Neutrinoless double beta decay classification in the LUX-Zeplin TPC

Theory, results, and future work

Andrey Solovov -- 2nd BigDataHEP meeting, Braga -- Feb. 13 2020

LZ and the interest of neutrinoless 2β decay

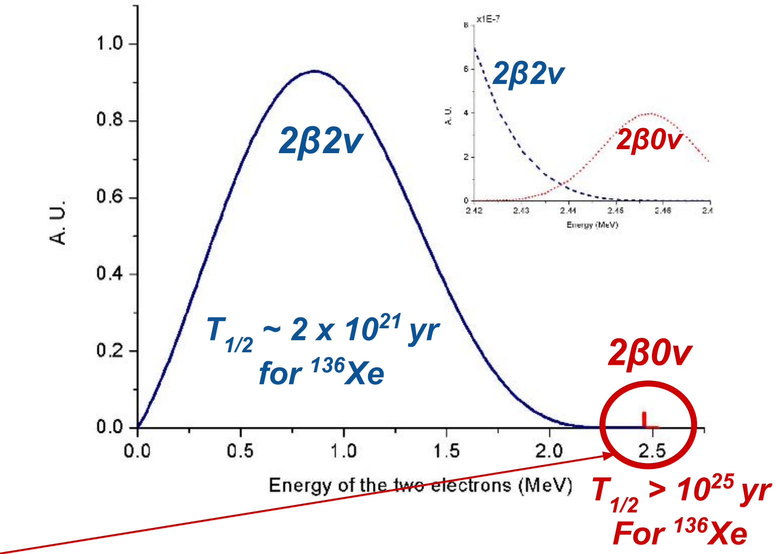
Standard *(2 neutrino)* double beta decay is very rare but allowed in SM, and it looks like this:

$$(A,Z) \rightarrow (A,Z+2) + 2e^- + 2\nu_e$$

Neutrinoless double beta decay *(NDBD)* could happen if the neutrino is its own antiparticle:

$$(A,Z) \rightarrow (A,Z+2) + 2e^-$$

If this is observed, and neutrinos are seen to be light enough, then the matter-antimatter asymmetry in the universe can be explained

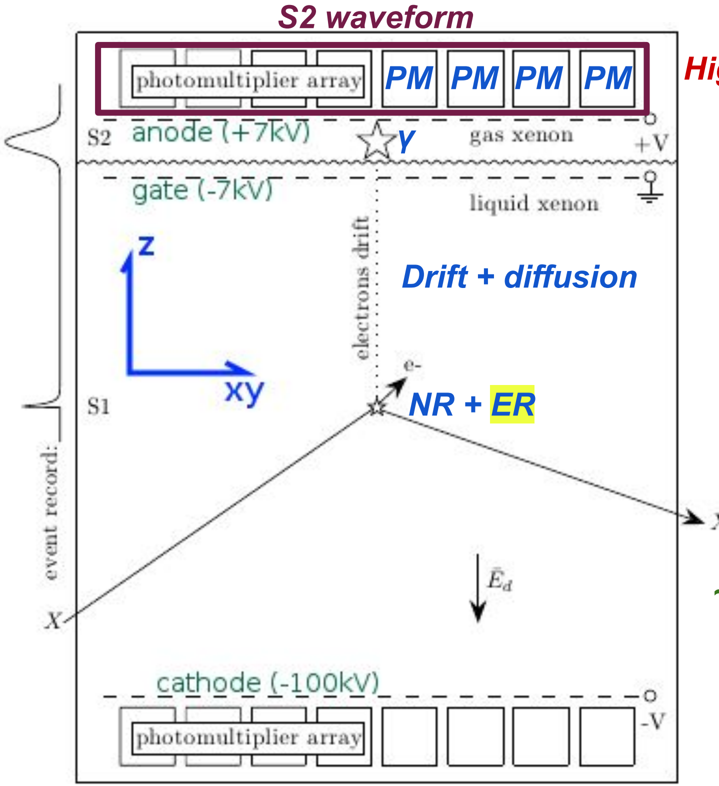


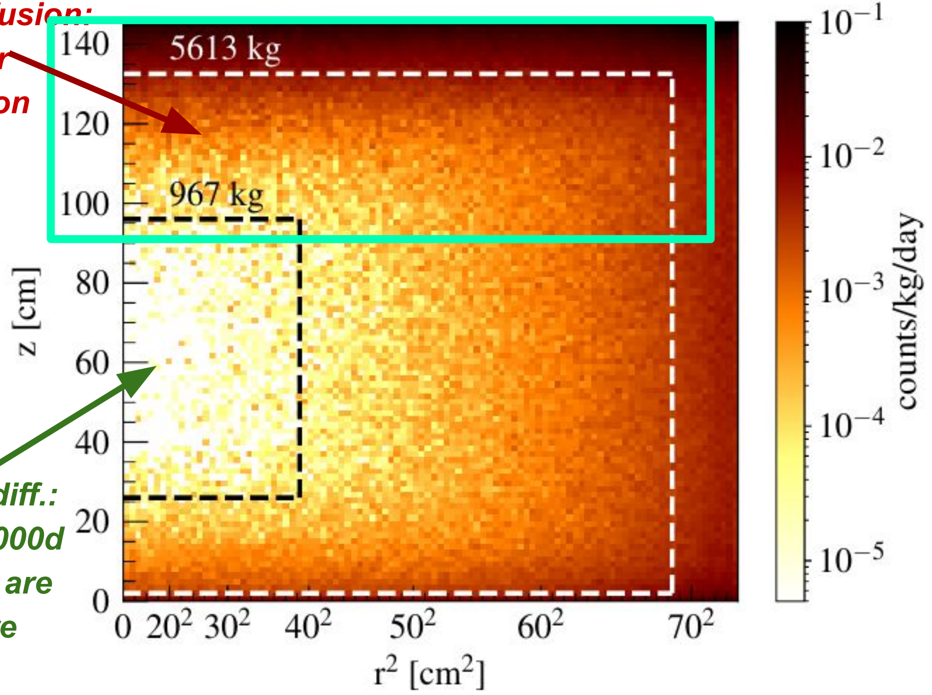
$T_{1/2}$ ~ 2 x 10^{21} yr for ^{136}Xe

$T_{1/2}$ > 10^{25} yr For ^{136}Xe

The LXe in LZ has a ~2.5 MeV ββ decay for ^{136}Xe, and LZ has an exposure of 1360 kg x years, competitive with KamLAND-Zen

LZ - the TPC and its backgrounds





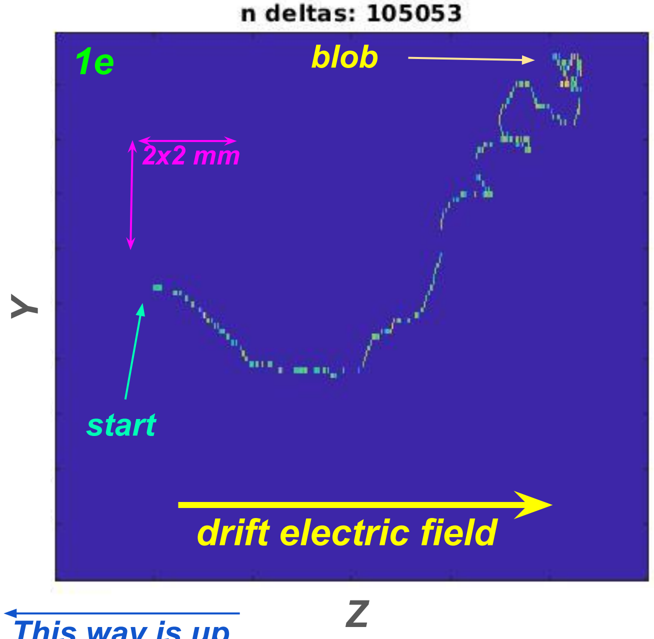
High bg, low diffusion: Need better discrimination

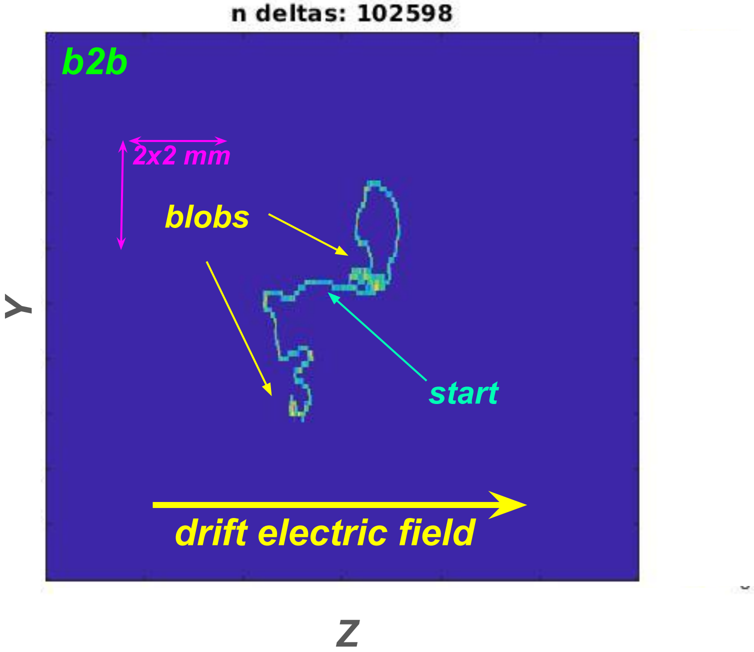
Low bg, high diff.: ~35 counts / 1000d NDBD counts are enough here

To find NDBD, must distinguish it from one dangerous background: a ~2.5 MeV electron.

