Feature extraction for the shashlyk calorimeter

Markus Preston, Per-Erik Tegnér

Gratefully acknowledging the help from
Stefan Diehl and the JLU Gießen group

PANDA Collaboration Meeting, GSI, 2019-11-06

The PANDA electromagnetic calorimeters

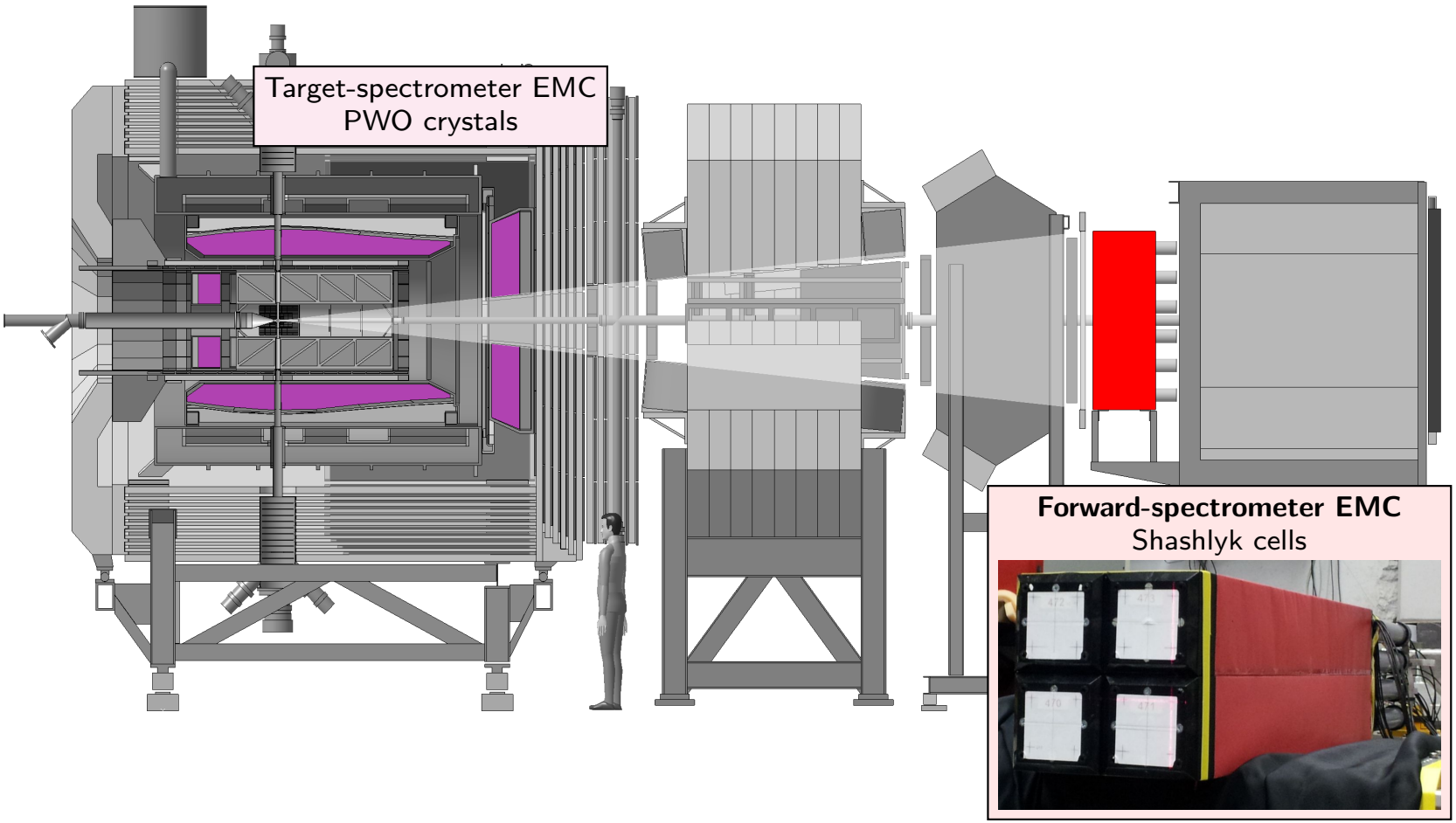
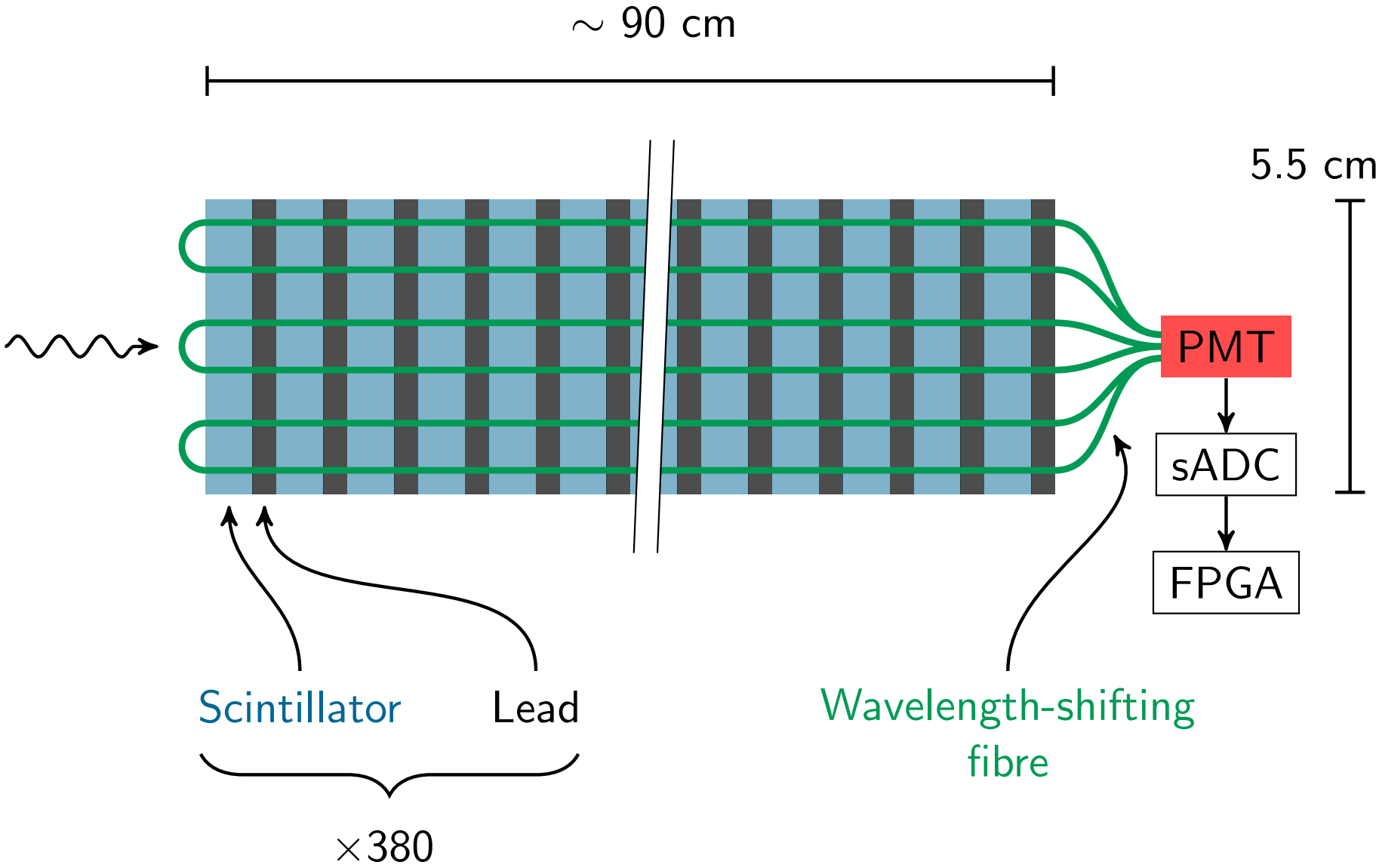


