Machine learning in LZ

P. Brás

1st BigDataHEP meeting @ Coimbra

2019-01-11

LZ data and pulses

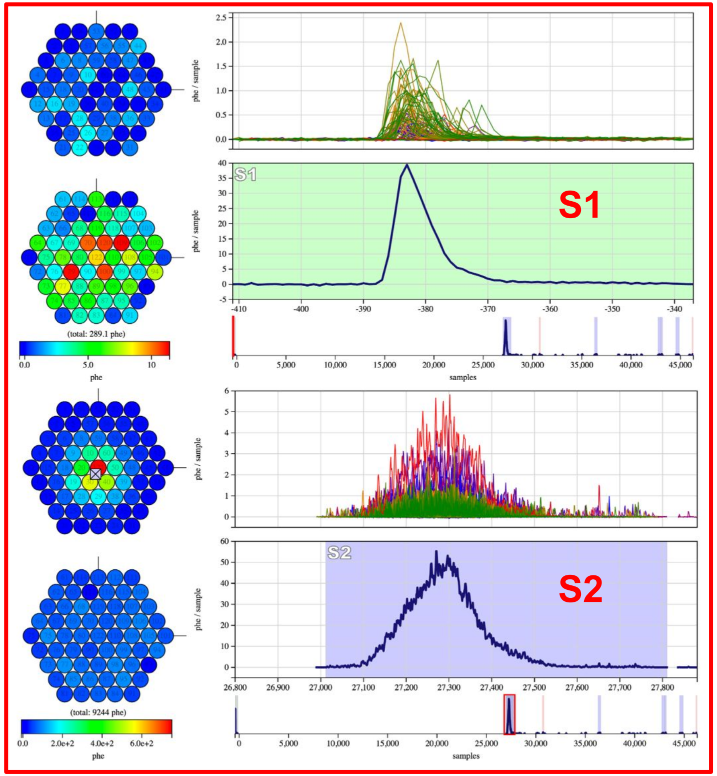
LZ is a dual-phase xenon TPC aimed at the detection of dark matter (WIMPs).

Two main signals are expected when an interaction is recorded:

- Prompt scintillation light **(S1 signal)**
- Delayed proportional scintillation from drifting ionization charge **(S2 signal)**

Identifying these pulses correctly allows for a full description of the event

LUX data



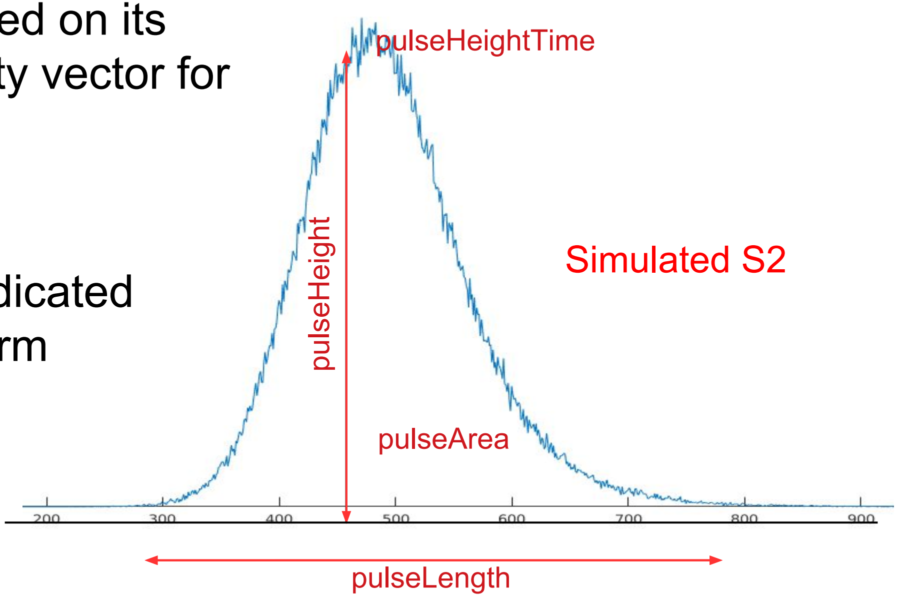
Pulse Classification for LZap

Goal: Identify the nature of a given pulse based on its geometrical parameters, returning a probability vector for all considered topologies +1

[S1, S2, SPE, SE, MPE, Other]

Input: 17 pulse parameters obtained by a dedicated algorithm (pulseParametrizer) OR full waveform

- Pulse area (pA)
- Pulse amplitude (pH)
- Pulse length (pL, pL90 - length at 90% area)
- Prompt fraction (pF)
  fraction of area at start of pulse (50, 100, 200, 500, 1k, 2k and 5k ns)
- Top-bottom asymmetry (TBA) - $(A_{top}-A_{bottom})/(A_{top}+A_{bottom})$
- Area fraction time (aft)
  time at XX% integrated area (5%, 25%, 50%, 75%, 95% area)