The task of classification

Our goal is to identify NDBD in LXe under the **ideal conditions**: i.e. distinguish waveforms produced by a single vertically inciding ~2.5 MeV electron *(aka 1e)* deposition from those produced by a deposition by two ~1.25 MeV vertical electrons emitted back-to-back *(aka b2b).*

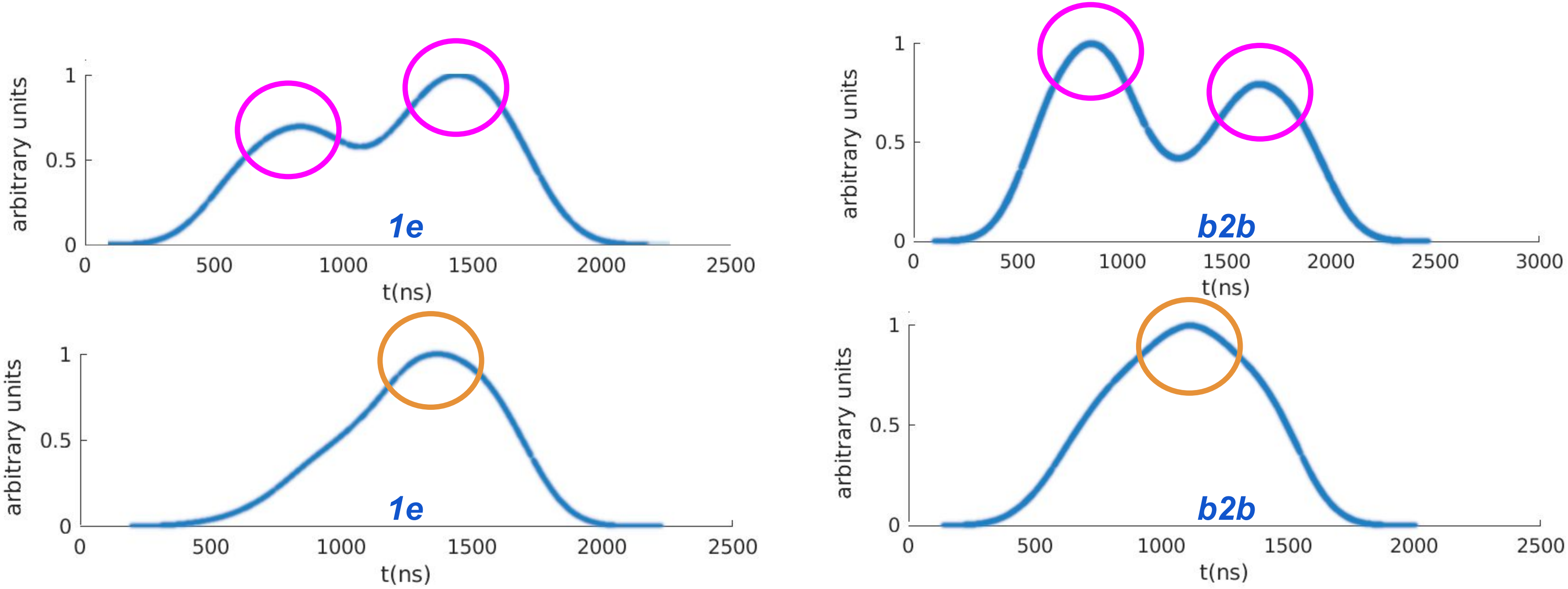




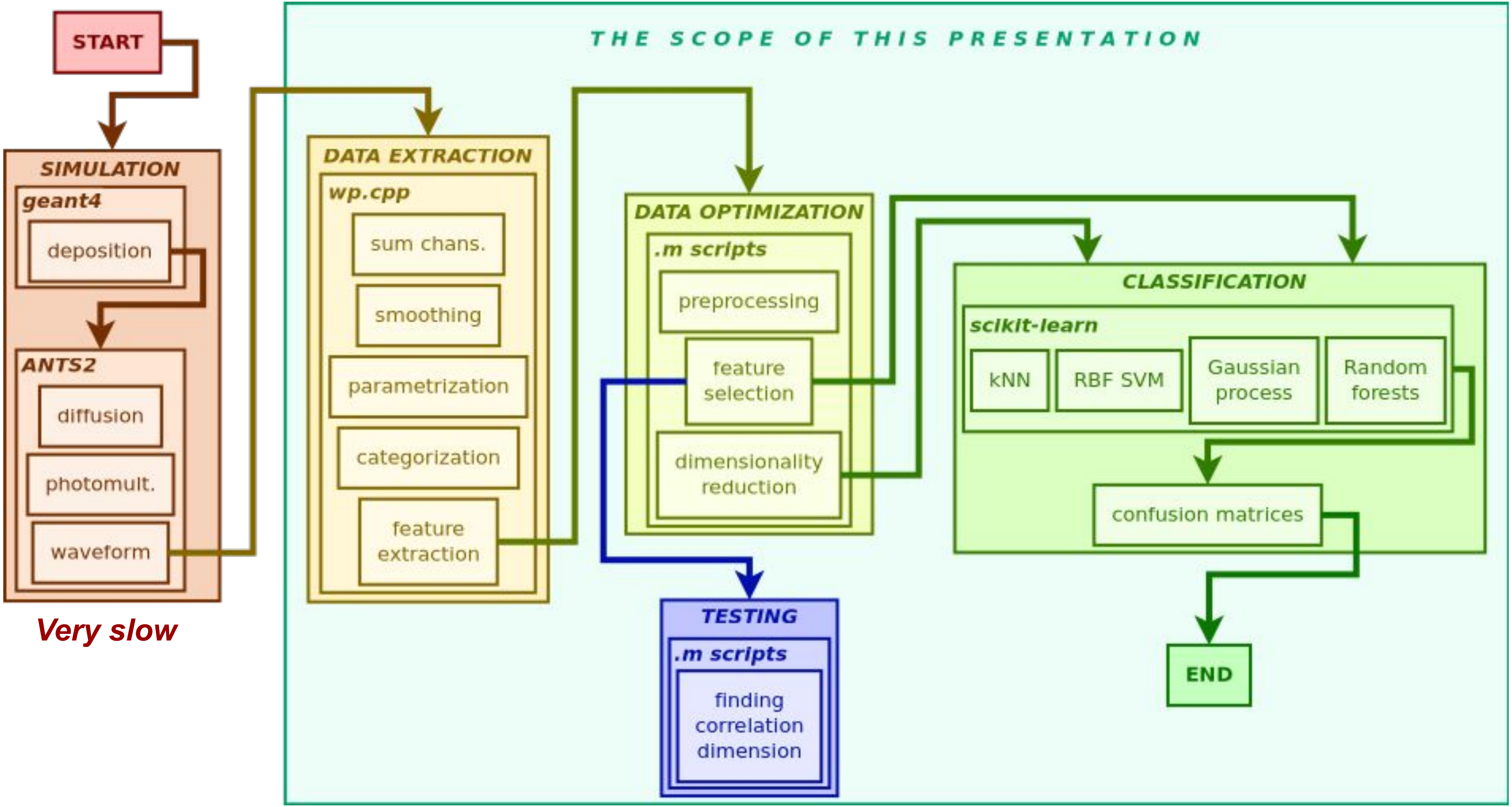
This way is up

The problem of classification

The morphological differences between the waveforms produced by the two types of event are not immediately evident:



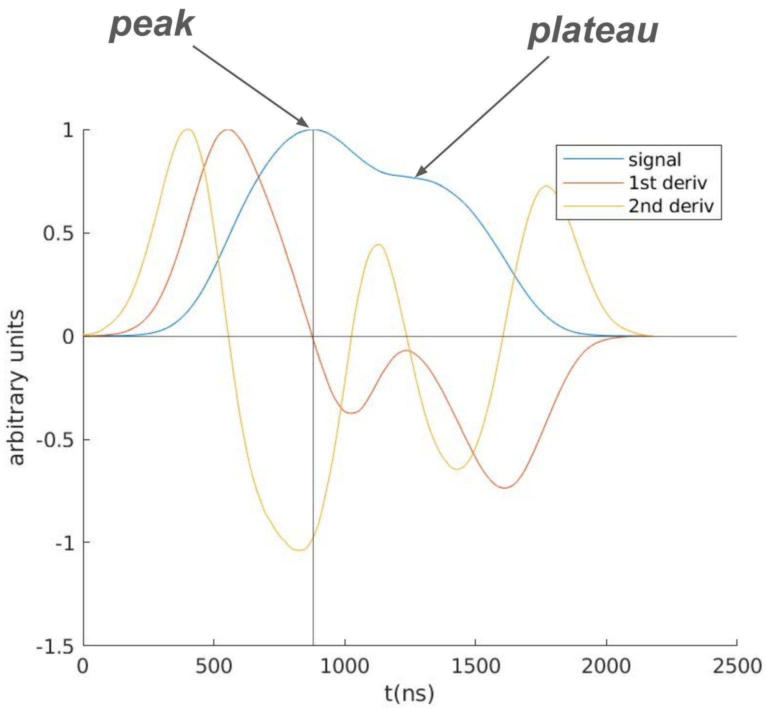
Solution → *parametrization (+ dimensionality reduction) + ML classification*


START
THE SCOPE OF THIS PRESENTATION
SIMULATION
geant4
deposition
ANTS2
diffusion
photomult.
waveform
DATA EXTRACTION
wp.cpp
sum chans.
smoothing
parametrization
categorization
feature extraction
DATA OPTIMIZATION
.m scripts
preprocessing
feature selection
dimensionality reduction
CLASSIFICATION
scikit-learn
kNN
RBF SVM
Gaussian process
Random forests
confusion matrices
TESTING
.m scripts
finding correlation dimension
END

Very slow

Currently extracted params

(Currently the waveform is the sum of the S2 from all PMTs, ie no XY discrimination for now)



b2b, 1e, etc.

- ID (*type* + *number)*
- Bremsstrahlung existence
- Cumulative signal area vs. time
- Heights and times of peaks and plateaus
- Area fraction times
- RMS width, RMS amplitude

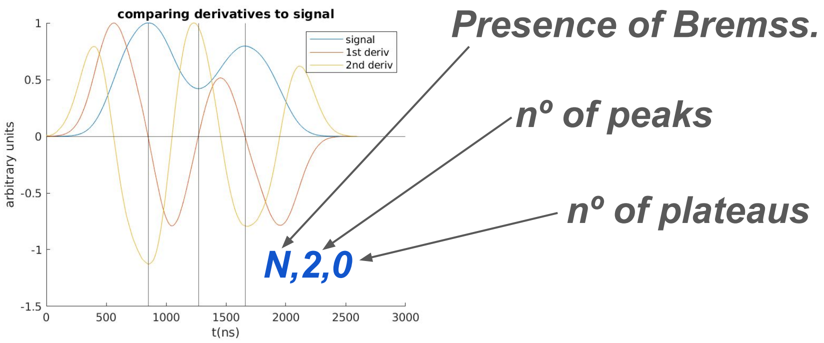
Amplitude unit → *photons detected*

Breaking into datasets via morphology

- Low likelihood of globally continuous parameters; More likely locally continuous

- Simple dimensionality reduction methods assume Gaussian-distributed linear combinations of linearly independent "latent variables" in the parameters

- **Solution**: dataset is divided up according to three characteristics:
  a. ***Presence of Bremsstrahlung;***
  b. ***# of peaks;***
  c. ***# of plateaus;***
- The nomenclature I chose to use to identify the subsets is exemplified to the right

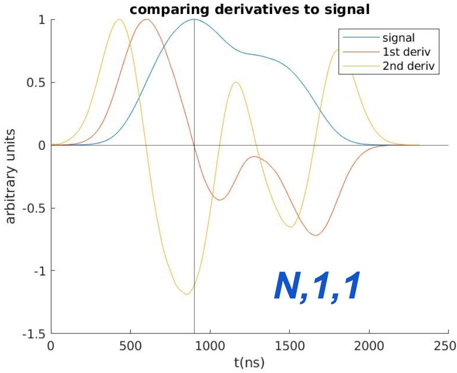


Presence of Bremss.

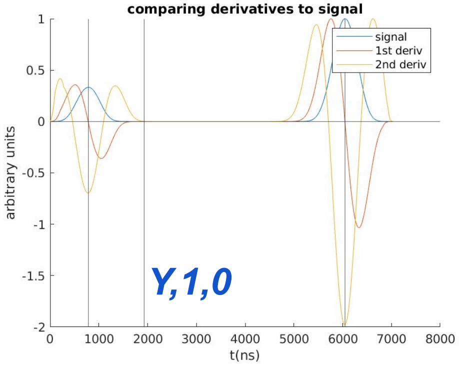
nº of peaks

nº of plateaus

N,2,0



N,1,1



Y,1,0

Feature selection and dimensionality reduction

Dimensionality reduction methods suffer with high dimensionality. Must add feature selection step

Chose seq. floating forward selection (SFFS)

It starts with a minimal feature set and sequentially adds or removes features depending on what ***maximizes*** the value of a certain ***criterion function***

The criterion function I chose is a measure of Euclidean distance of the two classes that is covariance-aware

Simple to implement and good enough performance

Dimensionality reduction performed using classical Multidimensional Scaling (MDS). Recommended for small samples with large dimensionality.

Classical MDS supposes that a centered dataset **Y** can be represented as an output dataset **X** in the space of the latent variables, by finding the orthogonal axis change that best preserves the pairwise scalar products of **Y** so:

Each point a column vector

$$\mathbf{Y} = \boxed{\mathbf{W}}\mathbf{X}$$

As wide, but "shorter"

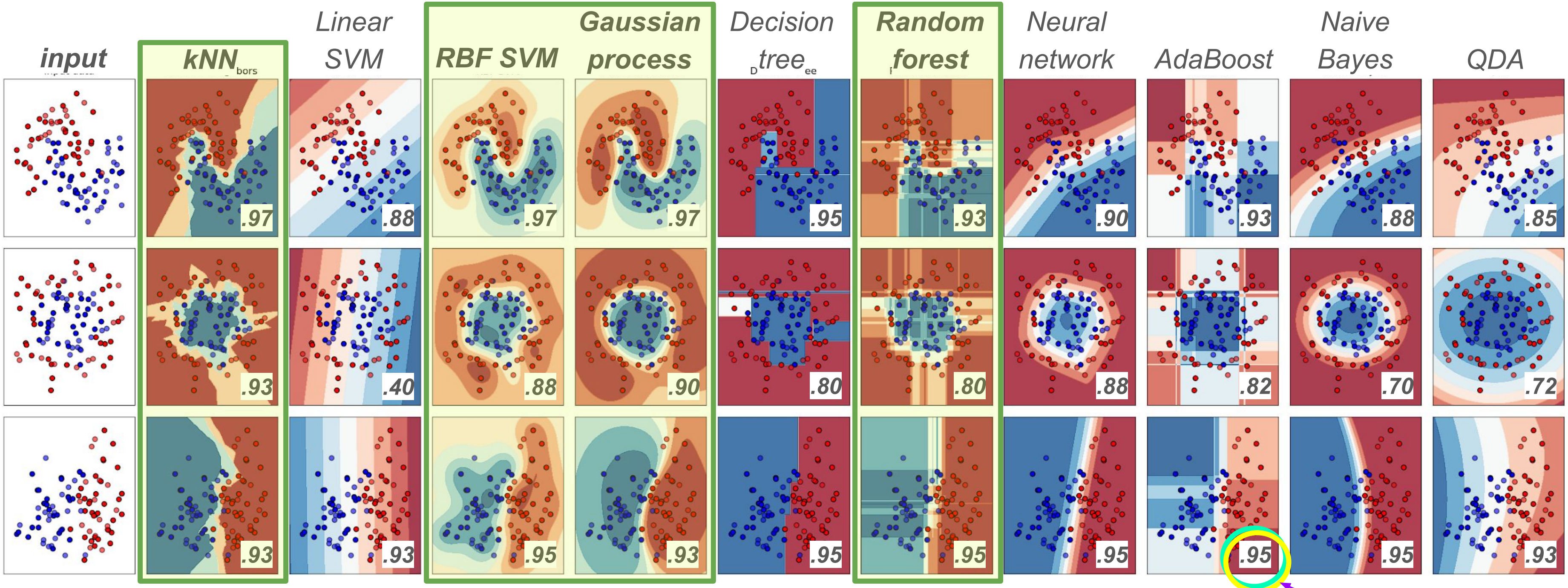
and:

$$\boxed{\mathbf{Y}^T\mathbf{Y}} \boxed{=} \mathbf{X}^T\mathbf{X} = \mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{X}$$