Figure adapted from https://panda.gsi.de/oldwww
Photo from S. Diehl, Ph.D. thesis, JLU Gießen (2016)

Detector structure

Side view of cell



Aim of this work

- Find an FPGA triggering/feature extraction algorithm, optimised with respect to:
  - Pulse identification (triggering)
  - Energy resolution
  - Time resolution
  - Pile-up identification/reconstruction

Aim of this work

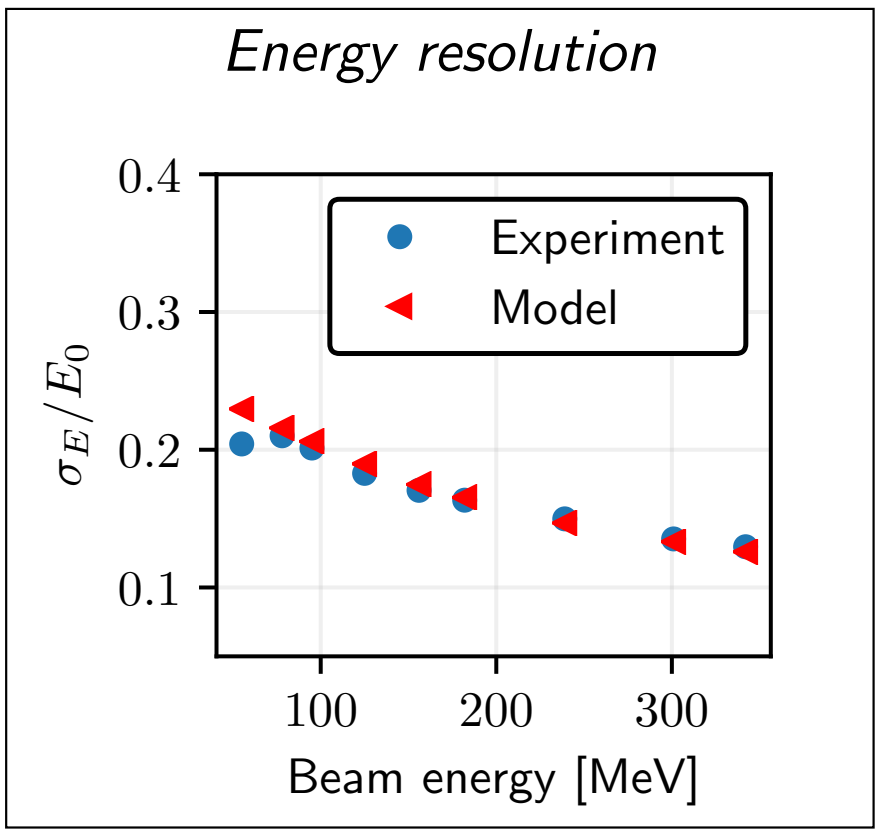
- Find an FPGA triggering/feature extraction algorithm, optimised with respect to:
  - Pulse identification (triggering)
  - Energy resolution
  - Time resolution
  - Pile-up identification/reconstruction
- How?
  - We have developed a Monte Carlo model of a 4×4 prototype (starting with Geant4). *Talk at June 2019 CM.*
  - Use the Monte Carlo model to generate pulses with known underlying energy, time and pile-up information. Evaluate performance of feature-extraction algorithms. **This talk.**

Recap of June presentation

- 2014 MAMI photon testbeam data provided by Gießen group.
- Model of detector, photon transport, photoelectron generation and readout developed at Stockholm University
  $\Rightarrow$ realistic description of event-by-event fluctuations.