Current Classifier in LZap

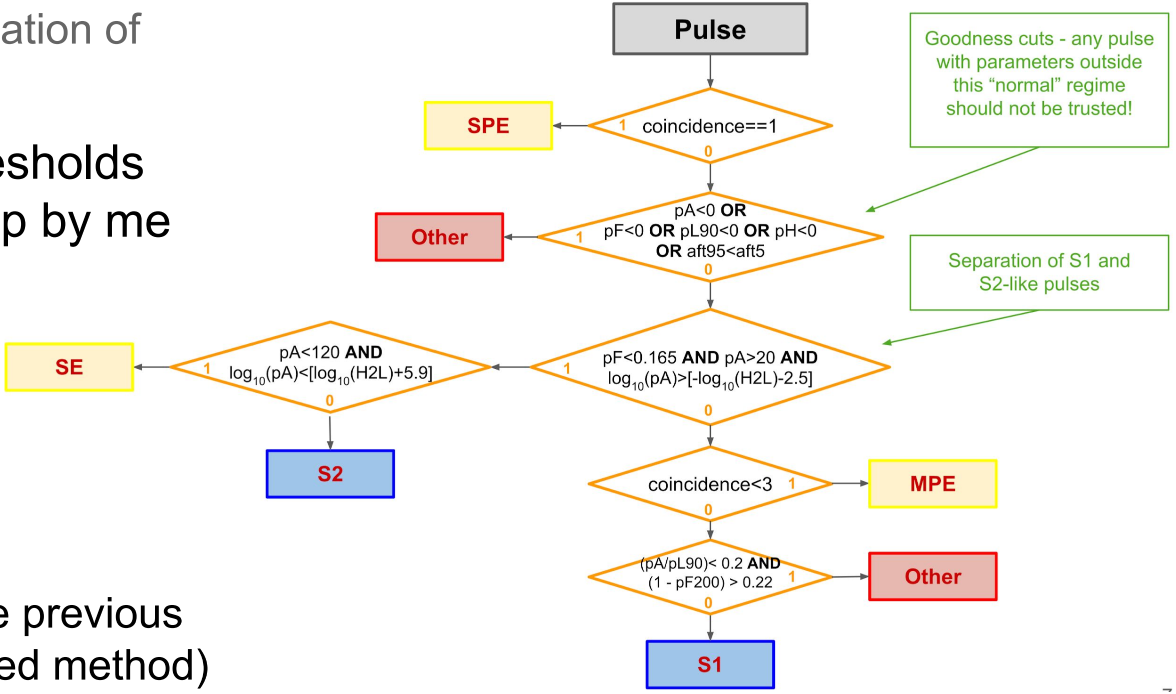
HADES

Heuristic Algorithm for Discrimination of Event Substructures

Basic classifier based on thresholds currently implemented in LZap by me

- Robust (simple)
- Easy to modify on the fly
- Benchmark method
- Purely categorical
- Efficiency > 99.5%

Created to solve a problem in the previous classifier (COMPACT - PDF-based method)



Machine Learning for Pulse Classification

Tools used:

1. **Keras** - F. Chollet et al. (2015) https://keras.io
    a. Artificial neural networks
    b. Convolution neural networks
2. **Scikit-learn** - Pedregosa et al., JMLR 12, pp. 2825-2830 (2011)
    a. Random Forest

Data used:

1. LUX simulated data: **176k** pulses from low-E events
2. LZ simulated data:
    a. Mock Data Challenge 1 data (clean pulses) - **300k** pulses from low-E events
    No visualization tools available & limited info on MCtruth
    b. MDC2 data (realistic waveforms) - **7.6M** pulses of 5 classes (actually 4)
    No pulse-level MCTruth available!!! - used results of HADES for training

ML for Pulse Classification - 1

Artificial neural networks
(single-layer perceptrons only)

Motivations:

1. Fast processing time
2. Powerful, robust, readily available and easy to use (e.g. Keras)
3. Can be easily implemented in the framework of LZ

Tests performed in LUX simulated data, LZ MDC1 and MDC2 simulated data

Convolution neural networks

Motivations:

1. Trying to bypass the “parametrizer” feeding it raw waveforms
2. Detect features beyond the geometrical parameters used now
3. Maybe incorporate pulse identification and classification in the same module

Tests performed with synthetic pulses produced to look like real S1 and S2 signals of different shapes and sizes

Classification NN using Keras

A simple neural net with 2x15 fully connected layers

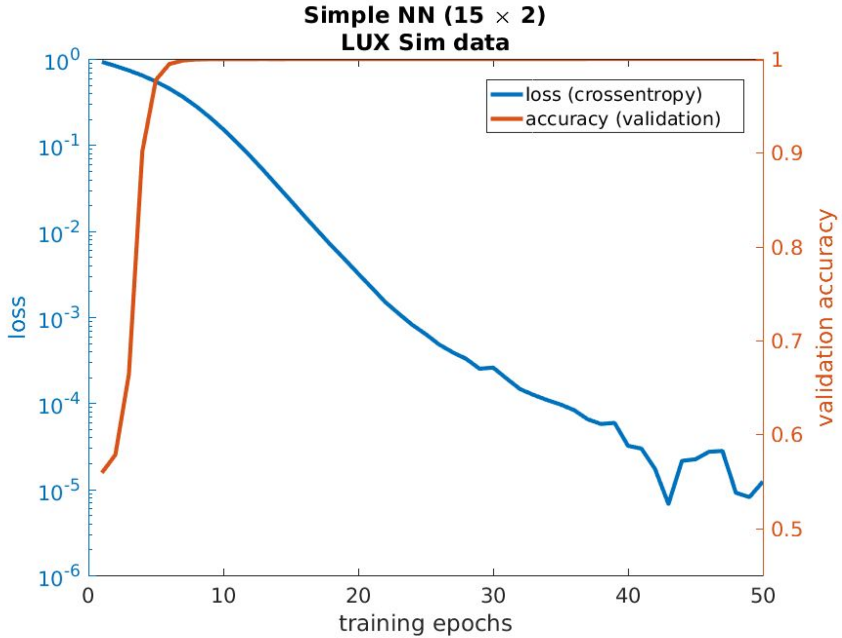
Trained and tested with LUX simulated data:

Clean s1, s2 and SE pulses (no SPE)

Input: 4 modified parameters **[pF TBA H2L pS]**

Output: probability vector **[s1 s2 other]**

- 87975 s1 and s2 pulses total
- 10% used for validation
- Trained using batches of 5000 samples
- **100% classification accuracy (<20 epochs)**
    - Pre-selected pulses (no SPEs or Others)
    - Small dataset, low diversity



Classification NN using Keras

A simple neural net with 2x15 fully connected layers

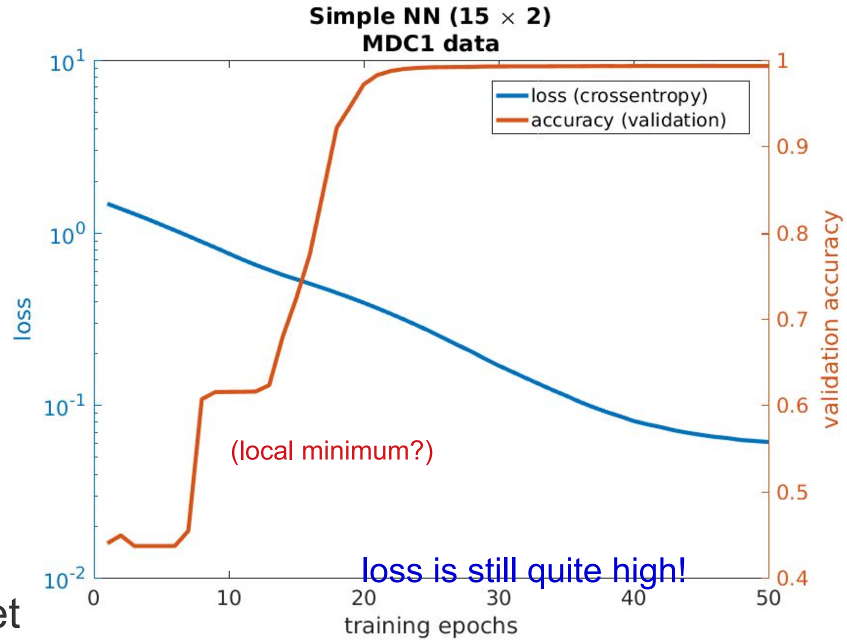
Trained and tested with pre-MDC1 data:

MCtruth labels (s1, s2, cherenkov, scint.)

