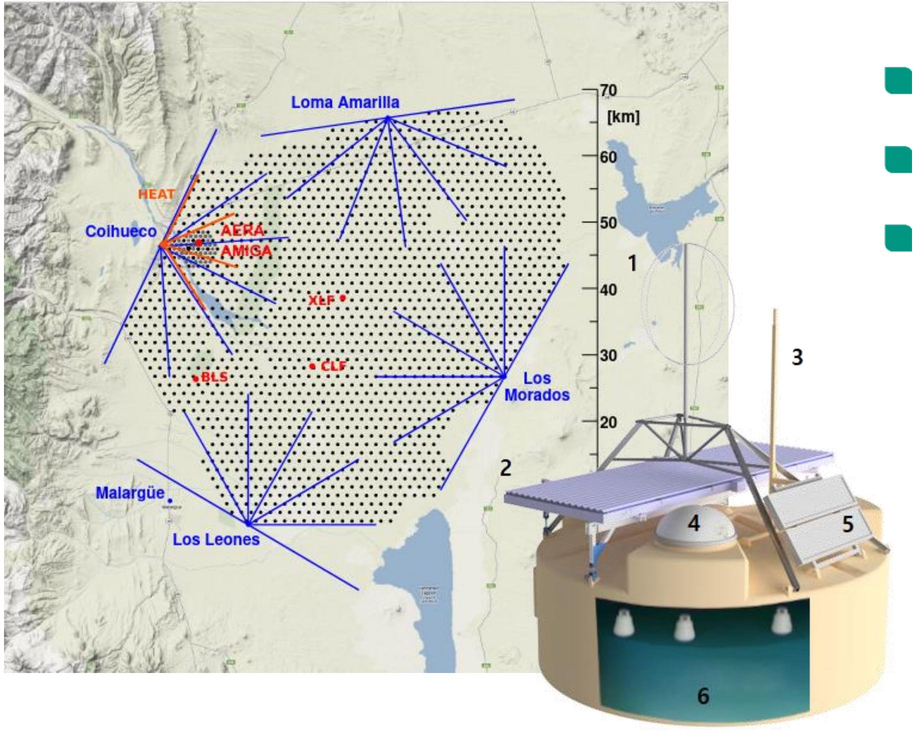
Potential of neural network triggers for the Water-Cherenkov detector array of the Pierre Auger Observatory

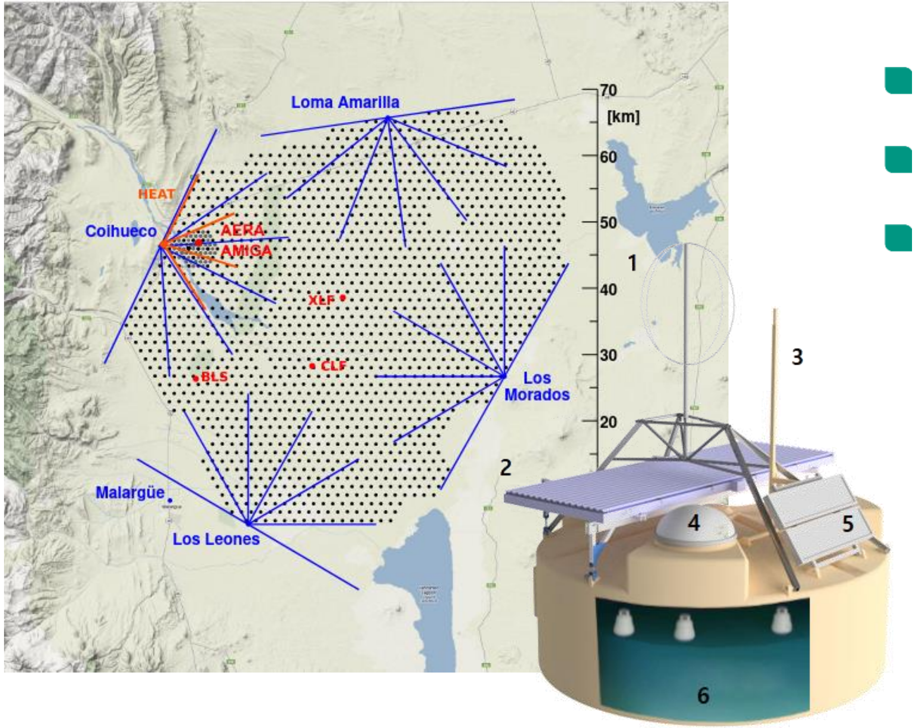
Paul Filip - High Energy Universe seminar 01.06.23

SD Array / trigger hierarchy / WCD time traces



- **Around ~1600 stations**
- **Triangular** $1.5\ \mathrm{km}$ **grid spacing**
- **Ongoing upgrade from UB → UUB**
  - **3 Water-Cherenkov detectors (WCD)**
  - 1 Surface scintillator detector (SSD)
  - 1 Radio antenna

SD Array / trigger hierarchy / WCD time traces

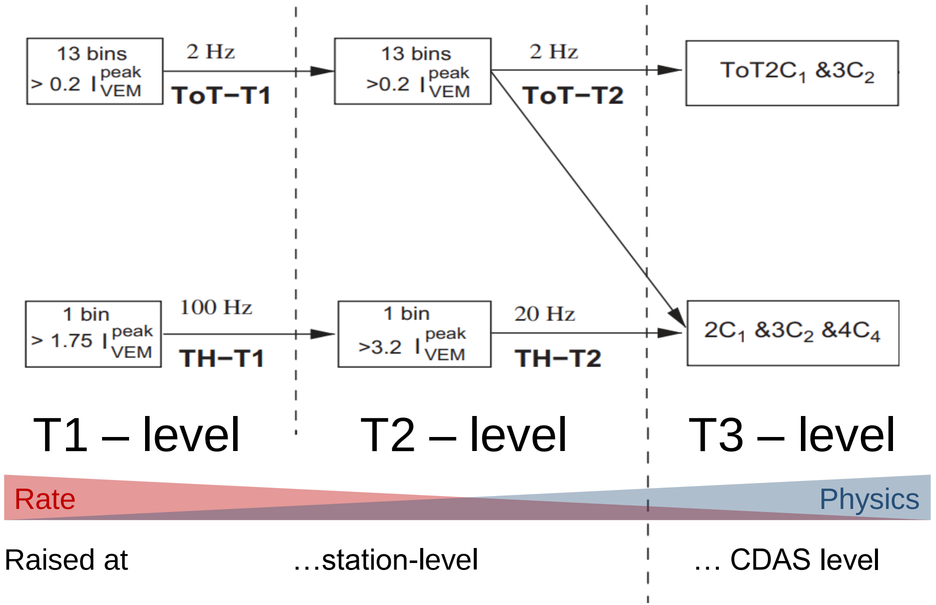


- **Around ~1600 stations**
- **Triangular $1.5\,\mathrm{km}$ grid spacing**
- **Ongoing upgrade from UB → UUB**
  - **3 Water-Cherenkov detectors (WCD)**
  - 1 Surface scintillator detector (SSD)
  - 1 Radio antenna

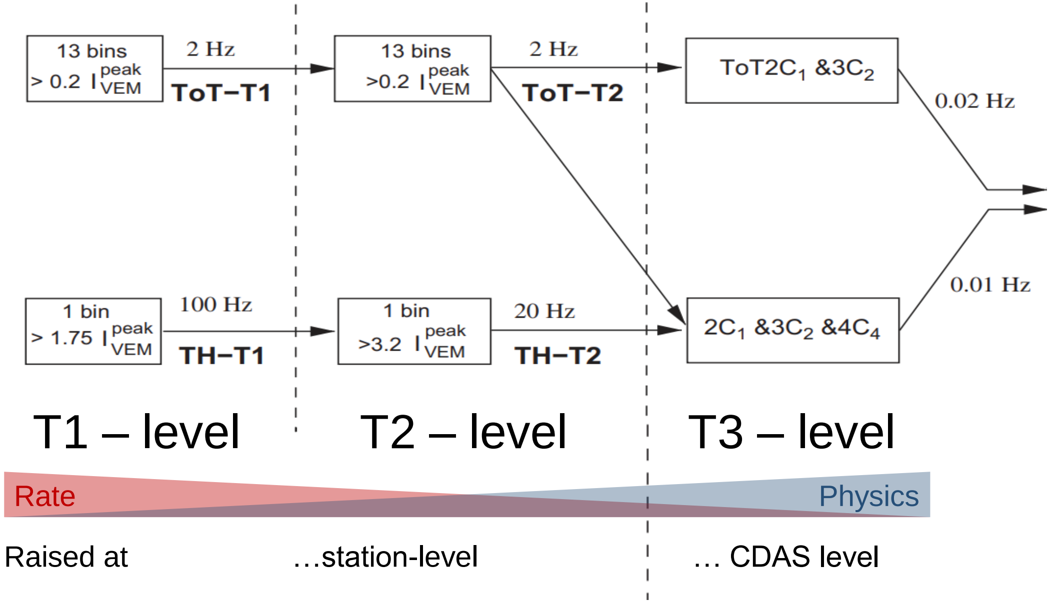
Too comput. expensive to read all measured data at all times!

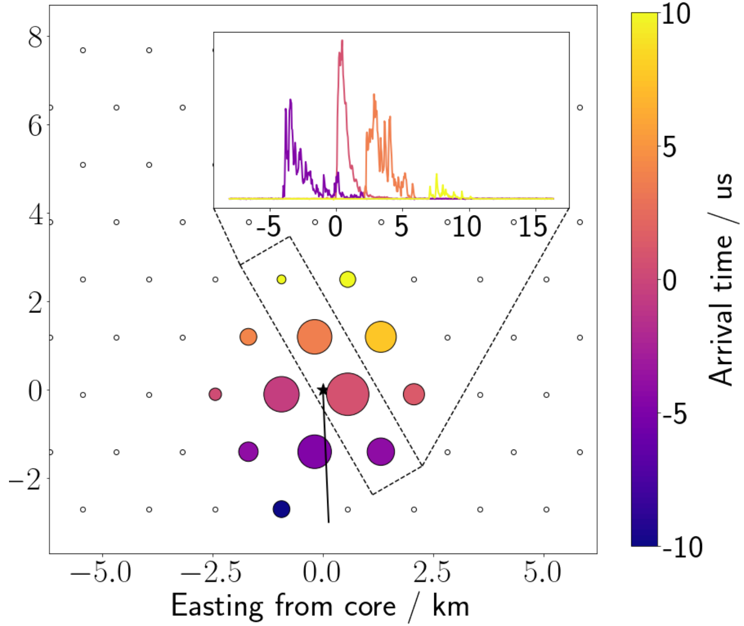
→ Implement trigger hierarchy

SD Array / **trigger hierarchy** / WCD time traces



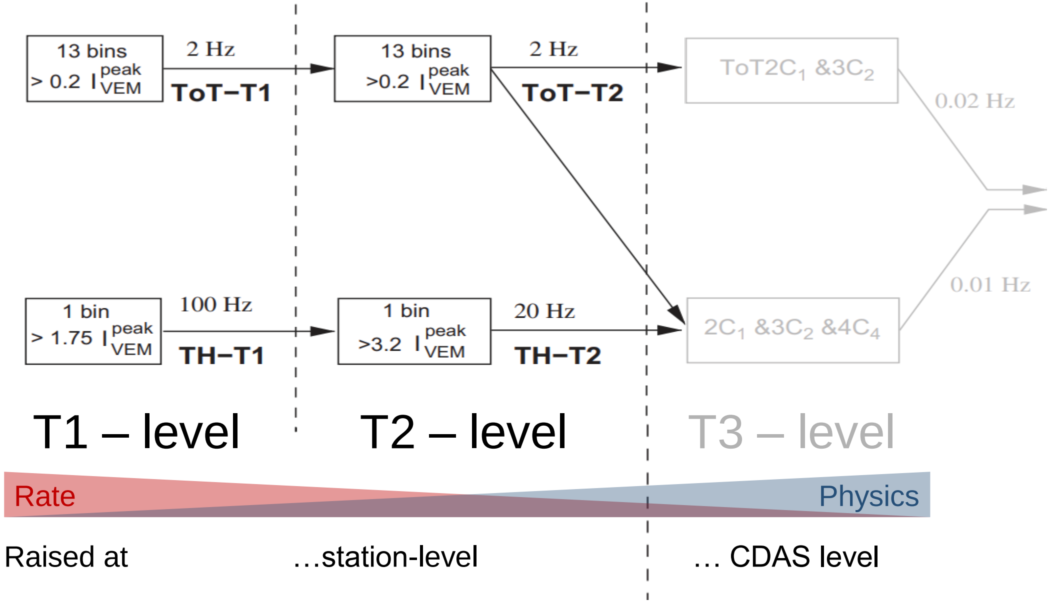
SD Array / **trigger hierarchy** / WCD time traces

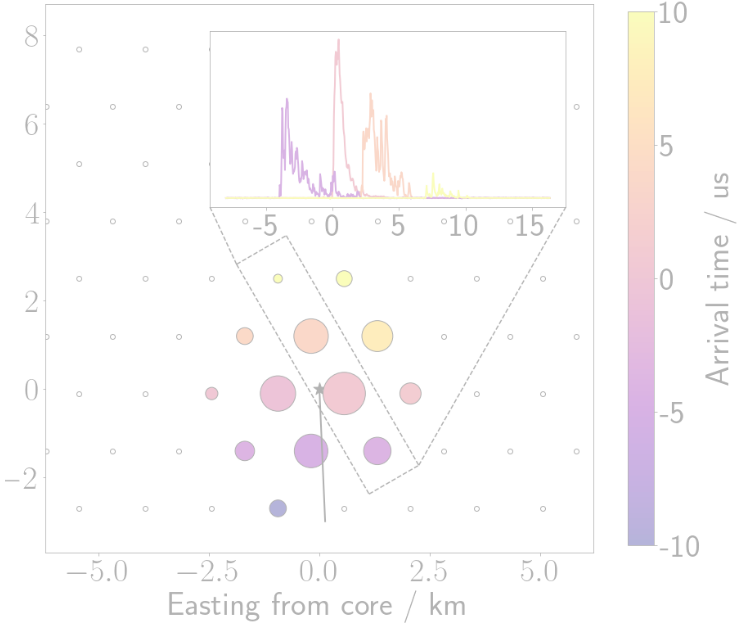




SD Array / **trigger hierarchy** / WCD time traces

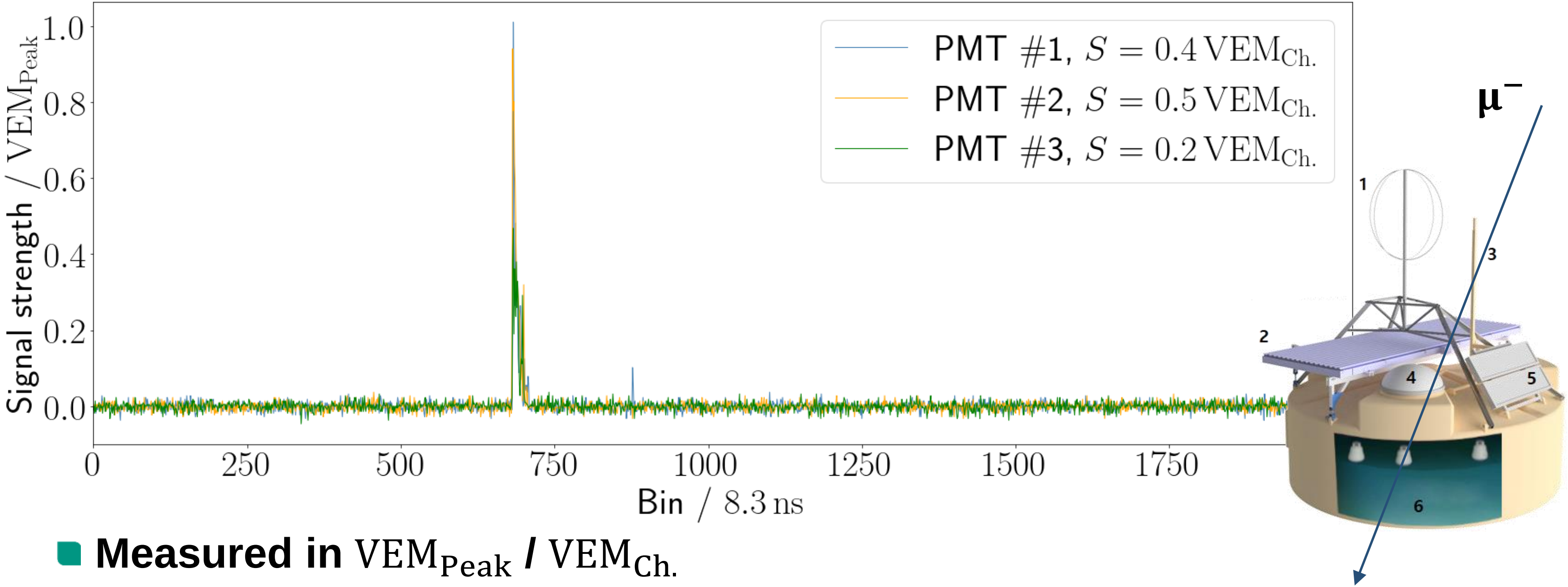
Karlsruhe Institute of Technology





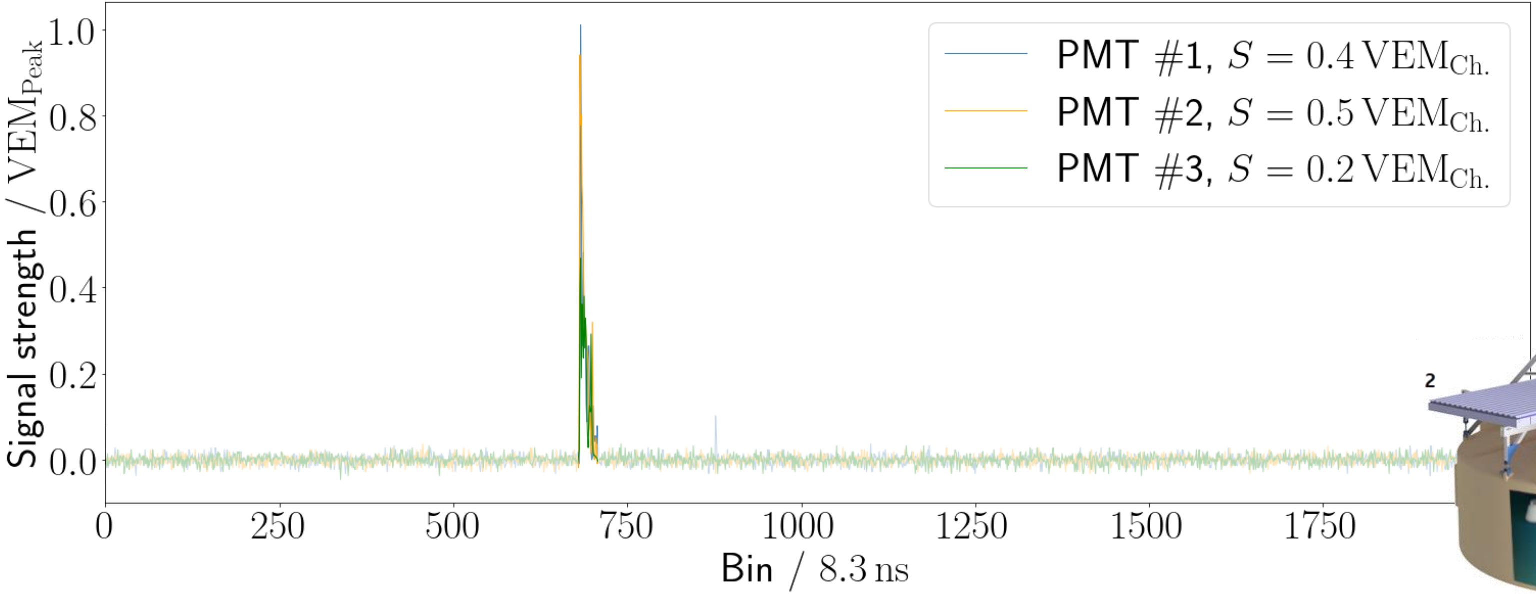
- **Threshold trigger (Th)**
- **Time over threshold (ToT) & ToT-like triggers**