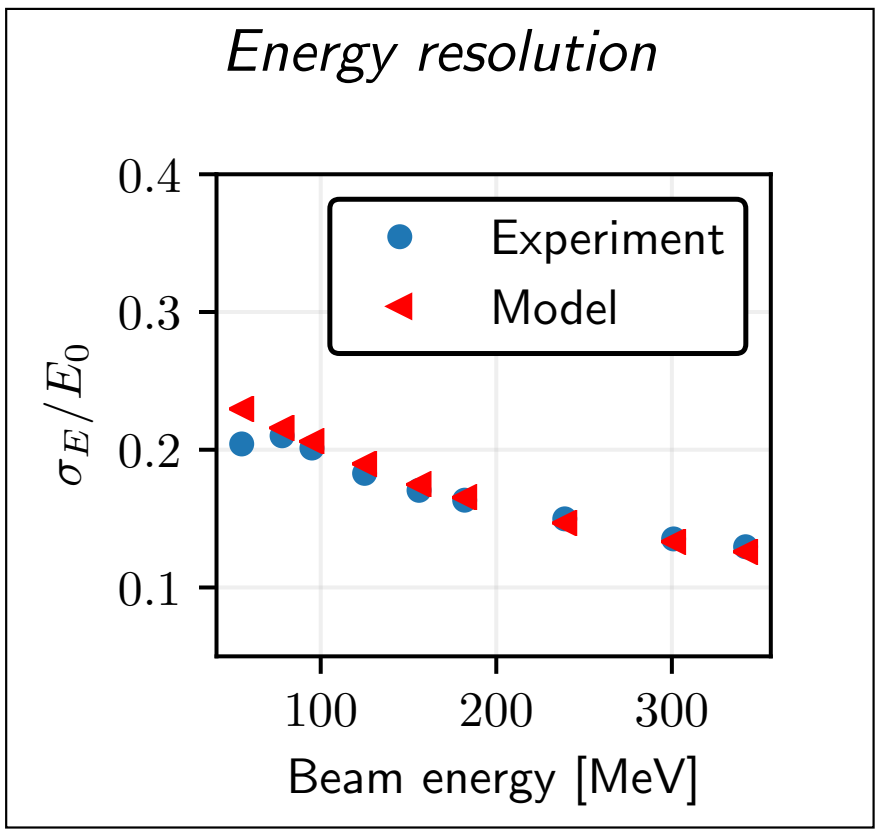
Recap of June presentation

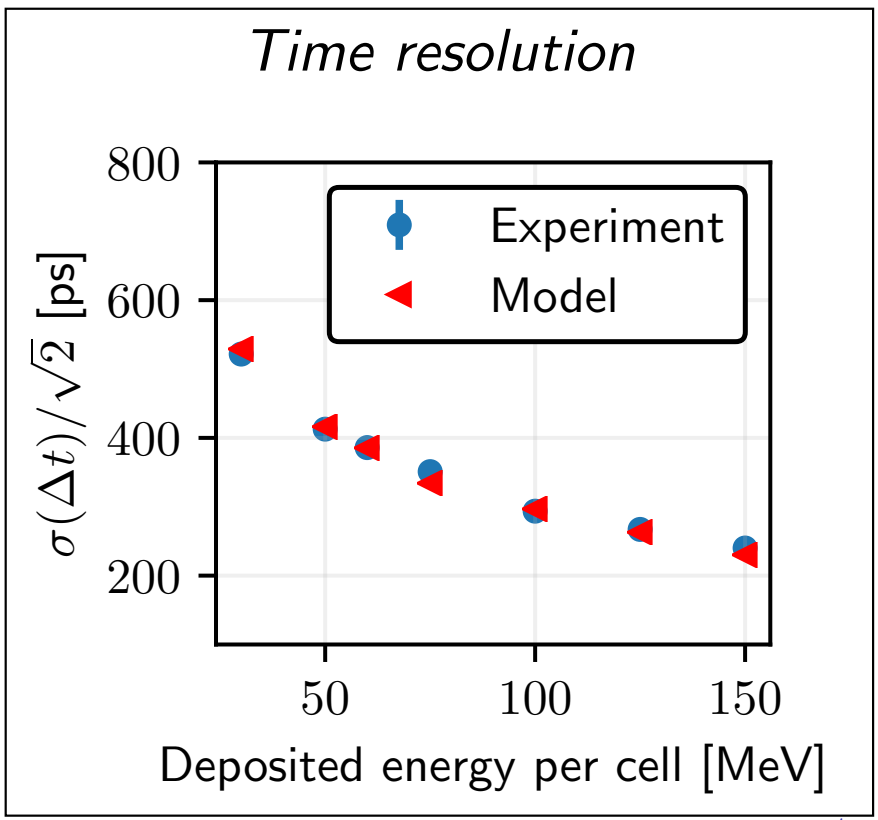
- 2014 MAMI photon testbeam data provided by Gießen group.
- Model of detector, photon transport, photoelectron generation and readout developed at Stockholm University
  $\Rightarrow$ realistic description of event-by-event fluctuations.
- Model validated:



Recap of June presentation

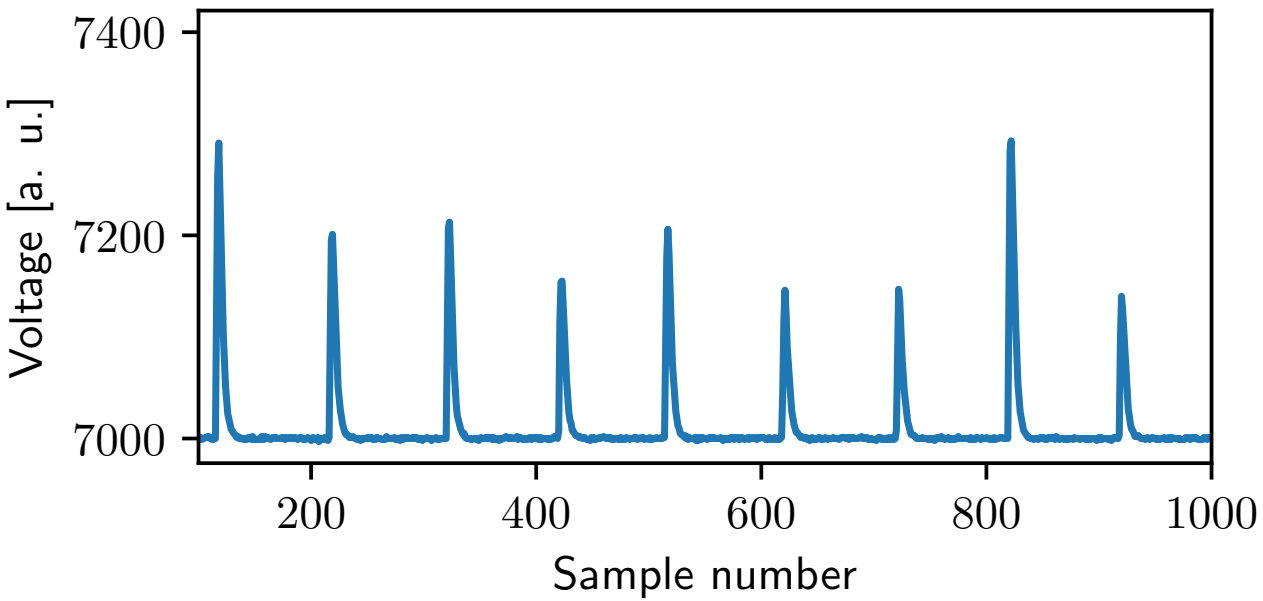
- 2014 MAMI photon testbeam data provided by Gießen group.
- Model of detector, photon transport, photoelectron generation and readout developed at Stockholm University
  $\Rightarrow$ realistic description of event-by-event fluctuations.
- Model validated:





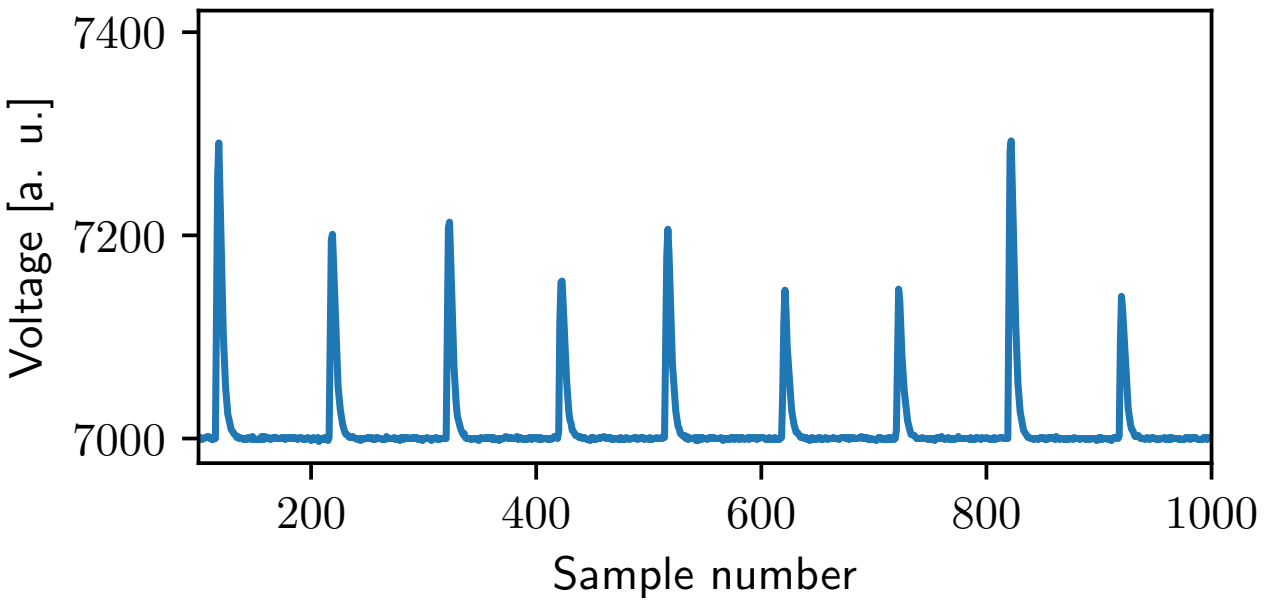
Evaluating feature extraction — method

1. Generate well-separated signals



Evaluating feature extraction — method

1. Generate well-separated signals

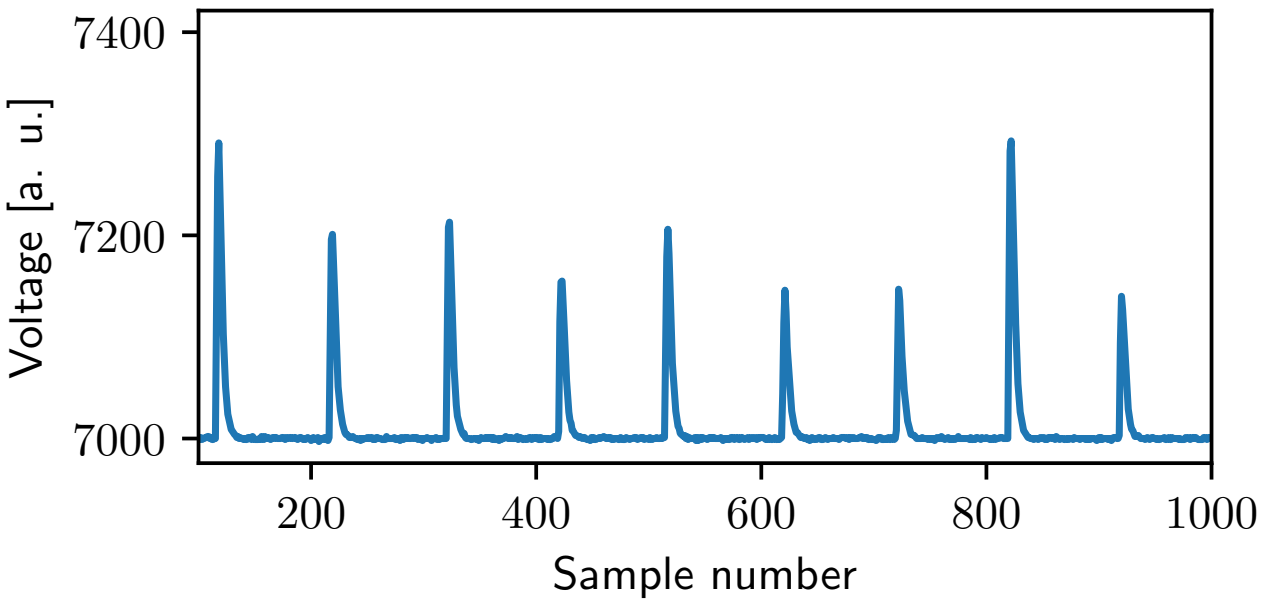


2. Use as input to VHDL code

Feature-extraction algorithms (pulse height, timing, ...)