Input: 4 modified parameters **[pF TBA H2L pS]**

Output: prob. vector **[s1 s2 chrk scnt other]**

- 299817 pulses total
- 10% used for validation
- Trained using batches of 20000 samples
- **99.32% accuracy (<35 epochs)**
    - Probably due to low statistics of pulses labeled “scintillation” (0.7%)
    - COMPACT* outperforms it for this dataset



*Complex Pulse Analysis and Classification Tool - previous LZap classifier

Classification NN using Keras

A neural net with 2x41 fully connected layers

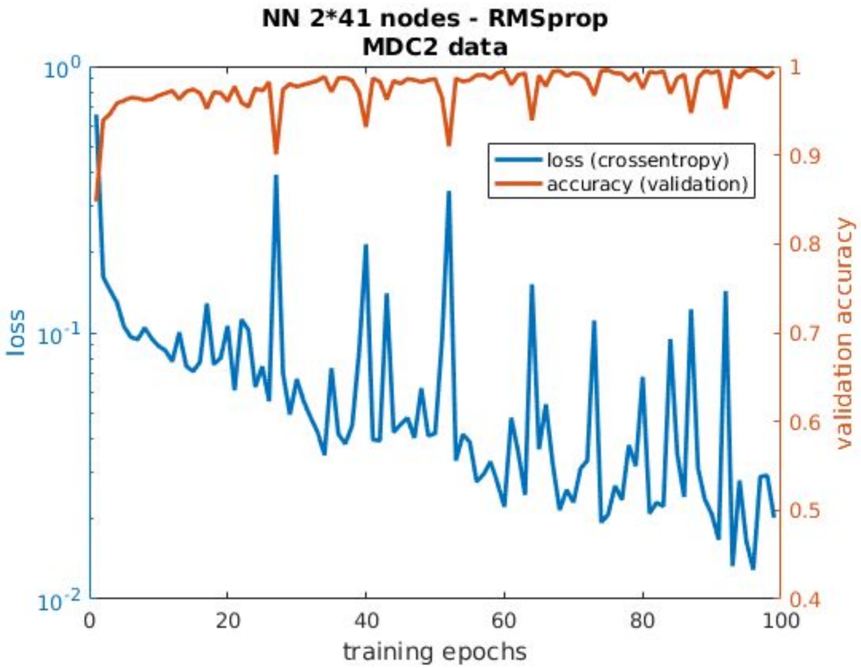
Trained and tested with MDC2 data:

MCtruth labels: (s1, s2, se)

Input: 16 modified parameters

Output: prob. vector **[s1 s2 se other]**

- 7.6M pulses total
- 10% used for validation
- Trained using batches of 10000 samples
- **~99.4% accuracy**
  - These pulses are more realistic
  - The NN efficiency sits on top of HADES efficiency!!

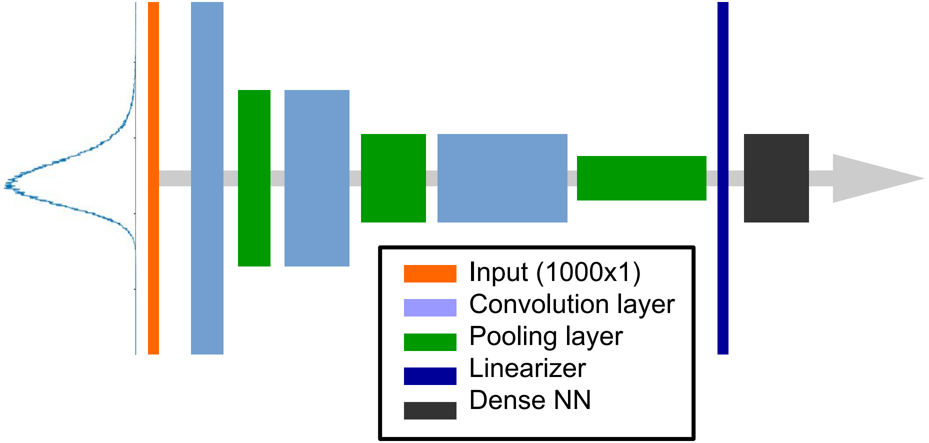


Convolution NNs

Looking into waveforms directly

Simple architecture:

1. Three pairs of convolution/pooling layers
   Generate 32 feature maps
2. Linearizer to shape the NN input
3. Dense 1500 layer NN for classification