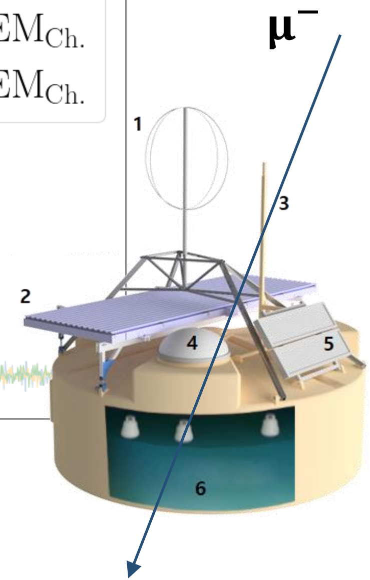
SD Array / trigger hierarchy / WCD time traces



- **Measured in** $\mathrm{VEM}_{\mathrm{Peak}}$ **/** $\mathrm{VEM}_{\mathrm{Ch.}}$
- $120\ \mathrm{MHz}$ **sampling frequency**

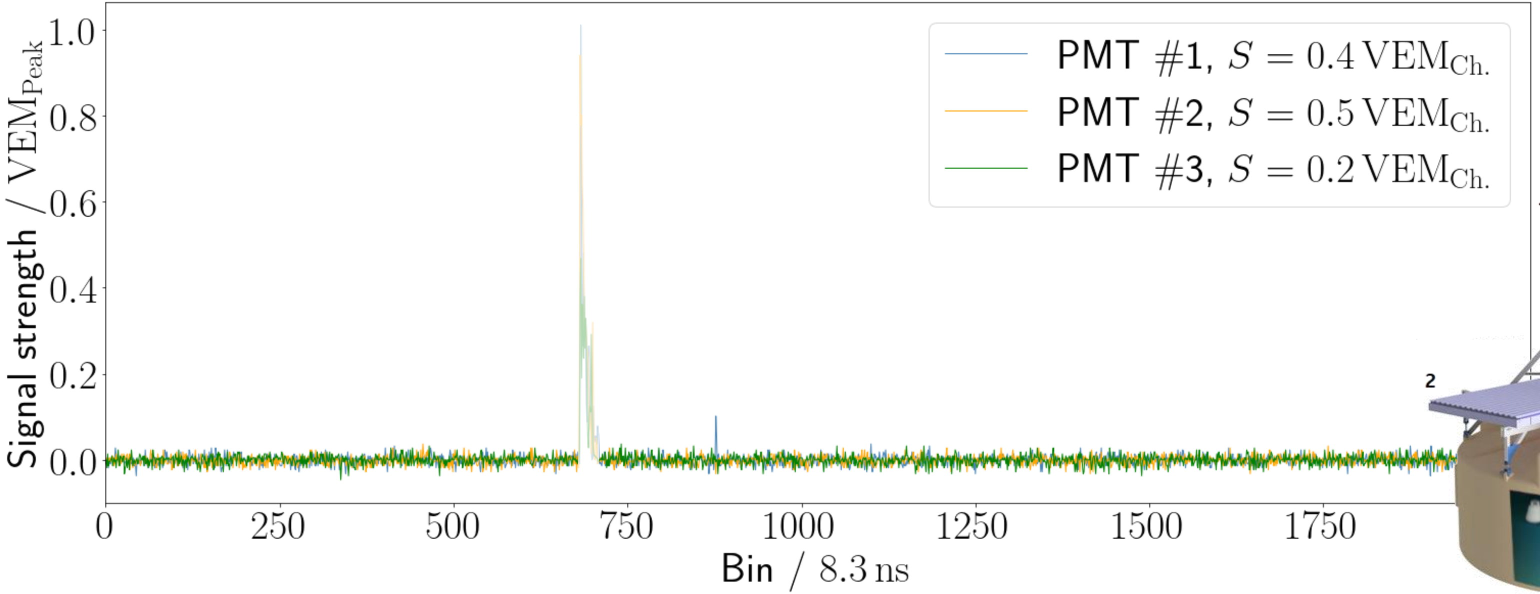
SD Array / trigger hierarchy / WCD time traces

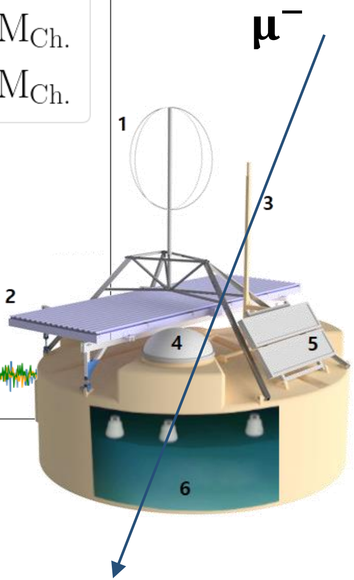




- **QGSJET-II.04 protons with** $16 \leq \log_{10}(\mathrm{E} / \mathrm{eV}) < 19.5$
- **Background electronic noise measured in field**

SD Array / trigger hierarchy / WCD time traces

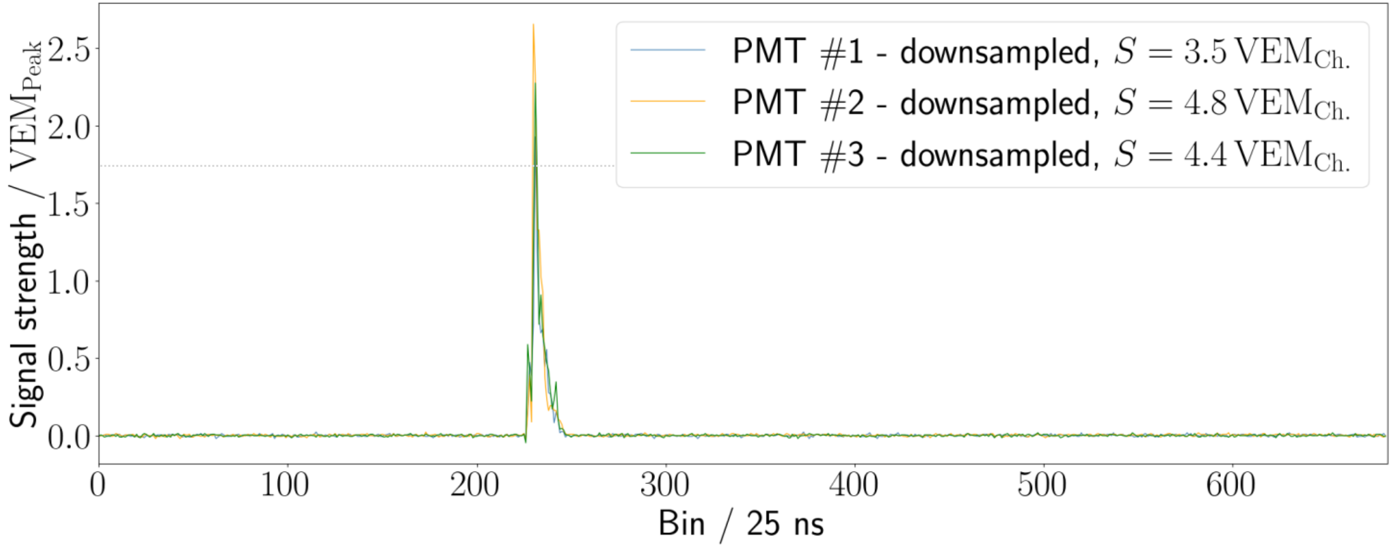




- **QGSJET-II.04 protons with** $16 \leq \log_{10}(\mathrm{E}\,/\,\mathrm{eV}) < 19.5$
- **Background electronic noise measured in field**

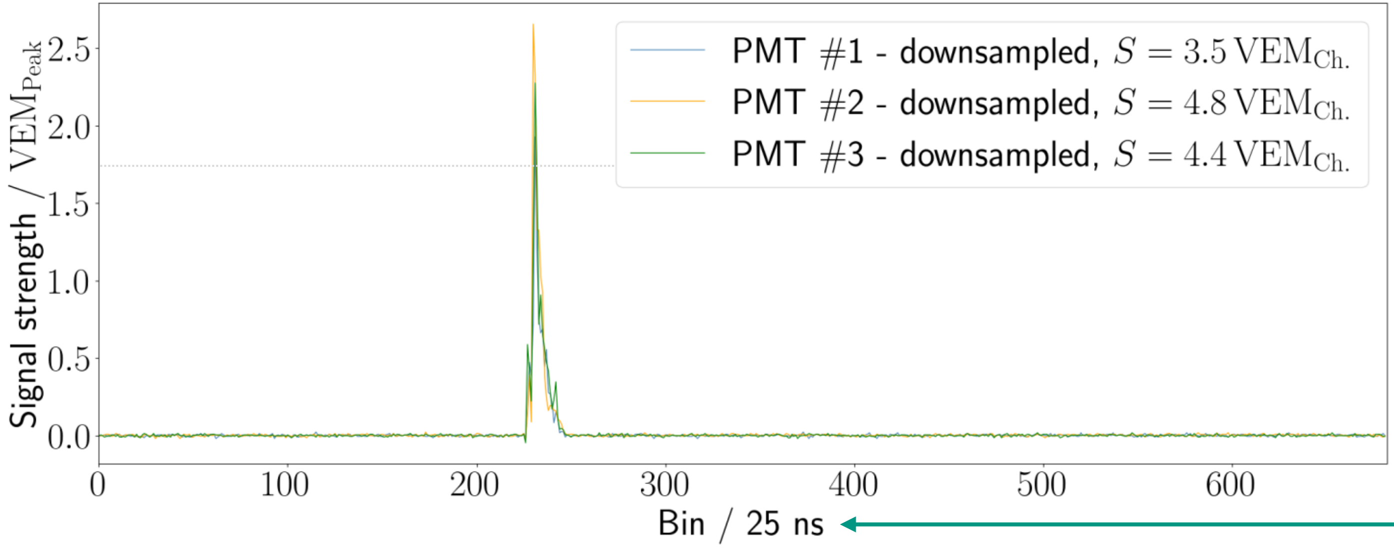
Current station-level trigger algorithms

- **Threshold trigger (Th)**
  - **PMTs register signal** $S \geq 3.2\ \mathrm{VEM_{Peak}}$ $(1.75\ \mathrm{VEM_{Peak}}$ **for T1**)
  - **Threshold must be exceeded simultaneously for all PMTs**



Current station-level trigger algorithms

- **Threshold trigger (Th)**
  - **PMTs register signal** $S \geq 3.2\ \mathrm{VEM_{Peak}}$ $(1.75\ \mathrm{VEM_{Peak}}$ **for T1**)
  - **Threshold must be exceeded simultaneously for all PMTs**

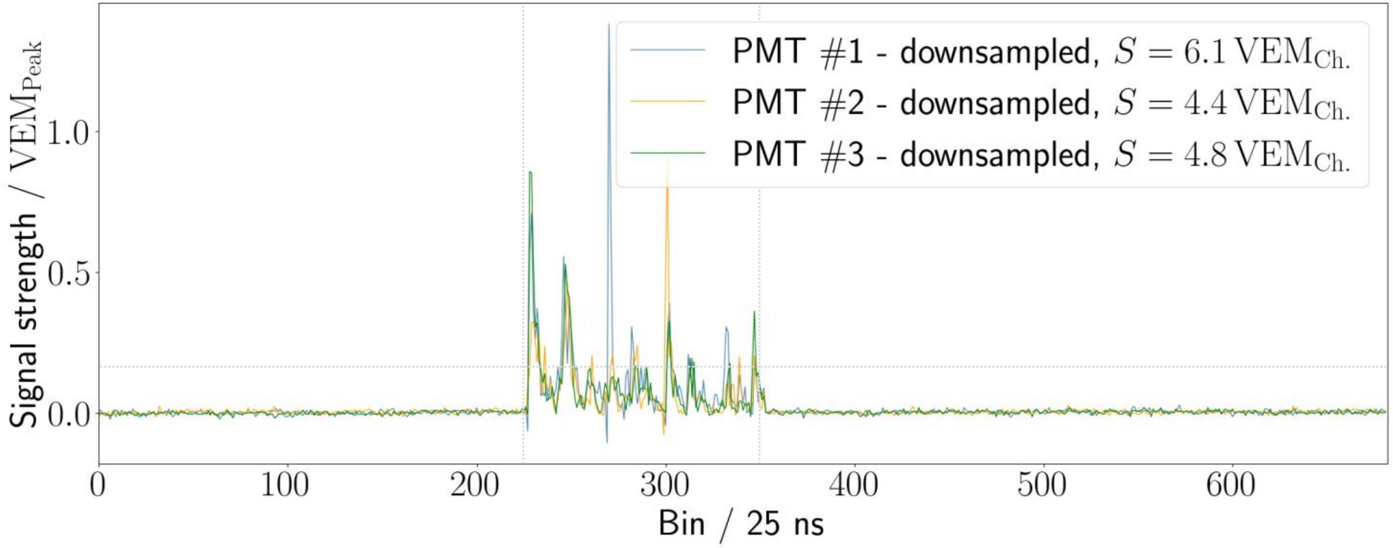


Different binning!

Trigger thresholds defined for **UB traces**; measured data must be filtered and downsampled

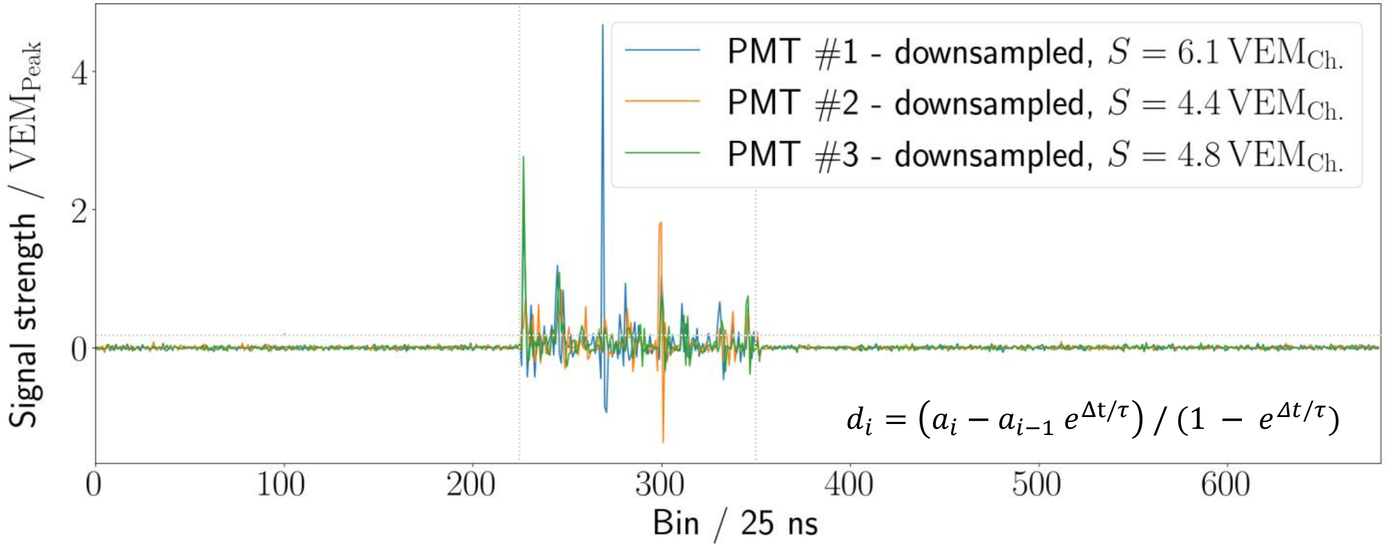
Current station-level trigger algorithms

- **Time over threshold (ToT)**
  - **More than 12 bins with $S \geq 0.2\ \mathrm{VEM_{Peak}}$ in any 120 bin window**
  - **At least 2 out of 3 PMTs meet above critera**



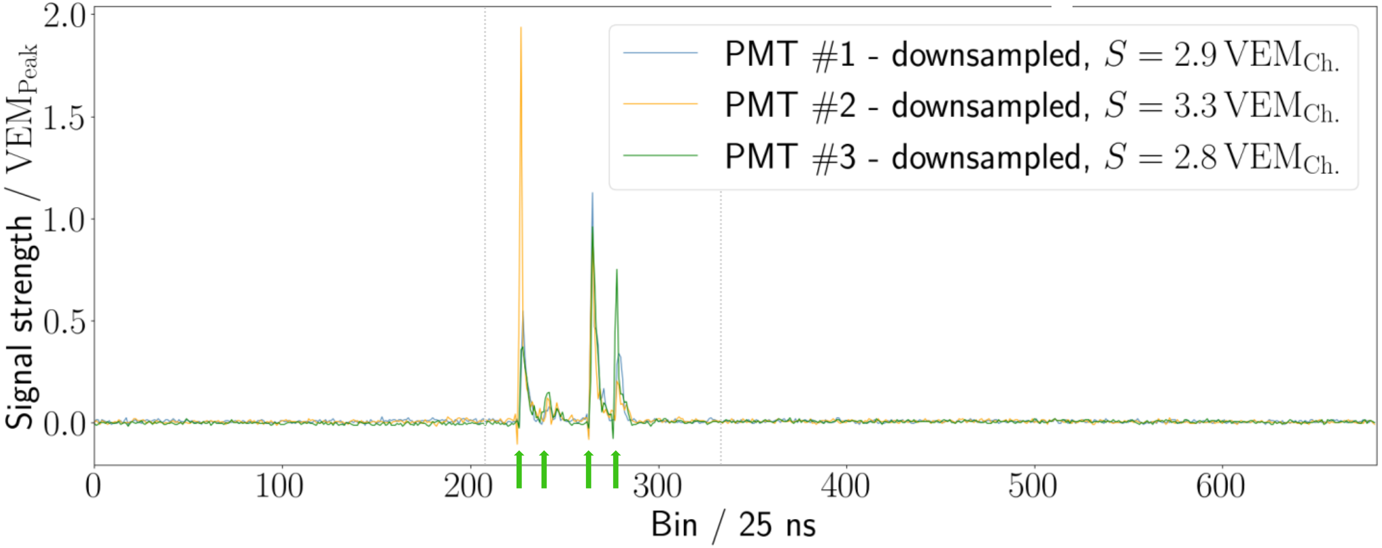
Current station-level trigger algorithms

