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

IAP HEU seminar 2024/01/25

Paul Filip, David Schmidt

Acknowledgements: Markus Roth, Darko Veberič, Dave Nitz

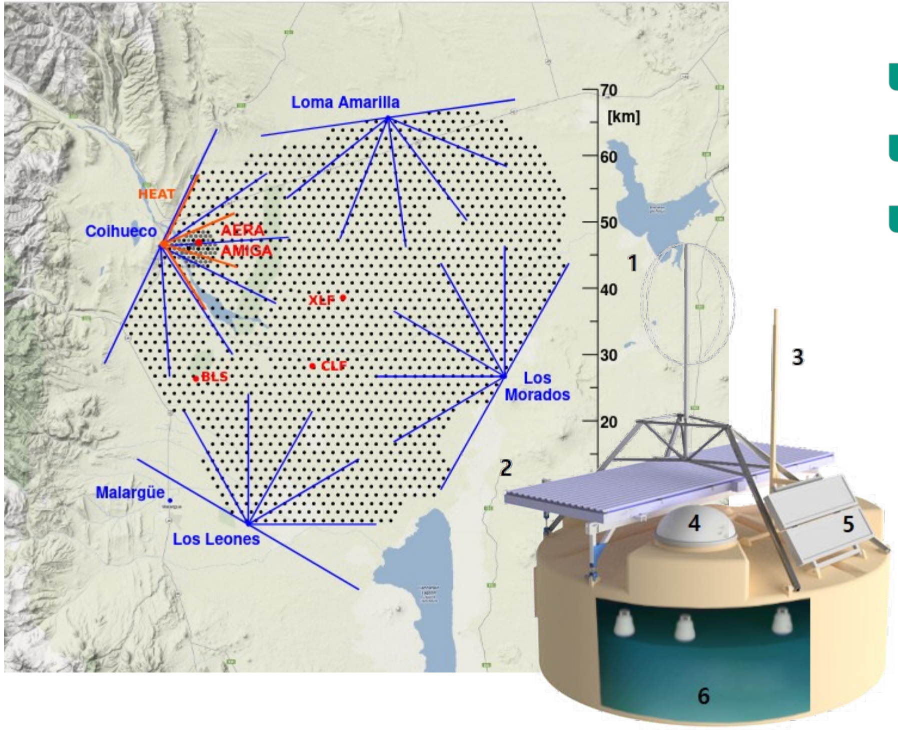
Goals

- Are NN-based triggers feasible SD/WCD triggers?
- Test feasibility with (for now) simulated proton showers

Requirements

- Limited computational power in FPGAs/UUB electronics
- Limited choice in candidate network architectures!
- Limited number of trainable parameters!

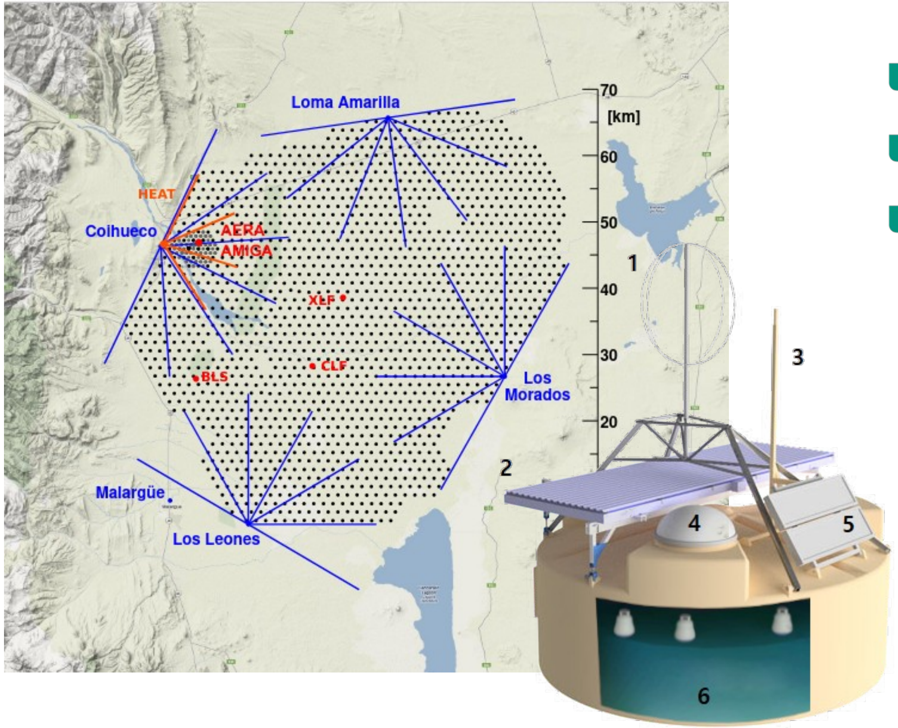
SD Array / trigger hierarchy / WCD time traces



- Around ~1600 stations
- Triangular 1500 m grid spacing
- Quasi 100% uptime

1 RD	2 SSD
3 Comms Antenna	4 Electronics Box
5 Solar Panel	6 WCD

SD Array / trigger hierarchy / WCD time traces



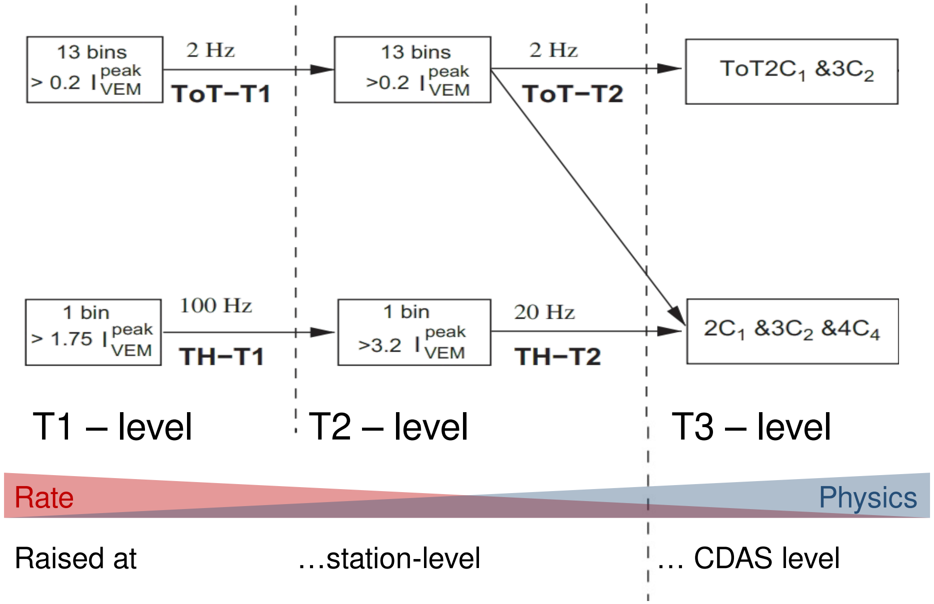
- **Around ~1600 stations**
- **Triangular 1500 m grid spacing**
- **Quasi 100% uptime**

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

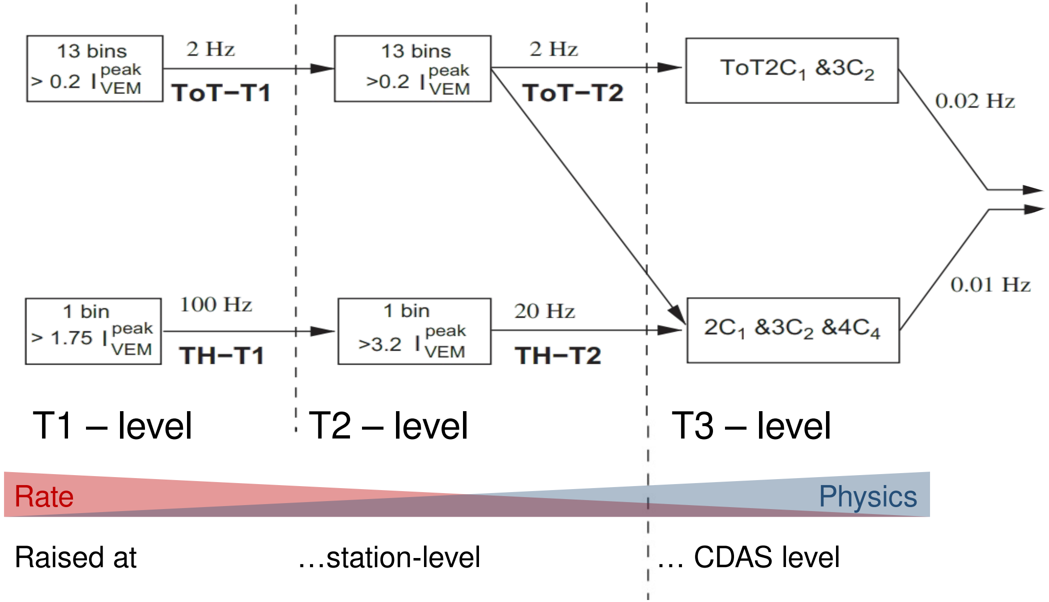
➡ Implement trigger hierarchy

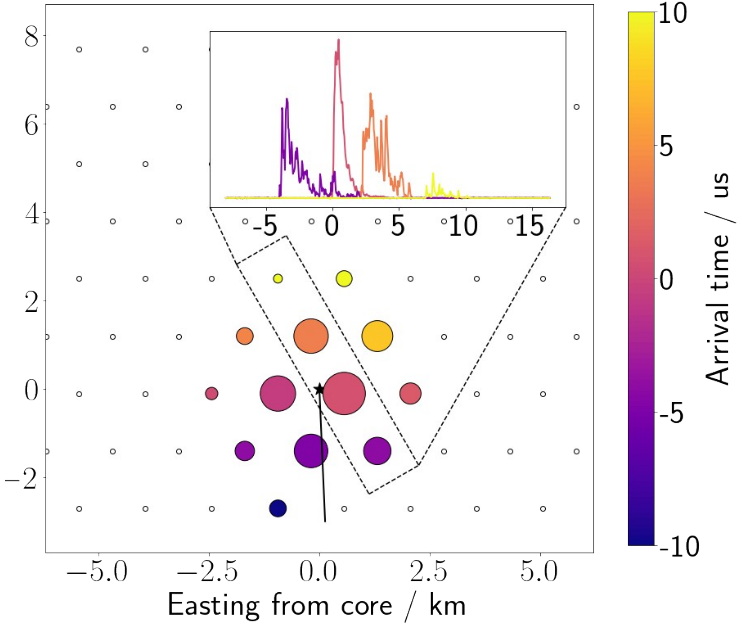
1 RD	2 SSD
3 Comms Antenna	4 Electronics Box
5 Solar Panel	6 WCD

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

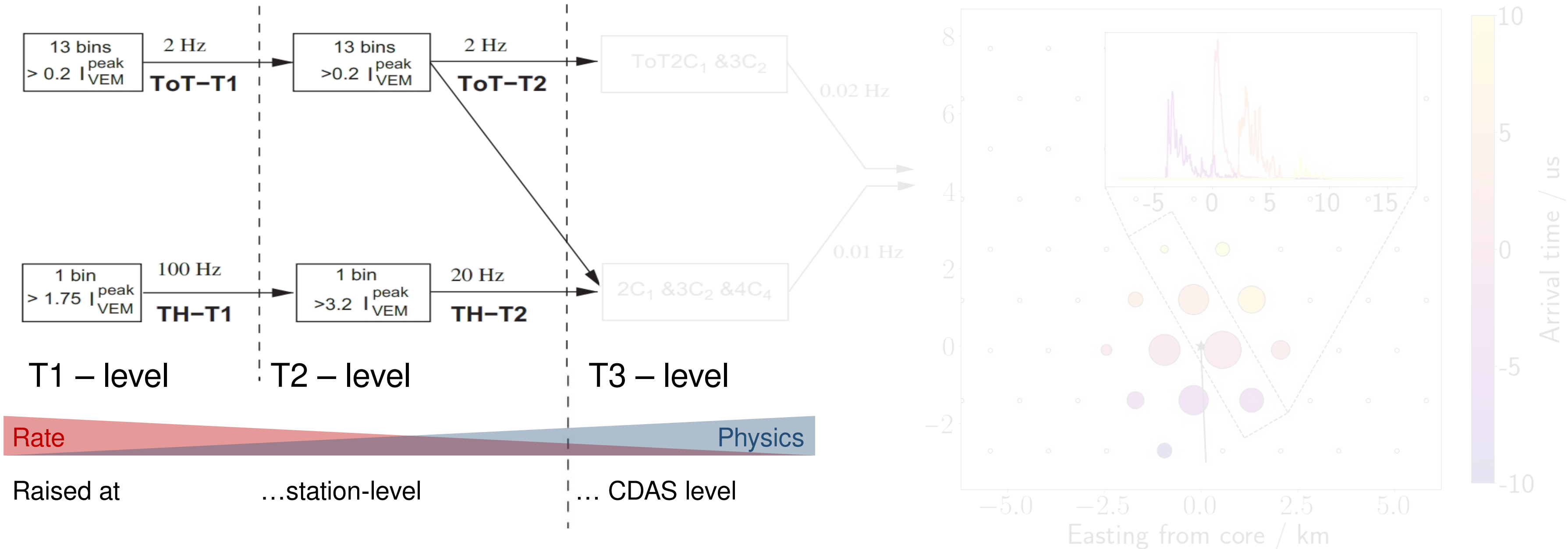


SD Array / **trigger hierarchy** / WCD time traces





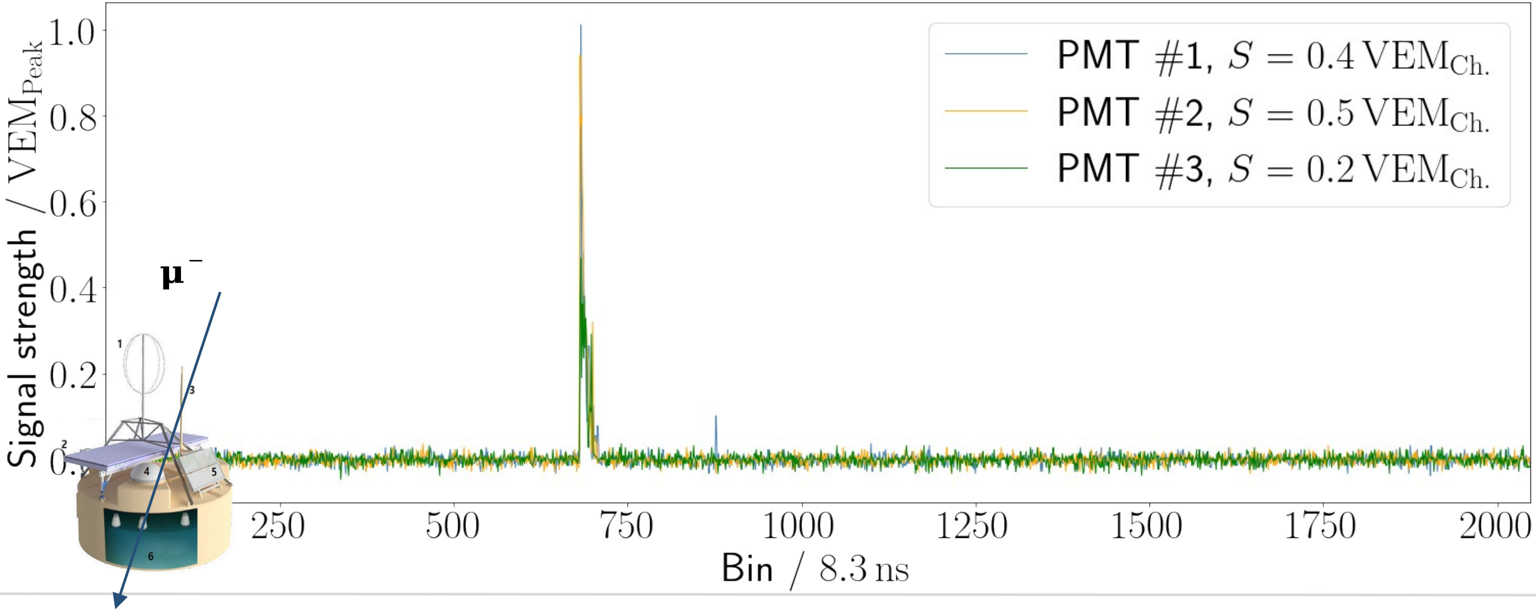
SD Array / **trigger hierarchy** / WCD time traces



- **Limited computational power in FPGAs/UUB electronics**
- **Limited choice in candidate network architectures/size**

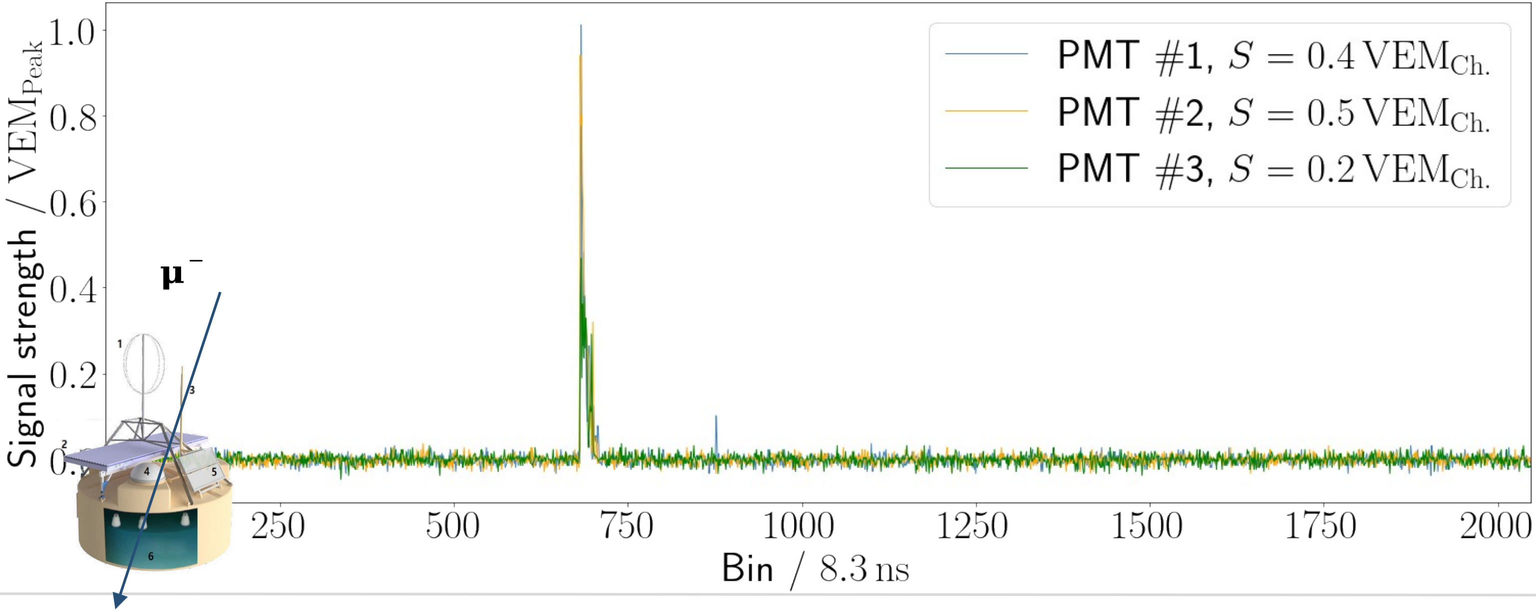
SD Array / trigger hierarchy / **WCD time traces**

