# Preface

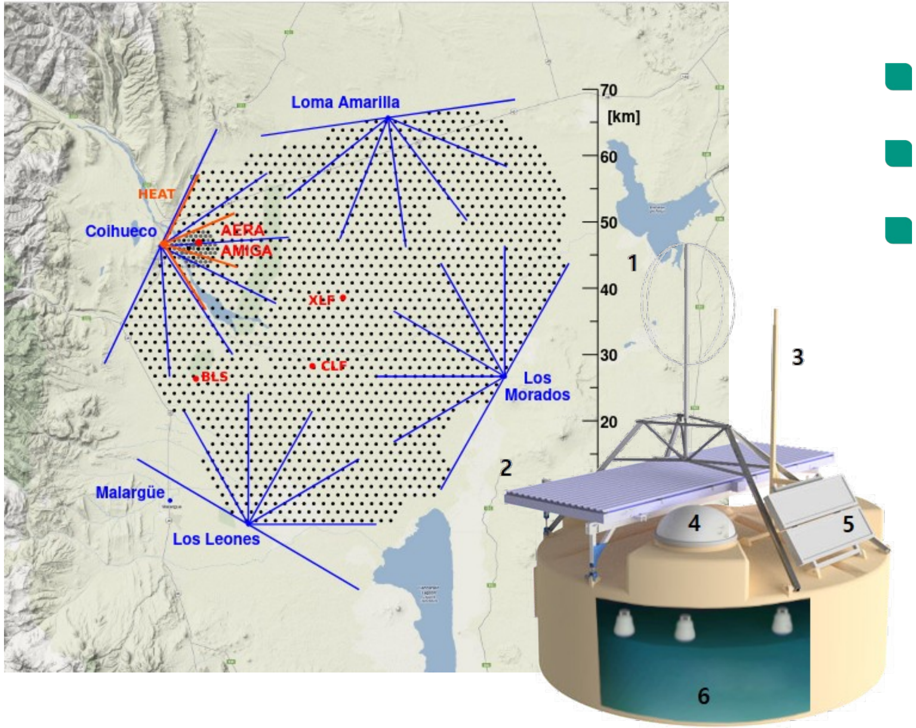
- PhD goals (defined so far, non-extensive list…)
  - Implement station-level algorithm for SSD calibration
  - Involve SSD in DAQ decision process (new triggers…)
  - Analyse gathered data for rare cosmic ray events (γ, υ, GZ-effect…)

- Hopefully more results to show next year ;)

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

Paul Filip – HIRSAP meeting 21.11.2023 – 22.11.2023

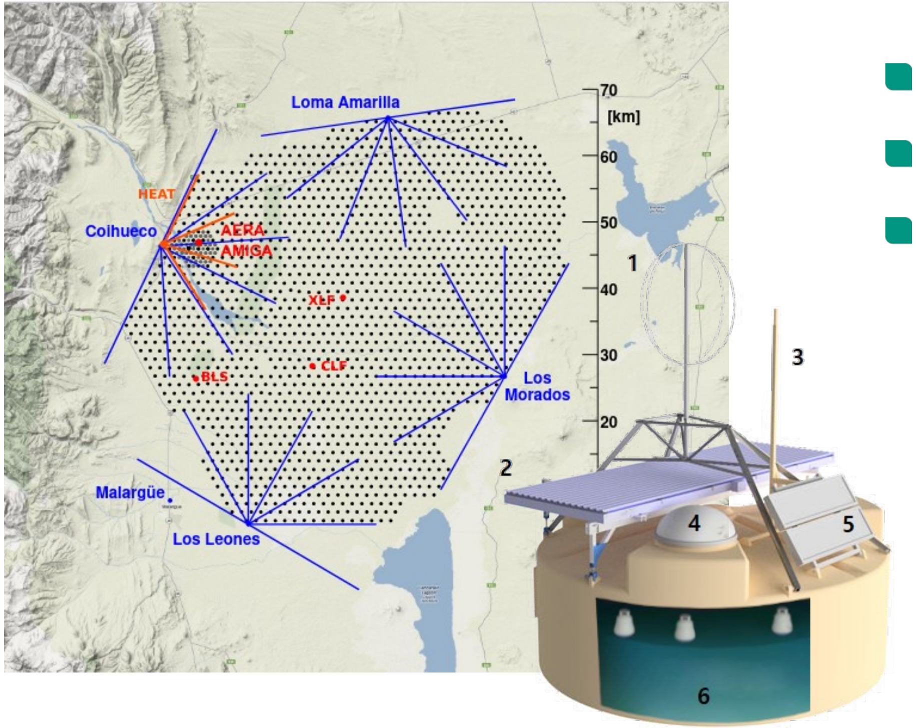
# SD Array / trigger hierarchy / WCD time traces




- Around ~1600 stations
- Triangular 1500 m grid spacing
- Upgrade from UB to UUB
  - 1 Water-Cherenkov detectors (WCD)
  - 1 Surface scintillator detector (SSD)
  - 1 Radio antenna (RD)

# SD Array / trigger hierarchy / WCD time traces

Karlsruhe Institute of Technology

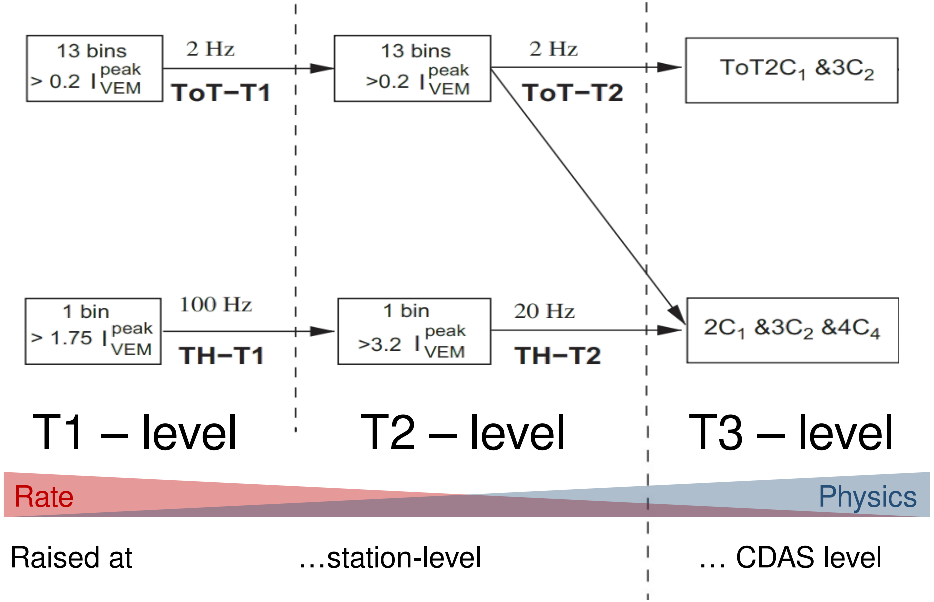



- Around ~1600 stations
- Triangular 1500 m grid spacing
- Upgrade from UB to UUB
  - 1 Water-Cherenkov detectors (WCD)
  - 1 Surface scintillator detector (SSD)
  - 1 Radio antenna (RD)

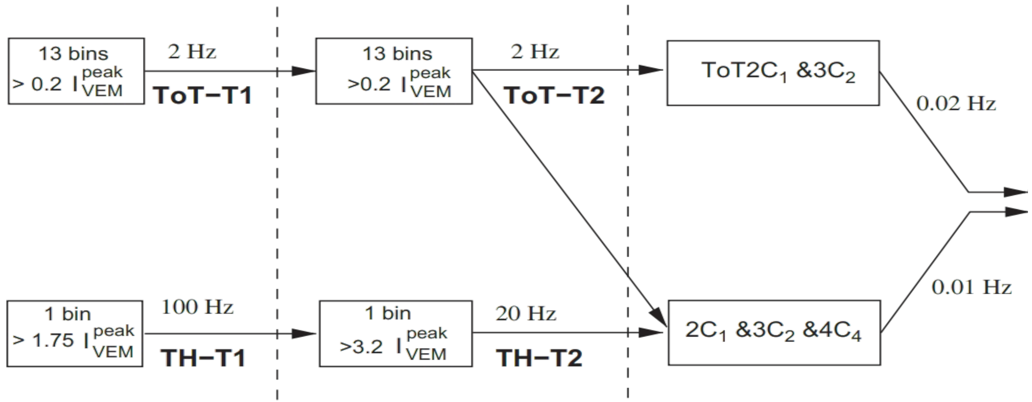
**Too comput. expensive to read all measured data at all times!**

**Implement trigger hierarchy**

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



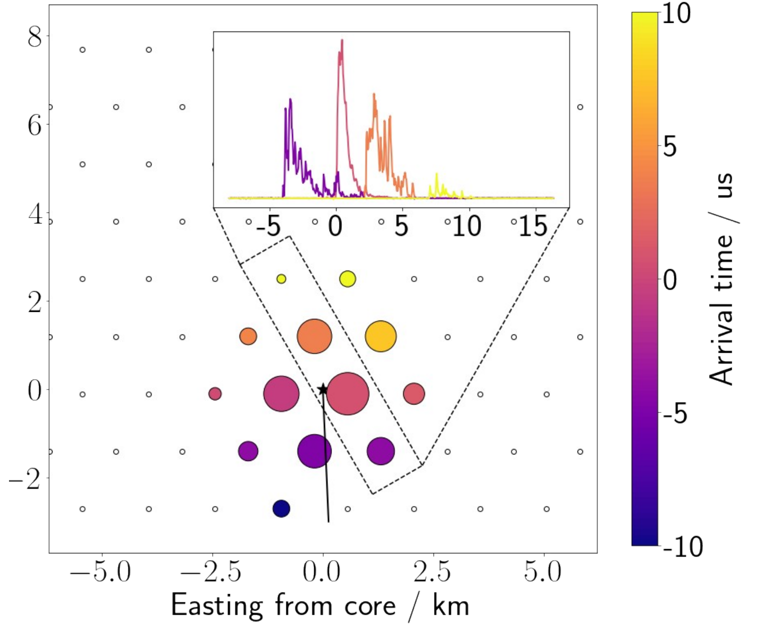
# SD Array / **trigger hierarchy** / WCD time traces



