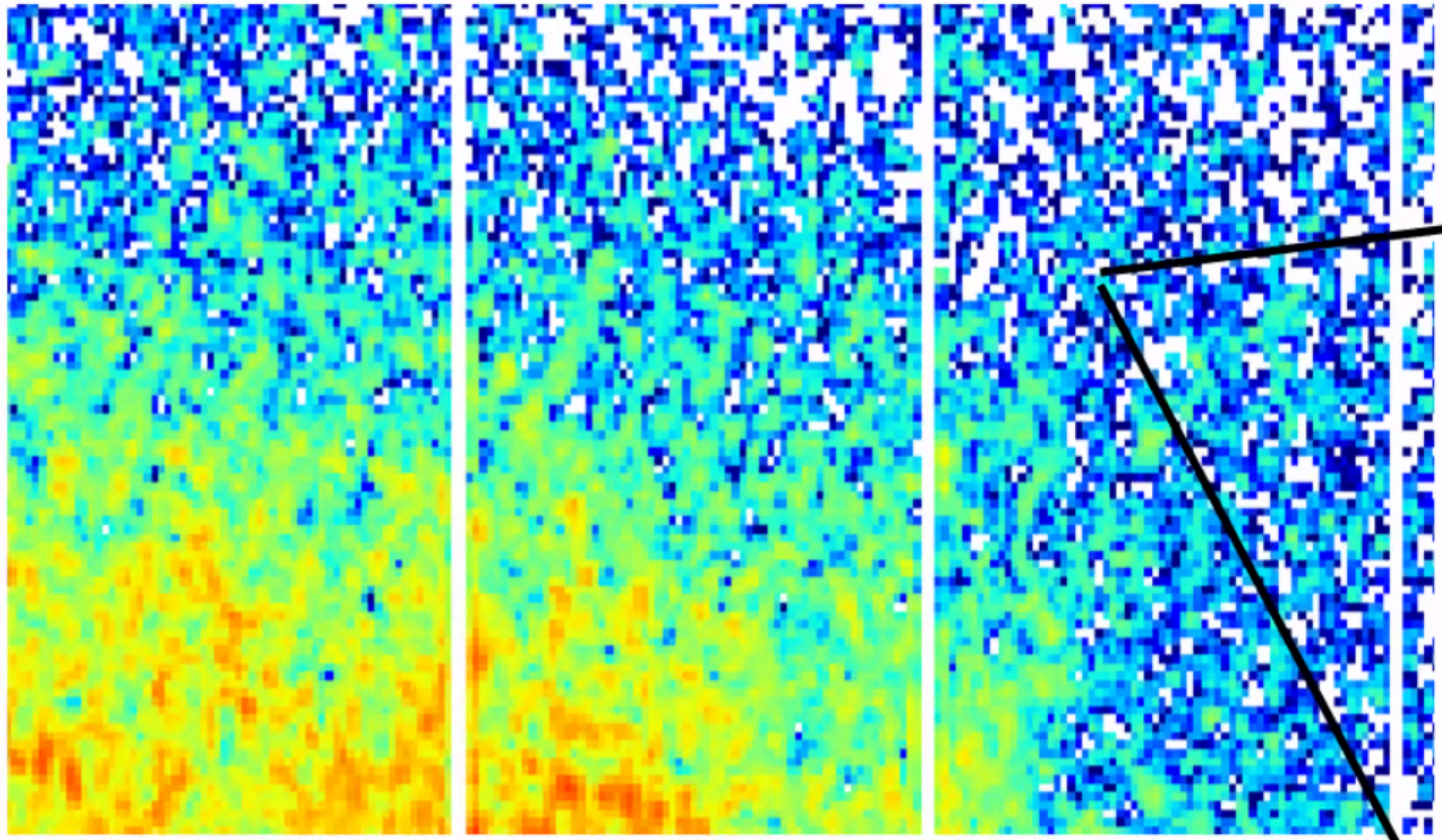
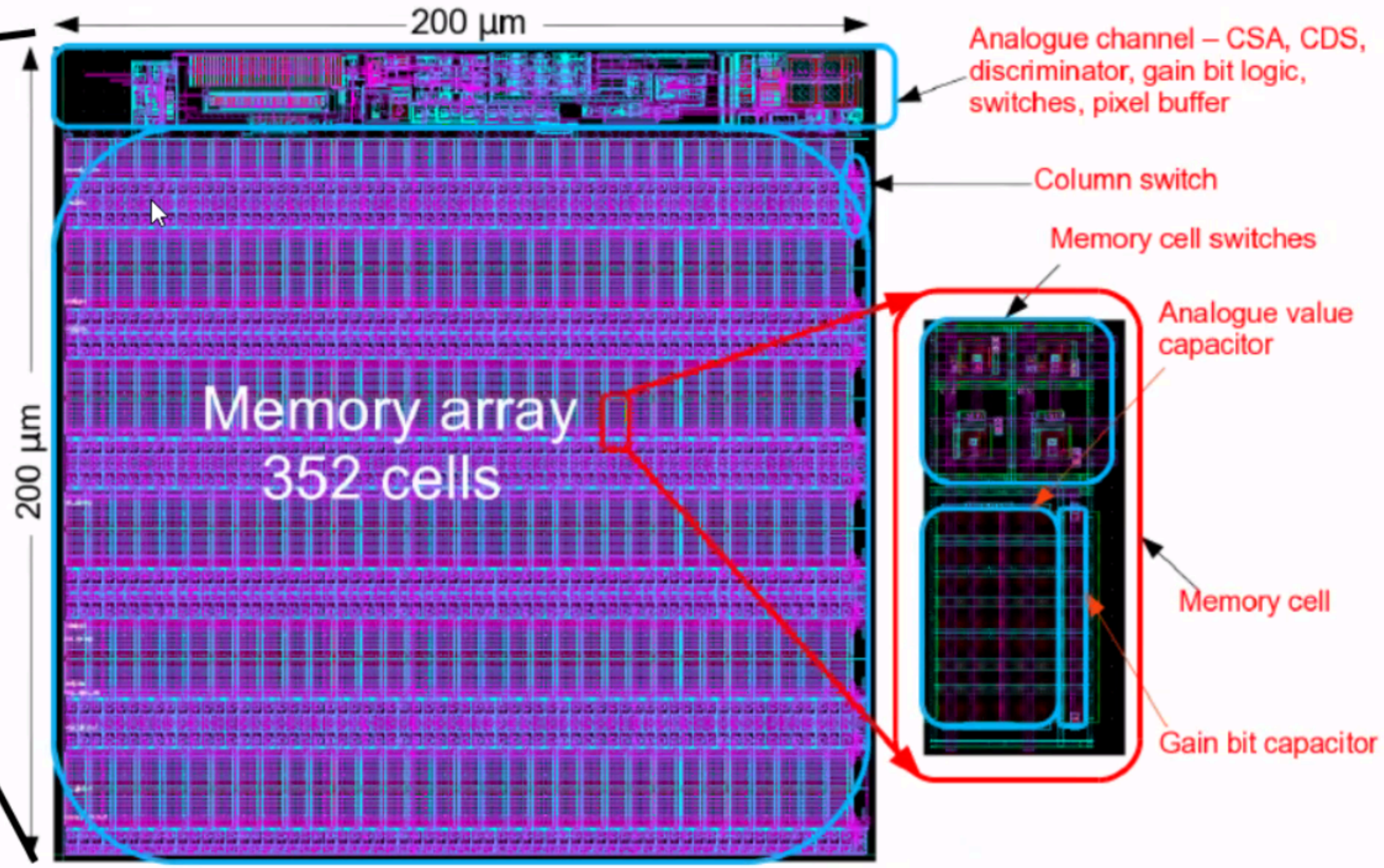
In pixel storage

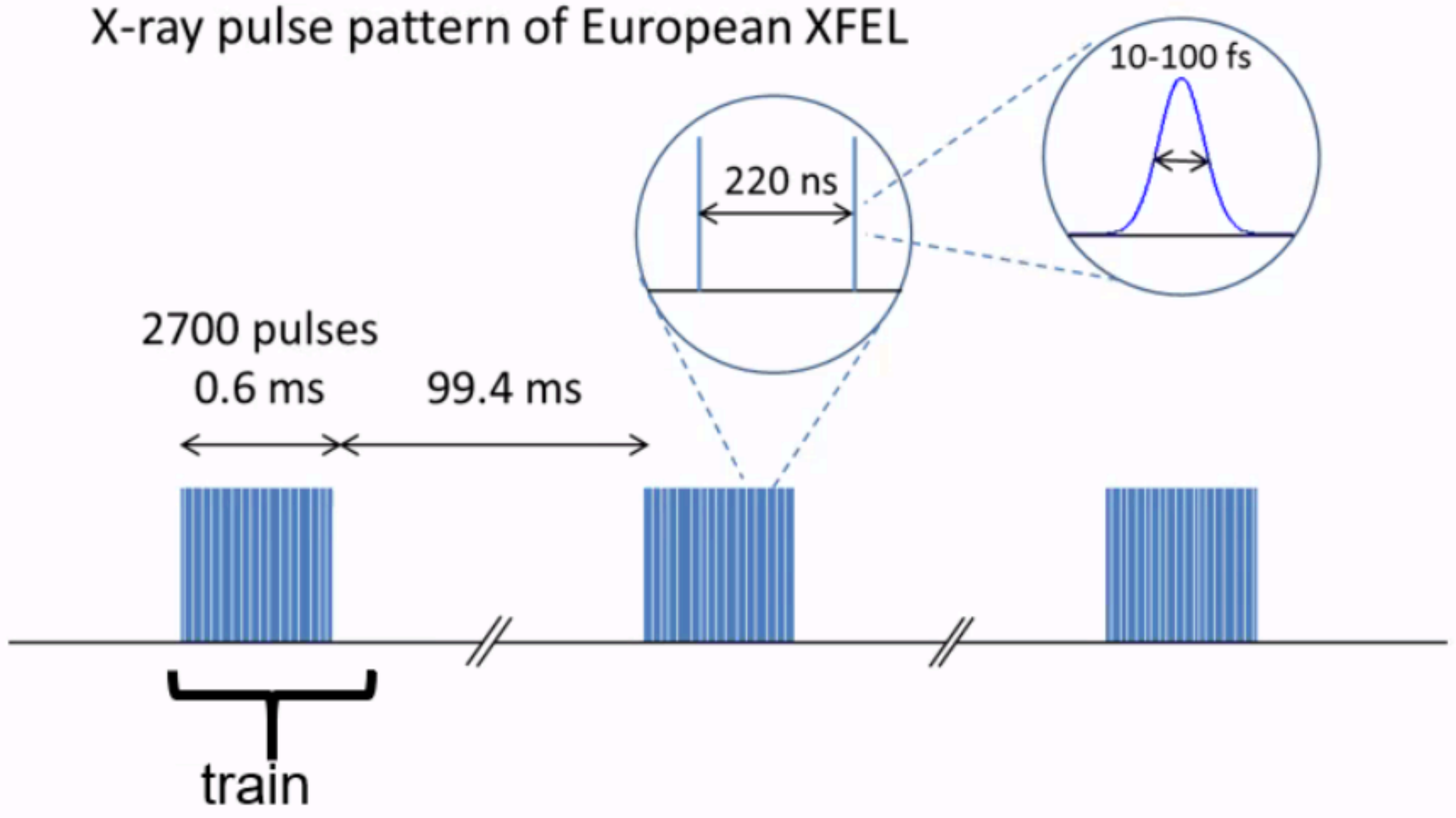
- Adaptive Gain Integrating Pixel Detector (AGIPD)
- 352 pulses can be stored for each train (4.5 MHz) and read-out between trains (10 Hz)

