QCAMi on Tenerife: Comparing Data to Simulations

CTAO | SST COLLABORATION

Adelaide University

Simon Lee
July 2026

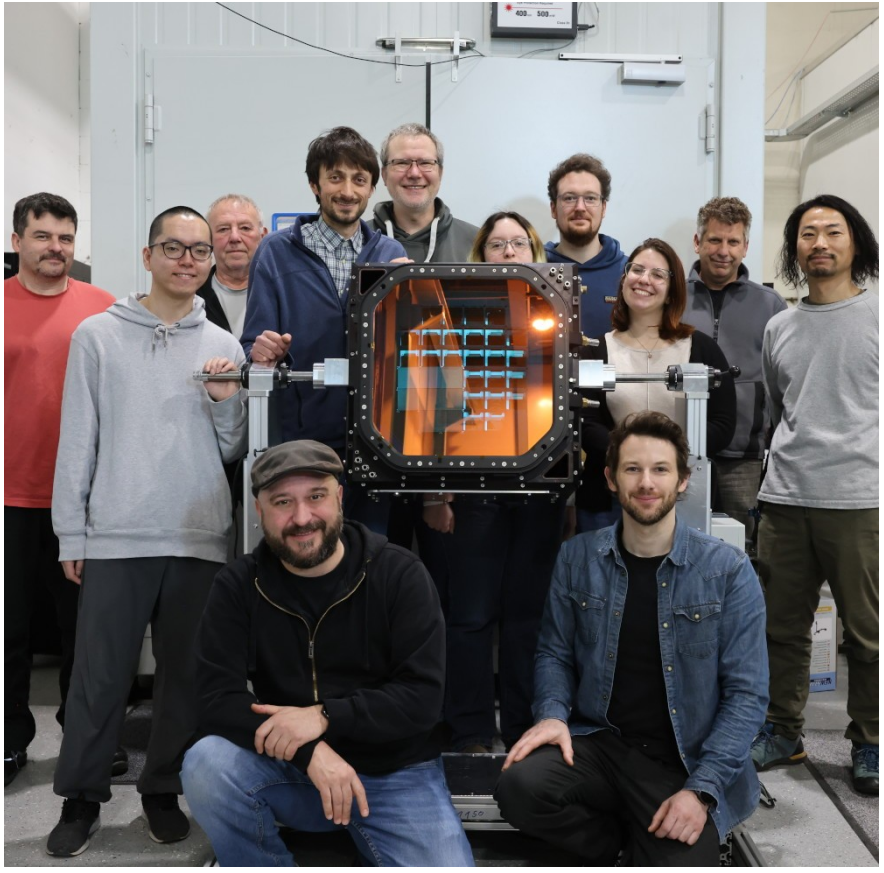
My titles will be a summary sentence of the slide

Presentation structure:

QCAM-i, the Tenerife campaign, and my goals:	3 slides
Lab calibration:	6 slides
Trigger investigation & MC setup:	9 slides
Results:	6 slides

QCAM-i, the Tenerife campaign, and my goals

The SSTCAM team built a quarter-camera



- Called “QCAM-i” (i for initial)
- No backplane (pixel modules are independent)
- 512 pixels (2048 in full camera)
- ~4° FoV
- Full mechanics, power, cooling

SSTCAM team with QCAM-i
[Richard White]

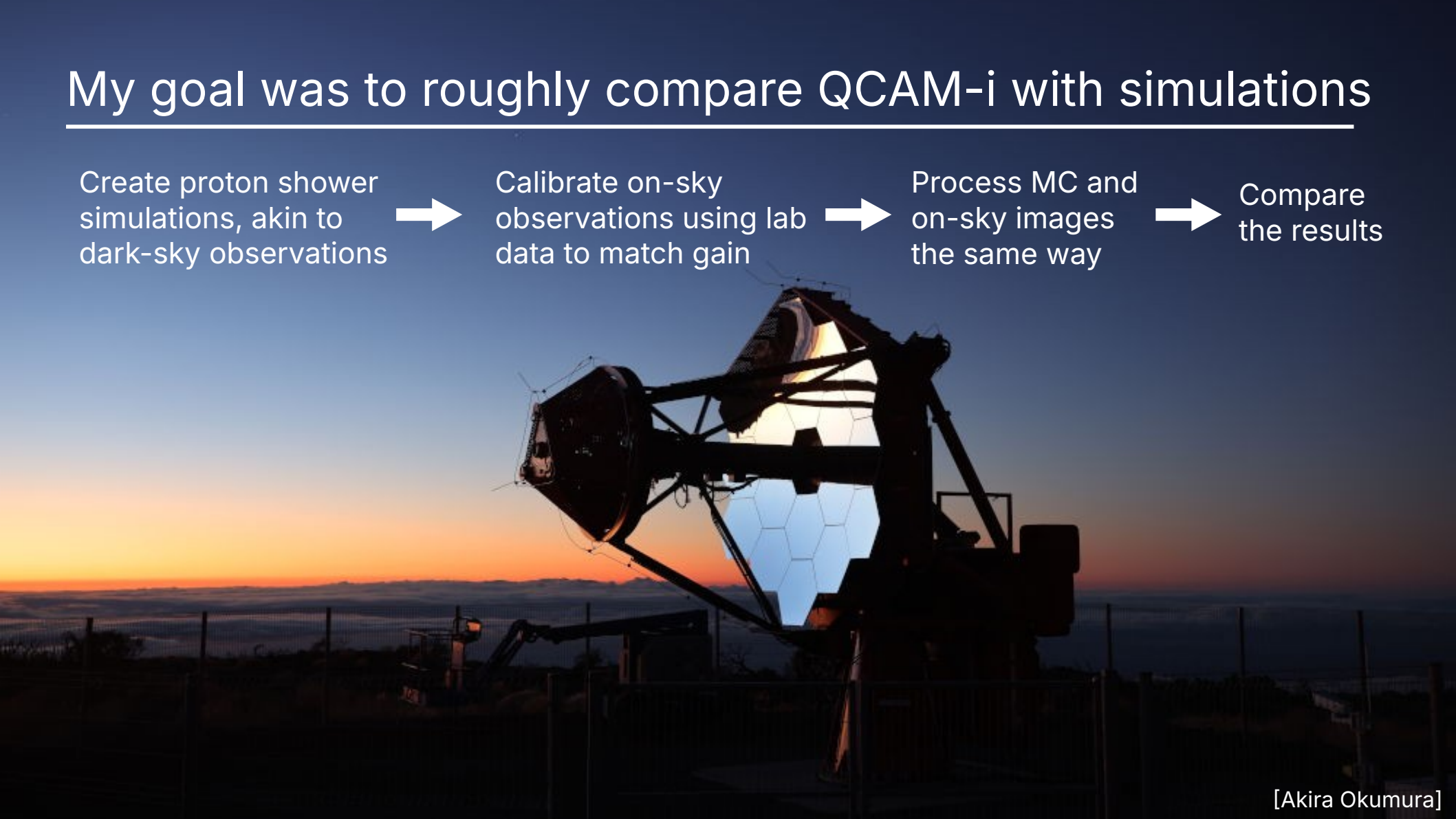
The SSTCAM team tested a QCAM-i on Tenerife

- July 7th – July 30th 2025
- QCAM-i was placed on ASTRI-2 on July 22nd
- Included chiller from Australian manufacturer LaserChill
- Several nights of observation, including dark sky, gamma-ray sources etc.

ASTRI-2 telescope with SST QCAM-i camera [Akira Okumura]

My goal was to roughly compare QCAM-i with simulations

Create proton shower simulations, akin to dark-sky observations → Calibrate on-sky observations using lab data to match gain → Process MC and on-sky images the same way → Compare the results

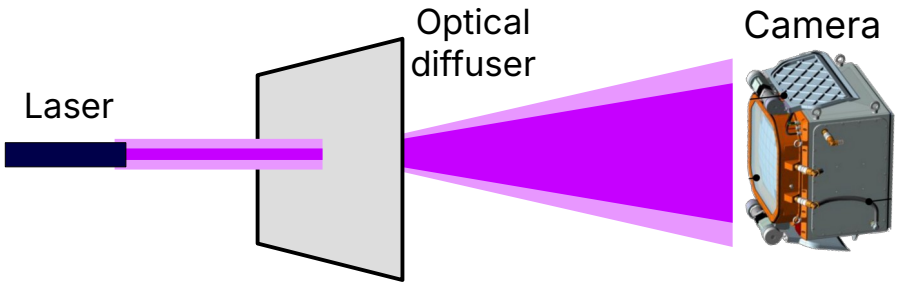


[Akira Okumura]

