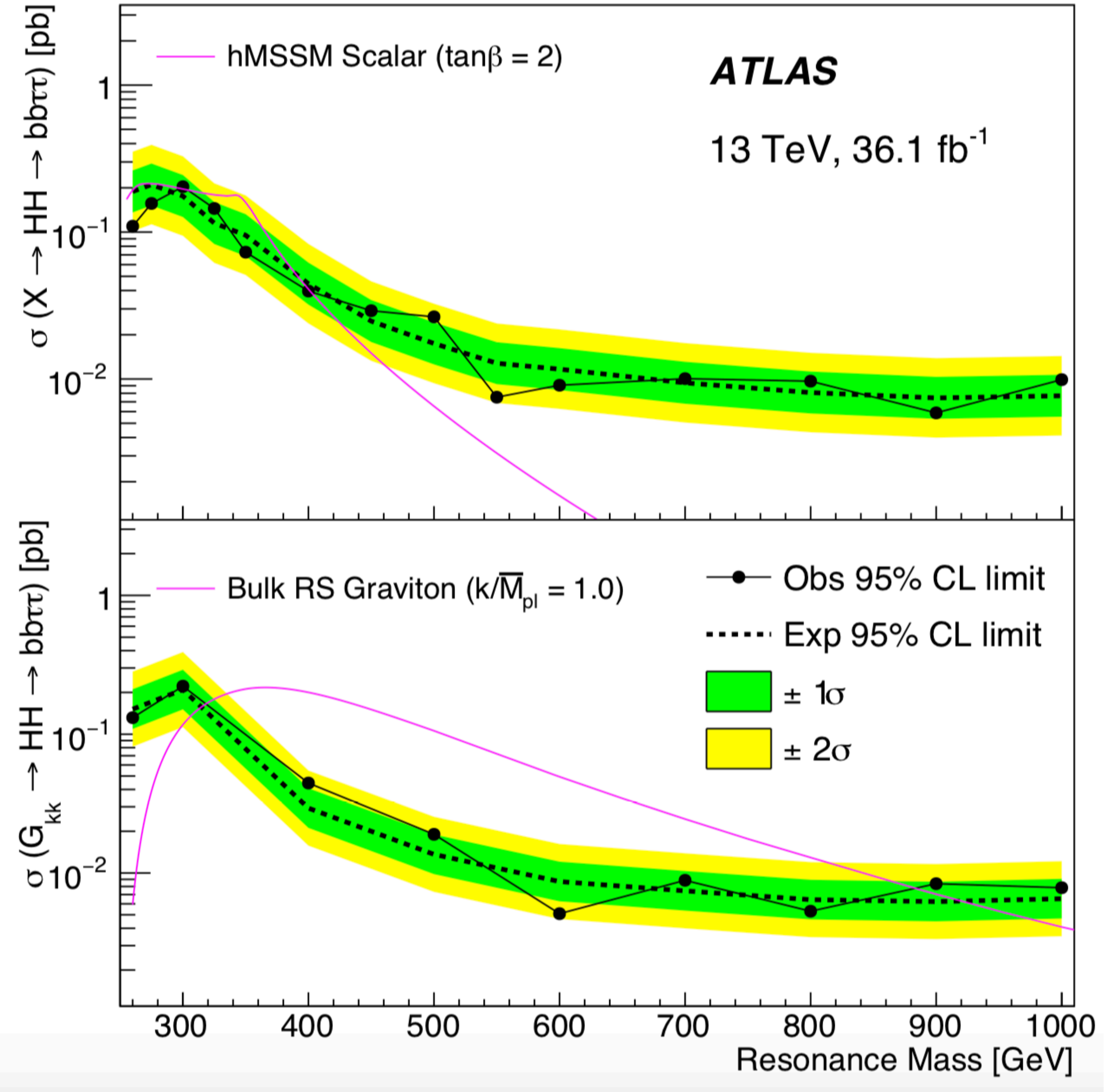


Bevan et al., proc CHEP 2016

*This general conclusion has been reached in one form or another by people studying BDTs vs SVMs and NNs vs SVMs for HEP problems. The take home message is that SVMs require less data to train in order to obtain a generalised result (follows from the fact there are fewer hyper-parameters to determine for SVMs vs other algorithms).

# EXAMPLES: HH→BB$\tau^+\tau^-$ (ATLAS - OFFICIAL RESULT)

- ATLAS recently reported limits on resonant and non-resonant production of HH via bb$\tau^+\tau^-$.



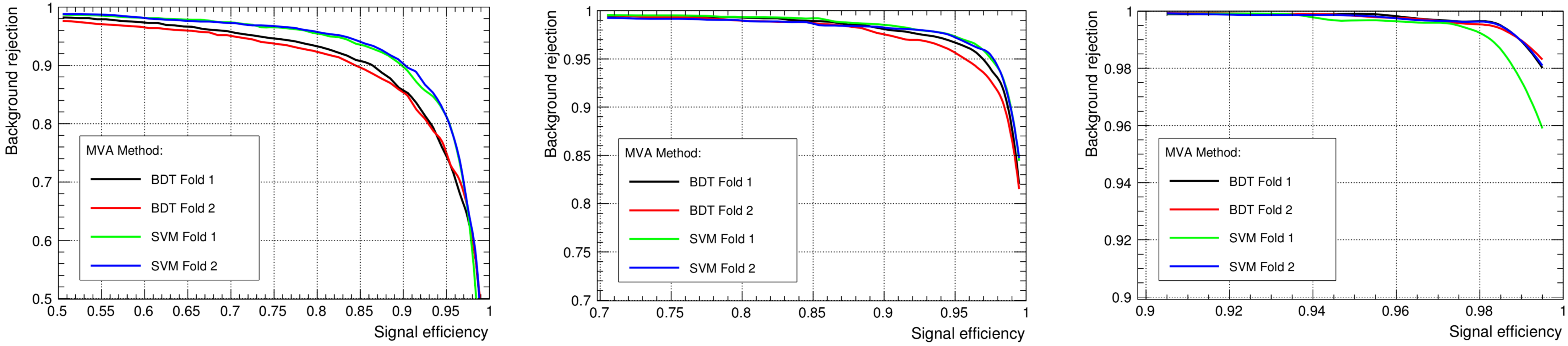
- The standard analysis shown here uses a BDT for both channels that contribute to the final state:
  - Two hadronically decaying $\tau$ leptons.
  - One hadronically and one leptonically decaying $\tau$.
- Results for the SM search are 12.7 times the Standard Model expected sensitivity.

https://arxiv.org/abs/1808.00336

# EXAMPLES: HH→BB$\tau^+\tau^-$ (ATLAS THESIS)

- A student working on this mode also looked at using SVMs (instead of BDTs) for the analysis.
- Similar performance obtained to the official result when using an SVM for both ROC curves and limit plots.

ROC curves for different mass points in the 2HDM search, using one of the trigger lines for the bb$\tau^+\tau^-$ channel.



(a) 2HDM ($m_H = 300\,\mathrm{GeV}$)

(b) 2HDM ($m_H = 500\,\mathrm{GeV}$)

(c) 2HDM ($m_H = 800\,\mathrm{GeV}$)

- SVMs less susceptible (than BDT) to overtraining for small samples.

T. Stevenson, CERN-THESIS-2018-119

# EXAMPLES: HH→BB$\tau^+\tau^-$ (ATLAS THESIS)

- A student working on this mode also looked at using SVMs (instead of BDTs) for the analysis.
- Similar performance obtained to the official result when using an SVM for both ROC curves and limit plots.

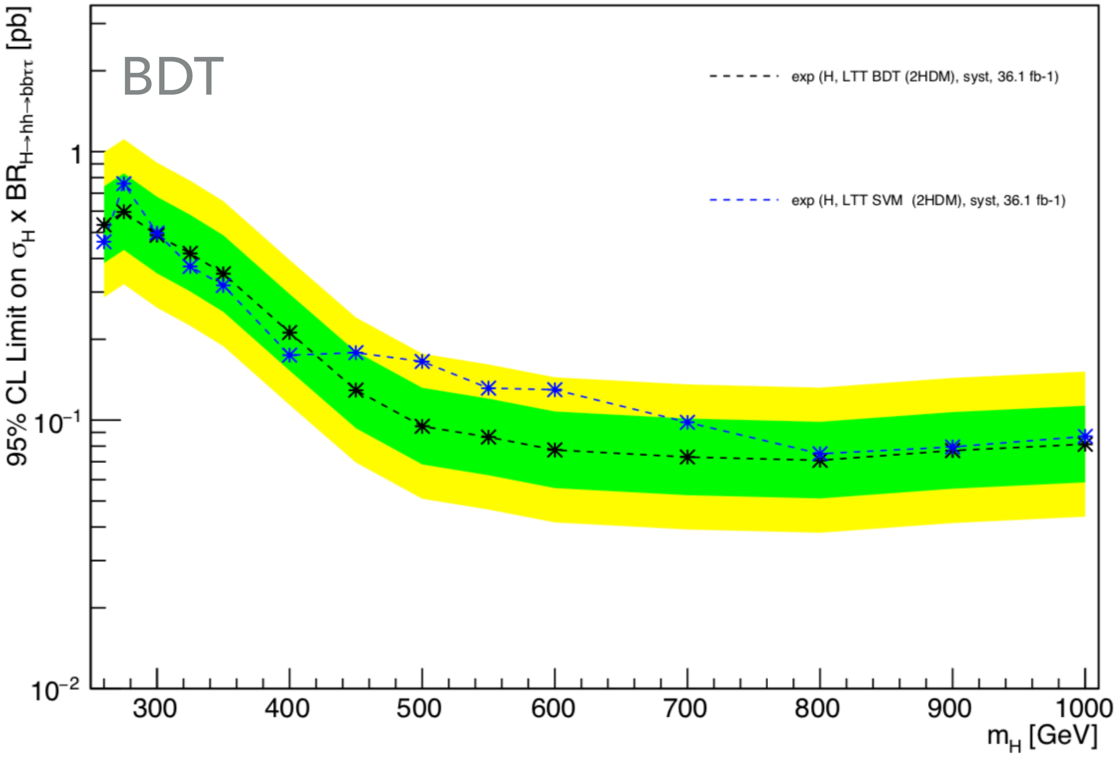


Figure 11.5: Expected limits for the BDT (black) and SVM (blue) at 95% C.L. on the cross-section times branching ratio of the 2HDM heavy scalar Higgs, $H \to hh \to bb\tau\tau$, process in the LTT channel.