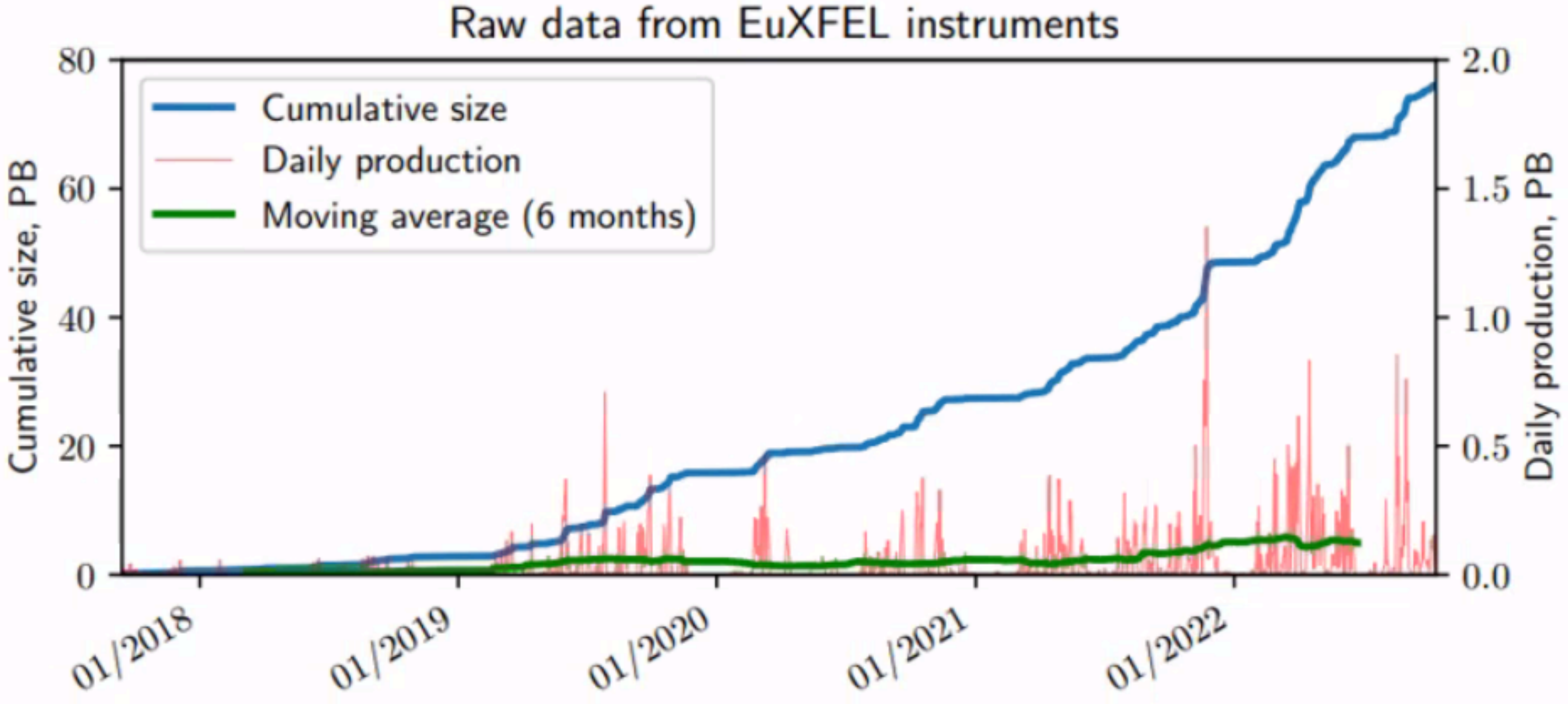




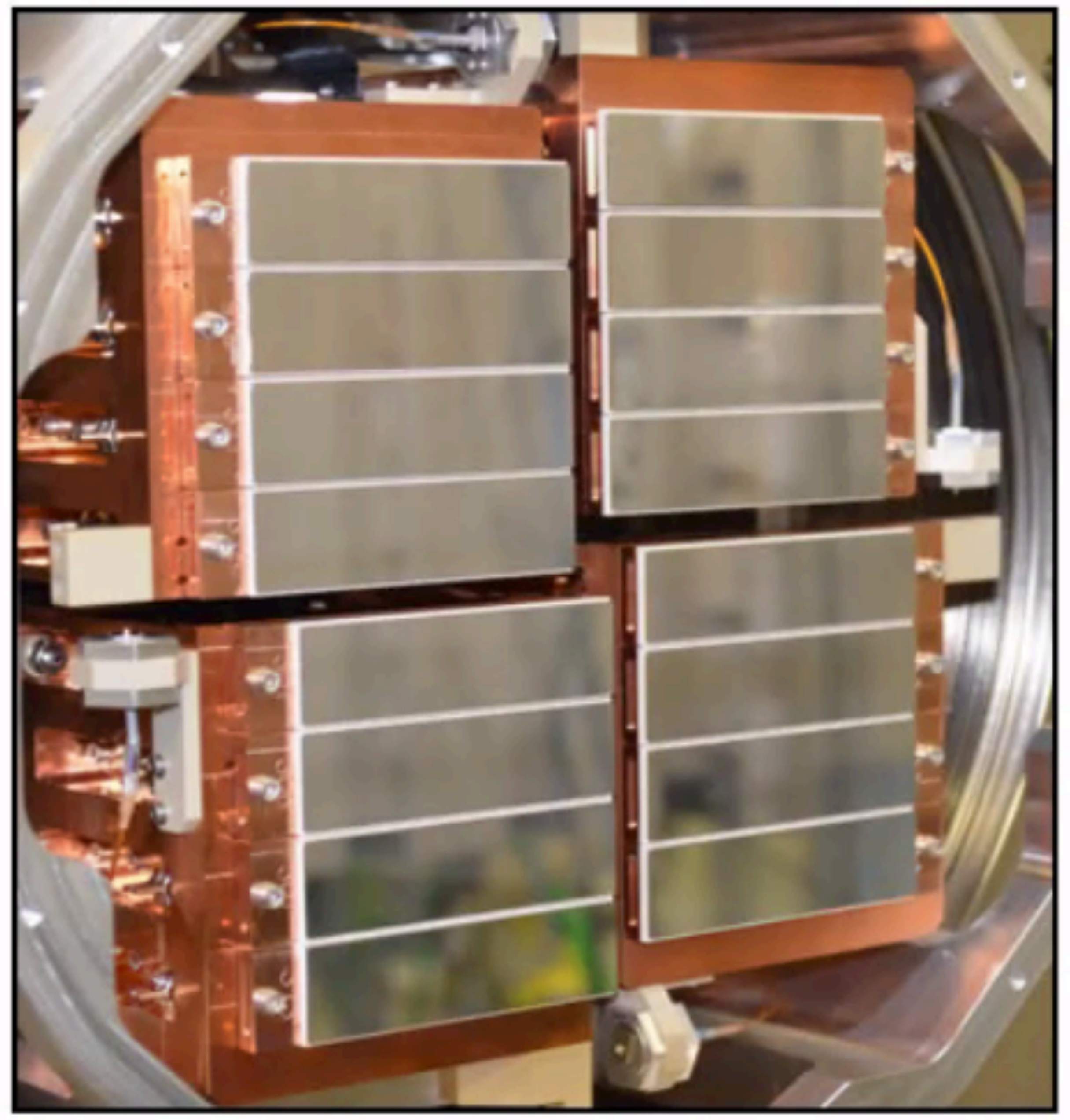
Allahgholi et al., J. Instr. (2015)

European XFEL





The best(?) week so far: 7 PB in 7 days.

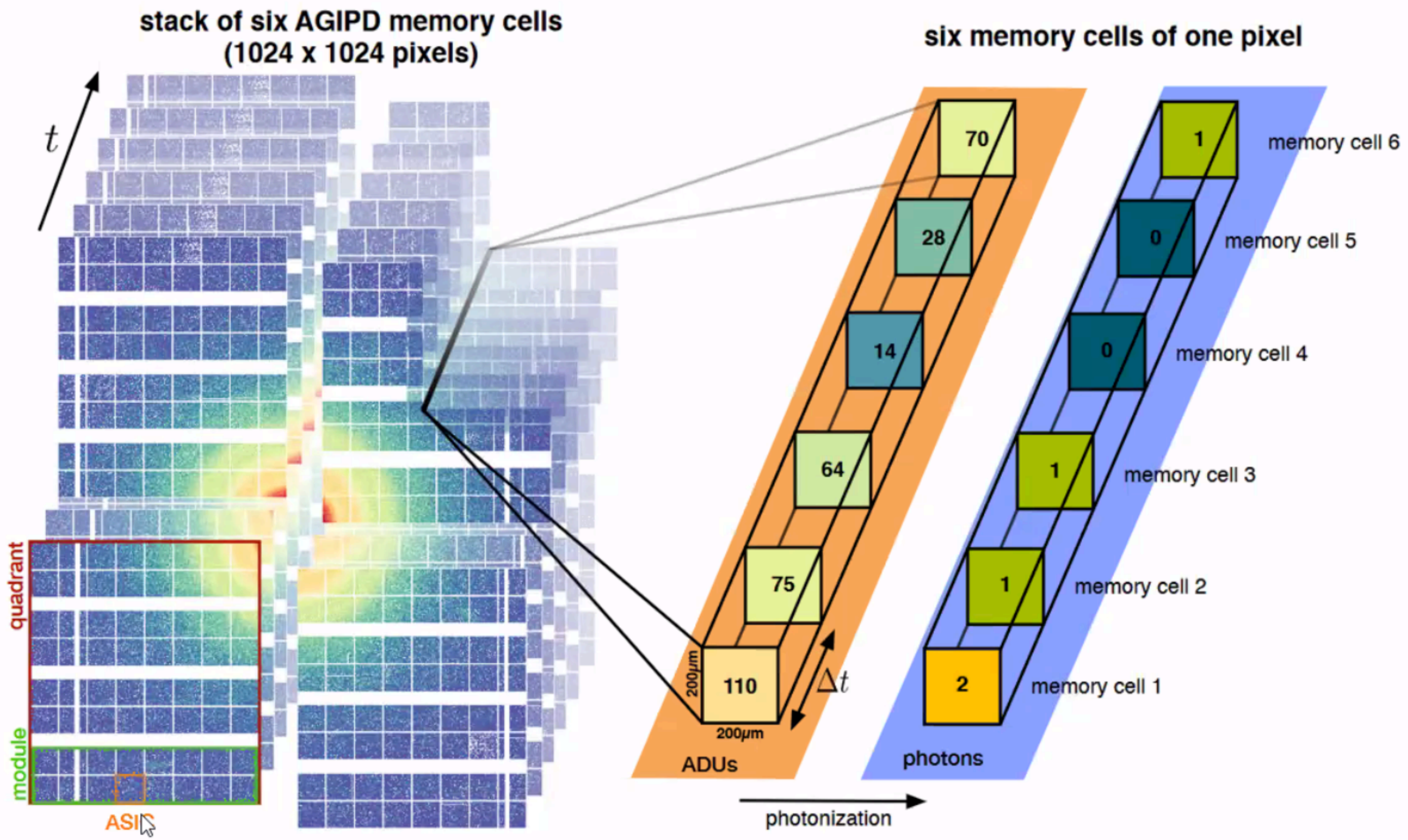


AGIPD 1M detector @ MID

- 1M Pixel
- pixel size 200x200um^2
- Capable of 4.5MHz
- 352 storage cells (352 images/train)
- Single photon sensitive
- Up to 10^4 photons/pixel with 3 gain stages

Detector	Sampling	Bandwidth
LPD 1M	5,120 Hz	86 Gbit/s
AGIPD 1M	3,520 Hz	118 Gbit/s
DSSC 1M	8,000 Hz	134 Gbit/s
ADQ412	$1.2 \cdot 10^6$ Hz	3 Gbit/s

- 4 quadrants
- 4 modules
- 2 x 8 ASICs
- 64 * 64 pixel
- 11*32 storage cells
  - *2*uint16*10 Hz
  - = 15 GB/s



Dallari et al. *Applied Sciences* 11.17 (2021): 8037

Out[4]: xarray.DataArray 'concatenate-8fa8b661b9b394489f8b1d0b88bad149'
(**trainId**: 1092, **pulseId**: 350, **module**: 16, dim_0: 512, dim_1: 128)

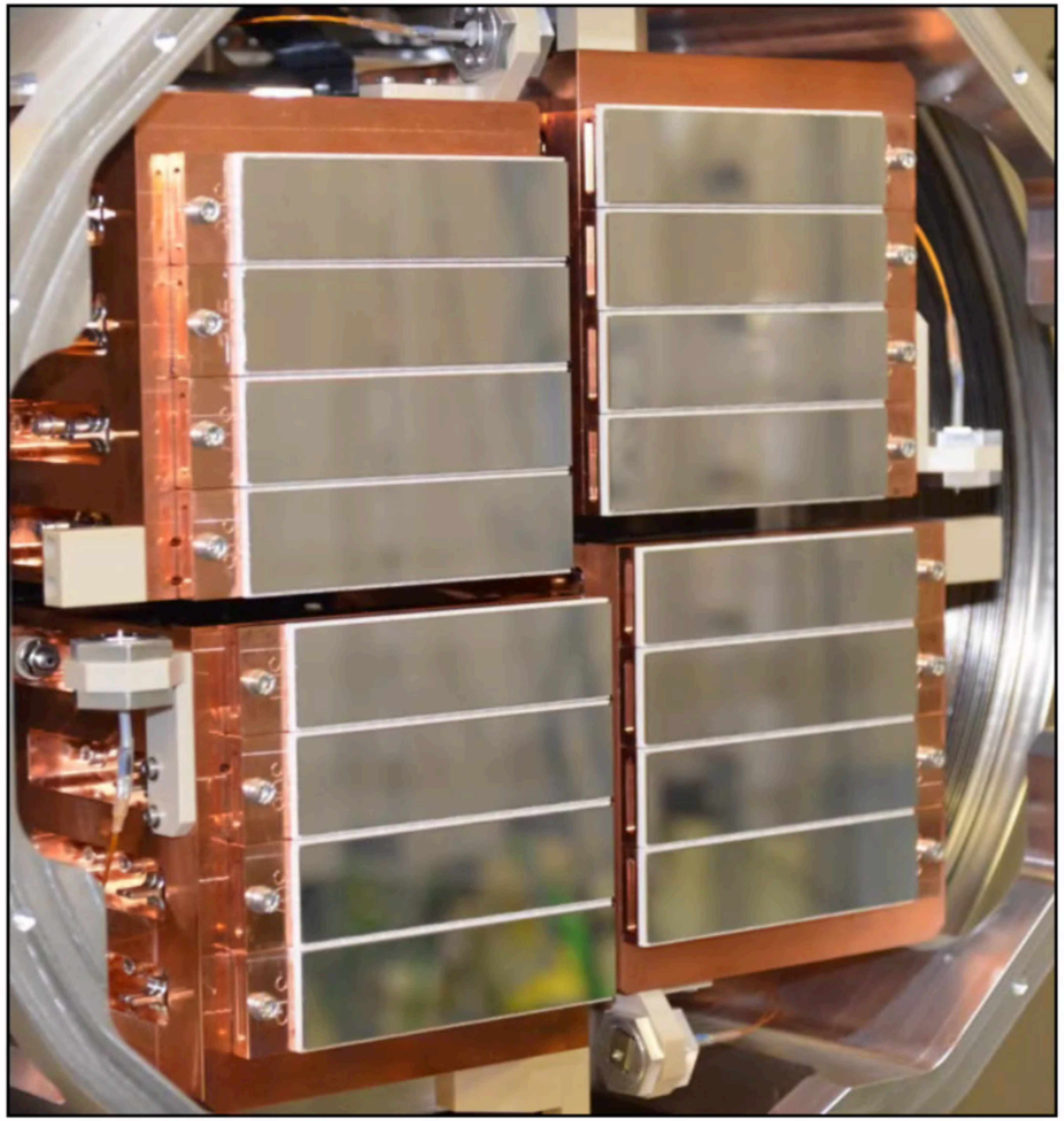
	Array	Chunk
Bytes	1.60 TB	1.74 GB
Shape	(1092, 350, 16, 512, 128)	(19, 350, 1, 512, 128)
Count	118406 Tasks	3584 Chunks
Type	float32	numpy.ndarray

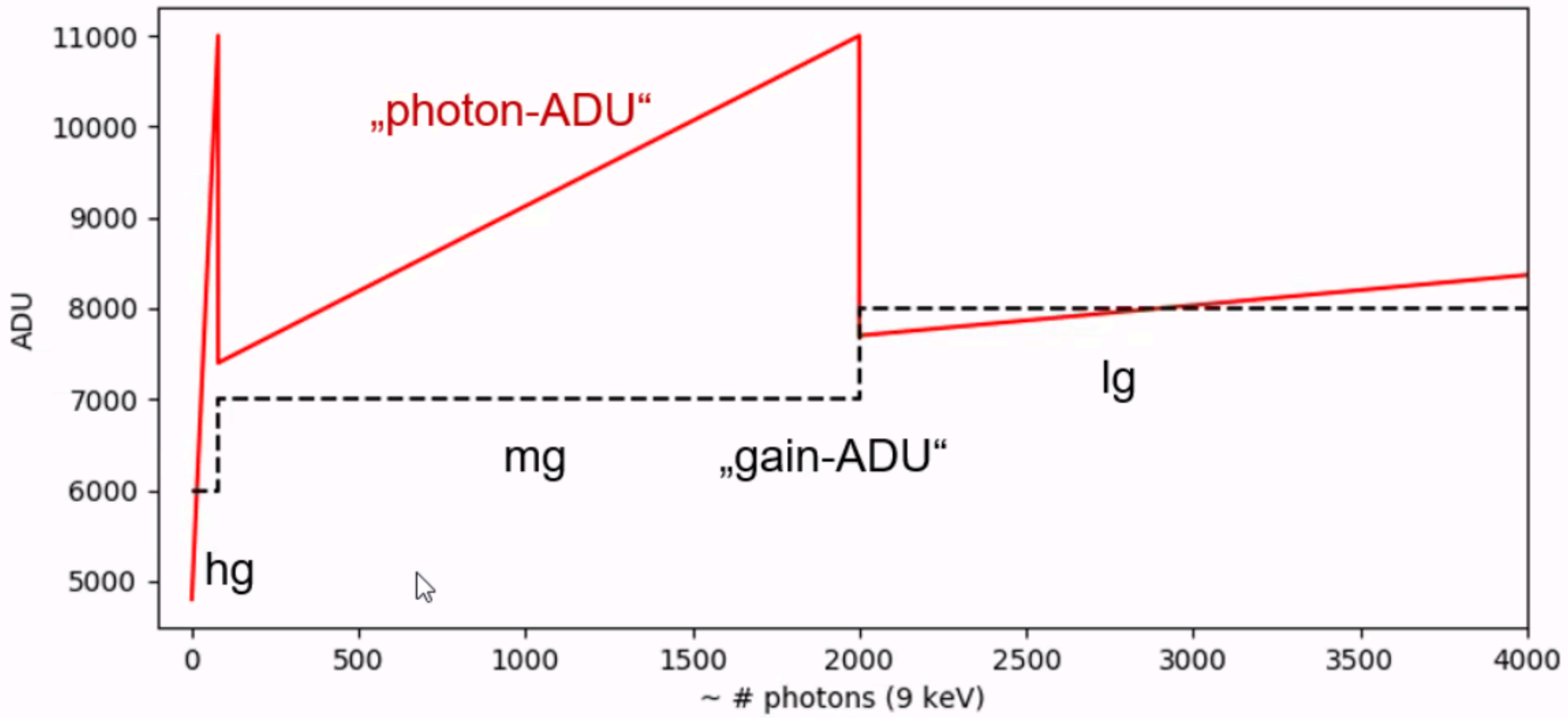
▾ Coordinates:

module	(module)	int64	0 1 2 3 4 5 6 ... 10 11 12 13 14 15
trainId	(trainId)	uint64	1474506456 ... 1474507596
pulseId	(pulseId)	uint64	2 4 6 8 10 ... 692 694 696 698 700

▸ Attributes: (0)