Evaluating feature extraction — method

1. Generate well-separated signals



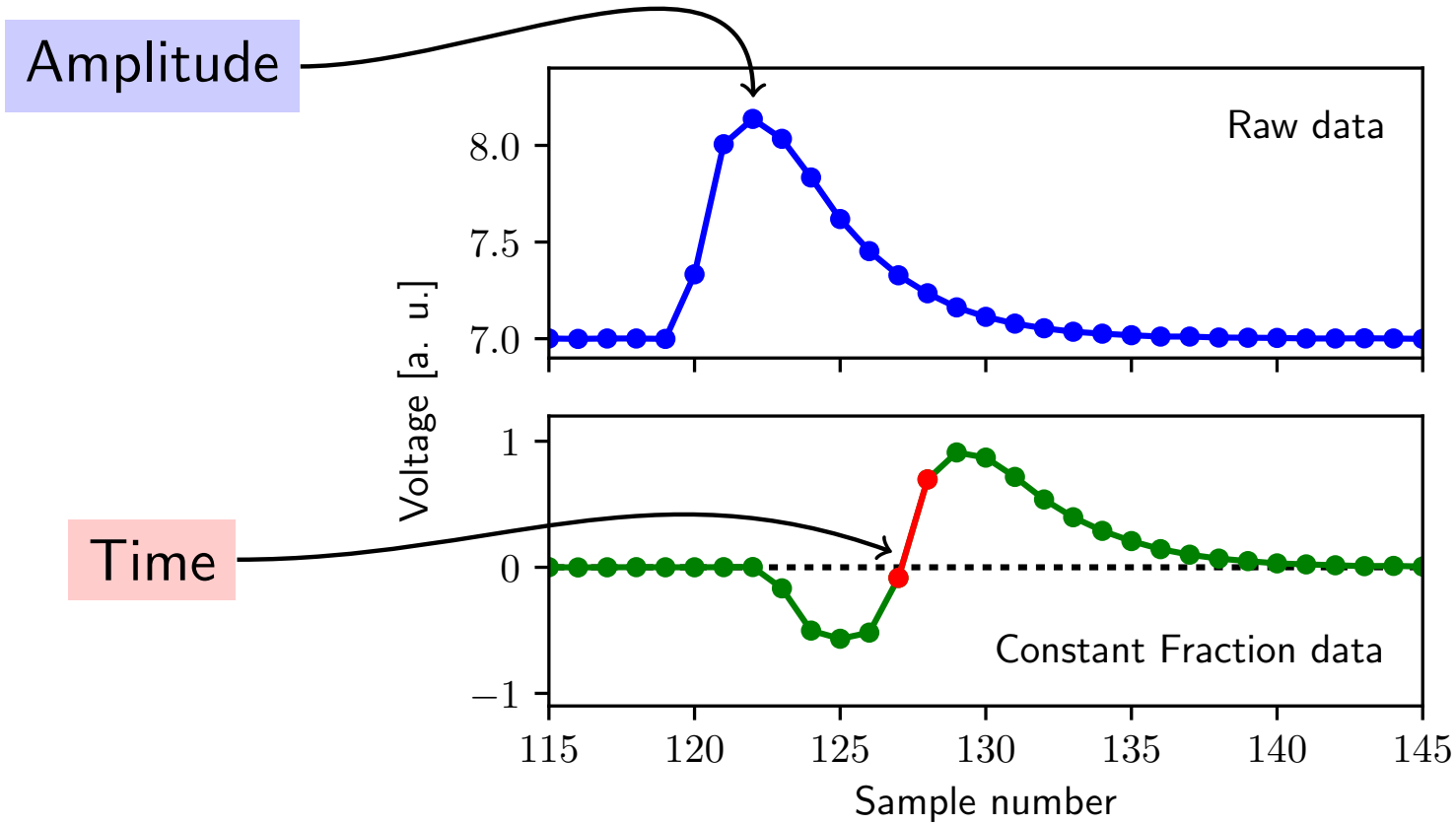
2. Use as input to VHDL code

Feature-extraction algorithms (pulse height, timing, ...)

3. Evaluate algorithm performance

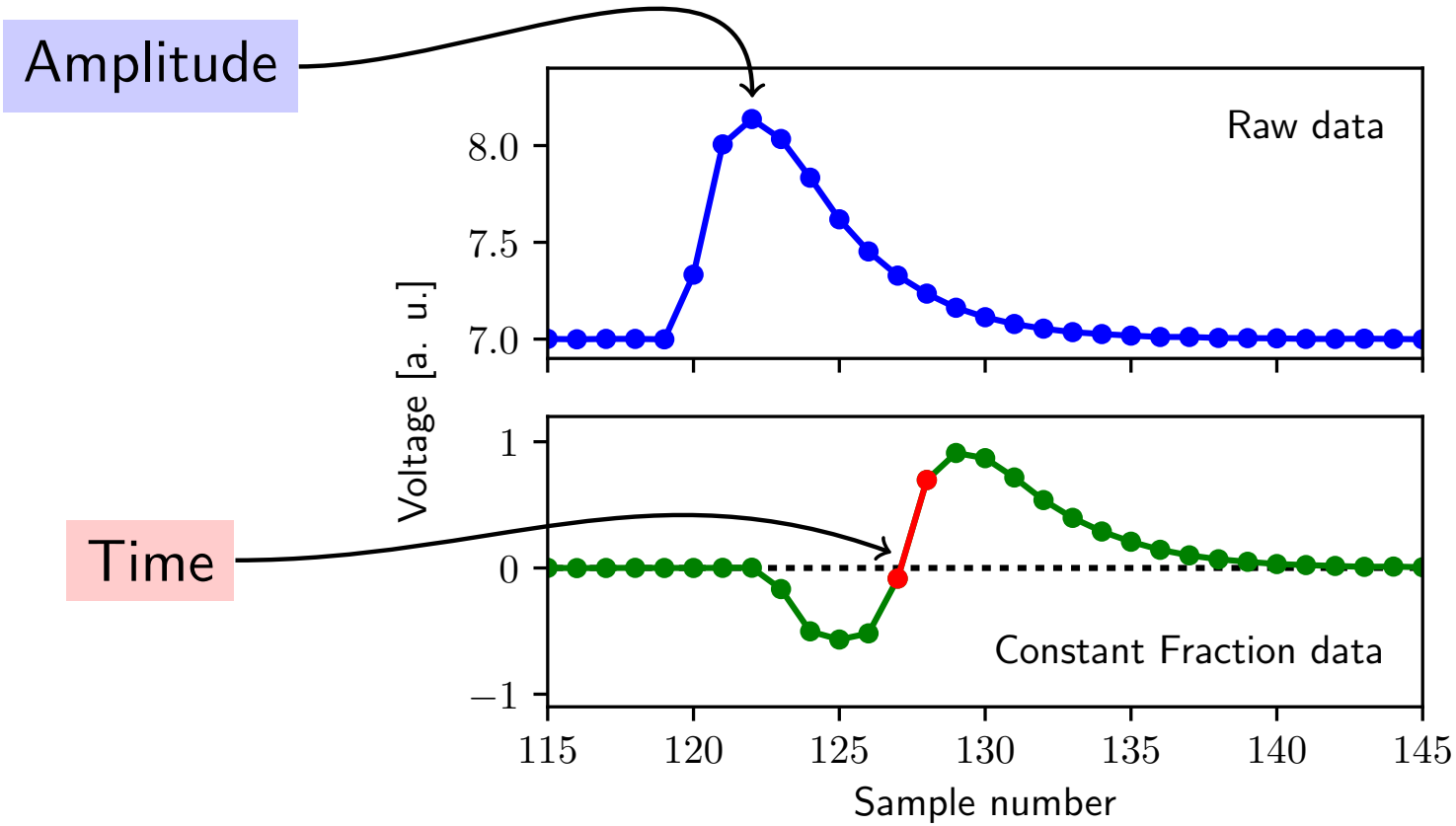
Existing algorithm

- Amplitude from highest sample in pulse
- Time from Constant Fraction method (linear interpolation)



Existing algorithm

- Amplitude from highest sample in pulse
- Time from Constant Fraction method (linear interpolation)



- Validity confirmed (analysis of 2014 testbeam data)
- Potential issues: (i) does not solve pile-up problem, (ii) relies on linear interpolation between two samples.

New algorithm

- Based on *optimal filter*, used for example in ATLAS
- Maximises signal-to-noise ratio (when noise = Gaussian)

New algorithm

- Based on *optimal filter*, used for example in ATLAS
- Maximises signal-to-noise ratio (when noise = Gaussian)
- Assume **fixed pulse shape** $g(t_i)$
- Free parameters: **Amplitude** A and **phase** τ

$$f(t_i) = Ag(t_i - \tau) \approx Ag(t_i) - A\tau g'(t_i)$$

New algorithm

- Based on *optimal filter*, used for example in ATLAS
- Maximises signal-to-noise ratio (when noise = Gaussian)
- Assume **fixed pulse shape** $g(t_i)$
- Free parameters: **Amplitude** A and **phase** τ

$$f(t_i) = Ag(t_i - \tau) \approx \boxed{Ag(t_i) - A\tau g'(t_i)}$$

Assume small $\tau \rightarrow$ linearise

New algorithm

- Based on *optimal filter*, used for example in ATLAS
- Maximises signal-to-noise ratio (when noise = Gaussian)
- Assume **fixed pulse shape** $g(t_i)$
- Free parameters: **Amplitude** A and **phase** τ

$$f(t_i) = Ag(t_i - \tau) \approx \boxed{Ag(t_i) - A\tau g'(t_i)}$$

Assume small $\tau \rightarrow$ linearise

- Construct Finite Impulse Response (FIR) filter with carefully selected sets of coefficients **a** and **b**.
- $A = \sum a_i S_i$
- $A\tau = \sum b_i S_i$