- **ToT deconvoluted (ToTd)**
  - **Deconvolute input data stream with exponential factor**
  - **Feed deconvoluted trace into ToT algorithm**

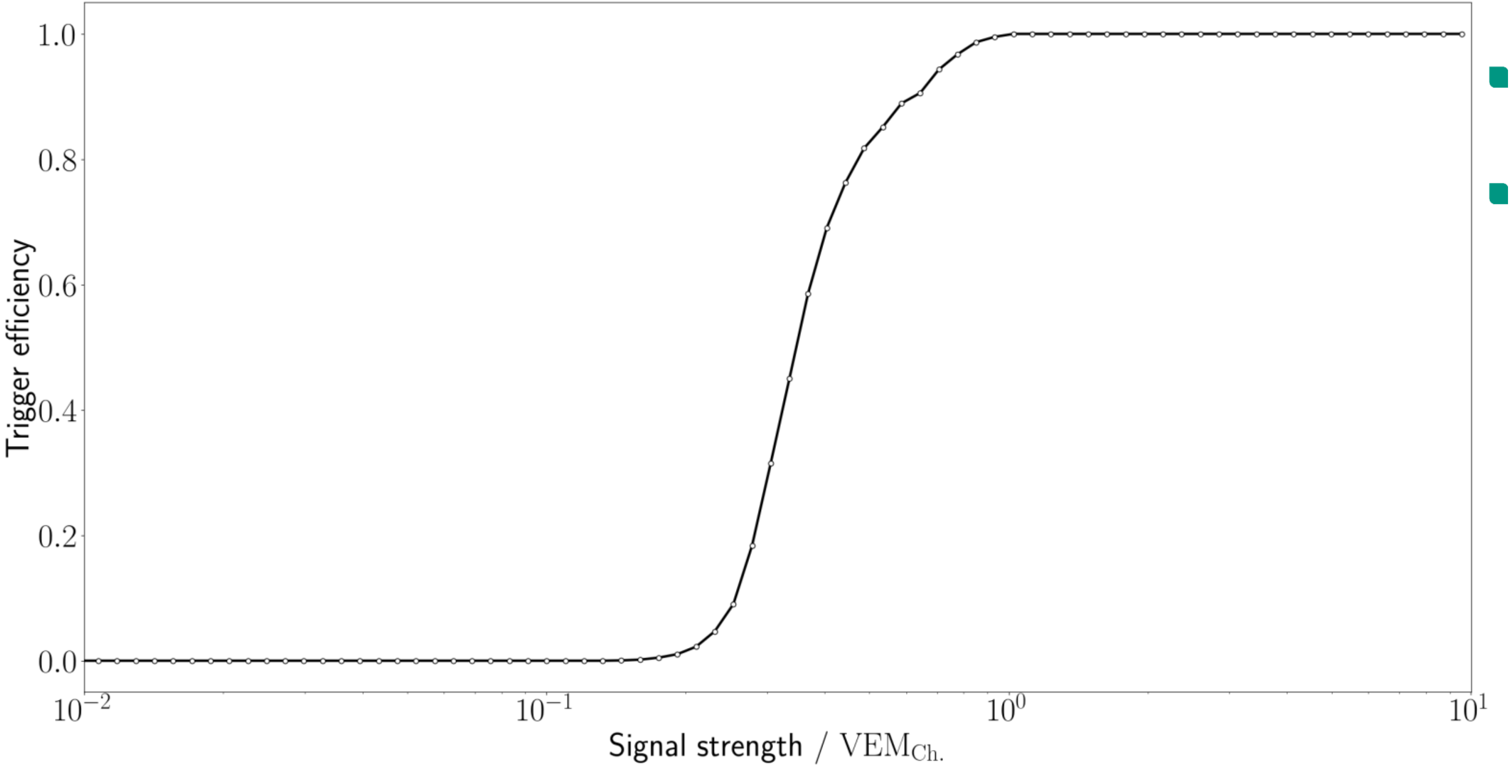


Current station-level trigger algorithms

- **Multiplicity of positive steps (MoPS)**
  - **Count number of rising flanks within 120 bin window**
  - **At least 2 PMTs have 4 (or more) rising flanks**

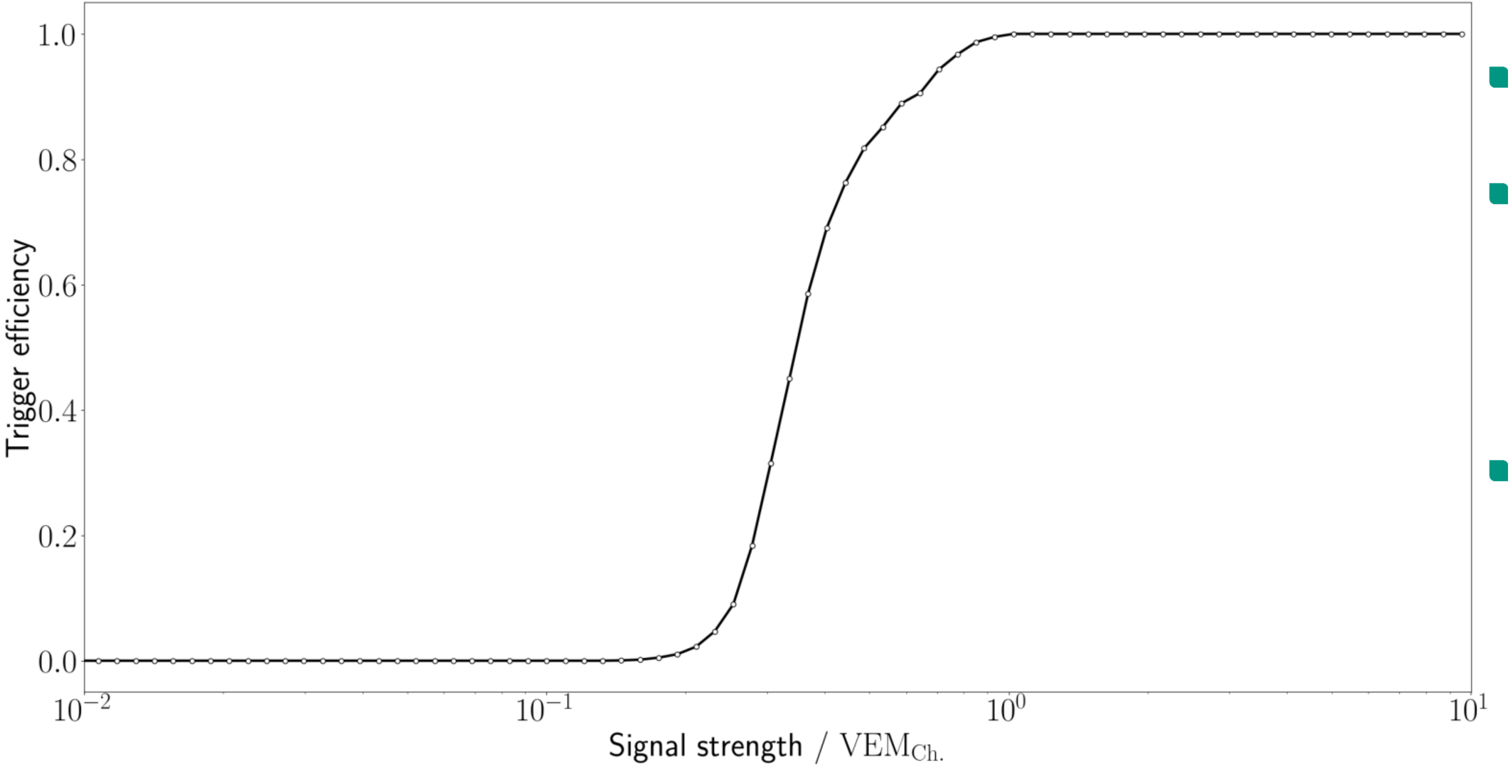


Trigger performance



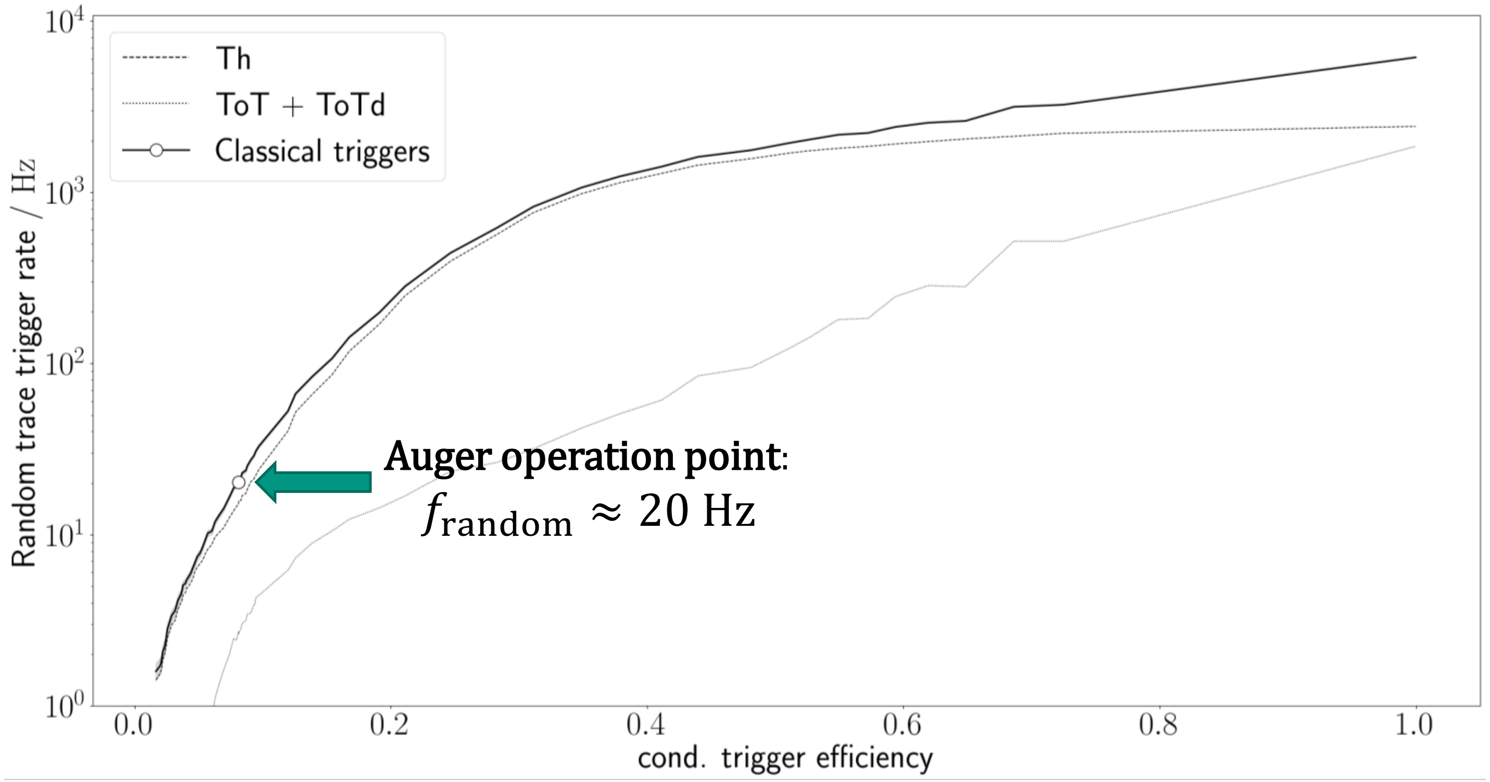
- **100% eff.** $\approx 1\ \mathrm{VEM}_{\mathrm{Ch.}}$ $\Leftrightarrow 3\ \mathrm{EeV}\ (10^{18.5}\ \mathrm{eV})$
- **Can we do better?**
  - **Photon search**
  - **Neutrino search**
  - **GZ effect**
  - **…**

Trigger performance



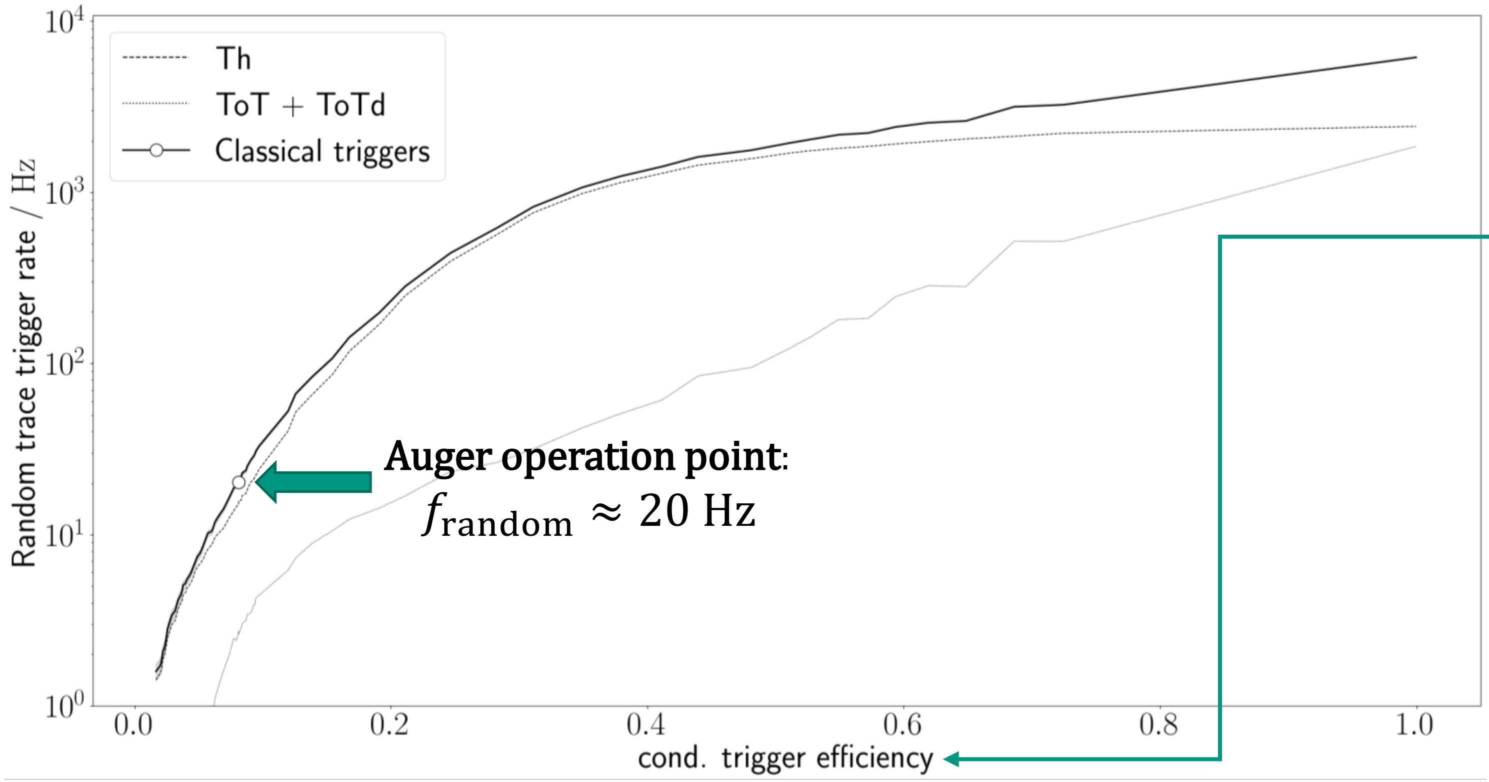
- **100% eff.** $\approx 1\ \mathrm{VEM_{Ch.}}$
  $\Leftrightarrow 3\ \mathrm{EeV}\ (10^{18.5}\ \mathrm{eV})$
- **Can we do better?**
  - **Photon search**
  - **Neutrino search**
  - **GZ effect**
  - **…**
- **Adjust thresholds**
  - **Better sensitivity**
  - **Worse specificity**
  - **What about SNR?**

Trigger performance



- Adjust thresholds
  - Better sensitivity
  - Worse specificity
  - What about SNR?
  - → gets way worse!
- Use ML triggers
  - Bayesian classifier
  - Neural networks
- Design limitation
  - Bandwidth limit
  - Performance limit
  - Storage limit

Trigger performance



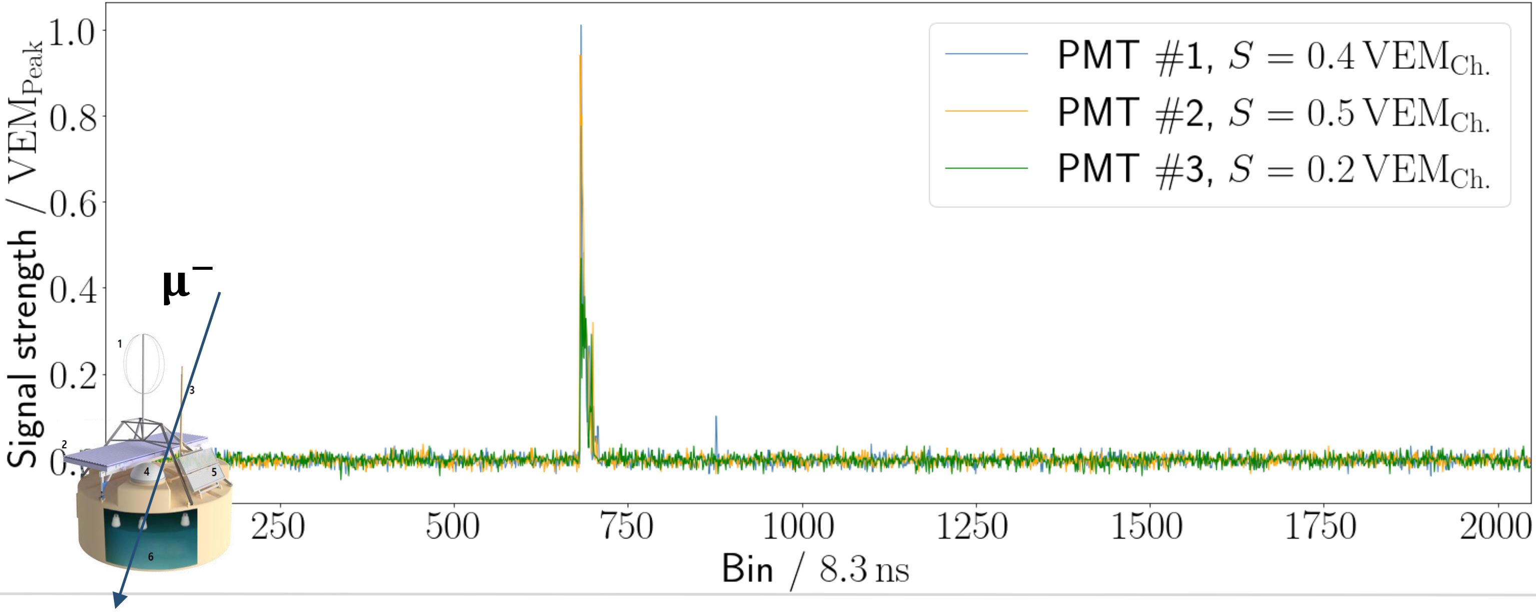
Cond. trigger efficiency

Ratio of air showers that ***are*** detected from all showers that ***can*** theoretically be detected