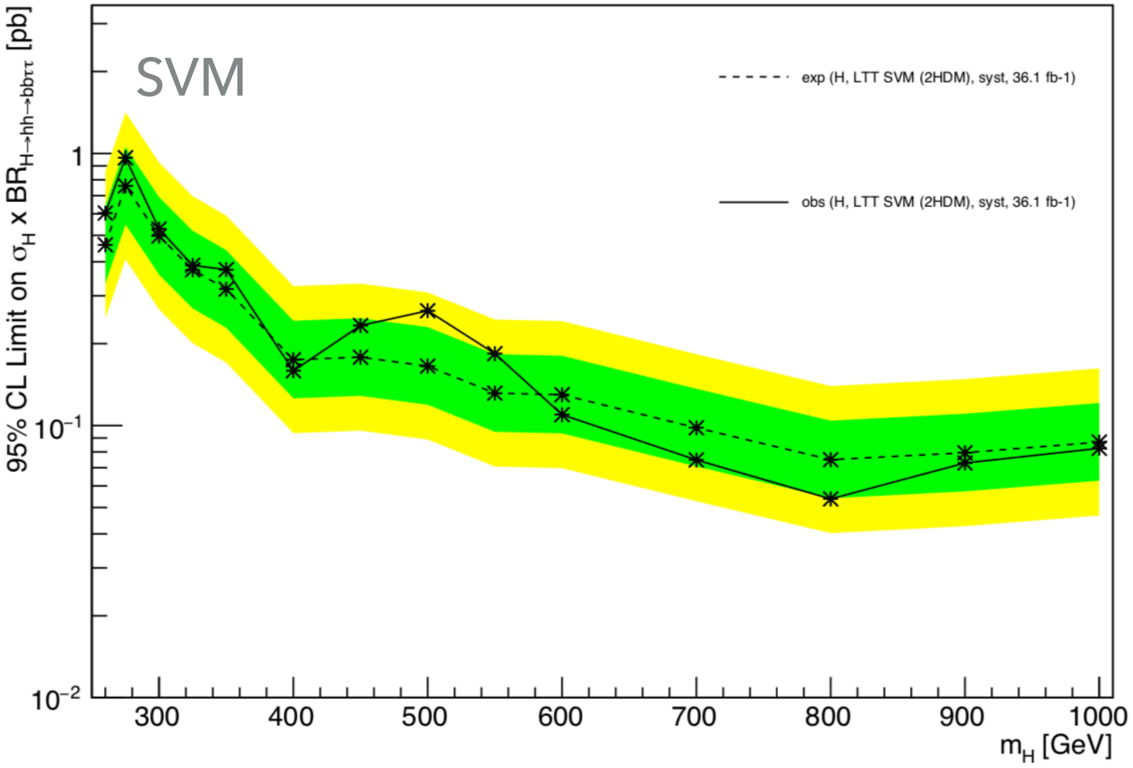


Figure 11.6: Expected (dashed black) and observed (solid black) limits using SVMs at 95% C.L. on the cross-section times branching ratio of the 2HDM heavy scalar Higgs, $H \to hh \to bb\tau\tau$, process in the LTT channel.

T. Stevenson, CERN-THESIS-2018-119

# EXAMPLES: SVM HINT APPLIED TO CMS DATA

- Uses libsvm with an RBF kernel function to optimise two parameters: C and Γ.
- Benchmark example of searching for top squark pair production with stops decaying into the lightest supersymmetric particle (LSP) and a top quark.
    - Could use the ROC area under the curve (AOC) to optimise on, but this is not directly related to the result being produced.
    - Instead use the Azimov estimate of the significance of the result as the figure of merit to compare and optimise performance on:

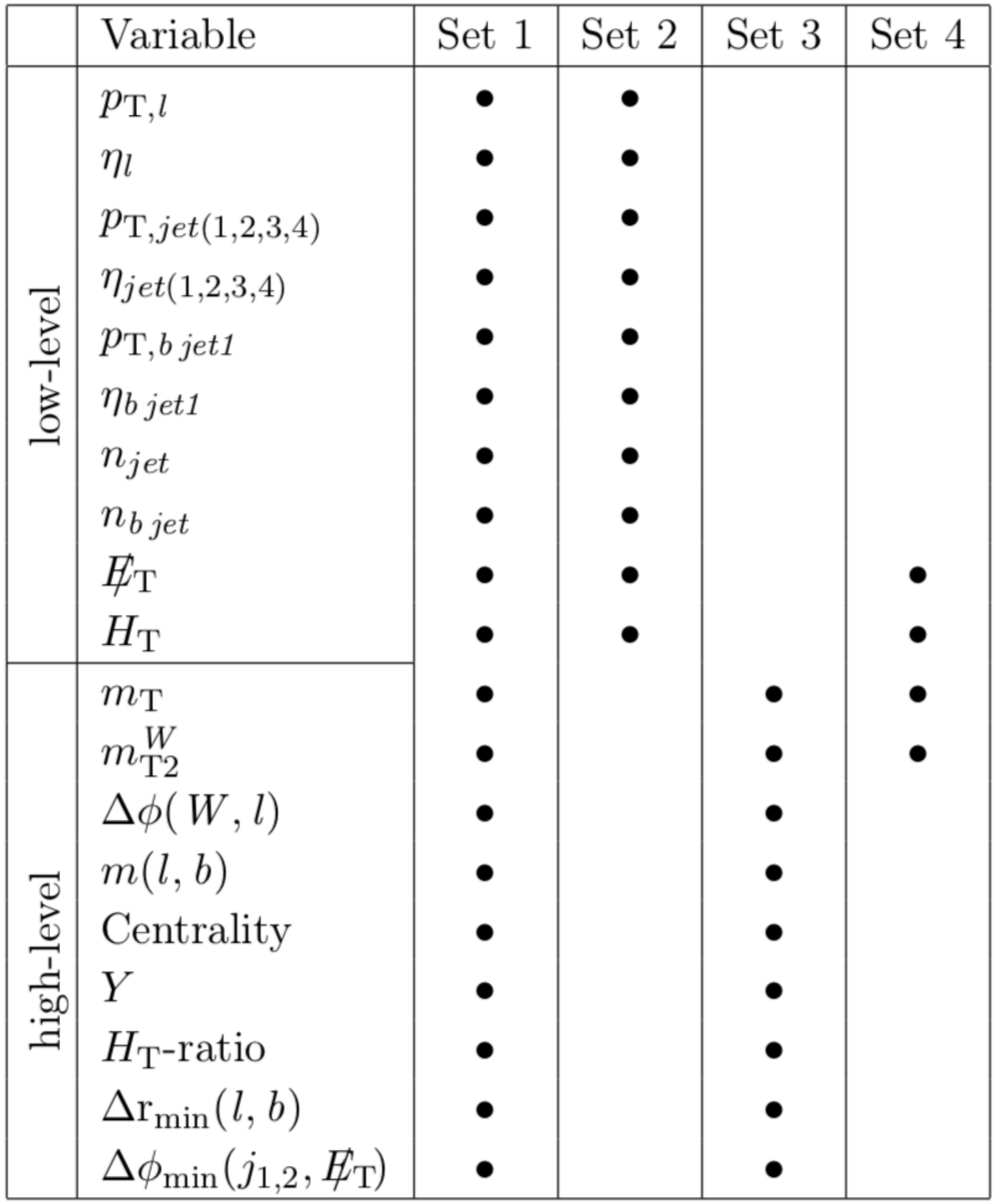
$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}\right]\right)\right]^{1/2}$$

This is the median discovery significance from the Poisson form of the signal (s) and background (b), with an uncertainty on the background of $\sigma_b$.

M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

# EXAMPLES: SVM HINT APPLIED TO CMS DATA

- The variable sets used for the SVM-HINT paper are



| | Variable | Set 1 | Set 2 | Set 3 | Set 4 |
|---|---|---|---|---|---|
| low-level | $p_{\mathrm{T},l}$ | ● | ● | | |
| | $\eta_l$ | ● | ● | | |
| | $p_{\mathrm{T},jet(1,2,3,4)}$ | ● | ● | | |
| | $\eta_{jet(1,2,3,4)}$ | ● | ● | | |
| | $p_{\mathrm{T},b\ jet1}$ | ● | ● | | |
| | $\eta_{b\ jet1}$ | ● | ● | | |
| | $n_{jet}$ | ● | ● | | |
| | $n_{b\ jet}$ | ● | ● | | |
| | $\not{E}_{\mathrm{T}}$ | ● | ● | | ● |
| | $H_{\mathrm{T}}$ | ● | ● | | ● |
| high-level | $m_{\mathrm{T}}$ | ● | | ● | ● |
| | $m_{\mathrm{T2}}^{W}$ | ● | | ● | ● |
| | $\Delta\phi(W,l)$ | ● | | ● | |
| | $m(l,b)$ | ● | | ● | |
| | Centrality | ● | | ● | |
| | $Y$ | ● | | ● | |
| | $H_{\mathrm{T}}$-ratio | ● | | ● | |
| | $\Delta \mathrm{r}_{\mathrm{min}}(l,b)$ | ● | | ● | |
| | $\Delta\phi_{\mathrm{min}}(j_{1,2},\not{E}_{\mathrm{T}})$ | ● | | ● | |

As with other work on using ML methods the expected result that the combination of high level and low level (derived and primitive) features provides better performance than using just one of those sets.