New algorithm

- Based on *optimal filter*, used for example in ATLAS
- Maximises signal-to-noise ratio (when noise = Gaussian)
- Assume **fixed pulse shape** $g(t_i)$
- Free parameters: **Amplitude** A and **phase** τ

$$f(t_i) = Ag(t_i - \tau) \approx \boxed{Ag(t_i) - A\tau g'(t_i)}$$

Assume small $\tau \rightarrow$ linearise

- Construct Finite Impulse Response (FIR) filter with carefully selected sets of coefficients **a** and **b**.
- $A = \sum a_i S_i$
- $A\tau = \sum b_i S_i$
- Sets of coefficients are fixed $\rightarrow$ calculated offline $\rightarrow$ FPGA implementation possible!

New algorithm — new developments

- *Optimal filter* method has been refined to address (i) free-running trigger nature of PANDA, (ii) pile-up reconstruction

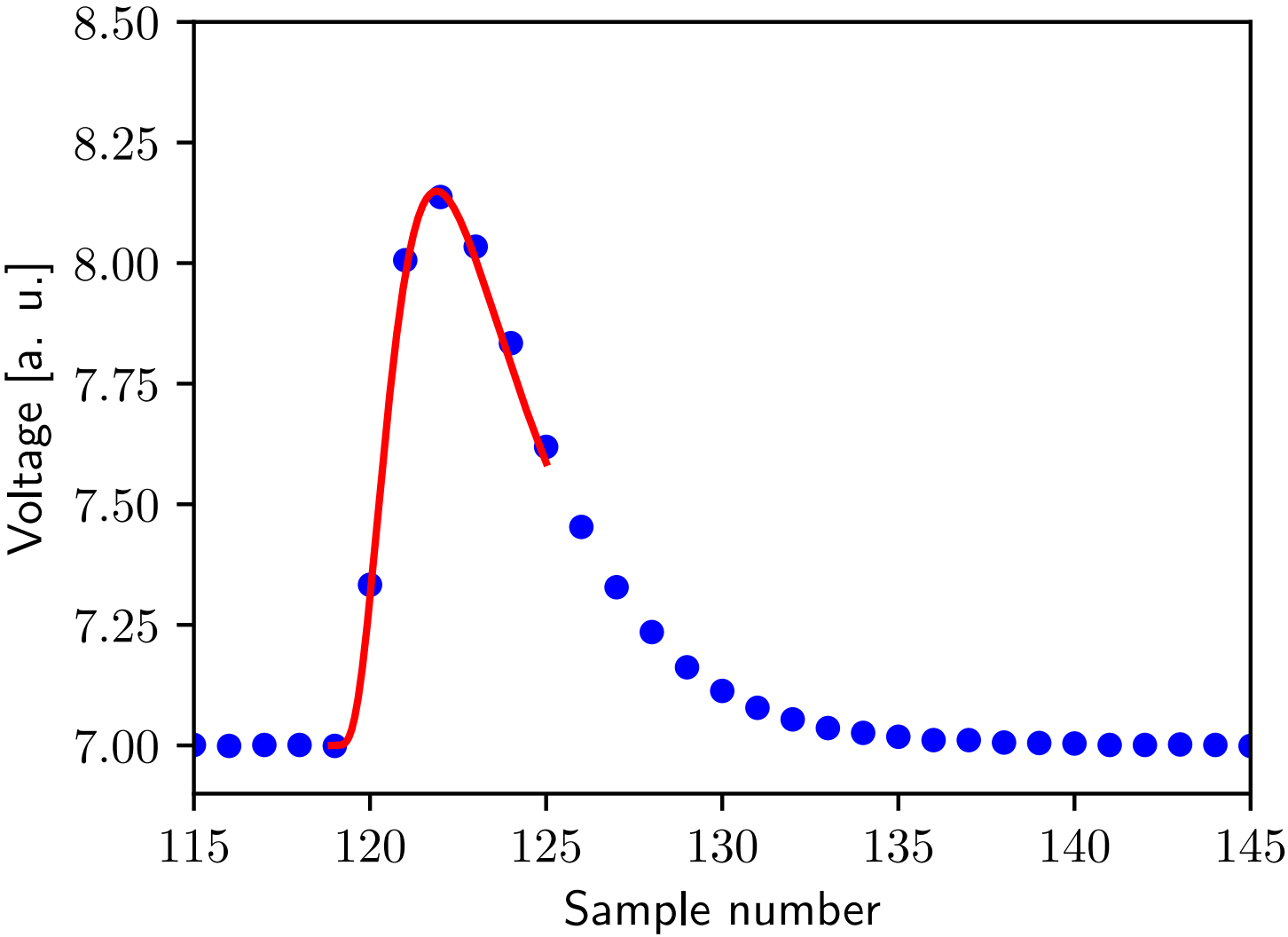
New algorithm — new developments

- *Optimal filter* method has been refined to address (i) free-running trigger nature of PANDA, (ii) pile-up reconstruction
- New developments:
  1. In PANDA, pulses not necessarily in phase with sampling clock.
     - Make first, crude, estimate of pulse phase using a **simplified** Constant Fraction algorithm (binary search).
     - Then, apply principles of optimal filter.