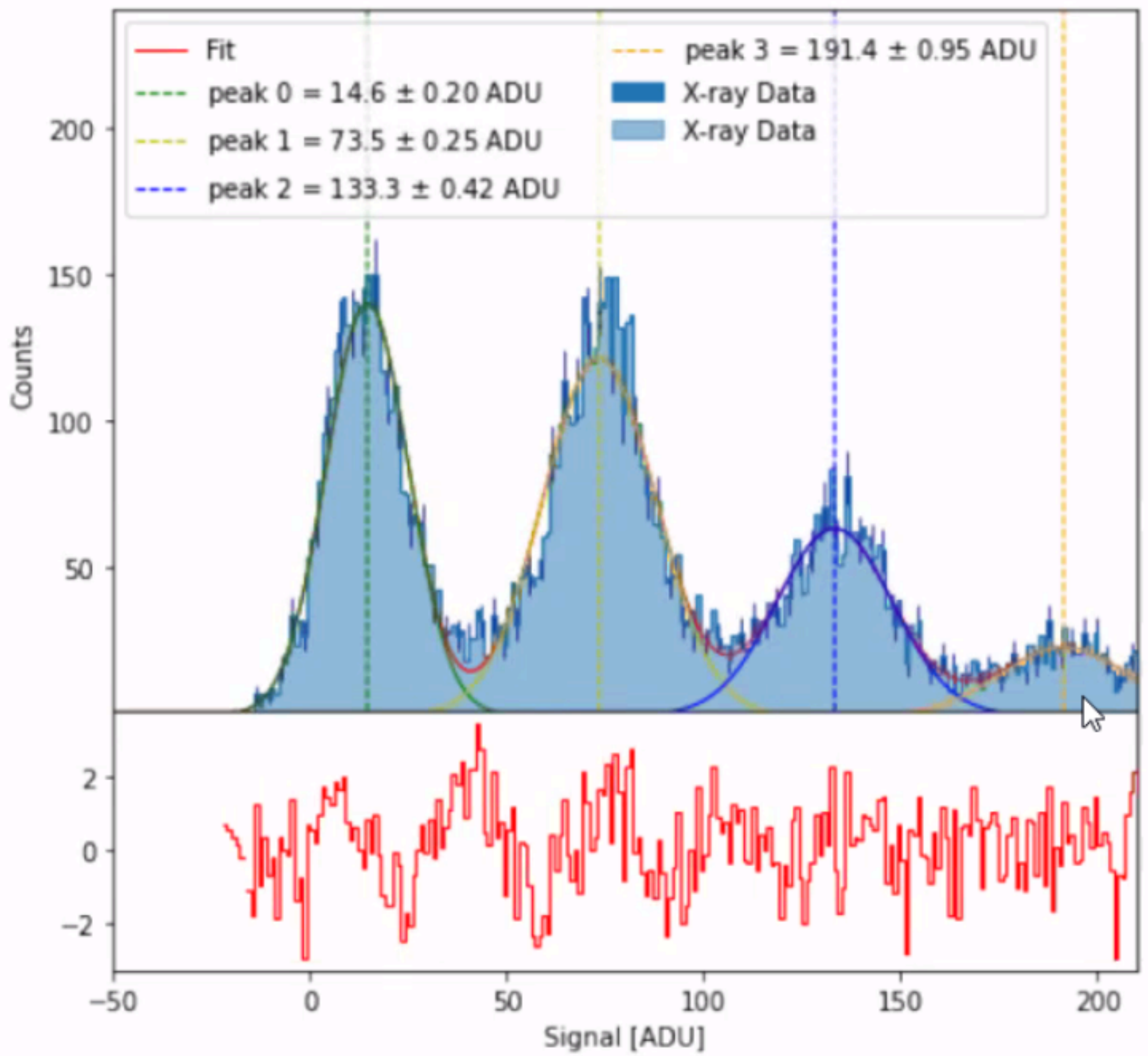
Adaptive Gain Integration Pixel Detector AGIPD



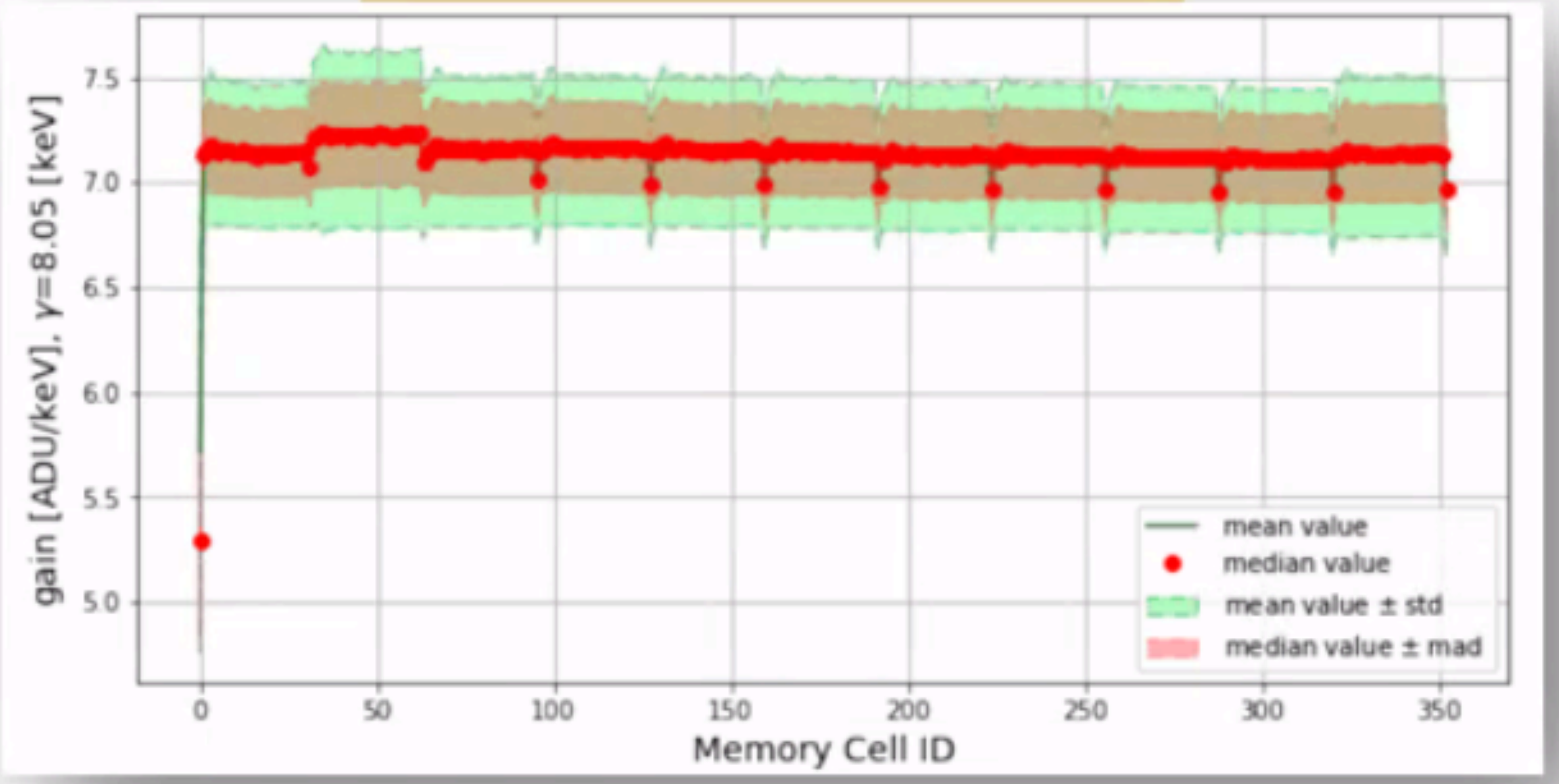


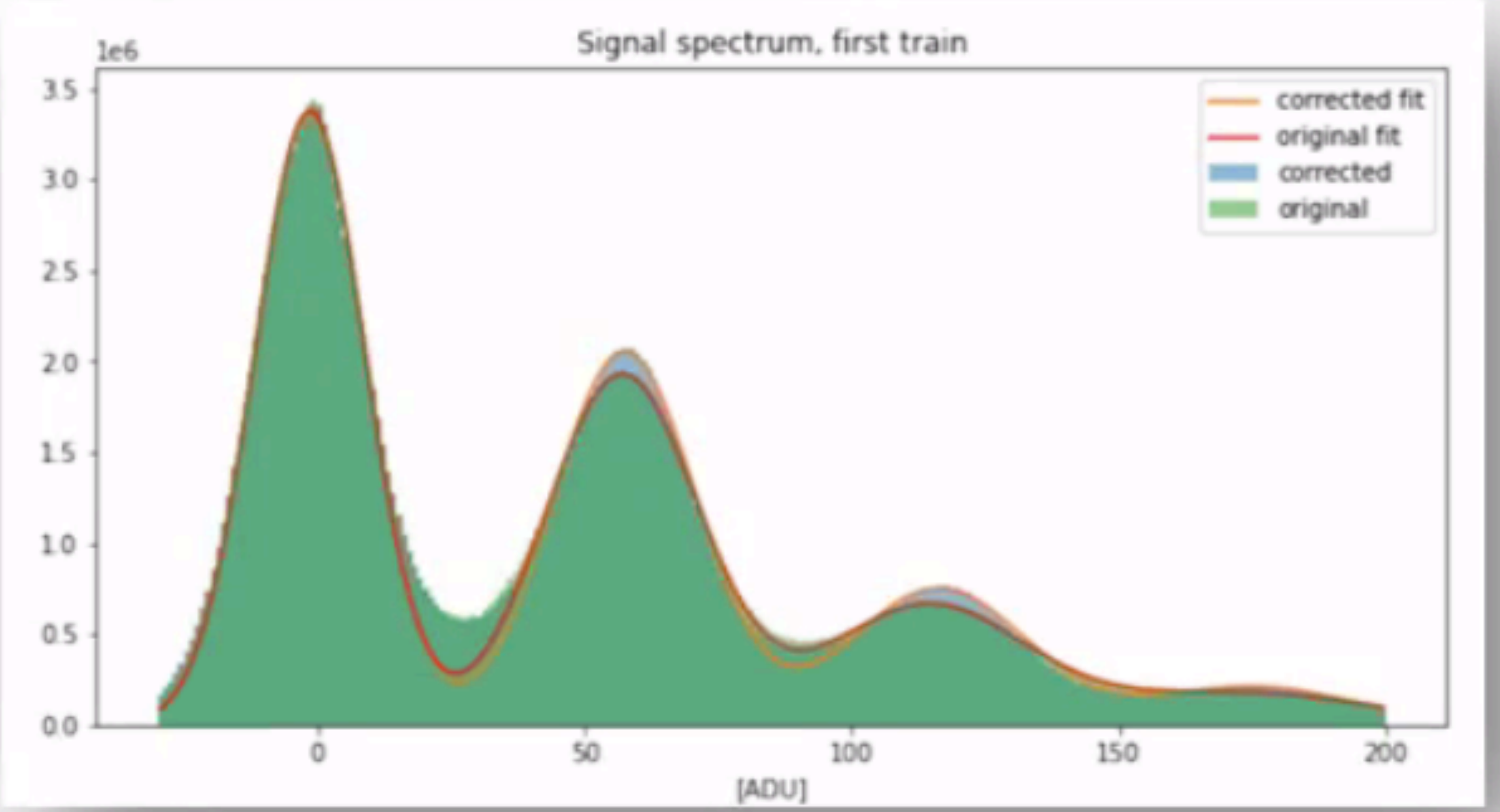
- Raw data:
  - 2 unit16 values per storage cell per pixel
    - ▶ Gain stage information, gainbit
- Proc data:
  - 1 float32 value proportional to the number of photons (in [keV])
  - (photonizing: uint16 -> number of photons)