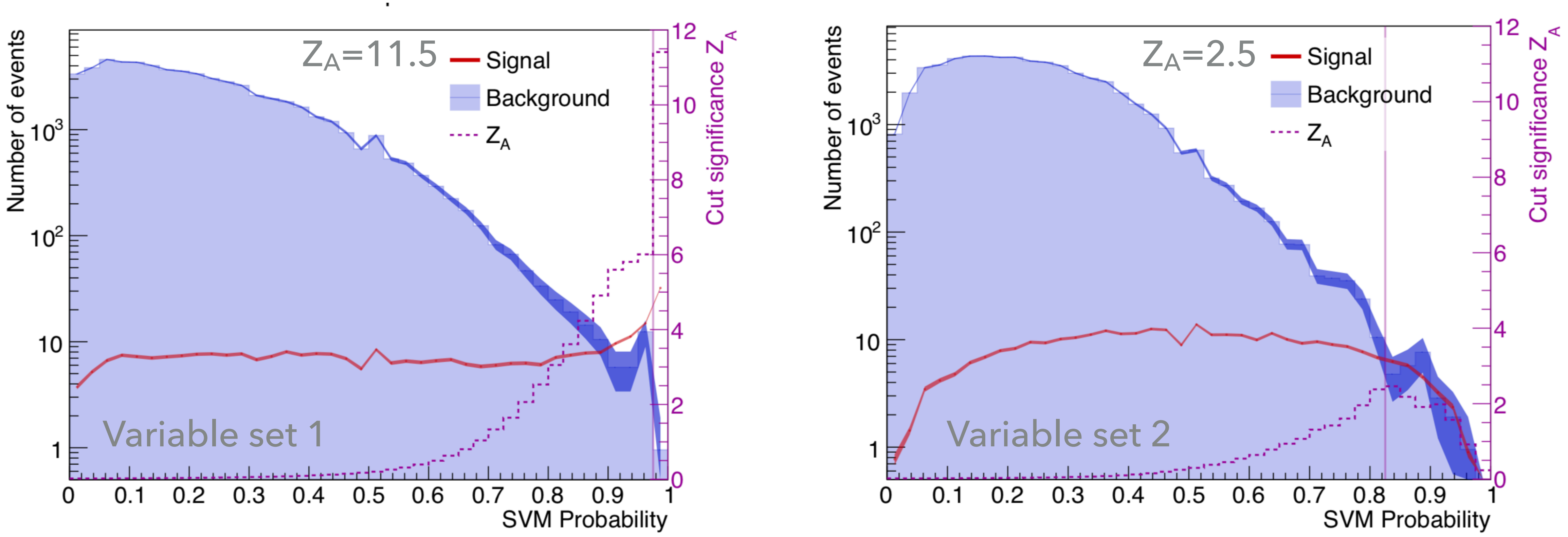
Results on the next two pages illustrate this.

M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

# EXAMPLES: SVM HINT APPLIED TO CMS DATA

- Results are turned into a probabilistic score using a sigmoid function:

$$P(y=1|\hat{f}) = \begin{cases} \frac{\exp(-t)}{1+\exp(-t)} & : t \equiv A + B\hat{f} \geqslant 0 \\ \frac{1}{1+\exp(t)} & : t < 0 \end{cases}$$

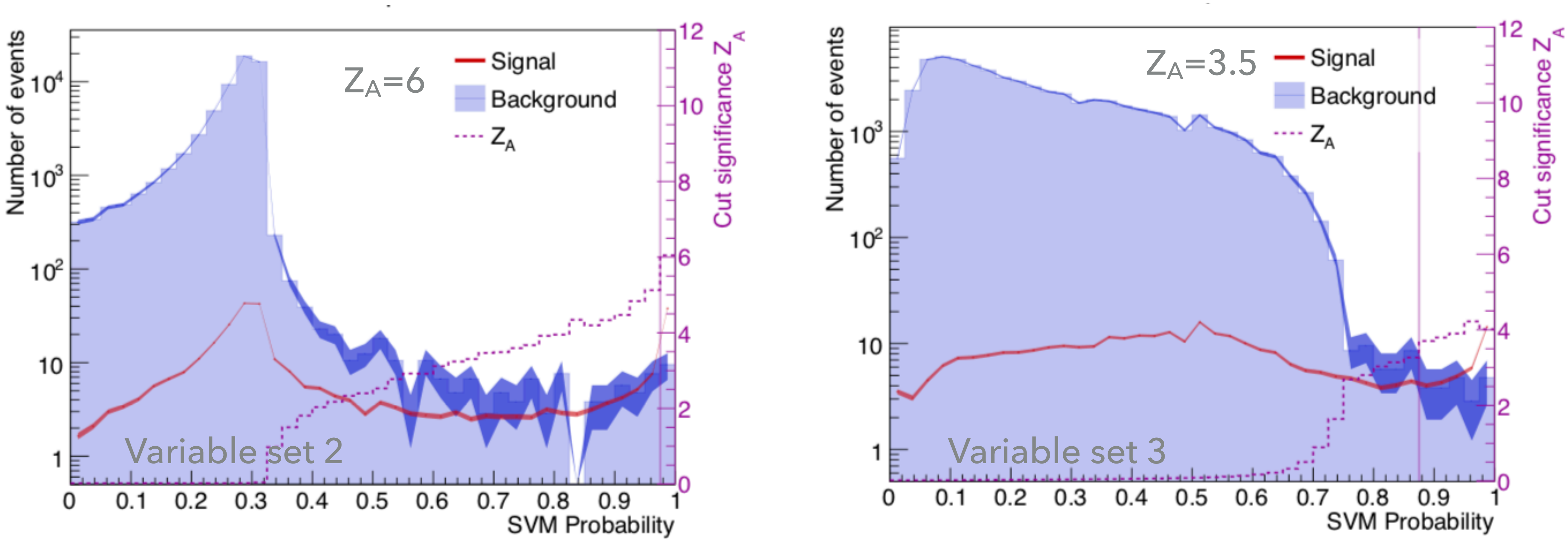


M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

# EXAMPLES: SVM HINT APPLIED TO CMS DATA

- Results are turned into a probabilistic score using a sigmoid function:

$$P(y=1|\hat{f}) = \begin{cases} \frac{\exp(-t)}{1+\exp(-t)} & : t \equiv A + B\hat{f} \geqslant 0 \\ \frac{1}{1+\exp(t)} & : t < 0 \end{cases}$$



M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

# SVMS: SUMMARY AND MISCELLANEOUS NOTES

- Use SVMs when:
  - You have small or very small training examples.
  - and you care about obtaining a generalised result (reproducibility of the output matters even if the data fed to the algorithm changes).
  - Computing time/resource (incl. memory) is not a problem.
- Do not use an SVM when:
  - You have a lot of training examples and/or very little computing resource.

# SVMS: SUMMARY AND MISCELLANEOUS NOTES

- We’ve looked at the hard and soft margin SVMs.
  - The algorithm stems from the same linear separation problem that is addressed by Rosenblatt’s perceptron paper.
  - However this focusses on how far an example is from the margin defining the separating hyperplane.
  - Can’t understand the mapping from the input feature space to the dual space (but we don’t have to).
- SVMs are widely used outside of HEP.
- They have been used for a broad range of physics studies in HEP, but the algorithm has not been widely adopted.
- There are specific reasons why you would or would not want to use the algorithm.
- Searches where you have limited training examples available (e.g. SUSY or Higgs BSM) are cases where you might want to look at the algorithm.

# KNN: K-NEAREST NEIGHBOURS

# K-NEAREST NEIGHBOURS (AKA K-MEANS)

- This is a clustering algorithm, and an example of unsupervised learning.
- Aim: determine the centroid positions C of K clusters in the data containing N examples using a Euclidean distance from the cluster mean to some data example.
- Optimisation: The variance of the clusters is minimised in order to determine the corresponding means of the cluster.

Following Section 14.3.6 of Hastie, Tibshirani, Friedman, Elements of Statistical Learning

# K-NEAREST NEIGHBOURS (AKA K-MEANS)

- Step 1:
  - Given C compute the total cluster variance and minimise this with respect to the means of the clusters.

$$\min_{c,\{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - m_k||^2$$

$x_i$: ith example
$N_k$: Number of examples in Kth cluster
$m_k$: Centroid of Kth cluster
k: Cluster index

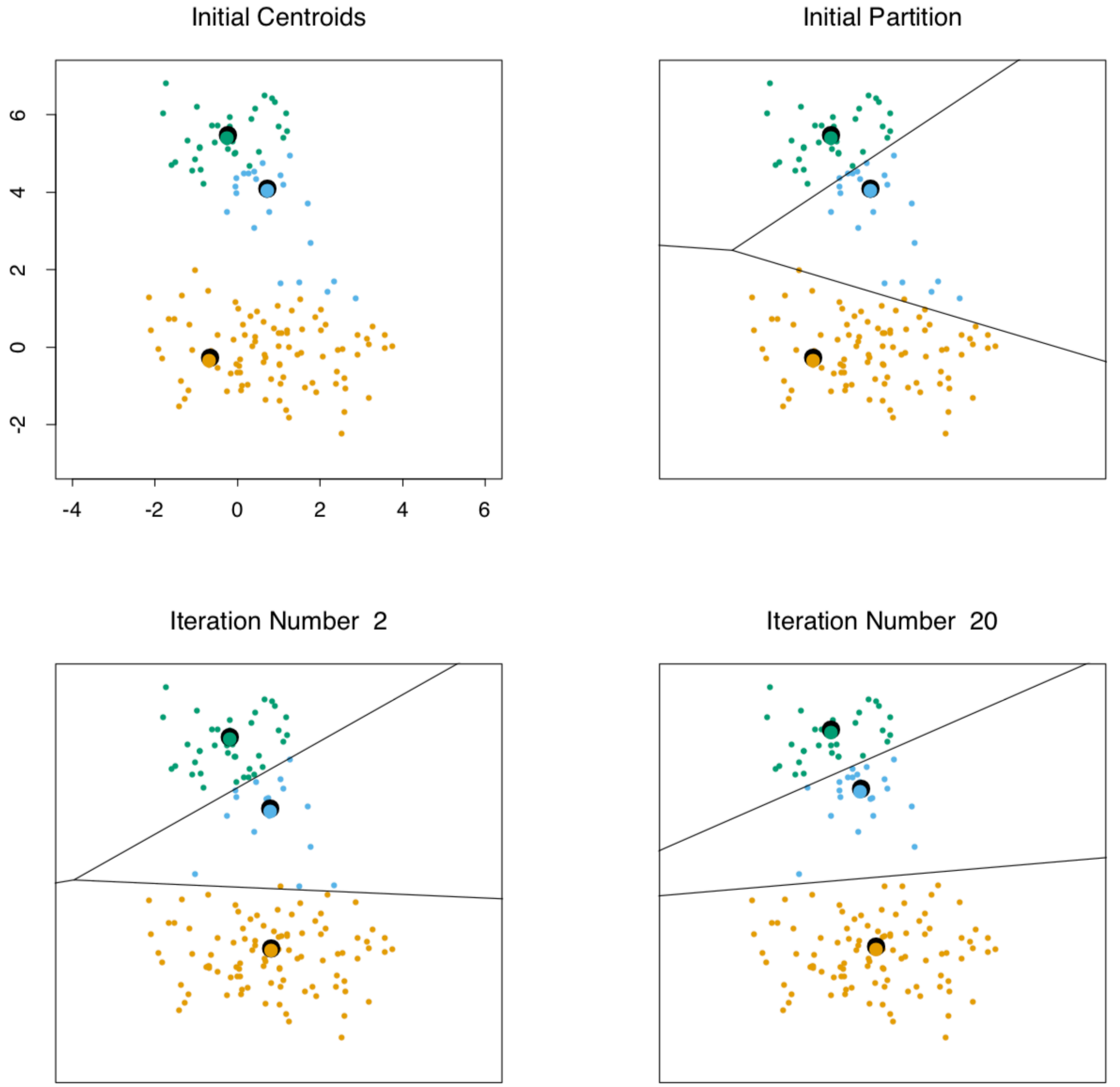
  - This gives the current mean positions of the clusters.
- Step 2:
  - Given a set of means m, minimise these by assigning elements to the closest current cluster mean. i.e.

$$C(i) = argmin_{1 \leq k \leq K} ||x_i - m_k||^2$$

- Step 3:
  - Iterate until the assignments stabilise.