Keep the P largest eigenvalues and corresponding eigenvectors

5. Binary classification → chosen algorithms

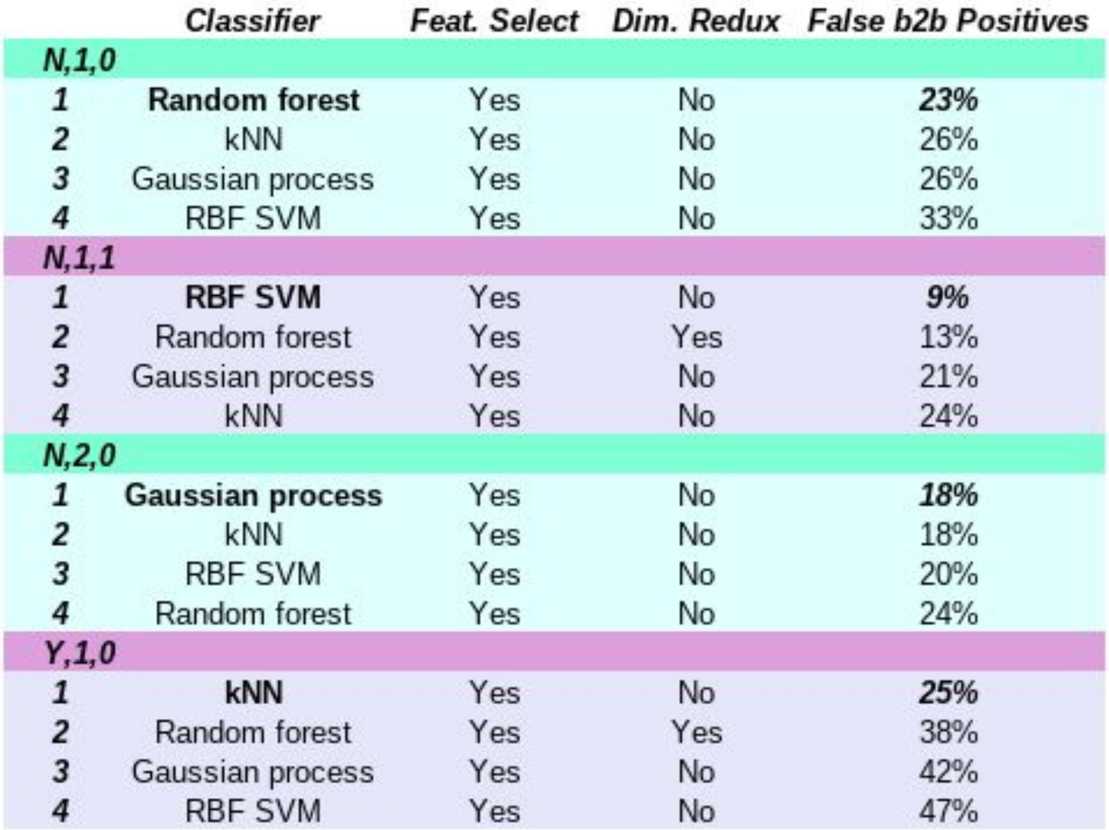
According to performance in the scikit-learn v0.21.3 classifier comparison page, 4 classifiers were chosen. The used classifier parameter values are the default ones



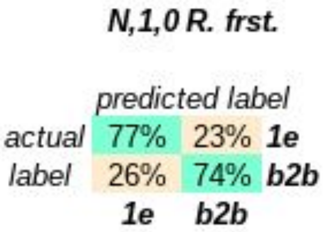
img src: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Fraction of correct predictions

Results → Best classifiers, confusion matrices

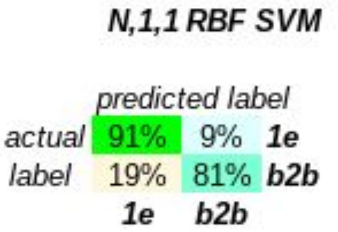


	Classifier	*Feat. Select*	*Dim. Redux*	*False b2b Positives*
N,1,0				
1	**Random forest**	Yes	No	***23%***
2	kNN	Yes	No	26%
3	Gaussian process	Yes	No	26%
4	RBF SVM	Yes	No	33%
N,1,1				
1	**RBF SVM**	Yes	No	***9%***
2	Random forest	Yes	Yes	13%
3	Gaussian process	Yes	No	21%
4	kNN	Yes	No	24%
N,2,0				
1	**Gaussian process**	Yes	No	***18%***
2	kNN	Yes	No	18%
3	RBF SVM	Yes	No	20%
4	Random forest	Yes	No	24%
Y,1,0				
1	**kNN**	Yes	No	***25%***
2	Random forest	Yes	Yes	38%
3	Gaussian process	Yes	No	42%
4	RBF SVM	Yes	No	47%



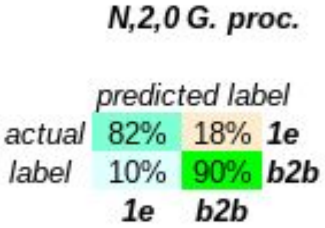
N,1,0 R. frst.

	predicted label		
actual	77%	23%	***1e***
label	26%	74%	***b2b***
	1e	***b2b***	



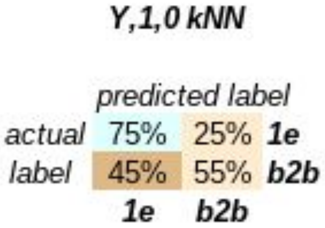
N,1,1 RBF SVM

	predicted label		
actual	91%	9%	***1e***
label	19%	81%	***b2b***
	1e	***b2b***	



N,2,0 G. proc.

	predicted label		
actual	82%	18%	***1e***
label	10%	90%	***b2b***
	1e	***b2b***	



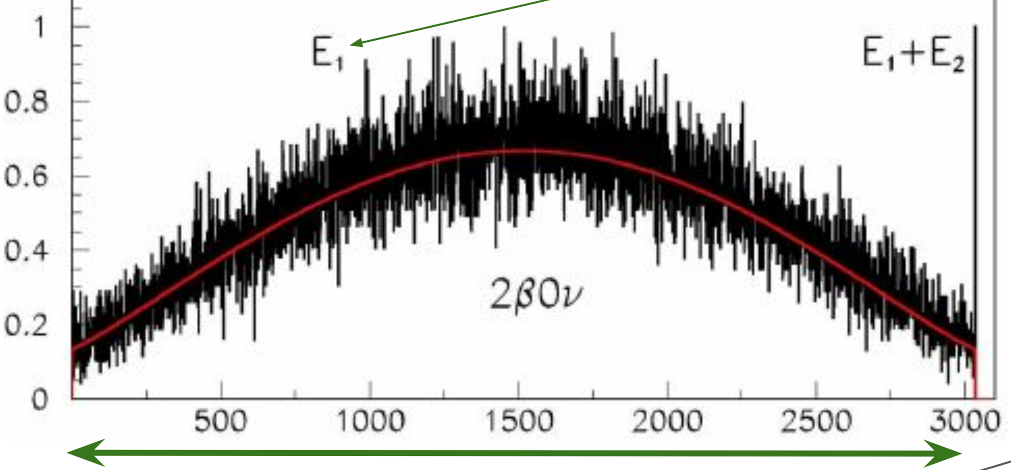
Y,1,0 kNN

	predicted label		
actual	75%	25%	***1e***
label	45%	55%	***b2b***
	1e	***b2b***	

May be usable at top of TPC

Future work:

Energy of one of the electrons



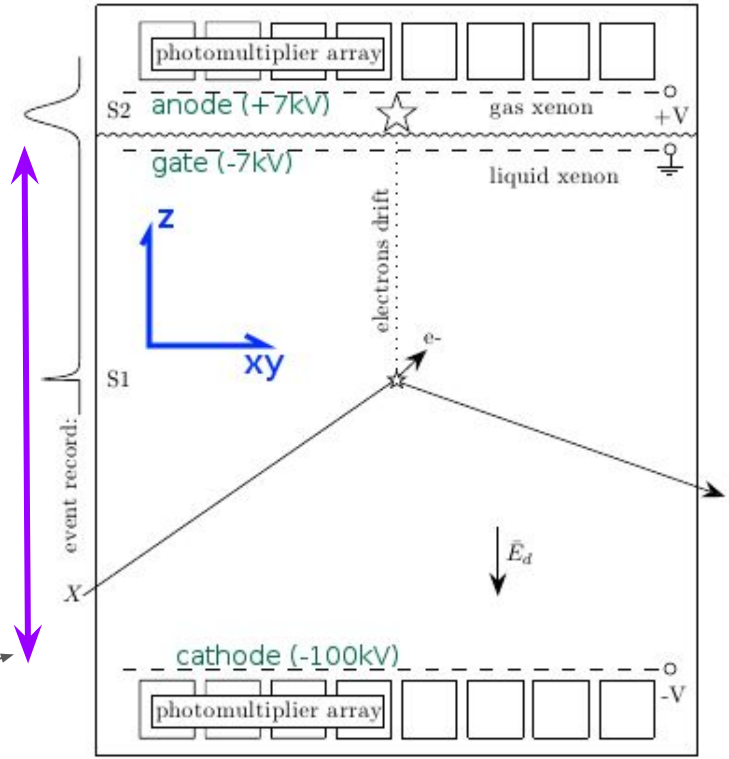
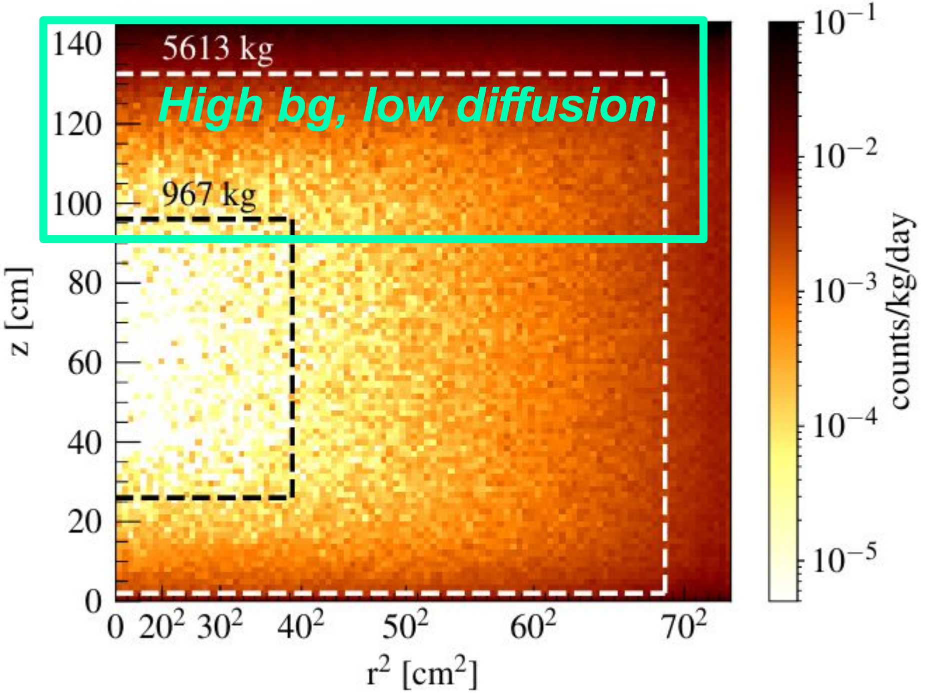


Figure 1: Schematic representation of a liquid xenon time projection chamber, showing interaction of a particle *X*. Primary scintillation photons (S1) are generated at the interaction vertex; electrons are also generated, and drifted to the gaseous xenon where they create proportional scintillation (S2). A typical event record corresponding to such an interaction is shown along the left of the diagram.

Move away from best case scenario: consider these

Conclusions

- *LZ has a competitive environment for NDBD*
- *NDBD overpowered by bg above low bg region*
- *Problem can be mitigated using ML classification*
- *Performed simulation of (NDBD + false-positive) dataset under ideal conditions (~10k each)*
- *Applied classification using 4 classifiers*
- *Good enough result to increase sim. complexity*

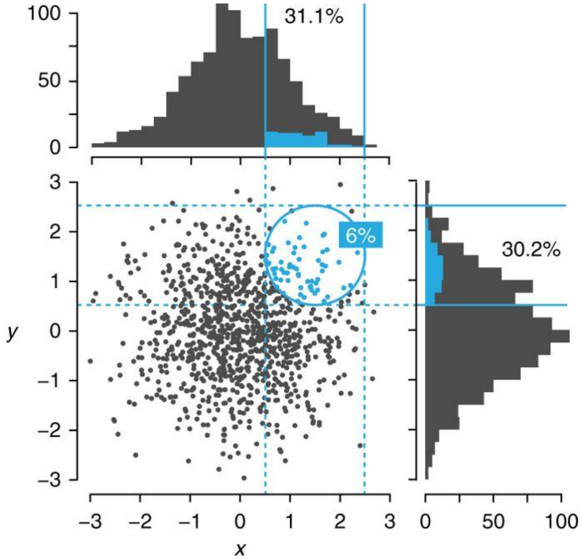


Thank you!

Appendices

The curse of dimensionality

More likely than not, the ability to effectively distinguish the two classes of events (1e vs. b2b) will be predicated on the use of **more parameters, rather than less**: the resulting increase in dimensionality improves the SNR. However it also brings along the risk of being subject to the curse of dimensionality. In a few words, the **curse of dimensionality** is a loss in the “descriptivity” of a dataset due to it becoming increasingly sparse with the rising number of parameters.



Altman, Krzywinski – Nat. Methods, 15, pg. 399–400 (2018)

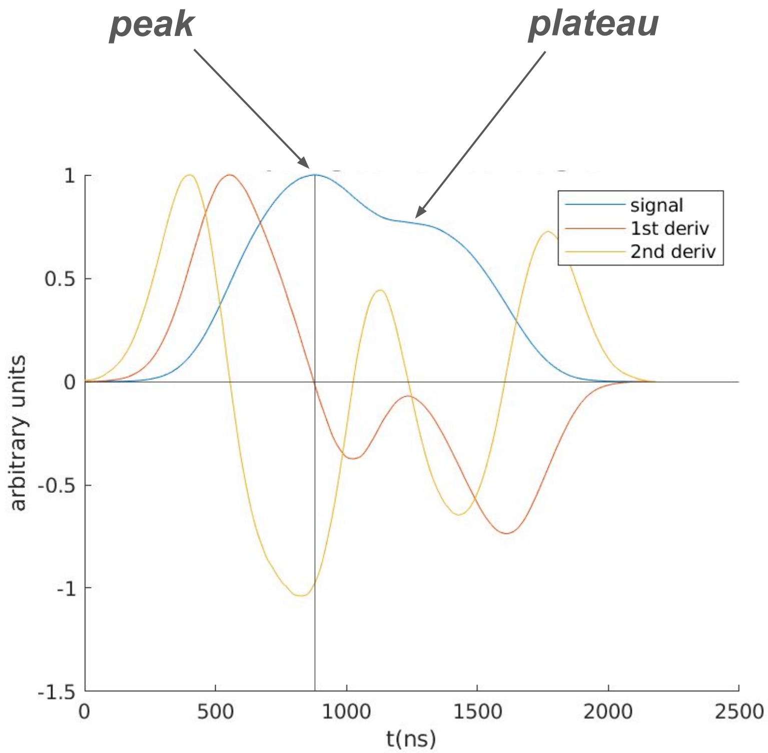
To mitigate the curse of dimensionality, it is astute to subject the data to a dimensionality reduction step. Dimensionality reduction, meanwhile, requires a prior feature selection step. The next slide shows an overview of the entire procedure I chose.

Chosen procedure from waveforms to classifier

Below is a summary. This presentation explains it in full:

1. From the ***waveform***, extract a set of “raw” ***parameters***, forming a raw dataset
    a. *Categorize **dataset** into **subdatasets** according to the values of some of the parameters*
2. Obtain a subdataset of ***features*** from parameters or parameter combinations
3. ***Feature selection*** on the feature subdataset to avoid curse of dimensionality
    a. *Check the **correlation dimension** of the subdataset*
4. On the feature-selected subdataset, perform ***dimensionality reduction***
5. On the dimension-reduced subdataset, test binary ***classification*** algorithms

1. W_form → params: (Pk. + Plat.) x (amp.’s + t’s)



Peak *(local maximum in signal)**:***

- *1st deriv → zero*
- *2nd deriv → below zero*

Plateau *(local abs. minimum in slope)**:***

- *2nd deriv → zero*
- *3nd deriv → above zero*

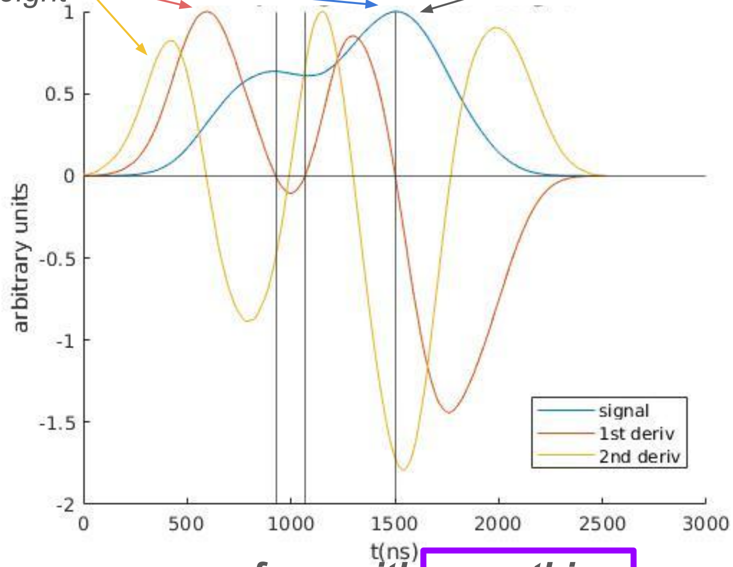
The parametrizer saves `std::vector`s containing pk. times / heights and plat. times / heights. The `.size()` of these corresponds to the # of pks. and plats. in the signal, respectively.