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



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

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



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

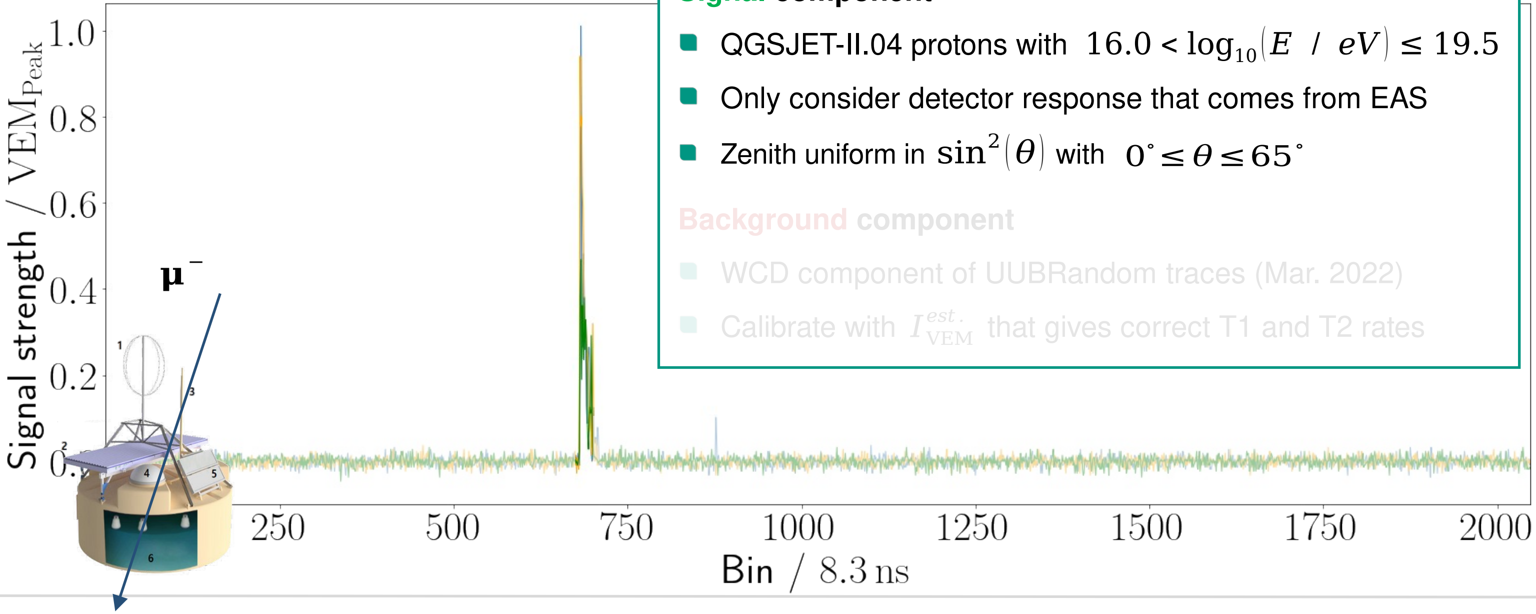
- **NN triggers**
  - **Feed labelled subset of trace to neural network architecture**
  - **Teach it to distinguish between Signal / Background**

Signal component

- QGSJET-II.04 protons with $16.0 < \log_{10}(E\ /\ eV) \leq 19.5$
- Only consider detector response that comes from EAS
- Zenith uniform in $\sin^2(\theta)$ with $0^\circ \leq \theta \leq 65^\circ$

Background component

- WCD component of UUBRandom traces (Mar. 2022)
- Calibrate with $I_{\mathrm{VEM}}^{est.}$ that gives correct T1 and T2 rates



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

Karlsruhe Institute of Technology

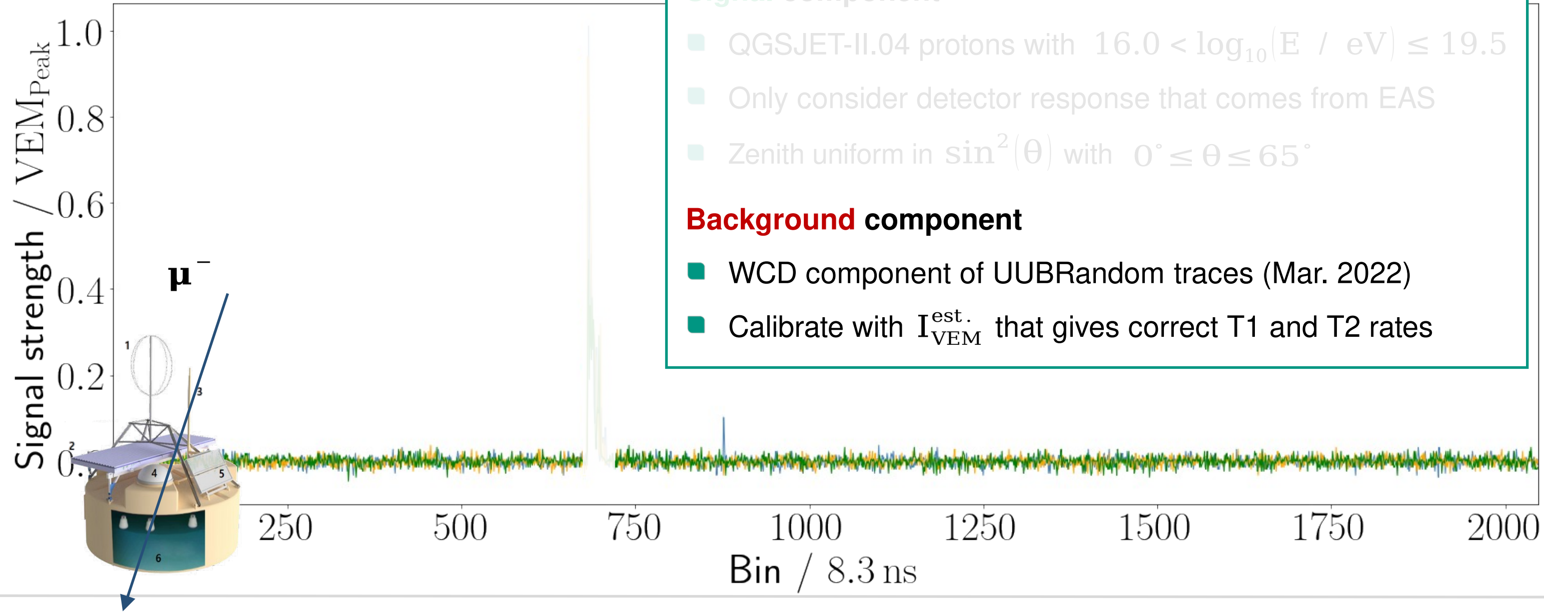
- **NN triggers**
  - **Feed labelled subset of trace to neural network architecture**
  - **Teach it to distinguish between Signal / Background**

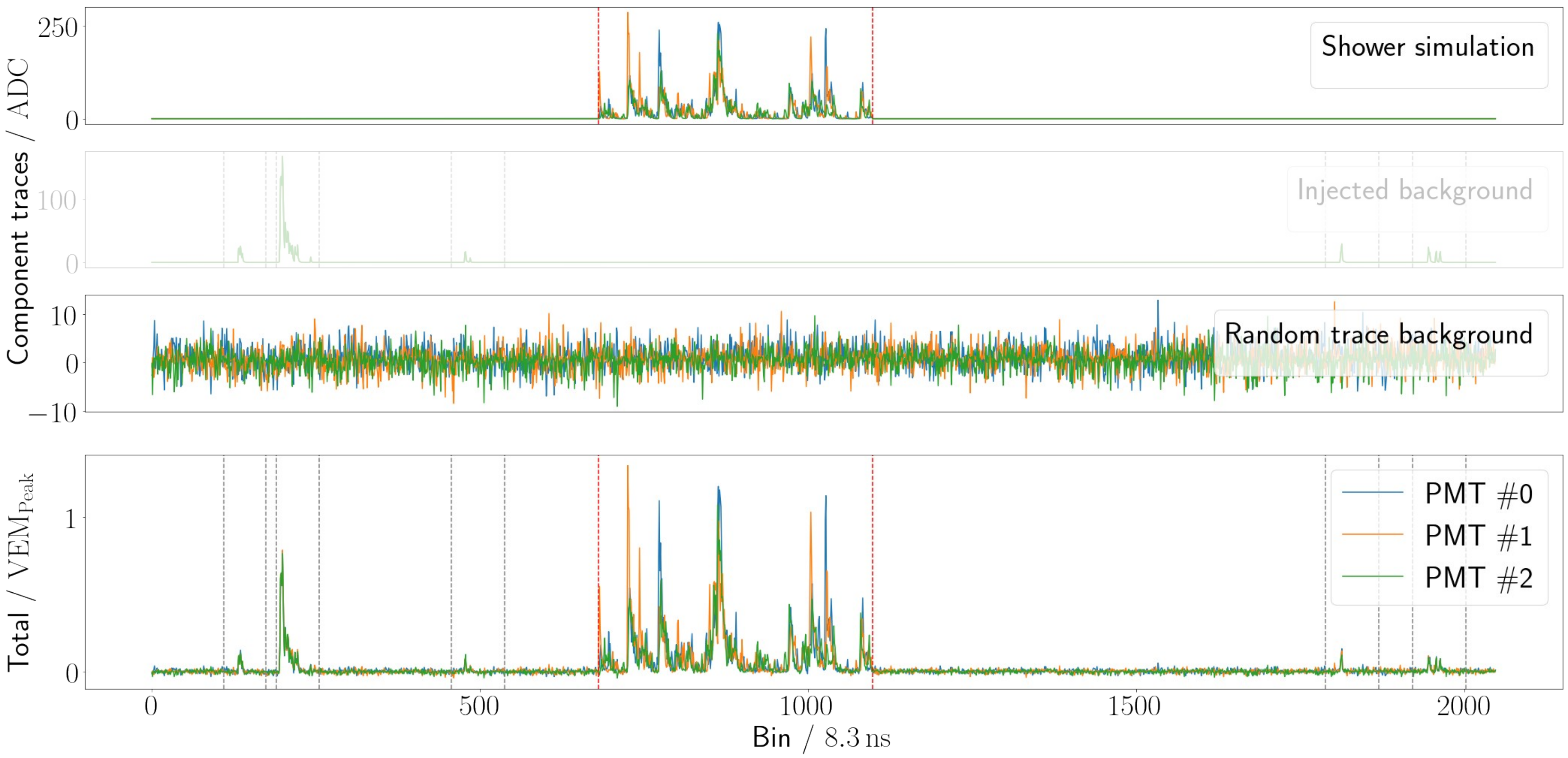
Signal component

- QGSJET-II.04 protons with $16.0 < \log_{10}(E\ /\ \mathrm{eV}) \leq 19.5$
- Only consider detector response that comes from EAS
- Zenith uniform in $\sin^2(\theta)$ with $0^\circ \leq \theta \leq 65^\circ$

Background component

- WCD component of UUBRandom traces (Mar. 2022)
- Calibrate with $I_{\mathrm{VEM}}^{\mathrm{est.}}$ that gives correct T1 and T2 rates




Shower simulation
Injected background
Random trace background
Component traces / ADC
Total / $\mathrm{VEM_{Peak}}$
PMT #0
PMT #1
PMT #2
Bin / $8.3\,\mathrm{ns}$
250
0
100
0
10
0
−10
1
0
0
500
1000
1500
2000

Network architectures

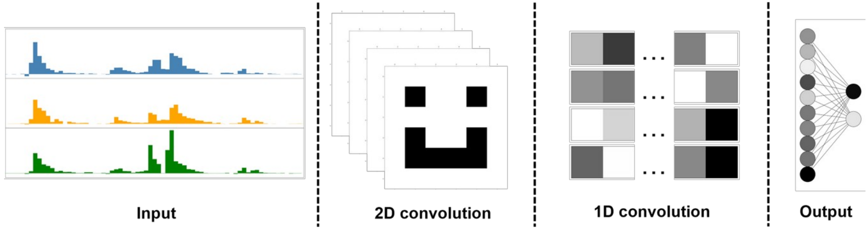
- 120 bins x 3 PMTs = 360 input values ⟹ 1 binary output (1 = Shower, 0 = Background)

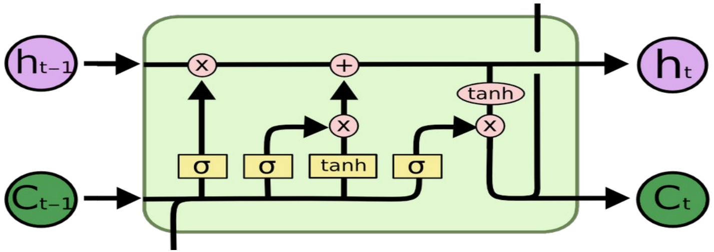
Convolutional neural networks (CNNs)

- 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/

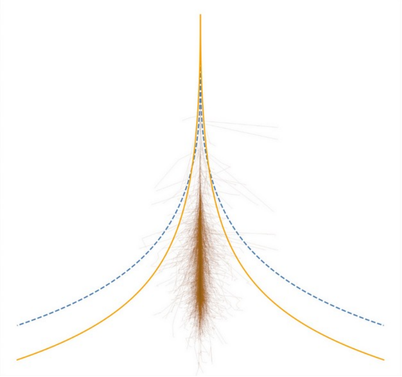
Estimating performance

- Have (after training neural network) trigger efficiency **given** signal: $\mathrm{P}(\mathrm{T2} \mid \mathrm{Signal})$
- Want trigger efficiency independent of EAS particles in tank: $\mathrm{P}(\mathrm{T2})$

Estimating performance

- Have (after training neural network) trigger efficiency **given** signal: $\mathrm{P}(\mathrm{T2} \mid \mathrm{Signal})$
- Want trigger efficiency independent of EAS particles in tank: $\mathrm{P}(\mathrm{T2})$
- Scale with lateral particle probability (LPP): $\mathrm{P}(\mathrm{Signal})$

$$\mathrm{LPP} = \min\left(1, \quad \right)^*$$



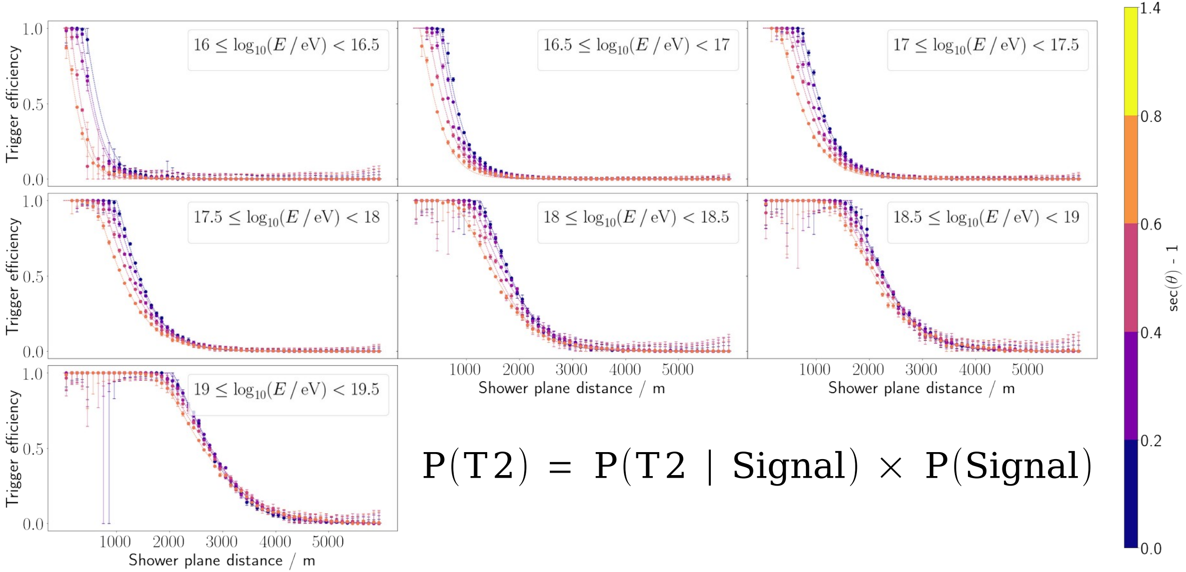
Lateral distribution function

$*$ not really, but helps with intuition

Estimating performance

- Have (after training neural network) trigger efficiency **given** signal: $\mathrm{P}(\mathrm{T2} \mid \mathrm{Signal})$
- Want trigger efficiency independent of EAS particles in tank: $\mathrm{P}(\mathrm{T2})$
- Scale with lateral particle probability (LPP): $\mathrm{P}(\mathrm{Signal})$

$$\mathrm{LPP} =$$



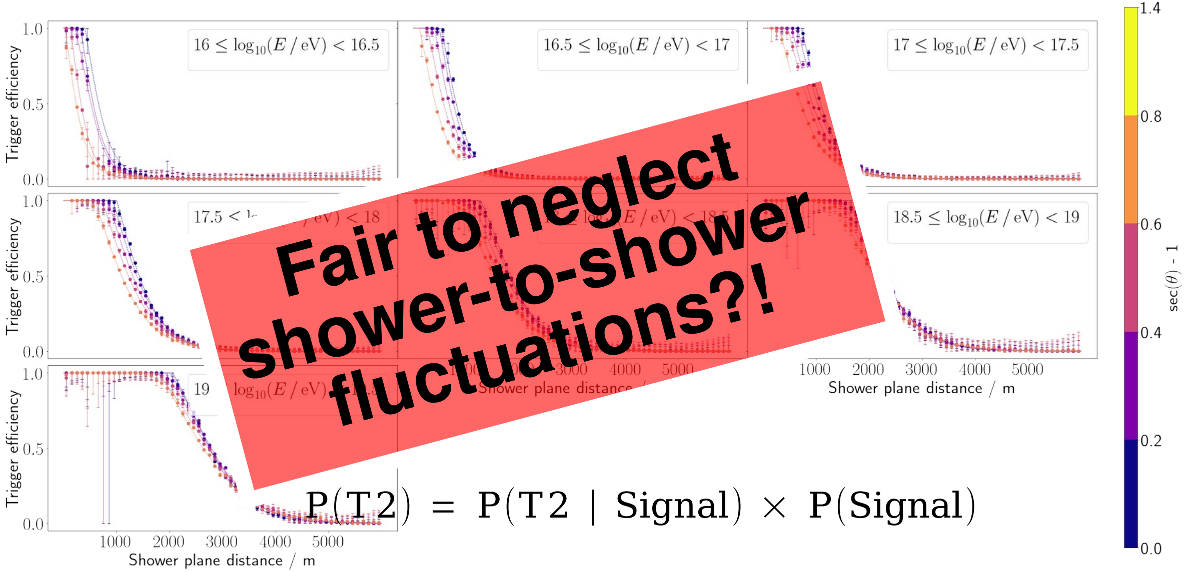
$$\mathrm{P}(\mathrm{T2}) = \mathrm{P}(\mathrm{T2} \mid \mathrm{Signal}) \times \mathrm{P}(\mathrm{Signal})$$