# K-NEAREST NEIGHBOURS (AKA K-MEANS)

- This example shows successive iterations of the K-means algorithm to a set of data with K=3.



This algorithm has the number of clusters, K, as a parameter.

Clustering results will depend on the choice of K.

Colour indicates example assignment to a given cluster.

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 14

# EXPLAINABILITY AND INTERPRETABILITY

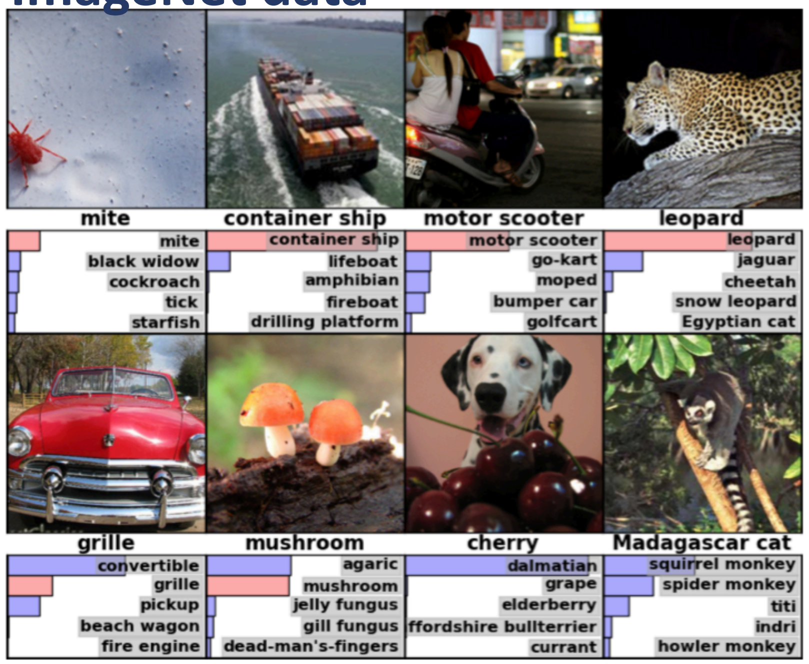
# EXPLAINABILITY AND INTERPRETABILITY

- The issue of how to explain the model, and how to interpret it is challenging.
  - e.g. why was a given prediction made?
    - Event classification / decision making
    - Real value prediction (e.g. signal strength in a score)
- There is no consensus on how to approach this problem; it is an active research area.
- Highlight just a few ways we can help to elucidate our models.

# EXPLAINABILITY AND INTERPRETABILITY

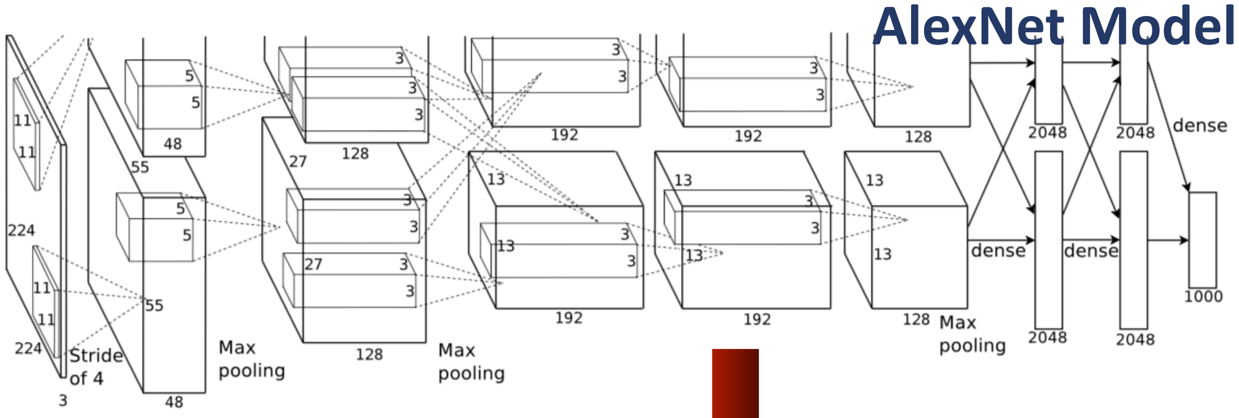
- CNN filter maps provide information about shapes and colour that can be used to interpret how features are identified.

ImageNet data

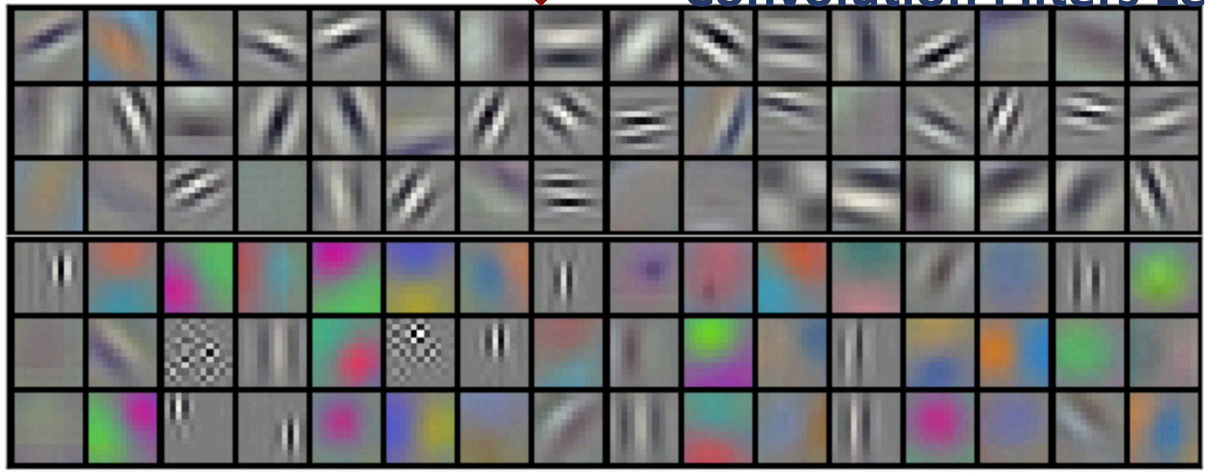


Label assignments

AlexNet Model



Convolution Filters Learned

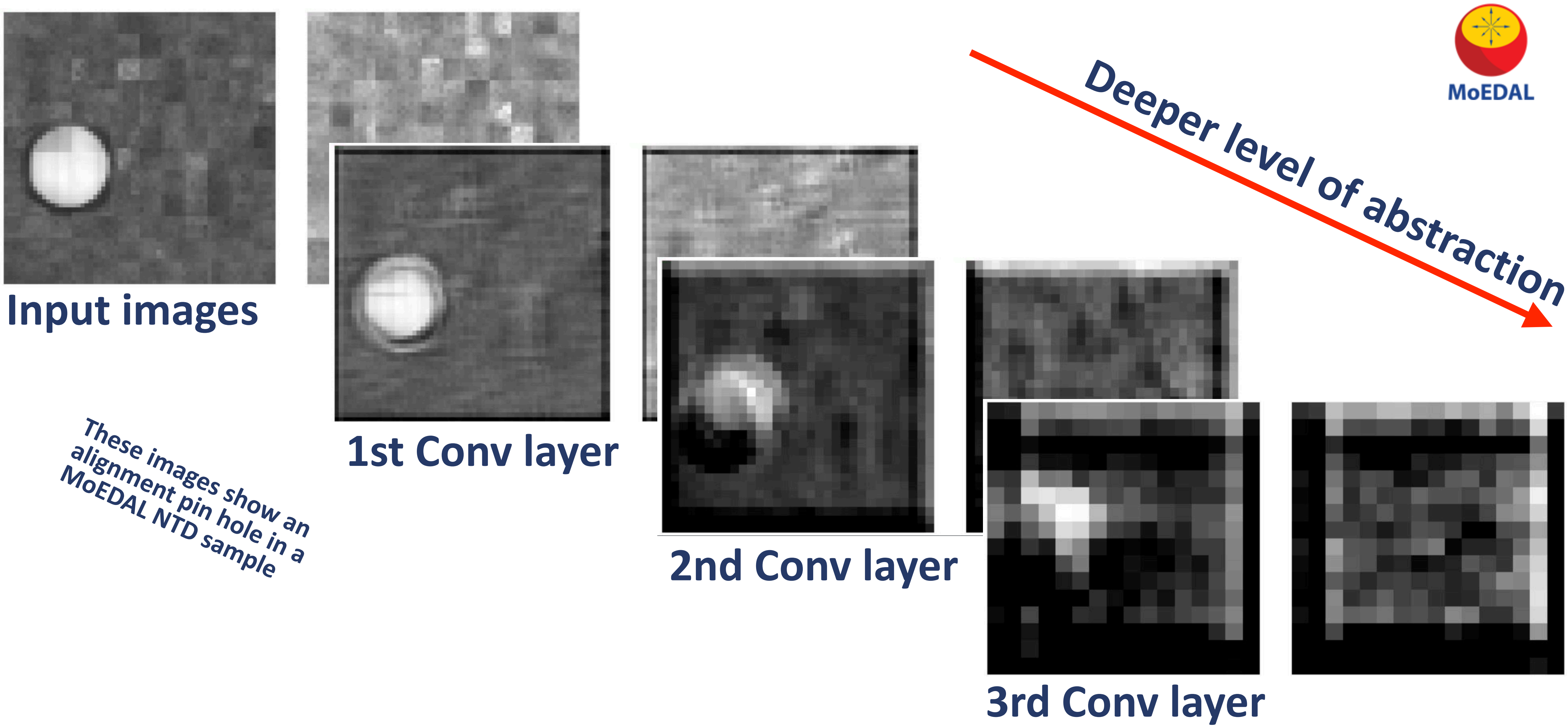


- Requires effort to "see what is happening in may cases"

Krizhevsky et al., Neural Information Processing Systems conference proceedings.

# EXPLAINABILITY AND INTERPRETABILITY

- Some problems have simpler filter interpretations.