To perform derivation, signal ***must be very smooth.*** *Smoothing on some point in the signal is currently done by averaging over that point’s neighborhood.*

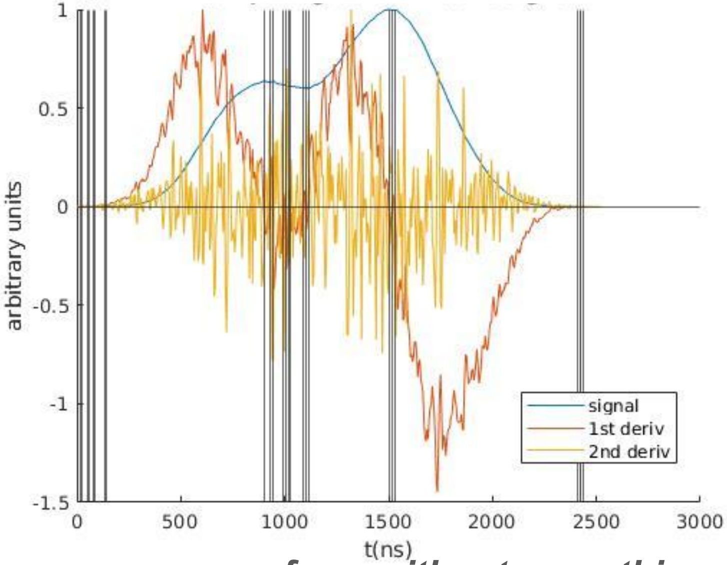
1. W_form → params: “Must be very smooth”

Normalized to max height

~28 x 10^6 photon hits



waveform with smoothing



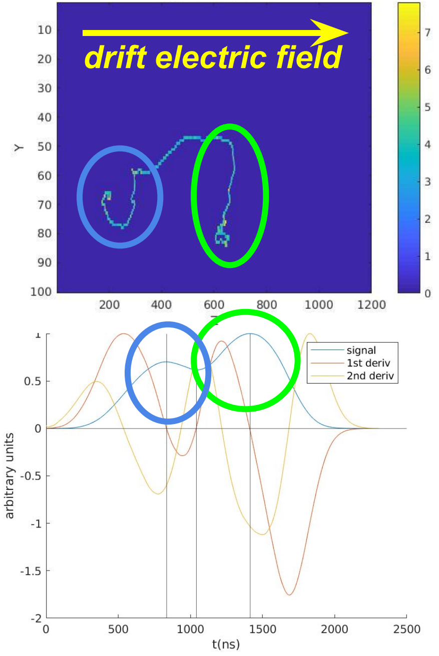
same waveform without smoothing

Procedure: *smooth the waveform with 20 neighbor moving average,*
get 1st deriv.; smooth 1st deriv with 30 neighbor moving avg,
get 2nd deriv.; smooth 2nd deriv with 40 neighbor moving avg

1. W_form → params: Bremsstrahlung existence

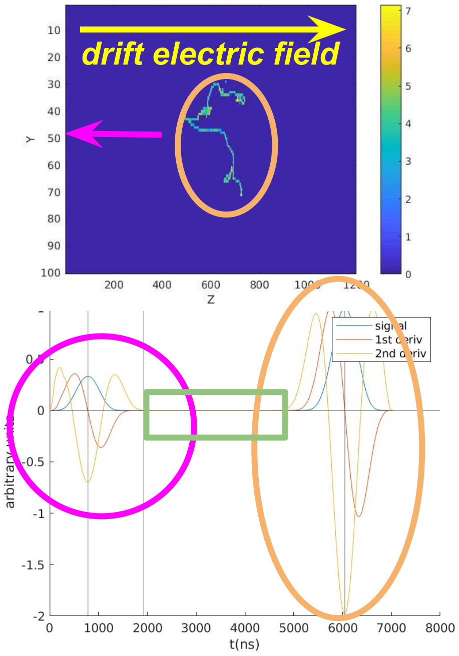
Bremsstrahlung photons result in secondary depositions and hence result in their own secondary pulses

# of deltas →**104488**



Ordinary case, ***no Brem***

# of deltas →**77986**



Brem. *(note smaller # of δ's)*

- If the photon is roughly vertical, and travels a certain distance before interacting, it results in a pulse well separated from the electron deposition pulse.
- The parametrizer currently detects Brem by the presence of trails of zeros between depositions
- The non-Brem deposition is currently taken to be the one giving the highest peak
- Once a non-Brem deposition is chosen, the secondary depositions are discarded from the signal

1. W_form → params: RMS width

A measure of the width of the signal performed by accounting for the deviations from the 50% area fraction time

Currently opted for over FWHM on account of seeming to be more descriptive of the shape of the signal

$$RMS_{Width} = \sqrt{\frac{1}{\boxed{S_{5-95}}} \times \sum_{j=j_5}^{j_{95}} s_j(t_j - \boxed{\tau})^2}$$

j_5 , j_{95} == *index for 5% and 95% area fraction times*

s_j == *signal at index j*

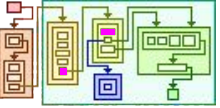
t_j == *j * sampling period (in ns)*

$$S_{5-95} = \sum_{j=j_5}^{j_{95}} s_j$$

Signal area from 5% to 95%

$$\sum_{j=0}^{\tau} s_j = S/2$$

τ == 50% area fraction time



2. Parameters → Features

Currently used features:

- Amplitude averages for different windows:
  - *Centered:*
    - [5 - 95]%, [10 - 90]%, [25 - 75]% area
  - *Left-leaning:*
    - [5 - 50]%, [5 - 75]%, [5 - 95]% area
  - *Right-leaning*
    - [10 - 95]%, [25 - 95]%, [50 - 95] % area
- Peak heights, peak times (after 5% area fraction)
- Max peak height, time of the max peak
- Plateau heights, plateau times (after 5% area frac.)
- RMS width
- RMS amplitude

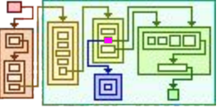
Roughly 24-30 features. All features preprocessed via normalization and division by the variance

Future features:

- Skewness
- More diverse windows
- Additional morphology? (e.g. straight lines)

Future considerations:

- *The waveform in the real case will likely have a lot of the morphological information erased by saturation. Possibly throughout all channels. It is important to think of features that will be resistant to this.*



3. Feature selection

Dimensionality reduction methods are sensitive to the curse of dimensionality. It is generally recommended to perform a feature selection step before moving into dimensionality reduction.

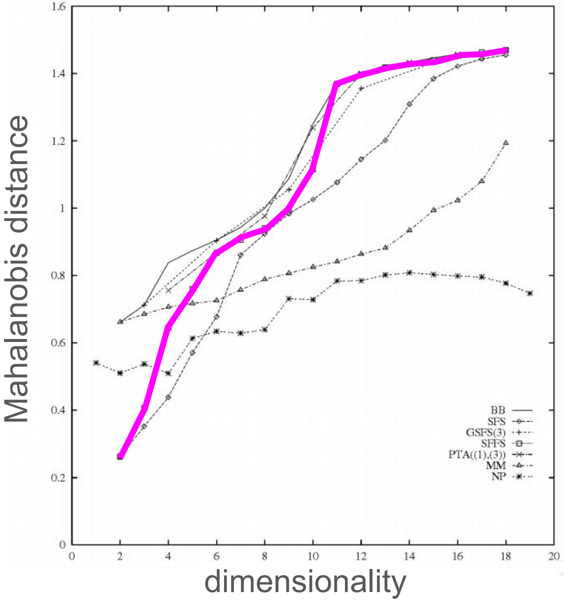
I chose sequential floating forward selection (SFFS)

SFFS Algorithm:

In: subdataset $\mathbf{T_x}$; ***Out***: feature-selected subdataset $\mathbf{T_y}$

- $\mathbf{T_y}$ begins as the two least correlated columns from $\mathbf{T_x}$
- **while(** Steps 1 and 2 together alter the columns in $\mathbf{T_y}$ **):**
    - **Step 1 → inclusion**
        - Of the columns in $\mathbf{T_x} \setminus \mathbf{T_y}$, concatenate to $\mathbf{T_y}$ the column whose inclusion in $\mathbf{T_y}$ maximizes the criterion function $\mathbf{J(T_y)}$
    - **Step 2 → conditional exclusion**
        - Find the column in $\mathbf{T_y}$ whose exclusion maximizes $\mathbf{J(T_y)}$
        - If $\mathbf{J(T_y)}$ in Step 2 is larger than in Step 1, remove that column from $\mathbf{T_y}$

Jain, Zongker -- IEEE Trans. Pattern Anal. Mach. Intell., Vol. 19, Nº 2, pp. 153 - 158



The criterion function is currently the Mahalanobis distance between the two classes (1e vs. b2b):

$$\mathbf{J(T_y)} = (\mu_1 - \mu_2)^t\ \Sigma^{-1}\ (\mu_1 - \mu_2)$$

- $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$ → *mean vectors of the two classes*
- $\boldsymbol{\Sigma}$ → common covariance matrix *(i.e. the average of the self-covariance matrices of the two classes)*