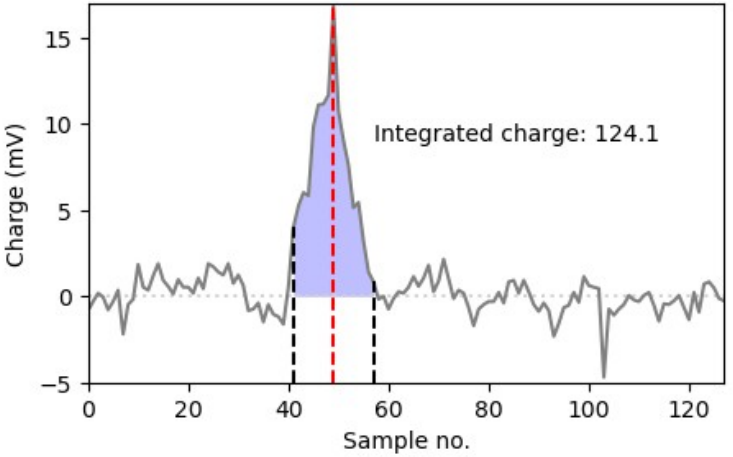
Lab calibration

In the lab, a laser flashes the camera to test the pixels

1: The camera is flashed with a calibrated laser

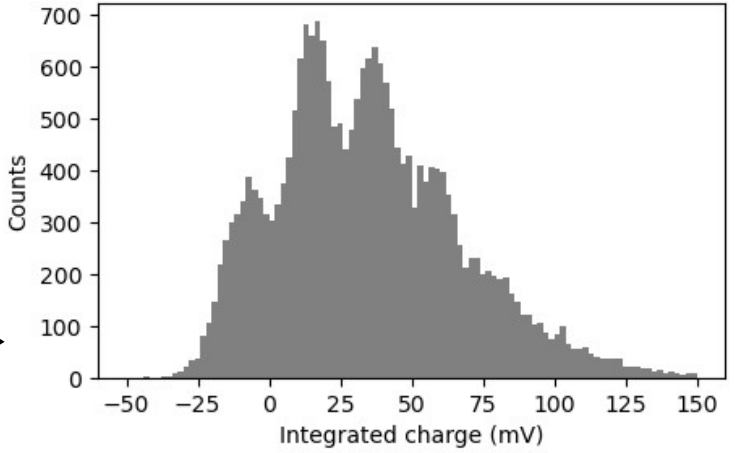


2: Example waveform and charge extraction

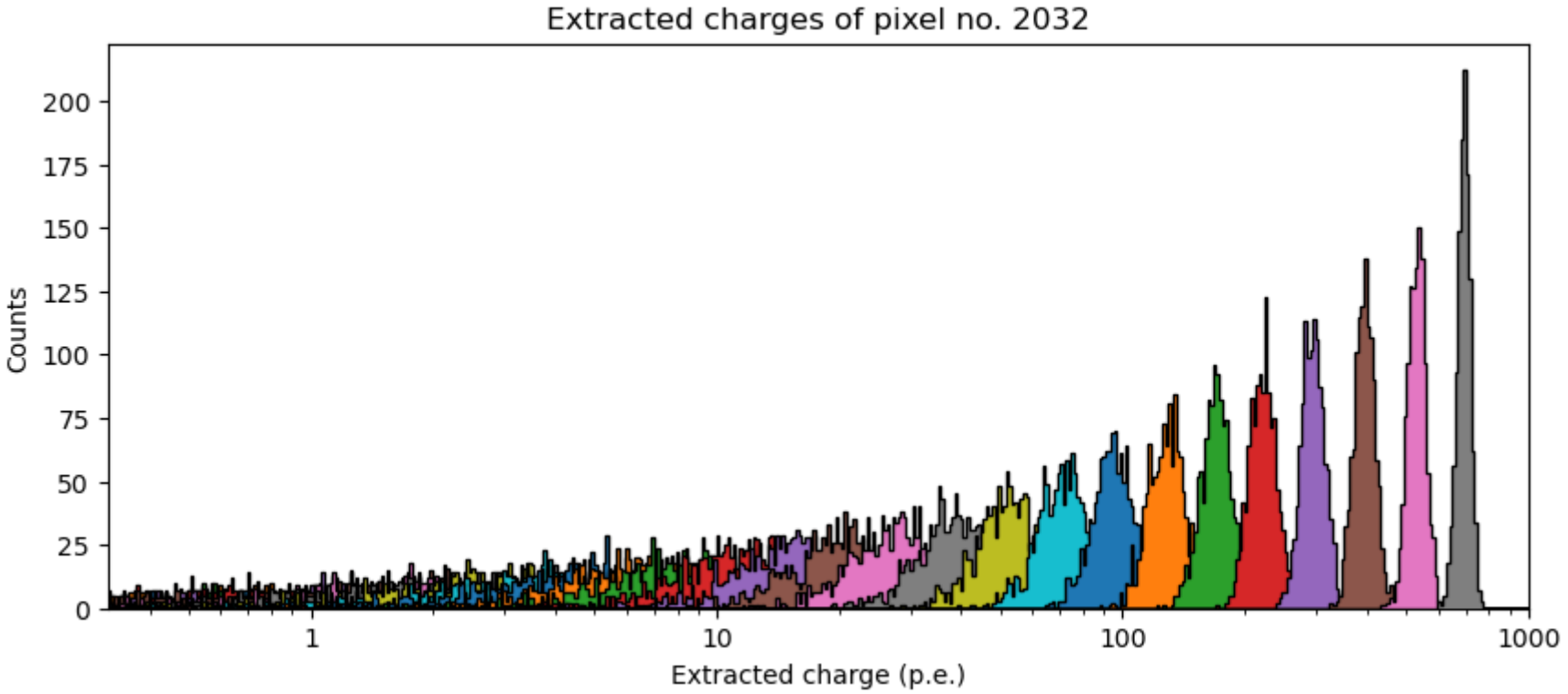


20k times

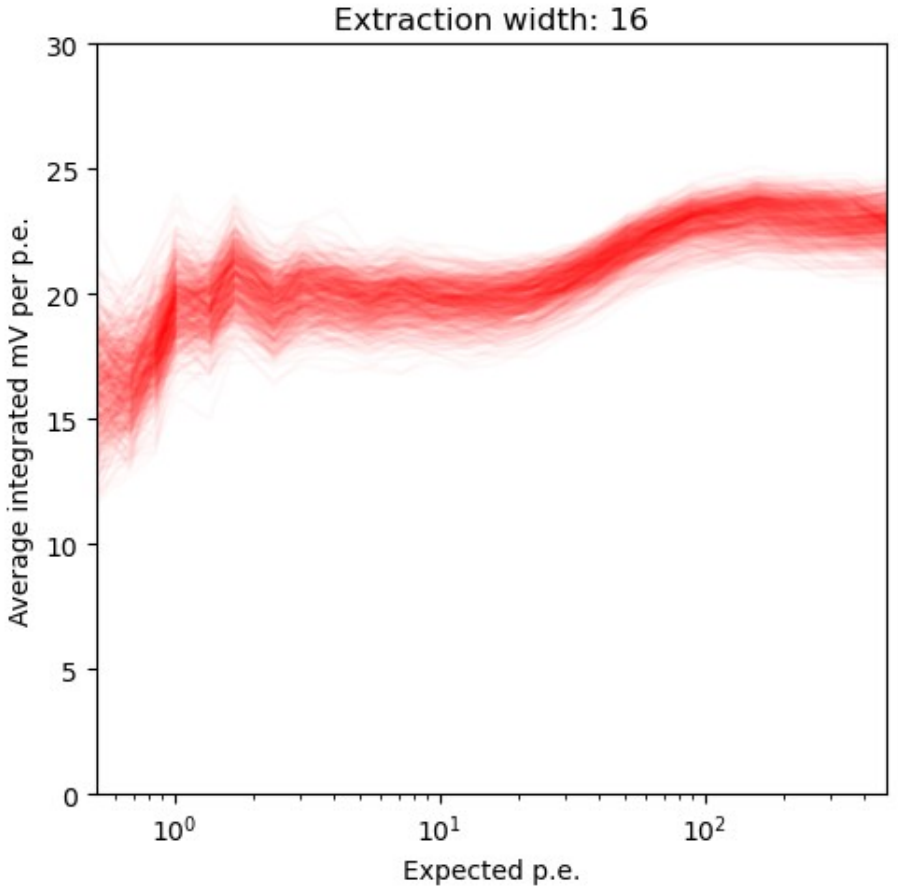
3: Distribution of extracted charges (for one pixel)



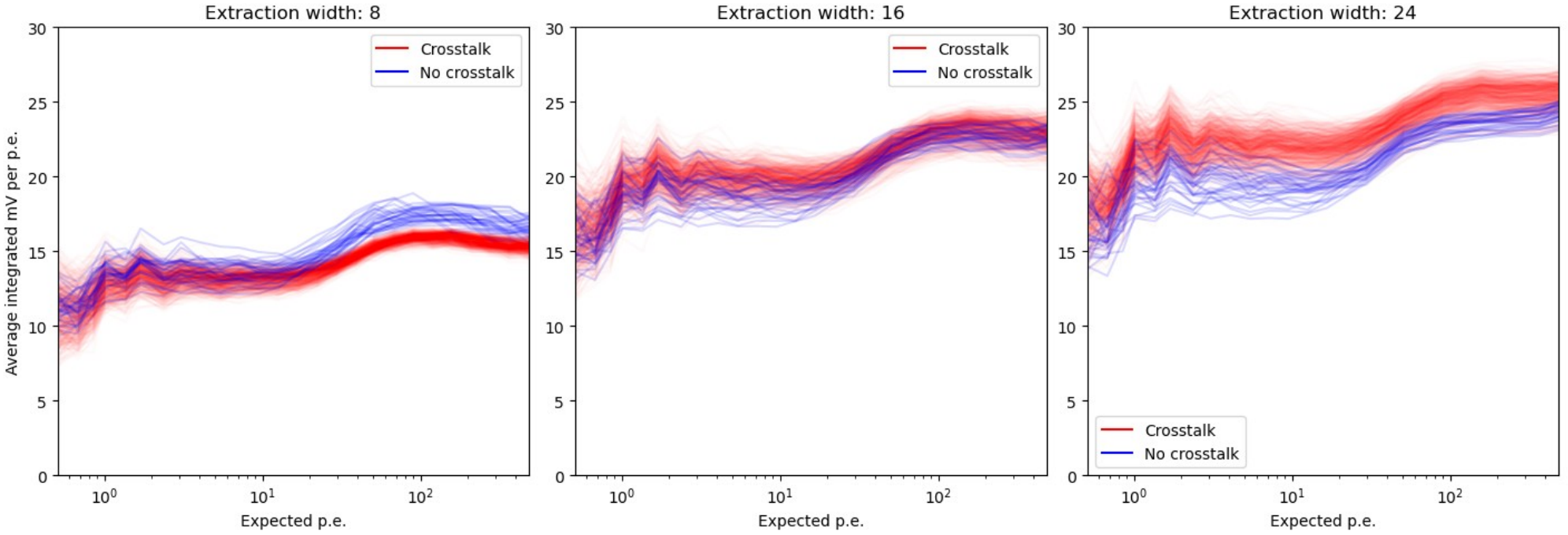
Laser flashes were done over a few orders of magnitude