Linearity: flatfields



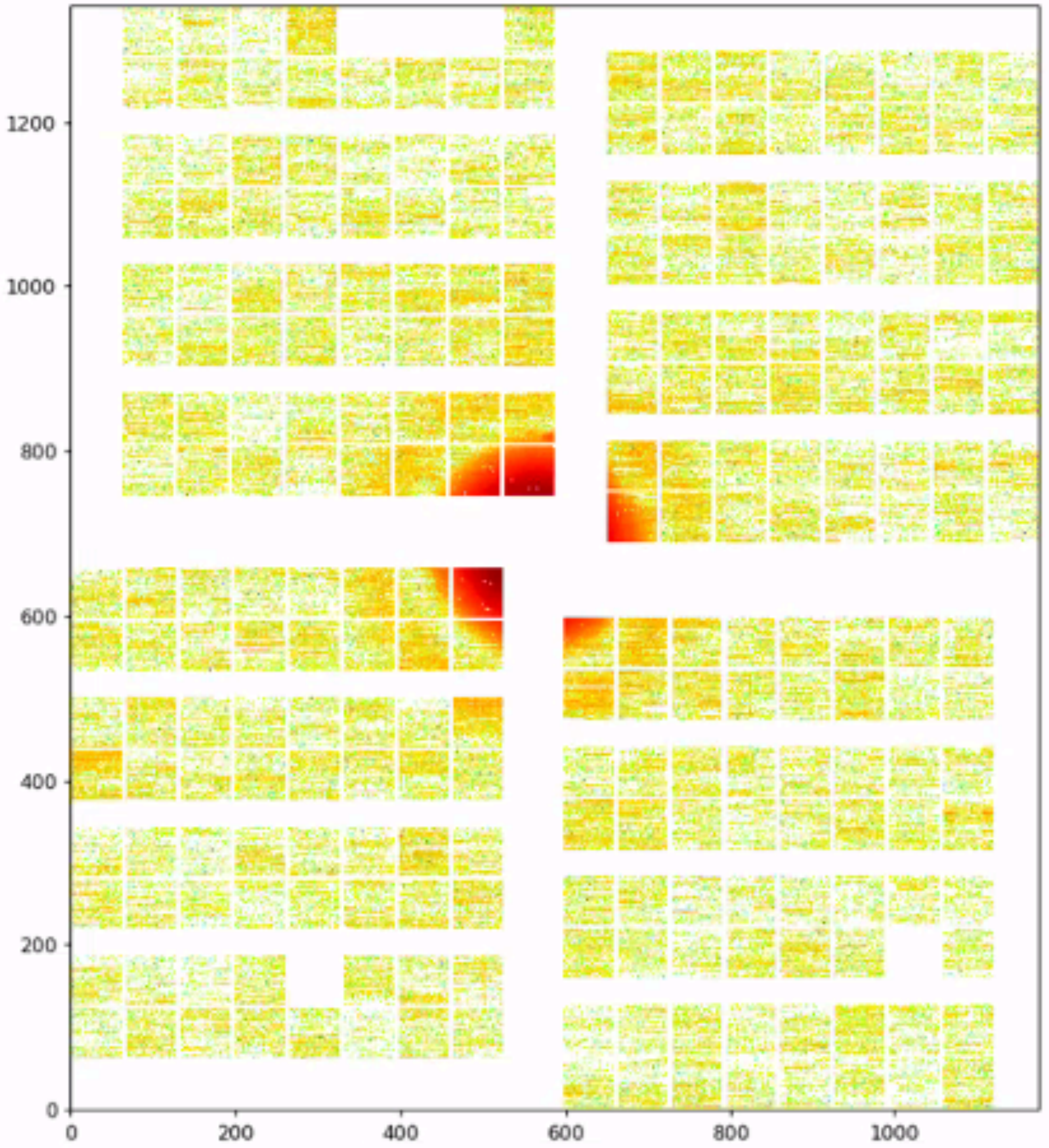
Gain values vs. mem.cell

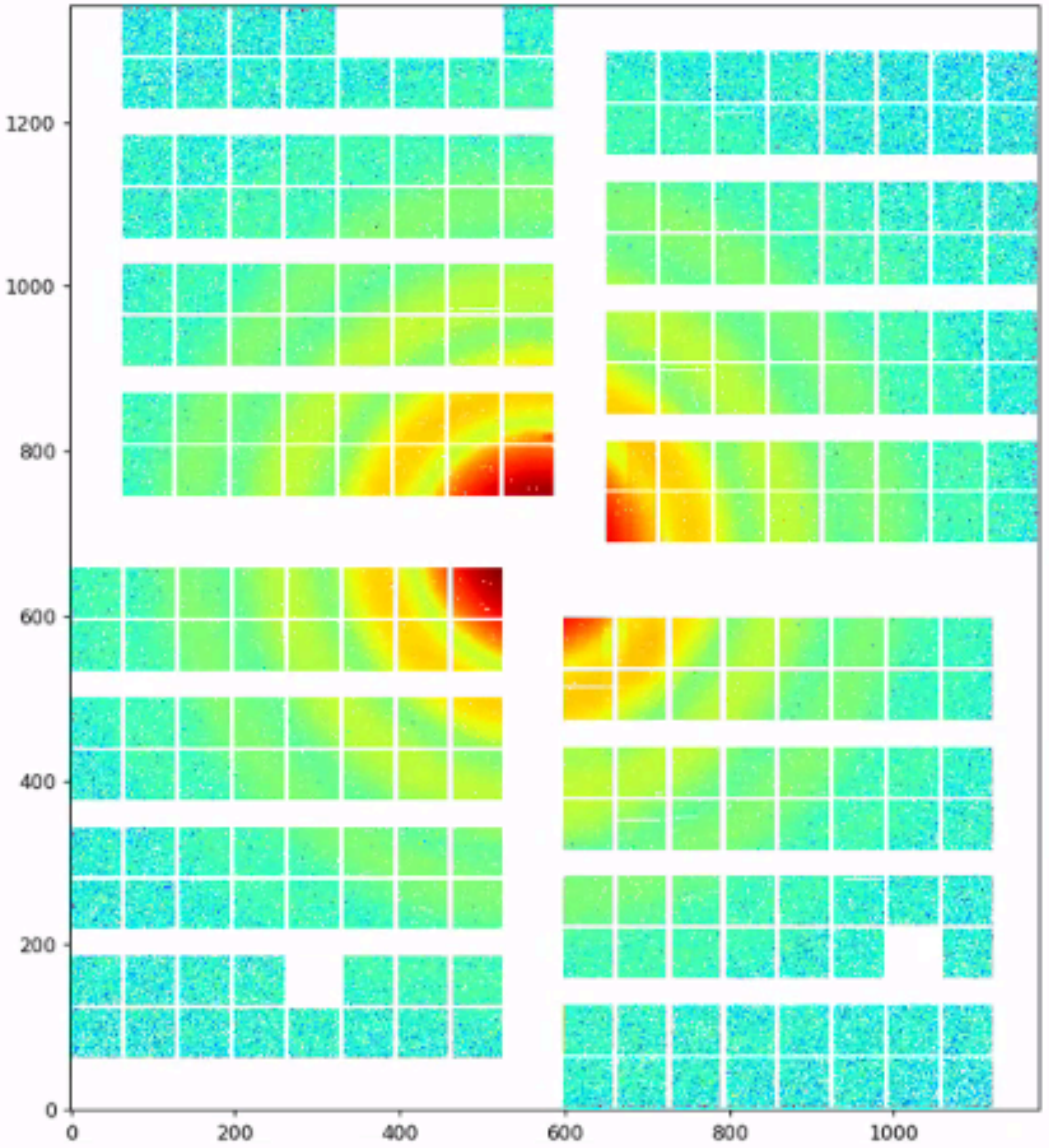




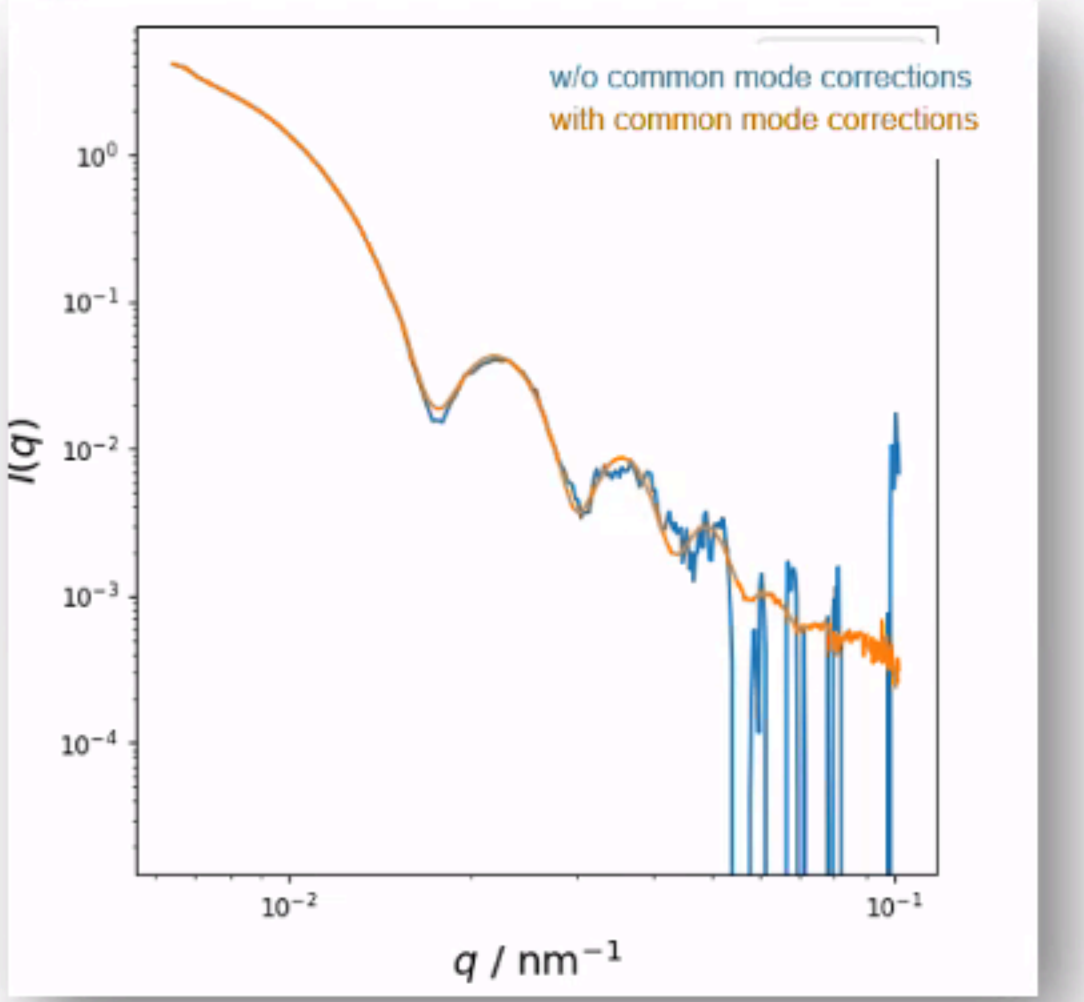
- Determination of linearity factor
  - Sparse Cu fluorescence data
- Development and validation automated histogram calculation and fitting
  - per storage cell
  - x 352 million
- Repeat per detector scenario
  - rep. rate, exp. time, gain mode
  - deposited into calibration constant database

Offset / pedestal: Common mode

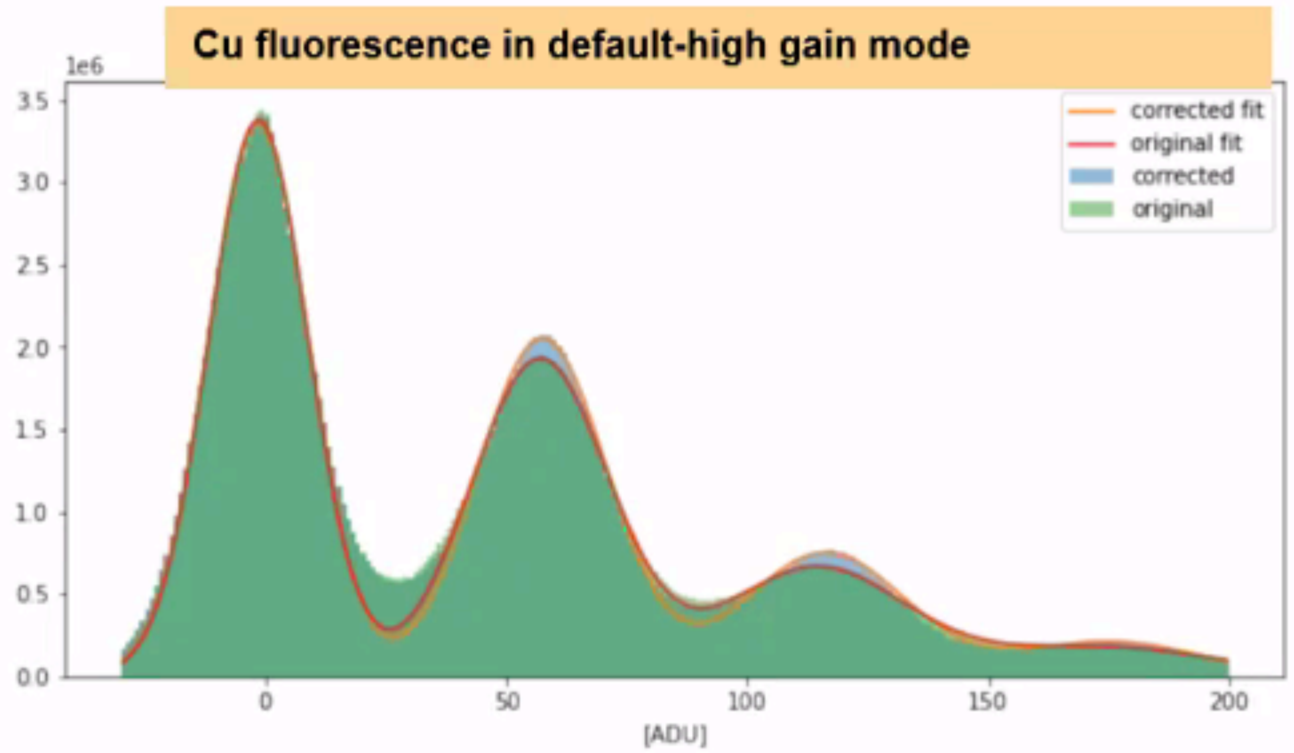


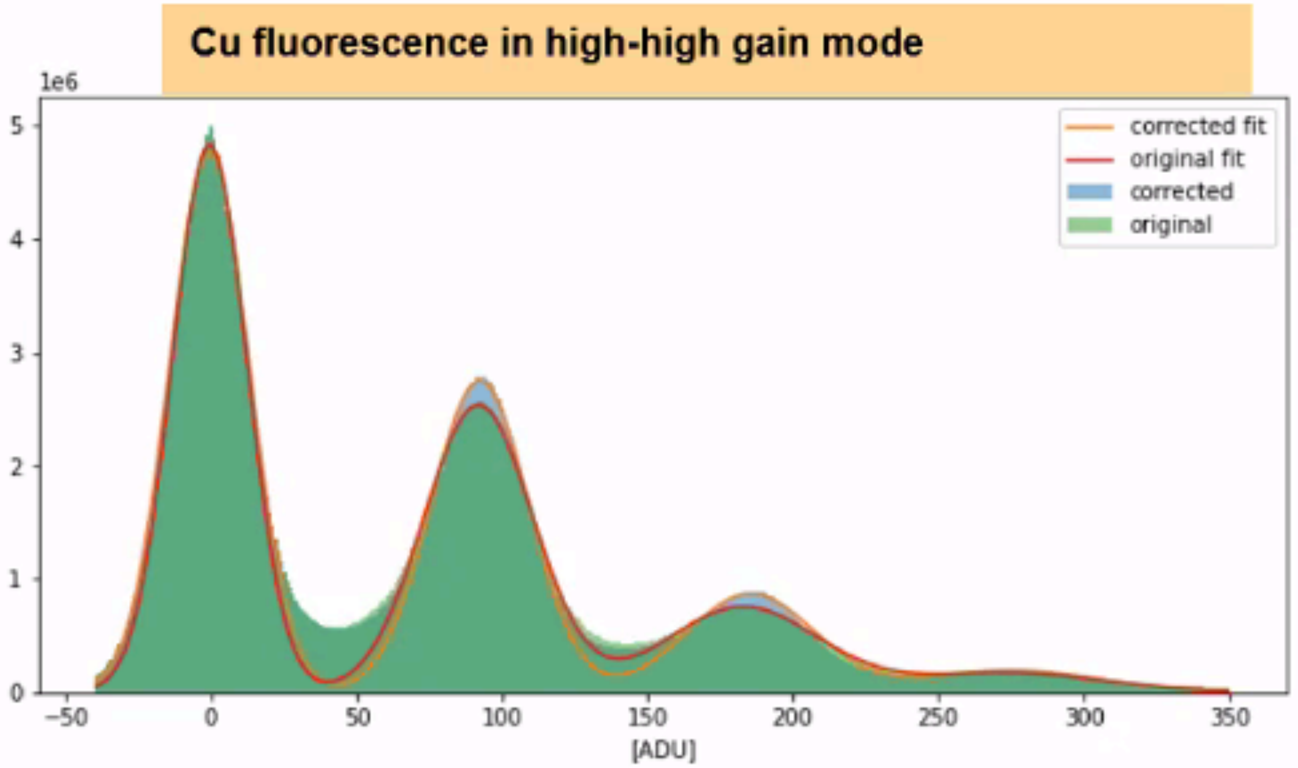


- Pedestals can fluctuate, drift and jump on various timescales
  - Single shot <-> hours
  - Deviations from “dark constant”
    - ▶ Non-zero contribution of dark pixels
- AGIPD common mode correction
  - More precise pedestal subtraction
  - Identifies empty pixel within single ASIC and block of 32 storage cells within one pixel