- **Use ML triggers**
  - **Bayesian classifier**
  - **Neural networks**
- **Design limitation**
  - **Bandwidth limit**
  - **Performance limit**
  - **Storage limit**

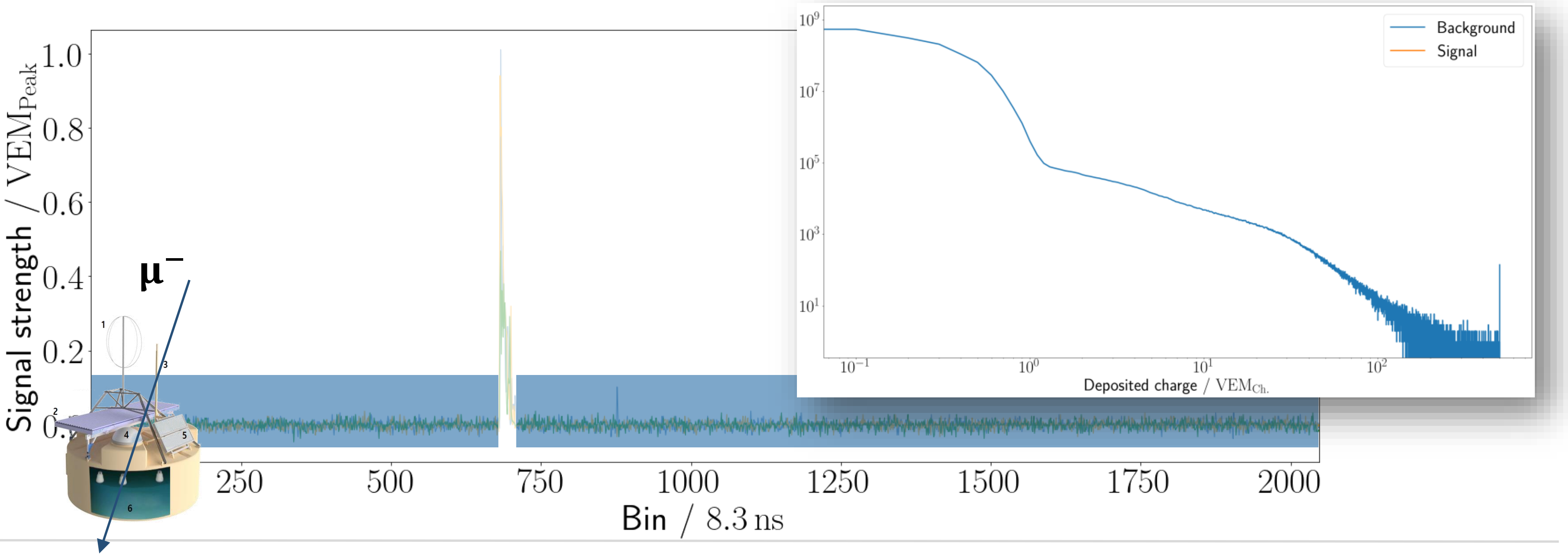
Intermezzo: Bayesian classifier

- Likelihood analysis of signal / background dataset only
- Use likelihood ratio λ_{LR} to distinguish



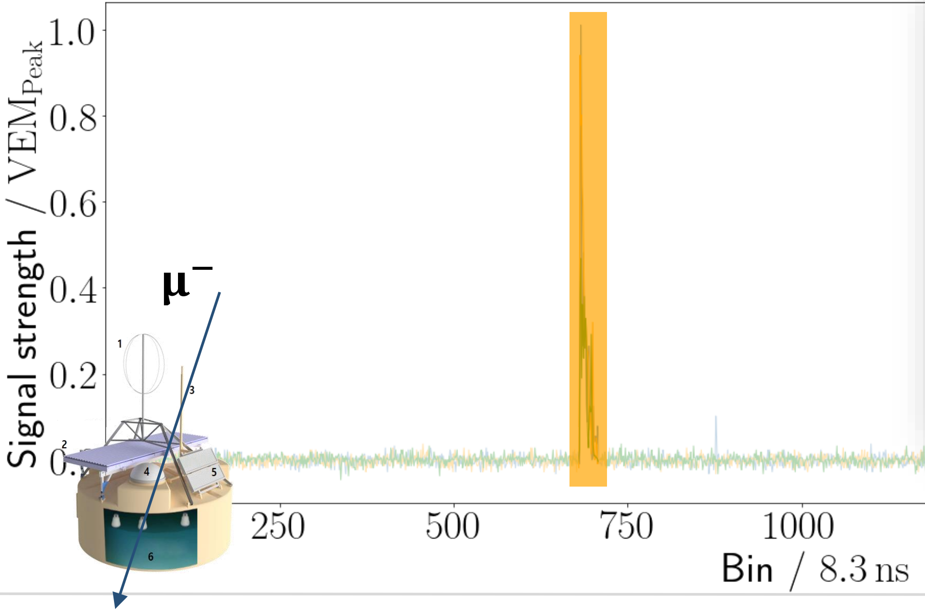
Intermezzo: Bayesian classifier

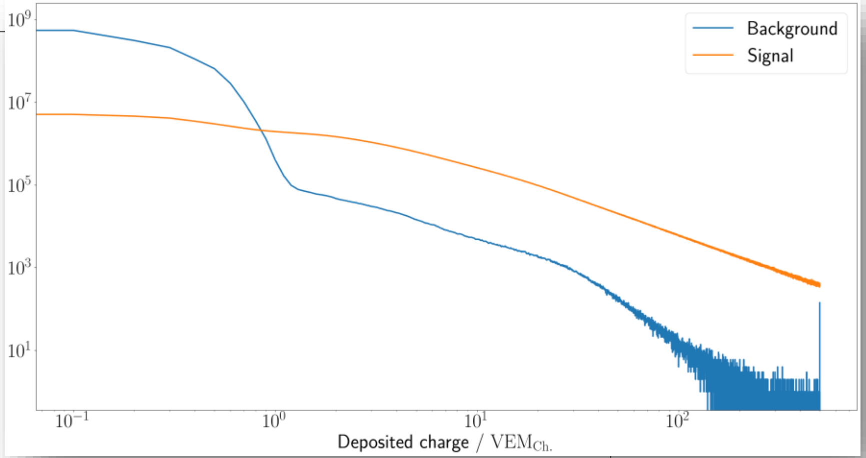
- **Likelihood analysis of signal / background dataset only**
- **Use likelihood ratio λ_{LR} to distinguish**



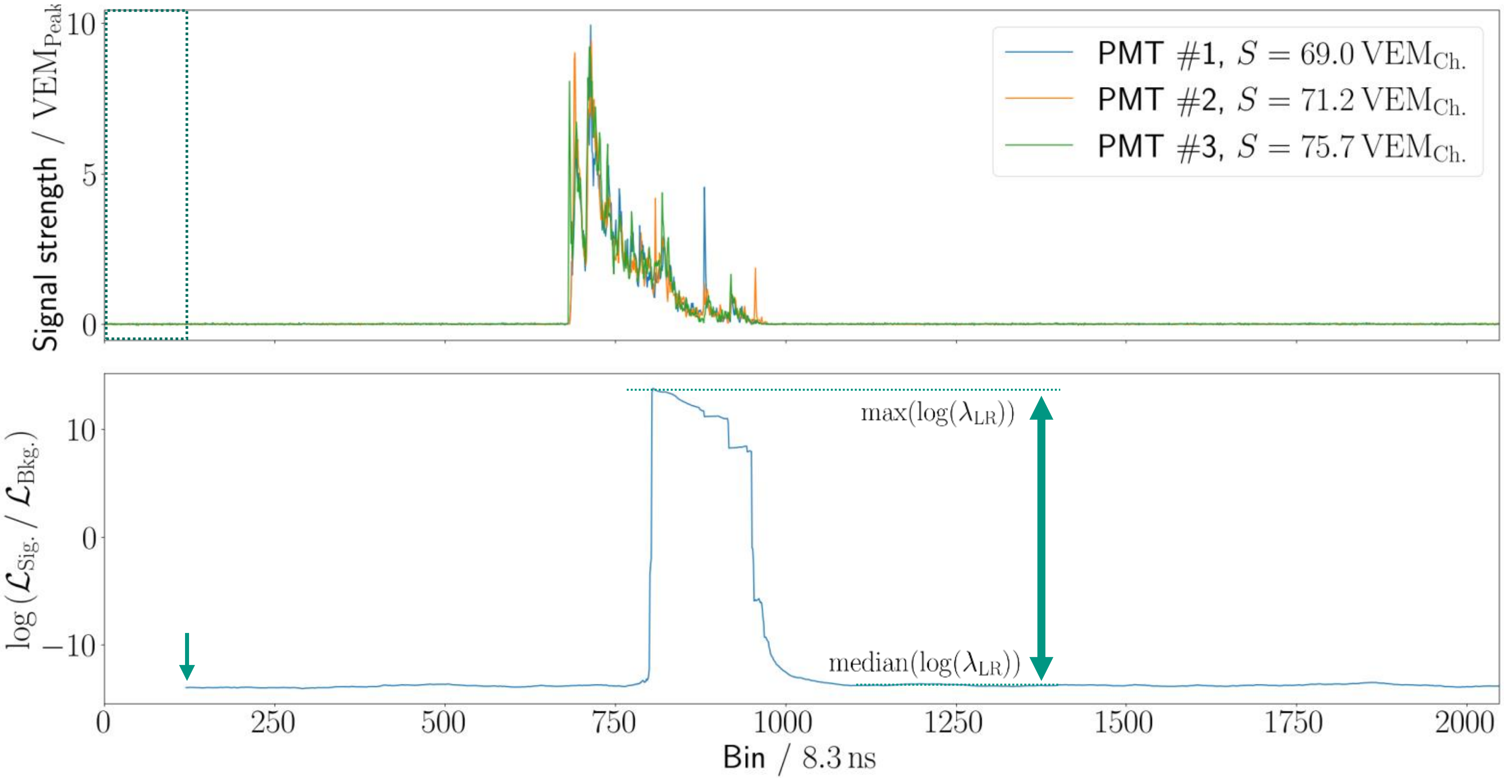
Intermezzo: Bayesian classifier

- Likelihood analysis of signal / background dataset only
- Use likelihood ratio λ_{LR} to distinguish



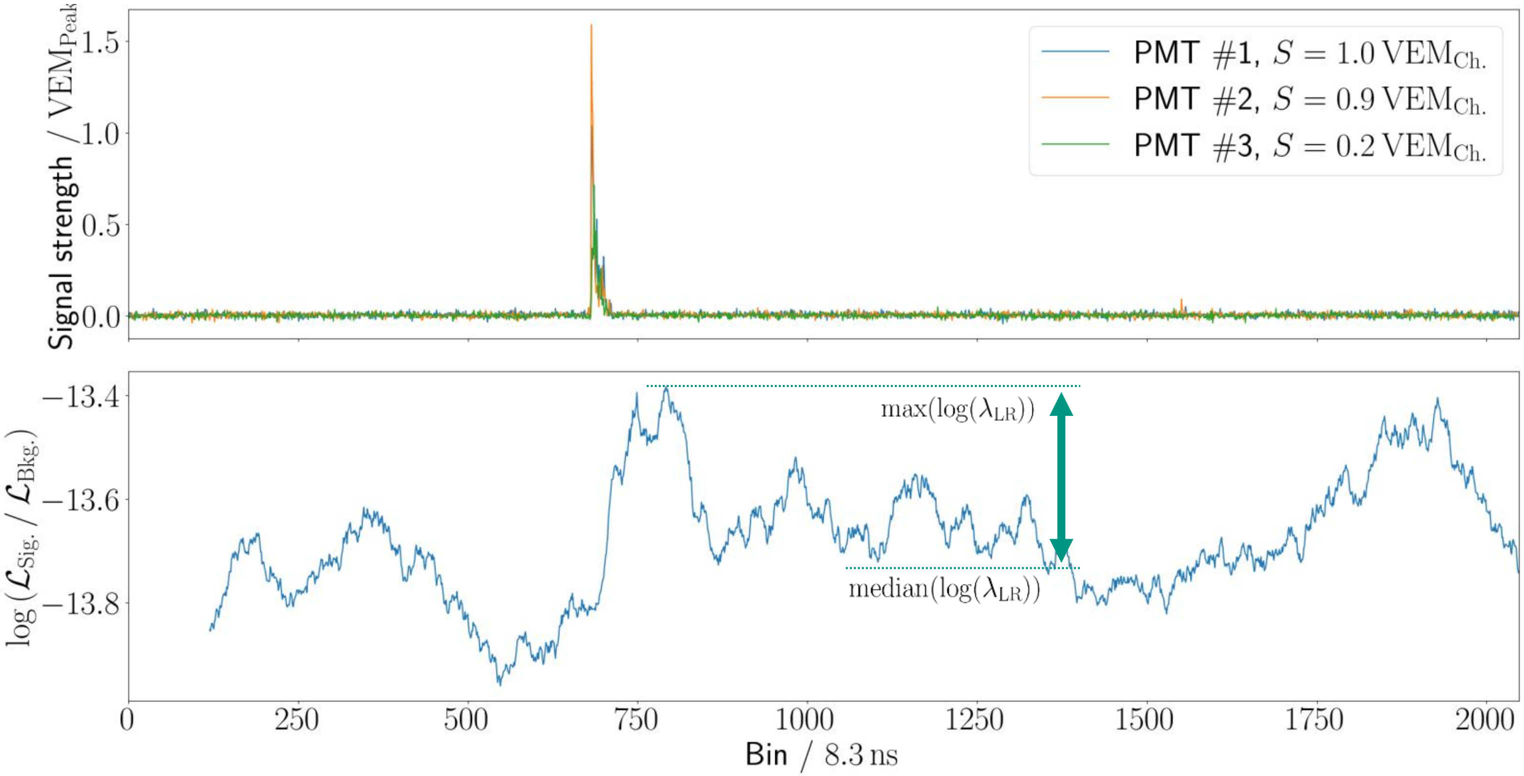


Intermezzo: Bayesian classifier



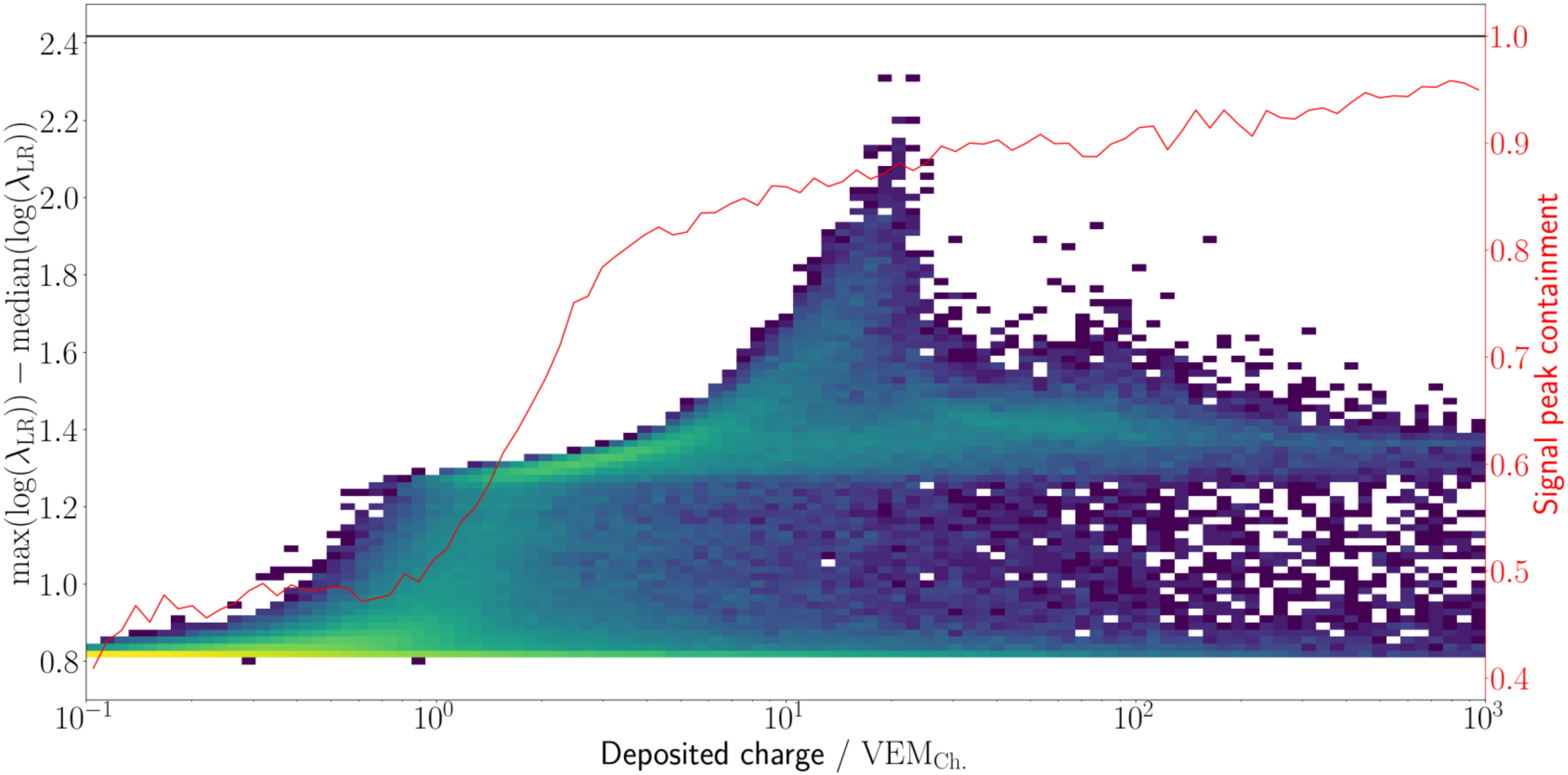
- **Use 120 bins for shape distinction**
- **Good resolution for large signals**