The extracted value per expected p.e. is not linear



I chose an extraction that balances out cross-talk effects



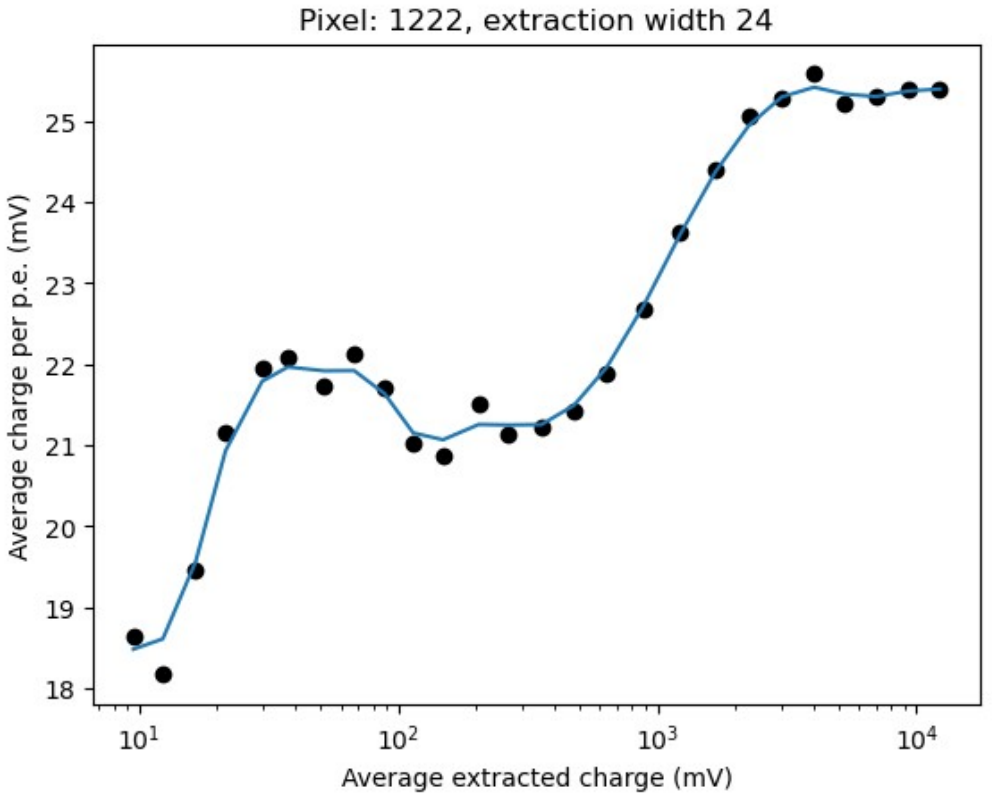
With a smaller width, less charge is extracted when there is crosstalk.

16 samples is the Goldilocks number apparently.

With a larger width, more charge is extracted when there is crosstalk.

I decide to use a per-pixel “transfer function”

Input: 230 integrated mV



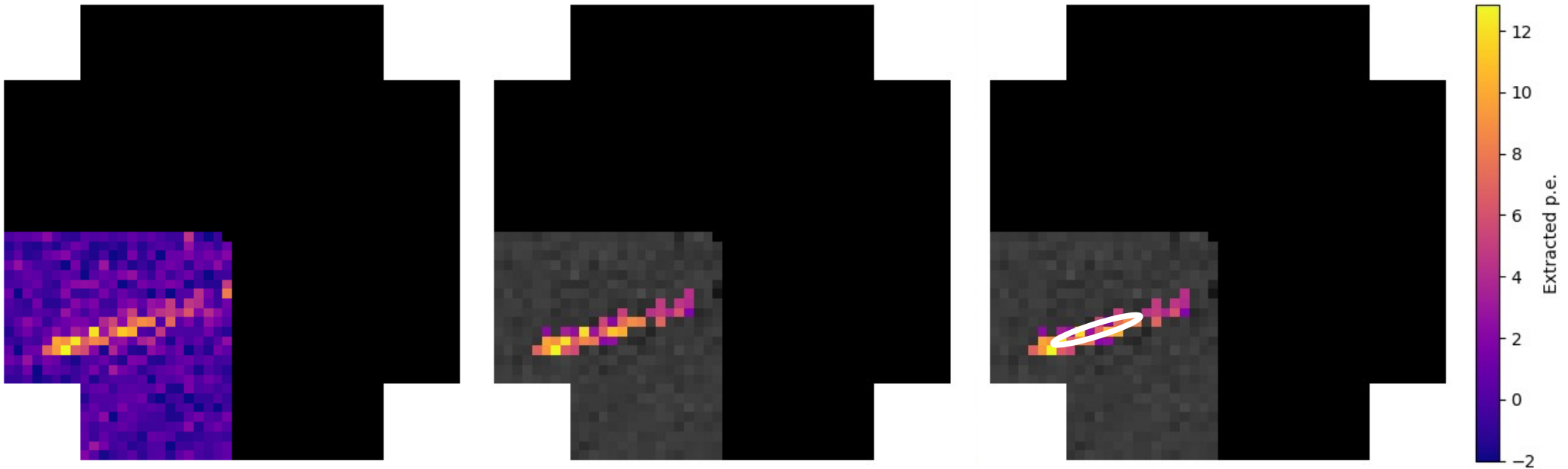
Output: 10.8 p.e.

Images were extracted, cleaned, and parameterised

1: Extracted image

2: Cleaned with “TailCuts”

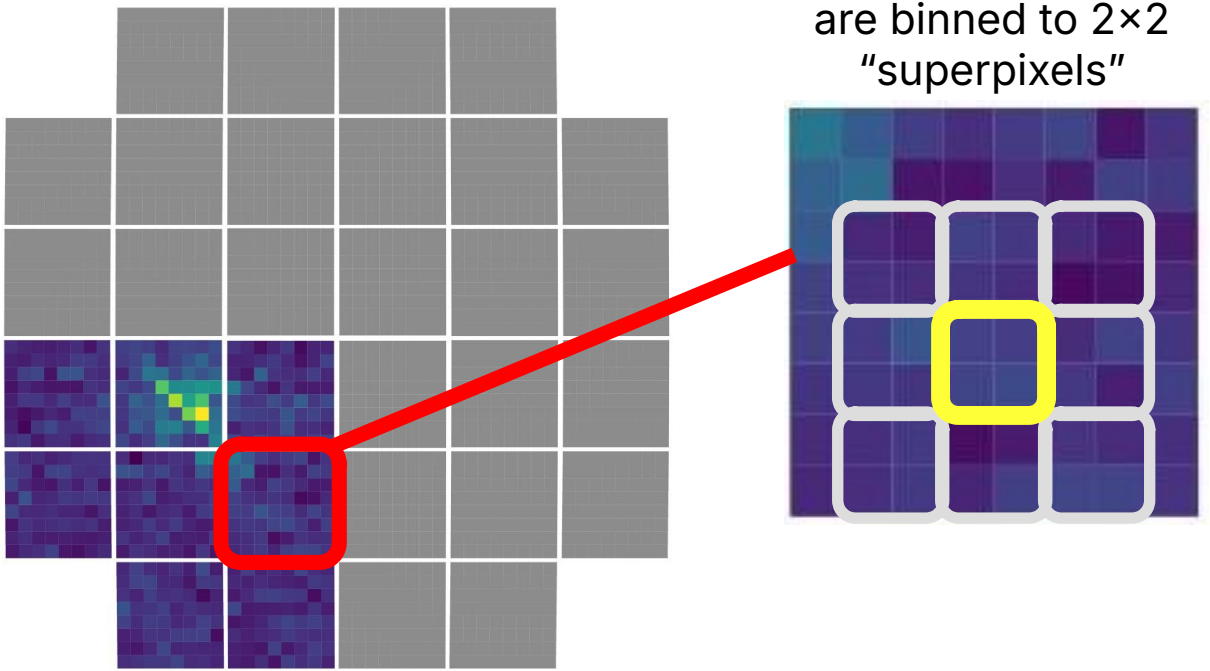
3: Get Hillas parameters



Trigger investigation & MC setup

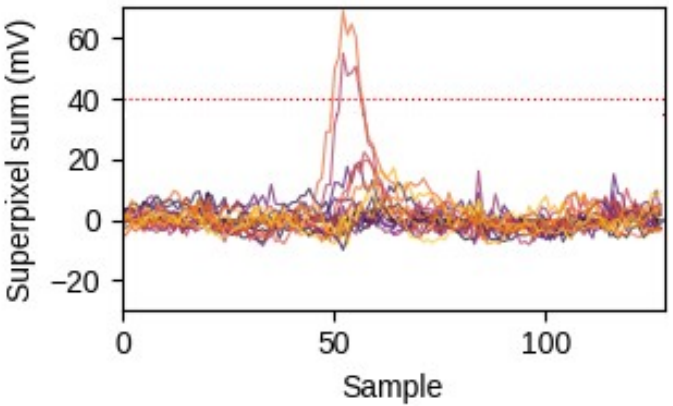
We next wanted to check the trigger

The camera is made up of modules with 8×8 pixels

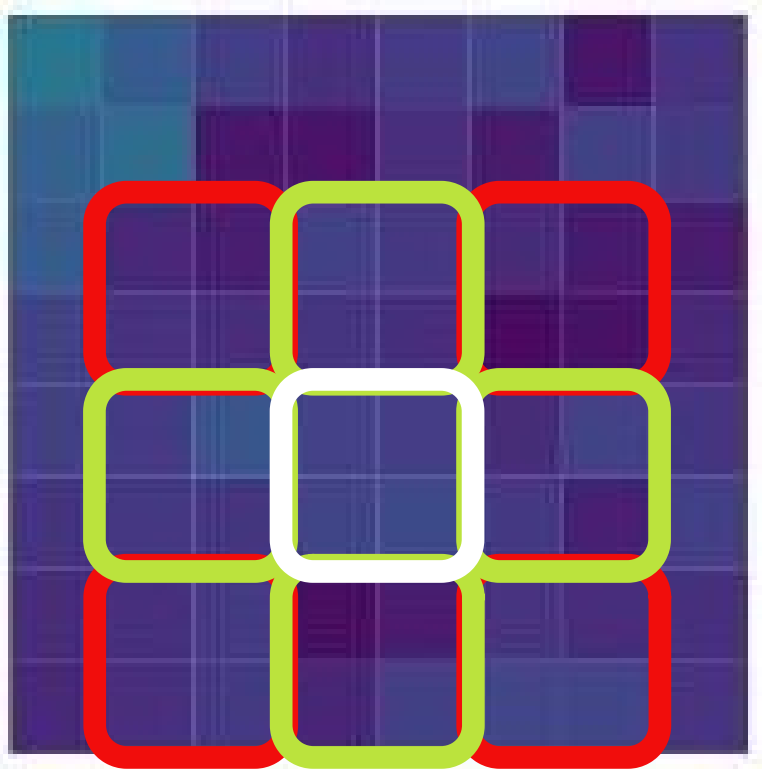


For trigger, the pixels are binned to 2×2 “superpixels”