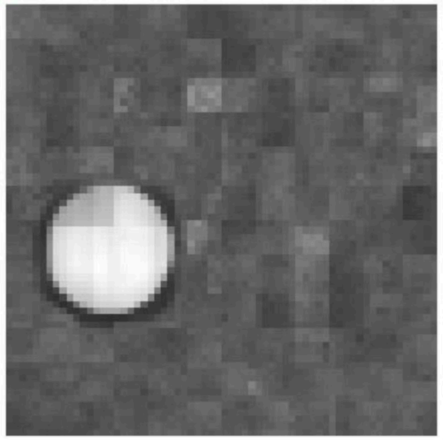
These images show an alignment pin hole in a MoEDAL NTD sample

A. Bevan, https://royalsocietypublishing.org/doi/10.1098/rsta.2019.0392

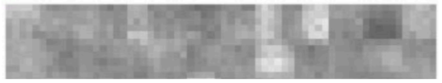
# EXPLAINABILITY AND INTERPRETABILITY

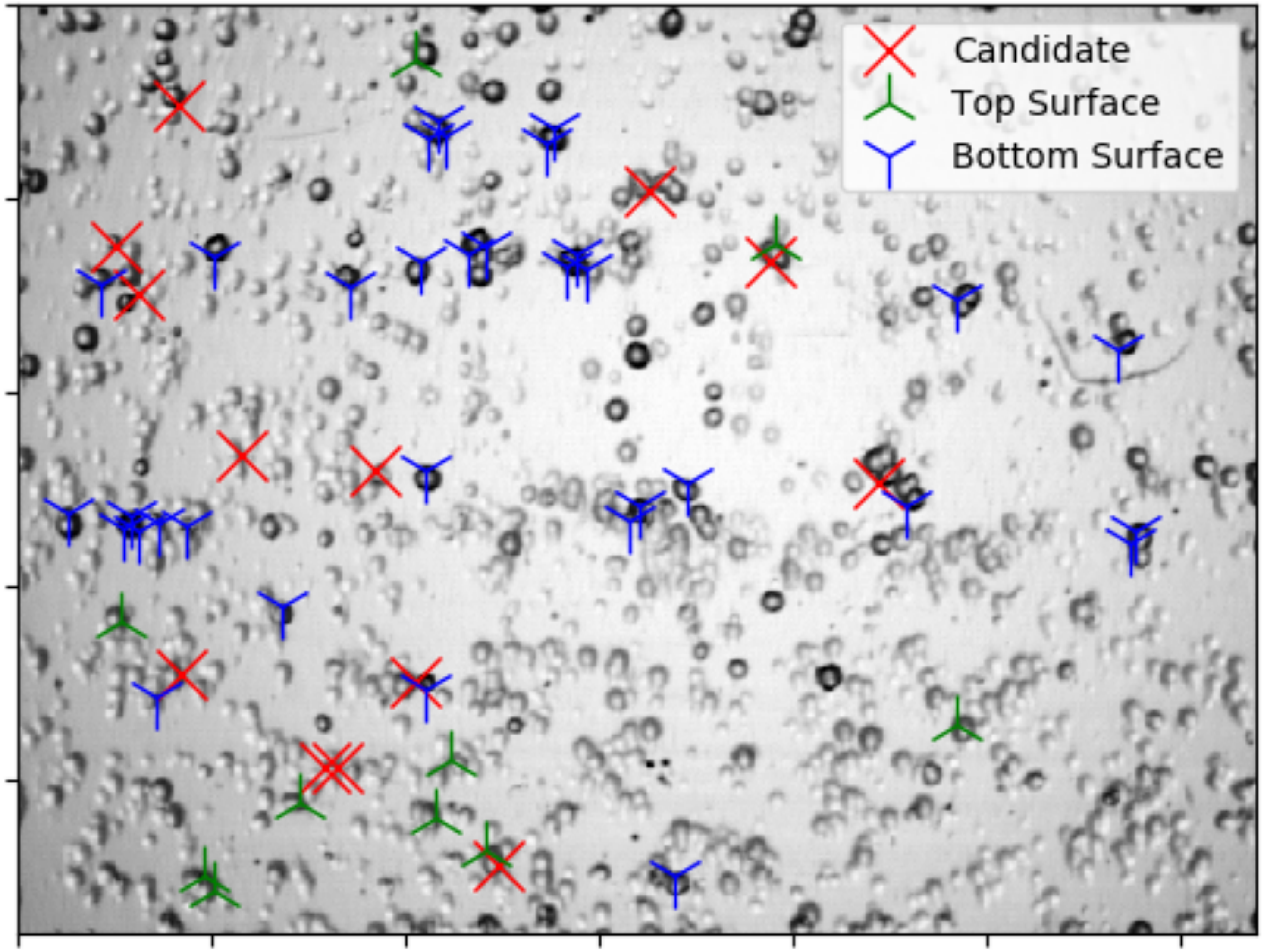
- Some problems have simpler filter interpretations.

MoEDAL



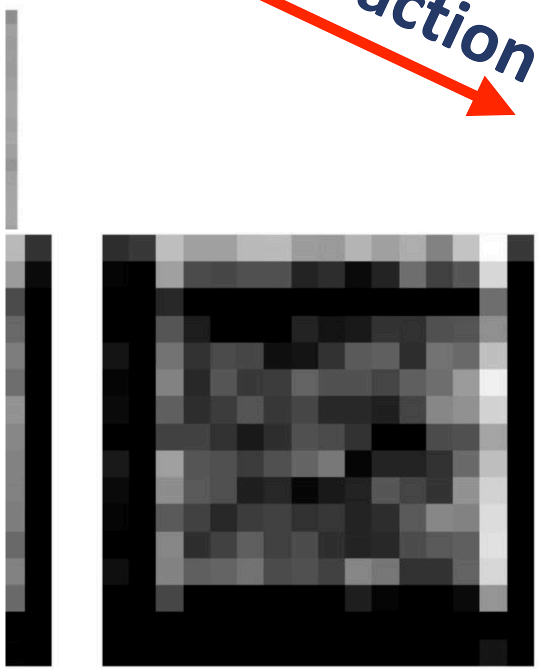
Input images





These images show
alignment pin hole in
MoEDAL NTD sample

level of abstraction

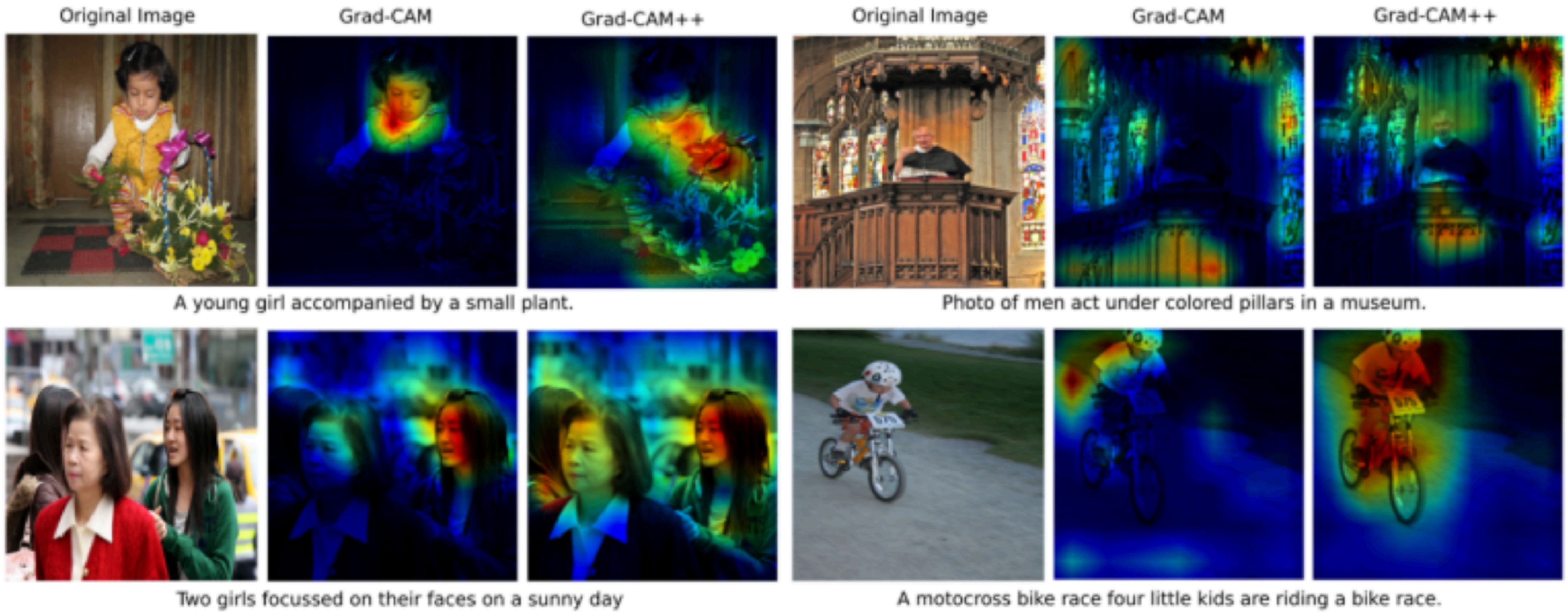


er

A. Bevan, https://royalsocietypublishing.org/doi/10.1098/rsta.2019.0392

# EXPLAINABILITY AND INTERPRETABILITY

- There are methods that use gradients and back-propagation to indicate which local regions of an image lead to a particular decision for CNNs: e.g. GradCam, Guided Back Propagation and variants thereof.



A young girl accompanied by a small plant.

Photo of men act under colored pillars in a museum.