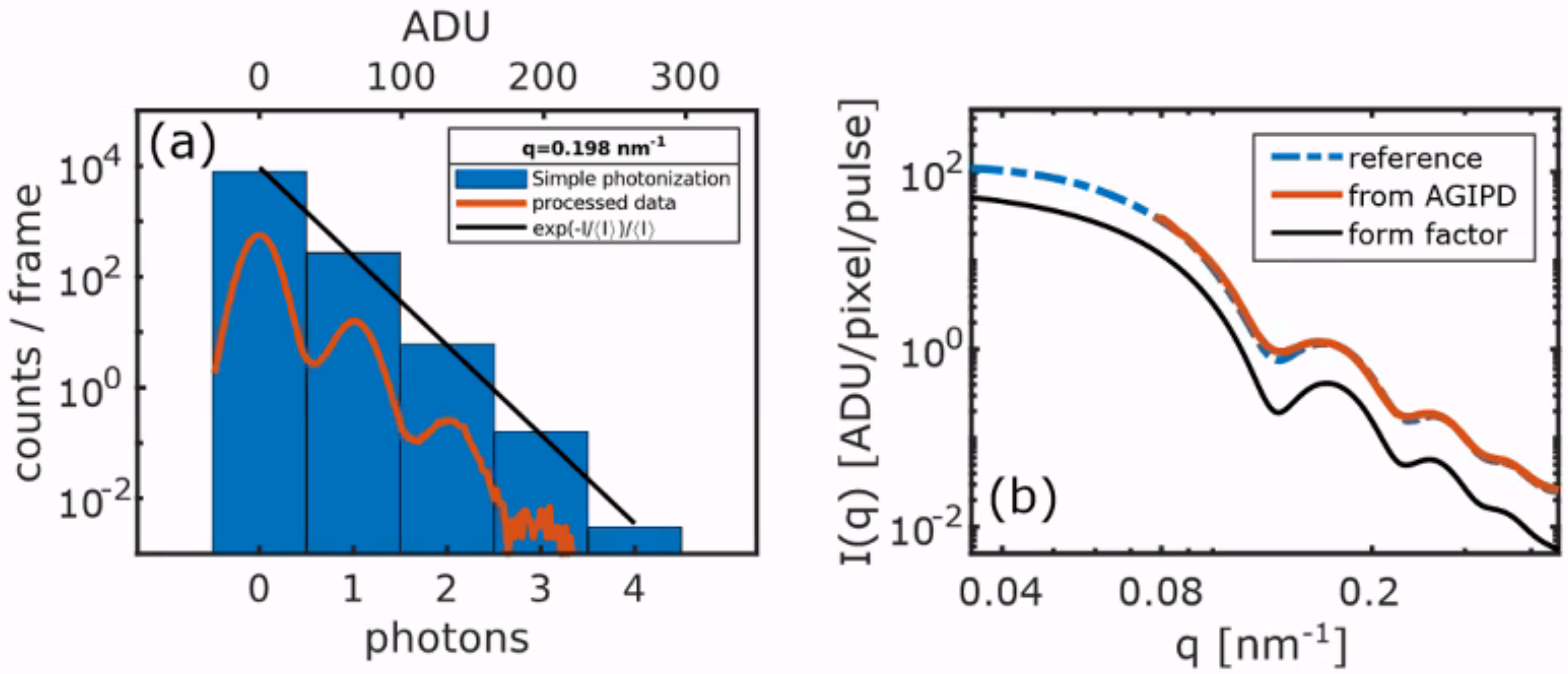


Towards better single photon sensitivity





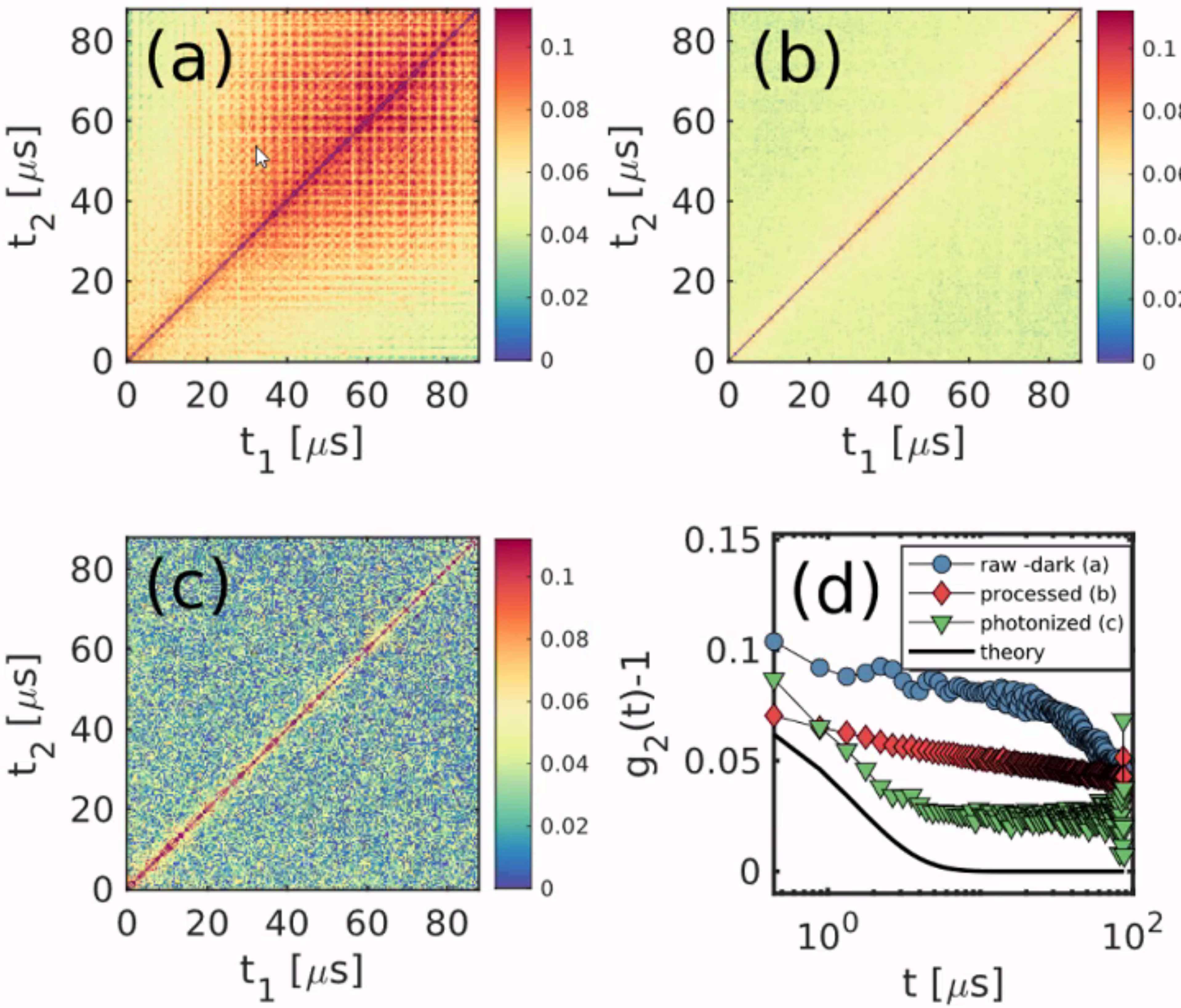
- Adapting detector operation to scenarios to different use cases
- Converting calibrated detector output [keV] to number of photons [#]
  - Simple binning works sufficiently well
    - No accounting for charge sharing
    - No dropletizing
  - "Photonizing"



Dallari et al. *Applied Sciences* 11.17 (2021): 8037

Influence of detector corrections on XPCS results

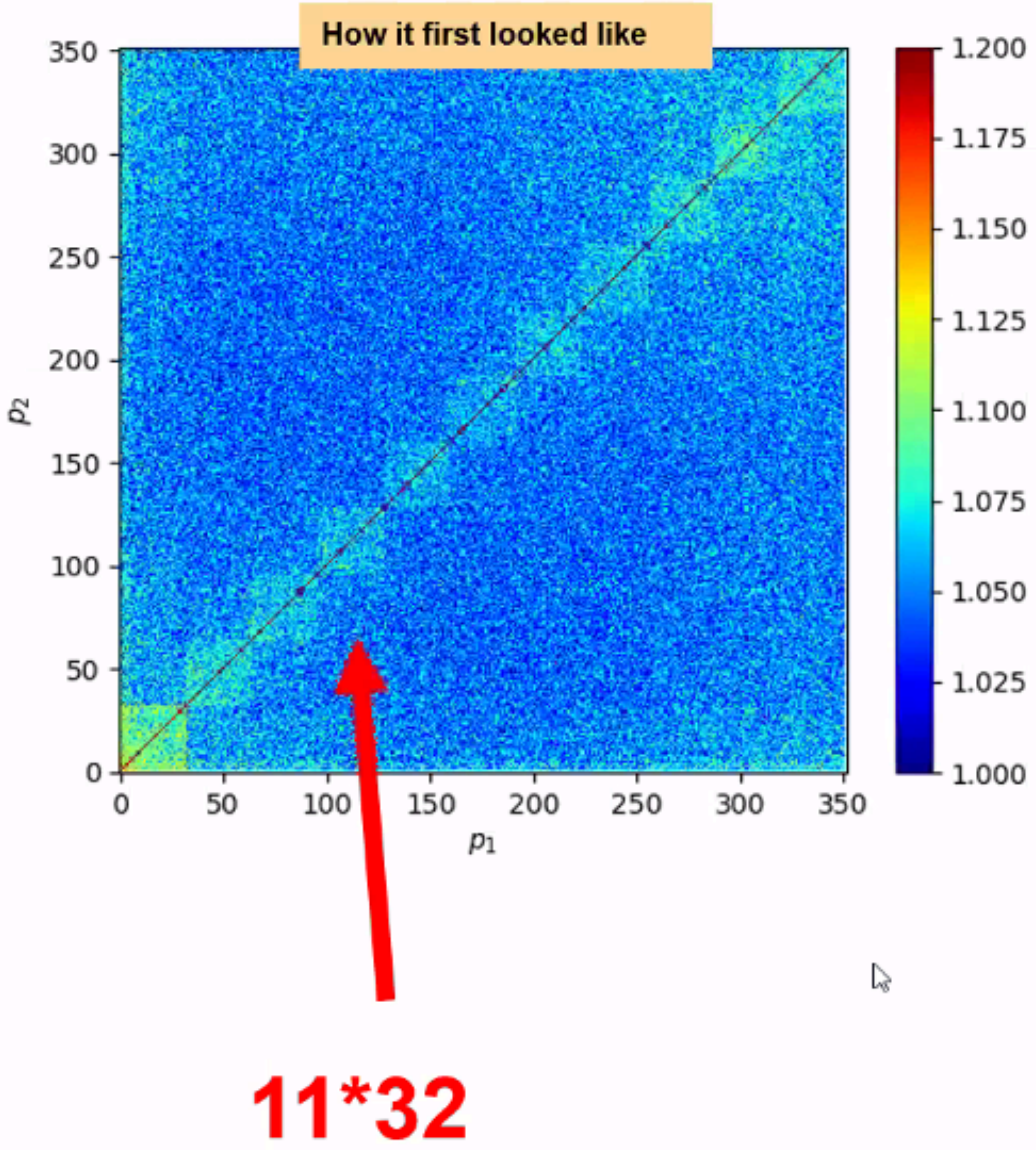
- Averaged two-time correlation functions (TTCFs) after different steps of data corrections
  - (a) dark subtraction only
  - (b) proc data (XFEL pipeline)
    - ▶ Common mode
    - ▶ Flatfield
  - (c) Photonized
- Improved corrections -> improved TTCFs
  - Good qualitative agreement to expected dynamics
- Remaining issues:
  - Baseline
  - Artifacts for low intensities
  - Only correctable in the data analysis, not the detector corrections

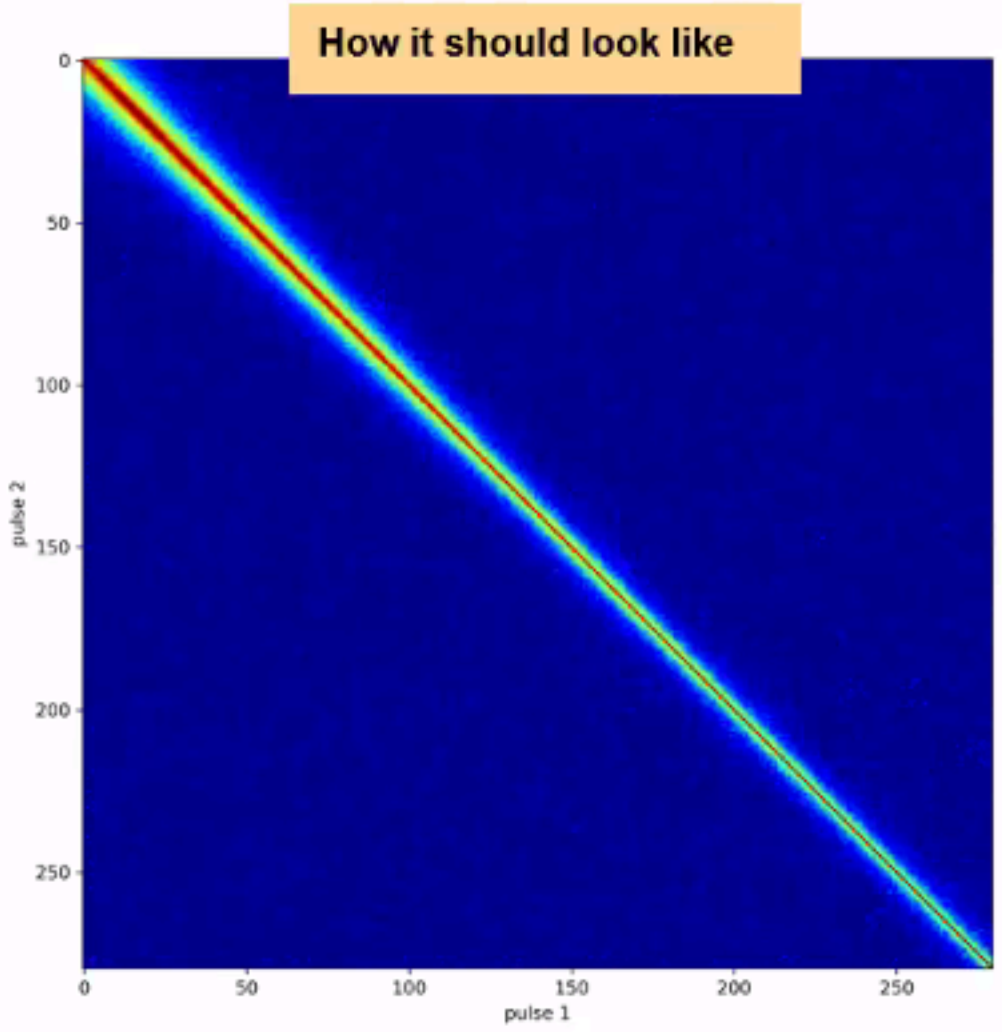


Dallari et al. *Applied Sciences* 11.17 (2021): 8037

squares

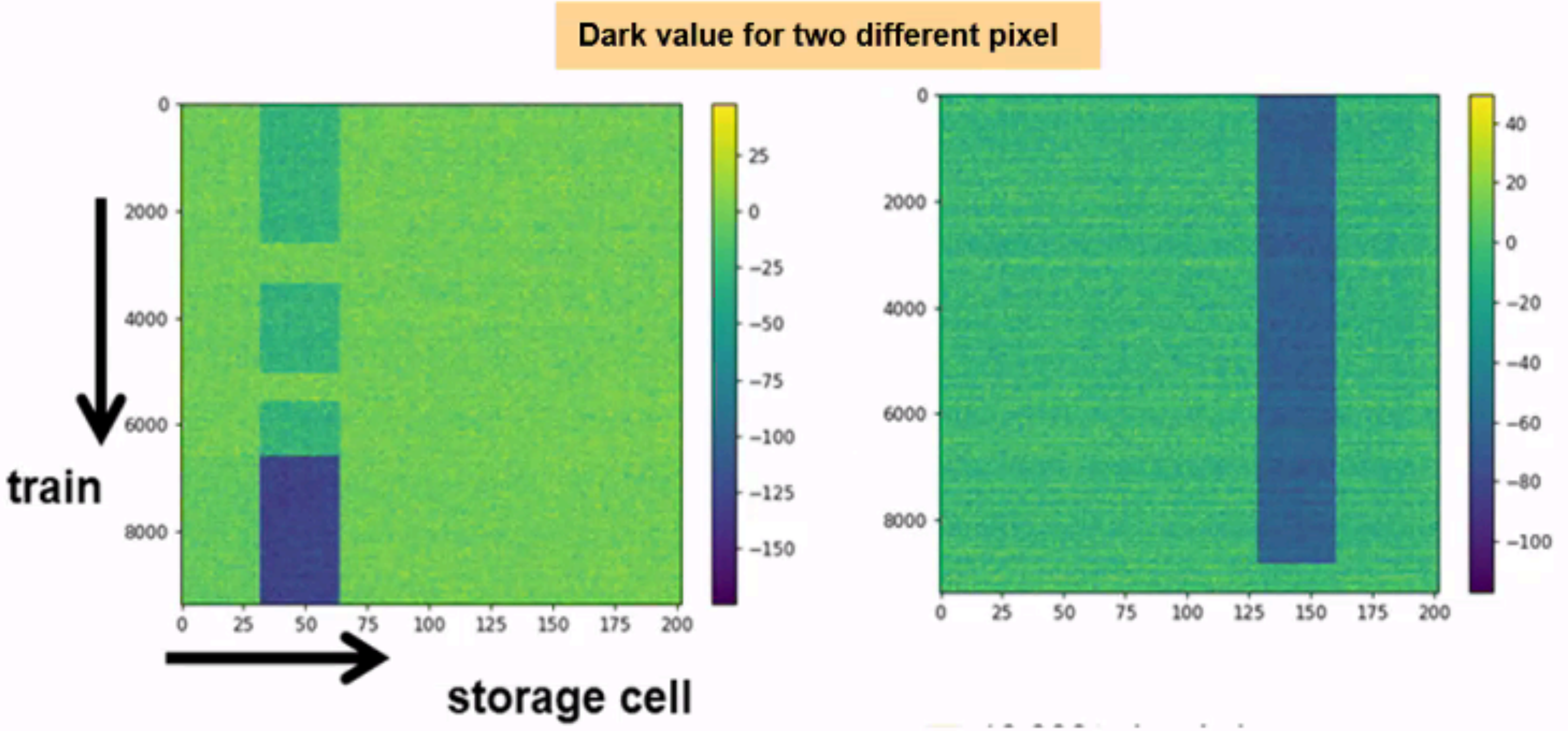
- Observation of squares in the TTCFs
  - Especially for low intensity experiments, with sparse scattering data on the detector (< 10^{-1} photons / pixel)
    - ▶ Most sensitive to small drifts in the pedestal