If two superpixels simultaneously surpass the threshold, it should trigger (i.e. save the data)



We noticed no diagonal superpixel pairs triggered

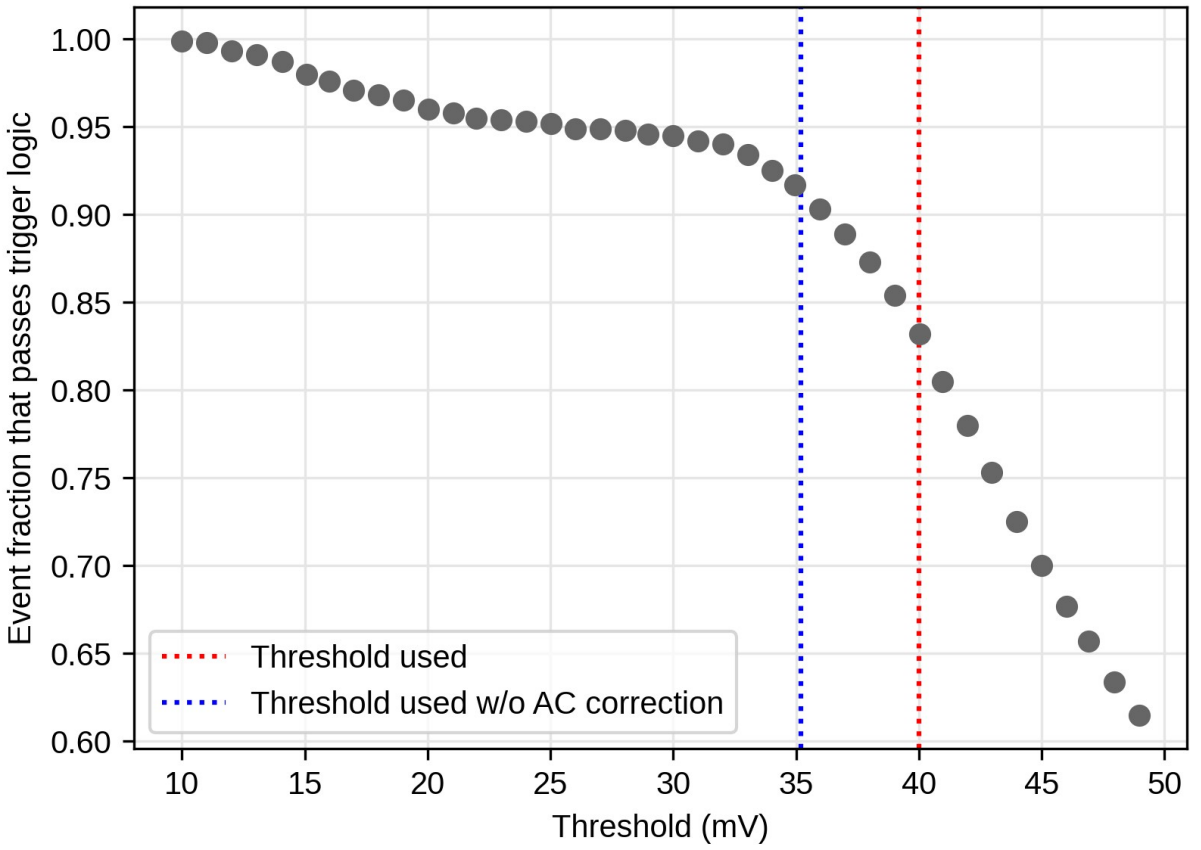


Turned out to be a code typo, needed to add in an extra argument

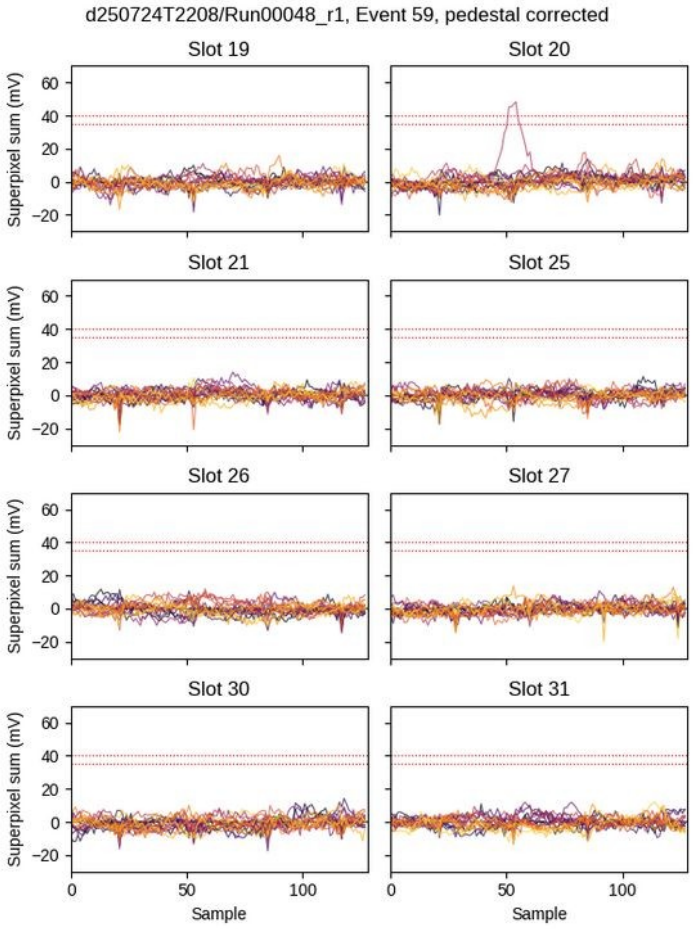
```
mapping_sm.get_neighbours(diagonal=True)
```

I adjusted the simulations to match

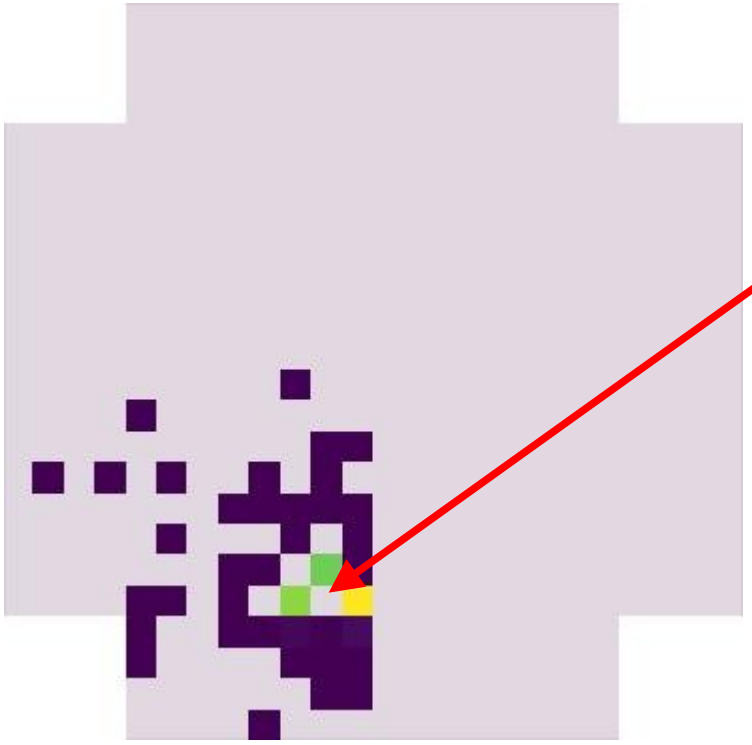
We also noticed ~5% of events didn’t pass trigger logic



Almost all these events involved one specific superpixel



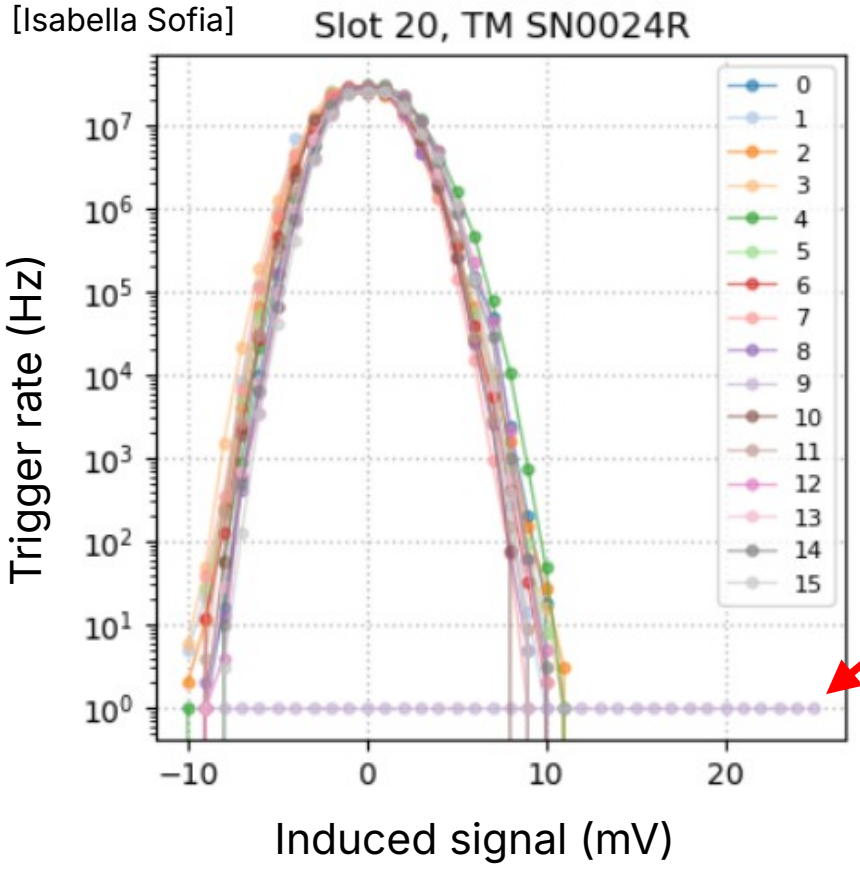
Distribution of solo-firing superpixels



Almost all event that “should not” have triggered, have pulses in SPs adjacent to this specific SP