Estimating performance

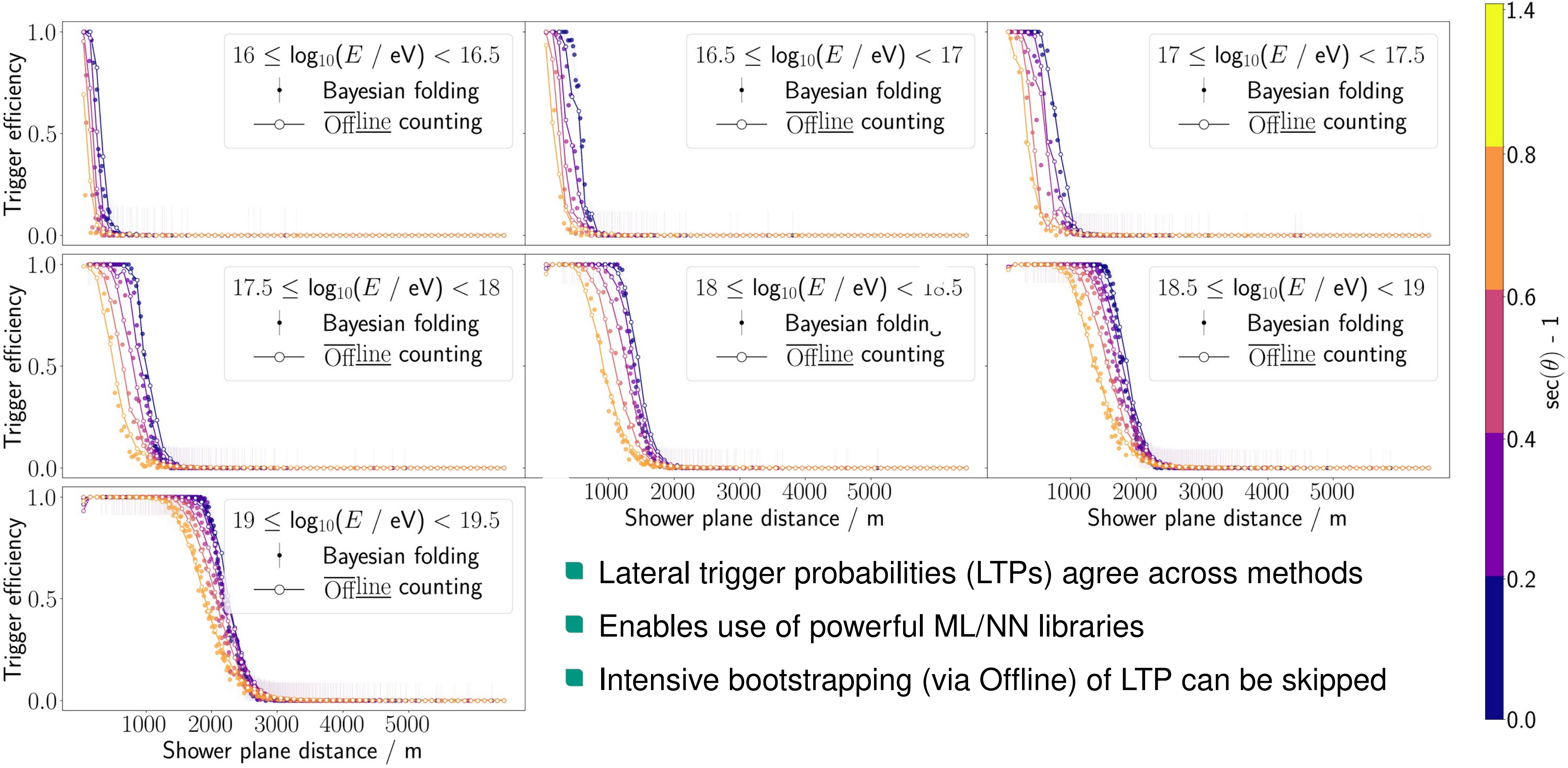
Karlsruhe Institute of Technology

- Have (after training neural network) trigger efficiency **given** signal: $\mathrm{P(T2 \mid Signal)}$
- Want trigger efficiency independent of EAS particles in tank: $\mathrm{P(T2)}$
- Scale with lateral particle probability (LPP): $\mathrm{P(Signal)}$

$$\mathrm{LPP} =$$

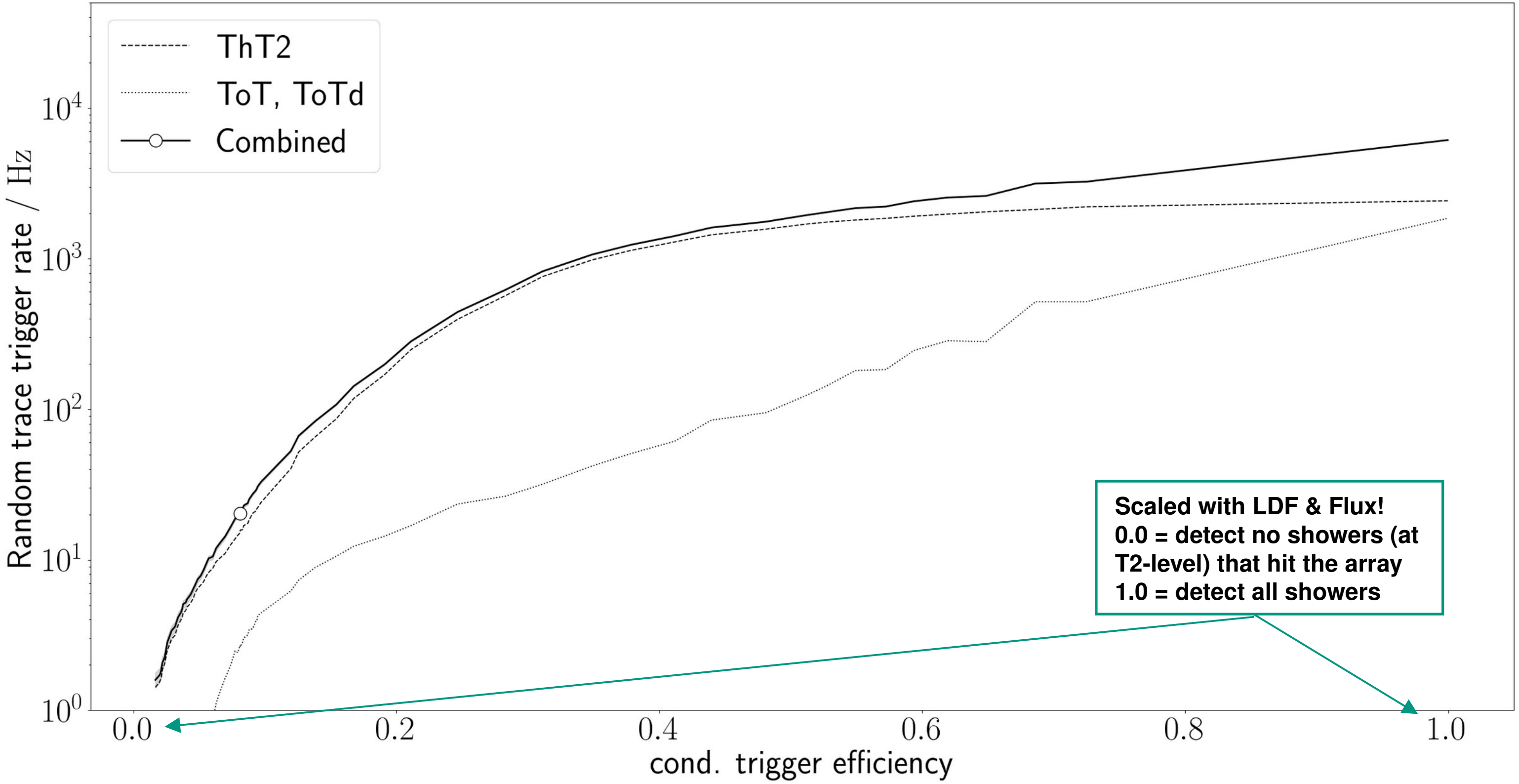


$$\mathrm{P(T2) = P(T2 \mid Signal) \times P(Signal)}$$



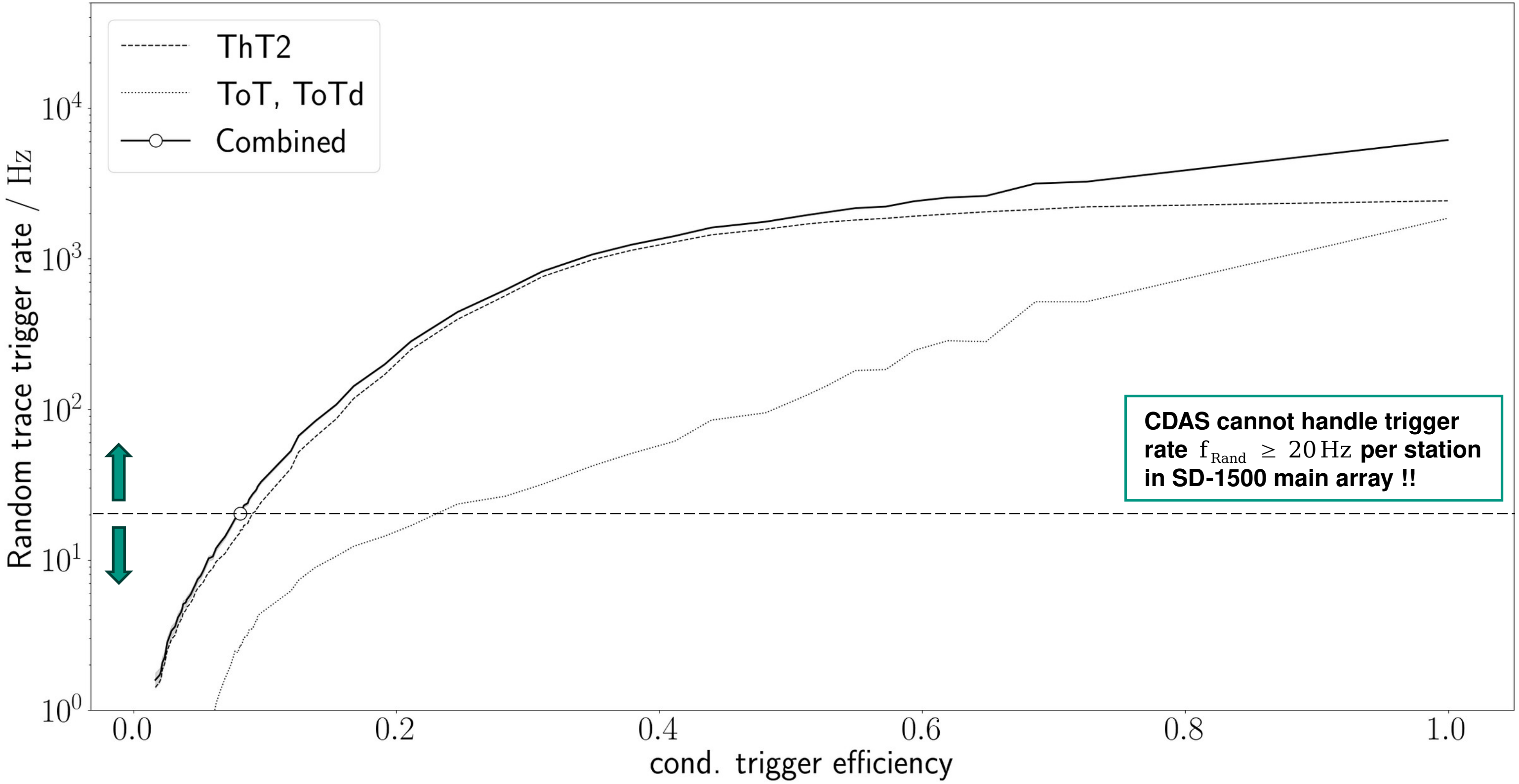
- Lateral trigger probabilities (LTPs) agree across methods
- Enables use of powerful ML/NN libraries
- Intensive bootstrapping (via Offline) of LTP can be skipped

Comparing trigger efficiency

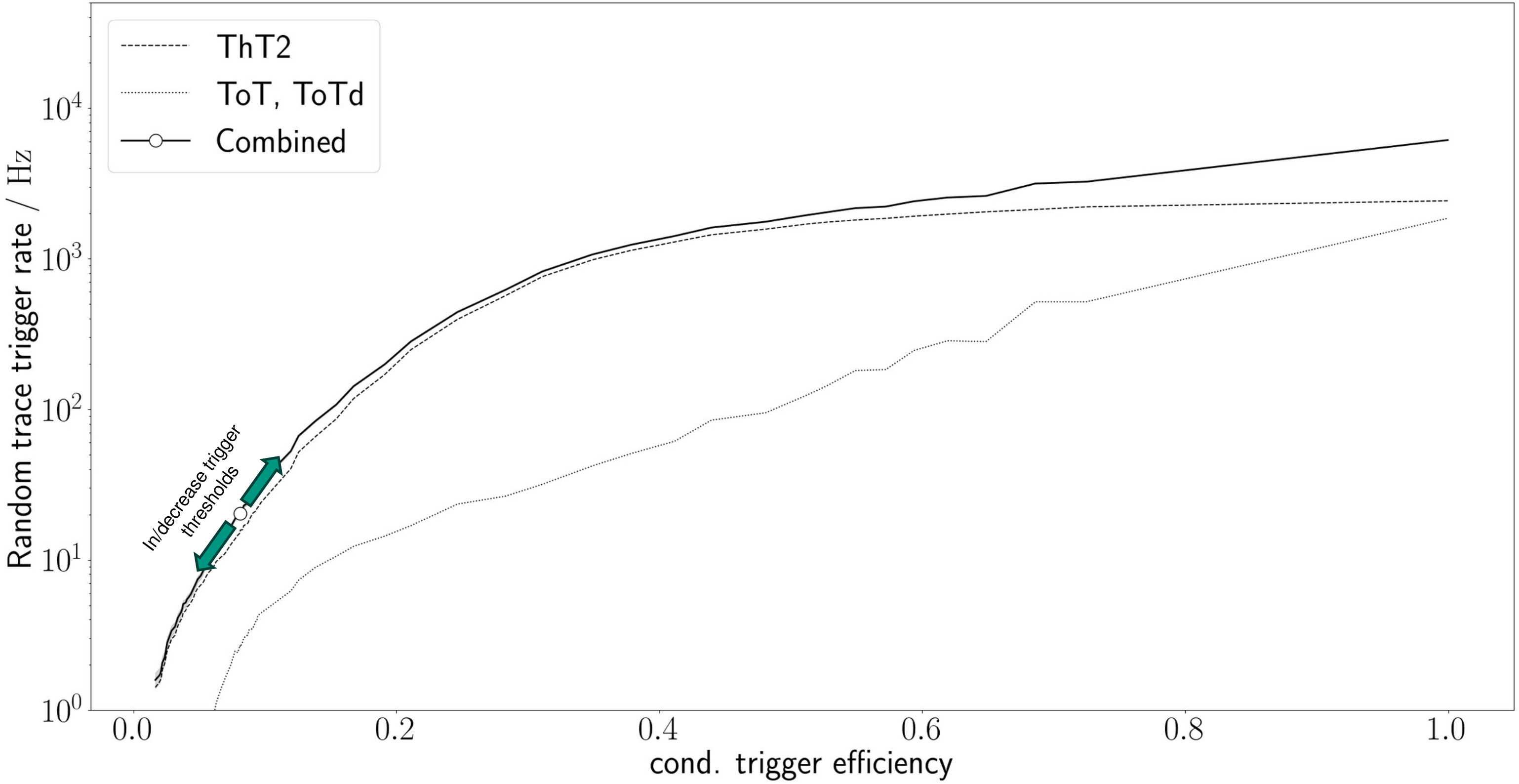


Scaled with LDF & Flux!
0.0 = detect no showers (at T2-level) that hit the array
1.0 = detect all showers