Intermezzo: Bayesian classifier



- **Use 120 bins for shape distinction**
- **Bad resolution for small signals**

Intermezzo: Bayesian classifier



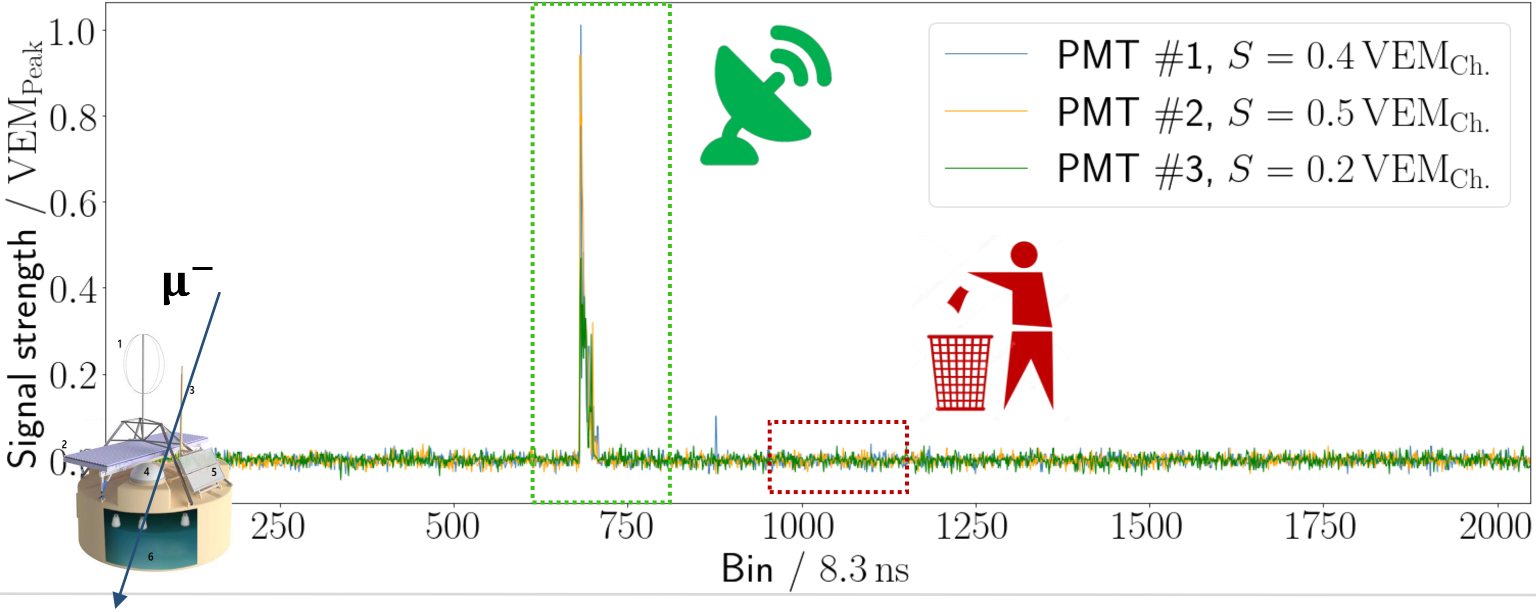
Conclusion: Bayesian classifier

- (Theoretically) promising…
  - Best singular test statistic (Neymann-Pearson-Lemma)
  - Simple / fast hardware implementation
  - Minimal diskspace required

- … but (realistically) much work needed
  - Trigger efficiency bad at small signals
  - Finetuning process of λ_{LR} complicated
  - Thoughts / Suggestions welcome… =)

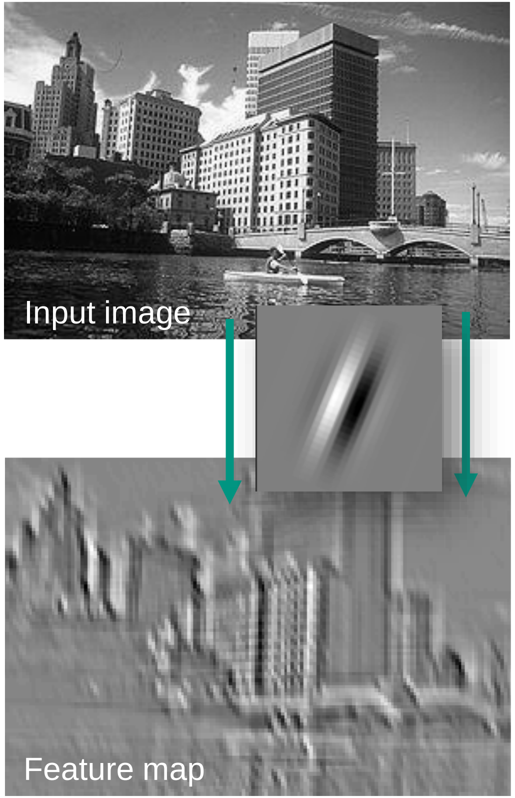
Neural network triggers

- Feed subset of trace to neural network architecture
- Teach it to distinguish between Signal / Background



Convolutional neural networks

- Specialized for image / object recognition
  - Different filters (matrices) scan parts of an image
  - Large output where filter and image look alike
  - Emergent object detection through multiple layers
- Treat WCD time traces as pictures
  - 3 PMTs represent image height
  - Temporal component as width



Taken from https://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/

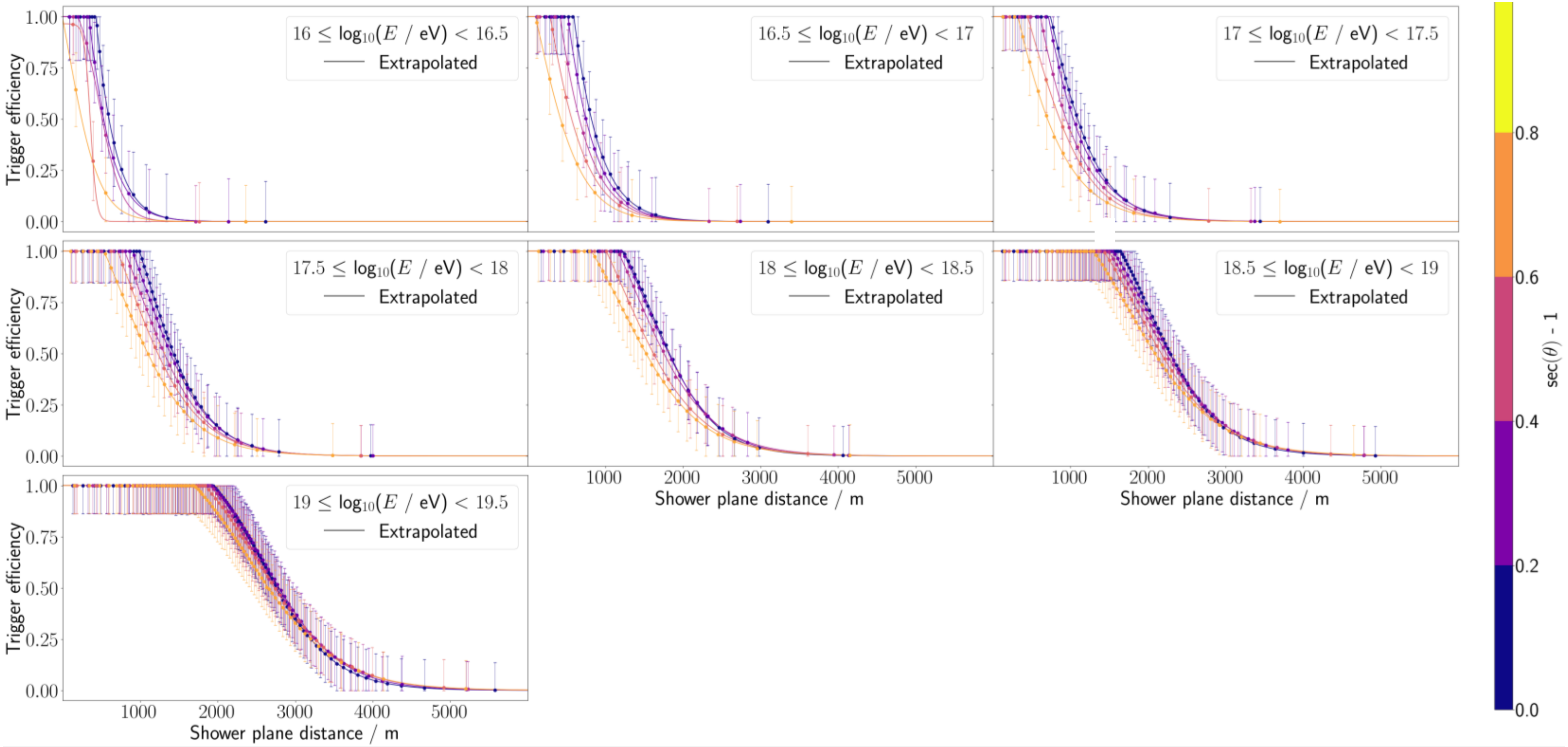
Convolutional neural networks

- Opt for 120 bin input size
  - Shown to work for ToT, ToTd, MoPS
  - Compromise network size and input information

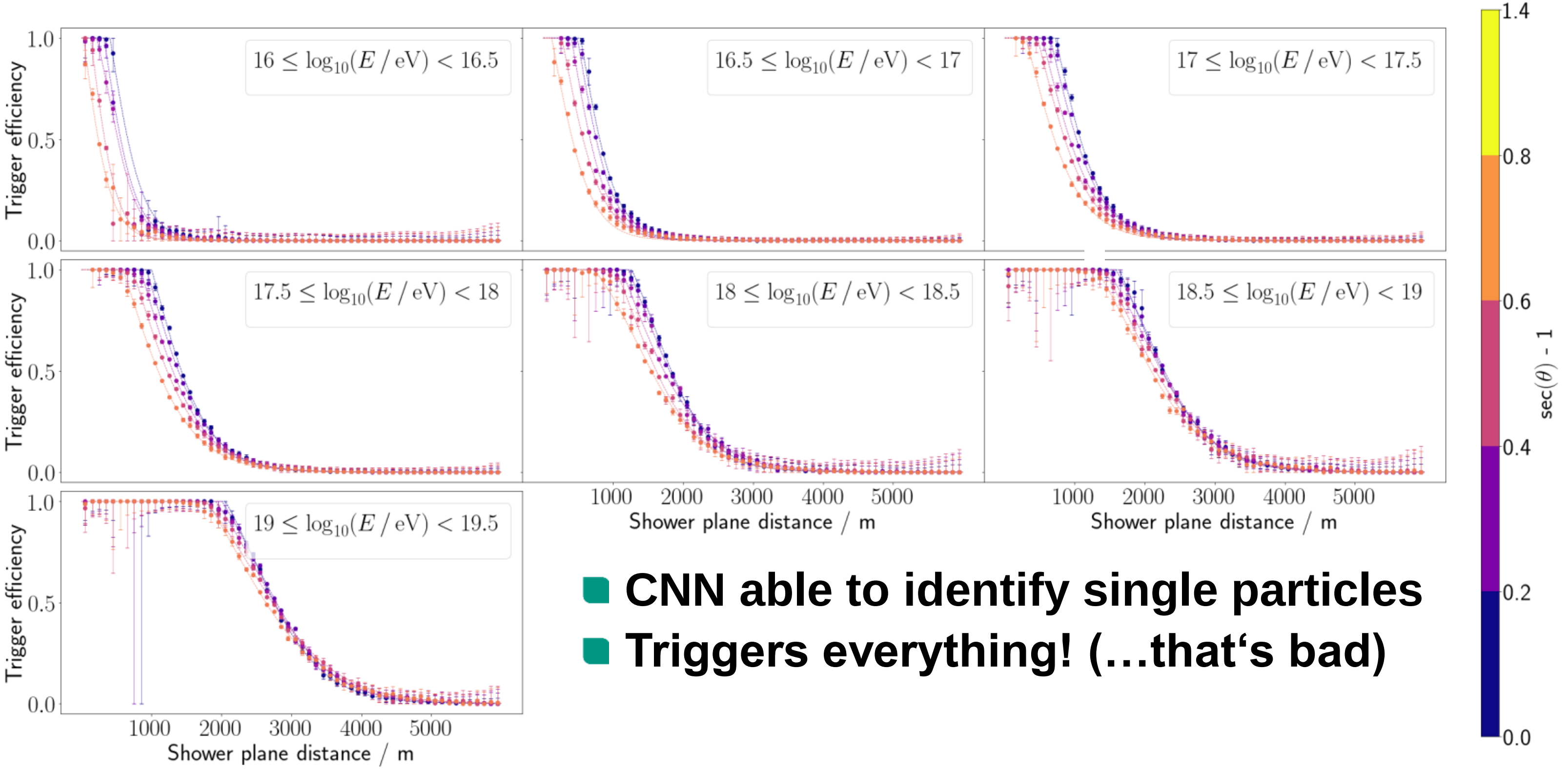
- Data motivated network architecture
  - 2D convolutional layer recognizes correlation between 3 PMTs
  - Dense layer propagates extracted features to output

- Evaluate performance for example architecture
  - Train on 32k simulated proton showers with $16.0 \leq \log_{10}(\mathrm{E} / \mathrm{eV}) < 19.5$

Convolutional neural networks

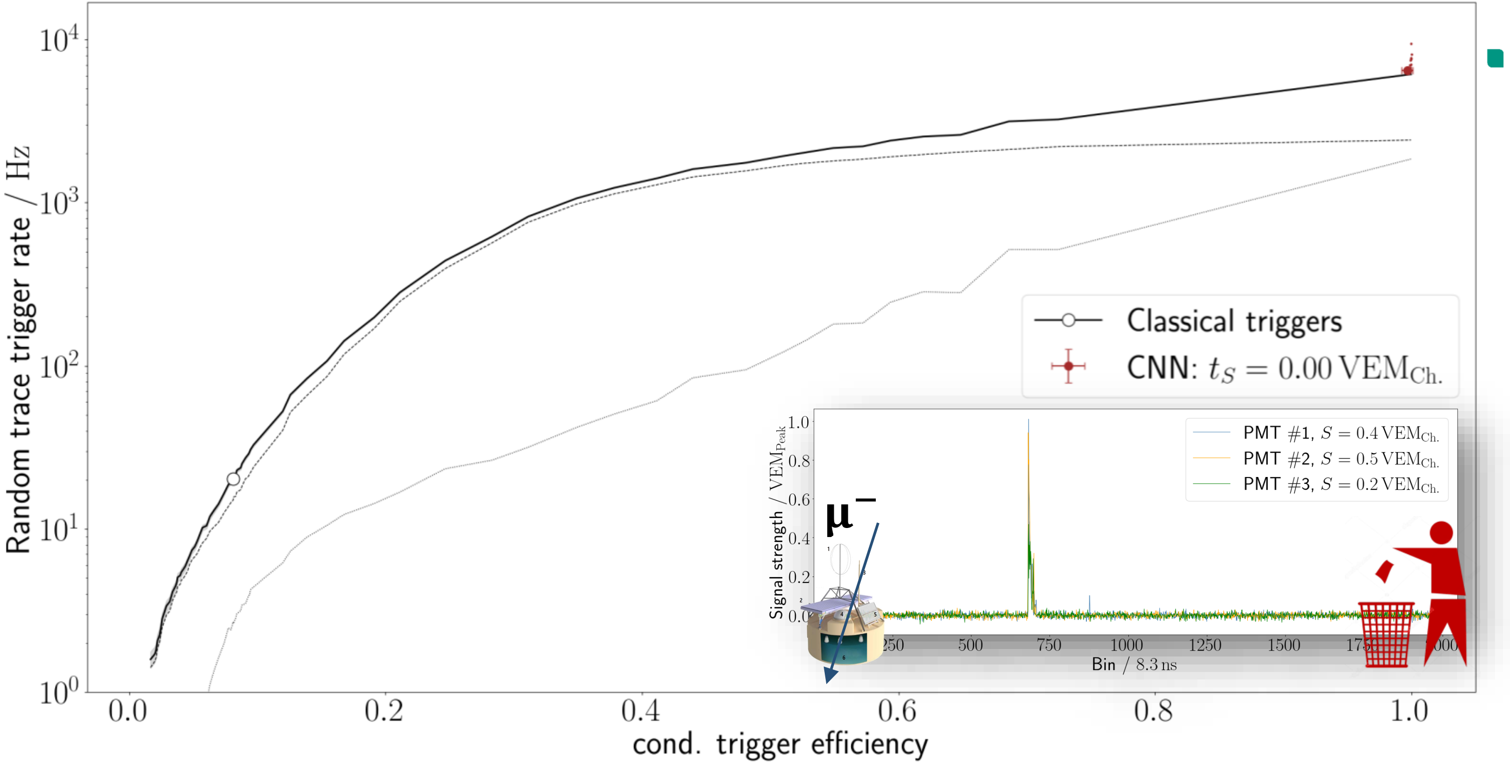


Convolutional neural networks



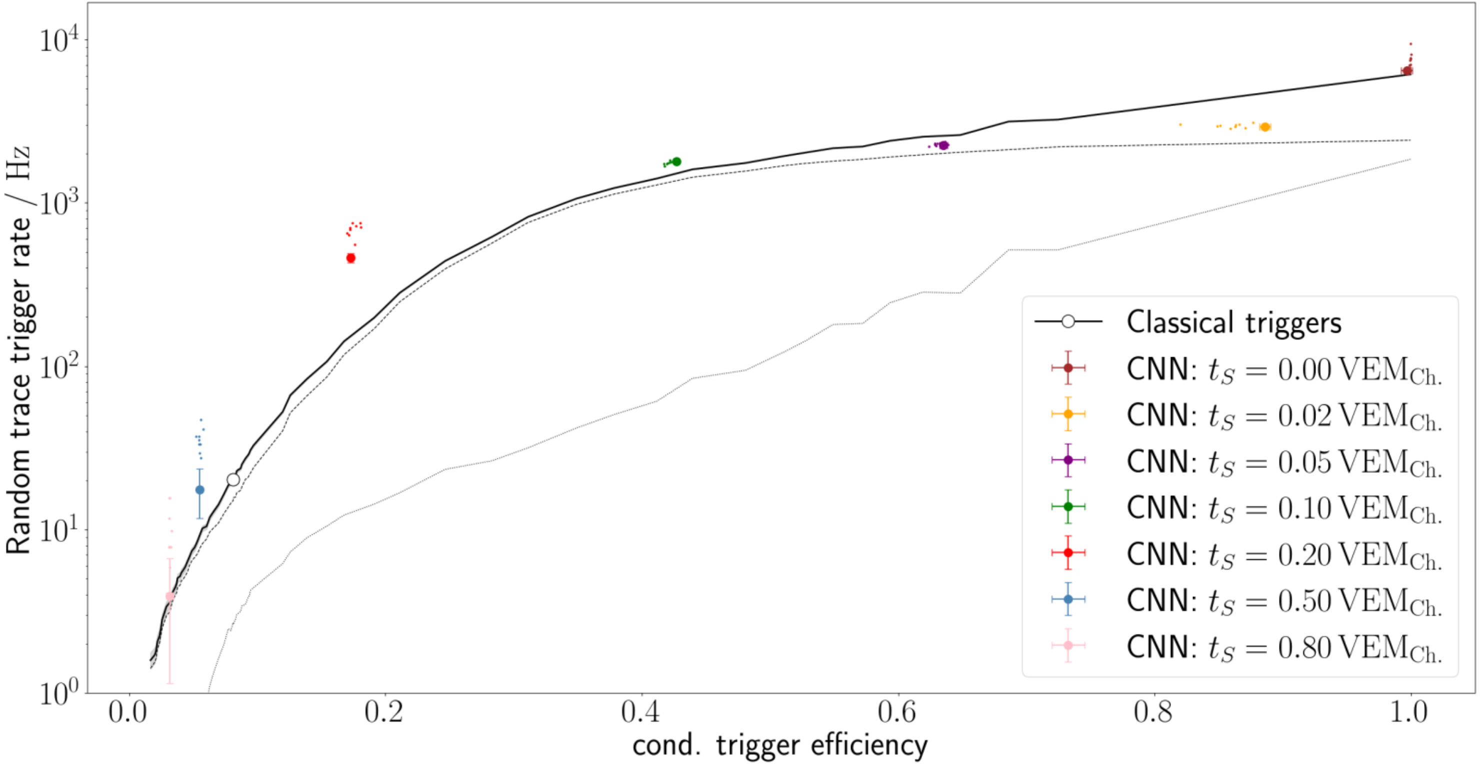
- CNN able to identify single particles
- Triggers everything! (…that‘s bad)