Turns out, this superpixel trigger was malfunctioning

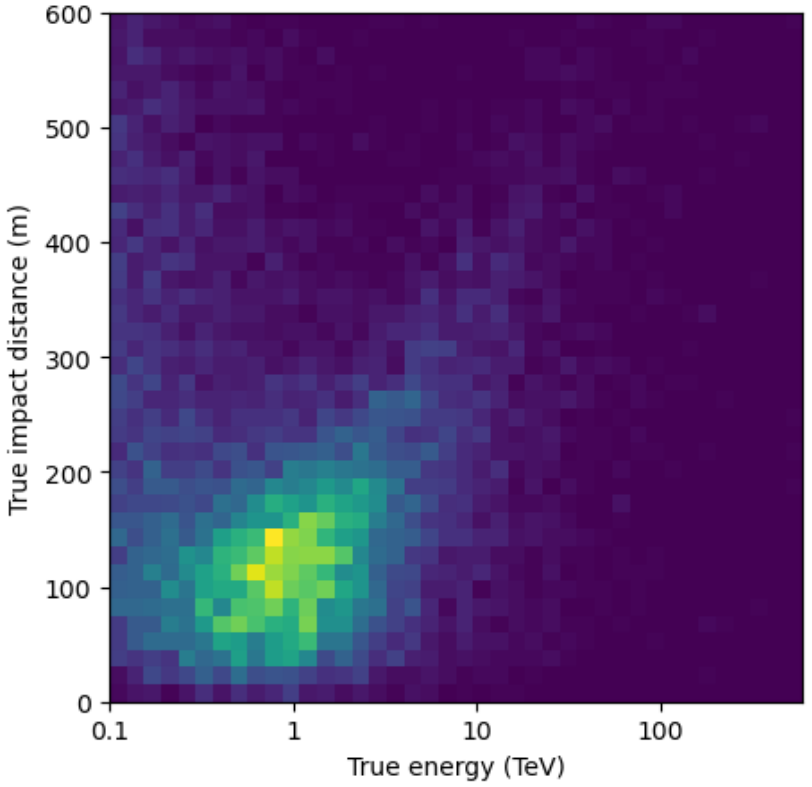
[Isabella Sofia]



Always sending trigger, regardless of signal

I adjusted the simulations to match

I simulated proton showers with this modified trigger



- 0.1–600 TeV
- -2.7 spectral index
- 20 deg zenith, 10 deg viewcone
- 1200m scatter radius
- 50k showers * 50 reuse * 100 runs = 250M simulated showers
- Equates to ~130s of observation time
- `./prod6_single_run SST LaPalma proton North`

I compared with dark field, clear sky observations

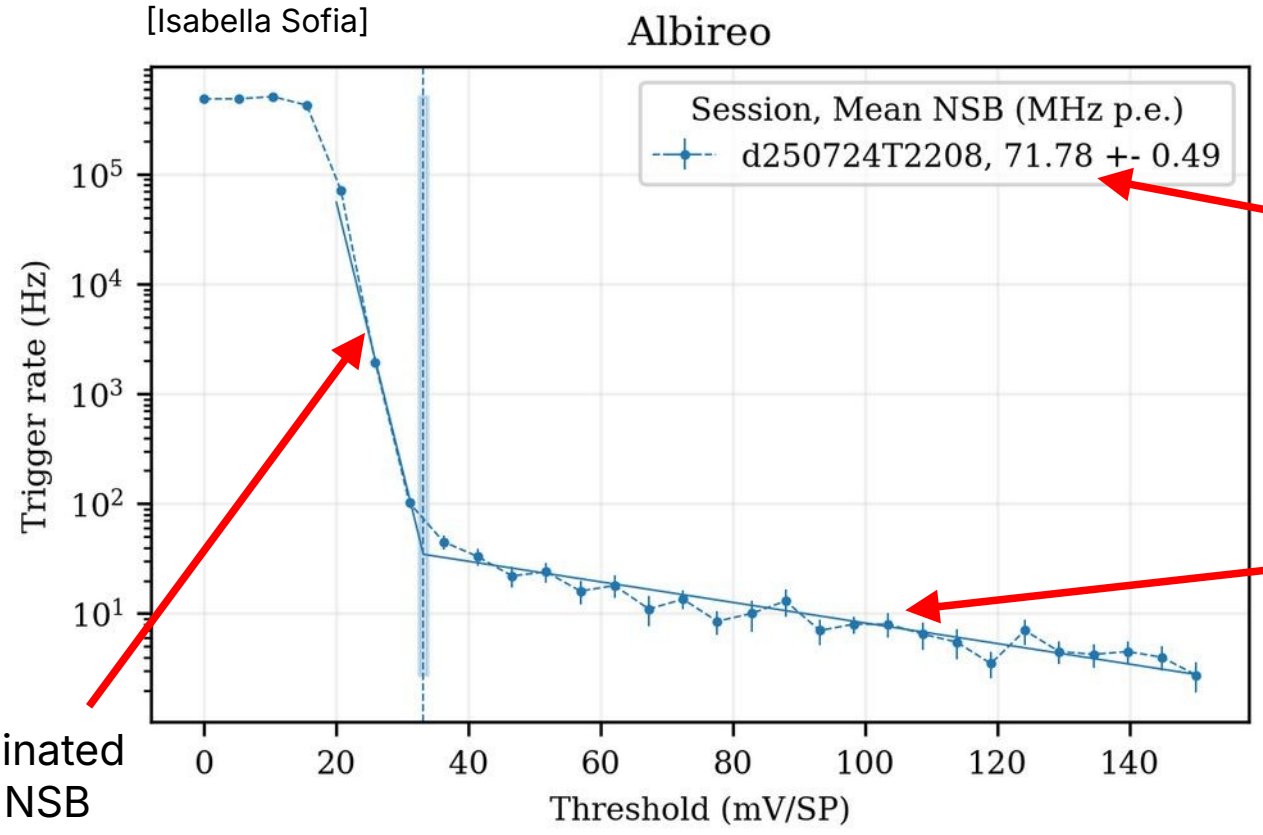
3 nights
10 runs
1.02 hours
96k triggers

of clear-sky,
dark field observations

ASTRI-2 at night
[Akira Okumura]

We also wanted to compare to Tenerife threshold scans

[Isabella Sofia]

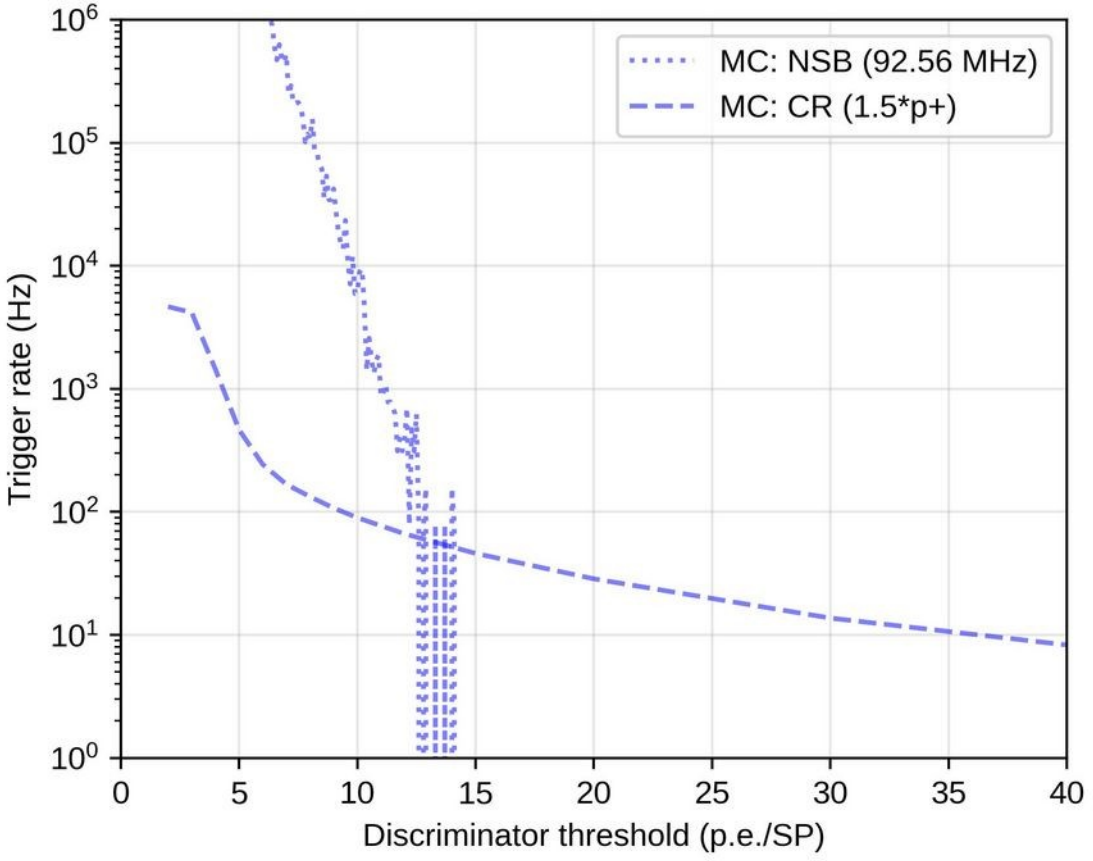


Different threshold scans were done at times with different Night Sky Background (NSB) rates