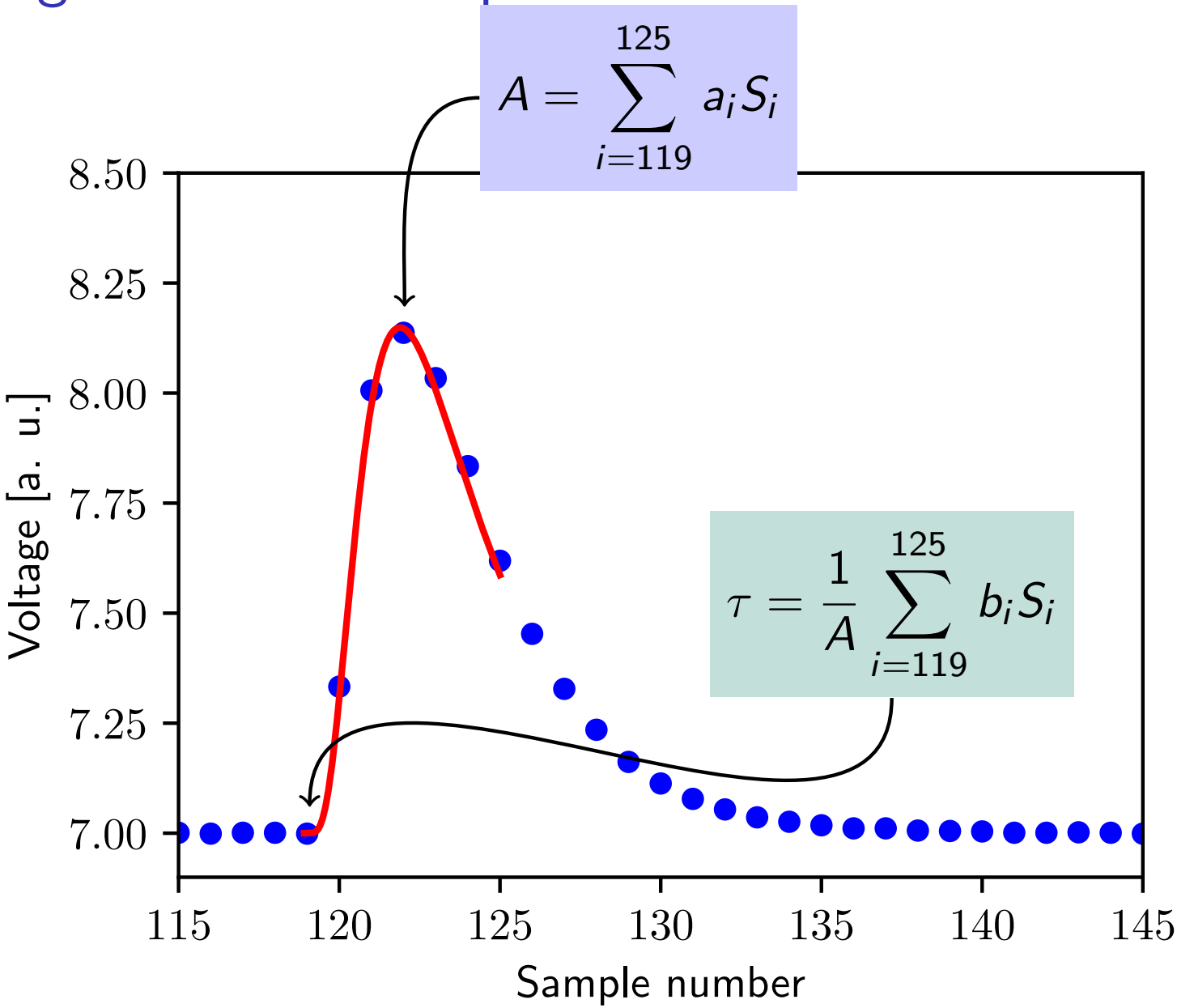
New algorithm — new developments

- *Optimal filter* method has been refined to address (i) free-running trigger nature of PANDA, (ii) pile-up reconstruction
- New developments:
  1. In PANDA, pulses not necessarily in phase with sampling clock.
     - Make first, crude, estimate of pulse phase using a **simplified** Constant Fraction algorithm (binary search).
     - Then, apply principles of optimal filter.
  2. Make pileup reconstruction/recovery possible.
     - Limit number of optimal filter coefficients to first part of pulse.
     - See later slides.