3.a Correlation dimension → implementation pt.1

*The correlation dimension is the q-dimension with **q = 2**. What is a q-dimension? Read below.*

The q-dimension is an extension of the concept of the fractal dimension*, itself a generalization of the intrinsic dimension of a topological space. Datasets, being embeddings on a topological manifold, are therefore fit to be described by such a concept. For a dataset of size **N**, below is the definition of q-dimension:

$$D_q = \lim_{\epsilon \to 0} \frac{\log \boxed{C_q(\epsilon)}}{(q-1)\log \epsilon}$$

*<b>q</b>-th order Minkowski norm on the **i**-th and **j**-th datapoint of the dataset. For **q = 2**, this is the Euclidean norm*

H(u)** returns **1** if **u >= 0**, else **0

$$C_q(\epsilon) \approx \frac{1}{N(N-1)} \sum_{\substack{i=j \\ i<j}}^{N} \boxed{H(\epsilon - \boxed{||\mathbf{y}(i) - \mathbf{y}(j)||_q})}$$

*It's a measure of the proportion of points that are within a **q**-th order Minkowski distance **ε** of each other*

The next slide explains how to calculate the correlation dimension in practice

*The fractal dimension is the q-dimension for q = 0, but that doesn't matter for us

3.a Correlation dimension → implementation pt.2

By definition, **D_2** is the correlation dimension:

$$d_{corr} = D_2 = \lim_{\epsilon \to 0} \frac{\log \boxed{\widehat{C}_2(\epsilon)}}{\log \epsilon}$$

However both the numerator and the denominator go to -∞, so by l'Hôpital's rule:

$$d_{corr} = D_2 = \lim_{\epsilon \to 0} \frac{\partial \log \widehat{C}_2(\epsilon)}{\partial \log \epsilon}$$

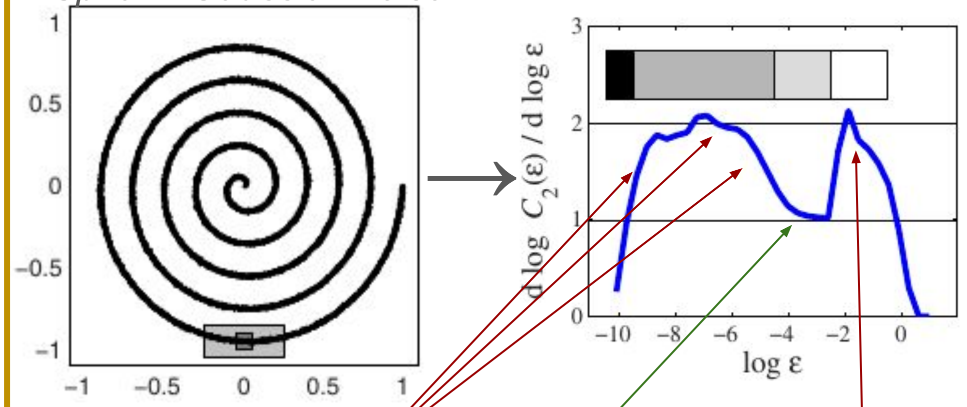
In practice, the scale-dependent correlation dimension is used instead:

$$\widehat{d}_{corr}(\epsilon_1, \epsilon_2) = \frac{\log \widehat{C}_2(\epsilon_2) - \log \widehat{C}_2(\epsilon_1)}{\log \epsilon_2 - \log \epsilon_1}$$

This scale-dependency allows us to select a "window" size where the intrinsic dimension is most relevant to us, so it's more useful than the fractal dimension, which just gives one number.

*In practice **$C_2(\varepsilon)$** is calculated by computing the Euclidean norms for all possible pairs of datapoints in the dataset, and then listing the proportion of those norms that are less than or equal to **ε***

Spiral + Gaussian noise



Too small, only "see" scattered points

Perfect! "See" spiral

Too large, "see" a 2D blurry circle

img. src: Lee, Verleysen -- Nonlinear Dimensionality Reduction, Springer, Fig 3.3.

4. Dimensionality reduction - getting latent vars

The method I'm currently using for dimensionality reduction is classical Multidimensional Scaling (MDS). Its use is recommended for the case of relatively small samples with large dimensionality.

Classical MDS supposes that a centered, **N** input dataset **Y**, with **D** features, can be represented as an output dataset **X**, with **P < D** latent variables, by finding the orthogonal axis change that best preserves the pairwise scalar products of **Y**, so:

***D x N** matrix* → $Y = \boxed{W}X$ ← ***P x N** matrix*

and:

$$\boxed{Y^TY}\boxed{=}X^TX = X^TW^TWX$$

Y

Y^TY being a square matrix, it can be eigenvalue decomposed, and so:

$$Y^TY = U\Lambda U^T$$

meaning that:

$$X^TX = U\Lambda U^T = (\Lambda^{1/2}U^T)^T(\Lambda^{1/2}U^T)$$

The latent variables will be the ones with the **P** largest corresponding eigenvalues, and then:

$$\hat{X} = I_{P\times N}\Lambda^{1/2}U^T$$

So the method consists of performing the eigenvalue decomposition of Y^TY, sorting it by the largest eigenvalues, selecting the P first ones and then constructing X that way. Next slide explains how to get W.

4. Dimensionality reduction - calculating W

Knowing how to find **W** is easier to understand with some background on Principal Component Analysis (PCA). Like MDS, PCA assumes **Y = WX**, they're equivalent, but PCA works by minimizing the reconstruction error:

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}}\, E_{\mathbf{y}}\left\{||\mathbf{y} - \mathbf{W}\boxed{\mathbf{W}^T\mathbf{y}}||_2^2\right\}$$

Expectation value over every vector **y** in **Y**

This is **x**. Given that **y = Wx** and **W^{T}W = 1**, then **W^{T}y = x**

Unpacking the Euclidean norm and simplifying:

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}}\left[\, E_{\mathbf{y}}\{\mathbf{y}^T\mathbf{y}\} - E_{\mathbf{y}}\{\mathbf{y}^T\mathbf{W}\mathbf{W}^T\mathbf{y}\}\right]$$

$$\mathbf{W} \approx \underset{\mathbf{W}}{\operatorname{argmax}}\ \operatorname{tr}(\mathbf{Y}^T\mathbf{W}\mathbf{W}^T\mathbf{Y})$$

By substituting **Y** by its singular value decomposition **Y = VΣU^T**, knowing that **U** and **V** are unitary, we get:

$$\mathbf{W} = \mathbf{V}\mathbf{I}_{D\times P}$$

From **Y^{T}Y = UΛU^T** *(note here that the **U** from the eigenvalue decomposition and the **U** from singular value decomposition are equal),* we find that **Λ = Σ^{T}Σ**. From there, the fact that **U** and **V** are unitary give us:

$$\mathbf{Y} = \mathbf{V}\Sigma\mathbf{U}^T \Rightarrow \mathbf{Y}\mathbf{U} = \mathbf{V}\Sigma$$

Now, because **Σ** is a **D x N** rectangular diagonal matrix, its pseudoinverse is **Σ^{-1} = (Λ$^{-1/2}$)I$_{N\ x\ D}$** , hence:

$$\mathbf{V} = \mathbf{Y}\mathbf{U}(\Lambda^{-1/2})\mathbf{I}_{N\times D}$$

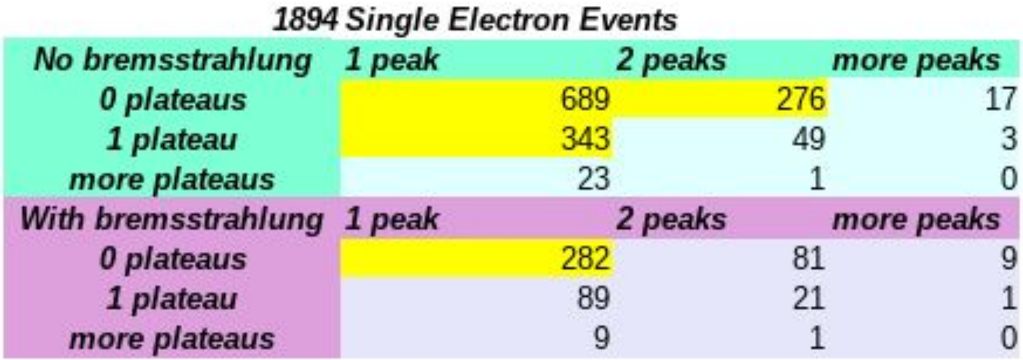
and:

$$\hat{\mathbf{W}} = \mathbf{Y}\mathbf{U}(\Lambda^{-1/2})\mathbf{I}_{N\times P}$$

By this method, the rotation matrix **W** can be retrieved strictly from the elements outputted by the MDS.