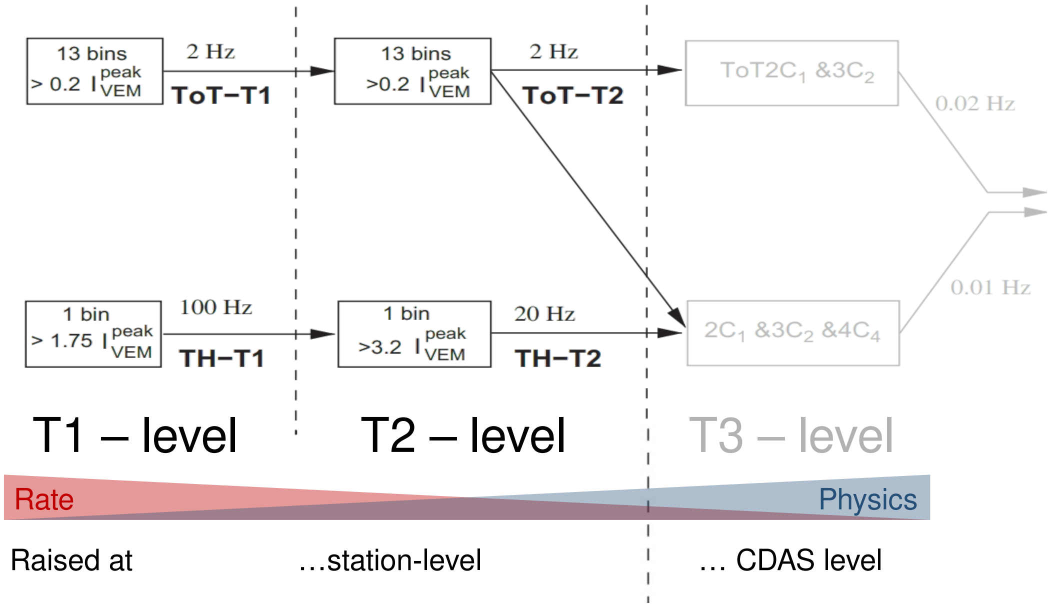

T1 – level | T2 – level | T3 – level

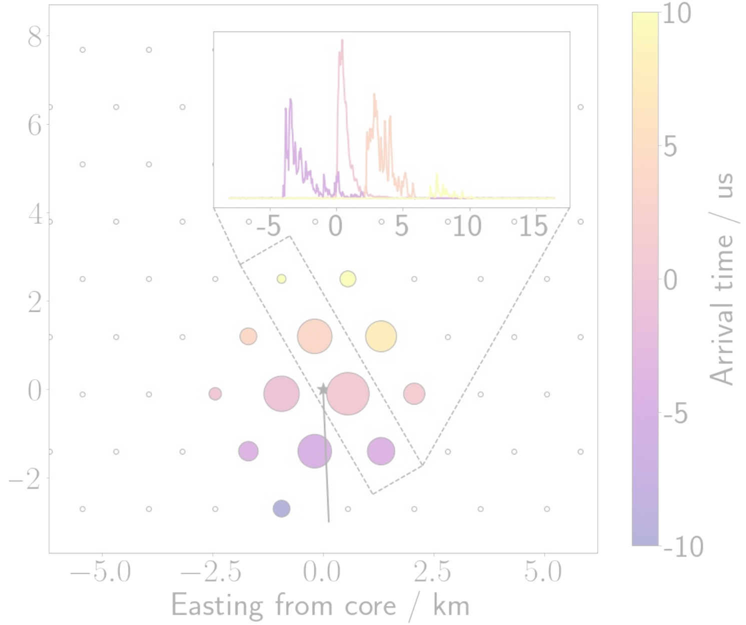
Rate | Physics

Raised at …station-level … CDAS level



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

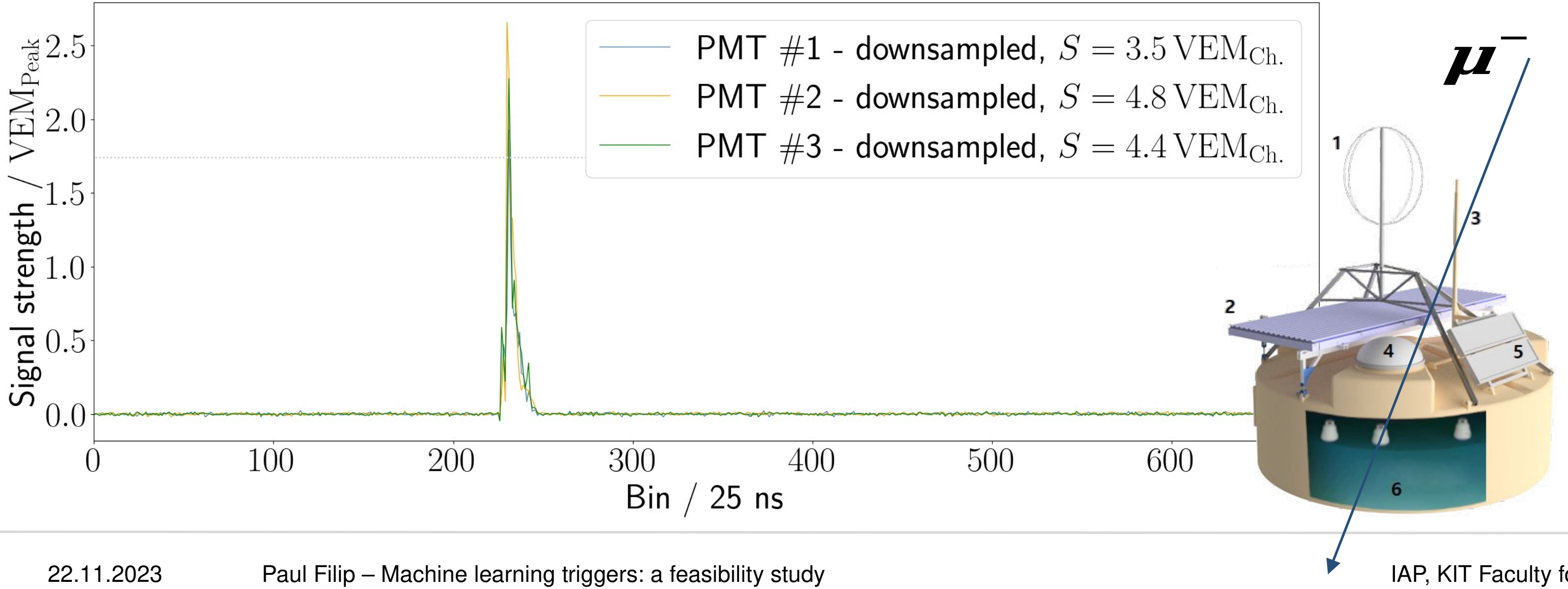


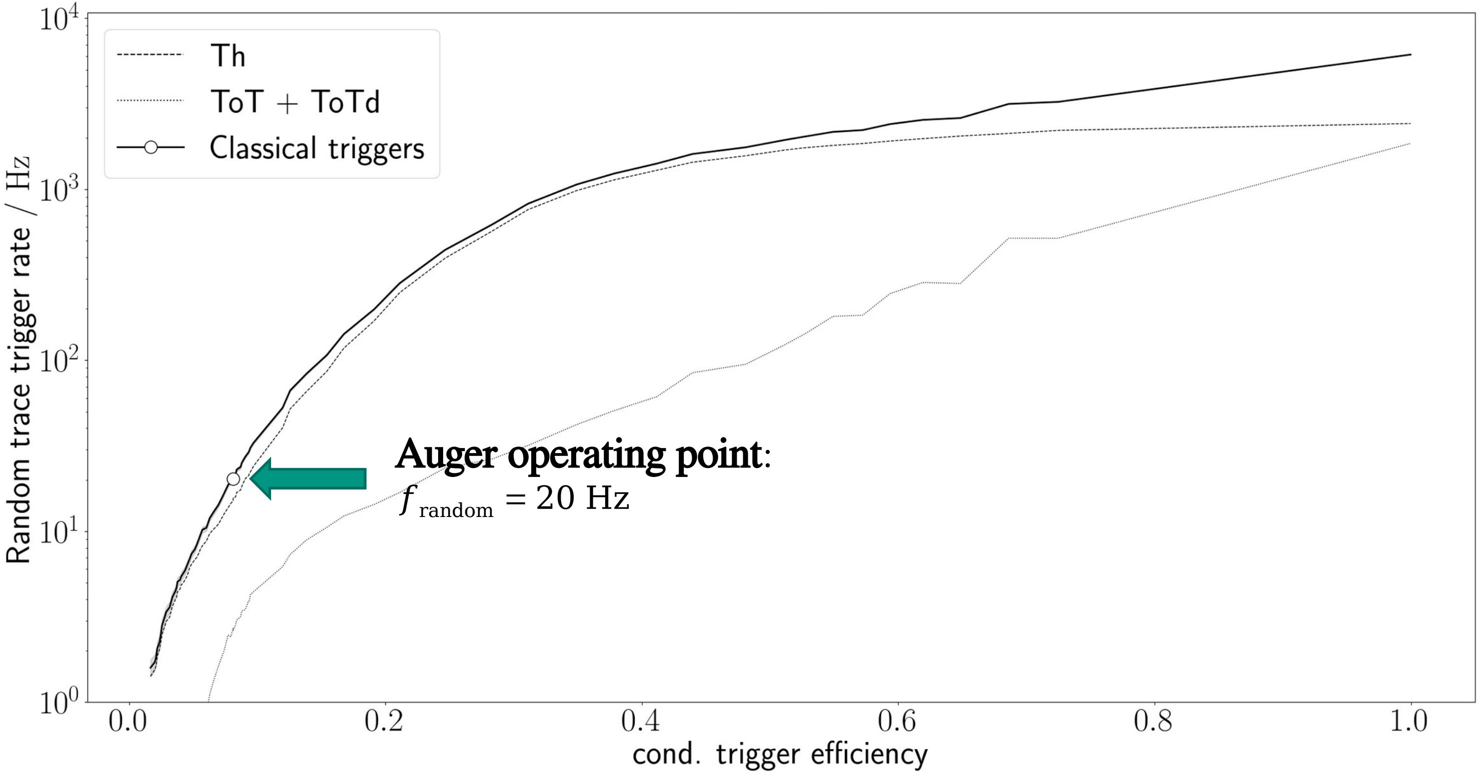

- **Threshold trigger (Th)**
- **Time over threshold (ToT) & ToT-like triggers**

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

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



# Trigger performance

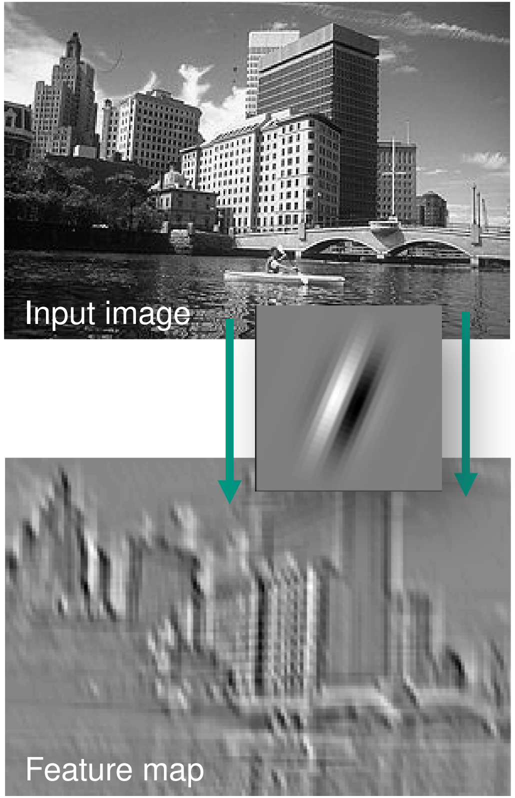



- Adjust thresholds
  - Better sensitivity
  - Worse specificity
  - What about SNR?

    gets way worse!
- Use ML triggers
  - Bayesian classifier
  - Neural networks
- Design limitation
  - Bandwidth limit
  - Performance limit
  - Storage limit

# Convolutional neural networks

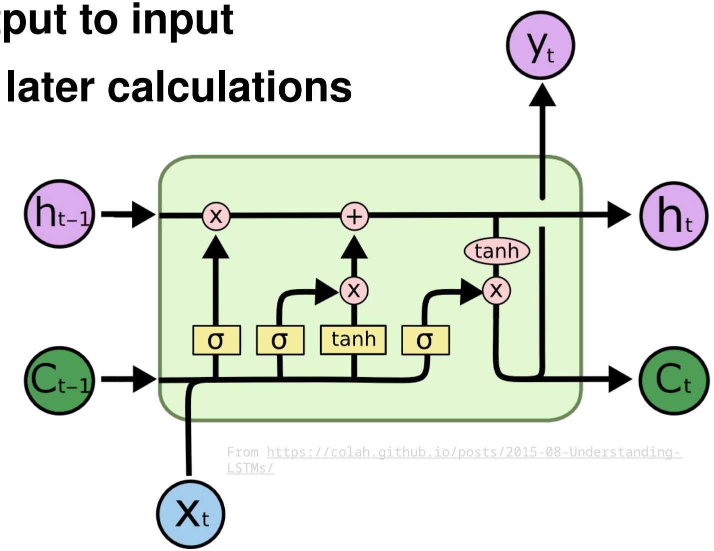
- **Specialized for image / object recognition**
  - **Different filters (matrices) scan parts of an image**
  - **Large output where filter and image look alike**
  - **Emergent object detection through multiple layers**
- **Treat WCD time traces as pictures**
  - **3 PMTs represent image height**
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




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

# Recurrent neural networks

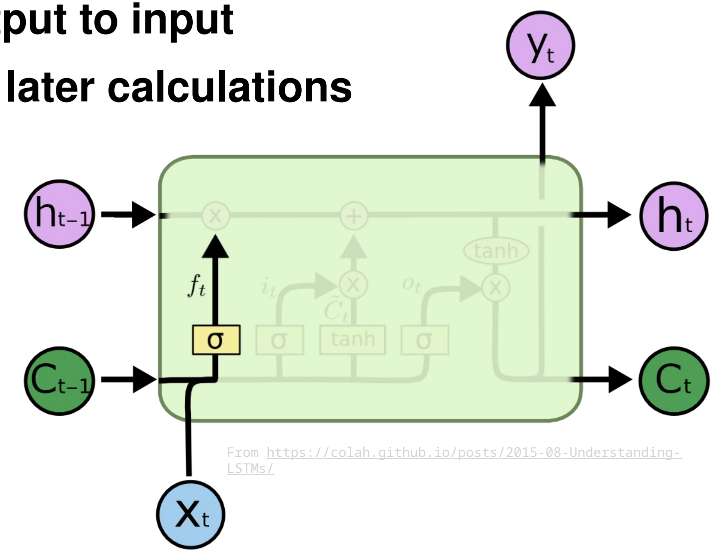
- **Long-Short-Term-Memory (LSTM) architecture**
  - **Has internal connections that point from output to input**
  - **Earlier processed information can influence later calculations**
  - **Treat time series very efficiently / elegantly**




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

# Recurrent neural networks

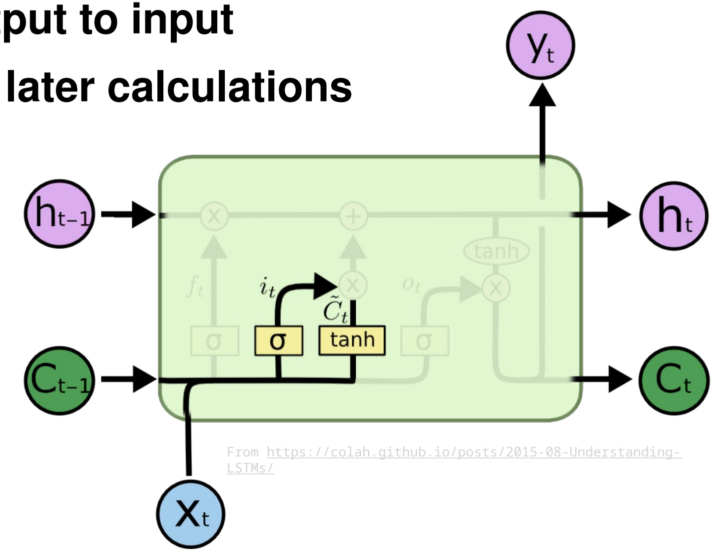
- **Long-Short-Term-Memory (LSTM) architecture**
  - **Has internal connections that point from output to input**
  - **Earlier processed information can influence later calculations**
  - **Treat time series very efficiently / elegantly**
- **Forget-Gate**
  - **What to keep from previous iterations**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

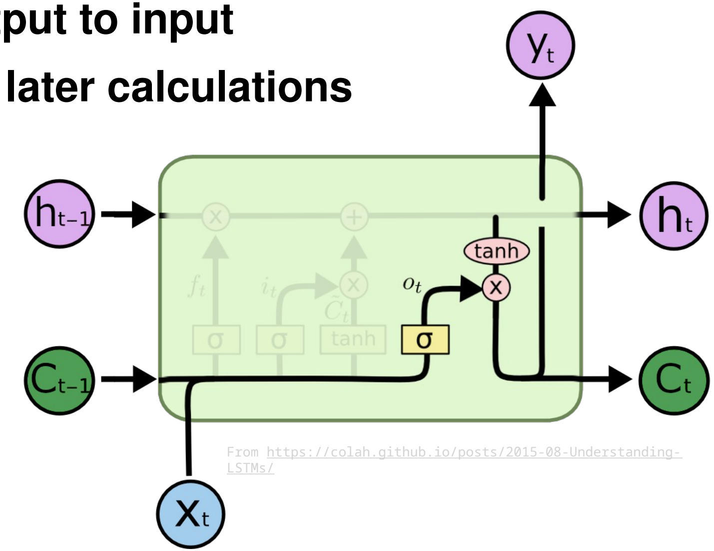
- **Long-Short-Term-Memory (LSTM) architecture**
  - **Has internal connections that point from output to input**
  - **Earlier processed information can influence later calculations**
  - **Treat time series very efficiently / elegantly**
- **Forget-Gate**
  - **What to keep from previous iterations**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