Jumping pixels and XPCS

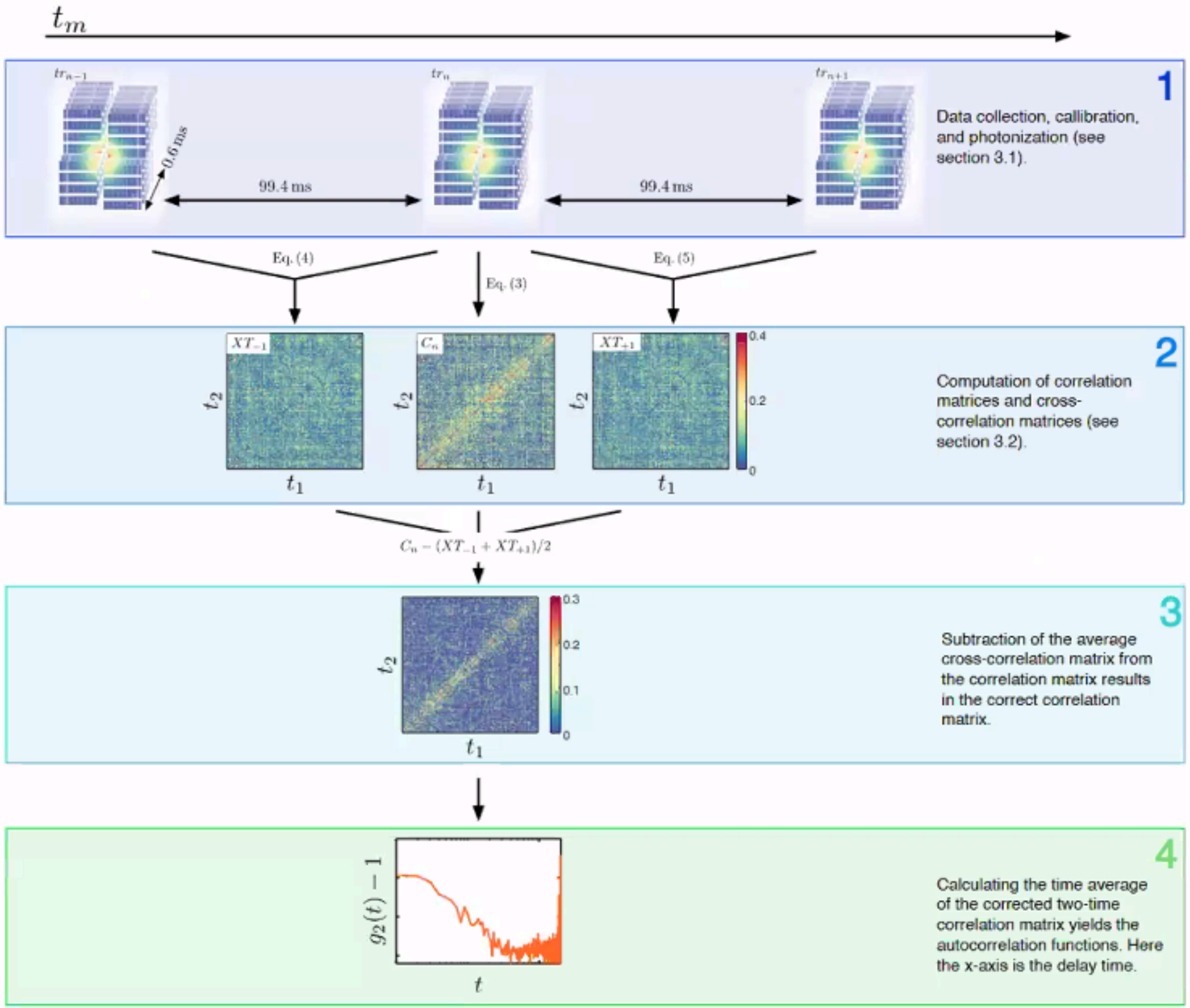
- Pedestals can fluctuate, drift and jump on various timescales
  - Jumps are rare but correlated
    - Blocks of 32 storage cells
  - Amplitude of jump: up to 1 photon
    - Negligible for most analysis
    - Serious artifact for XPCS
  - Most jumps corrected for by common mode correction
  - Remaining small but observable correlation of the pedestal of 32 storage blocks of each pixel
    - Can't be corrected in the detector calibration
    - Adaptation of XPCS data analysis



Cross-correlation correction

- Correlate what should be uncorrelated

- Inspired by corrections in fluctuation scattering
  - Spatial autocorrelation corrected by inter-frame correlation
- Here: Each train probes a new spot on the sample. Therefore speckle pattern from different trains should be uncorrelated.
  - If there is a correlation: From the detector and not the sample
- $g_2(q, tr_{n_1}, p_{m_1}, tr_{n_2}, p_{m_2}) = \frac{\langle I(q,tr_{n_1},p_{m_1}) I(q,tr_{n_2},p_{m_2})\rangle_q}{\langle I(q,tr_{n_1},p_{m_1})\rangle_q \langle I(q,tr_{n_2},p_{m_2})\rangle_q}$
  - Signal: $g_2(q, p_{m_1}, p_{m_2}) = \langle g_2(q, tr_{n_1}, p_{m_1}, tr_{n_2}, p_{m_2})\rangle_{tr_{n_1}=tr_{n_2}}$
  - Xcor: $g_2(q, p_{m_1}, p_{m_2}) = \langle g_2(q, tr_{n_1}, p_{m_1}, tr_{n_2}, p_{m_2})\rangle_{tr_{n_1}\neq tr_{n_2}}$



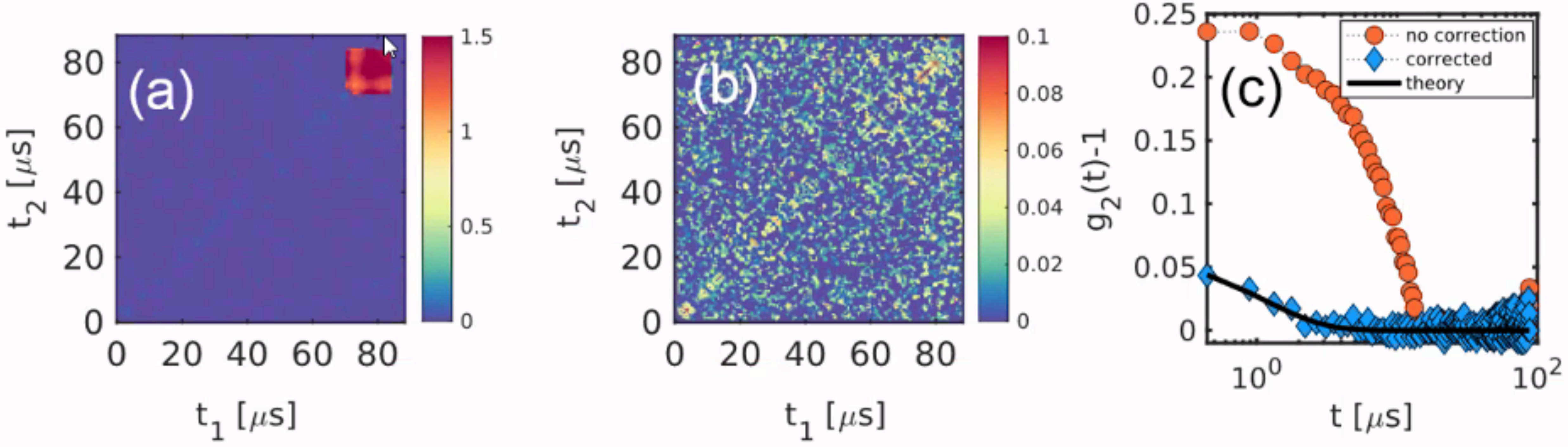
Madsen et al., *J. Synchrotron Rad.*, (2021). **28**, 637-649 Dallari et al. *Applied Sciences* 11.17 (2021): 8037

Cross-correlation correction

(a) Square shaped artifact in TTCF

(b) Cross-correlation corrected TTCF

(c) g_2 representation of the same two outputs, compared to the expected correlation function (theory)



Dallari et al. *Applied Sciences* 11.17 (2021): 8037

MHz XPCS at European XFEL

- MHz X-ray Photon Correlation Spectroscopy (XPCS)
  - European XFEL
  - Materials Imaging and Dynamics (MID) instrument
  - First user experiments
- Strategies for XPCS analysis at MID
  - Adaptive Gain Integration Pixel Detector (AGIPD)
  - Corrections
  - Analysis
- **Improvements and on-going developments**

Towards a general XPCS Pipeline

Felix Brauße, MID

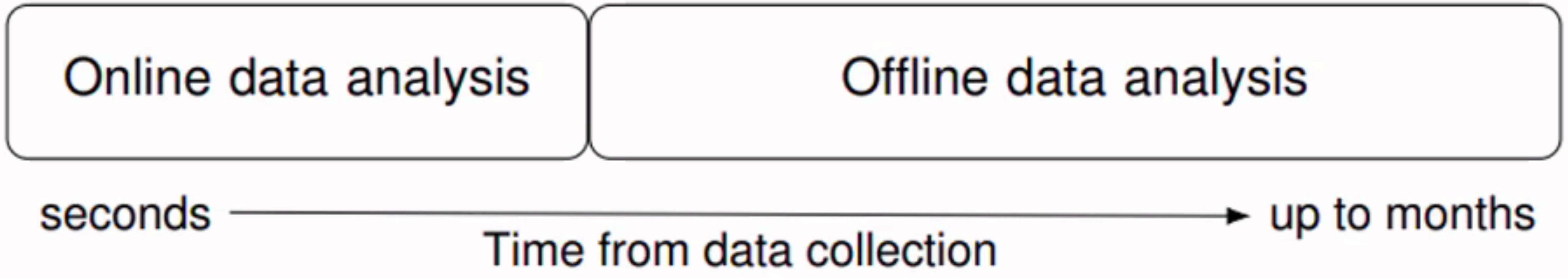
James Wrigley, DA

Wonhyuk Jo, MID

- Any of these blocks can be
  - Online or offline
  - CPU or GPU-based

- Incl. Photonization, sparsification, etc.

Offline and Online analysis



- High data rates are challenging
  - Data storage capacities
  - Long processing time prevent steering of experiment
- Online analysis
  - Stream data directly from the detector to calibration and analysis tools running on the online computing cluster at the instrument
- Offline analysis
  - Data transferred to the Maxwell HPC cluster @ DESY

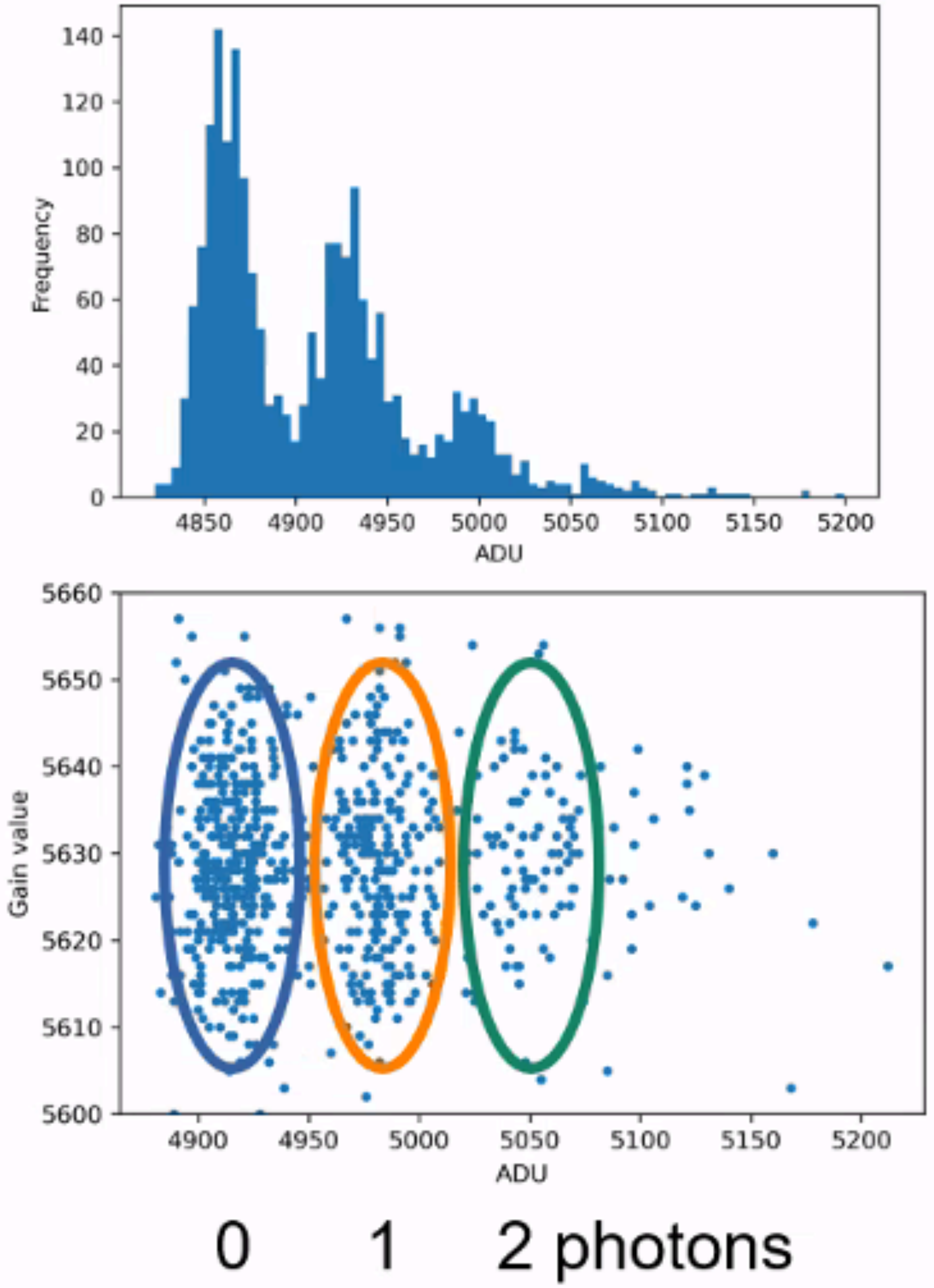
R&D Project GOAST: GPU-Computing for calibration & online analysis

- Online analysis of AGIPD data requires very efficient algorithms, especially when temporal or spatial correlations are desired due to the enormous amount of data per second

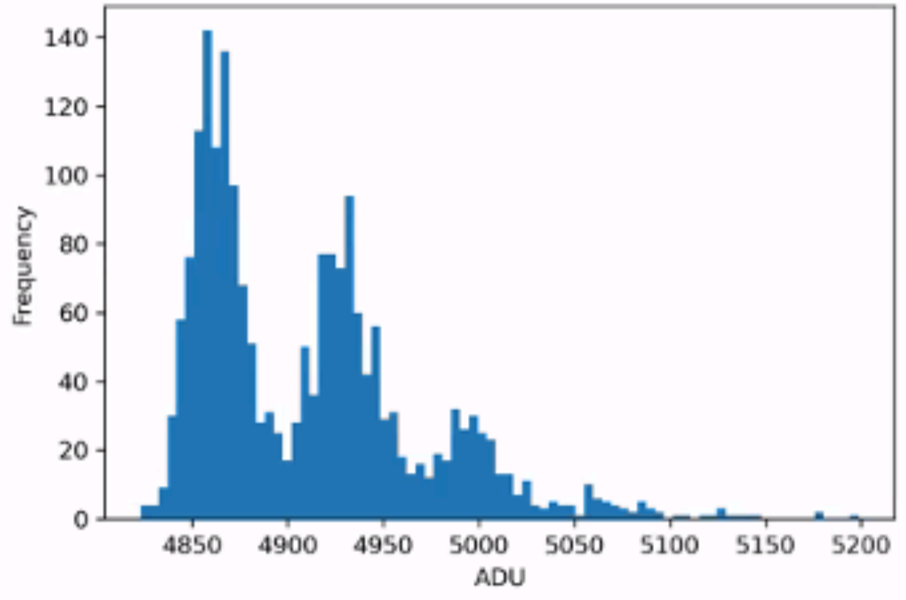
- We are exploring GPU-based **streaming algorithms,** which incrementally update calibration constants and analysis results with every processed image → ideal for live operation

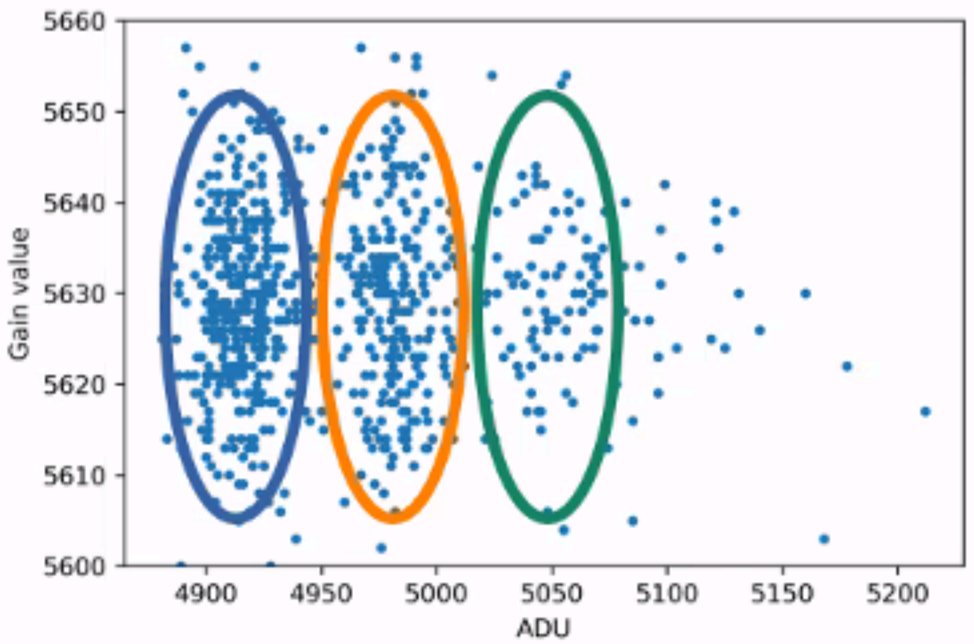
- Online calibration is based on **on-the-fly fitting** of a pixel's histogram (zero- and one-photon peak)

- At low count rates (<<1 phot/pix/pulse), we can use the GPU to remove all zero counts (saves storage on the order 1 minus count rate!)