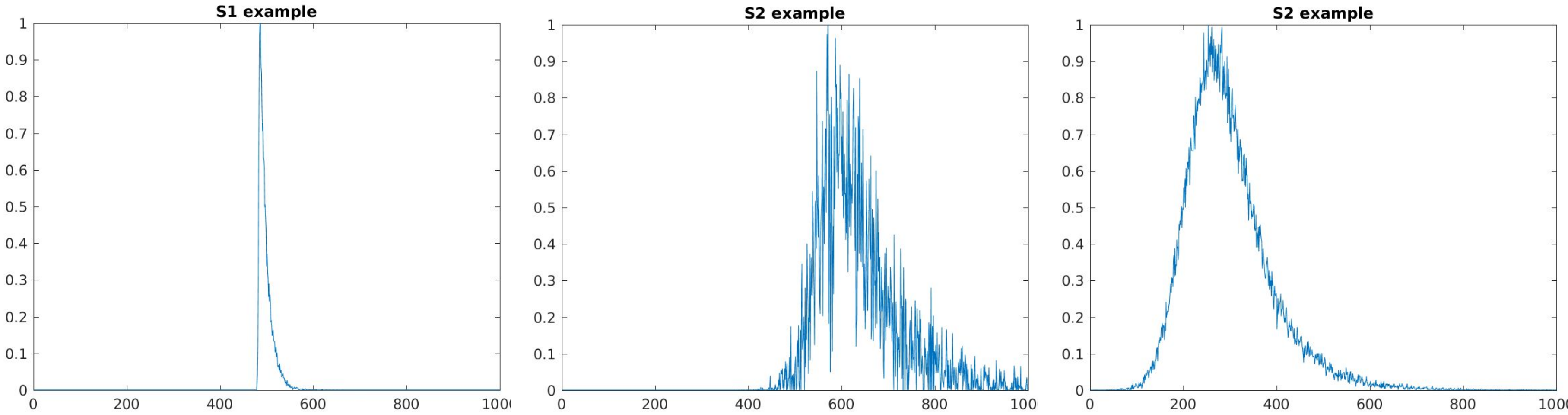



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 1, 1000, 8)        80
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 1, 1000, 8)        584
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 1, 250, 8)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 250, 16)        1168
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 1, 250, 16)        2320
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 62, 16)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 1, 62, 32)         4640
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 1, 62, 32)         9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 15, 32)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 480)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1500)              721500
_________________________________________________________________
dropout_1 (Dropout)          (None, 1500)              0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 4503
=================================================================
Total params: 744,043
Trainable params: 744,043
Non-trainable params: 0
```

Classification CNN using Keras - Input

Directly read a summed POD and determining the class of the pulse using a CNN

- Generated **20k synthetic s1** and **s2 pulses** for training
- Tried to **maintain pulse features** and have large **pulse diversity**
- Also included pulse pileup (S1+S2) to test the response



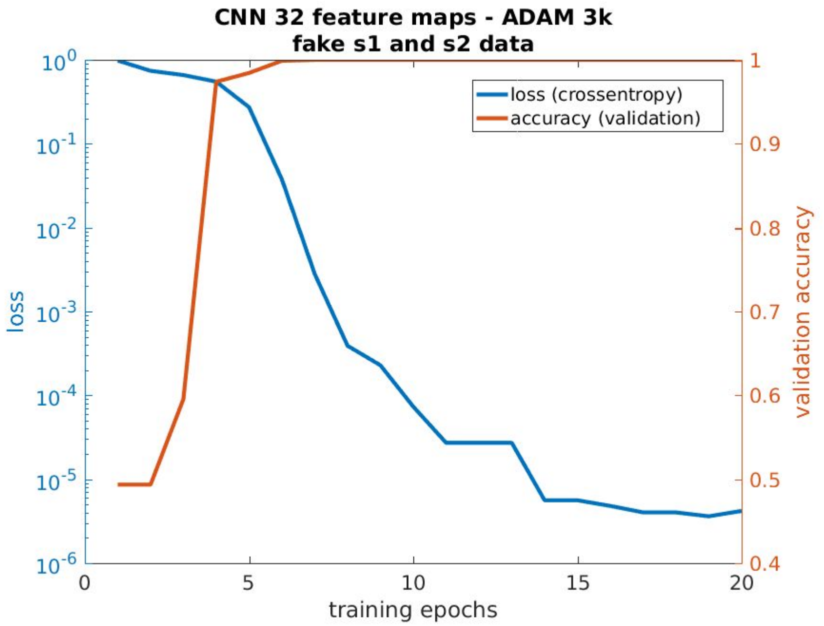