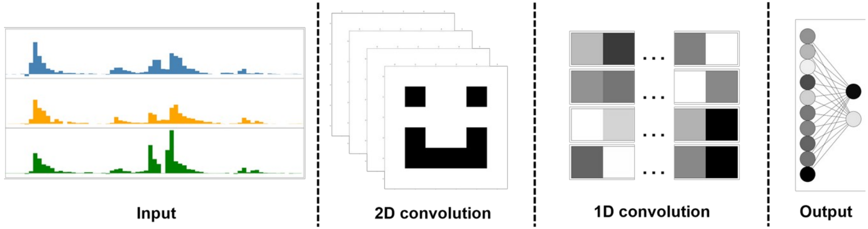

From https://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Network architectures

- 120 bins x 3 PMTs = 360 input values ⟹ 1 binary output (1 = Shower, 0 = Background)

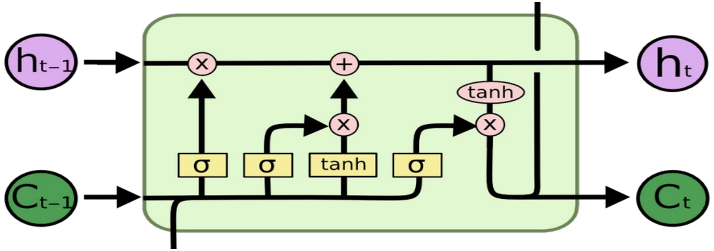
## Convolutional neural networks (CNNs)

- Good at recognizing objects in images
- Treat input data as 3x120 pixel image
- Output independent of signal position in window
- 1-2 convolutional layers with dense final layer
- 84 to 890 free trainable parameters



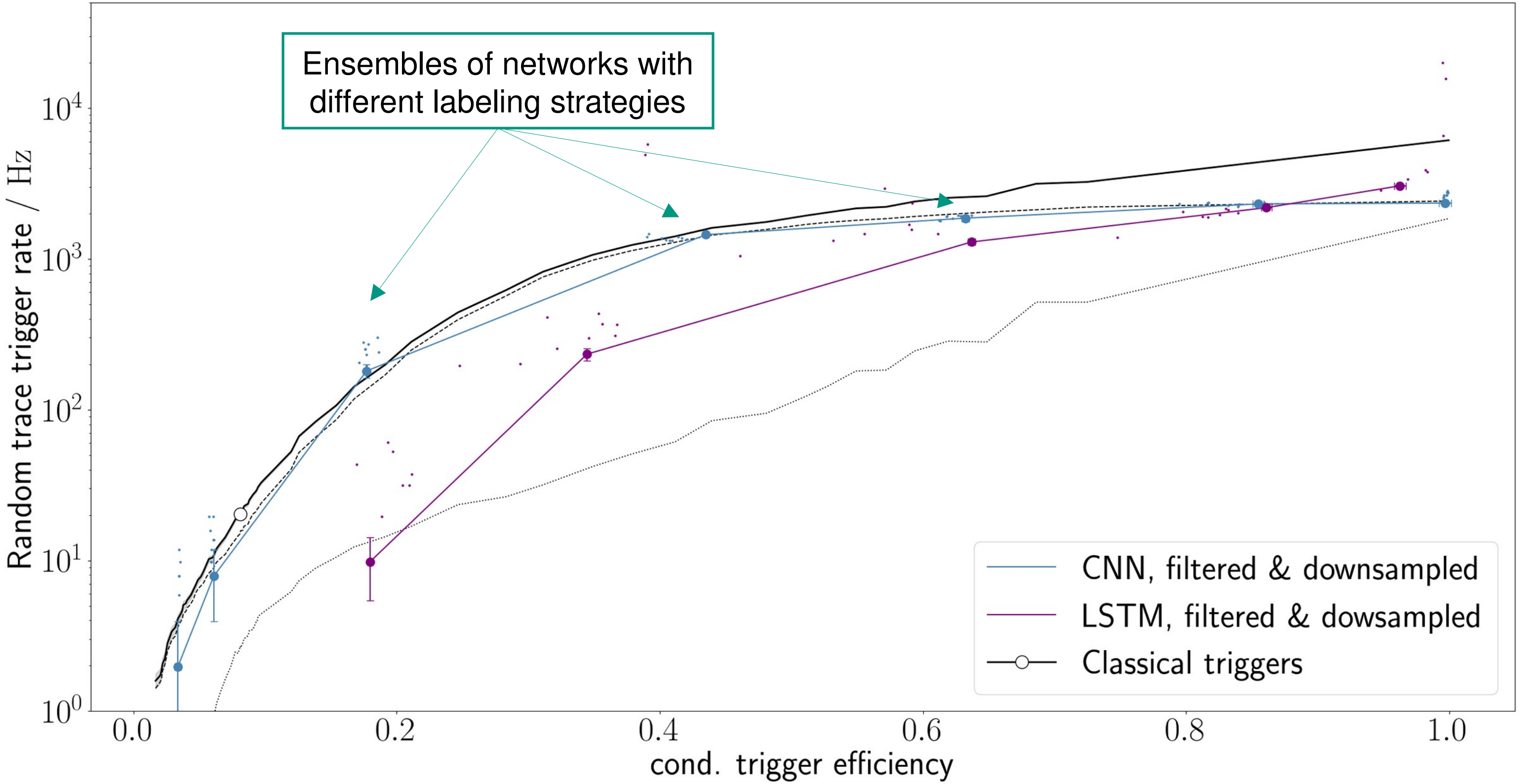

## Recurrent neural networks (LSTMs)

- Good at recognizing patterns sequential data
- Basic LSTM receives 1-dimensional input
- Implement 1 distinct LSTM for each PMT
- 12 to 44 free trainable parameters