Convolutional neural networks



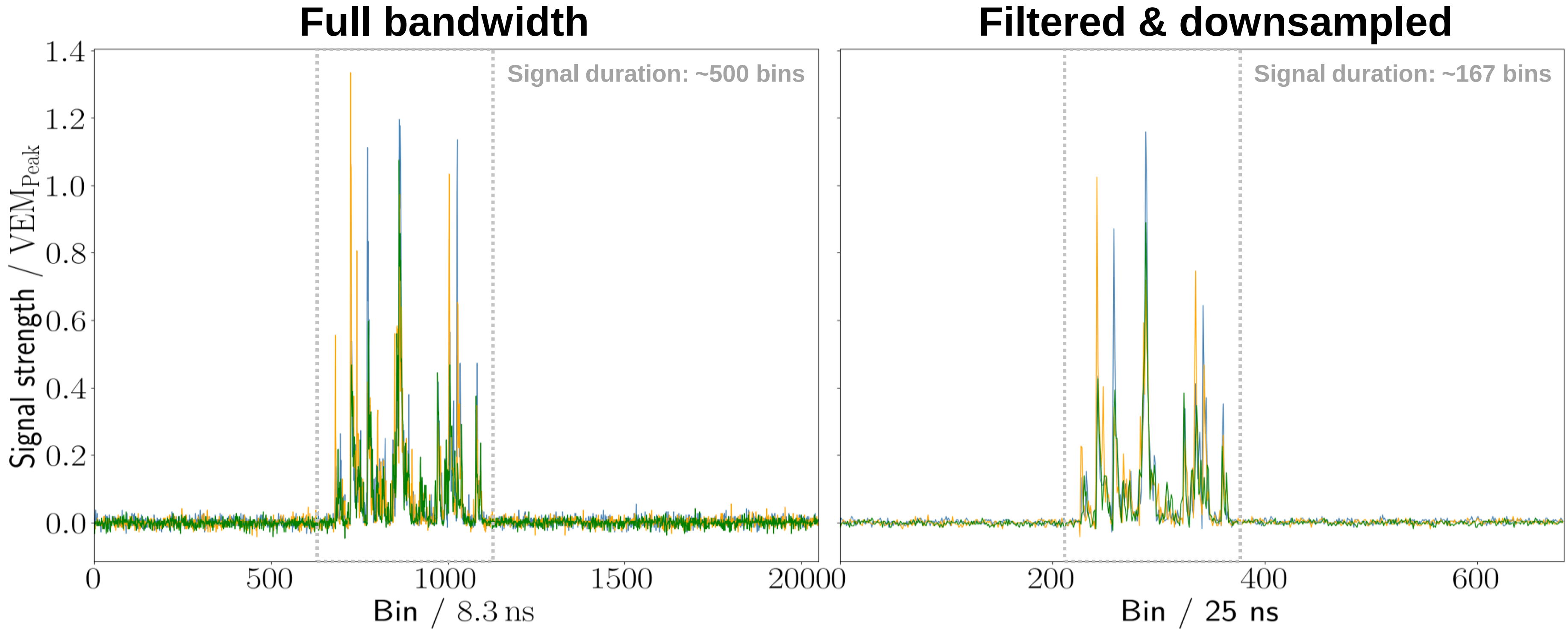
- Redefine „signal“
  - Ignore low $\mathrm{VEM_{Ch.}}$

Convolutional neural networks

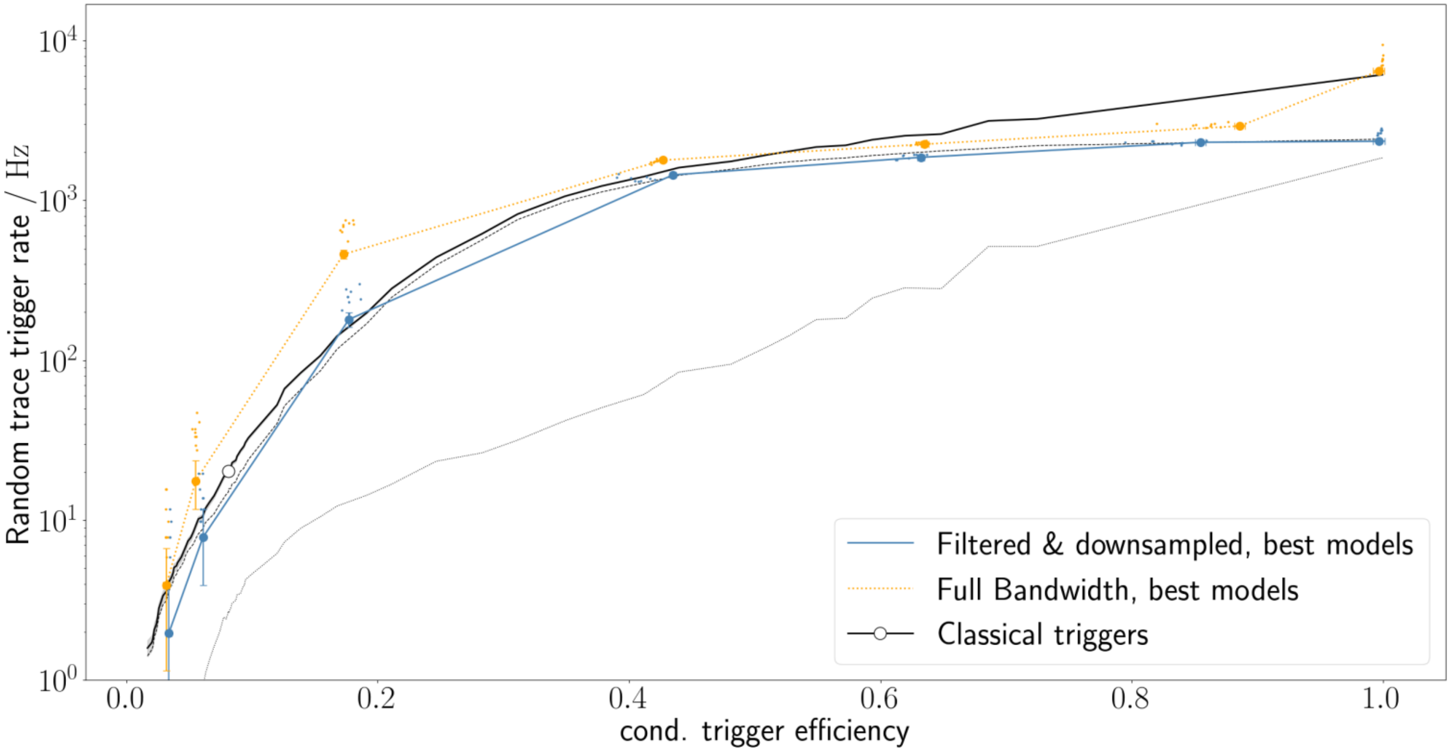


- **Redefine „signal“**
  - **Ignore low $\mathrm{VEM_{Ch.}}$**
  - **High $t_S \Rightarrow$ low rate**
- **Overall performance**
  - **CNN better at low t_S**
  - **Classical triggers better at high t_S**
- **Ideas for improving**
  - **Adjust architecture**
    - **Kernel size**
    - **…**
  - **Filter & downsample WCD time traces**

Intermezzo: Filtering and downsampling

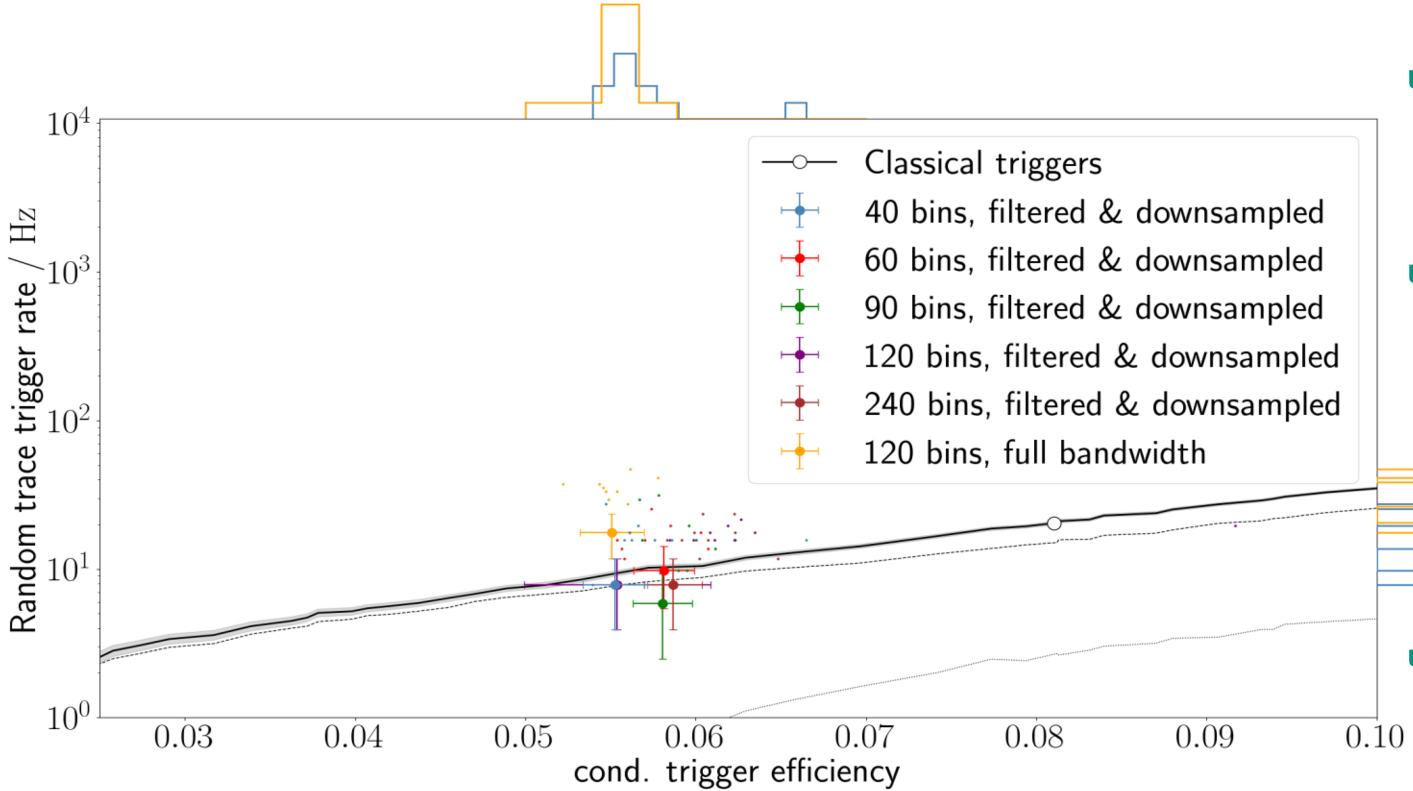


Convolutional neural networks



- **Filtered & downsampled traces preferred over full bandwidth traces at equal input size**

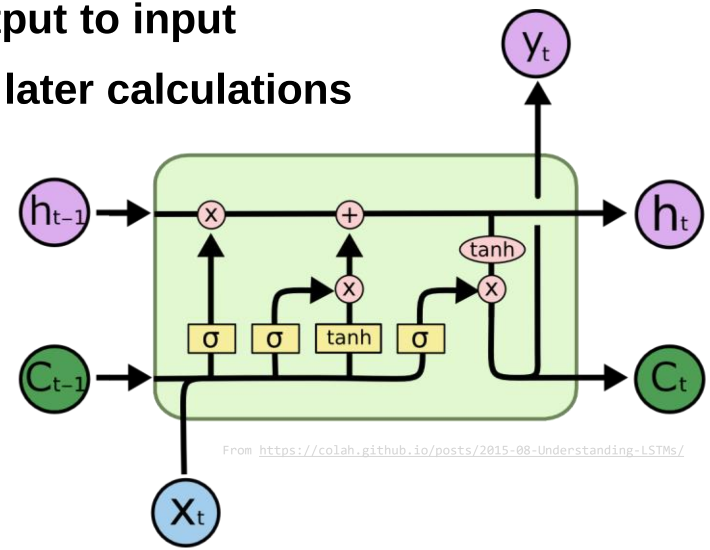
Convolutional neural networks



- Filtered & downsampled traces preferred over full bandwidth traces at equal input size
- Filtered & downsampled input preferred for equal time slices as well
- Smoothing of trace shape reduces electronic noise?

Recurrent neural networks

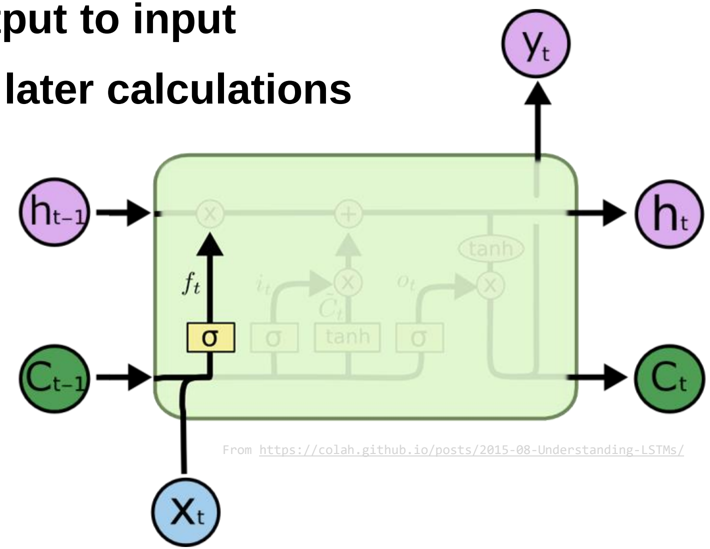
- Long-Short-Term-Memory (LSTM) architecture
  - Has internal connections that point from output to input
  - Earlier processed information can influence later calculations
  - Treat time series very efficiently / elegantly



From https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent neural networks

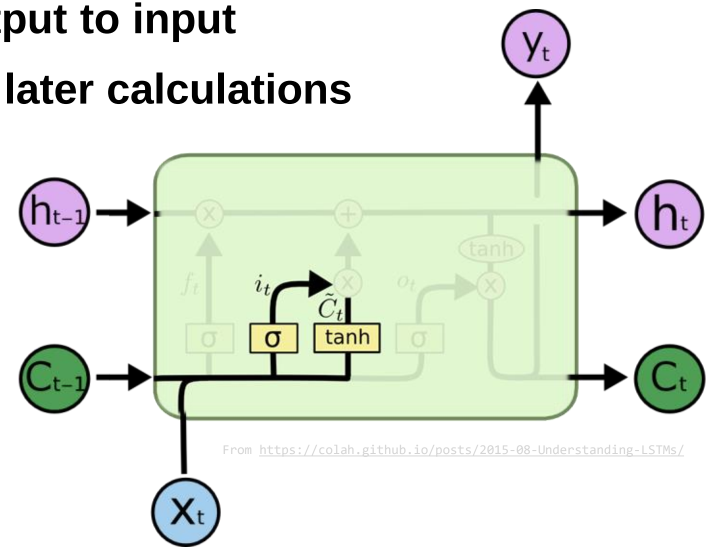
- Long-Short-Term-Memory (LSTM) architecture
  - Has internal connections that point from output to input
  - Earlier processed information can influence later calculations
  - Treat time series very efficiently / elegantly
- Forget-Gate
  - What to keep from previous iterations
- Input-Gate
  - What to save from this iteration
- Output-Gate
  - What to output from (updated) cell state



From https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent neural networks

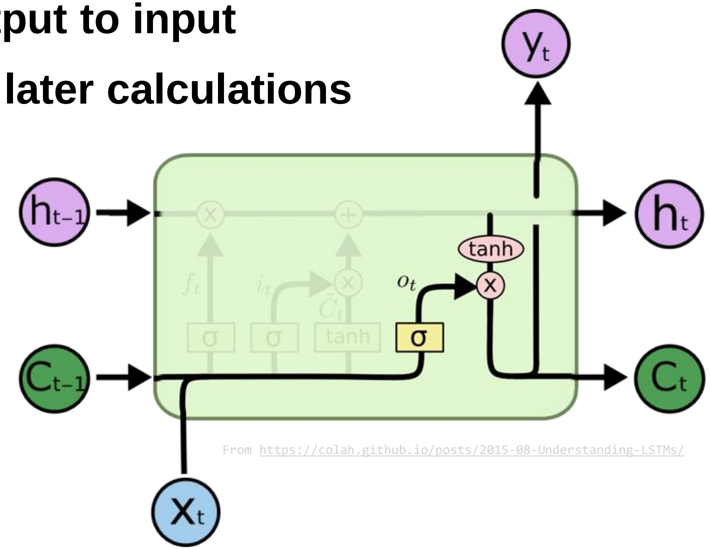
- Long-Short-Term-Memory (LSTM) architecture
  - Has internal connections that point from output to input
  - Earlier processed information can influence later calculations
  - Treat time series very efficiently / elegantly
- Forget-Gate
  - What to keep from previous iterations
- Input-Gate
  - What to save from this iteration
- Output-Gate
  - What to output from (updated) cell state