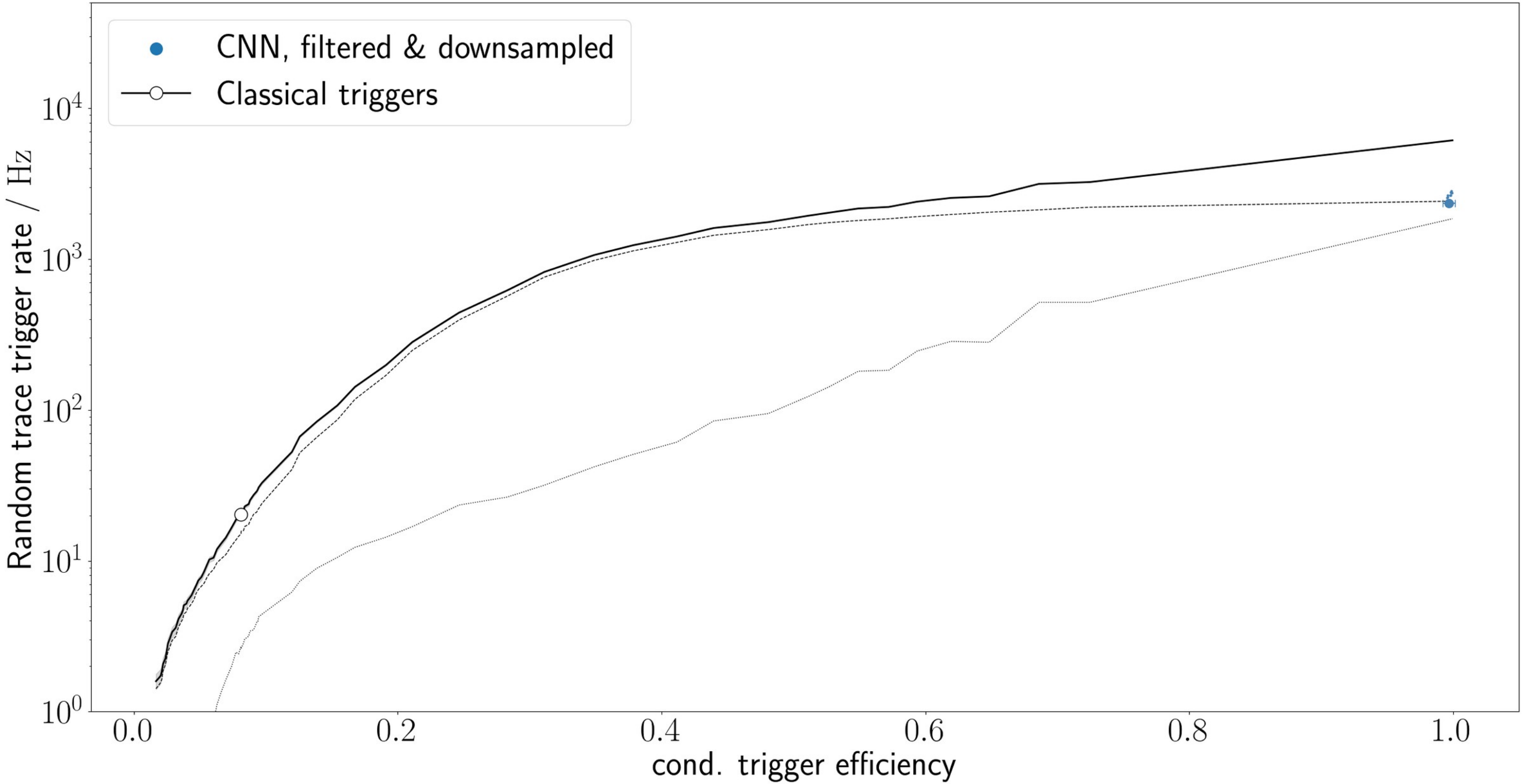
Comparing trigger efficiency



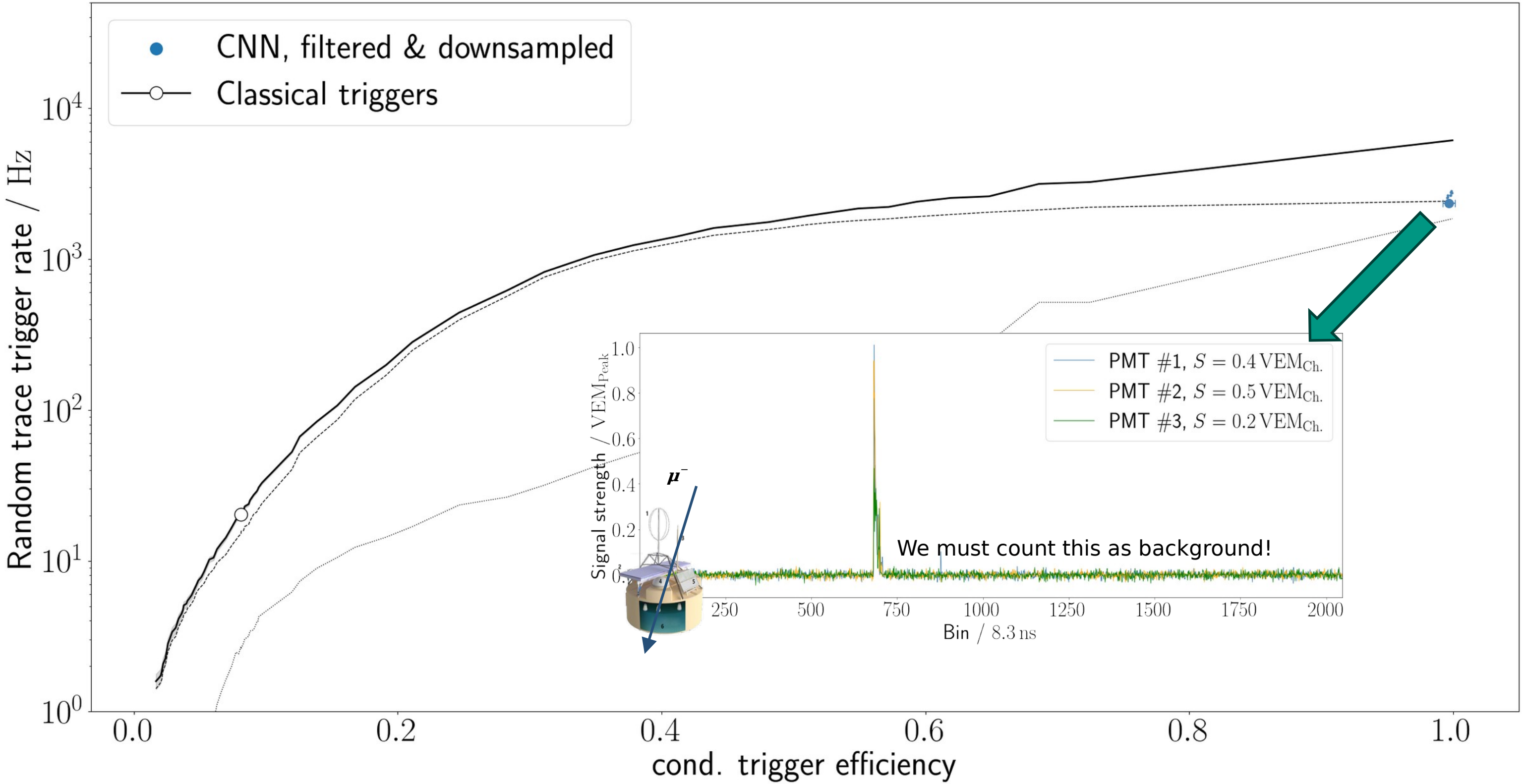
Comparing trigger efficiency



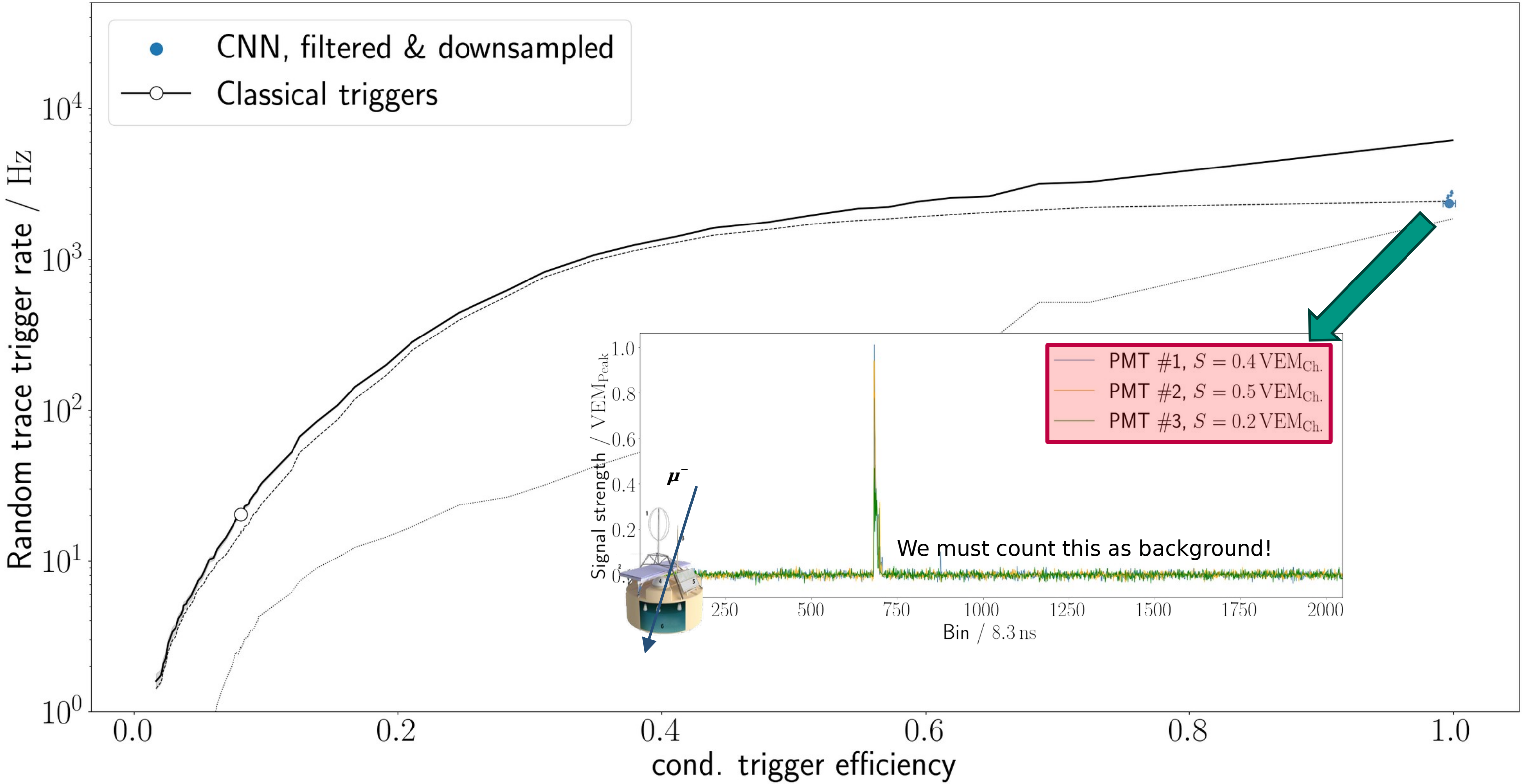
Comparing trigger efficiency



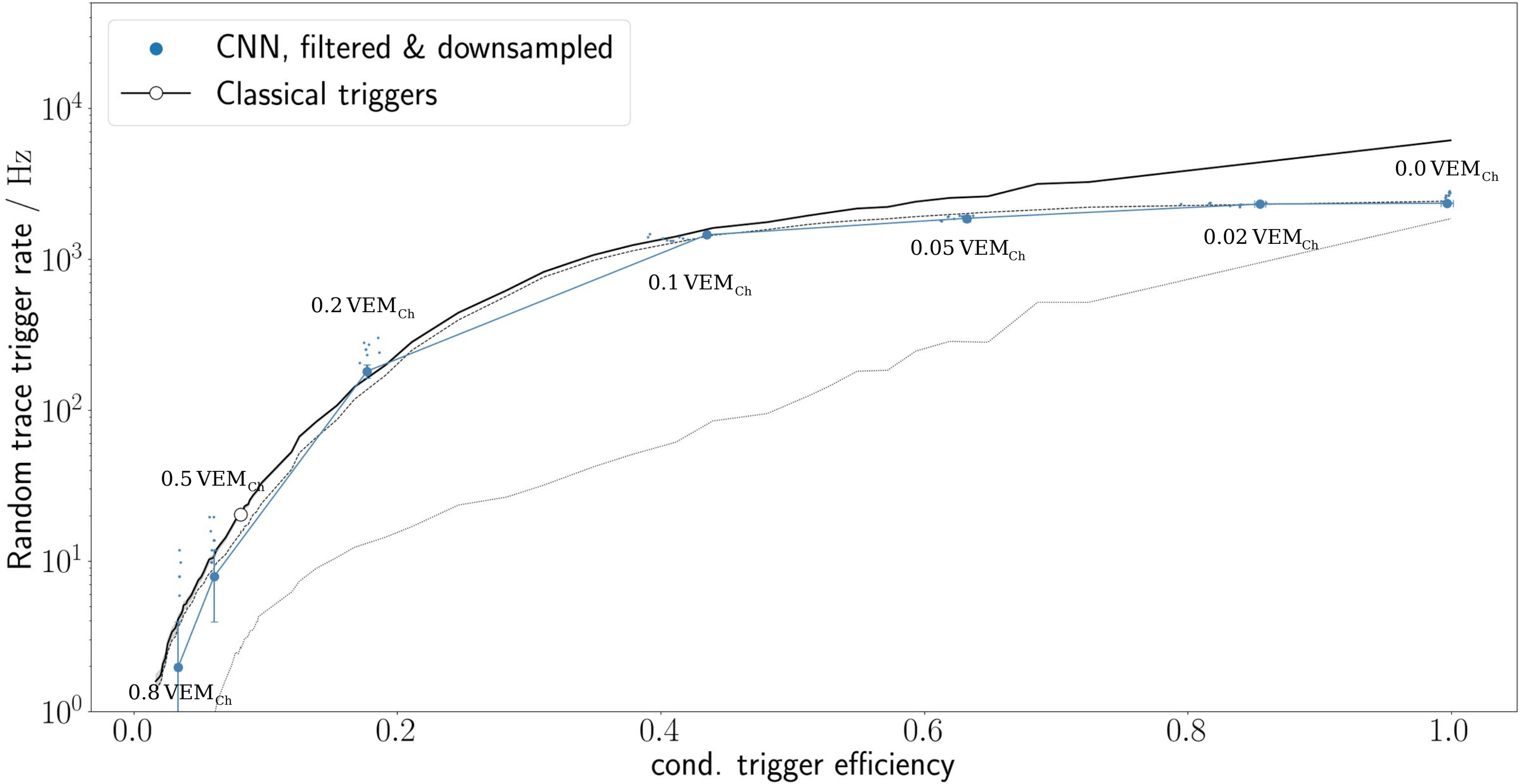
Comparing trigger efficiency



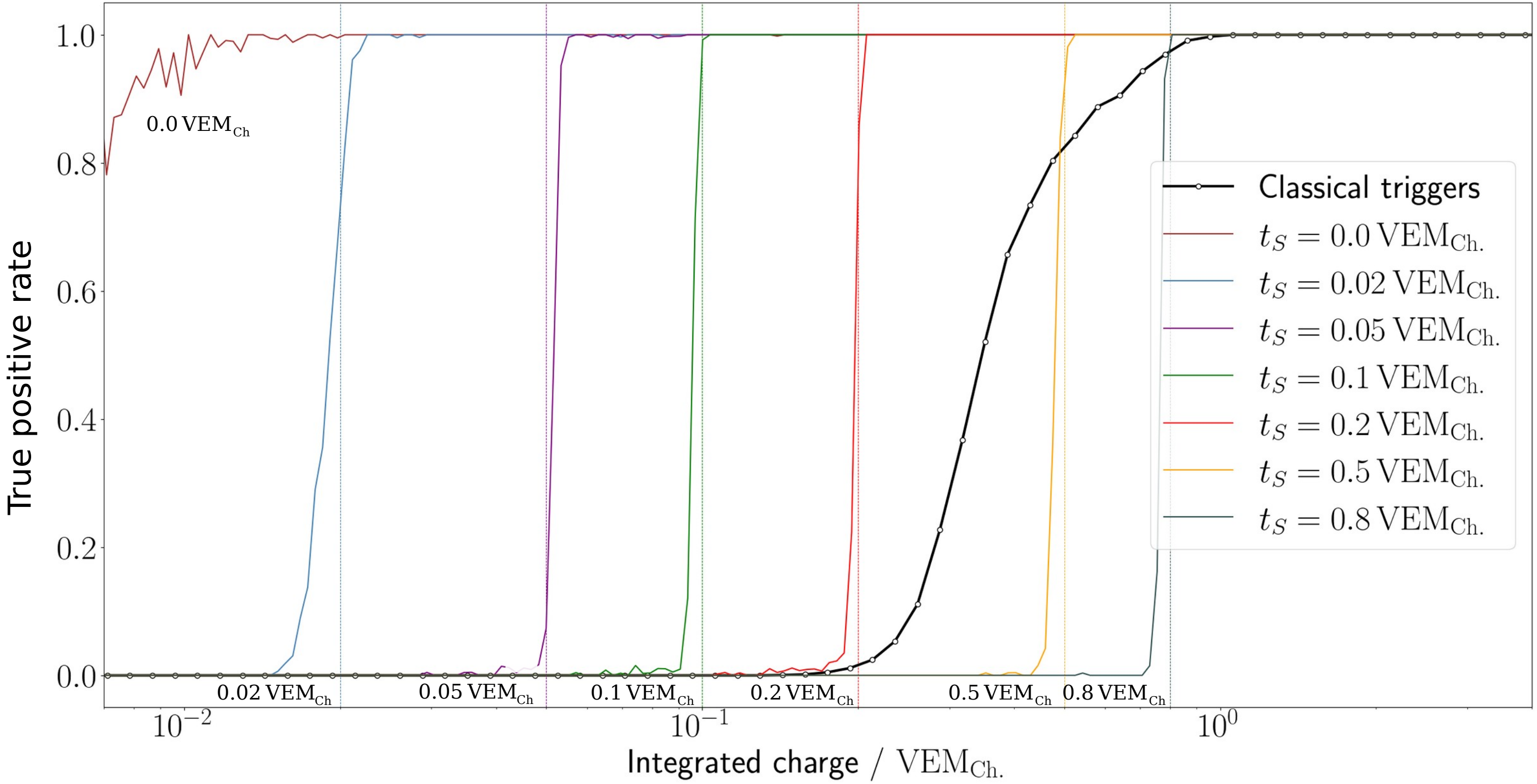
Comparing trigger efficiency



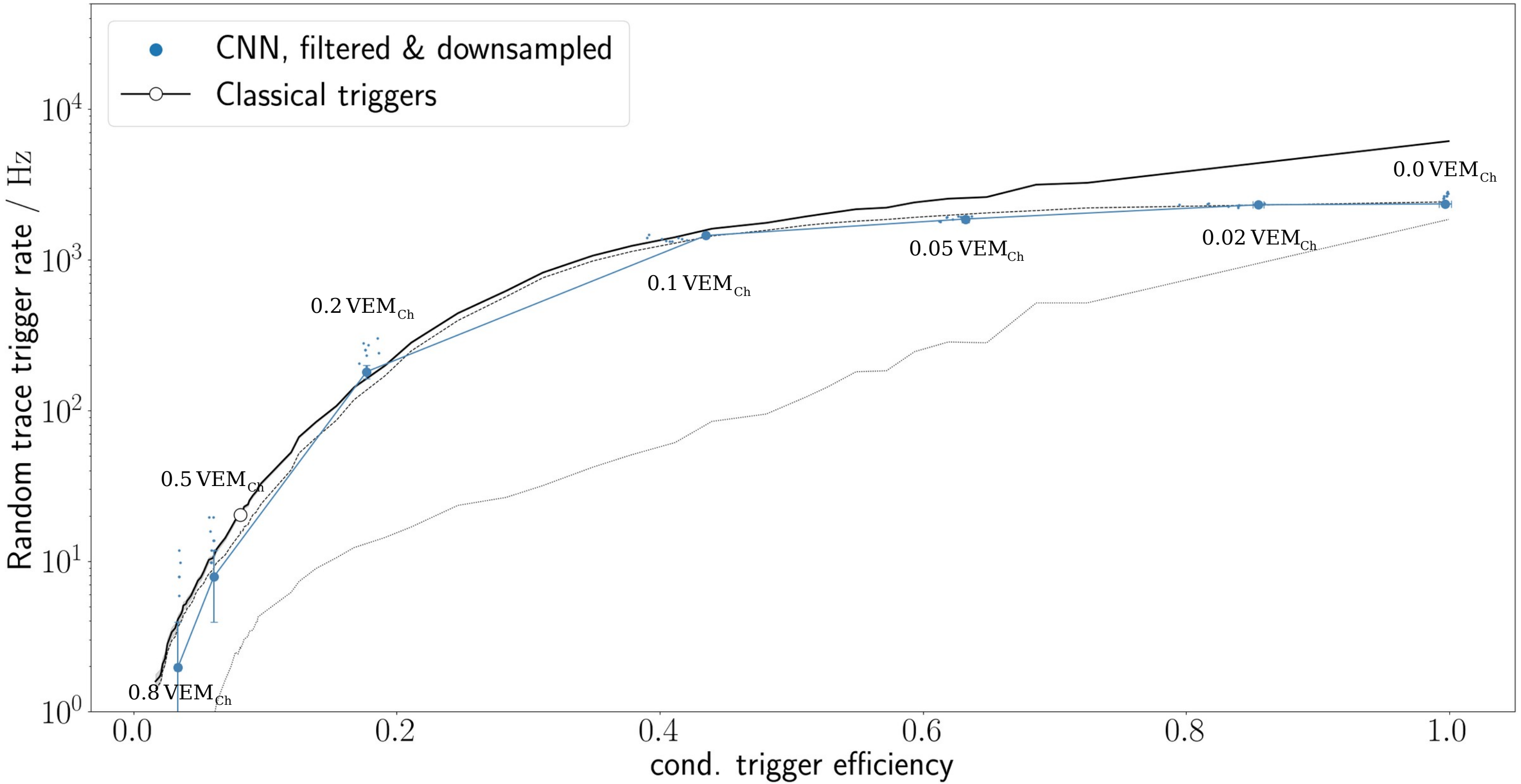
Comparing trigger efficiency



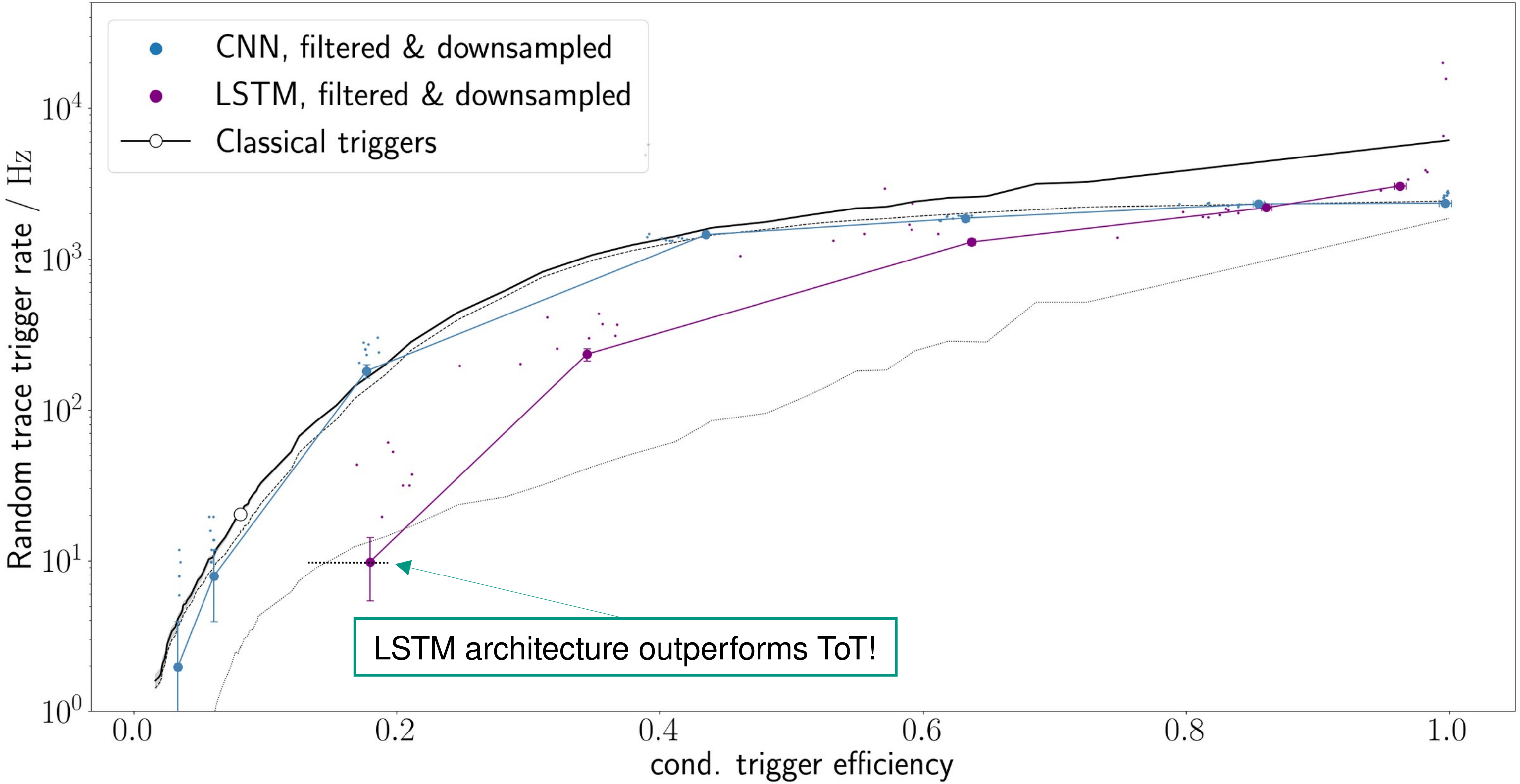
Sanity check – Charge cut



Comparing trigger efficiency

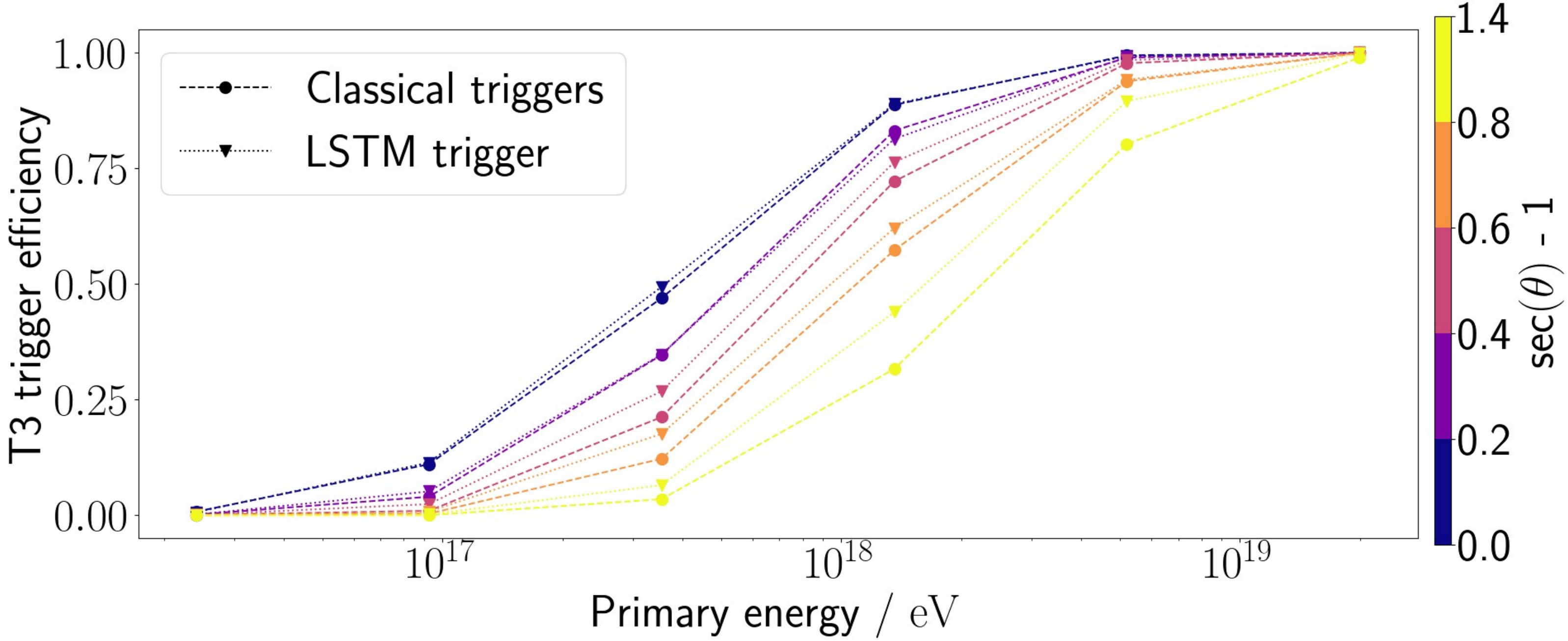


Comparing trigger efficiency



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