Felix Brauße, MID



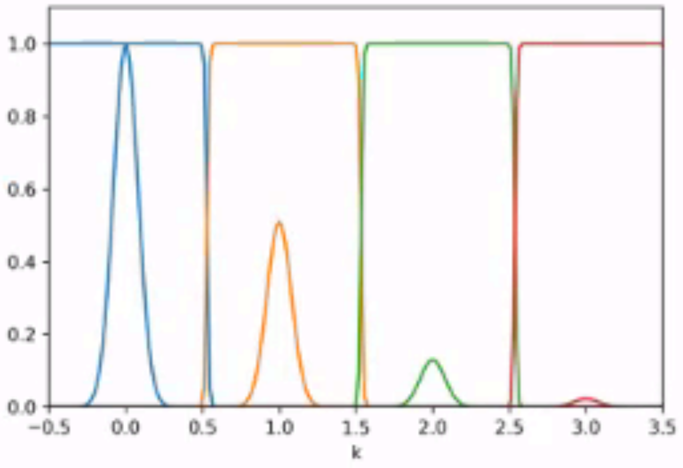


0 1 2 photons

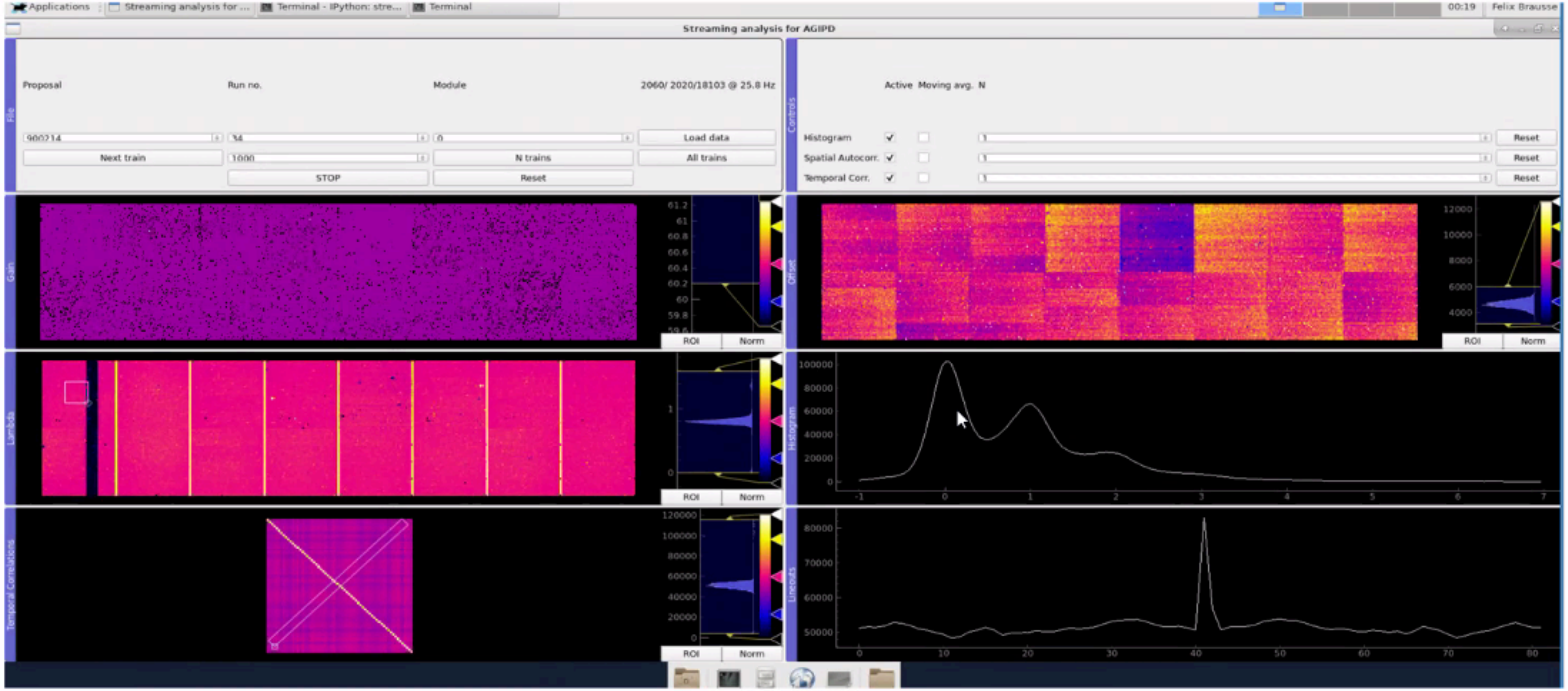
How Do We Fit the Model (without Histograms)?

- The Maximum-Likelihood principle tells us how; Maximize the likelihood function: $L(\boldsymbol{\theta},\mathbf{x})=\prod_i^n F(x_i,a,b,\lambda,\sigma)$
- Even better: Expectation Maximization: Exploit latent variables, here photon count *k*
- We can decompose $F(x_i,a,b,\lambda,\sigma)=\sum_k f_k(\boldsymbol{\theta},x_i)$ to get the likelihood function $L(\boldsymbol{\theta},\mathbf{x},\mathbf{k})=\prod_i^n \prod_k f_k(\boldsymbol{\theta},\mathbf{x})$
- Now we can optimize each component individually; using the weights

$T_{i,k}=\frac{f_k(\boldsymbol{\theta},x_i)}{\sum_k f_k(\boldsymbol{\theta},x_i)}$ we can use the Maximum-Likelihood update for a Gaussian: **the mean** $\mu_k^{(t+1)}=\frac{\sum_i T_{i,k}x_i}{\sum_i T_{i,k}}$

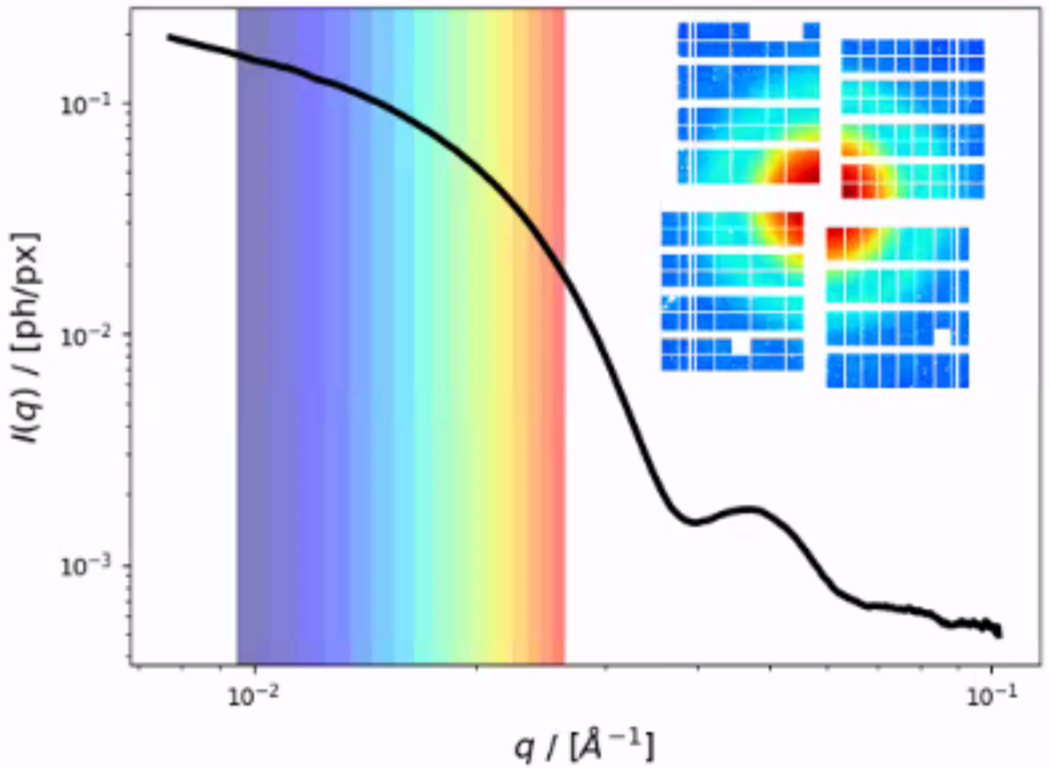


(this is the *k*-means algorithm)



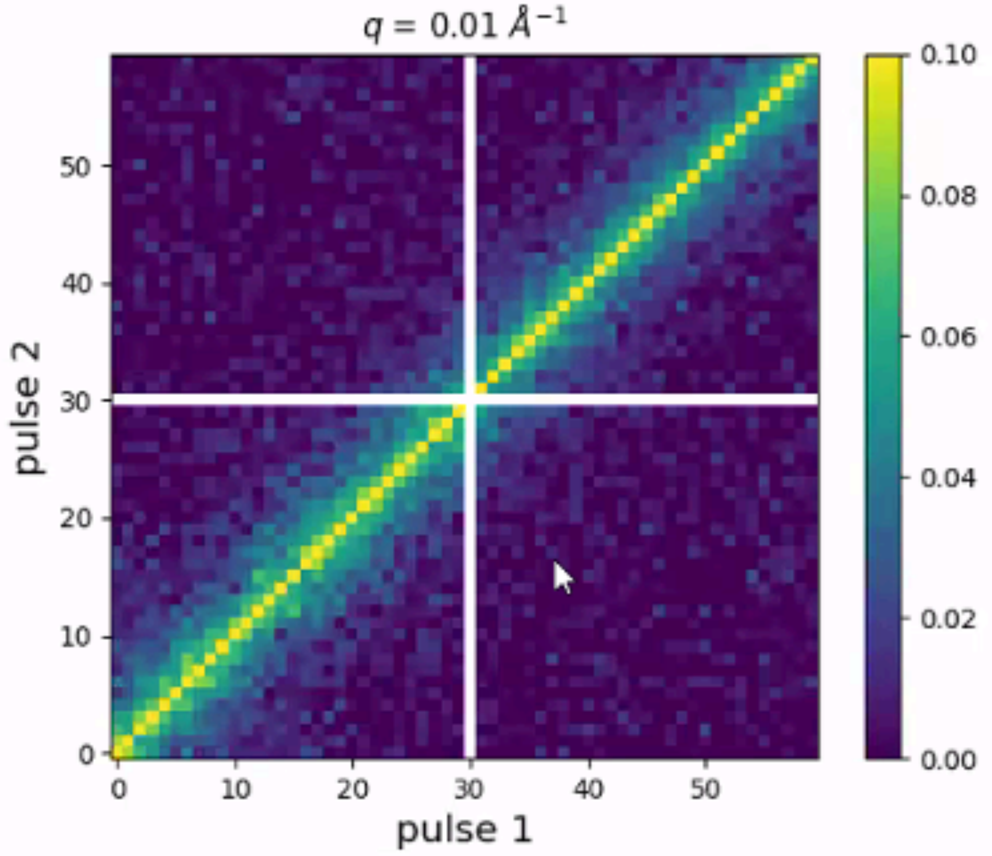
- Prototype of a streaming algorithm for calibration and online analysis under development together w/ experimental GUI for processing stored data
- Photon statistics, temporal correlations (XPCS) and spatial correlations (speckle analysis) implemented
- Processing rate > 100 Hz for 1 module, 352 memory cells (planned to run on GPU cluster to process all 16 modules)