Two girls focussed on their faces on a sunny day

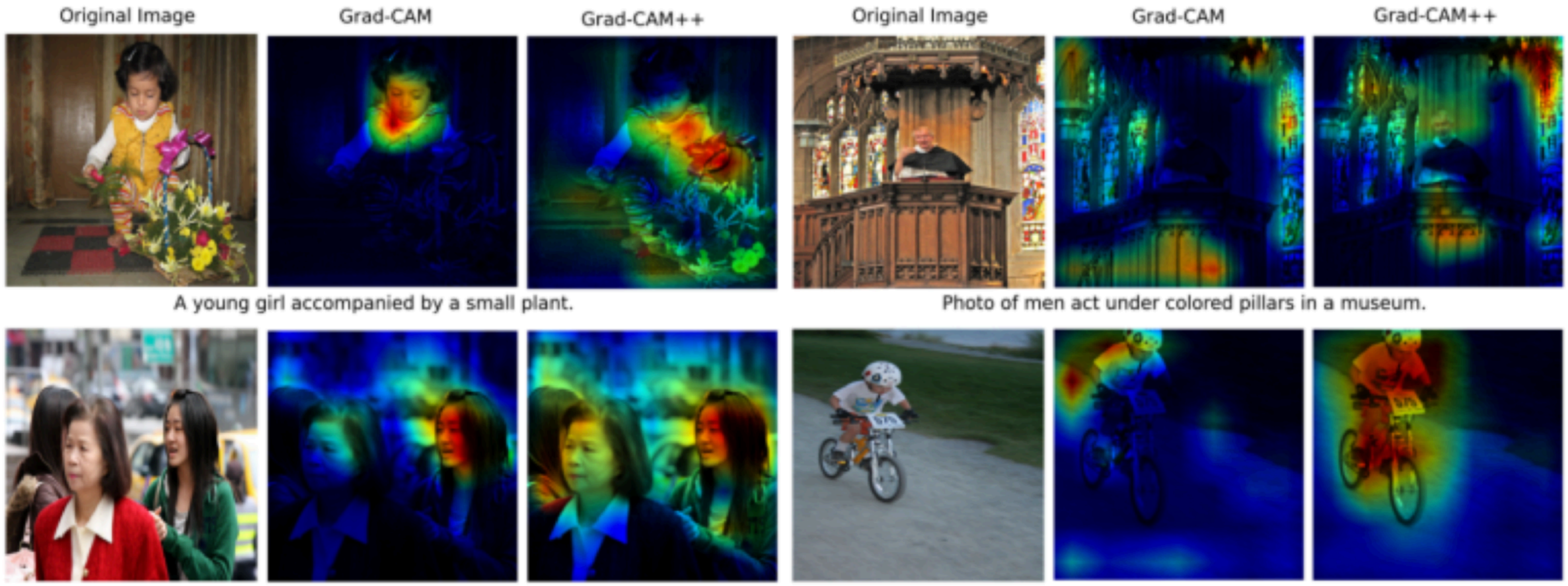
A motocross bike race four little kids are riding a bike race.

- There are also generalisations for DNNs.

A. Chattopadhyay et al., https://arxiv.org/abs/1710.11063

# EXPLAINABILITY AND INTERPRETABILITY

- There are methods that use gradients and back-propagation to indicate which local regions of an image lead to a particular decision for CNNs: e.g. GradCam, Guided Back Propagation and variants thereof.



A young girl accompanied by a small plant.

Photo of men act under colored pillars in a museum.

Two girls focussed on their faces on a sunny day

A motocross bike race four little kids are riding a bike race.

- There are also generalisations for DNNs.

A. Chattopadhyay et al., https://arxiv.org/abs/1710.11063

# EXPLAINABILITY AND INTERPRETABILITY

- Complicated models that rely on function approximation through deep abstractions, or implicit mappings into high dimensional feature spaces can be challenging to understand.
- Interpretation of their results can be straightforward or challenging.
- These however are one class of models; other machine learning algorithms can be more transparent (e.g. Decision Trees).
- Bayesian networks (not discussed here), require causal input in order to construct models, and are by construction easier to interpret than the methods discussed here.

DATA:

- MNIST
- CFAR-10
- CFAR-100
- KAGGLE
- UCI ML DATA REPOSITORY
- TIMIT
- RCV1-V2

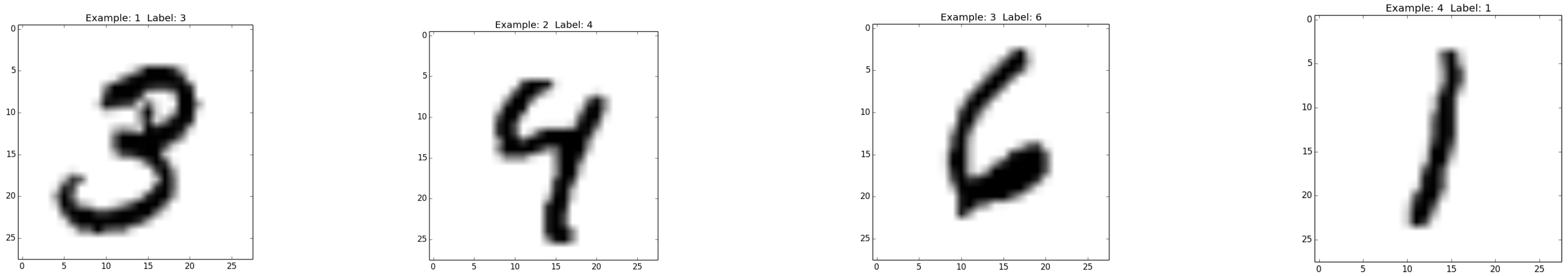
DEEP LEARNING USING LOW LEVEL FEATURES

CROSS VALIDATION

# APPENDIX

# APPENDIX: DATA — MNIST

- MNIST is a standard data set for hand writing pattern recognition. e.g. the numbers 1, 2, 3, … 9, 0

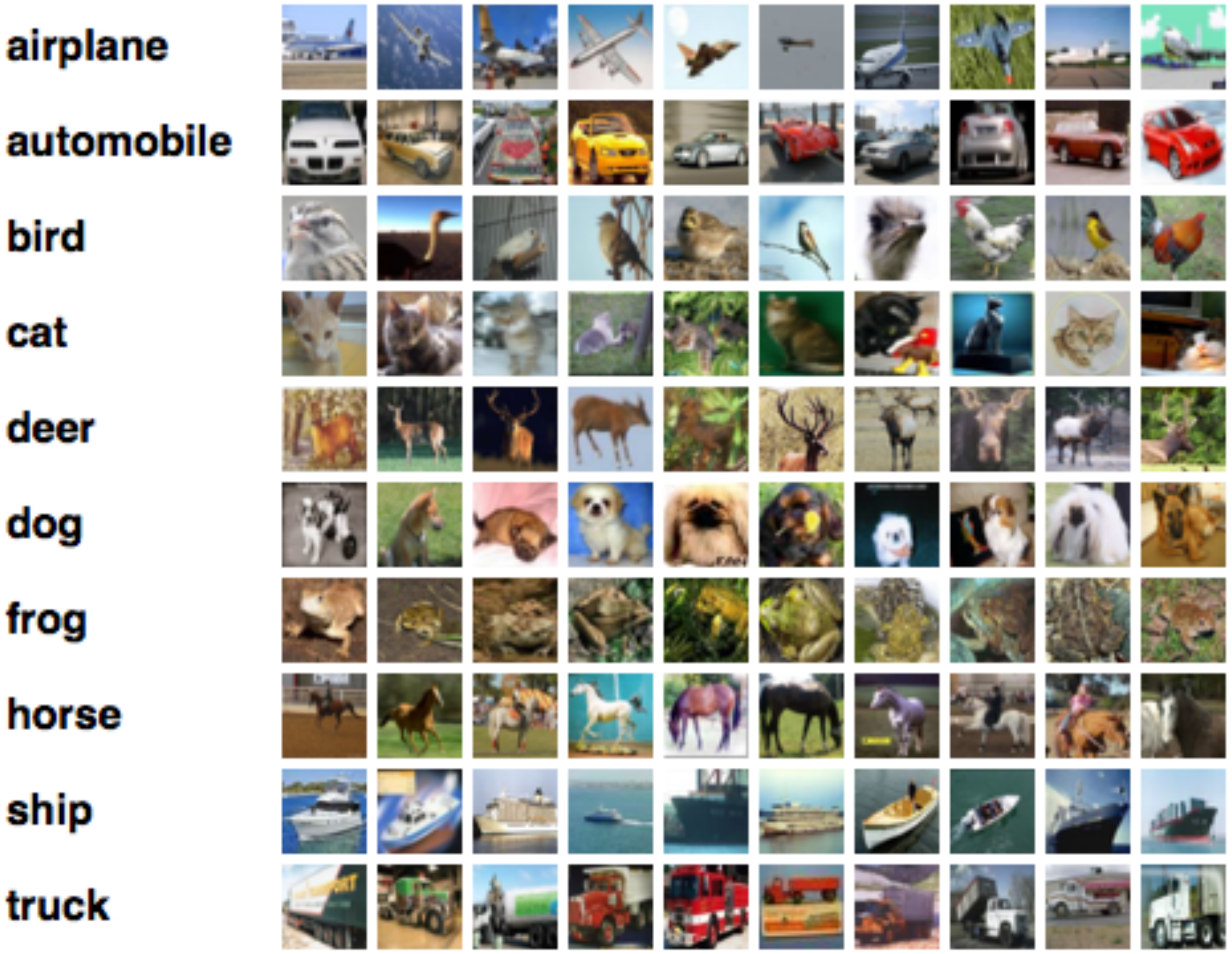


  - 60000 training examples
  - 10000 test examples
  - These are 8 bit greyscale images (one number required to represent each pixel)
  - Renormalise [0, 255] on to [0, 1] for processing.
  - Each image corresponds to a 28x28 pixel array of data.
  - For an MLP this translates to 784 features.