From https://colah.github.io/posts/2015-08-Understanding-LSTMs/

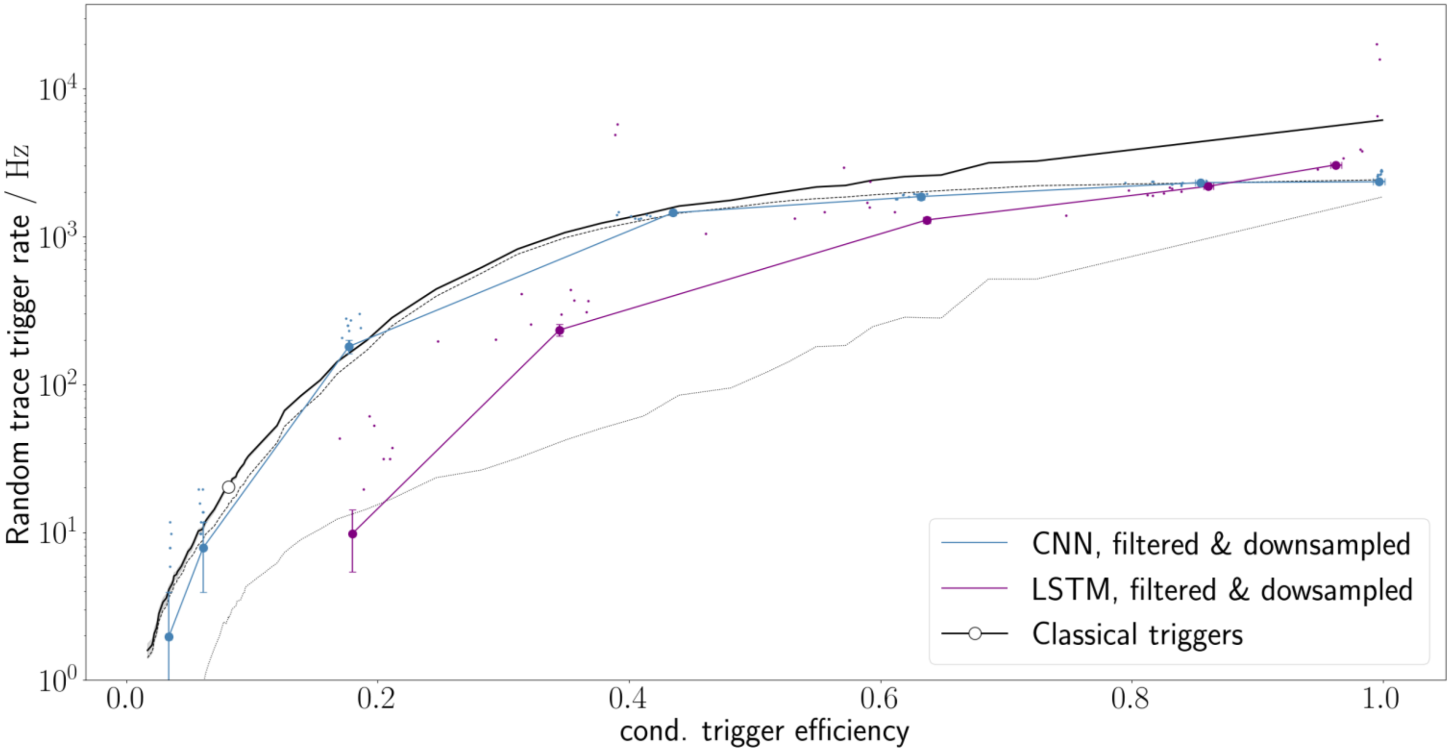
Recurrent neural networks

- Long-Short-Term-Memory (LSTM) architecture
  - Has internal connections that point from output to input
  - Earlier processed information can influence later calculations
  - Treat time series very efficiently / elegantly
- Forget-Gate
  - What to keep from previous iterations
- Input-Gate
  - What to save from this iteration
- Output-Gate
  - What to output from (updated) cell state



From https://colah.github.io/posts/2015-08-Understanding-LSTMs/

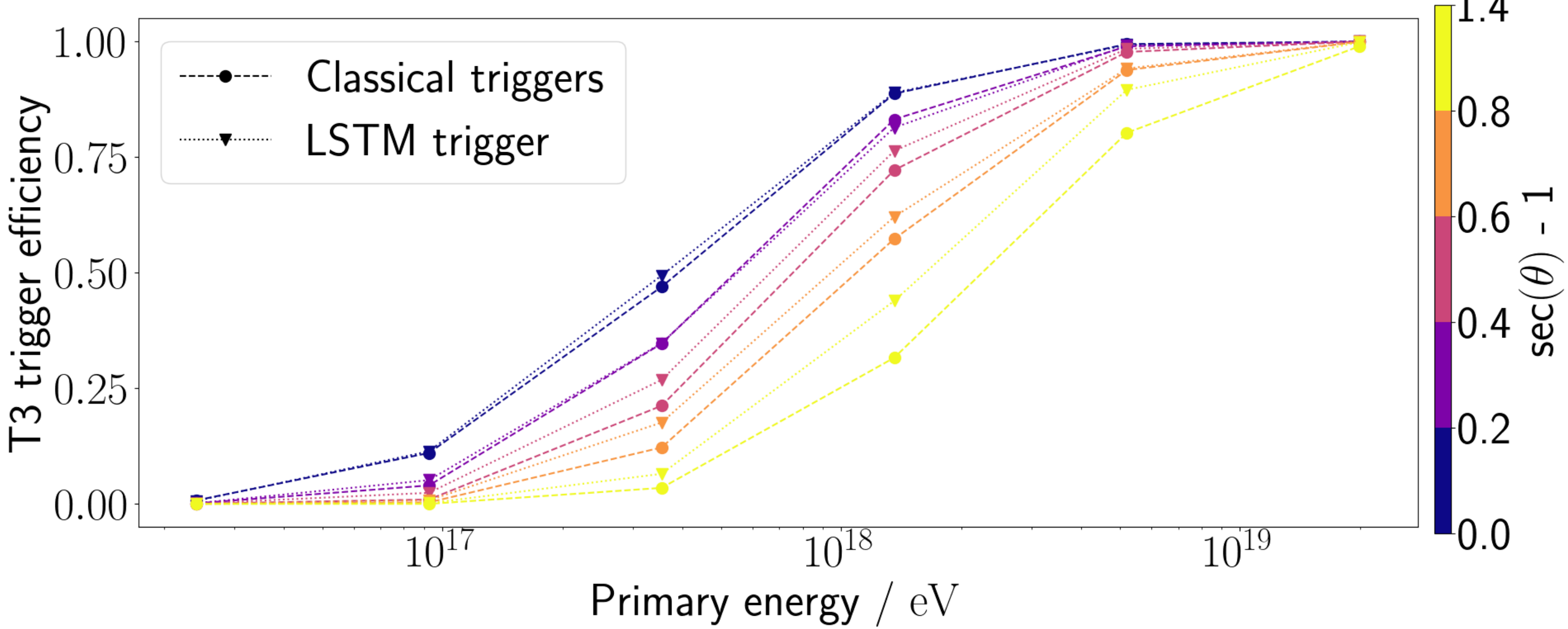
Recurrent neural networks



- **LSTM performance better than Th trigger**
- **Can be better than ToT for $t_S = 0.5\ \mathrm{VEM}_{\mathrm{Ch.}}$**
- **Just 44 parameters**
- **Effects on event level detection efficiency?**

Recurrent neural networks

- **Most drastic gains at inclinations** $\theta \approx 60°$ $(+16.5\%)$
- **Possibly higher gains at** $65° \leq \theta < 90°$ …



Summary / Outlook

- Test data-driven, machine learning concepts
  - Bayesian classifier promising, but needs lots of finetuning
  - Neural networks work out of the box but „too efficient“
  - Control trigger rate by implementing charge cut

- Convolutional neural networks
  - Performance of simple CNN architectures on par with Th-Trigger
  - Filtered & downsampled data preferred over full bandwidth input

- LSTM / recurrent neural networks
  - First results indicate better performance than ToT
  - Large gains in event detection efficiency at high shower angles

Summary / Outlook

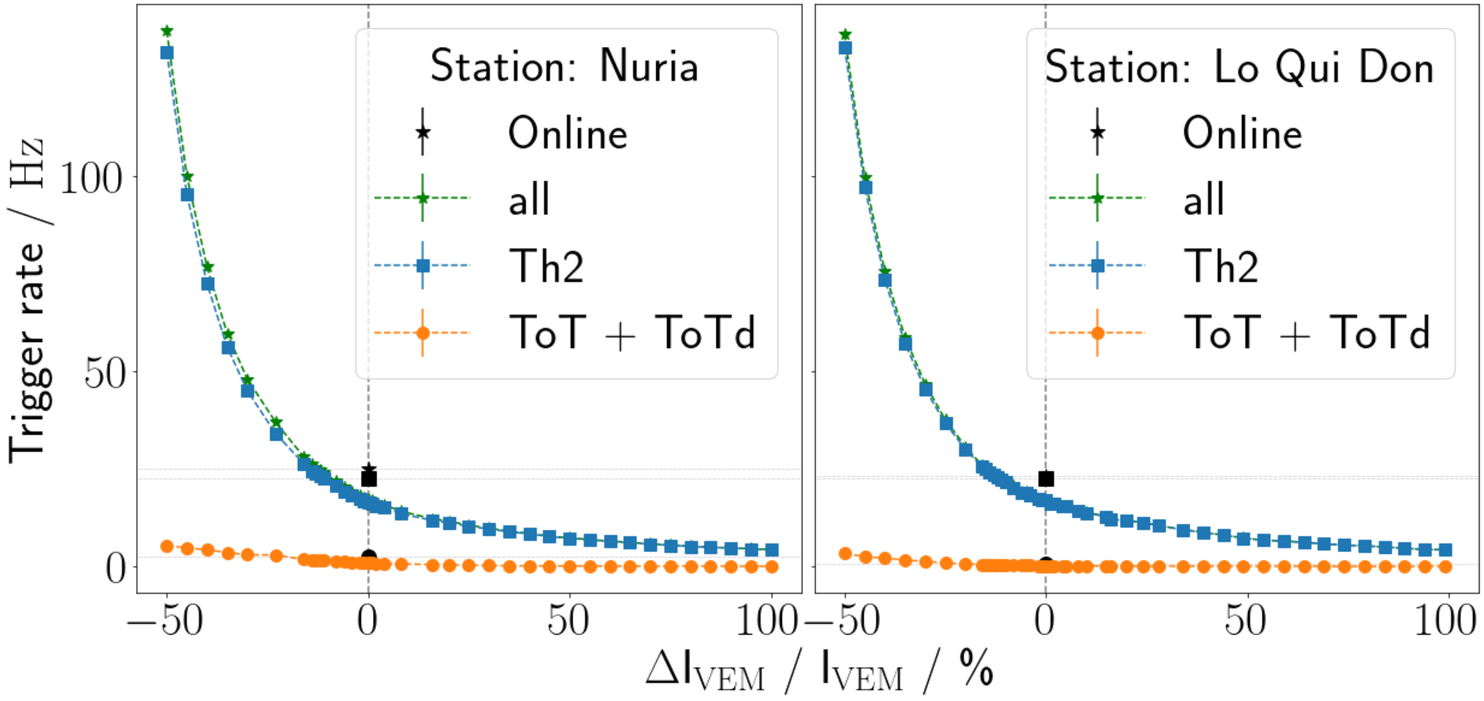
- **Lot of work needed until prototyping stage is left**
  - **Presented results stem from simulations only**
  - **No primary distinction, only data from protons considered**
  - **Only one hadronic interaction model (QGSJET-II.04)**

- **Ground work is completed**
  - **Key assumptions have been tested and verified to hold true**
  - **Analysis chain is implemented and ready to run**

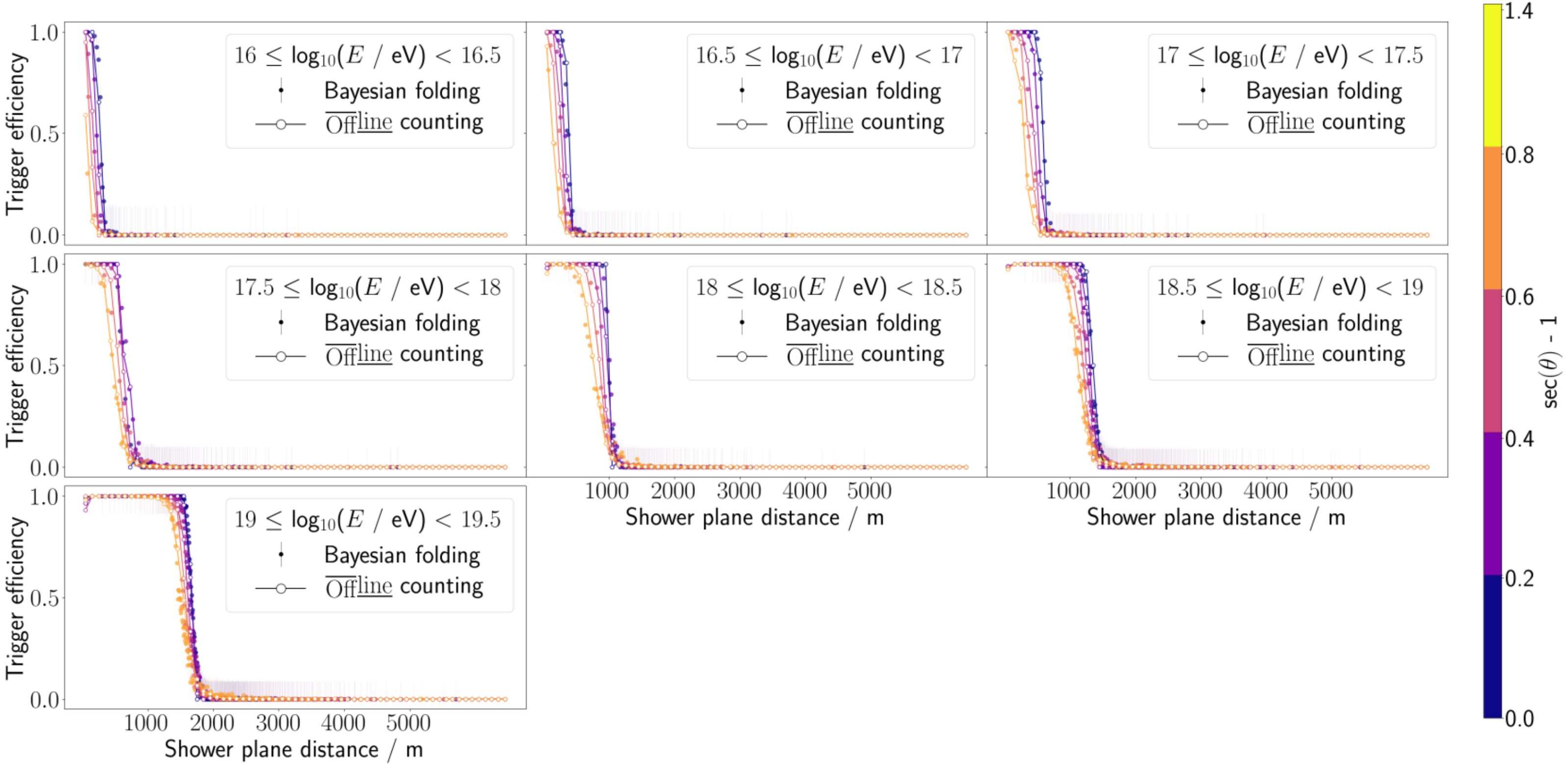
- **Upcoming dataset of easily accessible ($.csv$) WCD time traces**
  - **40k (proton primary) events, tagged by θ, ϕ, E, SPD, n_{μ}, …**
  - **Please tell us what other data you would be interested in**