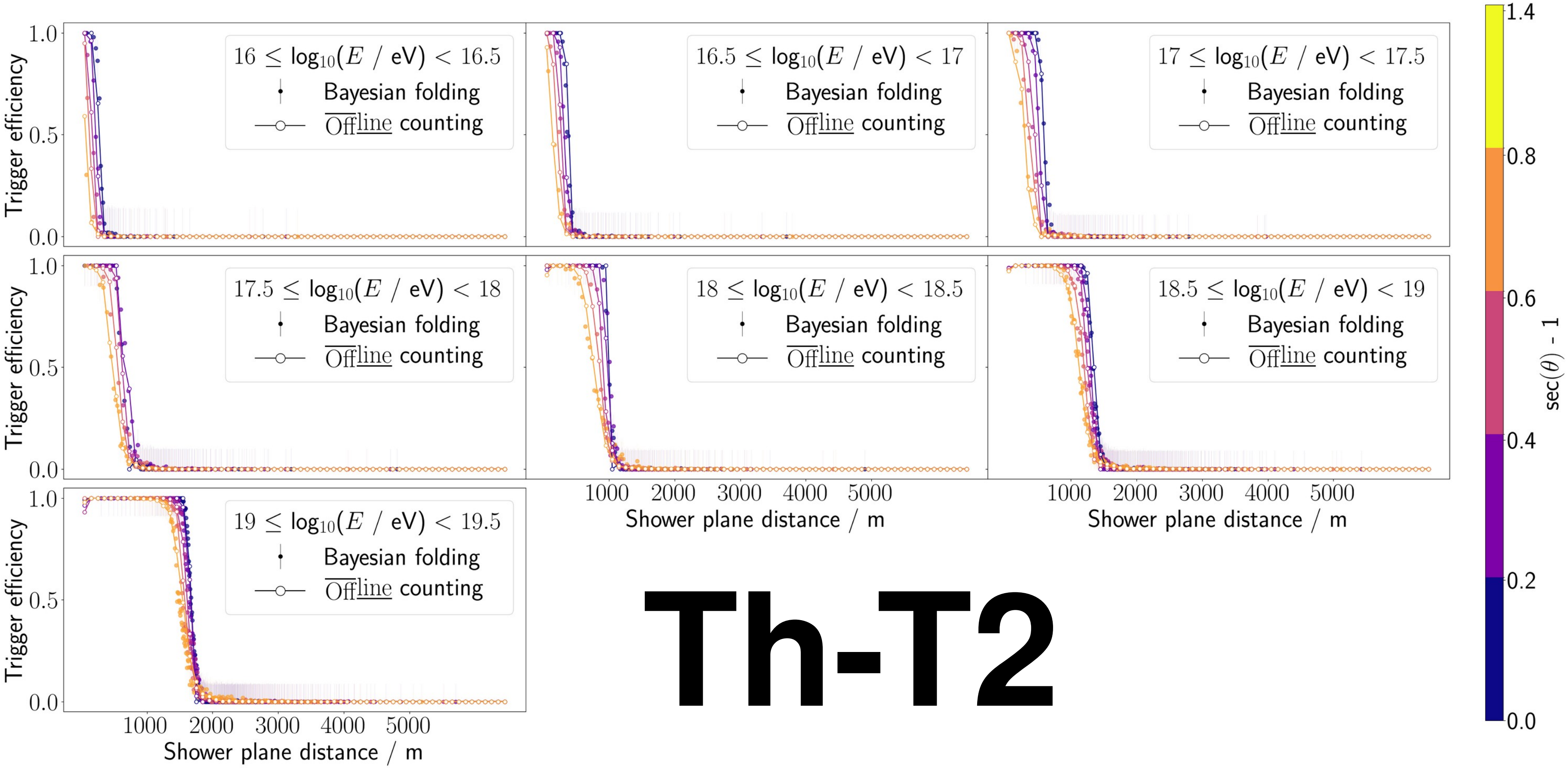
- **Most drastic gains at high inclinations**
- **Possibly higher gains at $\theta \geq 65^\circ$**



Summary

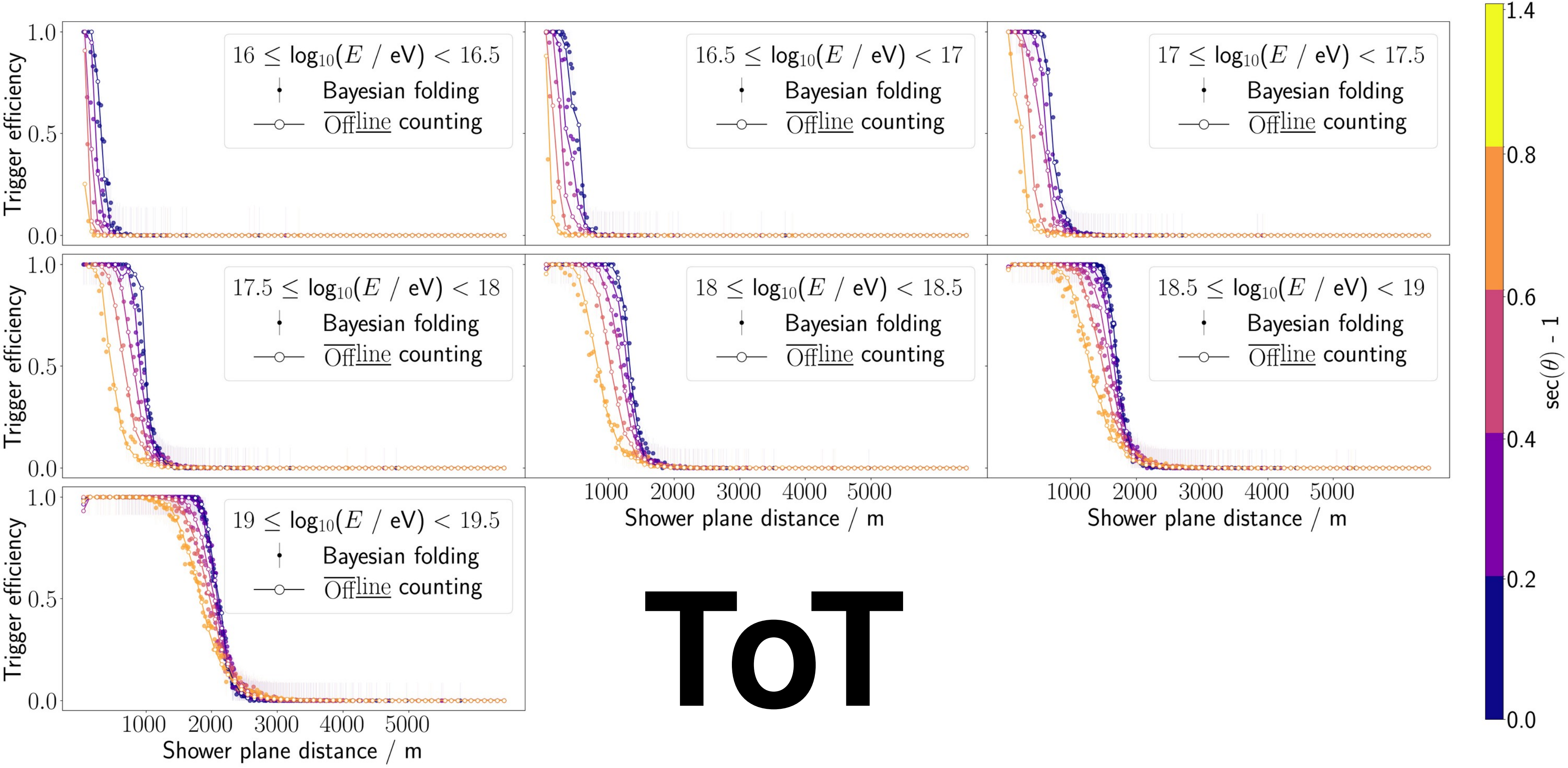
- Test data-driven, machine learning concepts
  - Analyse capability of shallow NNs (few parameters!) as SD T2-triggers
  - Consider convolutional (CNN) and simple recurrent neural networks (LSTM)
  - Verify performance of NNs with measured background data
  - Control trigger rate by implementing charge 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
  - Results indicate performance on par with or better than ToT
  - Gains in event detection efficiency at high shower angles