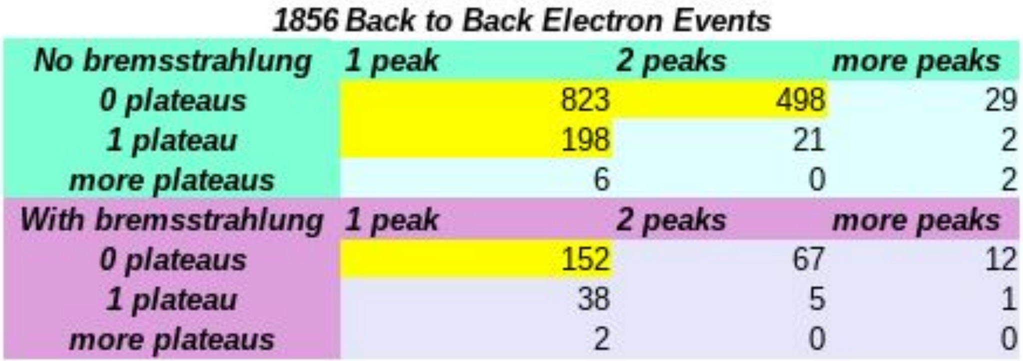
… I just realized that I could have obtained exactly the same expression through W = YX^{-1}

Results →currently obtained subdatasets



1894 Single Electron Events

No bremsstrahlung	1 peak	2 peaks	more peaks
0 plateaus	689	276	17
1 plateau	343	49	3
more plateaus	23	1	0
With bremsstrahlung	1 peak	2 peaks	more peaks
0 plateaus	282	81	9
1 plateau	89	21	1
more plateaus	9	1	0



1856 Back to Back Electron Events

No bremsstrahlung	1 peak	2 peaks	more peaks
0 plateaus	823	498	29
1 plateau	198	21	2
more plateaus	6	0	2
With bremsstrahlung	1 peak	2 peaks	more peaks
0 plateaus	152	67	12
1 plateau	38	5	1
more plateaus	2	0	0

Highlighted in yellow are the subdatasets with a minimally satisfactory amount of datapoints. For the classification it was chosen to have a 50/50 ratio of events of either class to prevent overfitting, meaning that the extra events in the class with the larger statistics were discarded, leaving us with the following datasets:

- ***N,1,0 - 1378 pts , 24 features***
- ***N,1,1 - 396 pts , 26 features***
- ***N,2,0 - 552 pts , 26 features***
- ***Y,1,0 - 304 pts , 24 features***

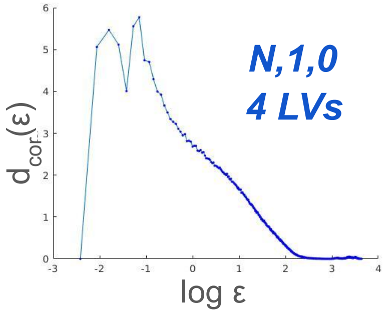
Results →Feature selection + dimens. reduction

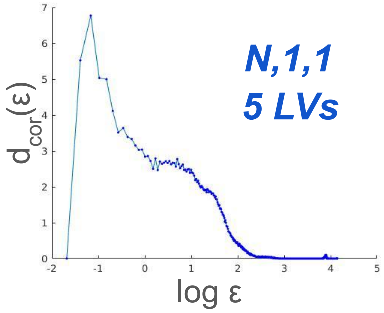
As described previously, feature selection was performed using the SFFS algorithm, with the Mahalanobis distance of the two classes as the criterion function. The resulting number of features and Mahalanobis distance are listed below.

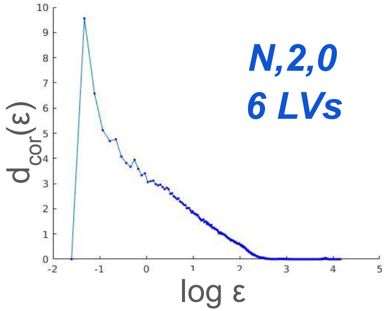
- ***N,1,0 - 14 features, 1.06 dist***
- ***N,1,1 - 17 features, 3.92 dist***
- ***N,2,0 - 16 features, 2.17 dist***
- ***Y,1,0 - 14 features, 1.25 dist***

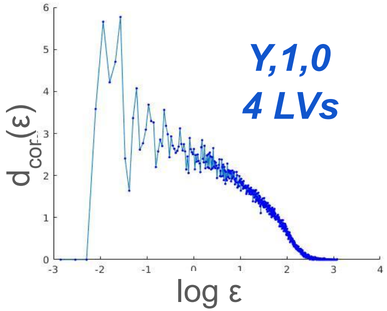
Presumably these distances are in units of variance, so the distance ranges between 1σ and 2σ.

The feature-selected datasets were then subjected to a dimensionality reduction step. The correlation dimension and actually used number of latent variables *(LVs)* are listed below:



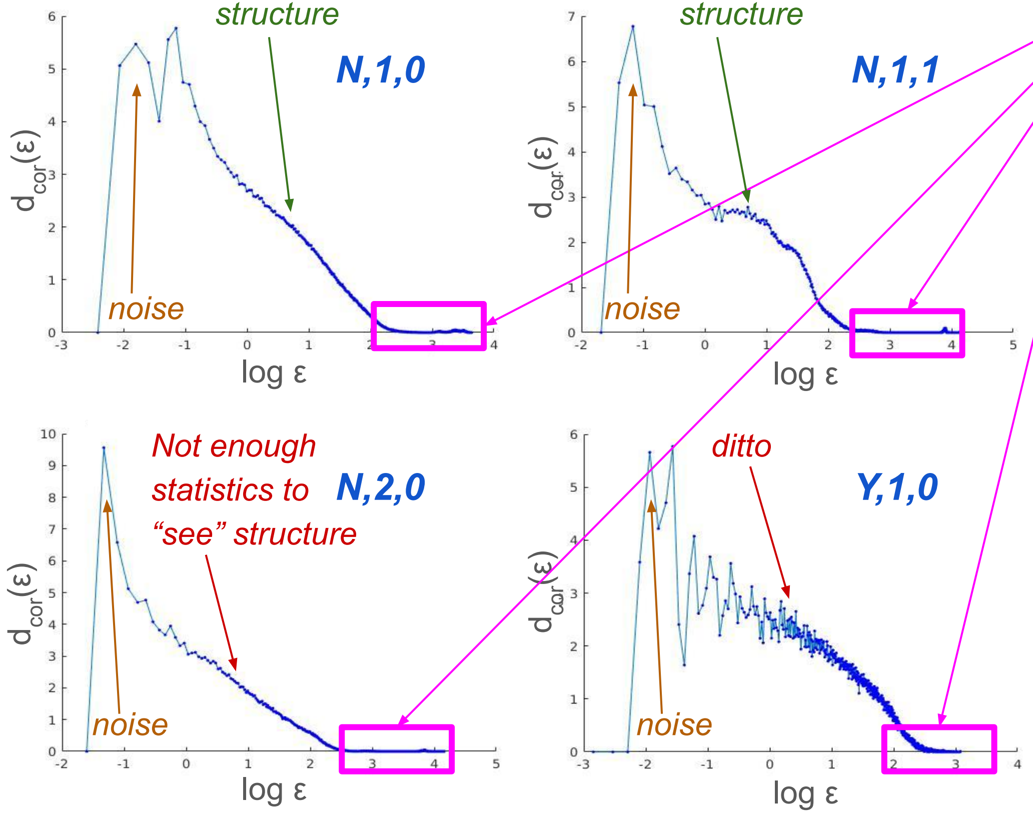


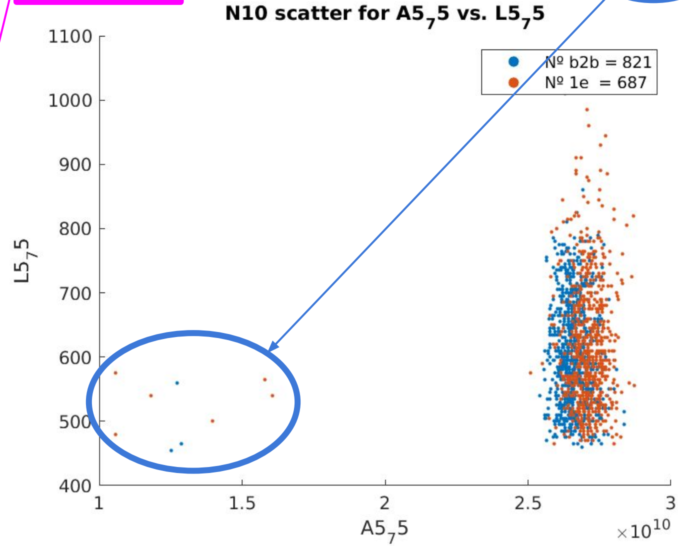




Correlation dimension results

This here is happening because of these.





I imagine that the outliers are because of Bremsstrahlung falling on the same z plane as the electron deposition, thus becoming “invisible”. Remove them?