http://yann.lecun.com/exdb/mnist/

# APPENDIX: DATA — CFAR-10

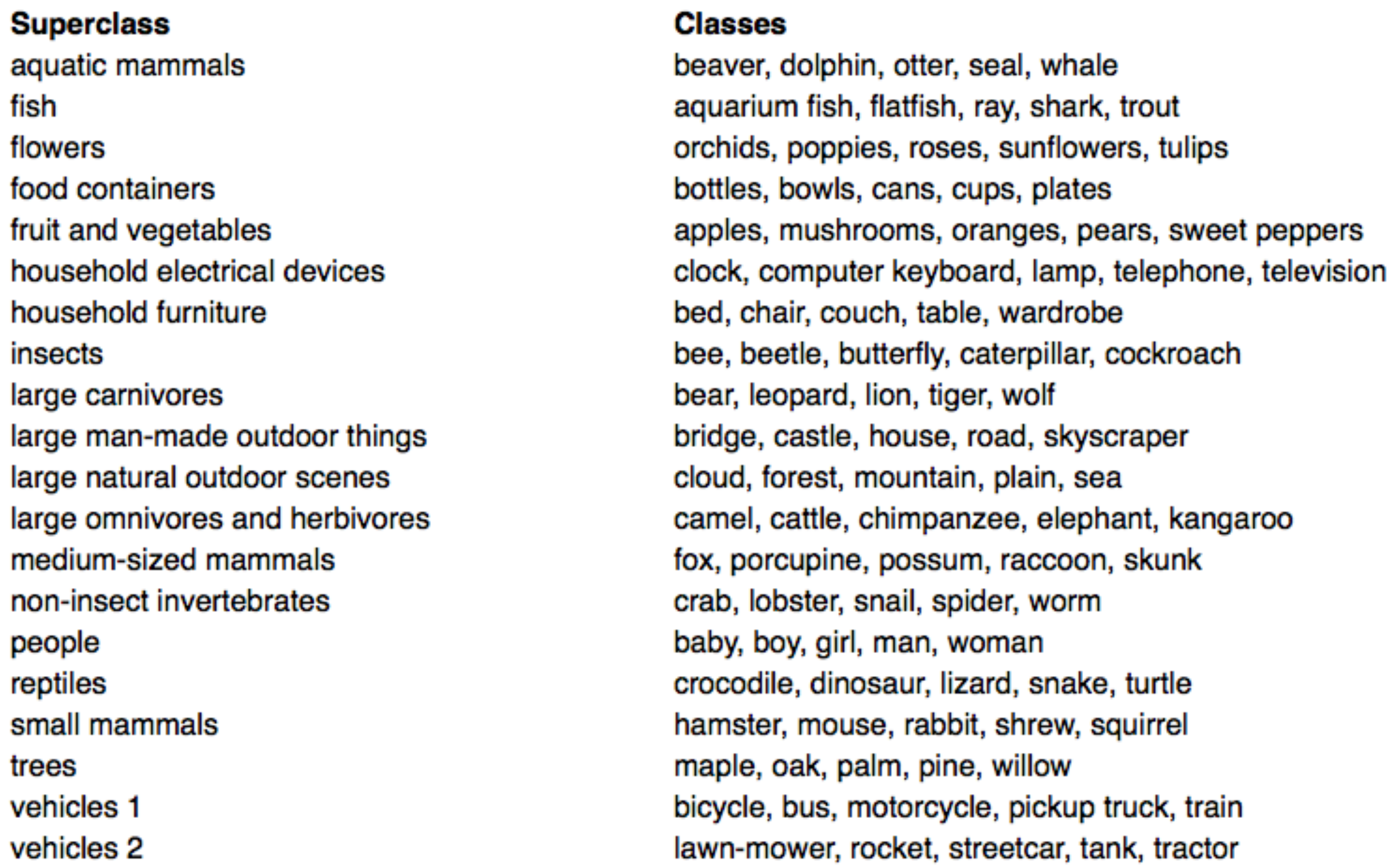
- 60k 32x32 colour images (so each image is a tensor of dimension 32x32x3).
- This is a labelled subset of an 80 million image dataset.
- 10 classes:



https://www.cs.toronto.edu/~kriz/cifar.html

# APPENDIX: DATA — CFAR-100

- 100 class variant on the CFAR10 sample:
- 32x32 colour images (so each image is a tensor of dimension 32x32x3).
- 100 classes:

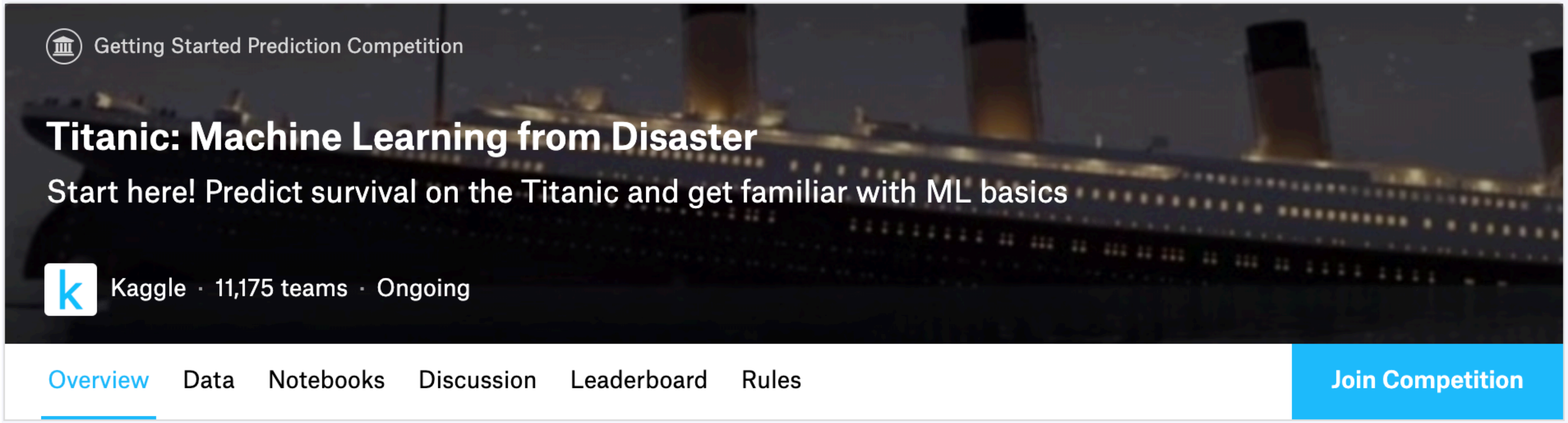


| Superclass | Classes |
|---|---|
| aquatic mammals | beaver, dolphin, otter, seal, whale |
| fish | aquarium fish, flatfish, ray, shark, trout |
| flowers | orchids, poppies, roses, sunflowers, tulips |
| food containers | bottles, bowls, cans, cups, plates |
| fruit and vegetables | apples, mushrooms, oranges, pears, sweet peppers |
| household electrical devices | clock, computer keyboard, lamp, telephone, television |
| household furniture | bed, chair, couch, table, wardrobe |
| insects | bee, beetle, butterfly, caterpillar, cockroach |
| large carnivores | bear, leopard, lion, tiger, wolf |
| large man-made outdoor things | bridge, castle, house, road, skyscraper |
| large natural outdoor scenes | cloud, forest, mountain, plain, sea |
| large omnivores and herbivores | camel, cattle, chimpanzee, elephant, kangaroo |
| medium-sized mammals | fox, porcupine, possum, raccoon, skunk |
| non-insect invertebrates | crab, lobster, snail, spider, worm |
| people | baby, boy, girl, man, woman |
| reptiles | crocodile, dinosaur, lizard, snake, turtle |
| small mammals | hamster, mouse, rabbit, shrew, squirrel |
| trees | maple, oak, palm, pine, willow |
| vehicles 1 | bicycle, bus, motorcycle, pickup truck, train |
| vehicles 2 | lawn-mower, rocket, streetcar, tank, tractor |

https://www.cs.toronto.edu/~kriz/cifar.html

# APPENDIX: DATA — KAGGLE

- Well known website for machine learning competitions; lots of problems and lots of different types of data.
- Also includes training material at:
  - https://www.kaggle.com/learn/overview
  - e.g. Intro to machine learning includes a data science problem on predicting titanic survivors from a limited feature space.
    - Since the outcome is known, this is a good sample of real world data to try out your data science skills.



Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

Kaggle · 11,175 teams · Ongoing

# APPENDIX: DATA — UCI ML DATA REPOSITORY

- Hundreds of data sets covering life sciences, physical sciences, CS / Engineering, Social Sciences, Business, Game and other categories of data.
- Different types of problem: including Classification, regression and clustering samples.
- Different types of data: e.g. Multivariate, univariate, time-series etc.
  - https://archive.ics.uci.edu/ml/datasets.php

# APPENDIX: DATA — TIMIT

- A corpus of acoustic-phonetic continuous speech data, provided with extensive documentation.
- Includes audio files and transcripts
- 630 speakers, each with 10 sentences, corresponding to a corpus of 25200 files (4 files per speaker).
- Total size is approximately 600Mb.

https://catalog.ldc.upenn.edu/LDC93S1

# APPENDIX: DATA — RCV1-V2

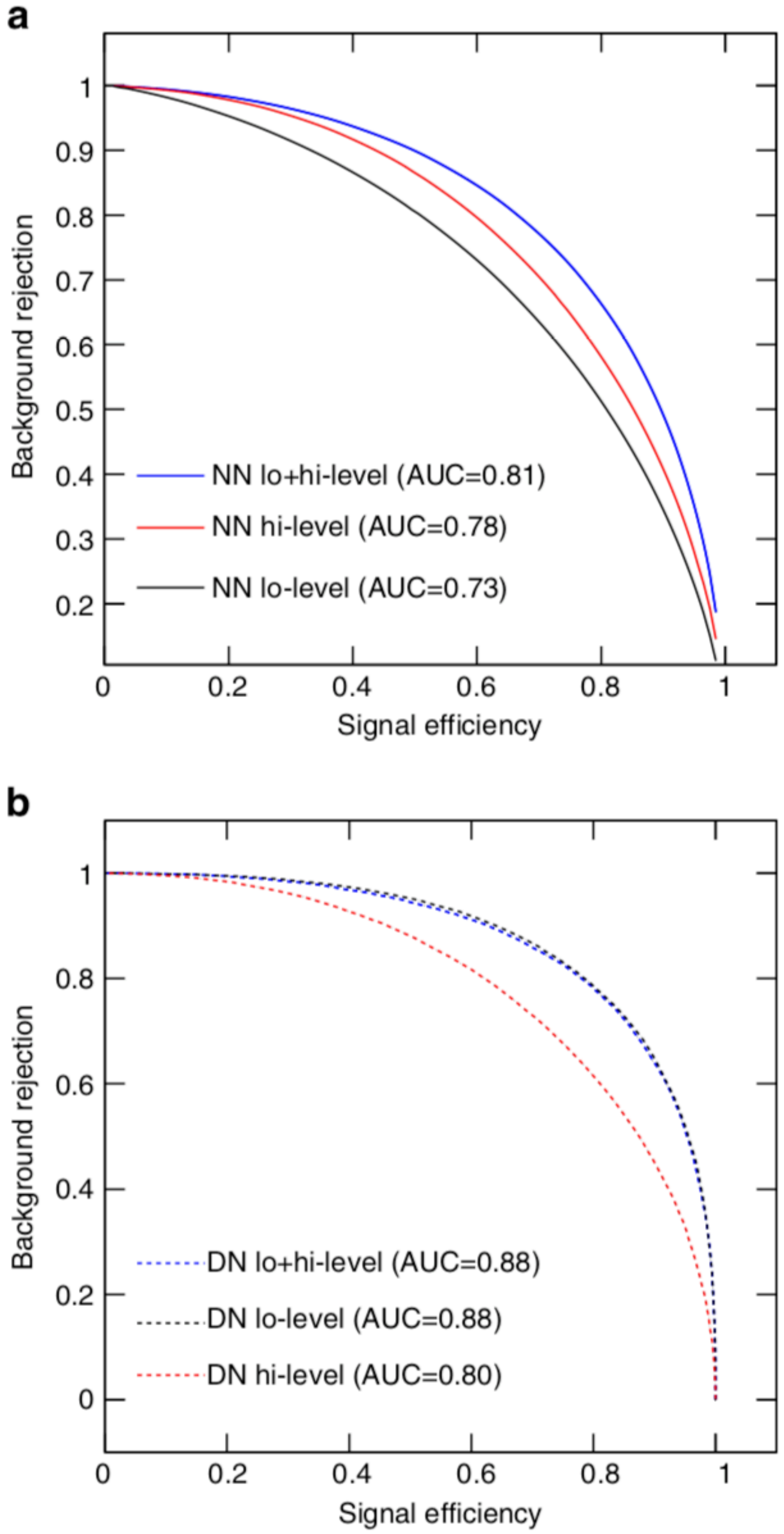
- RCV1: A New Benchmark Collection for Text Categorization Research