Classification CNN using Keras - results

Some results:

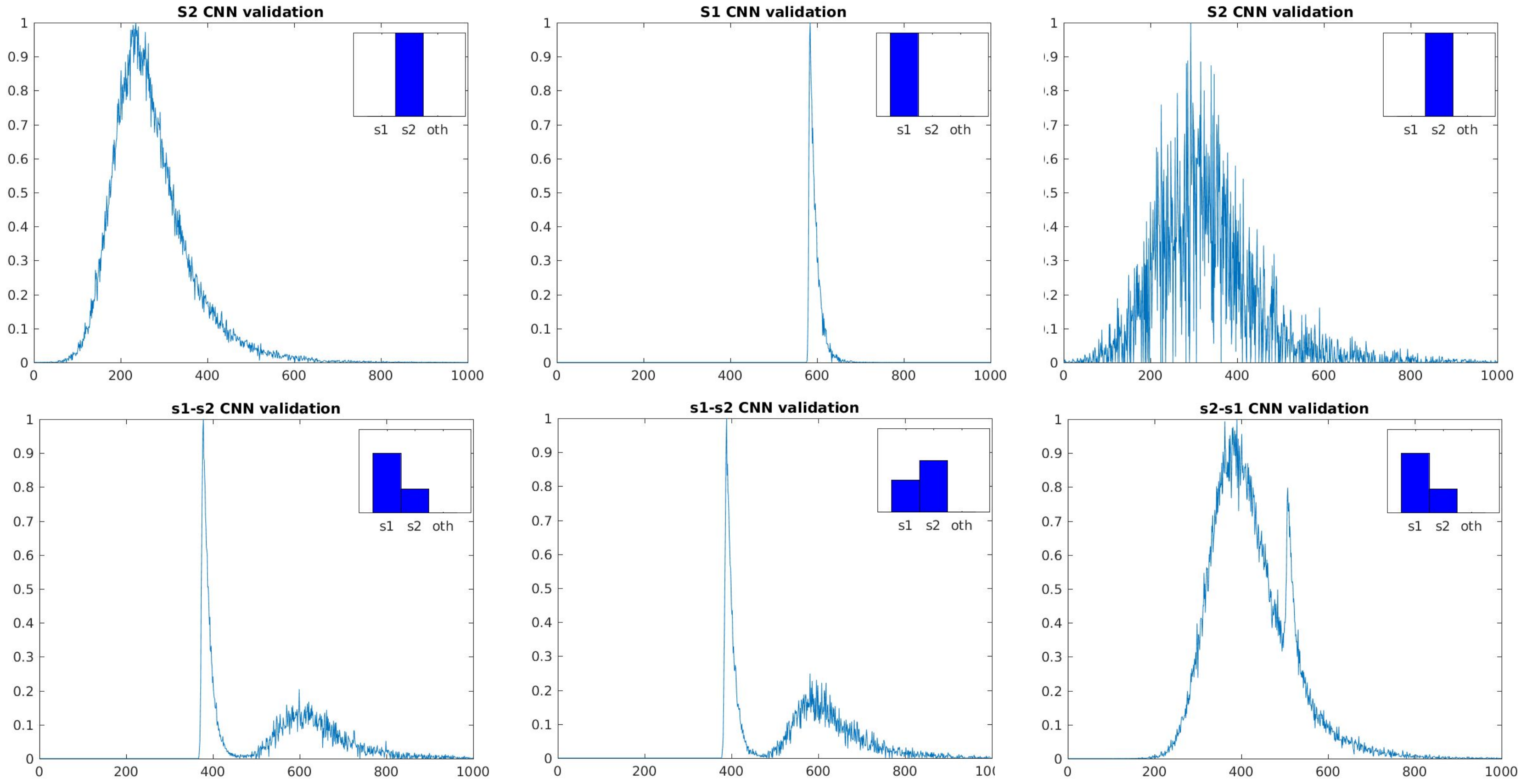
- **Data: 20k s1 and s2 waveforms**
  - 10% used for validation
- **100% classification accuracy for this dataset (< 10 epochs)**
  - Again, small dataset of well-behaved, synthetic pulses

Next:

- Increase the dataset
- Use realistic waveforms
- Check potential for pulse multiplicity test
  - Pileup identifier
  - NDBD signal discrimination (Andrey S.)



Some interesting results with the CNN



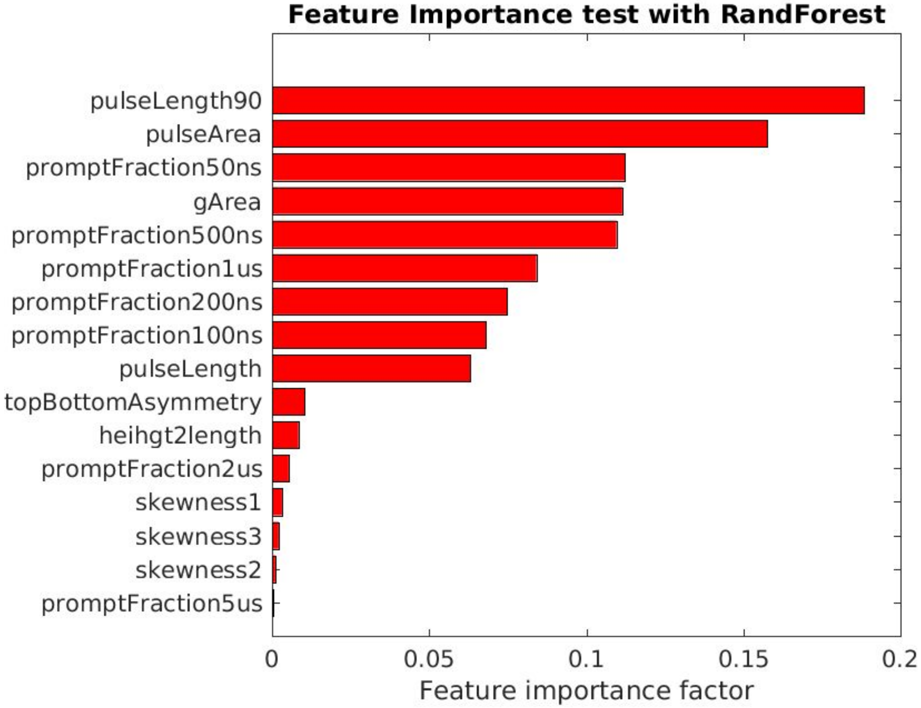
ML for Pulse Classification - 2

Random Forests
Motivations:

1. Great classification power
2. Resistant to overfitting
3. Return the classification error/variance
4. **Ability to determine the strongest discriminant features**

Tests performed with MDC2 simulated data:

- 85% of 7.6M pulses used for training
- 313 estimators in the forest
- **99.97% efficiency on top of HADES**



Summary

- **The final goal is to build a pulse Classifier that can handle realistic LZ data**
- **Simple neural networks** can achieve higher accuracy than specialized classification algorithms, given that LUX and pre-MDC1 data is well understood
- **Convolutional neural nets** can also achieve high accuracy without the need for pulse parametrization
- **Random Forests** can also achieve great results - **efficiency > 99.9% achievable**
- Testing new methods (K-means, dimensional reduction, clustering algos, etc...)
- Not having MCTruth on MDC2 data available is a drawback, but alternative paths are being explored:
    - Unsupervised learning
    - Confusion test comparisons with HADES

Thank you!

Confusion matrix for random forest and MDC2 data

Predicted Class	S1	S2	SE
Actual Class			
S1	5312	0	0
S2	0	4459	0
SE	1	1	5227

Pulse Classifier Overview

- Inputs:
  - Pulse Parameters (Physics::PulseParameters)
- Outputs:
  - Pulse Classifications (Physics::PulseClassifications)

1. Classification done at **pulse-level only**!
   a. No pulse correlations considered. Looks at each pulse object alone.
2. HG and LG channels currently using the same classification criteria.
3. Two PulseClassifier modules live within LZap
   a. COMPACT (PDFs) - disabled due to low classification efficiency
   b. HADES (cuts) - currently being used, robust

HADES will have tunable parameters for the cuts on the steering files

The COMPACT algorithm - why it failed on MDC2

Requirements of this algorithm (stated when first presented):

1. PDFs must be created using pulses with classification known a-priori
   a. Usage of simulated data to generate PDFs - requires pulse-level MCTruth
2. Only continuous pulse parameters can be used to build the PDFs
   a. Discrete parameters can't be used. However, this algorithm can (and should) be complemented with other decision criteria, exploiting all possible pulse parameters.
3. Low statistics in PDF creation decreases efficiency (see slide 5)
   a. Some features in low-sampled regions increase the error of the interpolated probability.
4. The product of probabilities is only valid if parameters are independent