Backup



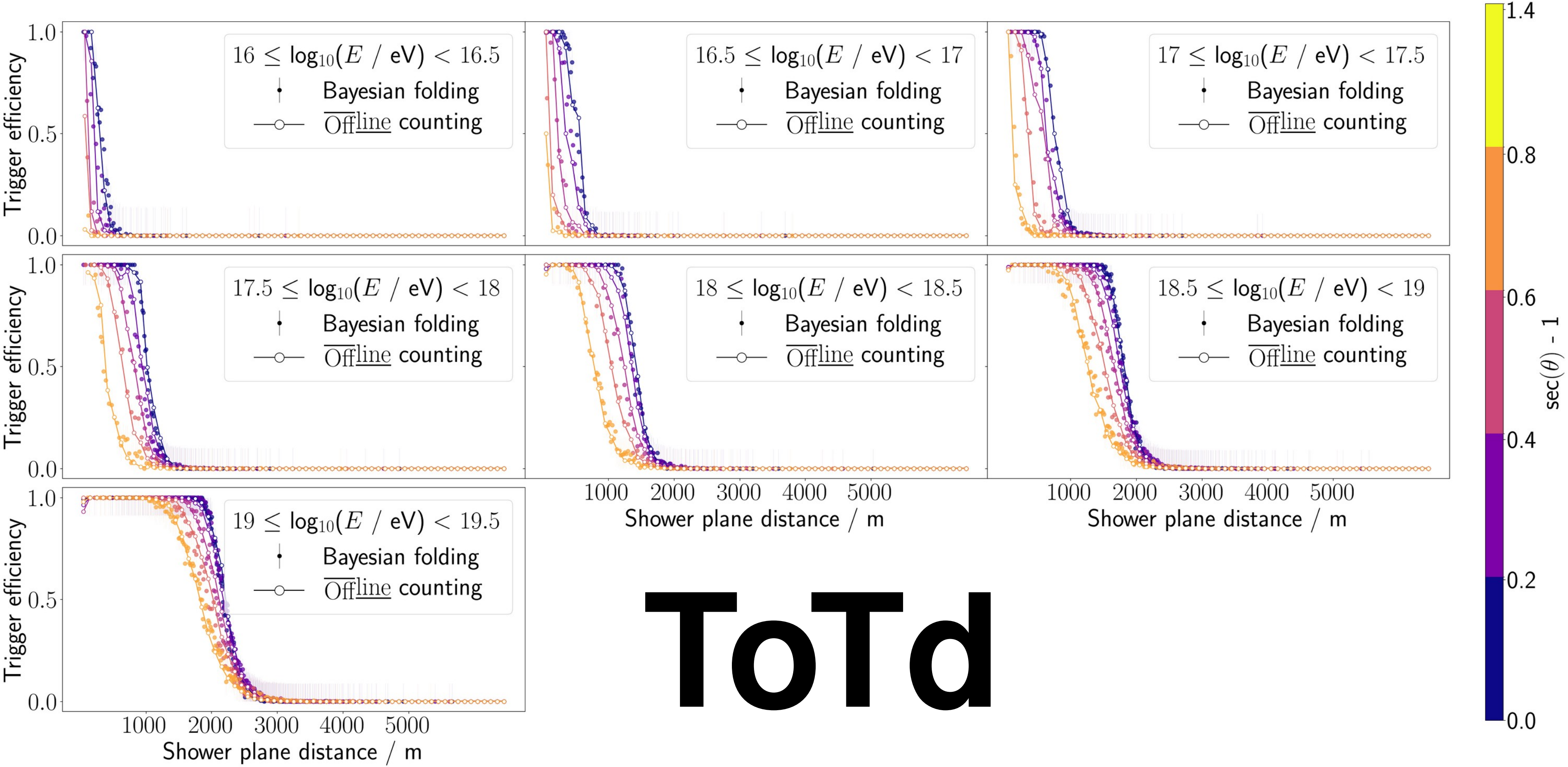
Trigger efficiency
$16 \leq \log_{10}(E\ /\ \mathrm{eV}) < 16.5$
$16.5 \leq \log_{10}(E\ /\ \mathrm{eV}) < 17$
$17 \leq \log_{10}(E\ /\ \mathrm{eV}) < 17.5$
$17.5 \leq \log_{10}(E\ /\ \mathrm{eV}) < 18$
$18 \leq \log_{10}(E\ /\ \mathrm{eV}) < 18.5$
$18.5 \leq \log_{10}(E\ /\ \mathrm{eV}) < 19$
$19 \leq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

Th-T2

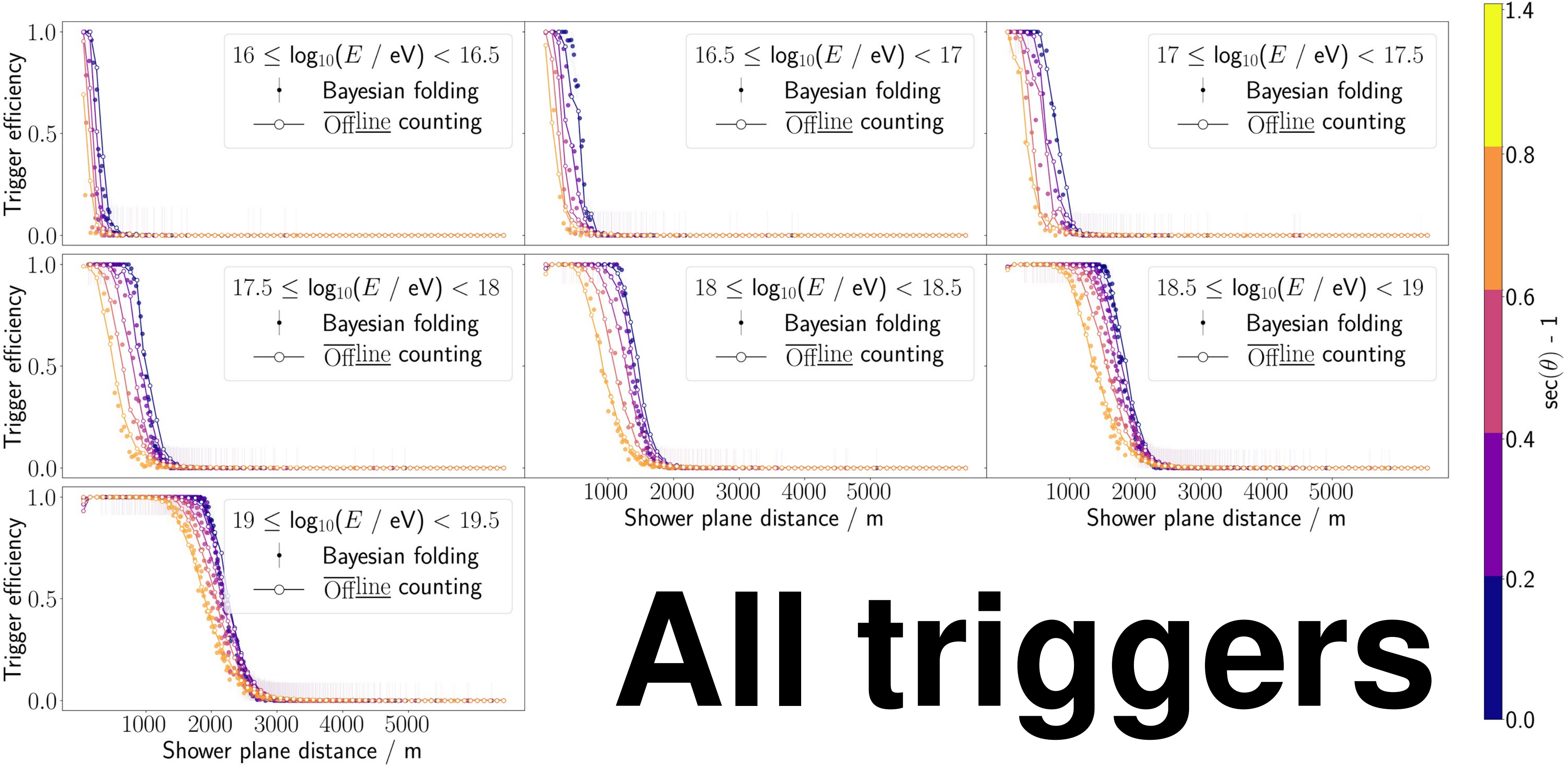


Trigger efficiency
1.0
0.5
0.0
$16 \leq \log_{10}(E\ /\ \mathrm{eV}) < 16.5$
Bayesian folding
Offline counting
$16.5 \leq \log_{10}(E\ /\ \mathrm{eV}) < 17$
$17 \leq \log_{10}(E\ /\ \mathrm{eV}) < 17.5$
$17.5 \leq \log_{10}(E\ /\ \mathrm{eV}) < 18$
$18 \leq \log_{10}(E\ /\ \mathrm{eV}) < 18.5$
$18.5 \leq \log_{10}(E\ /\ \mathrm{eV}) < 19$
$19 \leq \log_{10}(E\ /\ \mathrm{eV}) < 19.5$
1000
2000
3000
4000
5000
Shower plane distance / m
$\sec(\theta)$ - 1
0.0
0.2
0.4
0.6
0.8
1.4

ToT

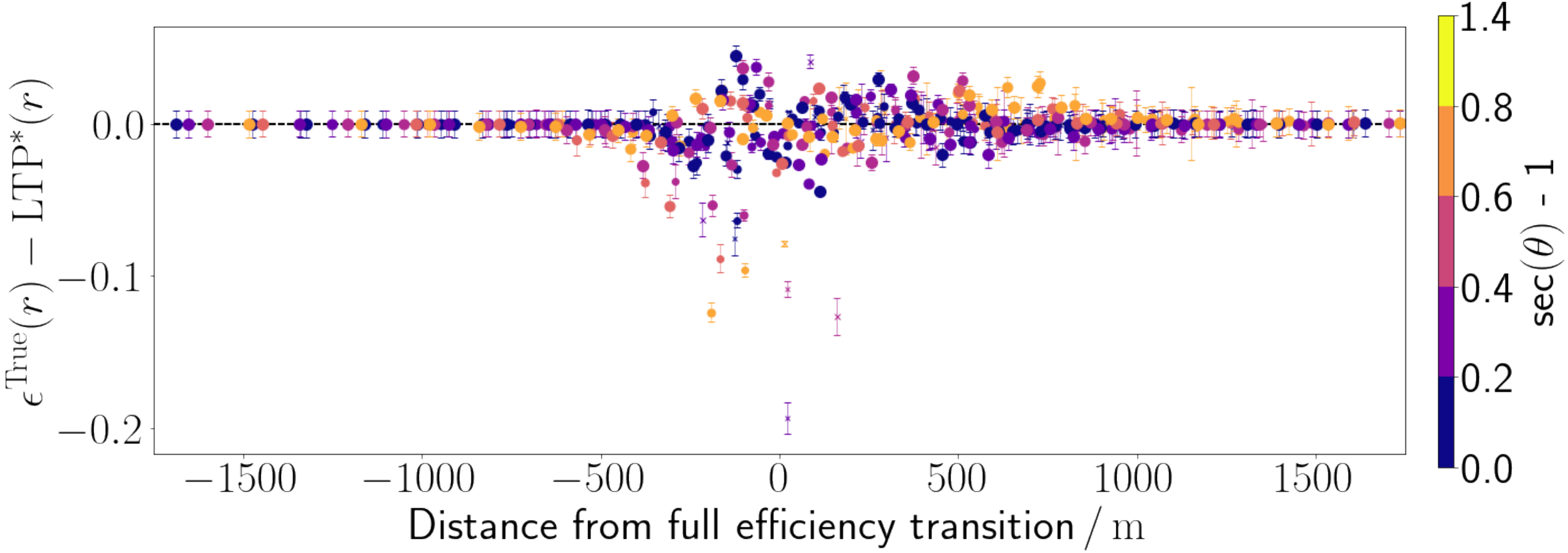

$16 \leq \log_{10}(E$ / eV$) < 16.5$
$16.5 \leq \log_{10}(E$ / eV$) < 17$
$17 \leq \log_{10}(E$ / eV$) < 17.5$
$17.5 \leq \log_{10}(E$ / eV$) < 18$
$18 \leq \log_{10}(E$ / eV$) < 18.5$
$18.5 \leq \log_{10}(E$ / eV$) < 19$
$19 \leq \log_{10}(E$ / eV$) < 19.5$
Bayesian folding
Offline counting
Trigger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