- A detailed description of this text categorisation data set can be found in: http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf

http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

# APPENDIX: DEEP LEARNING USING LOW LEVEL FEATURES

- Baldi et al. have reported the ability for a deep network to learn additional information from low level features over and above the high level features; doing function approximation from energy and momenta.
    - 2.6 million (100k) training (validation) examples.
    - 5 layer network with 300 hidden units in each layer.
    - learning rate 0.05 and weight decay coef. of $10^{-5}$.
    - Improves discovery significance over and above a NN.
- Good illustration, is not a realistic scenario as:
    - No systematics included.
    - Relies on very large training samples (unrealistic for many LHC scenarios).
    - FOM optimised is the AUC - we measure limits, cross sections and parameters relating to decay properties or fundamental quantities of the (SM) model.
    - Anecdotally I've found smart learning (SL) and deep learning (DL) perform equally well in many scenarios with realistic HEP Monte Carlo/data control sample constraints. SL algs. are less resource hungry than DL ones.



Baldi, Sadowski, Whiteson: DOI: 10.1038/ncomms5308.

# APPENDIX: CROSS VALIDATION

- In statistics cross validation is used to understand the mean and variance of estimations of model predictions from data.
  - The bias will be irreducible and mean that the predictions made will have some systematic effect related to the average output value.
  - The variance will depend on the size of the training sample.
  - The central limit theorem tells us that:

If one takes N random samples of a distribution of data that describes some variable x, where each sample is independent and has a mean value $\mu_i$ and variance $\sigma_i^2$, then the sum of the samples will have a mean value M and variance V where:

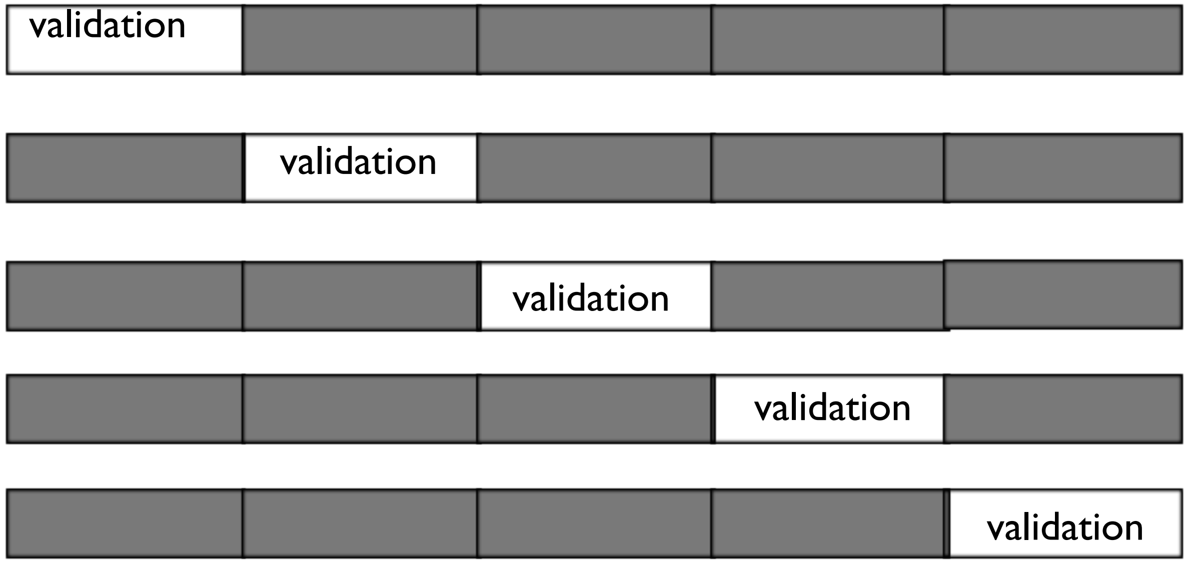
$$M = \sum_{i=1}^{N} \mu_i$$

$$V = \sum_{i=1}^{N} \sigma_i^2$$

Geisser, S. (1975). The predictive sample reuse method with applications. J. Amer. Statist. Assoc., 70:320–328.
For a review of cross validation see: S. Arlot and A. Celisse, Statistics Surveys Vol. 4 4079 (2010).

# APPENDIX: CROSS VALIDATION

- Application of this concept to machine learning can be seen via k-fold cross validation and its variants*
  - Divide the data sample for training and validation into k equal sub-samples.
  - From these one can prepare k sets of validation samples and residual training samples.
  - Each set uses all examples; but the training and validation sub-sets are distinct.
  - One can then train the data on each of the k training sets, validating the performance of the network on the corresponding validation set.

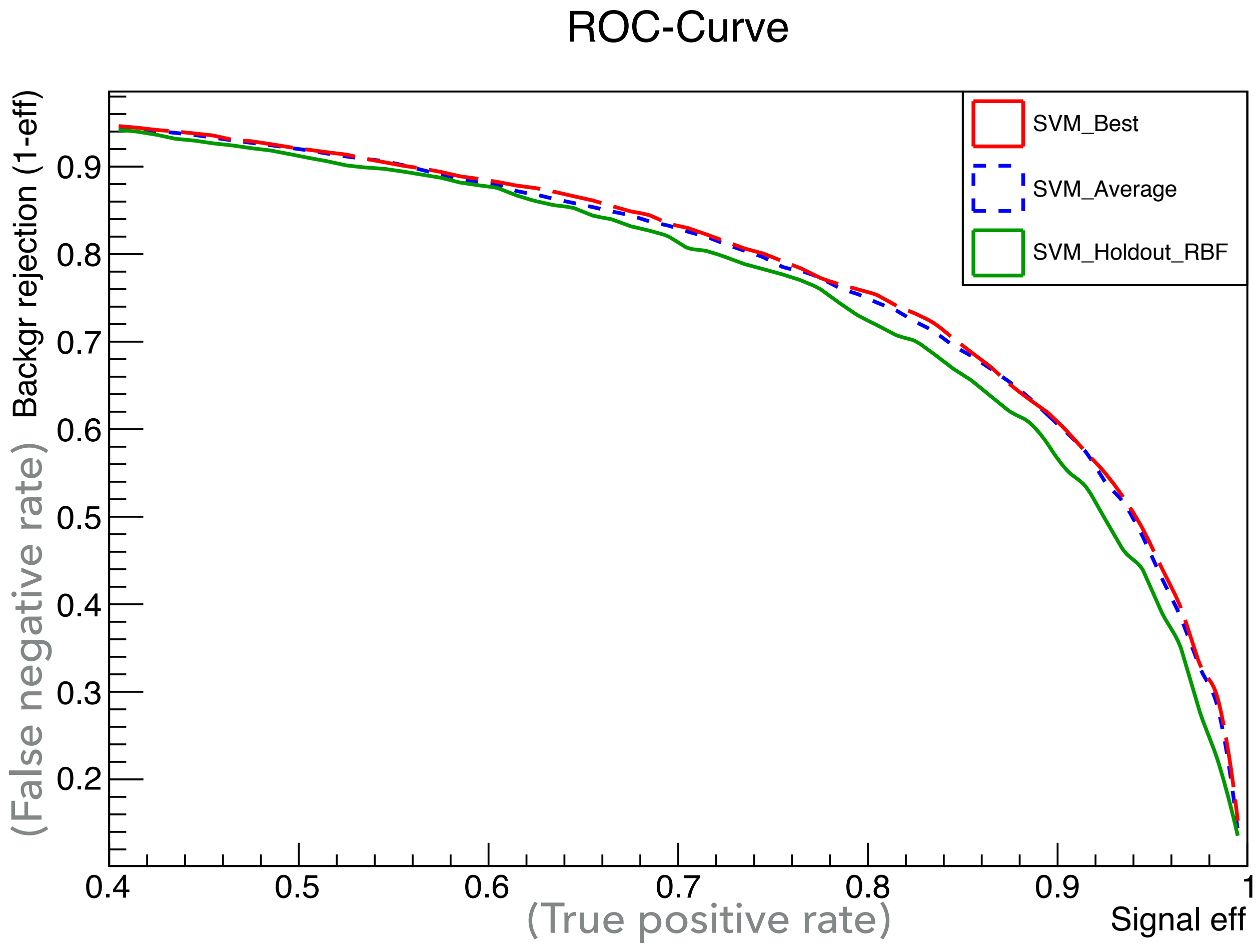


*Variants include the extremes of leave 1 out CV and Hold out CV as well as leave p-out CV. These involve reserving 1 example, 50% of examples and p examples for testing, and the remainder of data for training, respectively.

Geisser, S. (1975). The predictive sample reuse method with applications. J. Amer. Statist. Assoc., 70:320–328.
For a review of cross validation see: S. Arlot and A. Celisse, Statistics Surveys Vol. 4 4079 (2010).

# APPENDIX: CROSS VALIDATION

- Application of this concept to machine learning can be seen via k-fold cross validation and its variants*
  - The ensemble of response function outputs will vary in analogy with the spread of a Gaussian distribution.
  - This results in family of ROC curves; with a representative performance that is neither the best or worst ROC.
  - The example shown is for a Support Vector Machine, but the principle is the same.
  - It is counter-intuitive, but the robust response comes from the average, not the best performance using the ROC FOM.



*Variants include the extremes of leave 1 out CV and Hold out CV as well as leave p-out CV. These involve reserving 1 example, 50% of examples and p examples for testing, and the remainder of data for training, respectively.

Geisser, S. (1975). The predictive sample reuse method with applications. J. Amer. Statist. Assoc., 70:320–328.
For a review of cross validation see: S. Arlot and A. Celisse, Statistics Surveys Vol. 4 4079 (2010).