The COMPACT algorithm - why it failed on MDC2

1. The main problem: due to lack of MCTruth, PDFs were build with MDC1 data
   a. MDC2 data radically different from MDC1 data - also PulseFinder changed!
   b. A lot of S1s being tagged as SEs and even more SEs being tagged as S1s
   c. A lot of good pulses were being classified as others (PDF incompatibility)
2. Other problems:
   a. SPEs not being handled correctly due to coincidence bug on PulseParametrizer
   b. Severe over-splitting of the tails of large S2s, which were tagged as S2s (SS tree unusable)

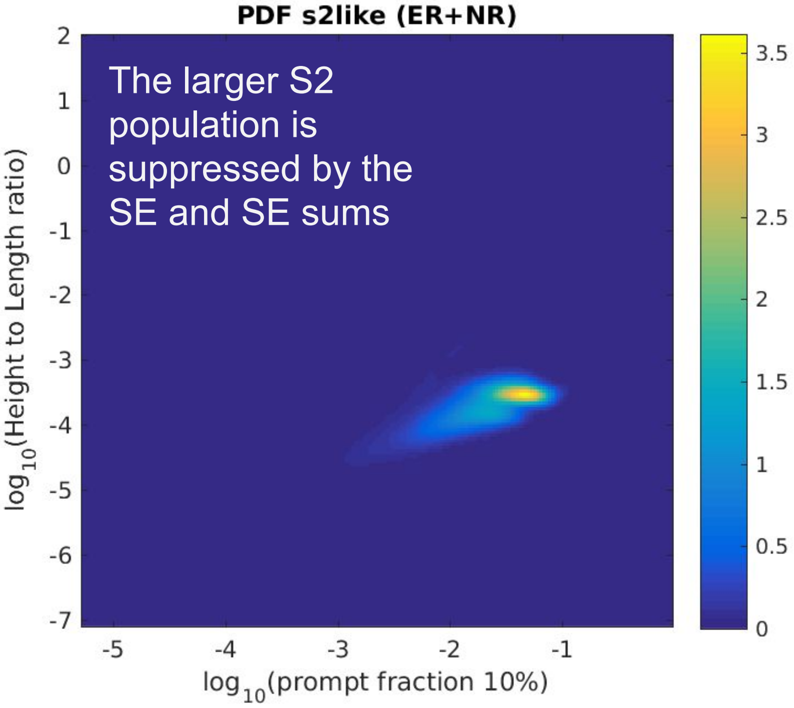
Some patches done to the module (and later withdrawn):

- Build new PDFs with MDC2 data
- SPE cut on area -> damaged efficiency for small S1s
- New class “OtherS2” to include tails -> damaged efficiency for small S2s

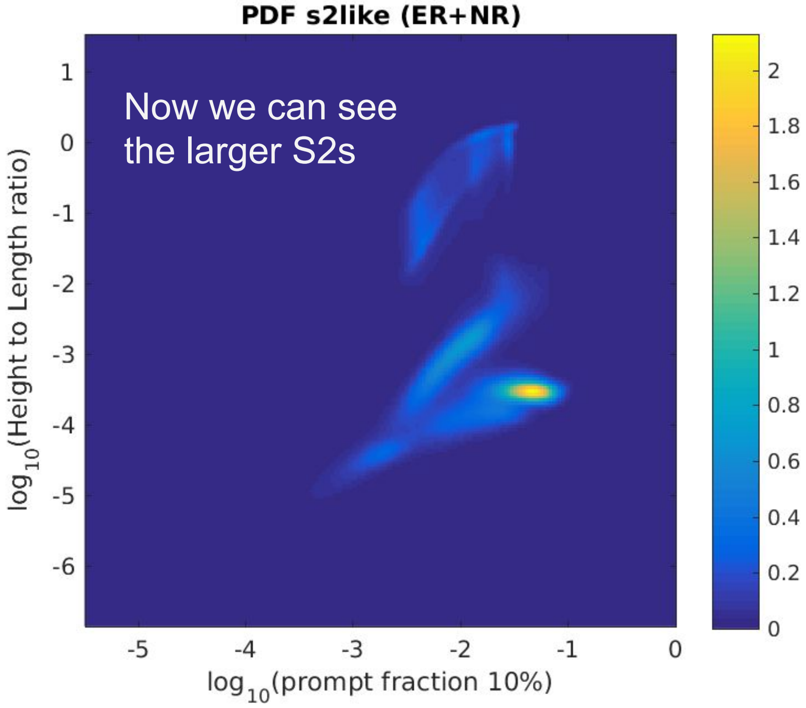
Suppression of S2 features in PDFs

Higher abundance of e-trains and tails cases S2 features to get severely suppressed, normalization required!

Before normalization



After normalization



Pulse Classifier HADES

Heuristics Algorithm for Discrimination of Event Substructures (HADES)
Born out of the necessity to have reliable classifications for building PDFs with MDC2 data

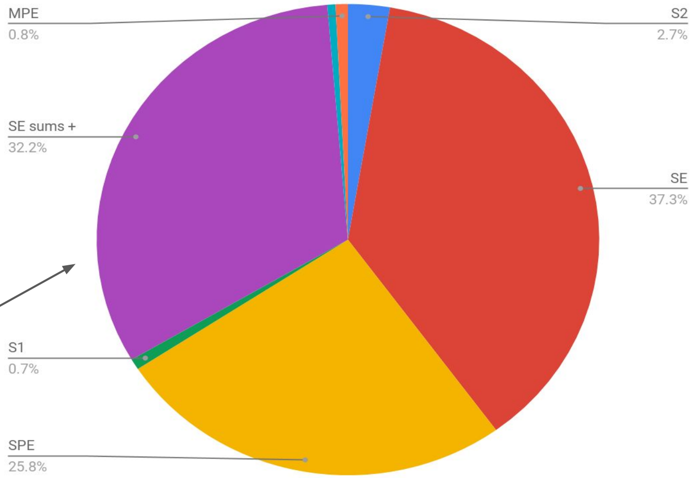
- Cut-based algorithm (similar to the one LUX had)
  - Purely heuristic and done by eye, the very opposite of COMPACT
- Currently implemented on LZap and tested
  - Robust, easy to understand and highly tunable