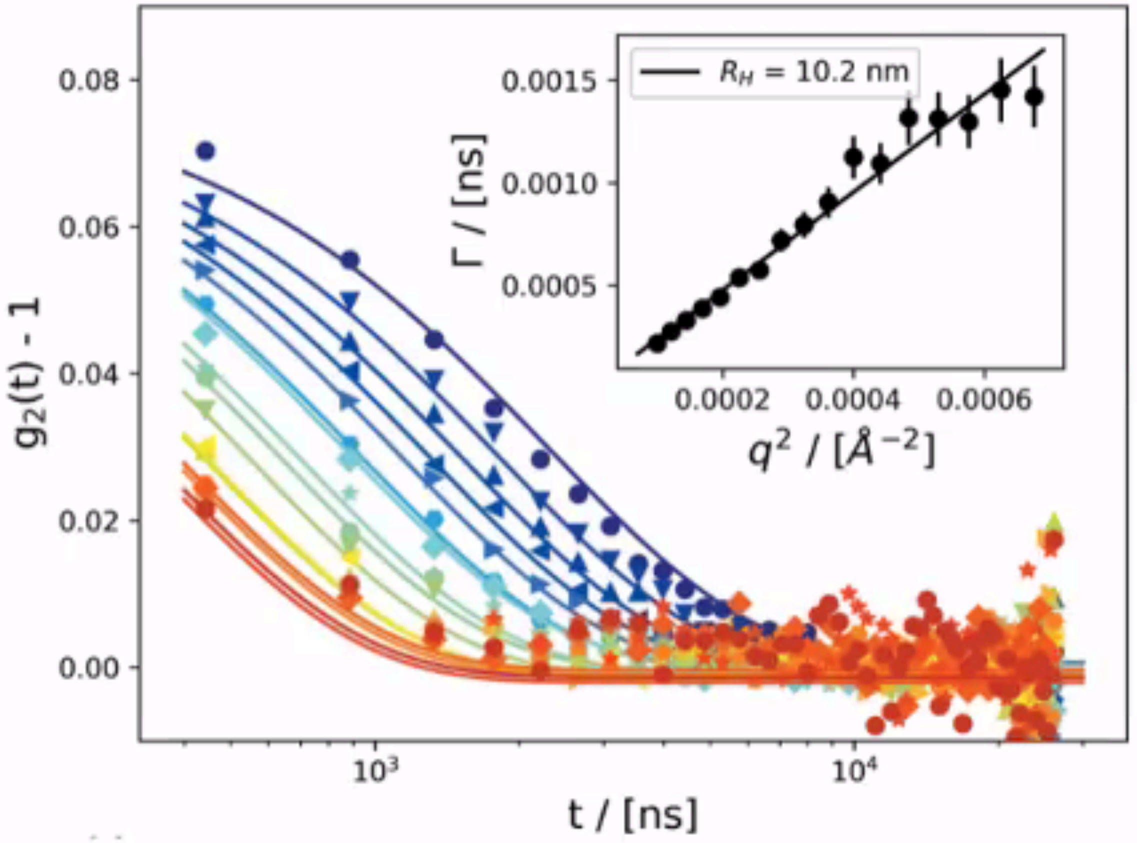
Results for Silica Particles



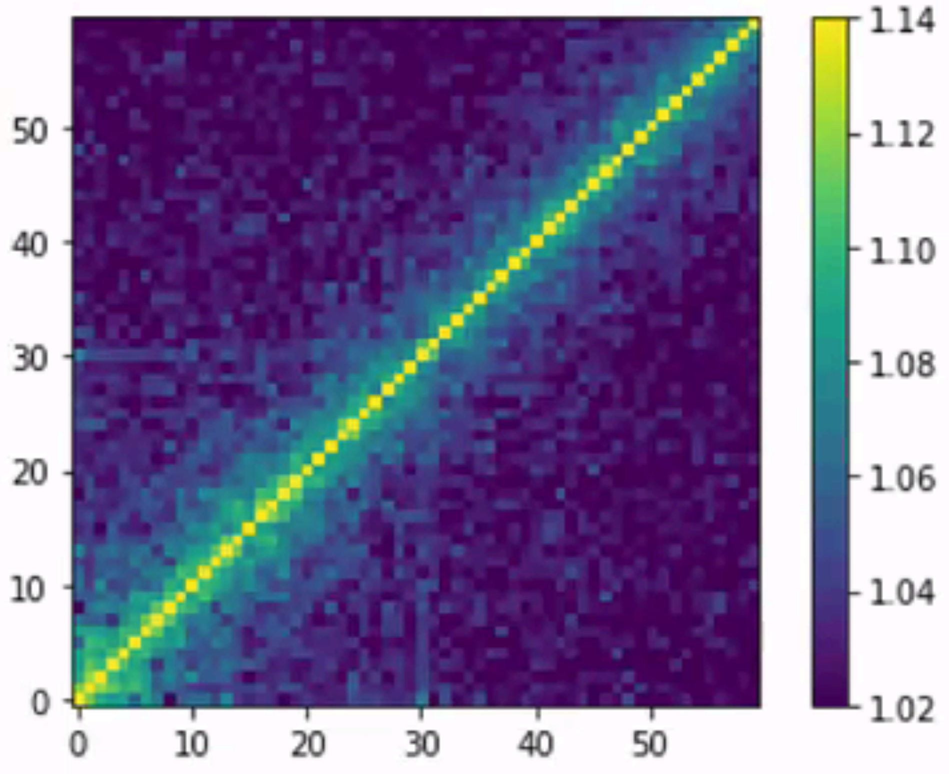
Published

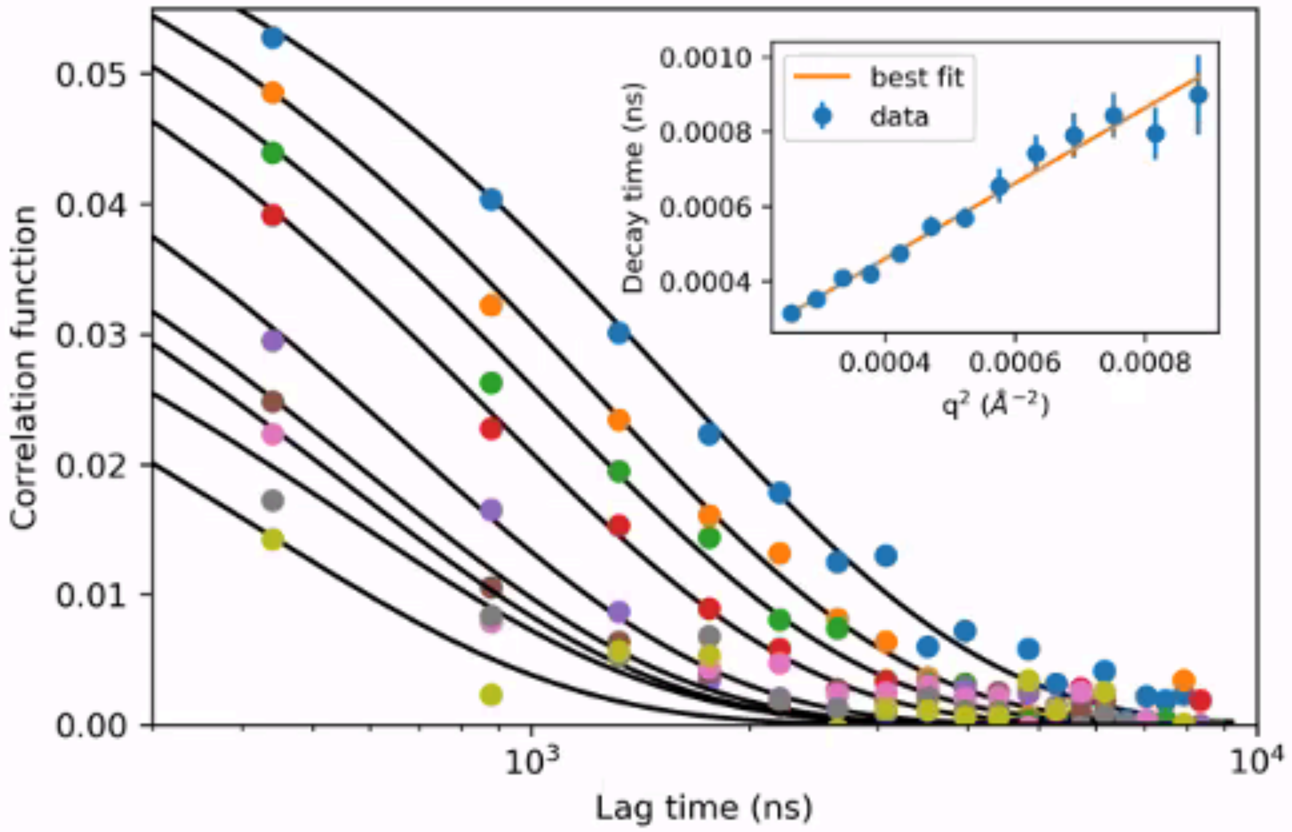
J. Synchrotron Rad. (2021). 28, 637





Streaming Algorithm





Towards a general XPCS Pipeline

- First approaches based on DASK
  - Fast and easy parallelization with distributed scheduler (multiple nodes) on Maxwell cluster
    - ▶ Good for data exploration and detector commissioning
    - ▶ With 10 nodes on the Maxwell cluster, full XPCS analysis in <30min for 5 minutes worth of data
  - Too resource draining and instable for building a larger workflow / analysis pipeline onto it

- Reduced on a single node
  - Start from mean and standard deviation of detector image for beam centering and pixel masking
    - ▶ ~ 5 hrs on 1 node for 5 minutes worth of data
  - Run TTCs, possibly w/ different parameters
    - ▶ 56 hrs on 1 node for 5 minutes worth of data

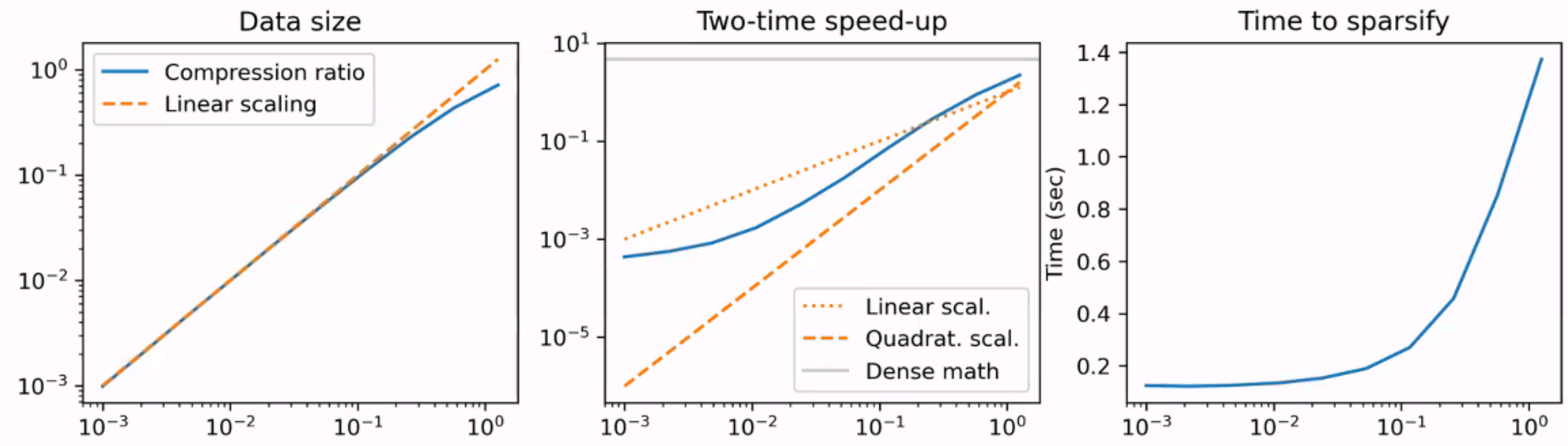
“On-line inspired” processing

- Maximize number of processes (~100 on 1 node)
- 1 process acesses 1 file (per module per 256 trains)
  - Read 1 train at a time, do streaming math
  - -> disk access is well distributed, no memory problems
- Mean and standard deviation take **15 minutes** for all 16 modules
- Dense TTCs: ~ 30 mins
- Sparse TTCs at 1x10^{-2} ph/pix/pulse: **~9 mins; speedup of ~x300**

$$\begin{aligned} \delta_i &= x_i - \mathrm{EMA}_{i-1} \\ \mathrm{EMA}_i &= \mathrm{EMA}_{i-1} + \alpha \cdot \delta_i \\ \mathrm{EMVar}_i &= (1-\alpha)\left(\mathrm{EMVar}_{i-1} + \alpha \cdot \delta_i^2\right) \end{aligned}$$

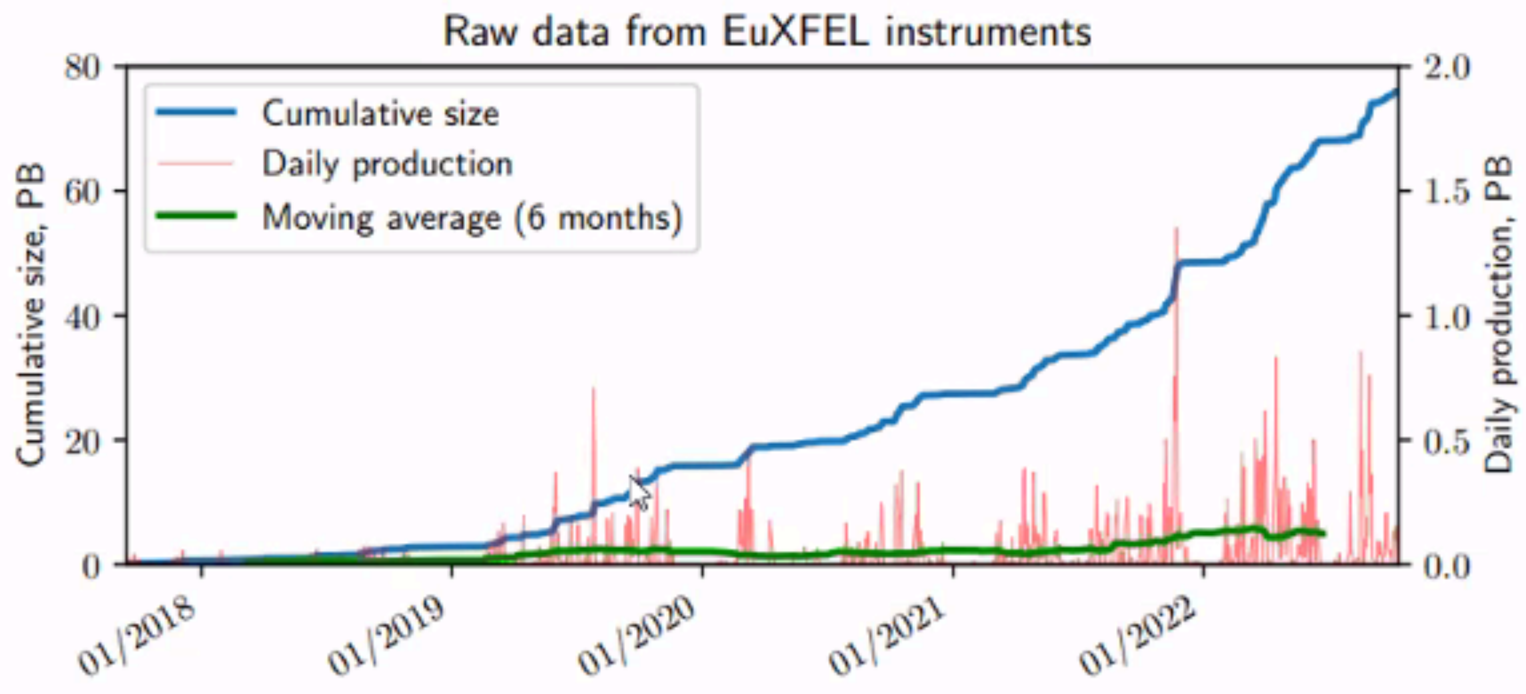
Sparse Data and Sparse Math

- Calibrated data is 32 bit float
- For maximum efficiency we use bit packing
  - 16 bits pixel, 9 bits storageCell, 7 bits value (i.e. up to 127 photons)

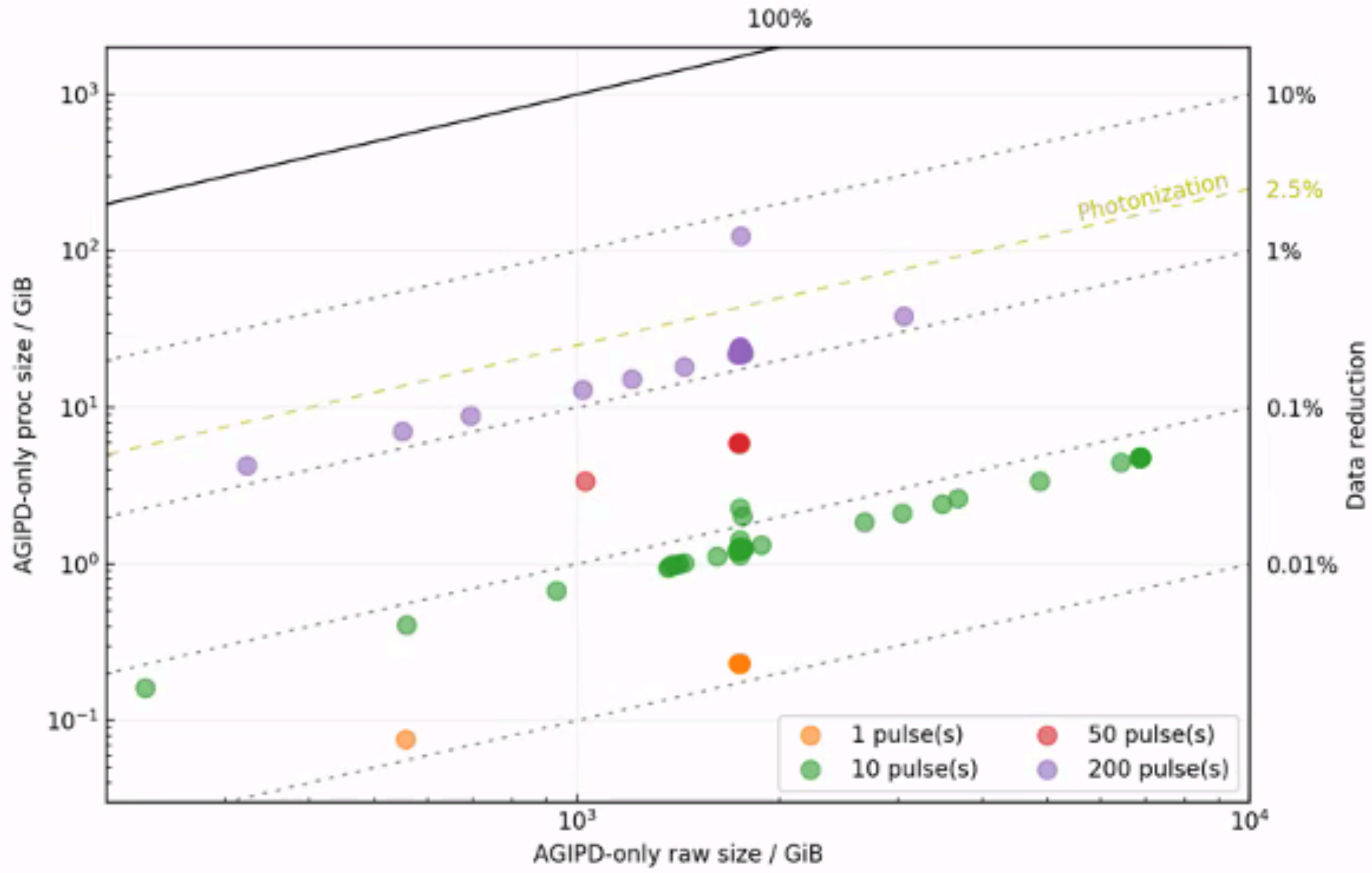


Data reduction

- The current rate of data collection is not sustainable
  - Emphasized stability over performance during the start of the facility
- Improving data correction and reduction routines
  - Can't always change the detector operation scenario, e.g. changing the number of pulses in a train
    - Automatic removal of empty frames
    - "photonization" float32 -> uint16
    - XPCS pipeline for automated reduction to 1D I(q) or g_2, 2D TTCF outpus



The best(?) week so far: 7 PB in 7 days.



Summary

- MHz XPCS is a well established technique at MID of EuXFEL
  - Initial struggles due to large data rates and limited (beam-)time for detector characterization
  - Many issues could be tackled by improving the detector corrections as well as the analysis strategy
  - Current developments on the software side target automation, shorter run times, data reduction and possibly improved signal-to-noise ratio for low intensity speckle data

- Developments on instrumentation side
  - Non-sequential pulse pattern (double-shot, rolling-bunch-pattern, split&delay) XPCS
    - Shorter timescales < 220 ns
    - Improved signal-to-dose and signal-to-noise ratios for radiation sensitive samples
  - Wide-angle (atomic length scale) XPCS remains challenging
    - Ongoing efforts in commissioning of hard X-ray self-seeding (improved longitudinal coherence lenght) & focussing