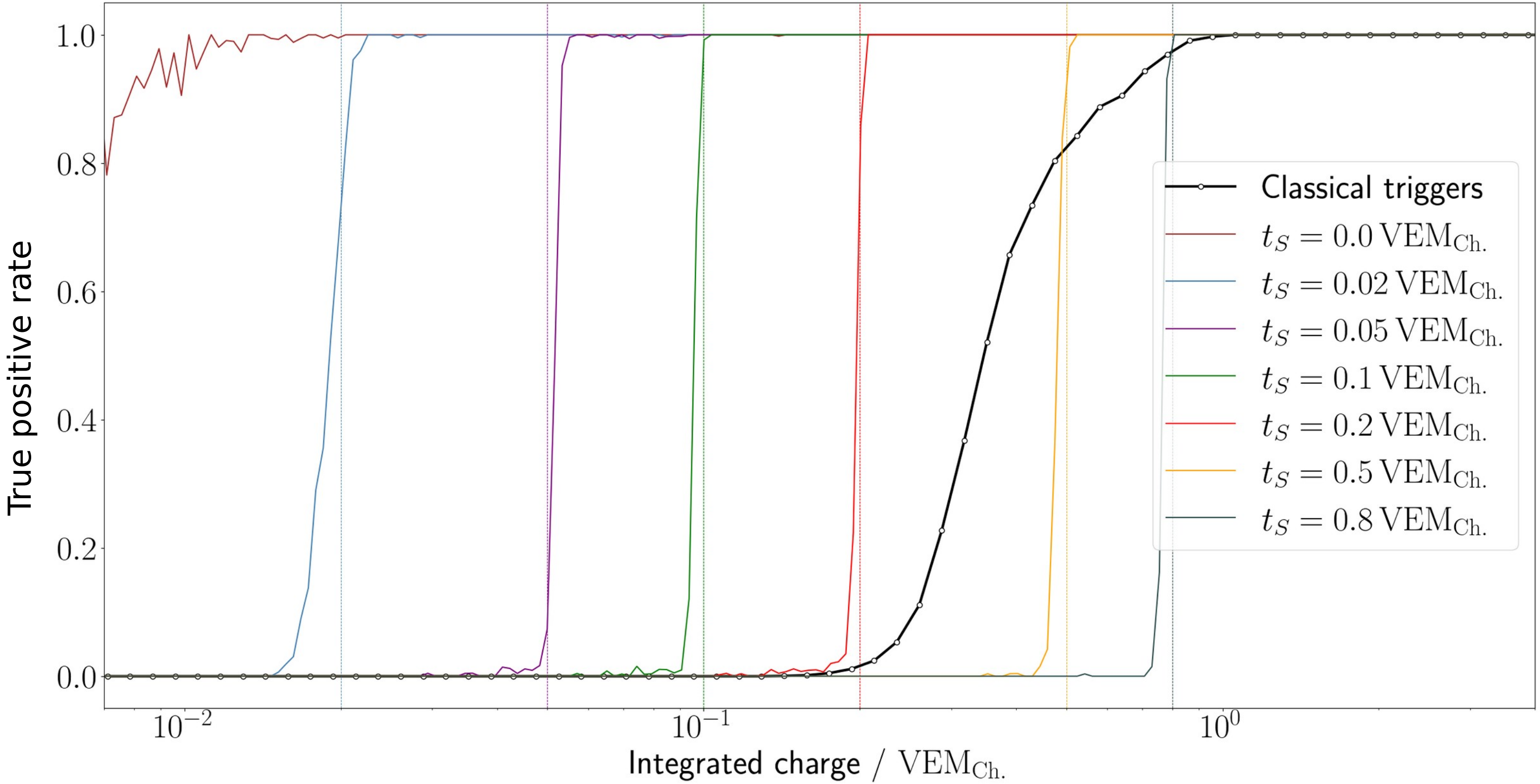



From https://colah.github.io/posts/2015-08-Understanding-LSTMs/

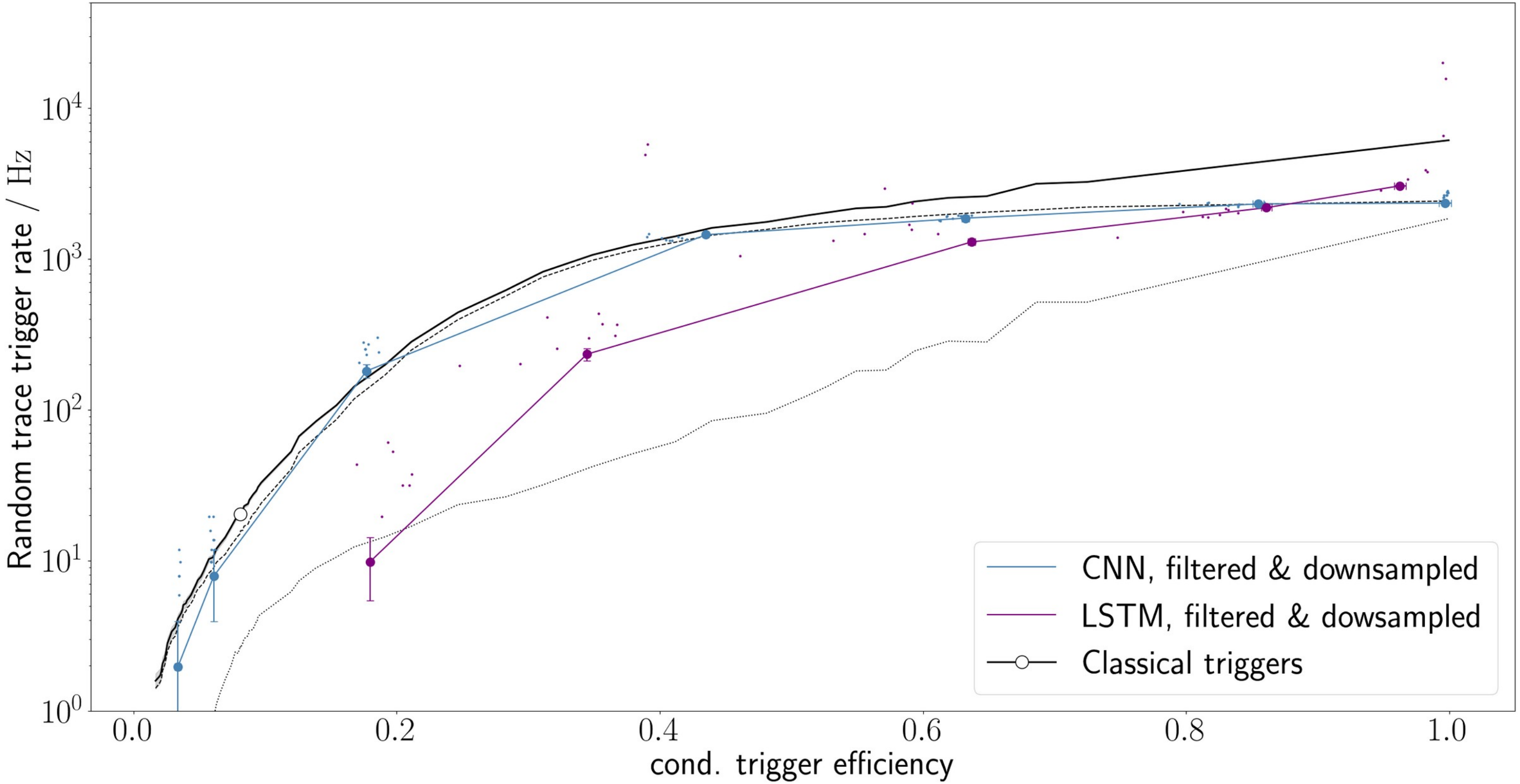
# Training results



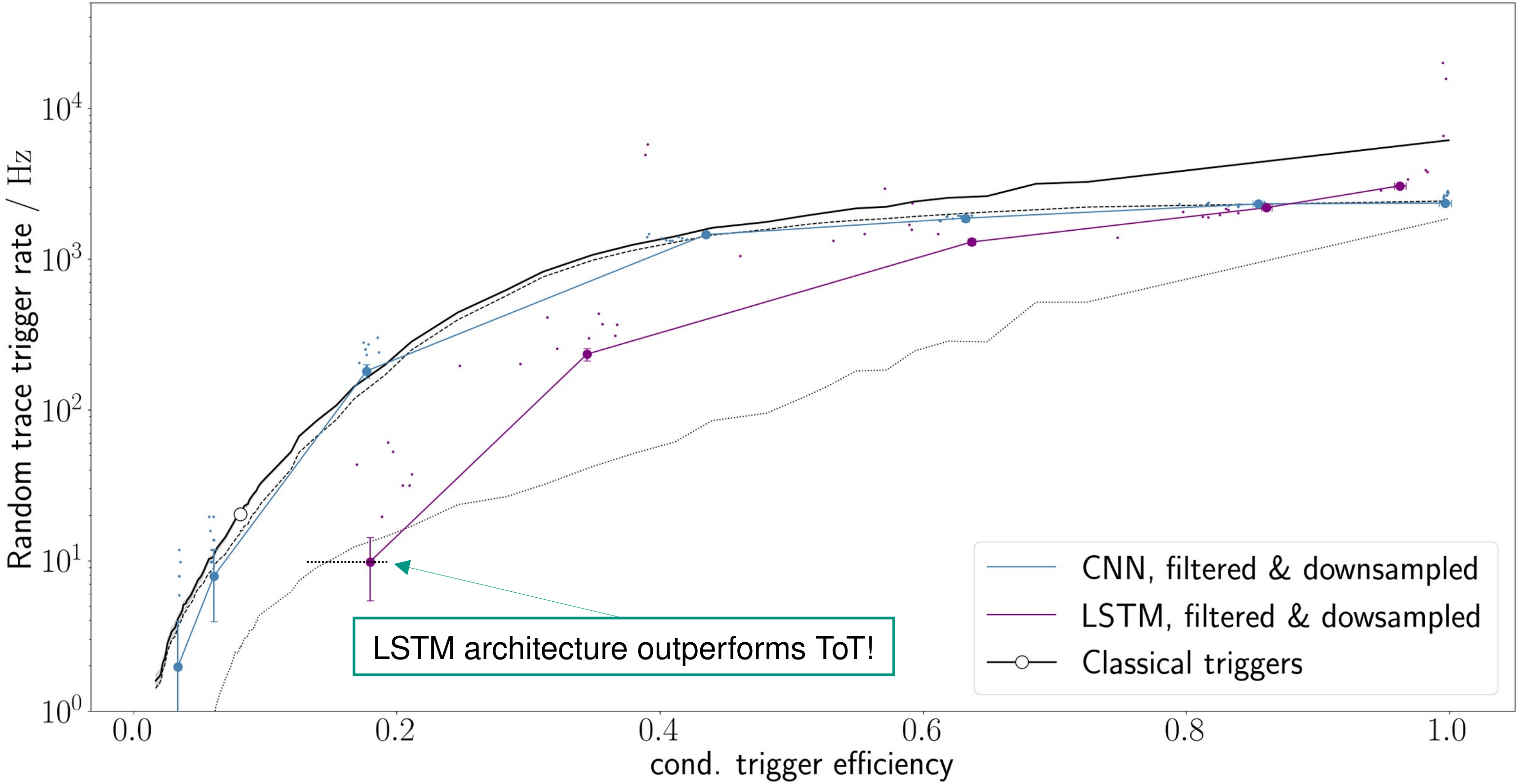
# Training results



# Training results

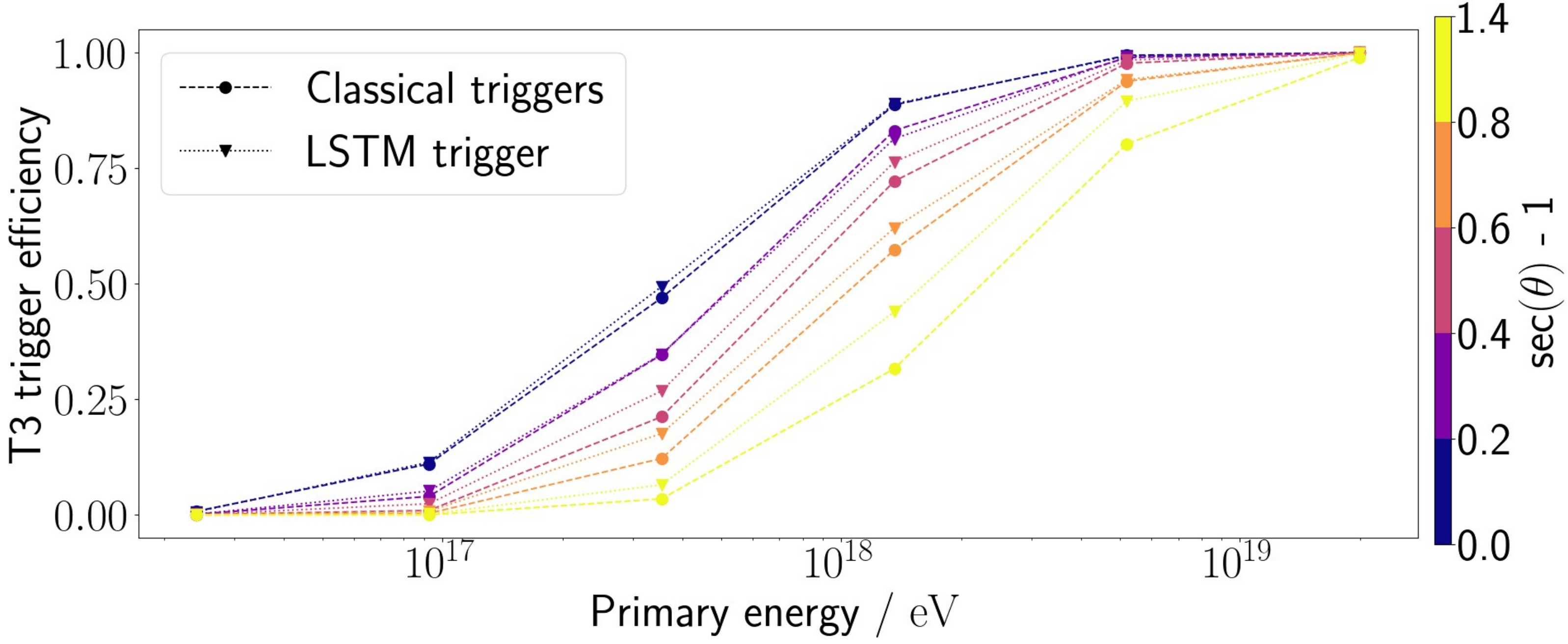


# Training results



# Resulting LSTM T3 efficiencies at $t_s = 0.5\,\mathrm{VEM}$

- **Most drastic gains at high inclinations**
- **Possibly higher gains at $\theta \geq 65^{\circ}$**



# Summary / Outlook

- Test data-driven, machine learning concepts
  - Analyse capability of shallow NNs (few parameters!) as SD T2-triggers
  - Consider convolutional (CNN) and simple recurrent neural networks (LSTM)
  - Verify performance of NNs with measured background data
  - Control trigger rate by implementing charge cut

- Convolutional neural networks
  - Performance of simple CNN architectures on par with Th-Trigger
  - CNN architecture has worse performance than ToT-trigger
  - Filtered & downsampled data preferred over full bandwidth input

- LSTM / recurrent neural networks
  - First results indicate performance on par with or better than ToT
  - Gains in event detection efficiency at high shower angles

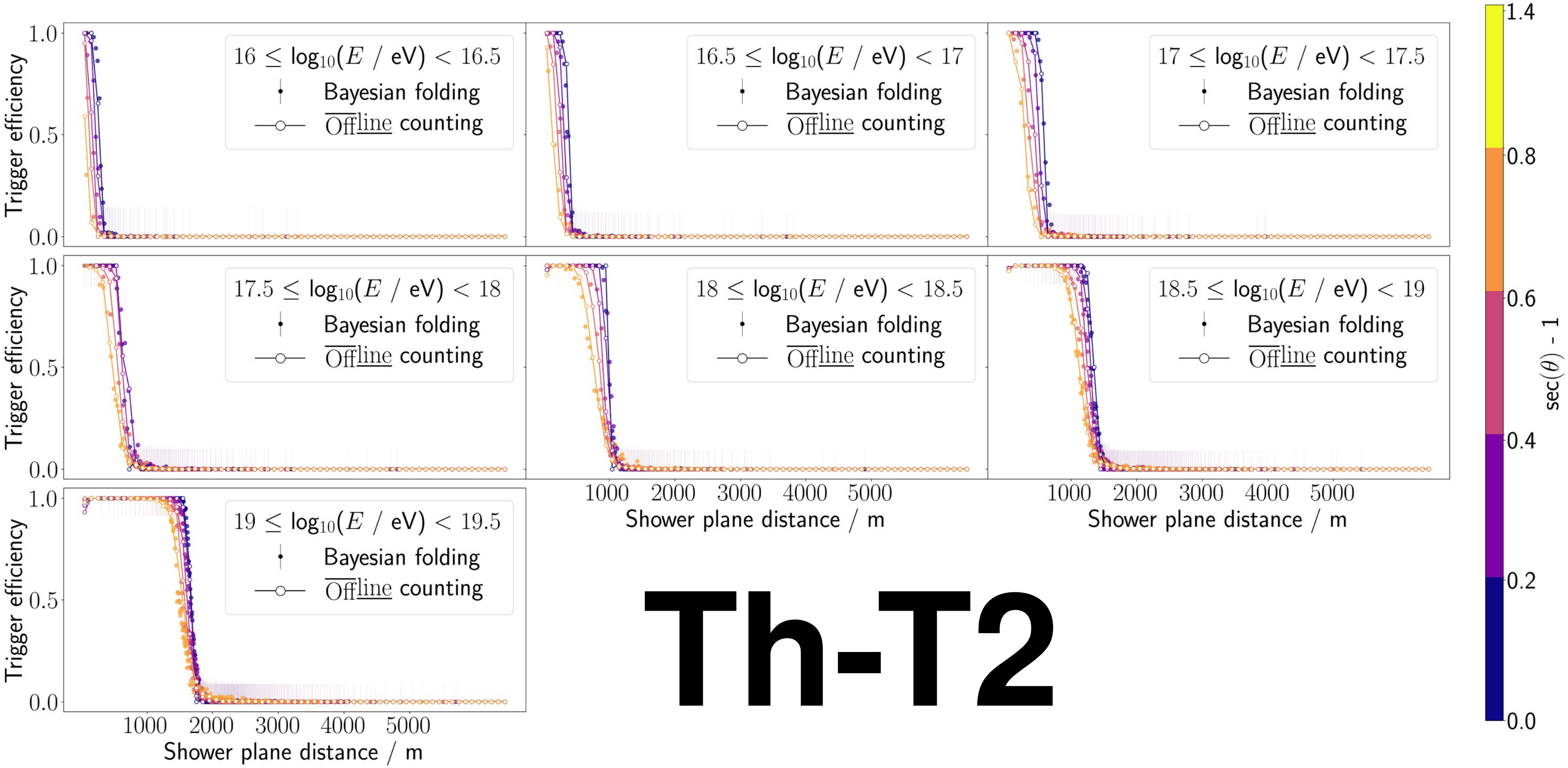
# Summary / Outlook

- **Lot of work needed before prototyping stage**
  - **Presented results in signal detection efficiency stem from simulations only**
  - **No primary distinction, only data from protons considered**
  - **Only one hadronic interaction model (QGSJET-II.04)**
  - **No steps taken yet to implement this on our FPGAs (feasibility?)**
- **Ground work is completed**
  - **Bayesian folding simplifies analysis and yields the same results as Offline**
  - **Analysis chain is implemented and ready to run**
  - **Neural networks show potential for SD T2-triggers**