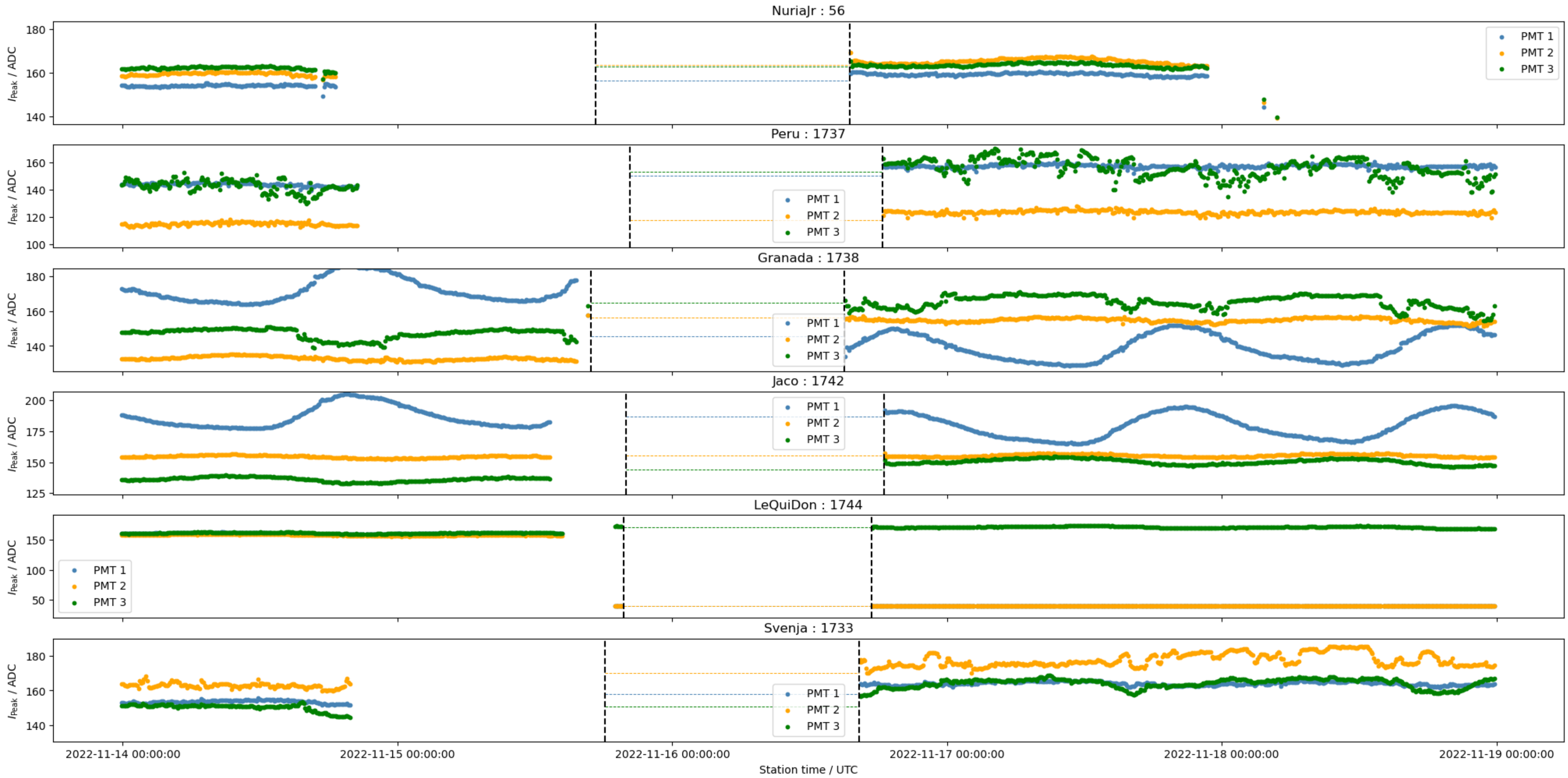
ToTd



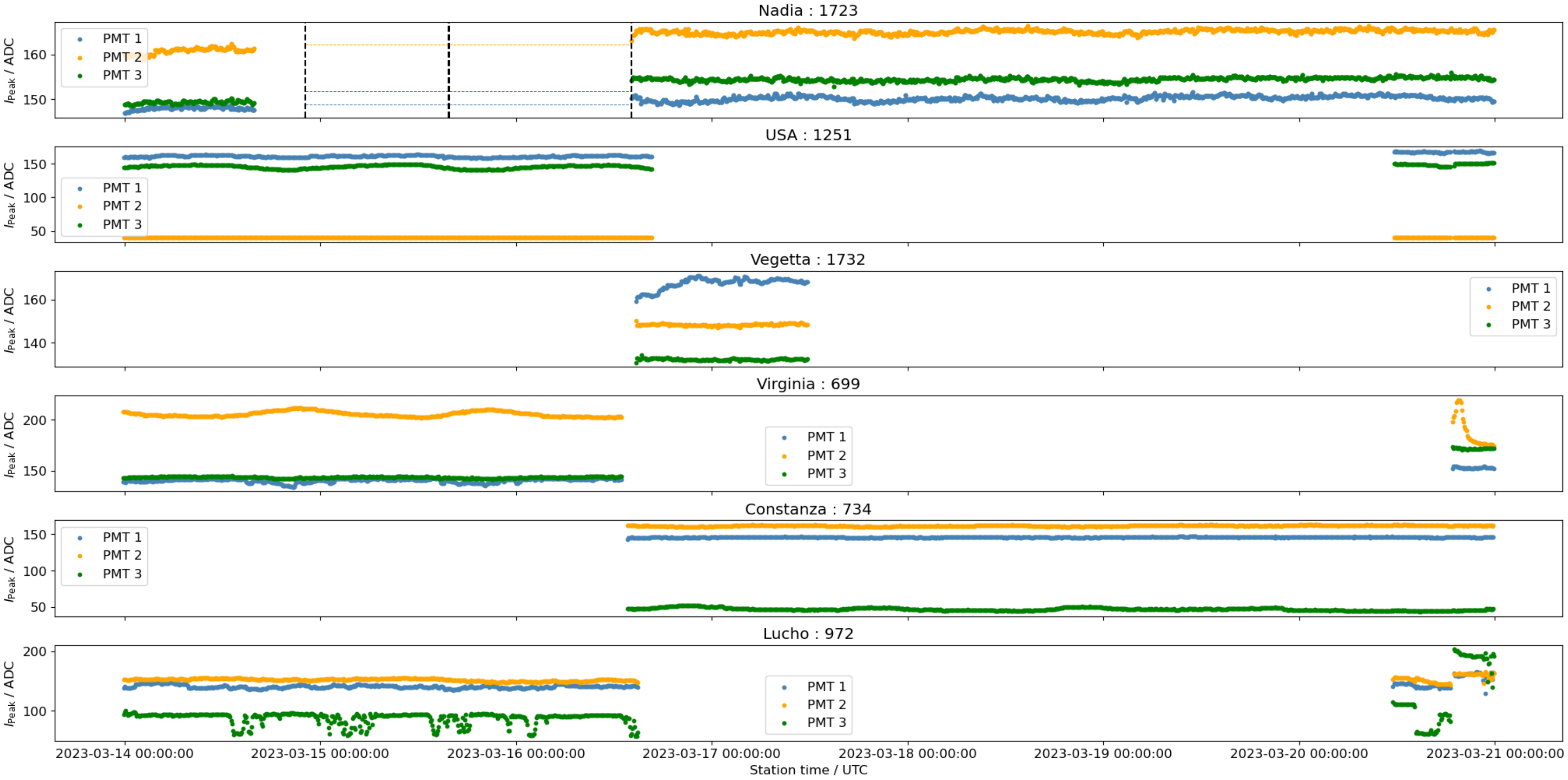
All triggers

Residuals – LTP fitfunction

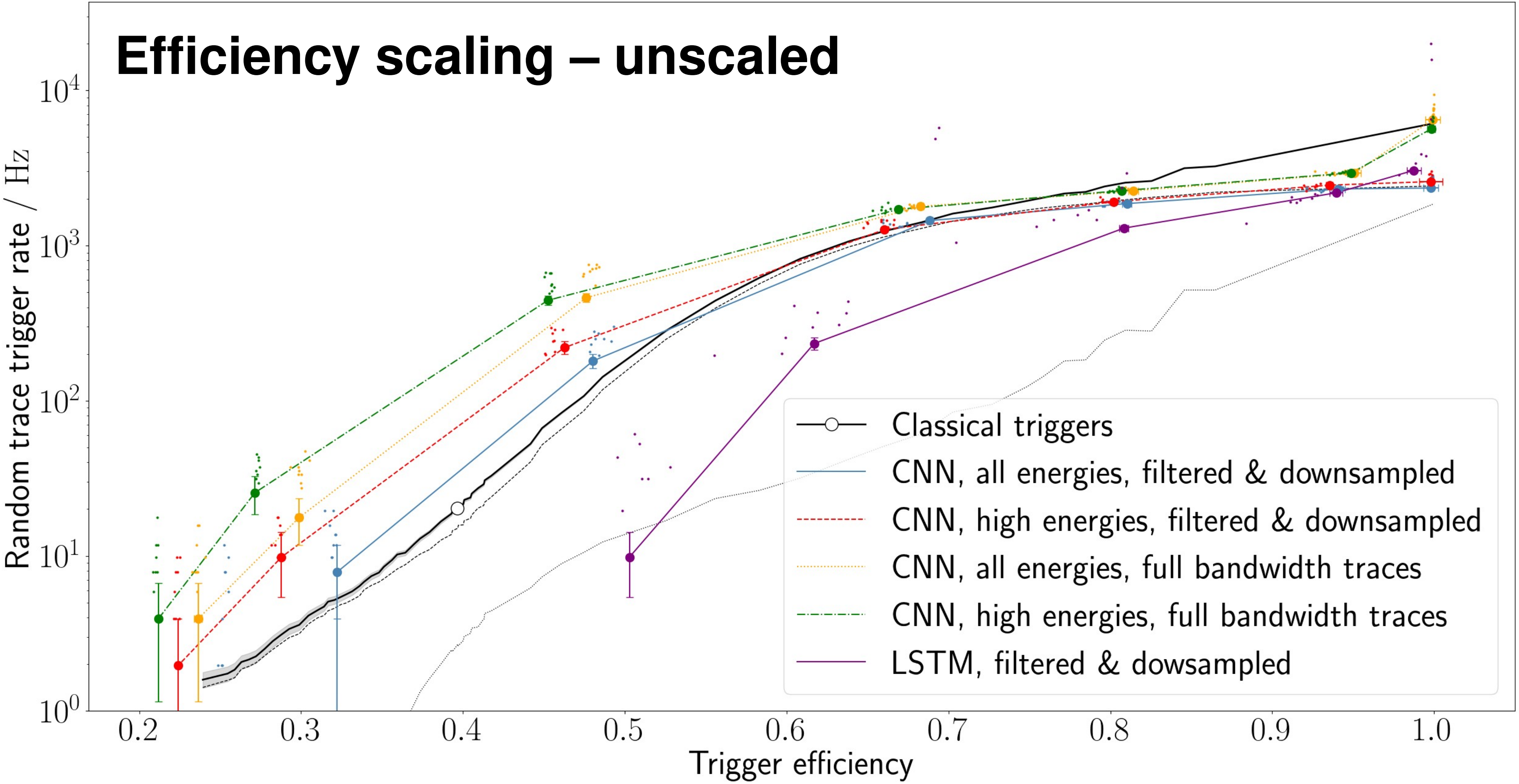




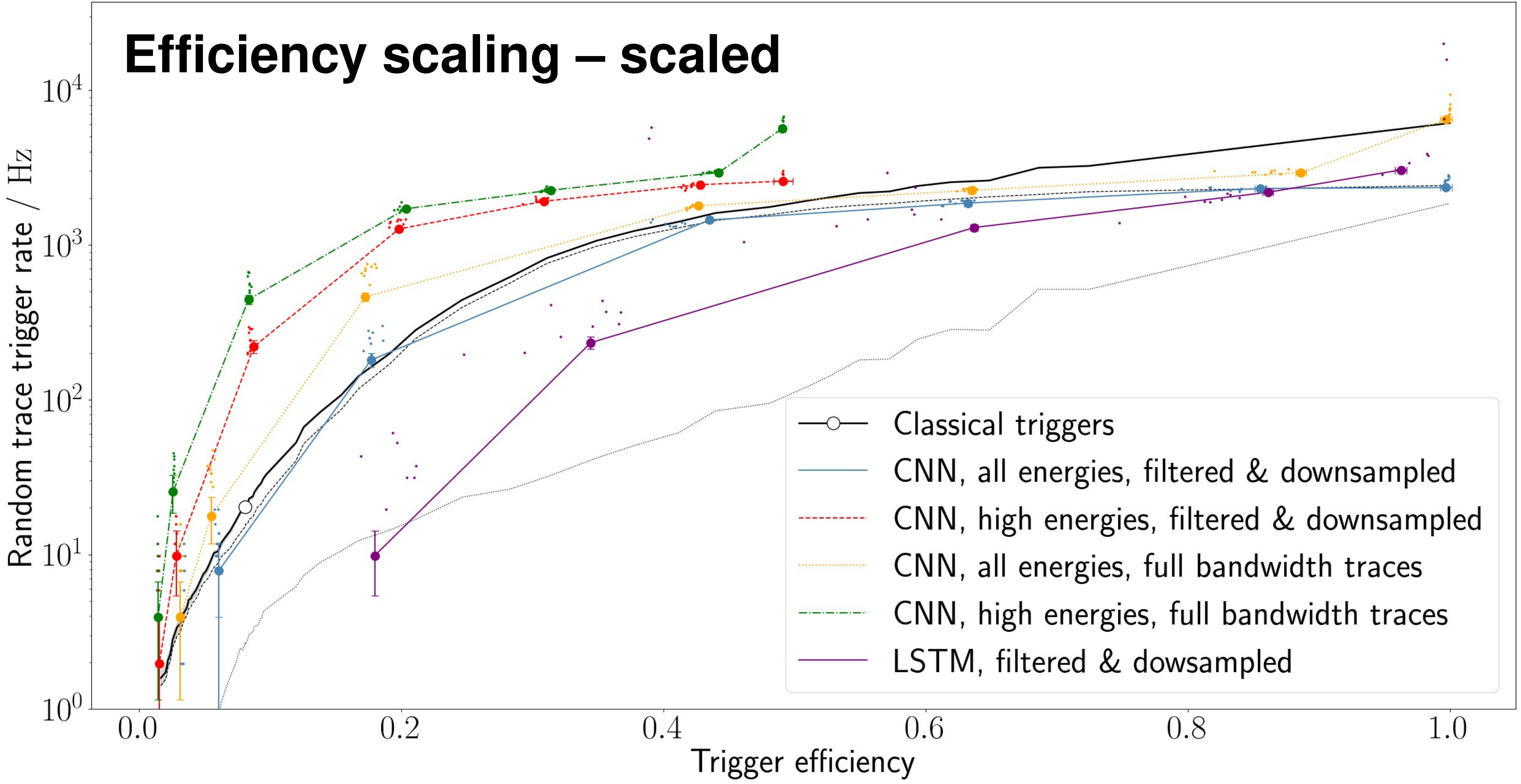
NuriaJr : 56
Peru : 1737
Granada : 1738
Jaco : 1742
LeQuiDon : 1744
Svenja : 1733
PMT 1
PMT 2
PMT 3
I_{Peak} / ADC
2022-11-14 00:00:00
2022-11-15 00:00:00
2022-11-16 00:00:00
2022-11-17 00:00:00
2022-11-18 00:00:00
2022-11-19 00:00:00
Station time / UTC