  - Early results look good - easily outperforms COMPACT
- Cuts are fairly basic and can be improved (WIP)
  - S1s still permeated SE phase-space - easily removed with pulse length cut

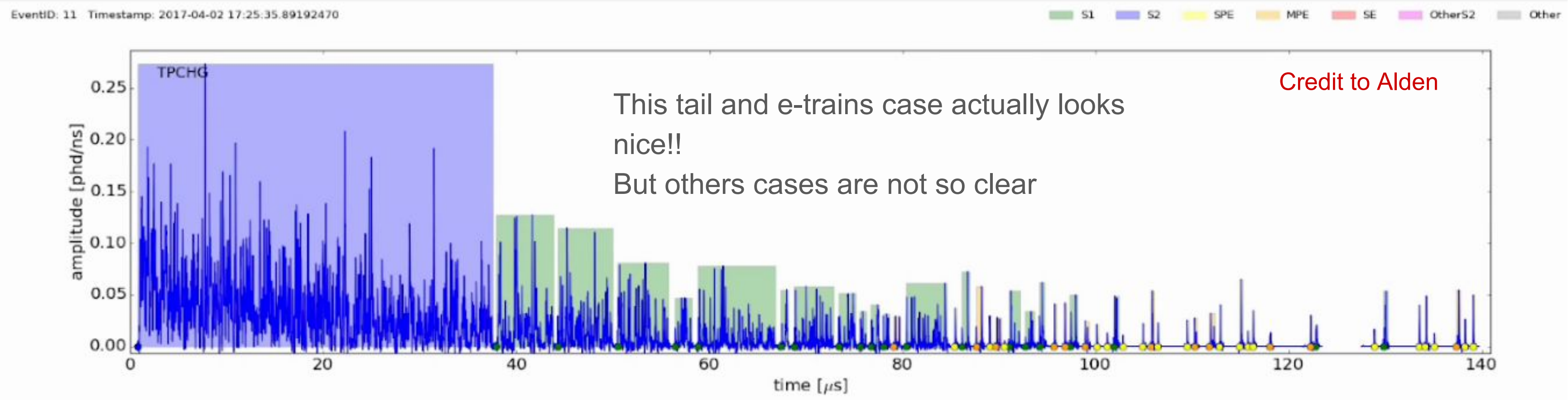
MDC2 Data Quality

Do we want the Pulse Finder to split S2 tails so strongly in the future?

This includes SE sums, tails and very small S2s.
They are classified as S2s since 3.8.0

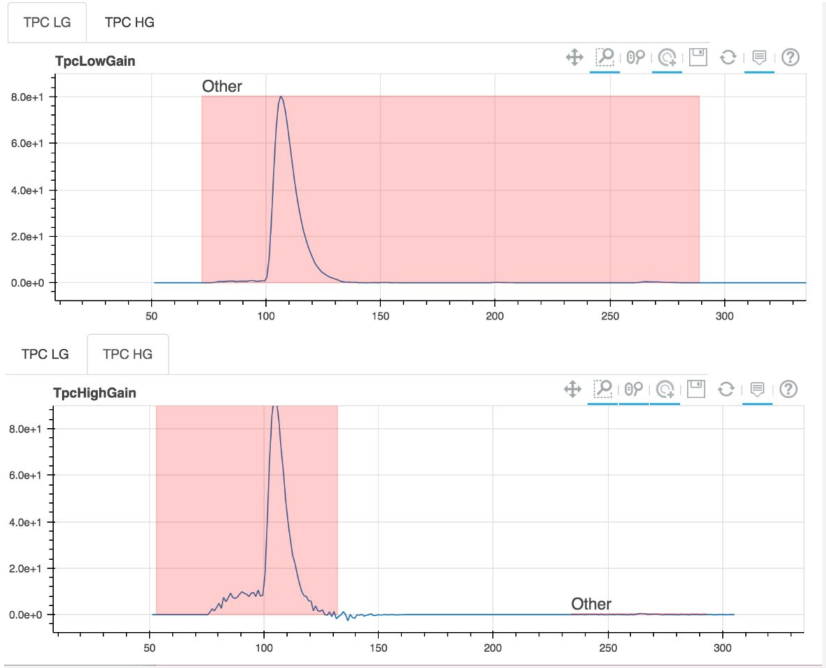
BG data classifications with HADES

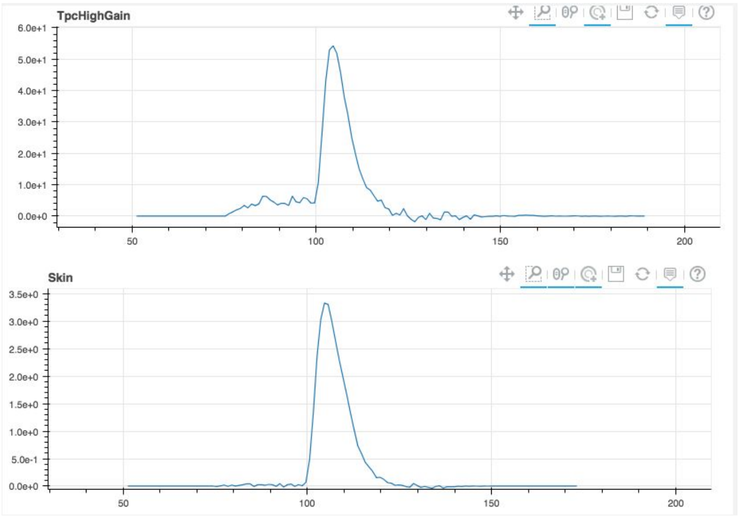




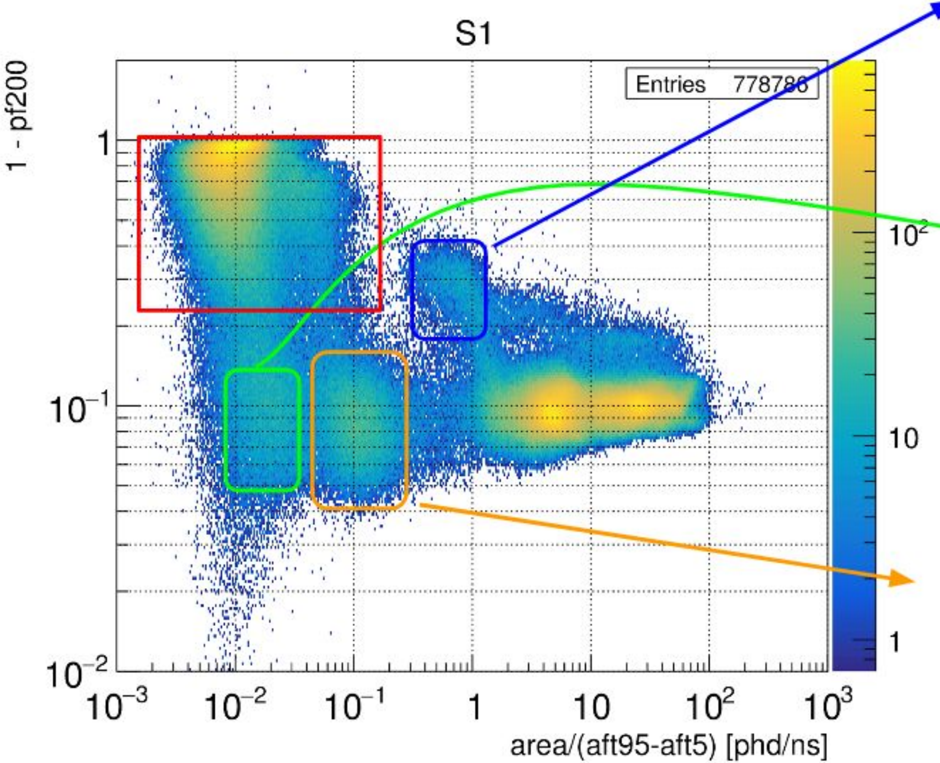
This tail and e-trains case actually looks nice!!
But others cases are not so clear

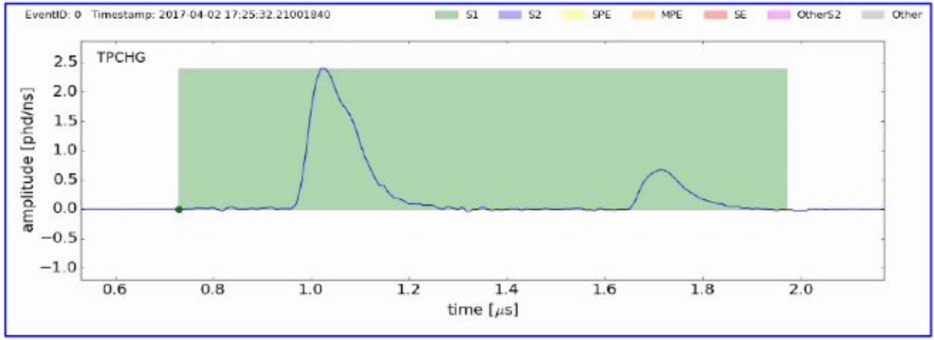
Credit to Alden

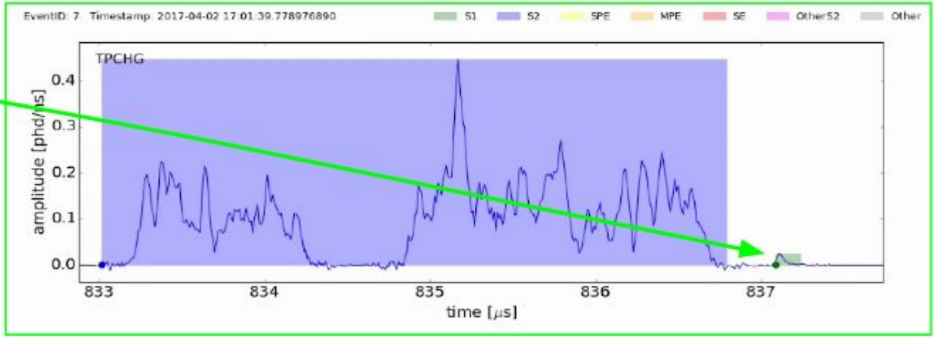

TPC LG
TPC HG
TpcLowGain
Other
8.0e+1
6.0e+1
4.0e+1
2.0e+1
0.0e+0
50
100
150
200
250
300
TPC LG
TPC HG
TpcHighGain
8.0e+1
6.0e+1
4.0e+1
2.0e+1
0.0e+0
Other
50
100
150
200
250
300

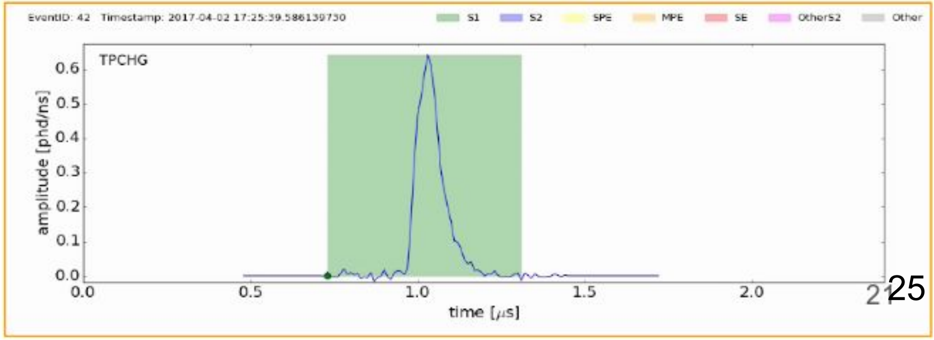

TpcHighGain
6.0e+1
5.0e+1
4.0e+1
3.0e+1
2.0e+1
1.0e+1
0.0e+0
50
100
150
200
Skin
3.5e+0
3.0e+0
2.5e+0
2.0e+0
1.5e+0
1.0e+0
5.0e-1
0.0e+0
50
100
150
200

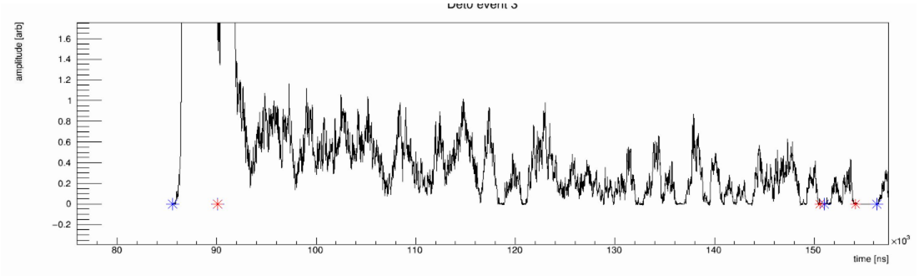
What are the other blobs?

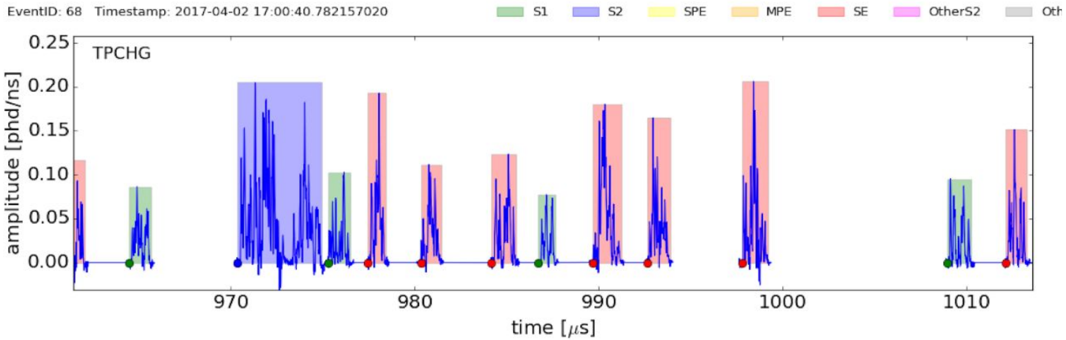


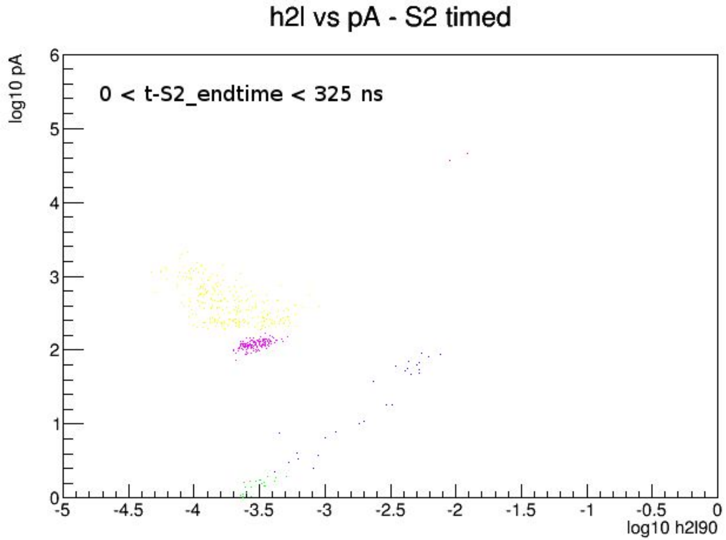







amplitude [arb]
time [ns]


EventID: 68 Timestamp: 2017-04-02 17:00:40.782157020
S1
S2
SPE
MPE
SE
OtherS2
Oth
TPCHG
amplitude [phd/ns]
time [μs]


h2l vs pA - S2 timed
0 < t-S2_endtime < 325 ns
log10 pA
log10 h2l90