Dominated by cosmic rays

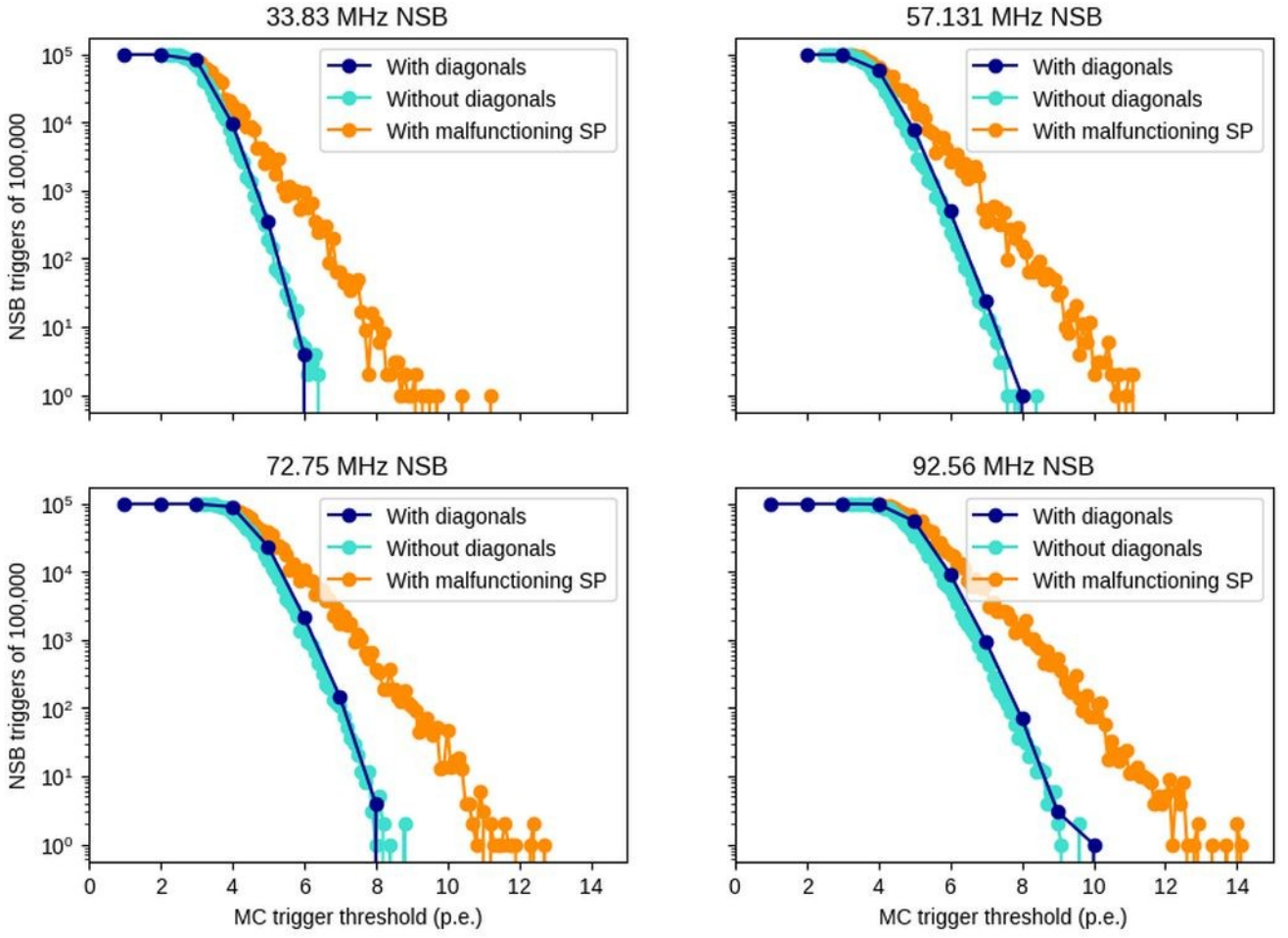
Dominated by NSB

I simulated protons and NSB separately

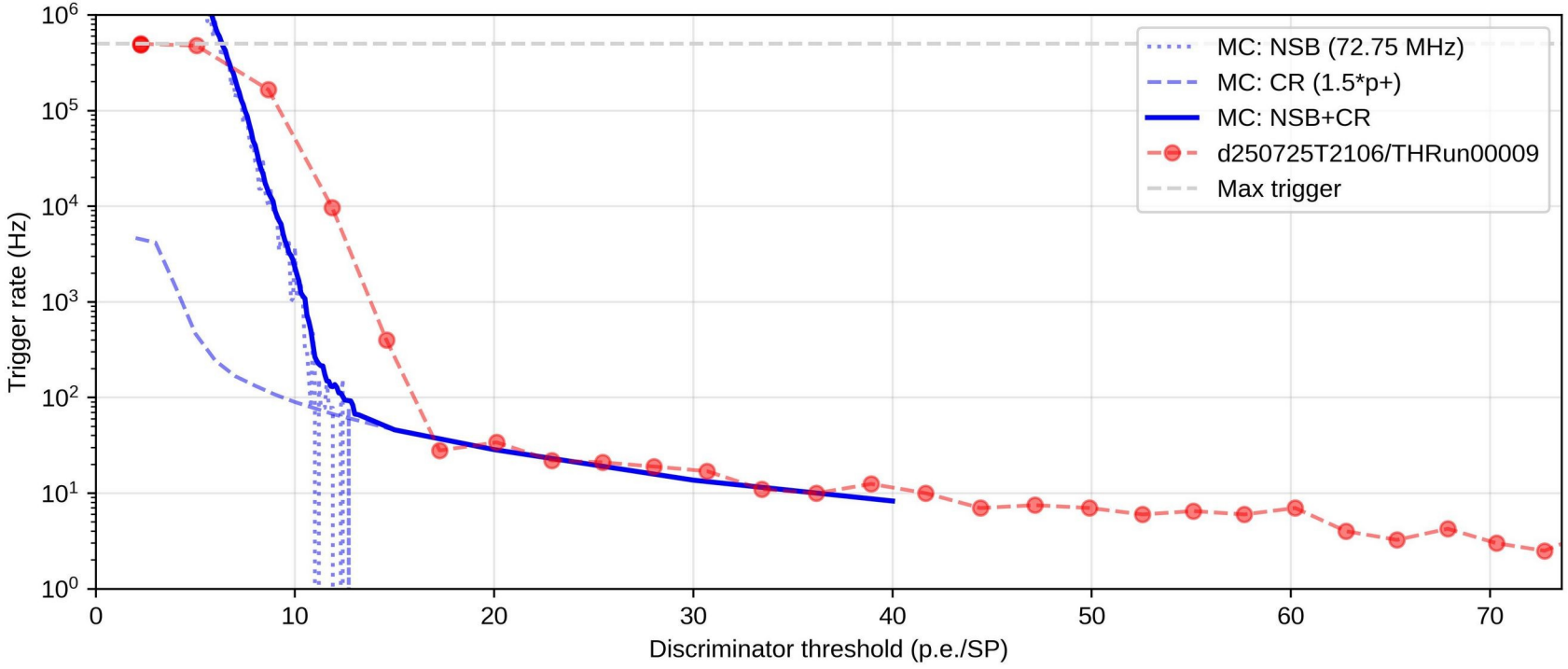


Results

Trigger changes made a big difference to simulations

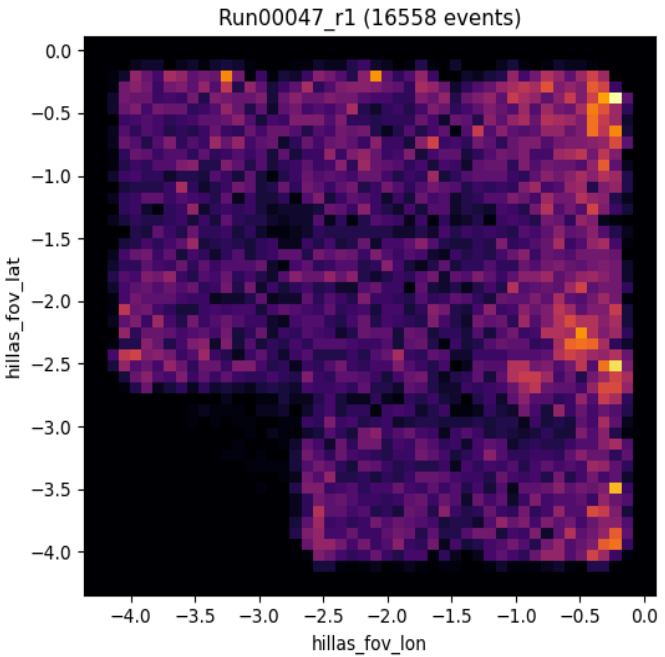


Here’s how simulations compare to real threshold scans

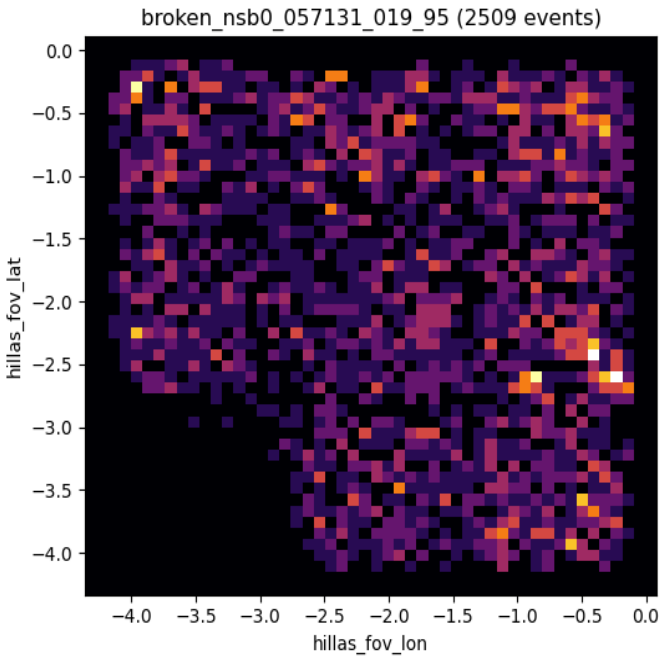


Here are distributions of the extracted centres of gravity

One on-sky run



Simulation

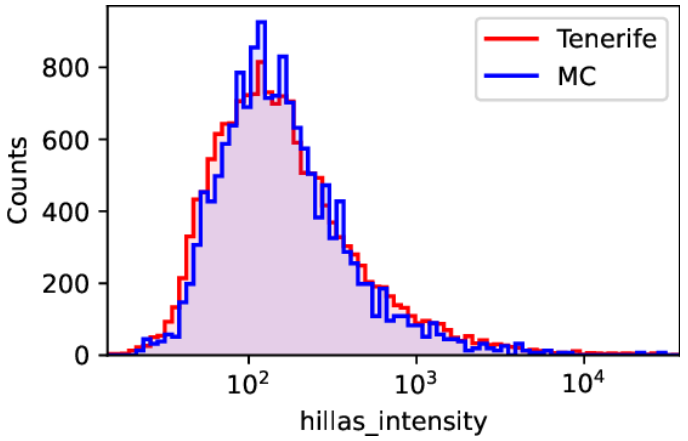


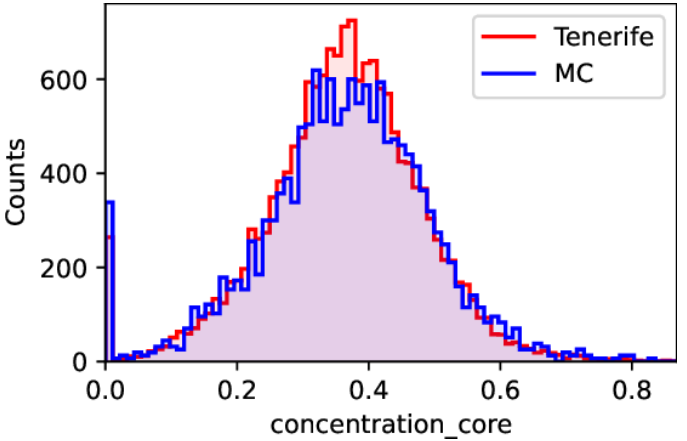
Note:

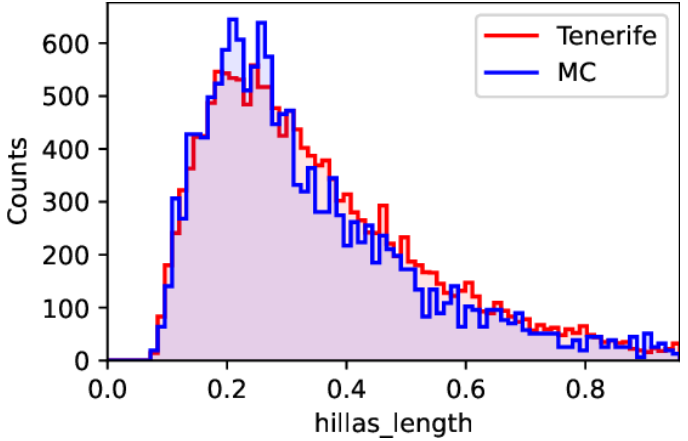
- “Pillows” of disjoint target modules

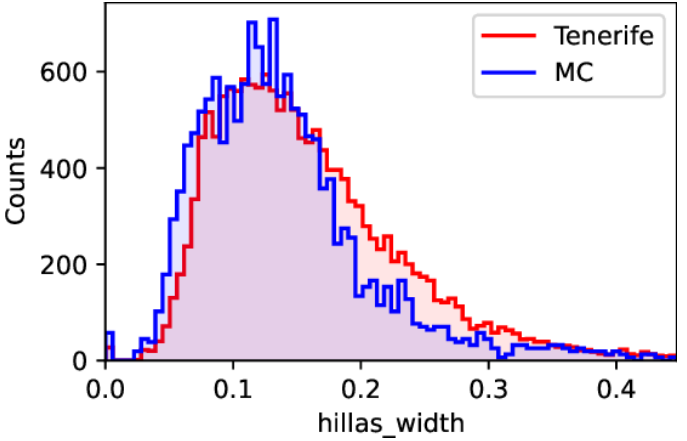
- Cluster of three points around superpixel with malfunctioning trigger

Here’s how the Hillas parameters compare

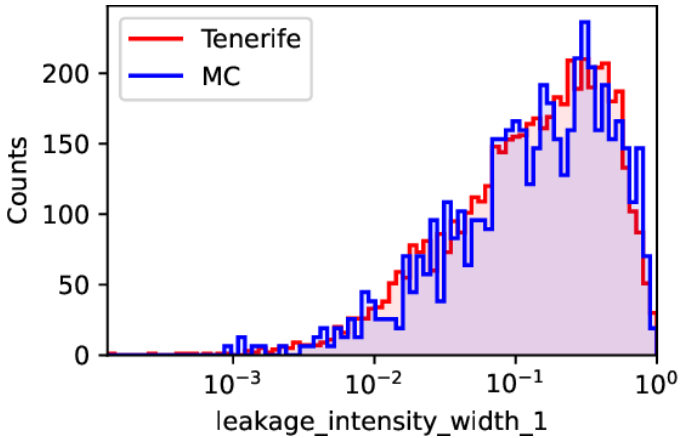


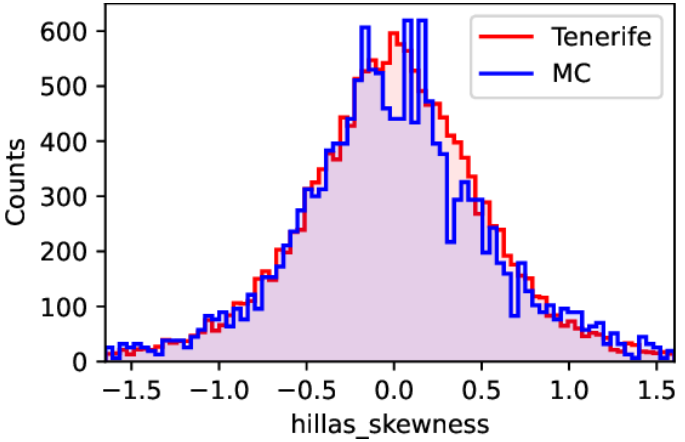


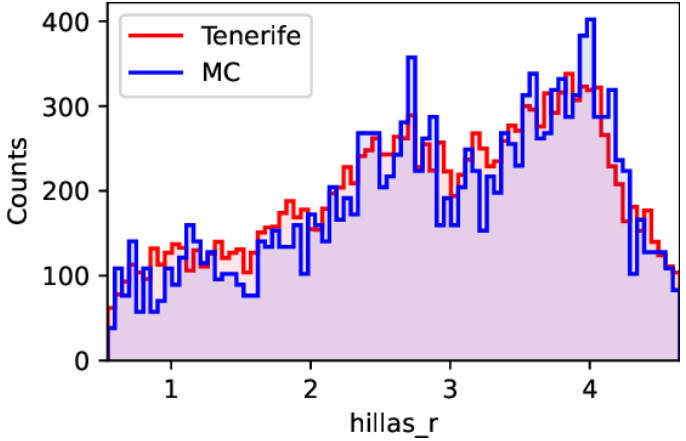


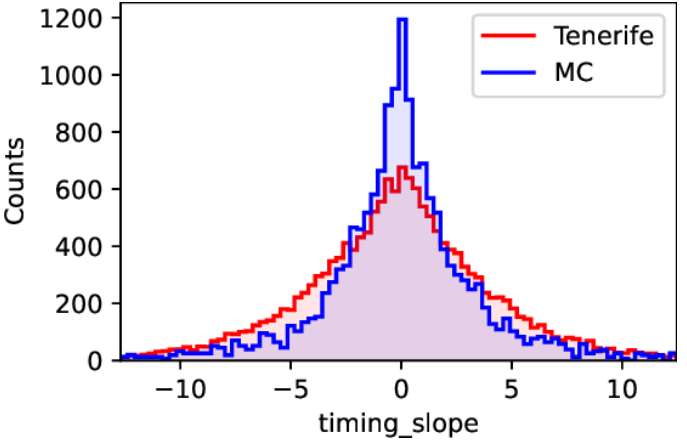


Here’s how the Hillas parameters compare



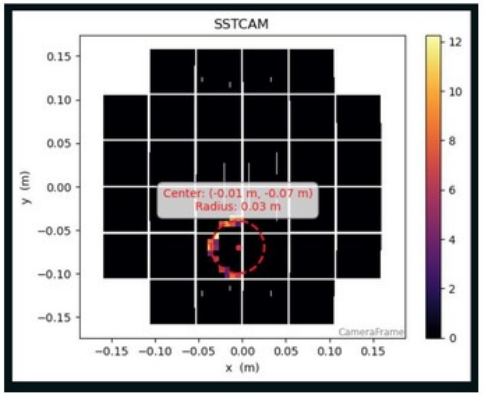






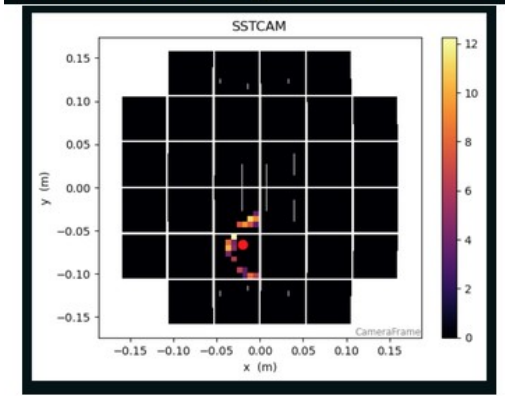
Muon identification and rate comparisons were made

From: Michail Kravchenko @ DESY



Check distance from
Circle's geometric center

```
if (6 < len(trig_sp) < 20)  and (not is_pixels_at_center_c):
```



Check distance from
Pixels' center of mass

Full camera simulation

ring_completeness = 0.7

```
Detected muons : 1861
Injected Muon Rate (R_inj) : 1.7373 Hz
Detected Muon Rate (R_det) : 0.6219 Hz
```

Centroid depletion check (0.4 * max_d)

```
Detected muons : 1861
Injected Muon Rate (R_inj) : 1.7373 Hz
Detected Muon Rate (R_det) : 0.6312 Hz
```