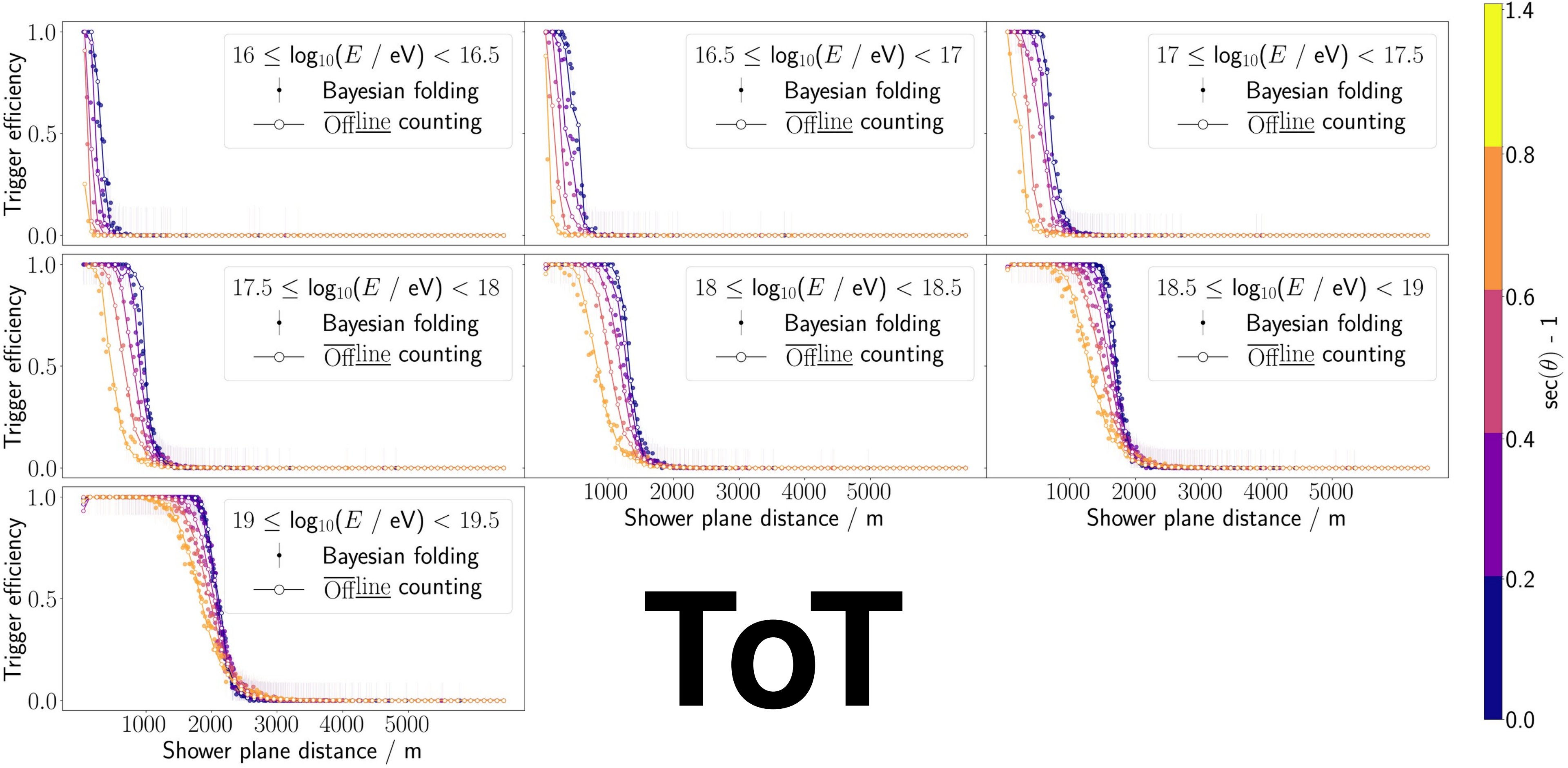
Karlsruhe Institute of Technology

PIERRE AUGER OBSERVATORY

# Backup

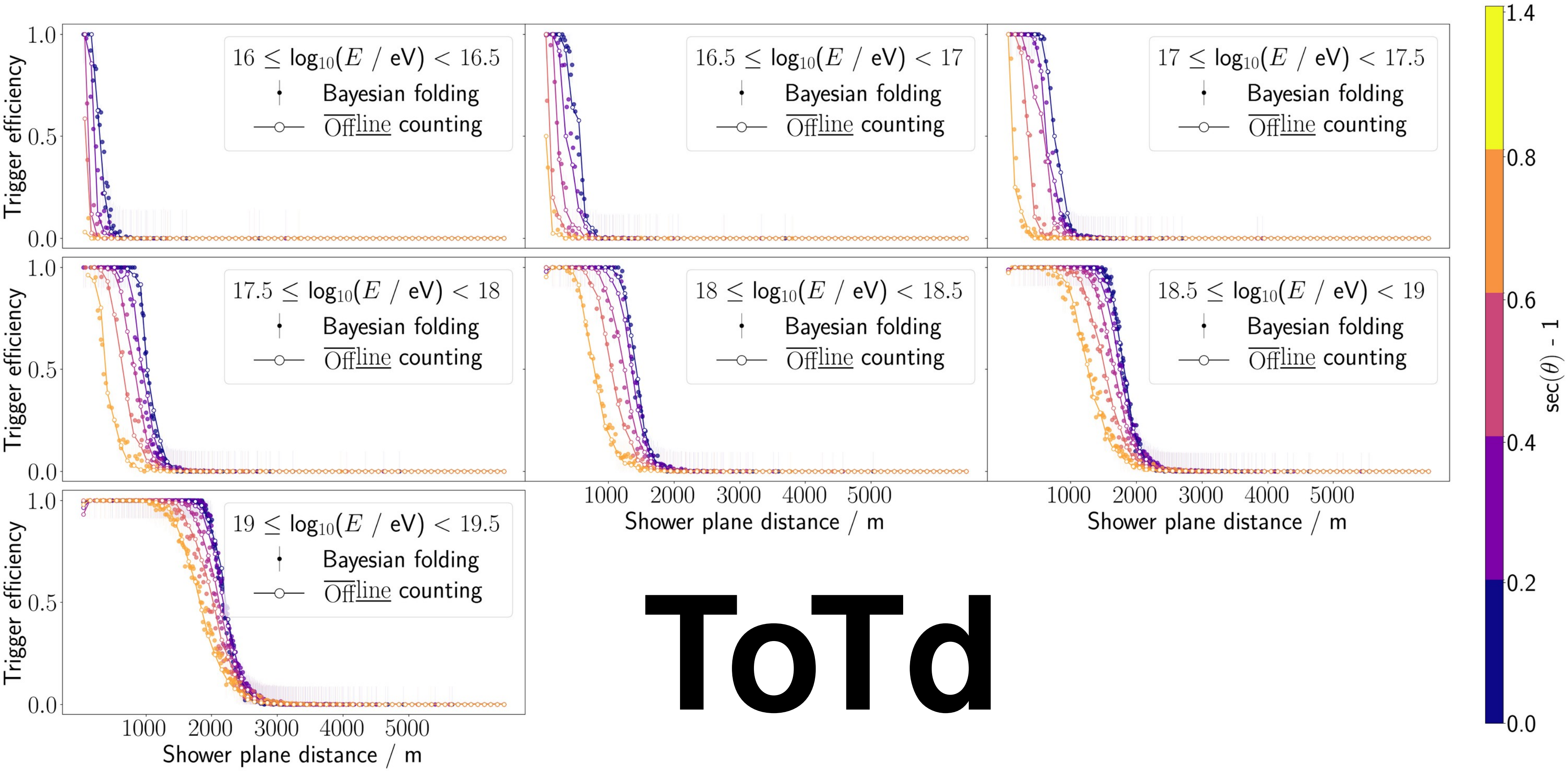

16 ≤ log10(E / eV) < 16.5
16.5 ≤ log10(E / eV) < 17
17 ≤ log10(E / eV) < 17.5
17.5 ≤ log10(E / eV) < 18
18 ≤ log10(E / eV) < 18.5
18.5 ≤ log10(E / eV) < 19
19 ≤ log10(E / eV) < 19.5
Bayesian folding
Offline counting
Trigger efficiency
Shower plane distance / m
sec(θ) - 1

# Th-T2



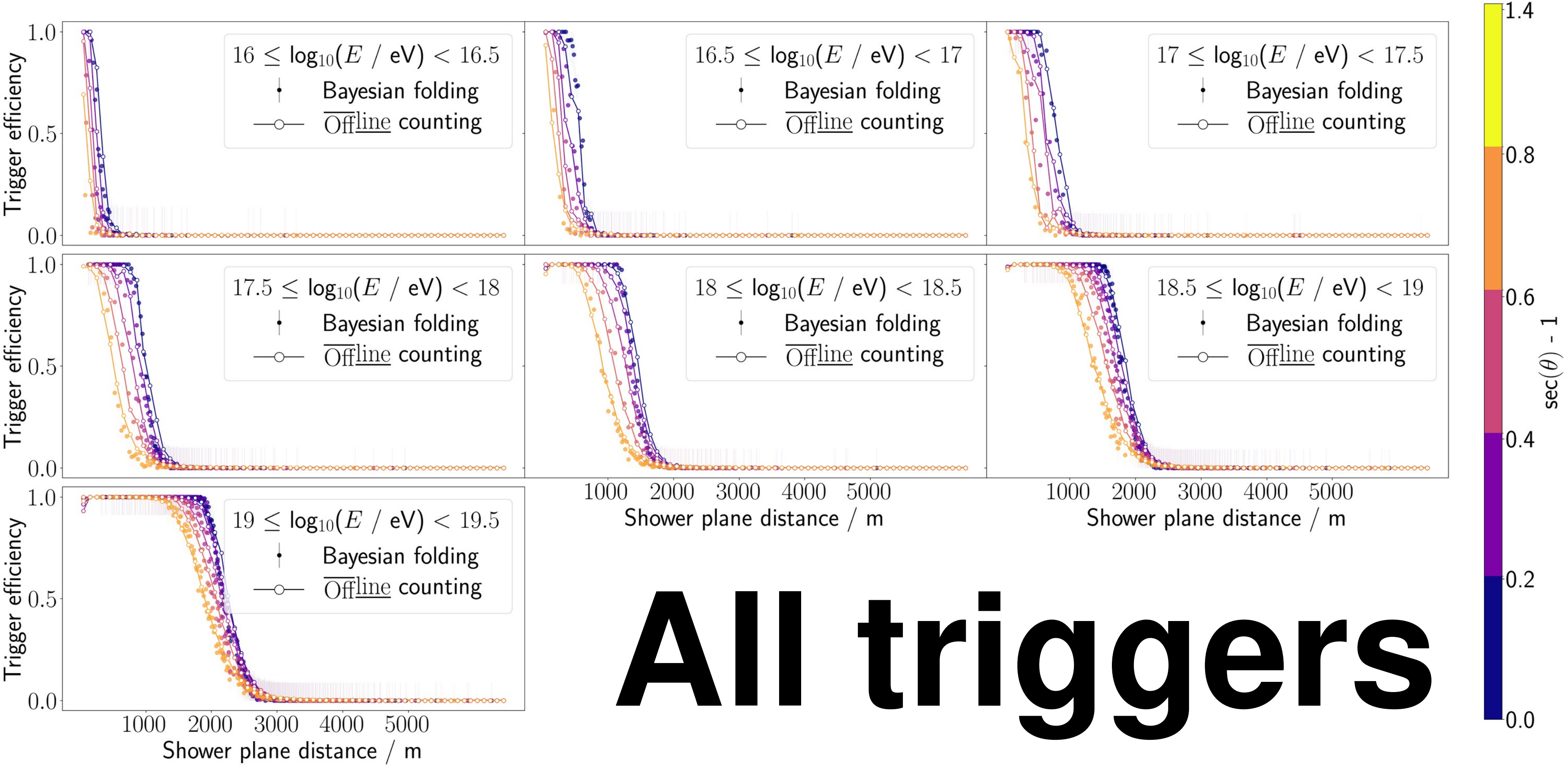
Trigger efficiency
1.0
0.5
0.0
16 ≤ log10(E / eV) < 16.5
16.5 ≤ log10(E / eV) < 17
17 ≤ log10(E / eV) < 17.5
17.5 ≤ log10(E / eV) < 18
18 ≤ log10(E / eV) < 18.5
18.5 ≤ log10(E / eV) < 19
19 ≤ log10(E / eV) < 19.5
Bayesian folding
Offline counting
1000
2000
3000
4000
5000
Shower plane distance / m
sec(θ) - 1
0.0
0.2
0.4
0.6
0.8
1.4


# ToT

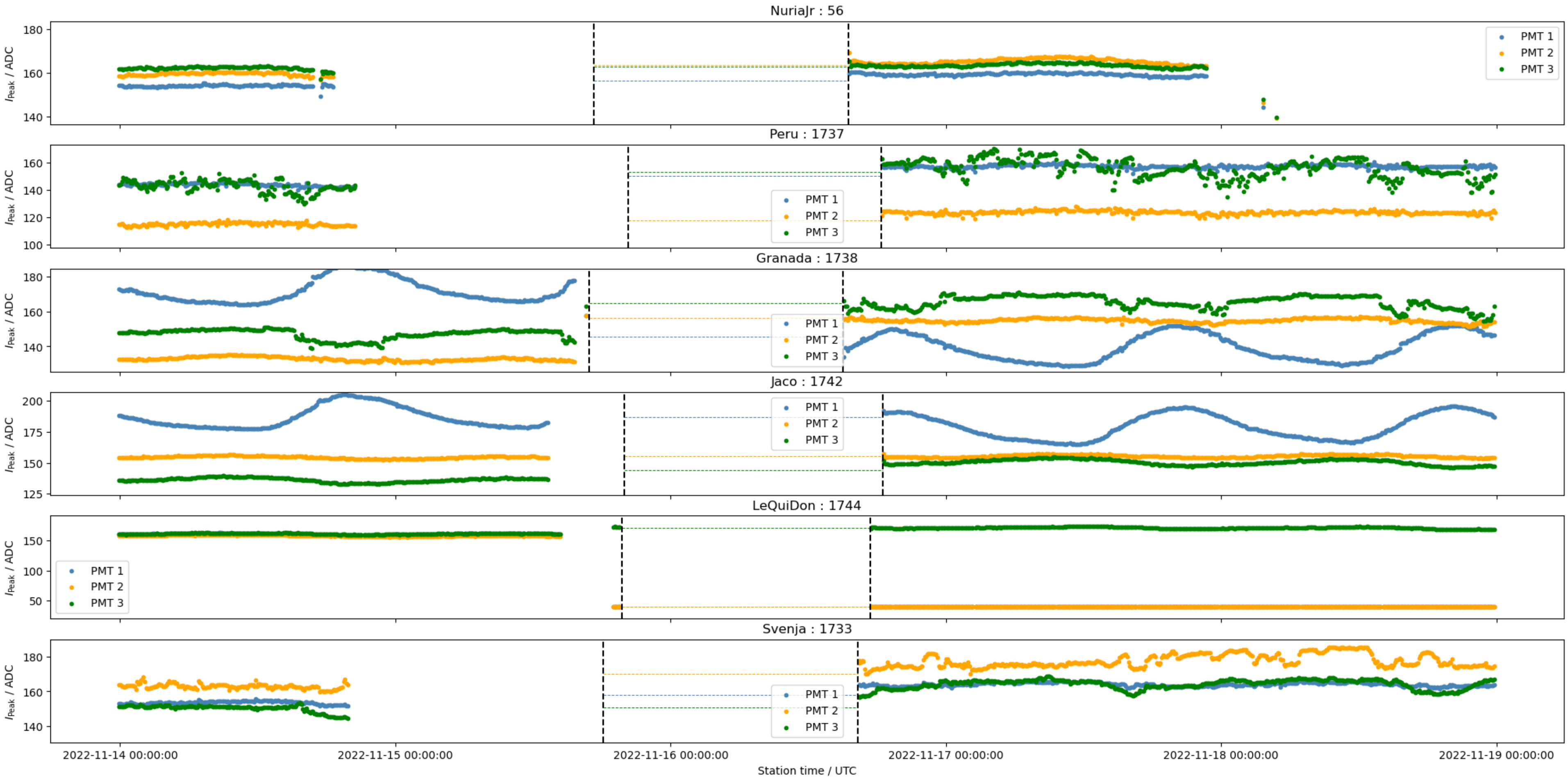


$16 \leq \log_{10}(E$ / eV$) < 16.5$
Bayesian folding
$\overline{\text{Off}}\underline{\text{line}}$ counting
$16.5 \leq \log_{10}(E$ / eV$) < 17$
$17 \leq \log_{10}(E$ / eV$) < 17.5$
$17.5 \leq \log_{10}(E$ / eV$) < 18$
$18 \leq \log_{10}(E$ / eV$) < 18.5$
$18.5 \leq \log_{10}(E$ / eV$) < 19$
$19 \leq \log_{10}(E$ / eV$) < 19.5$
Trigger efficiency
Shower plane distance / m
$\sec(\theta)$ - 1