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

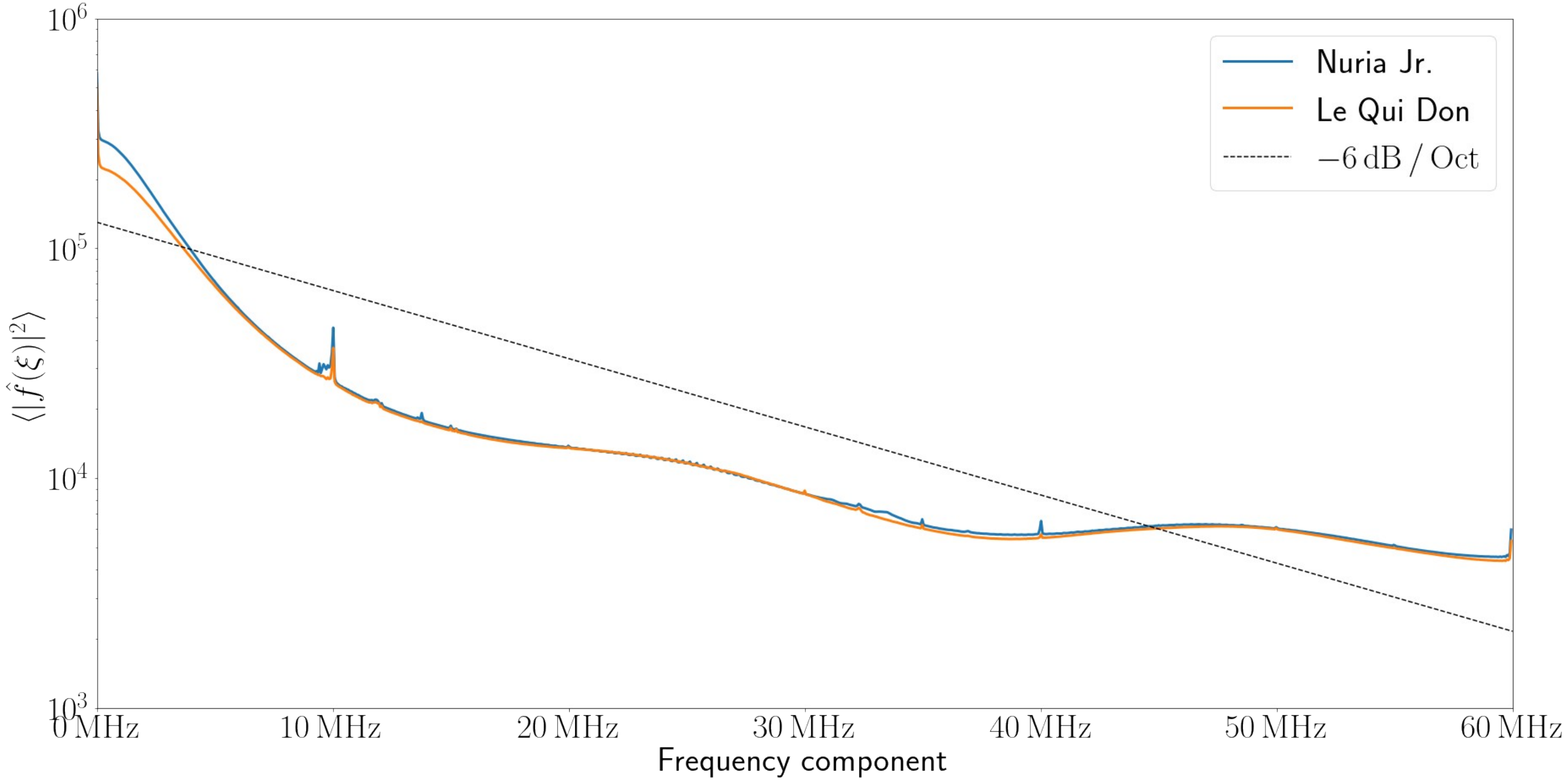
Efficiency scaling – unscaled



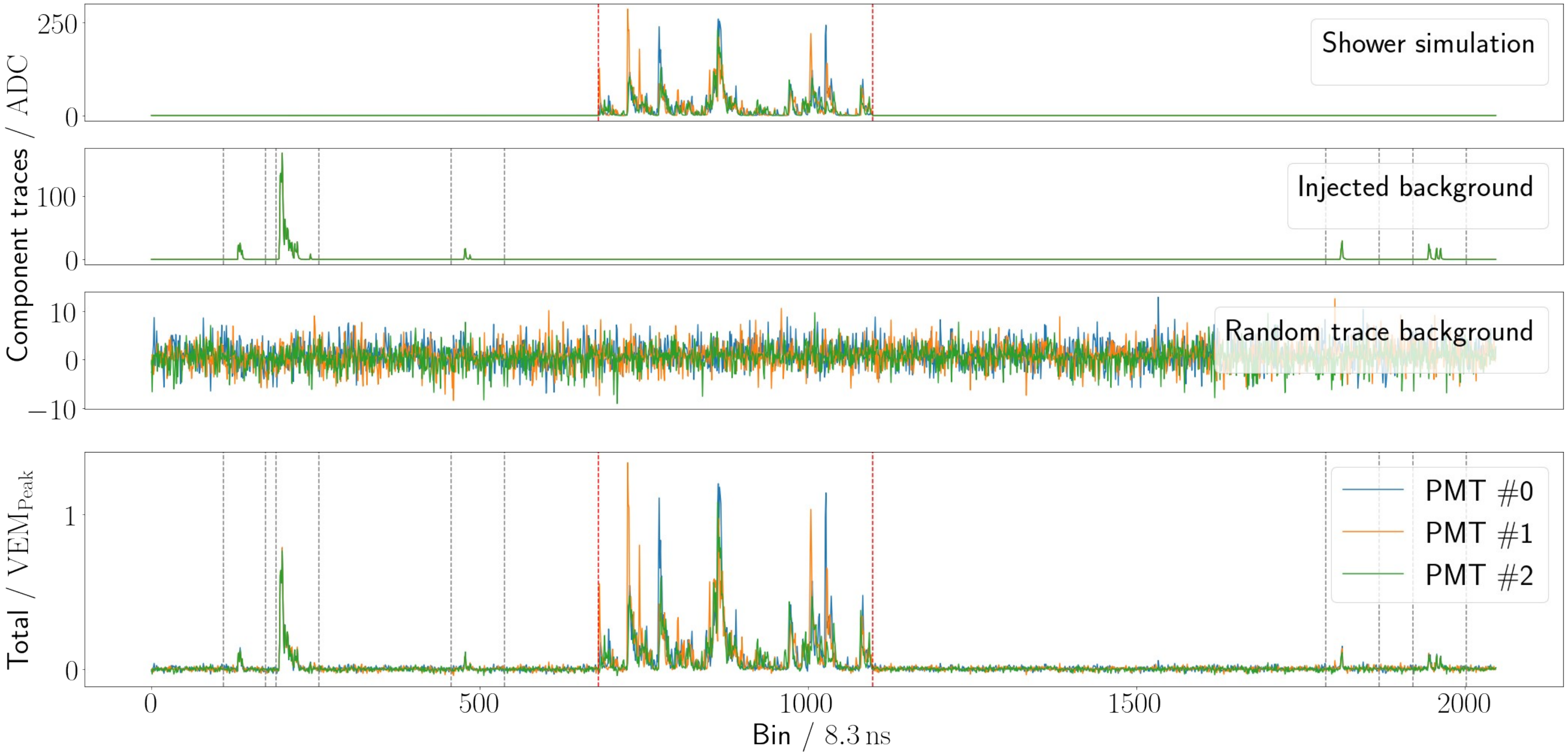
Efficiency scaling – scaled



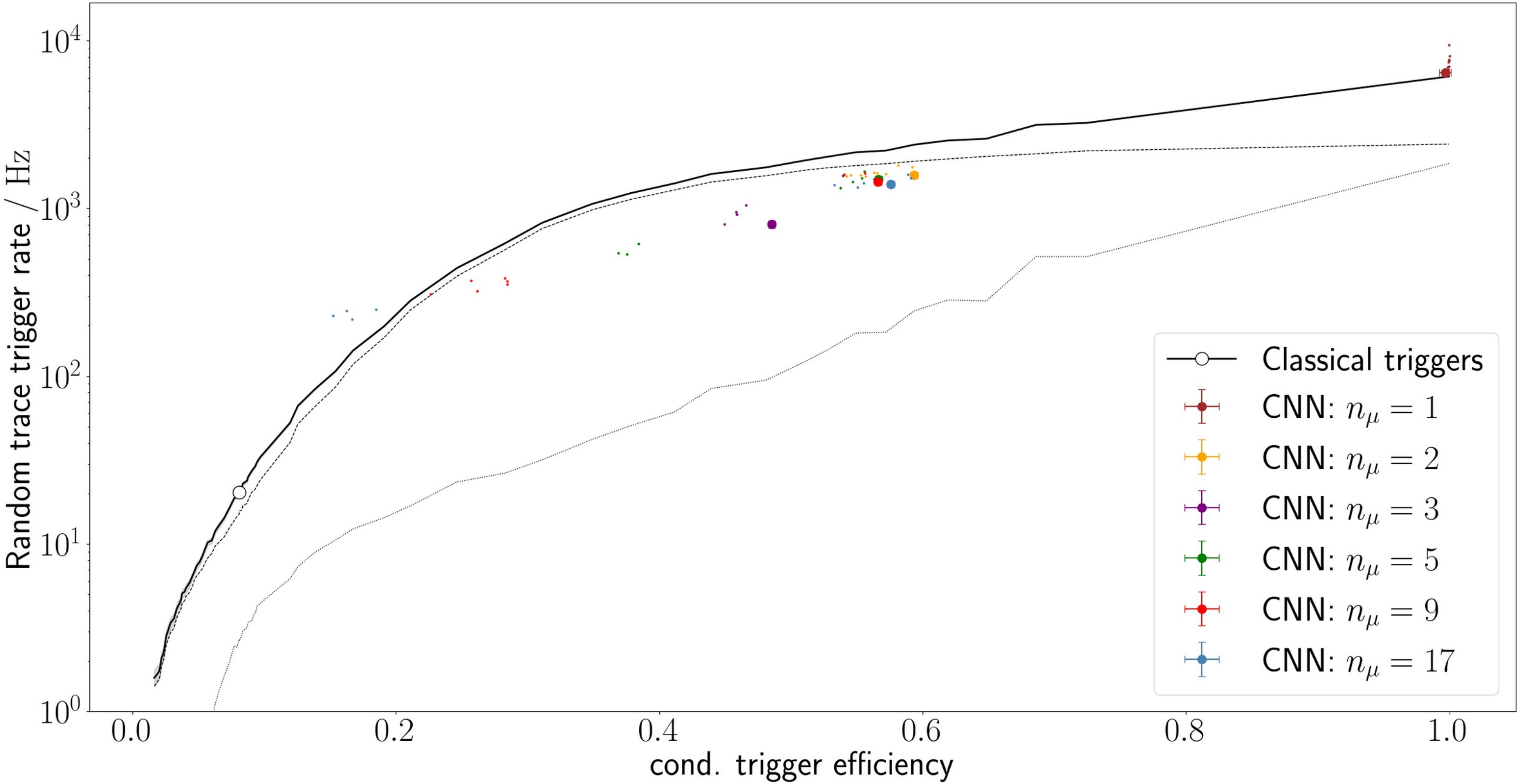
Random traces – Power spectrum



Trace building

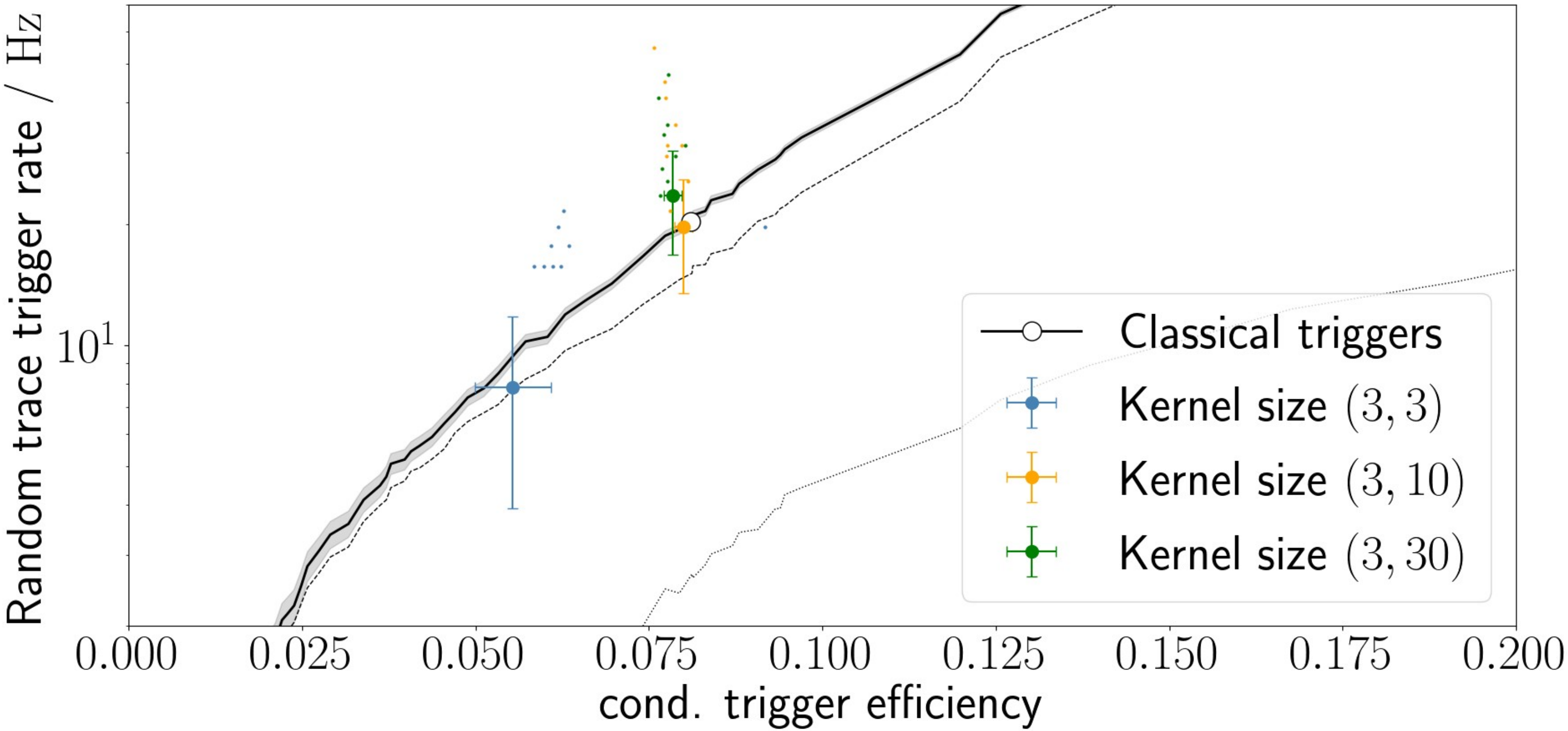


Muon cut



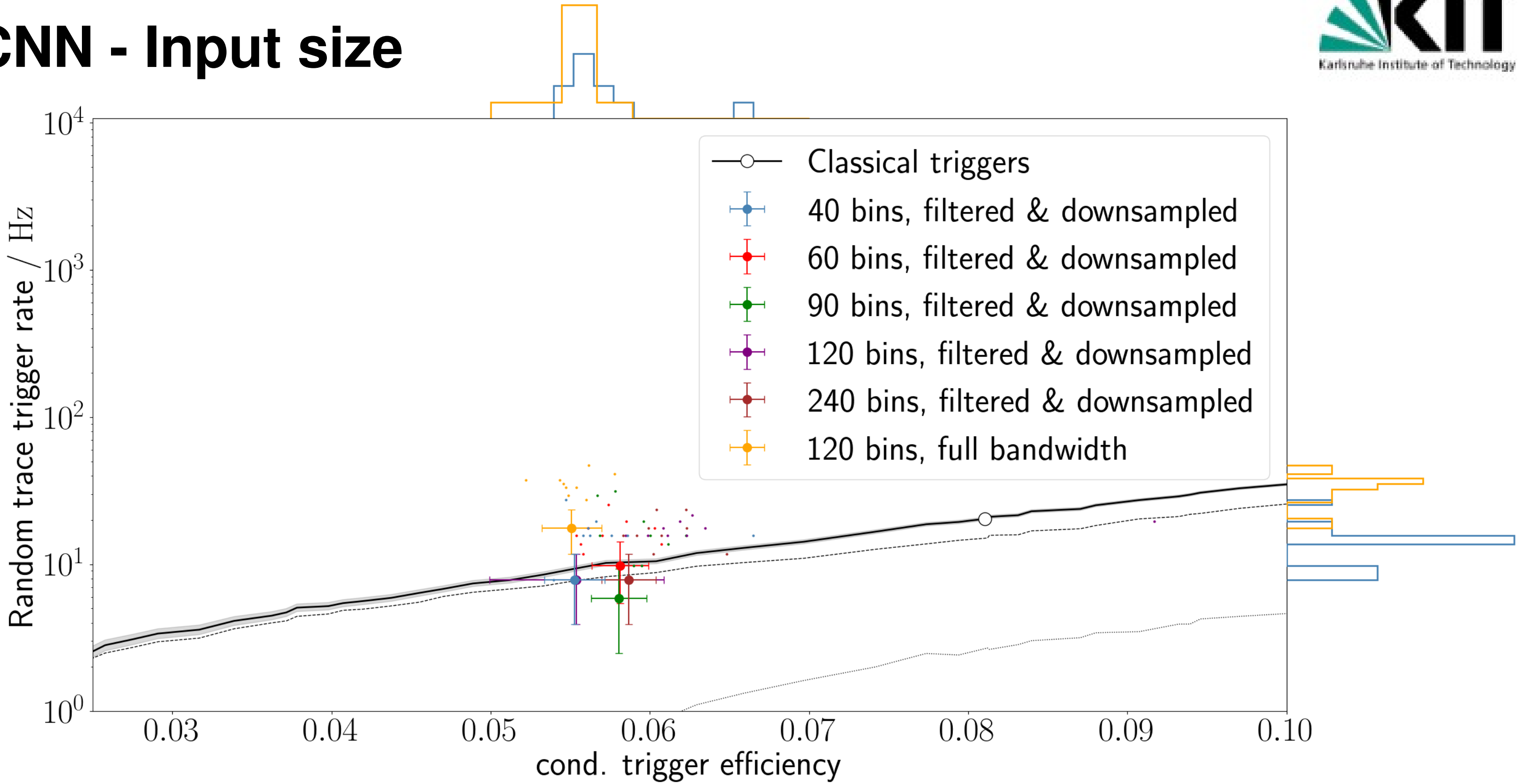
Random trace trigger rate / Hz
cond. trigger efficiency
10^0
10^1
10^2
10^3
10^4
0.0
0.2
0.4
0.6
0.8
1.0
Classical triggers
CNN: $n_\mu = 1$
CNN: $n_\mu = 2$
CNN: $n_\mu = 3$
CNN: $n_\mu = 5$
CNN: $n_\mu = 9$
CNN: $n_\mu = 17$

Kernel size

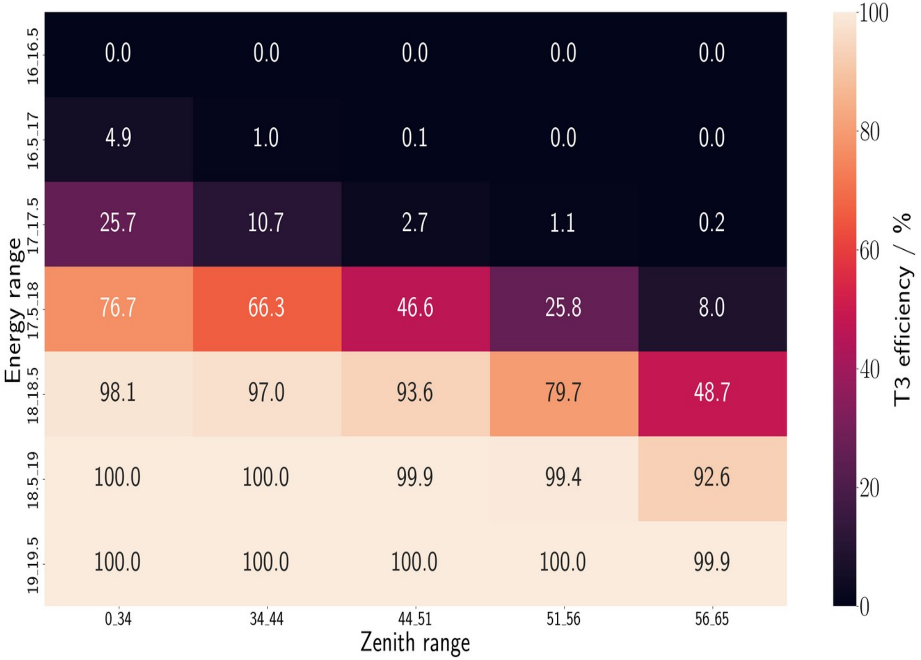


Random trace trigger rate / Hz
10^1
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
cond. trigger efficiency
Classical triggers
Kernel size $(3,3)$
Kernel size $(3,10)$
Kernel size $(3,30)$

CNN - Input size



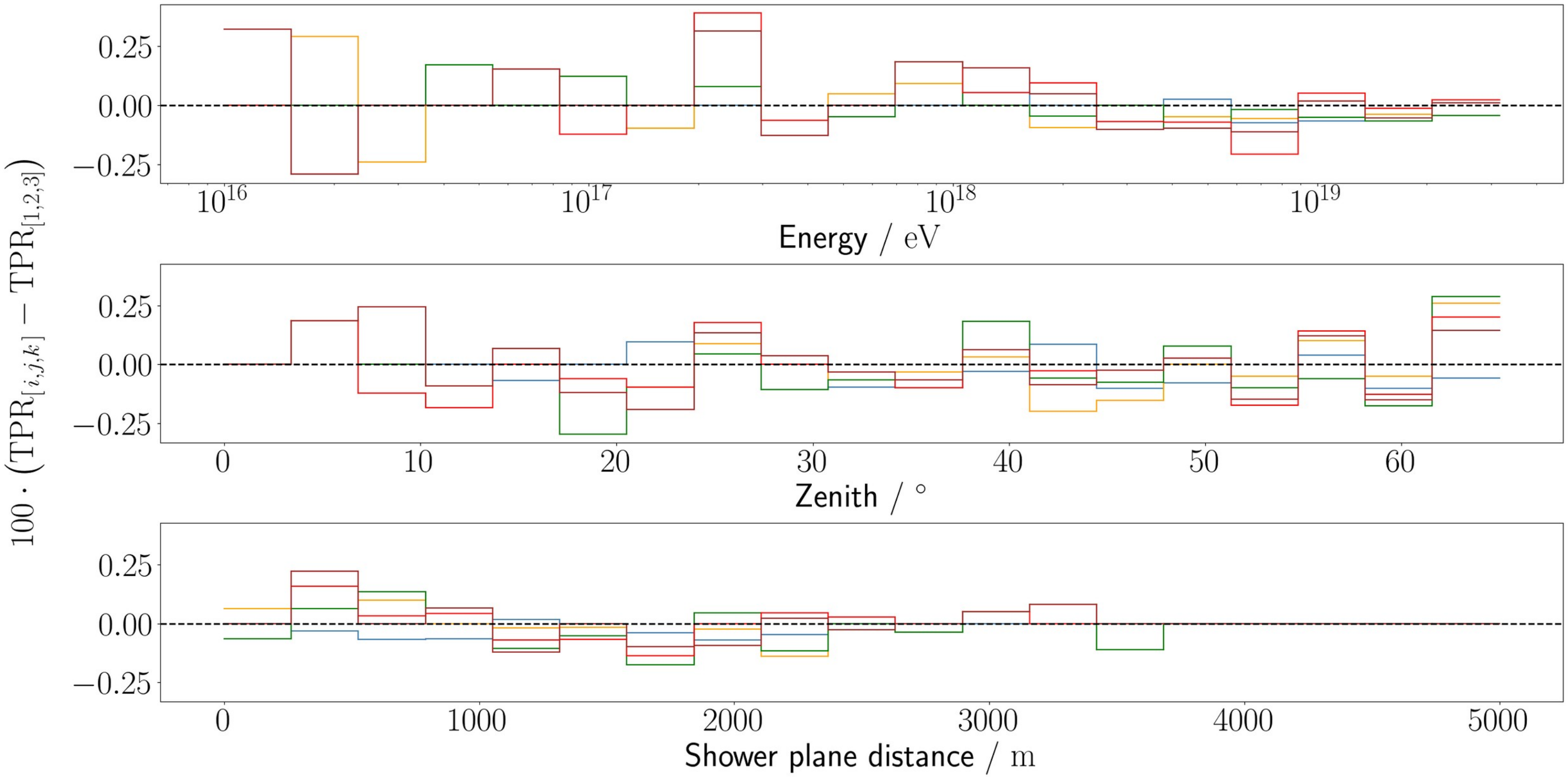
T3 efficiency calculation



Offline approach

Bayesian folding

LSTM permutations



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**