0.63 Hz × ¼ camera
× 0.7 edge effect = 0.110 Hz

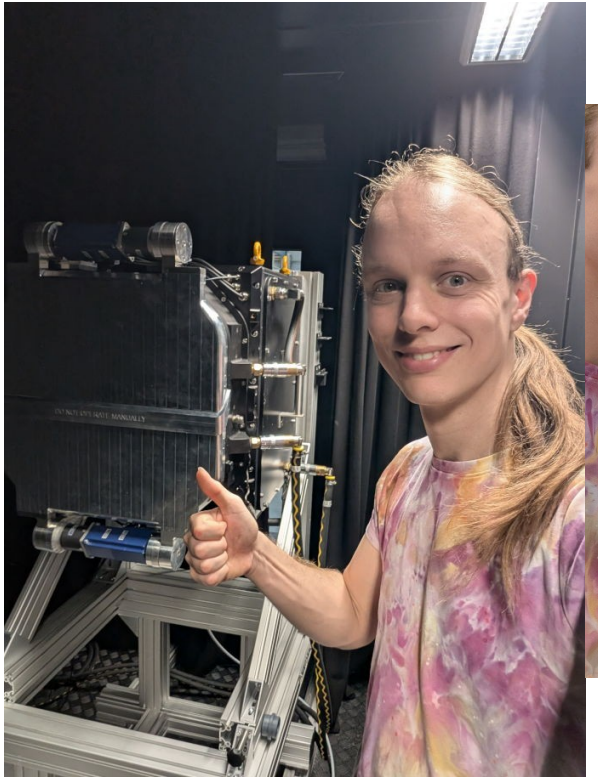
QCAM-i

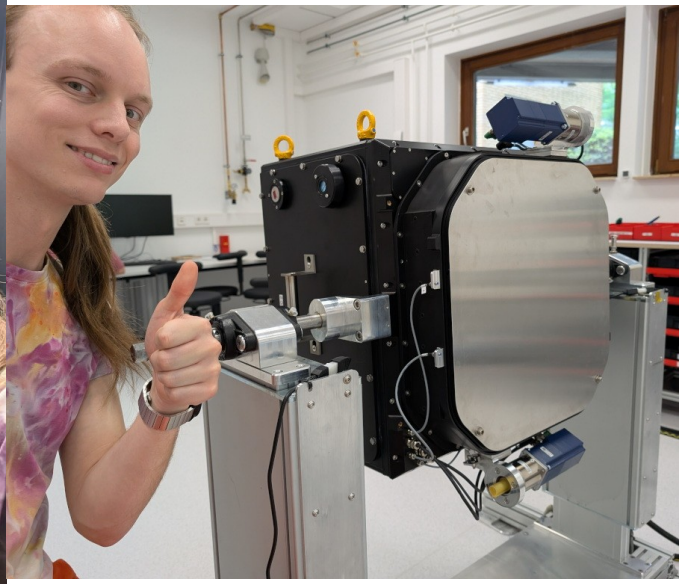
n_showers | 5000

Detected muons : 207
Detected Muon Rate (R det) : 0.0751 Hz

That’s what I’ve been up to

- Preliminary QCAM-i simulations compare well with on-sky data
- We’re currently in the process of verifying simulation parameters
- We’ll make single SST simulations to prepare for ECAM-i in Chile

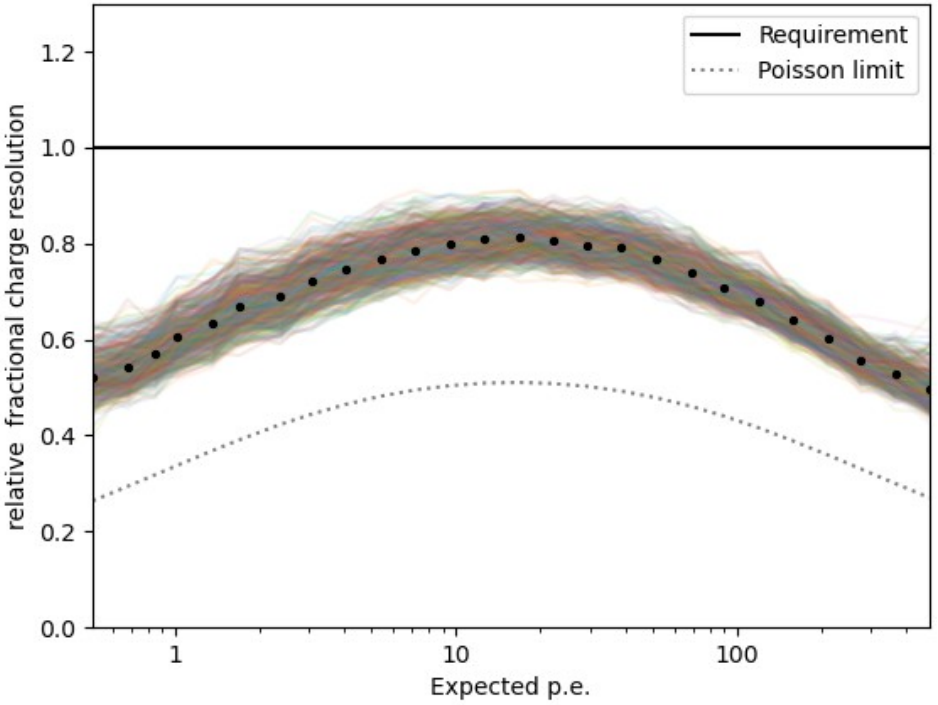




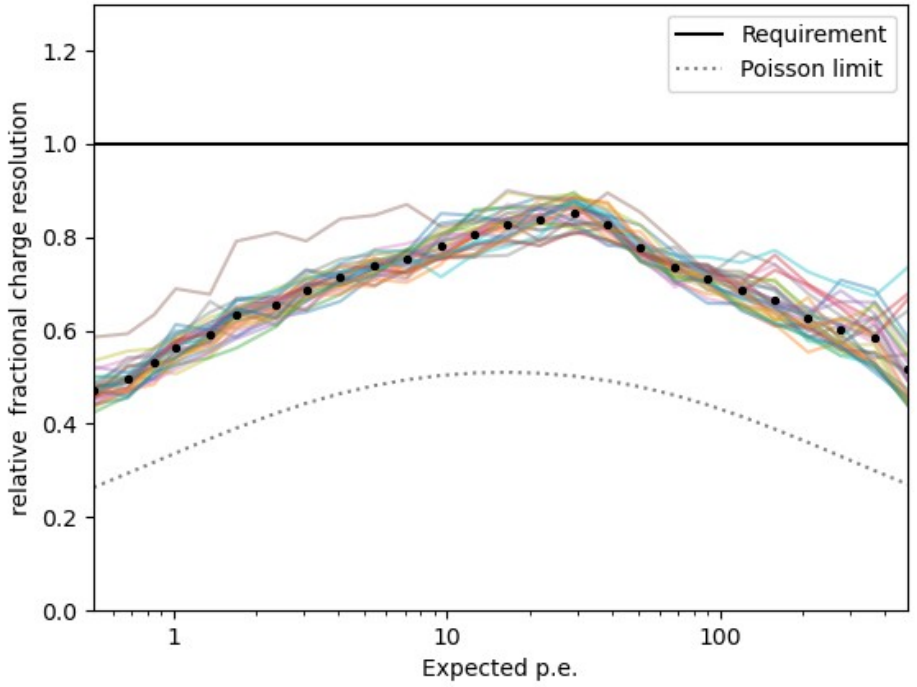
Backup

Charge resolution requirements

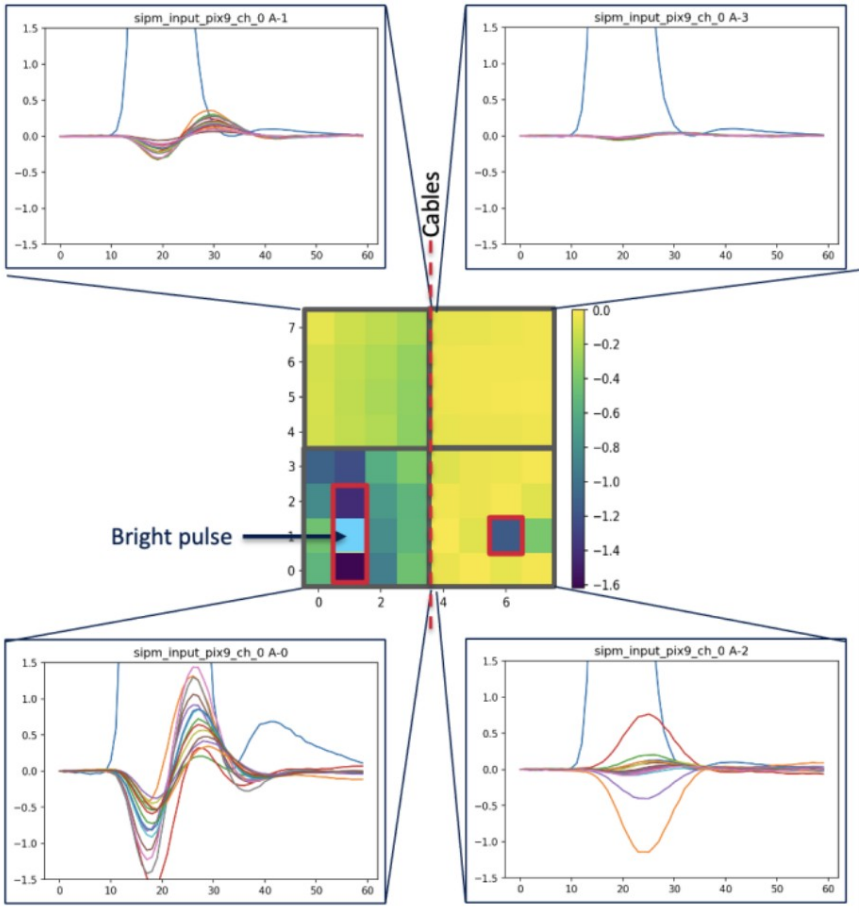
Transfer function derived with all pixels on i.e. with cross-talk



Transfer function derived with 1 pixel on per ASIC, applied to all pixels on



Backup



Extract from
“SST Camera Status Update”
[Richard White]

Backup

```
area = ((cf[0]['max_scatter_range']*u.m)**2 * np.pi).to(u.cm**2)
solid_angle = 2*np.pi*(1-np.cos(cf[0]['max_viewcone_radius']*u.deg)) * u.sr

index=-2.7
N=(9.6e-9 / (u.GeV * u.cm**2 * u.s * u.sr))

integral_pt1 = (1*u.TeV).to(u.GeV) * N * (1 / (index + 1))
integral_pt2 = ((e_max/u.TeV)**(index + 1) - (e_min/u.TeV)**(index + 1))
integral = integral_pt1 * integral_pt2

particle_rate = integral*area*solid_angle

mc_time = n_showers / particle_rate
ten_time = 557.1491014957428 * u.s
mc_scale = (ten_time/mc_time).value
```