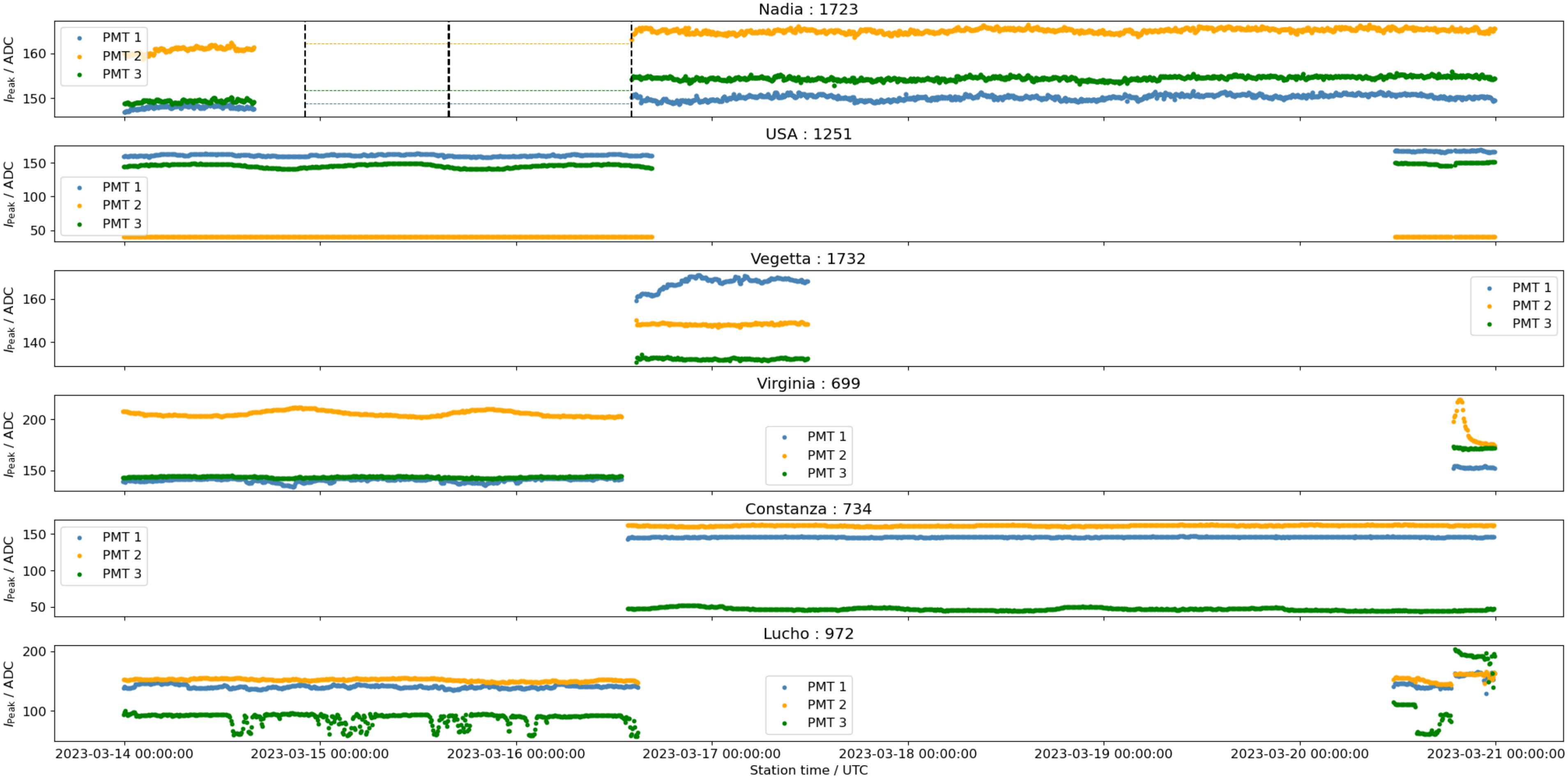

# ToTd



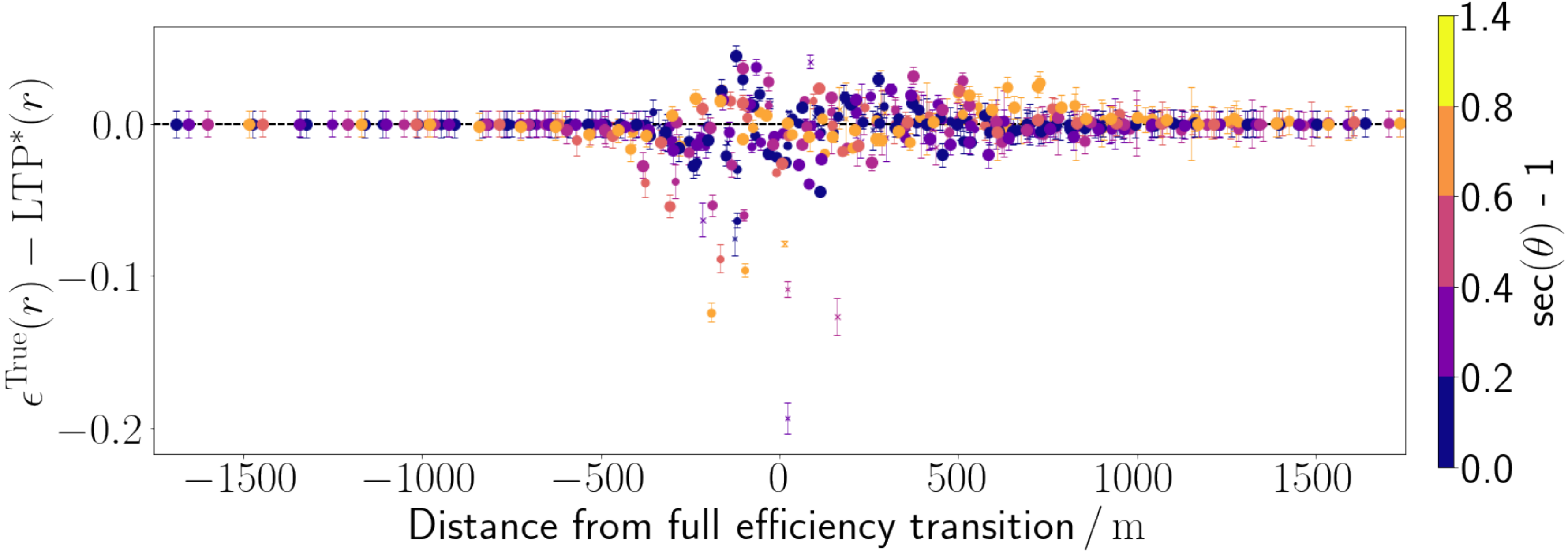

# All triggers



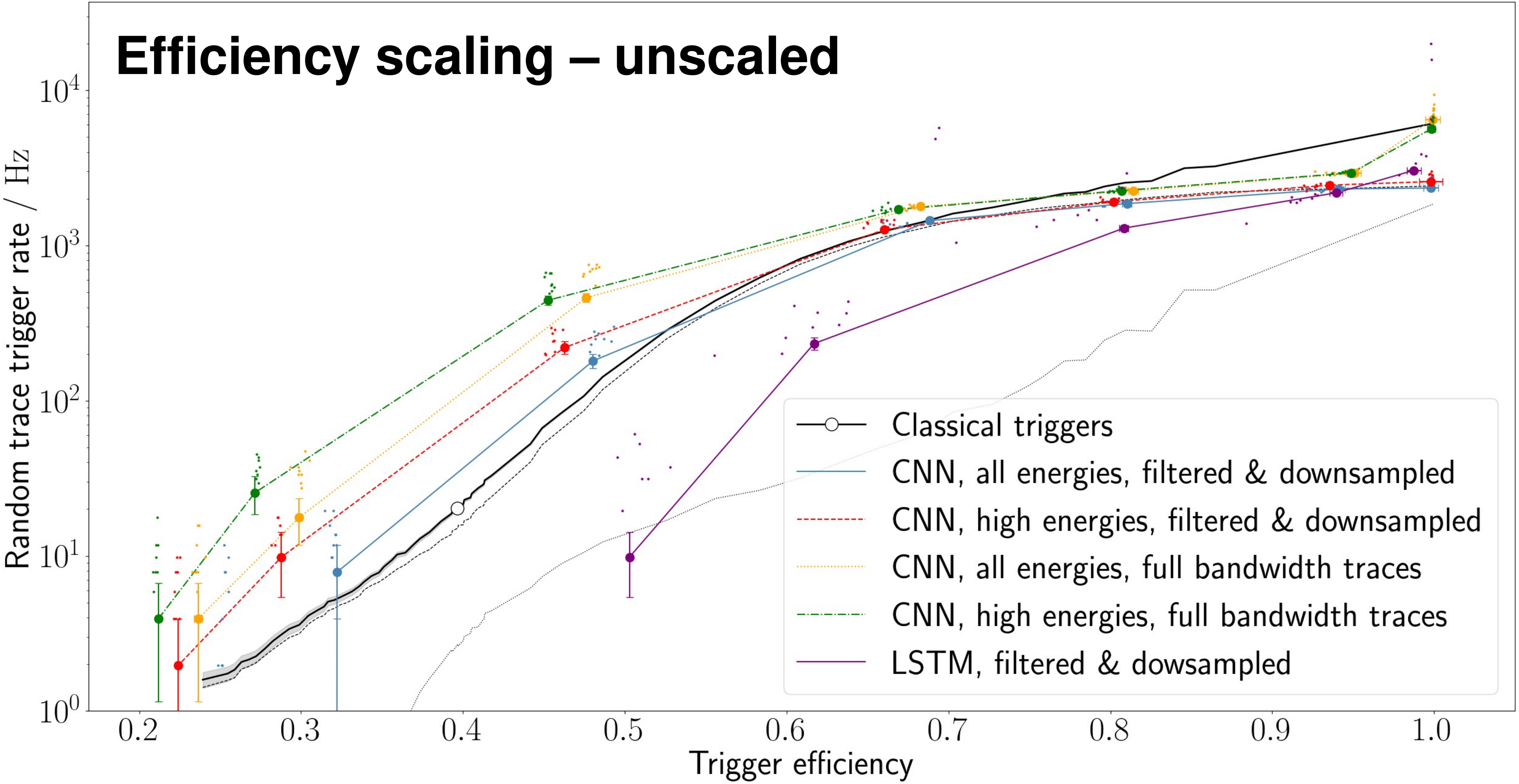
NuriaJr : 56
Peru : 1737
Granada : 1738
Jaco : 1742
LeQuiDon : 1744
Svenja : 1733
PMT 1
PMT 2
PMT 3
$I_{Peak}$ / ADC
2022-11-14 00:00:00
2022-11-15 00:00:00
2022-11-16 00:00:00
2022-11-17 00:00:00
2022-11-18 00:00:00
2022-11-19 00:00:00
Station time / UTC


Nadia : 1723
USA : 1251
Vegetta : 1732
Virginia : 699
Constanza : 734
Lucho : 972
PMT 1
PMT 2
PMT 3
$I_{Peak}$ / ADC
Station time / UTC
2023-03-14 00:00:00
2023-03-15 00:00:00
2023-03-16 00:00:00
2023-03-17 00:00:00
2023-03-18 00:00:00
2023-03-19 00:00:00
2023-03-20 00:00:00
2023-03-21 00:00:00