Backup

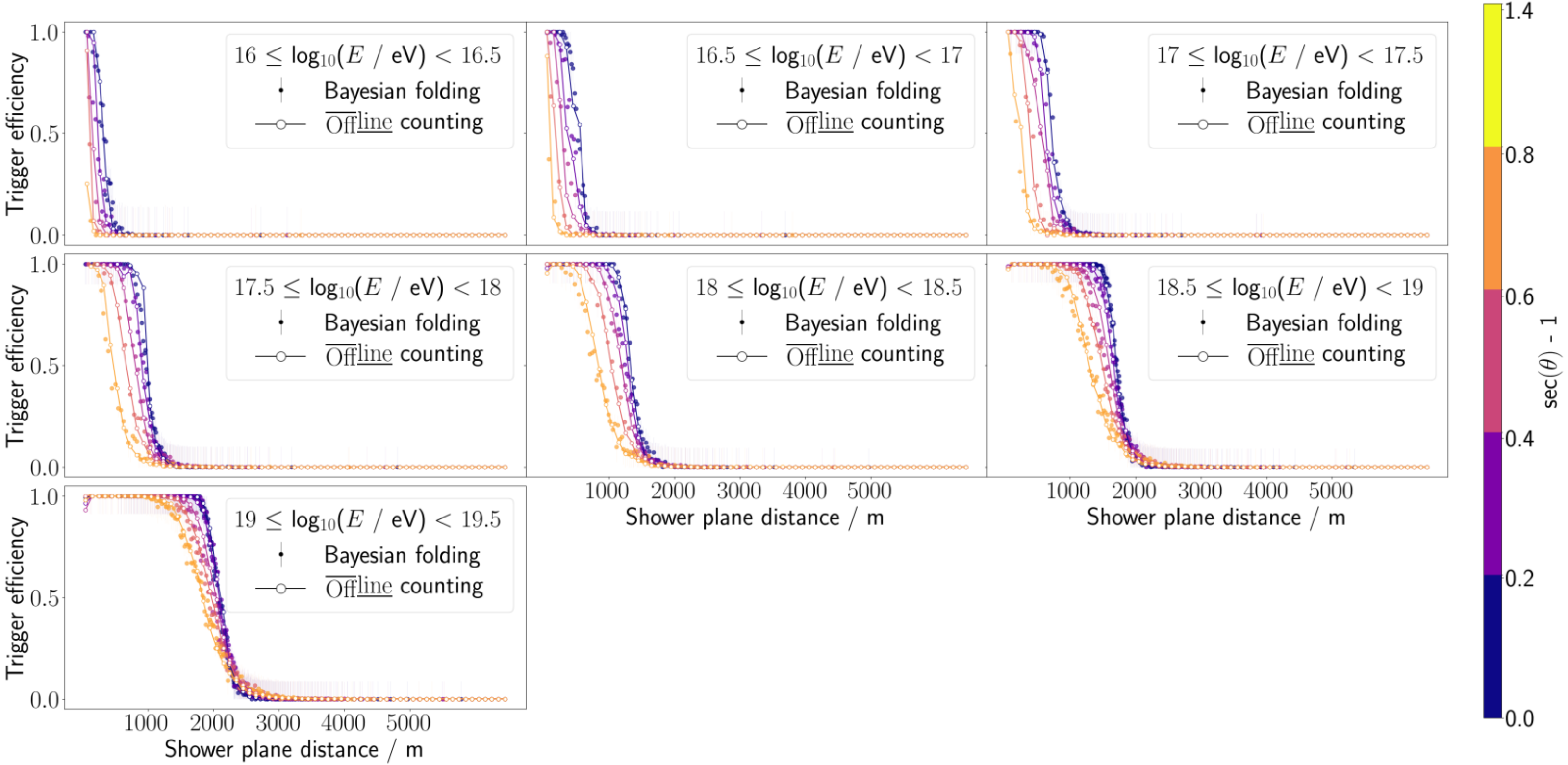
Random traces – Calibration



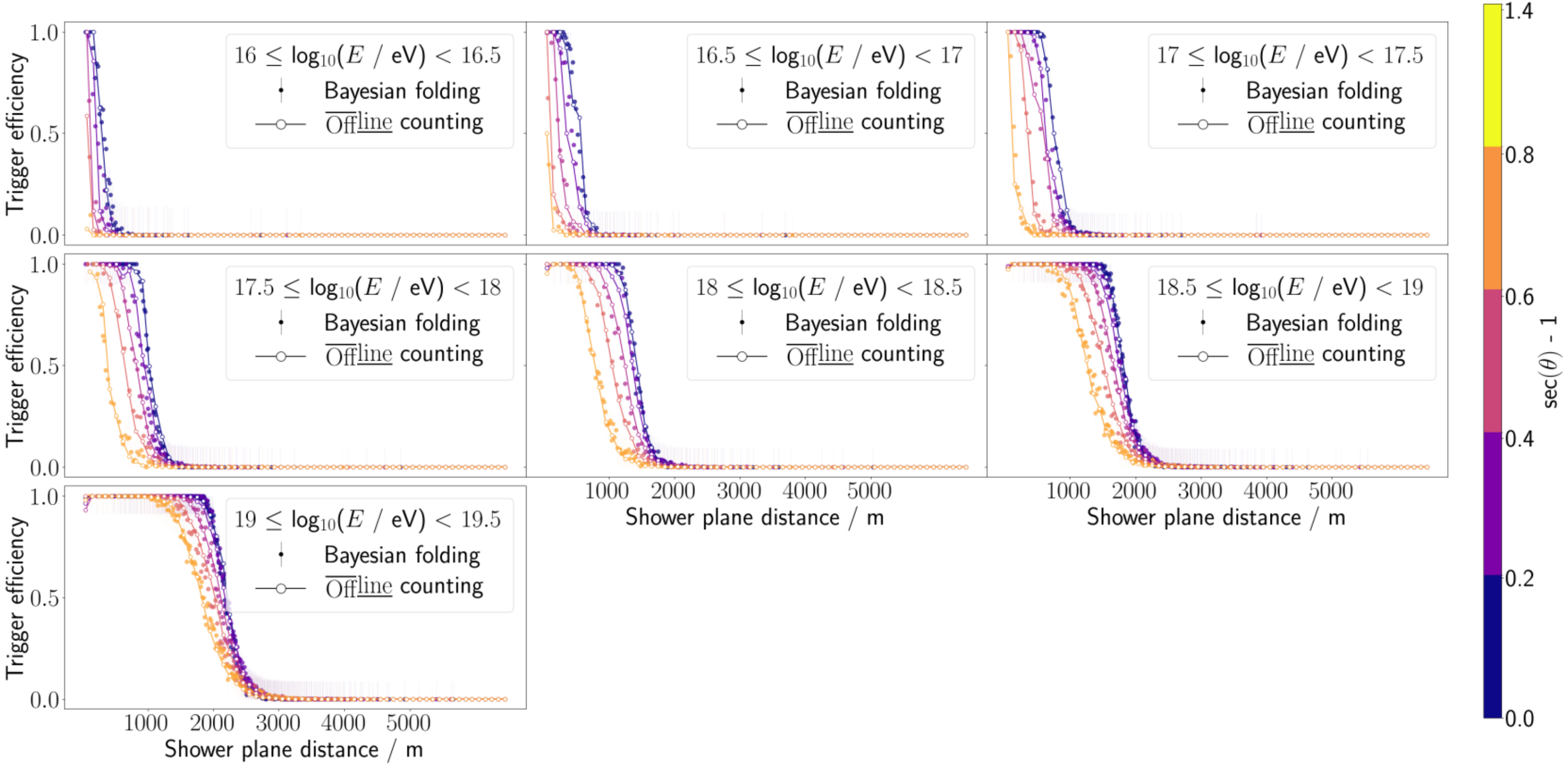
Random traces – Calibration



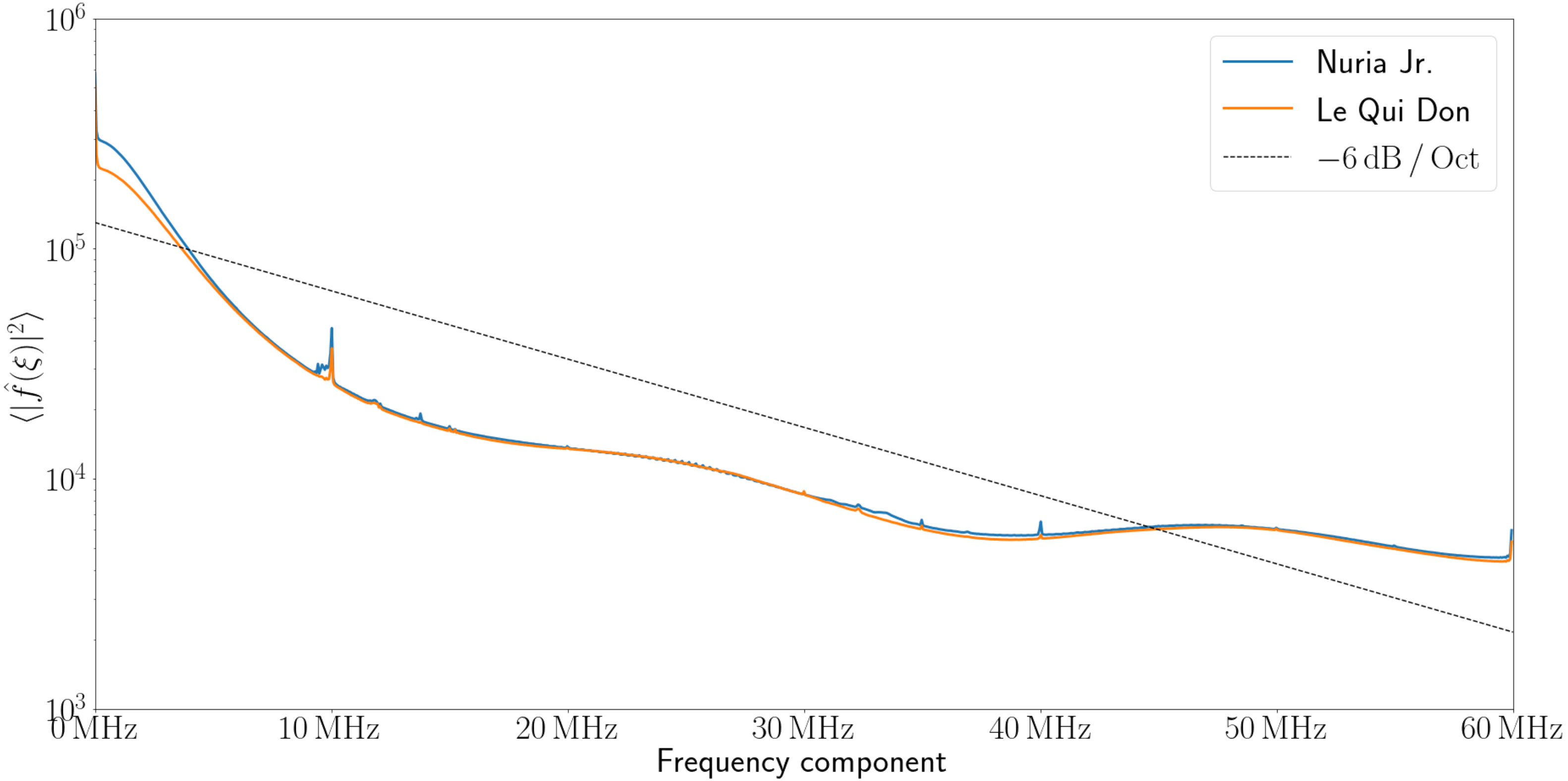
Random traces – Calibration



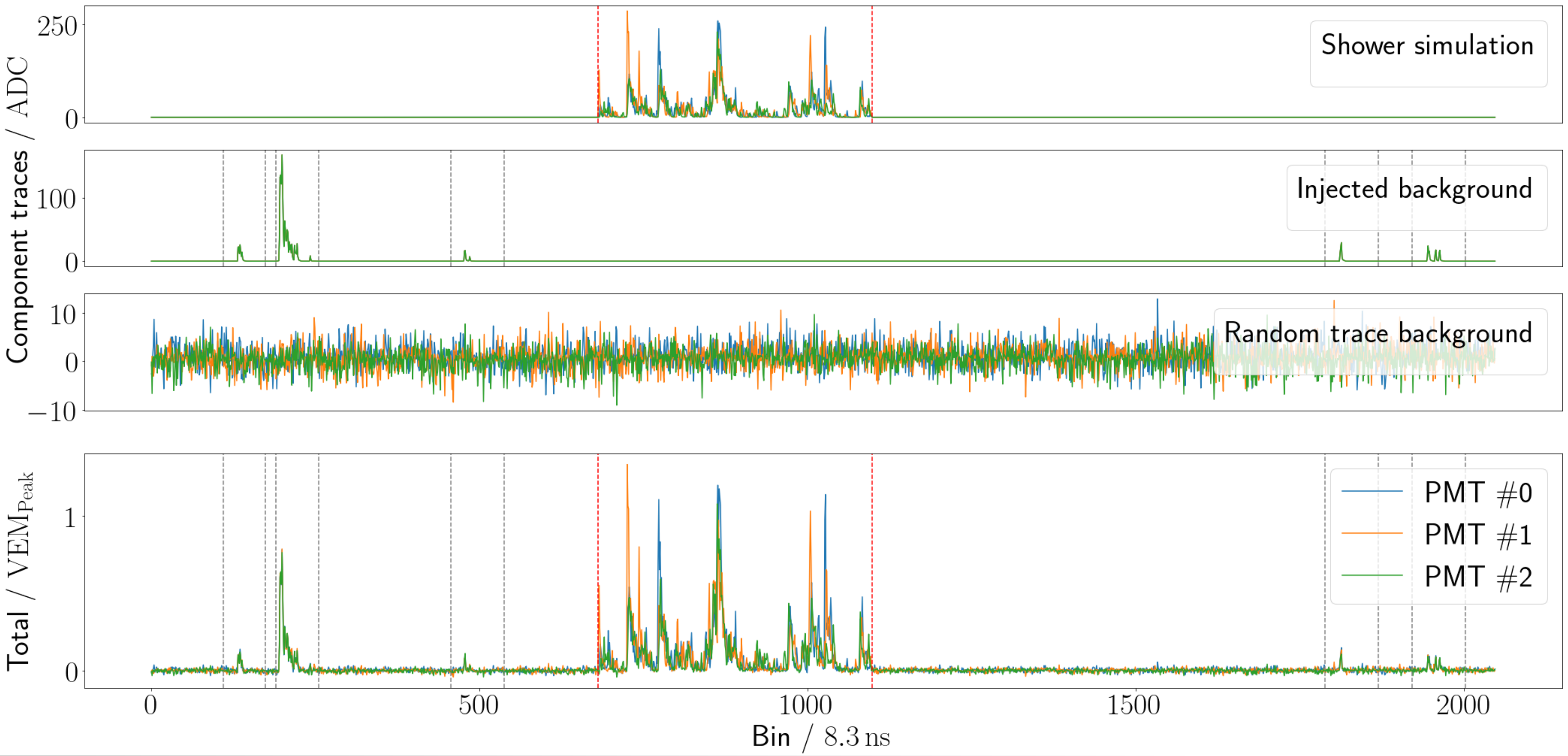
Random traces – Calibration



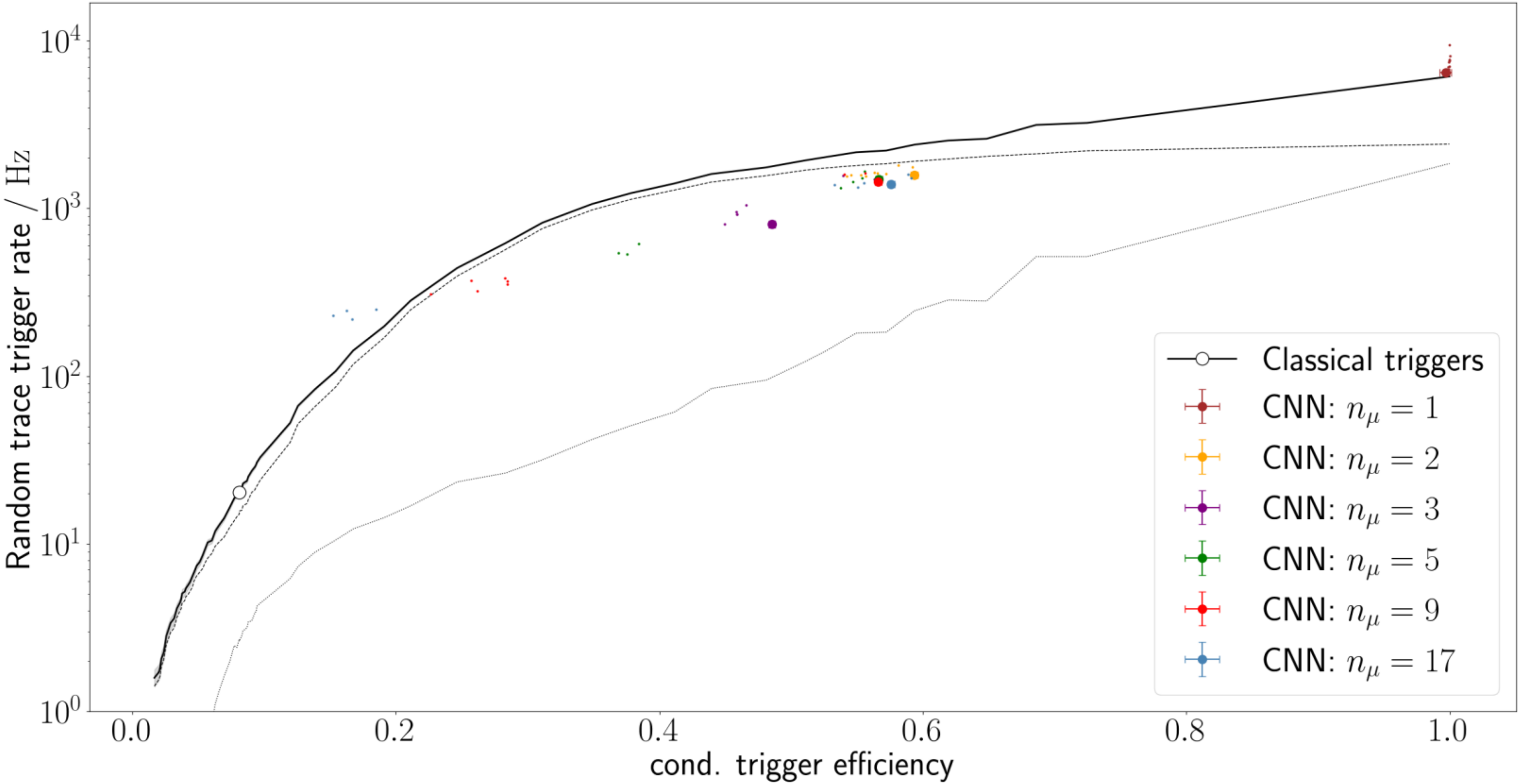
Random traces – Power spectrum



Trace building

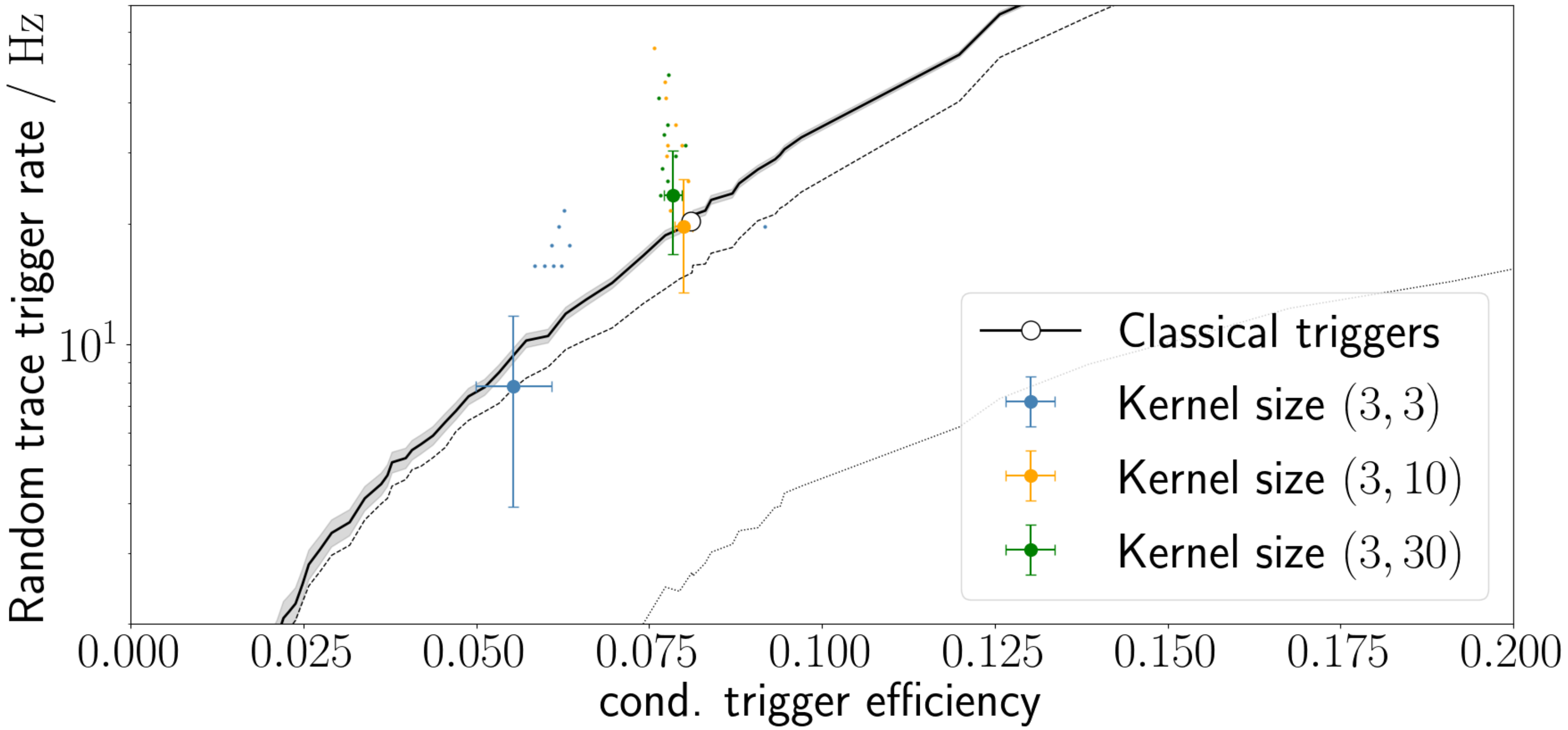


Muon cut



Random trace trigger rate / Hz
cond. trigger efficiency
10^0
10^1
10^2
10^3
10^4
0.0
0.2
0.4
0.6
0.8
1.0
Classical triggers
CNN: $n_\mu = 1$
CNN: $n_\mu = 2$
CNN: $n_\mu = 3$
CNN: $n_\mu = 5$
CNN: $n_\mu = 9$
CNN: $n_\mu = 17$

Kernel size



Random trace trigger rate / Hz
10^1
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
cond. trigger efficiency
Classical triggers
Kernel size $(3, 3)$
Kernel size $(3, 10)$
Kernel size $(3, 30)$

Network parameters

Type	Input size	Kernel size	n_{train}	w/ dense extension
CNN	$(3, 120)$	$(3, 3)$	**140**	**834**
CNN	$(3, 120)$	$(3, 10)$	**216**	**534**
CNN	$(3, 120)$	$(3, 30)$	**444**	**714**
CNN	$(3, 40)$	$(3, 3)$	**84**	**210**
CNN	$(3, 60)$	$(3, 3)$	**100**	**290**
CNN	$(3, 90)$	$(3, 3)$	**120**	**390**
CNN	$(3, 240)$	$(3, 3)$	**220**	**890**
LSTM	$(3, 120)$	–	**12**	(single layer)
LSTM	$(3, 120)$	–	(three layers)	**44**

LSTM permutations