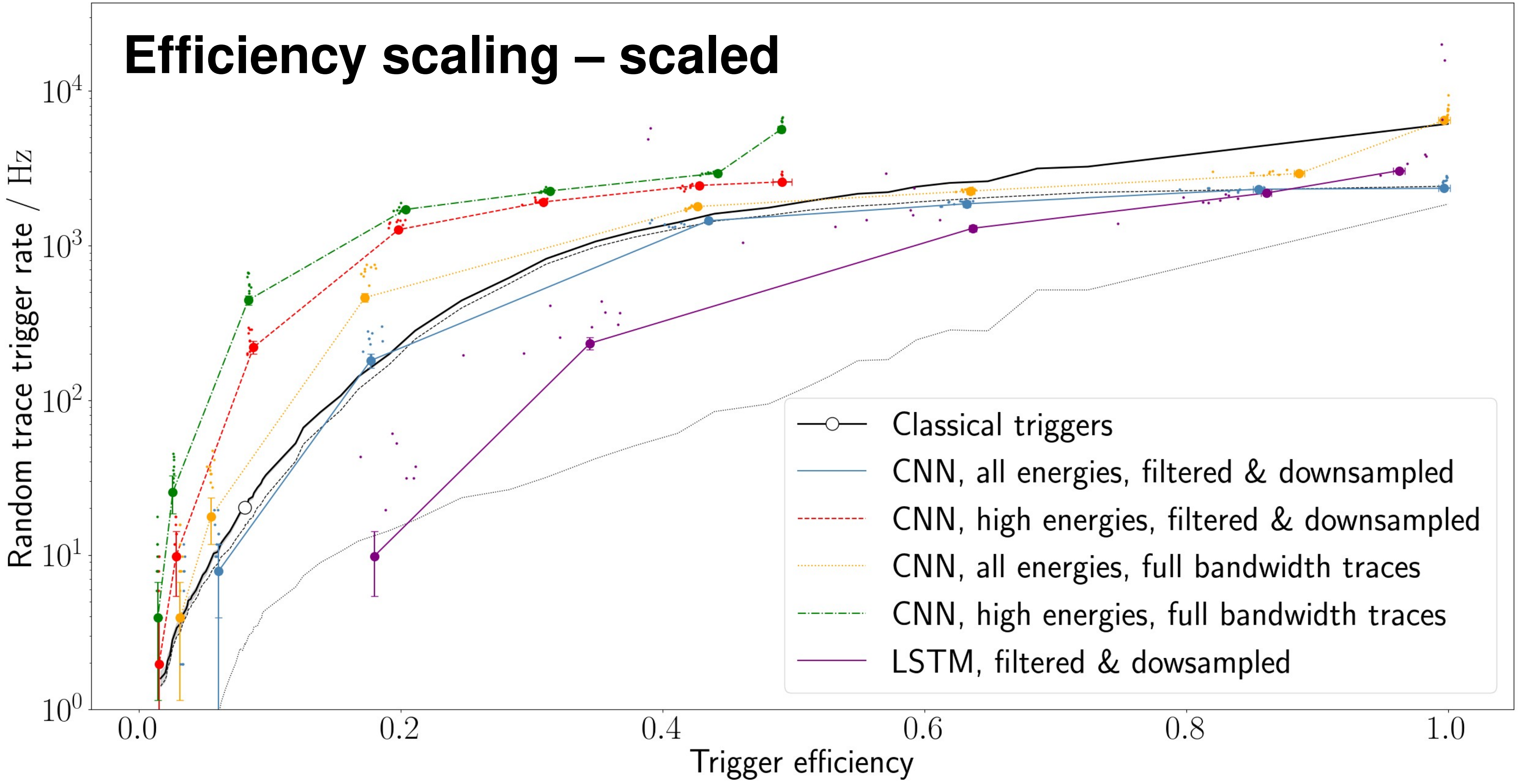
# Residuals – LTP fitfunction



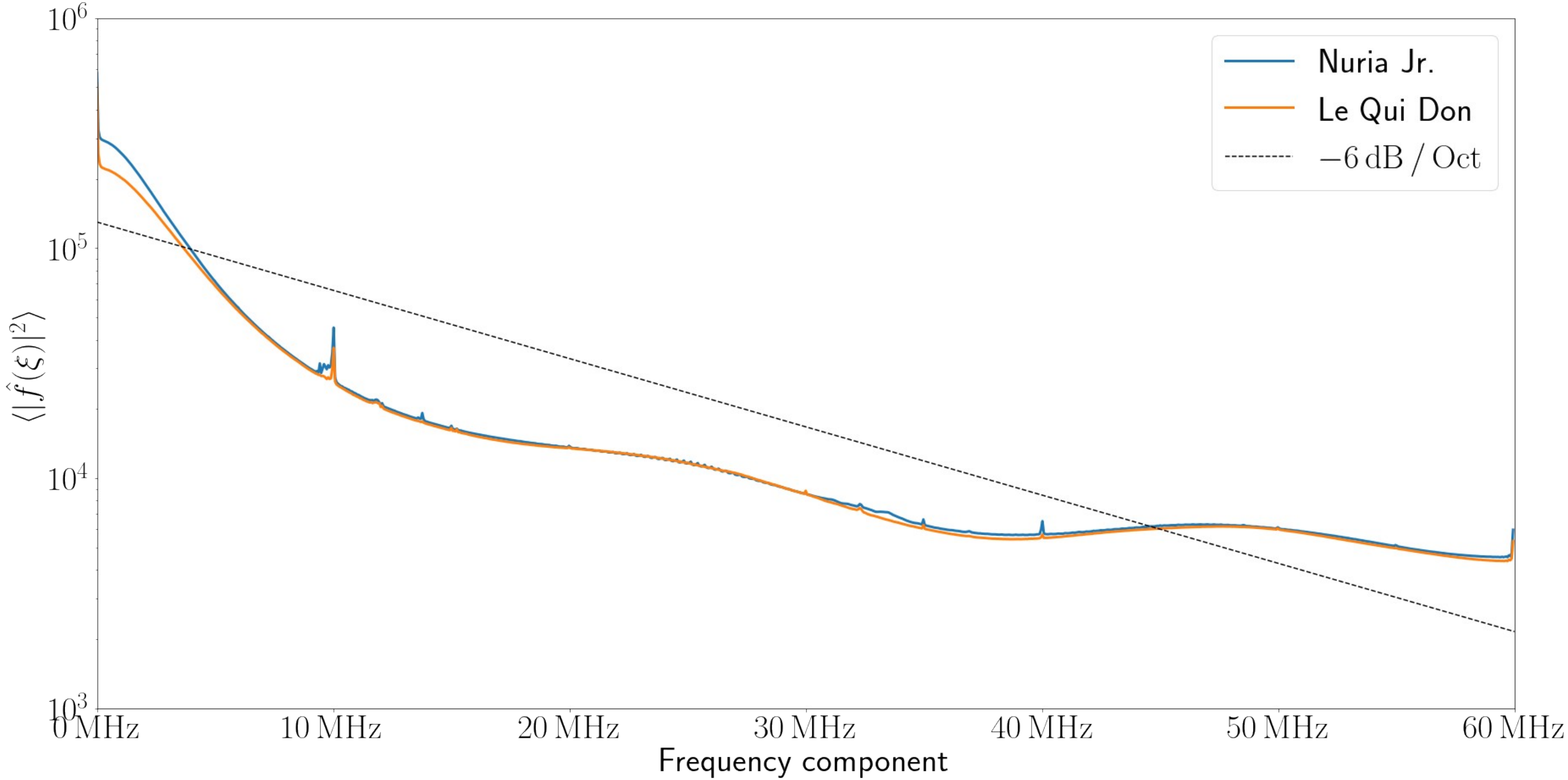
# Efficiency scaling – unscaled



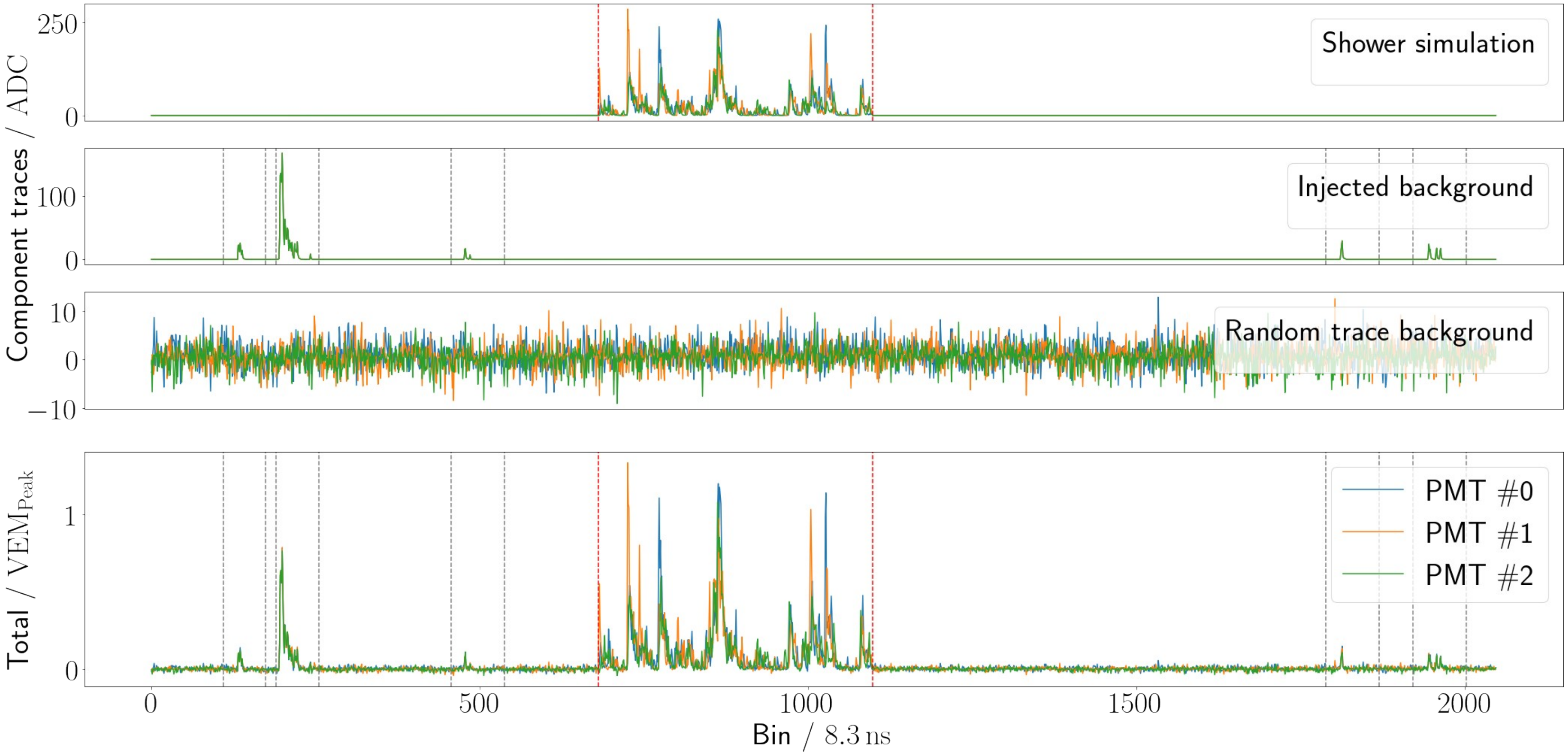
# Efficiency scaling – scaled



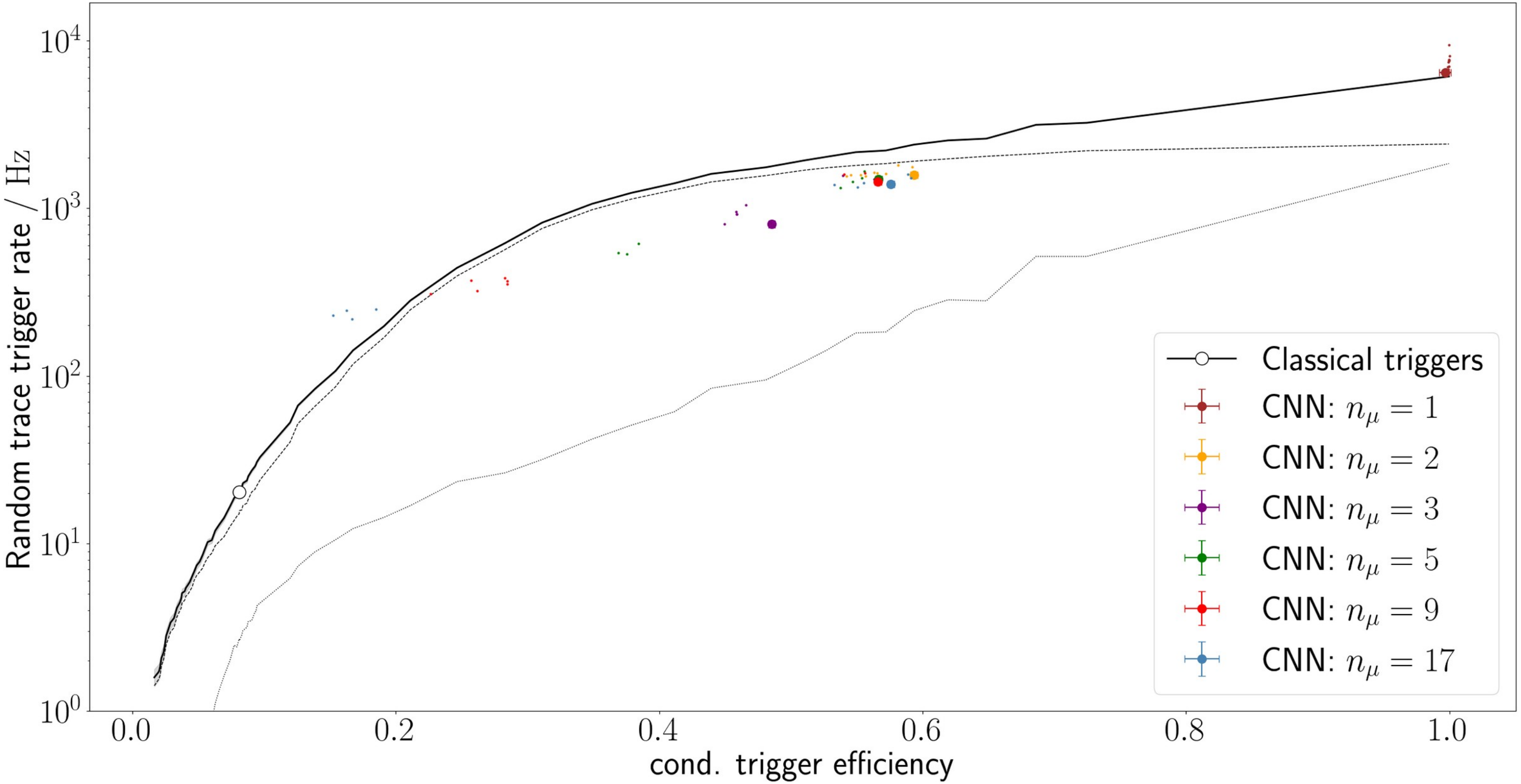
# Random traces – Power spectrum



# Trace building

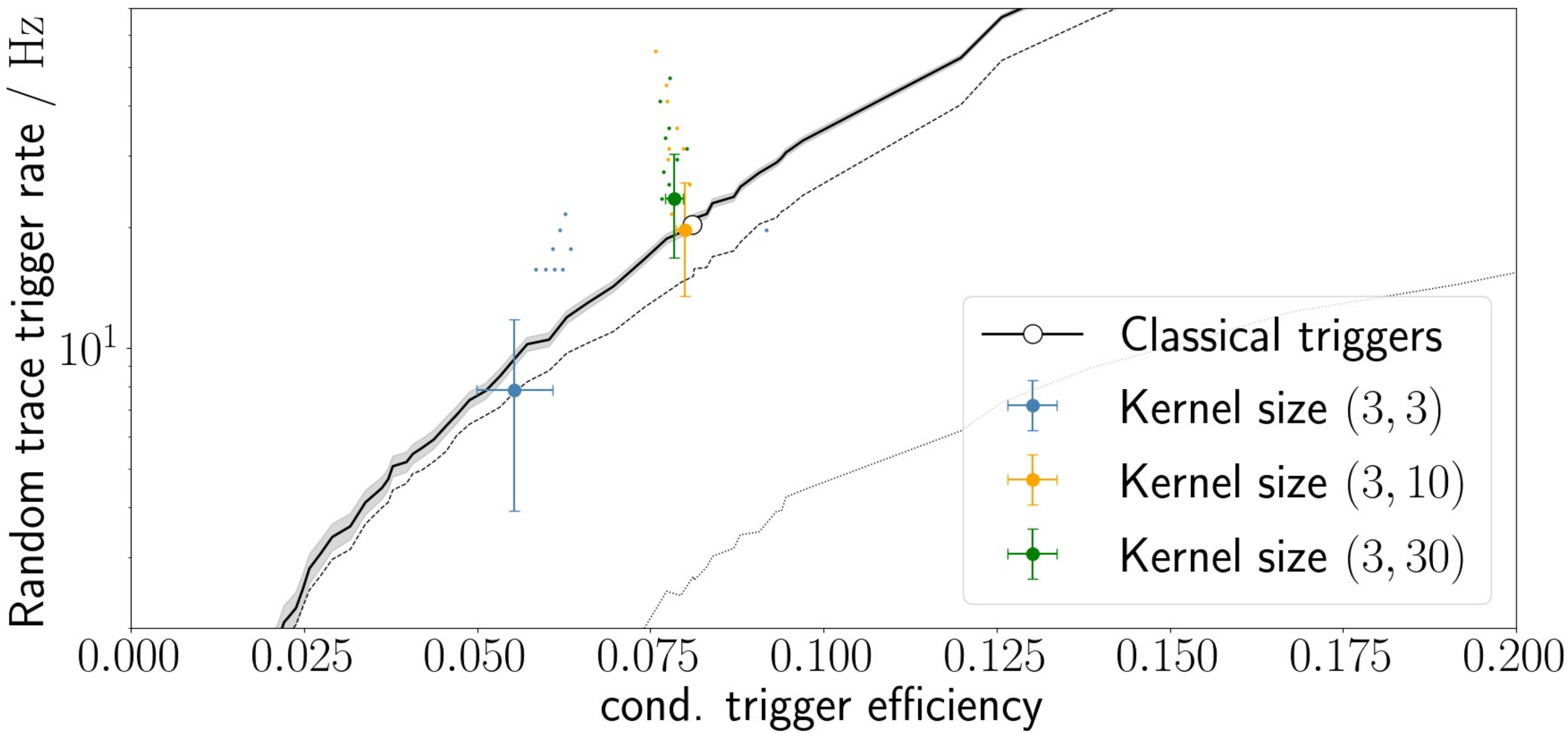


# Muon cut



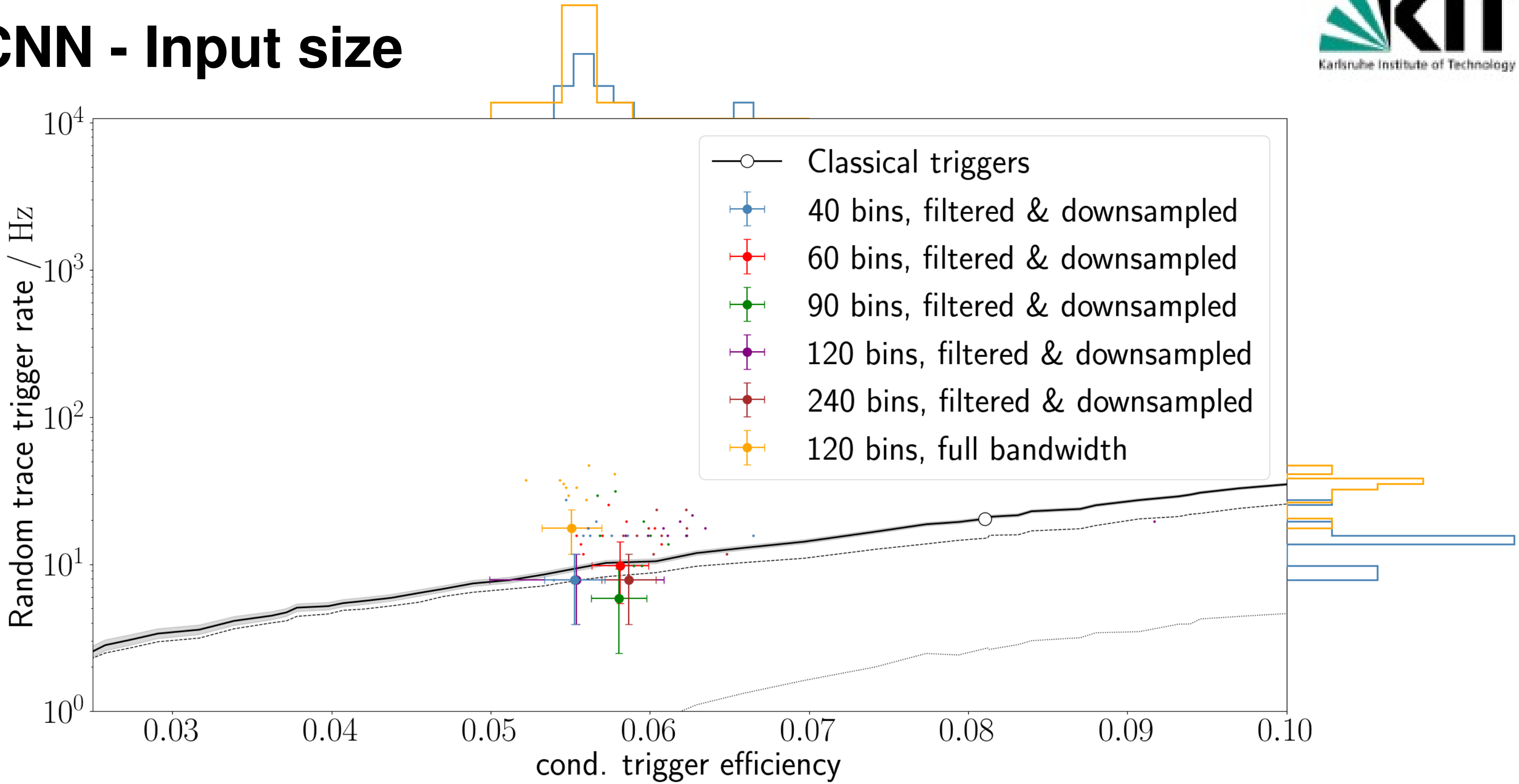
Random trace trigger rate / Hz
$10^4$
$10^3$
$10^2$
$10^1$
$10^0$
0.0
0.2
0.4
0.6
0.8
1.0
cond. trigger efficiency
Classical triggers
CNN: $n_\mu = 1$
CNN: $n_\mu = 2$
CNN: $n_\mu = 3$
CNN: $n_\mu = 5$
CNN: $n_\mu = 9$
CNN: $n_\mu = 17$

# Kernel size

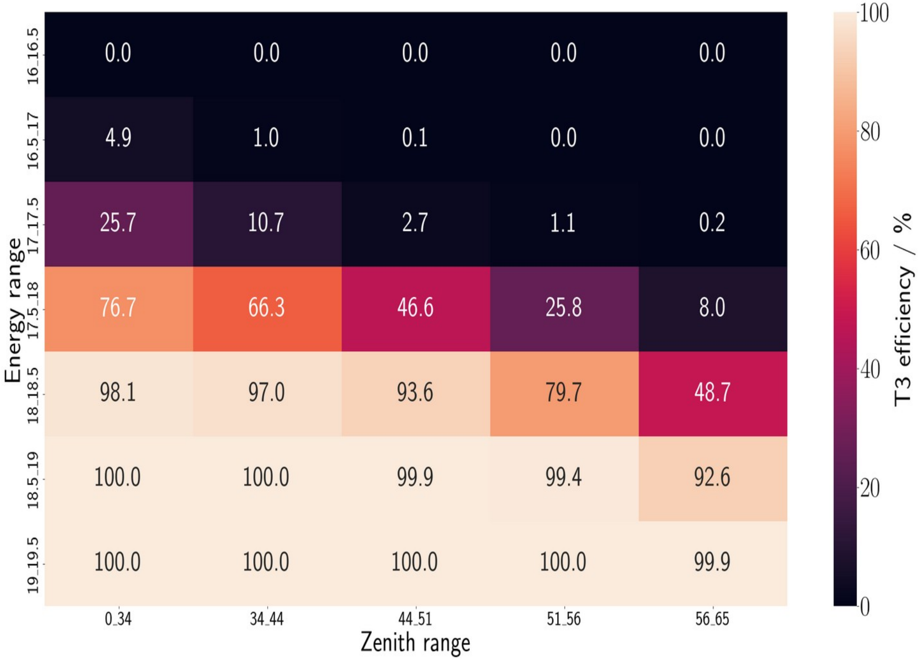


Random trace trigger rate / Hz
$10^1$
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
cond. trigger efficiency
Classical triggers
Kernel size $(3, 3)$
Kernel size $(3, 10)$
Kernel size $(3, 30)$

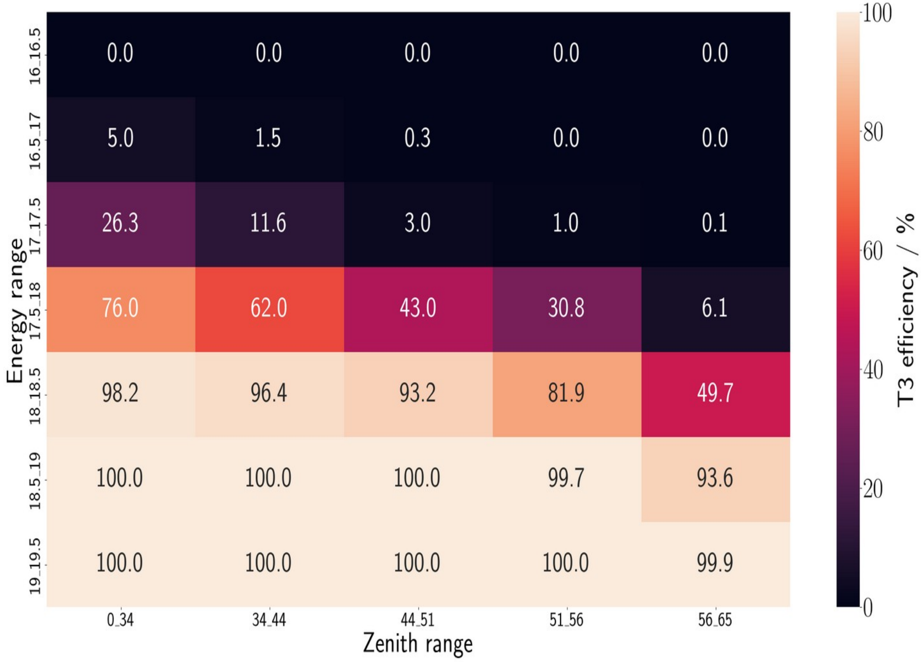
# CNN - Input size



# T3 efficiency calculation

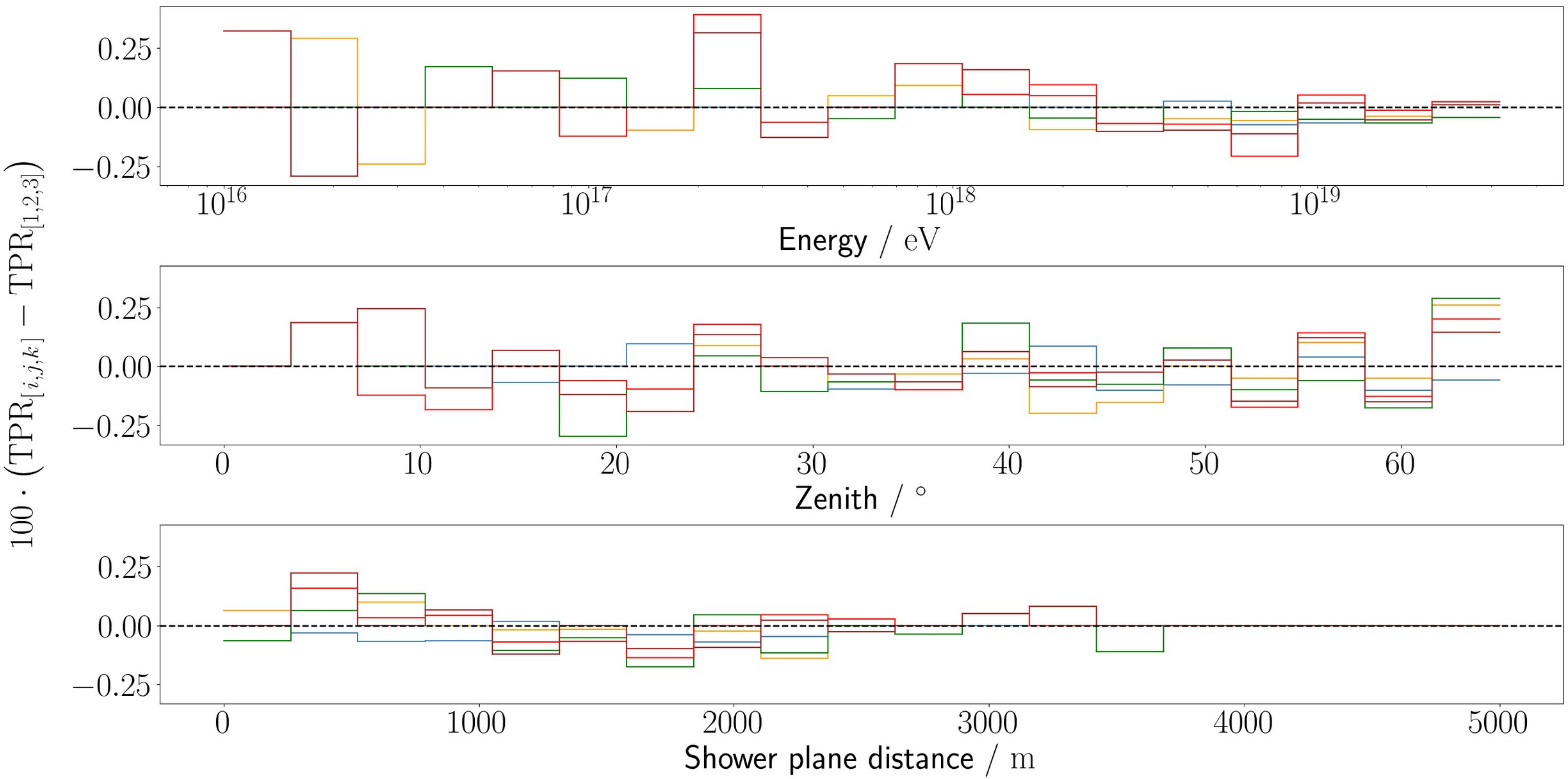



**Offline approach**




**Bayesian folding**

# LSTM permutations



# Network parameters

| Type | Input size | Kernel size | $n_{\text{train}}$ | 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** |