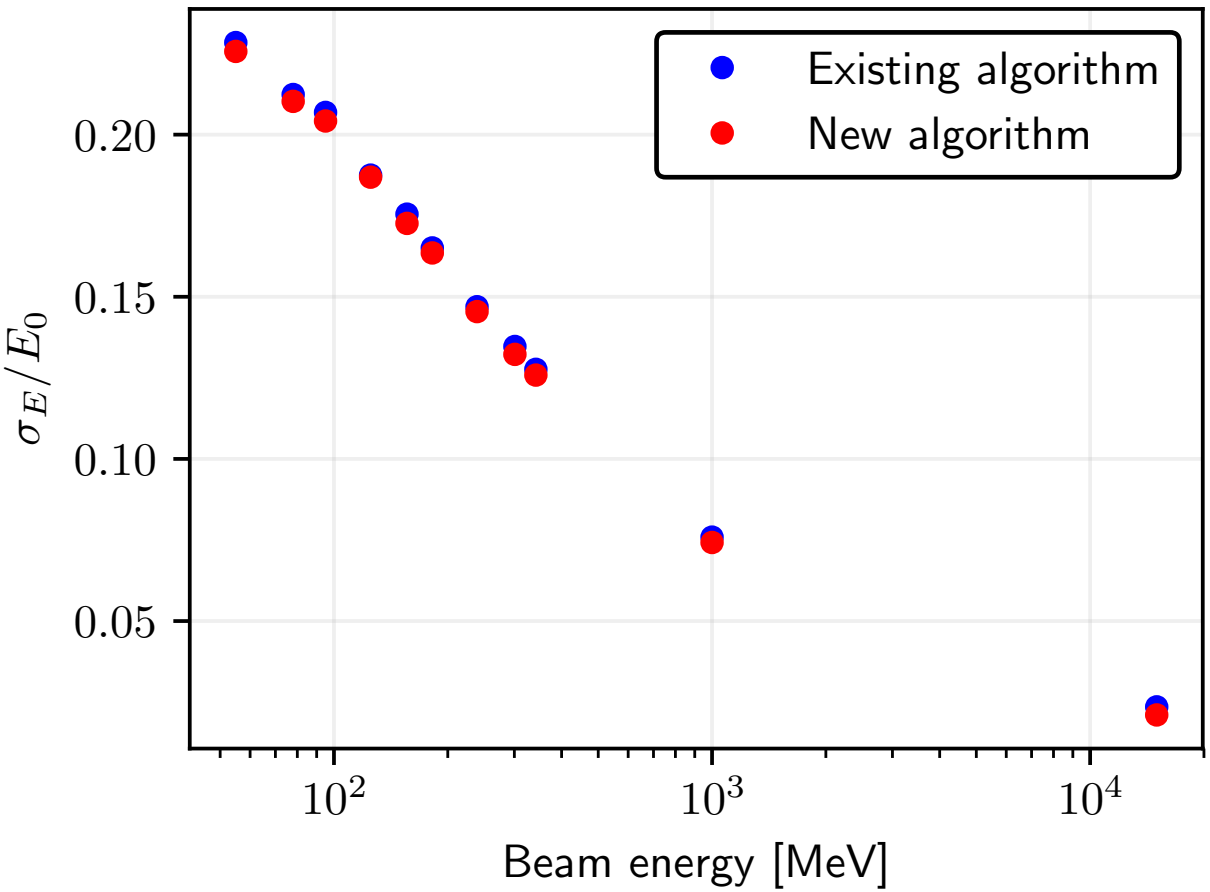
New algorithm — example



New algorithm — example

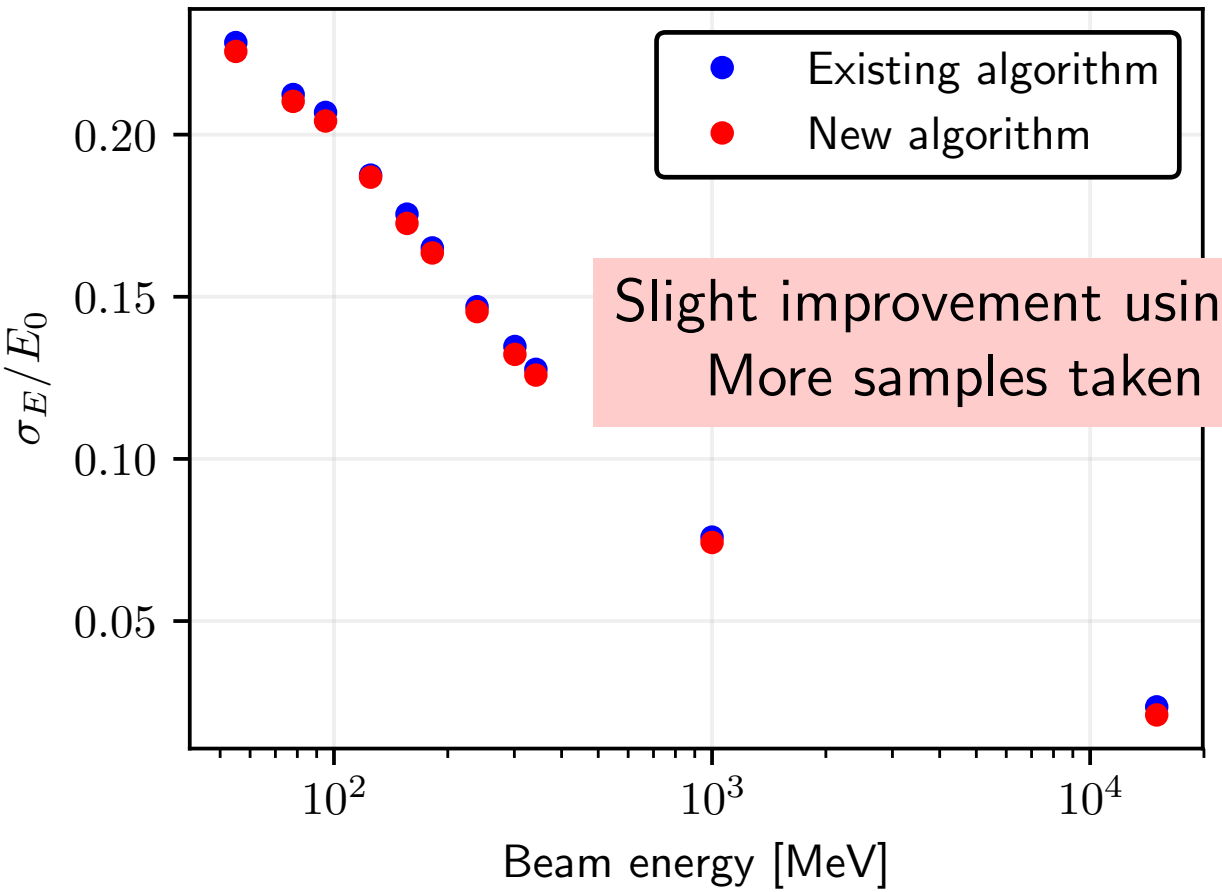


Results — energy resolution



Relative energy resolution of single cell determined by fitting Novosibirsk distribution to pulse-height (amplitude A) spectra.

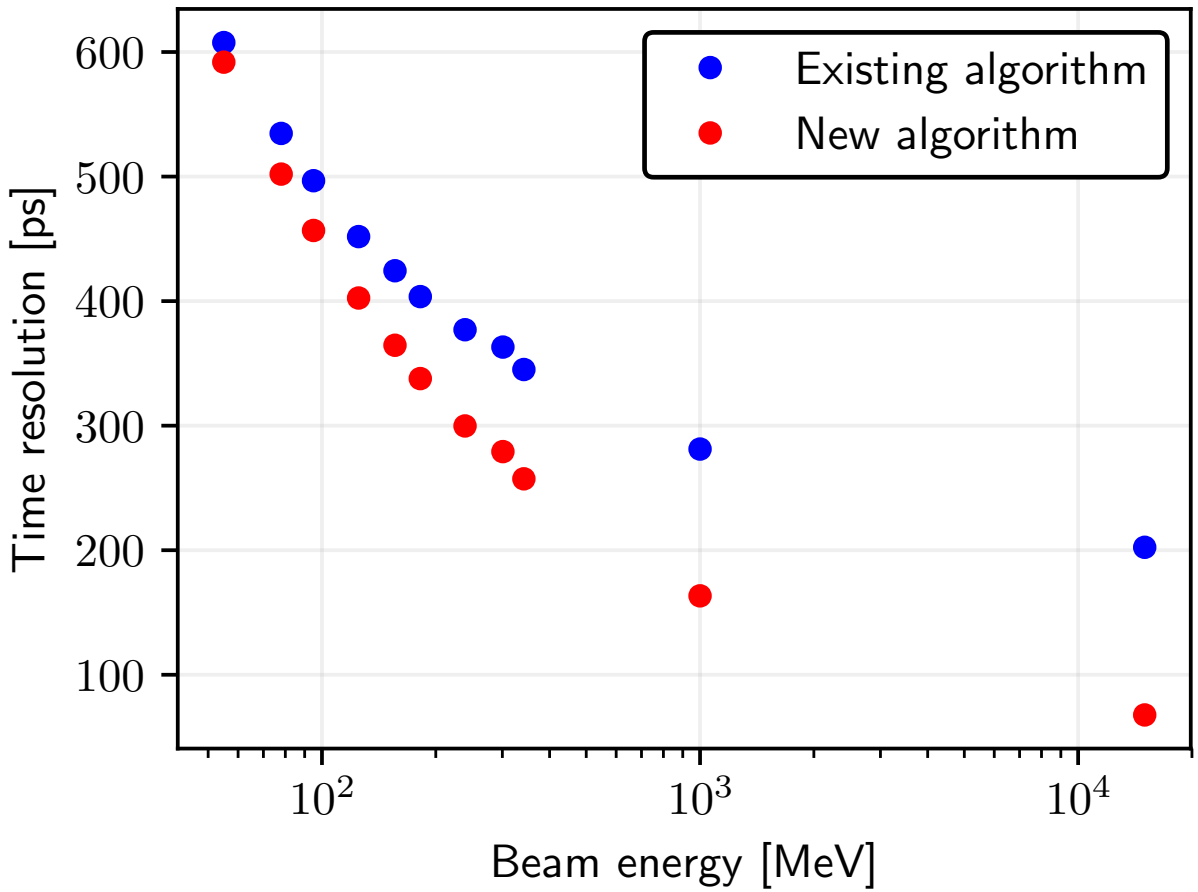
Results — energy resolution



Slight improvement using new algorithm
More samples taken into account

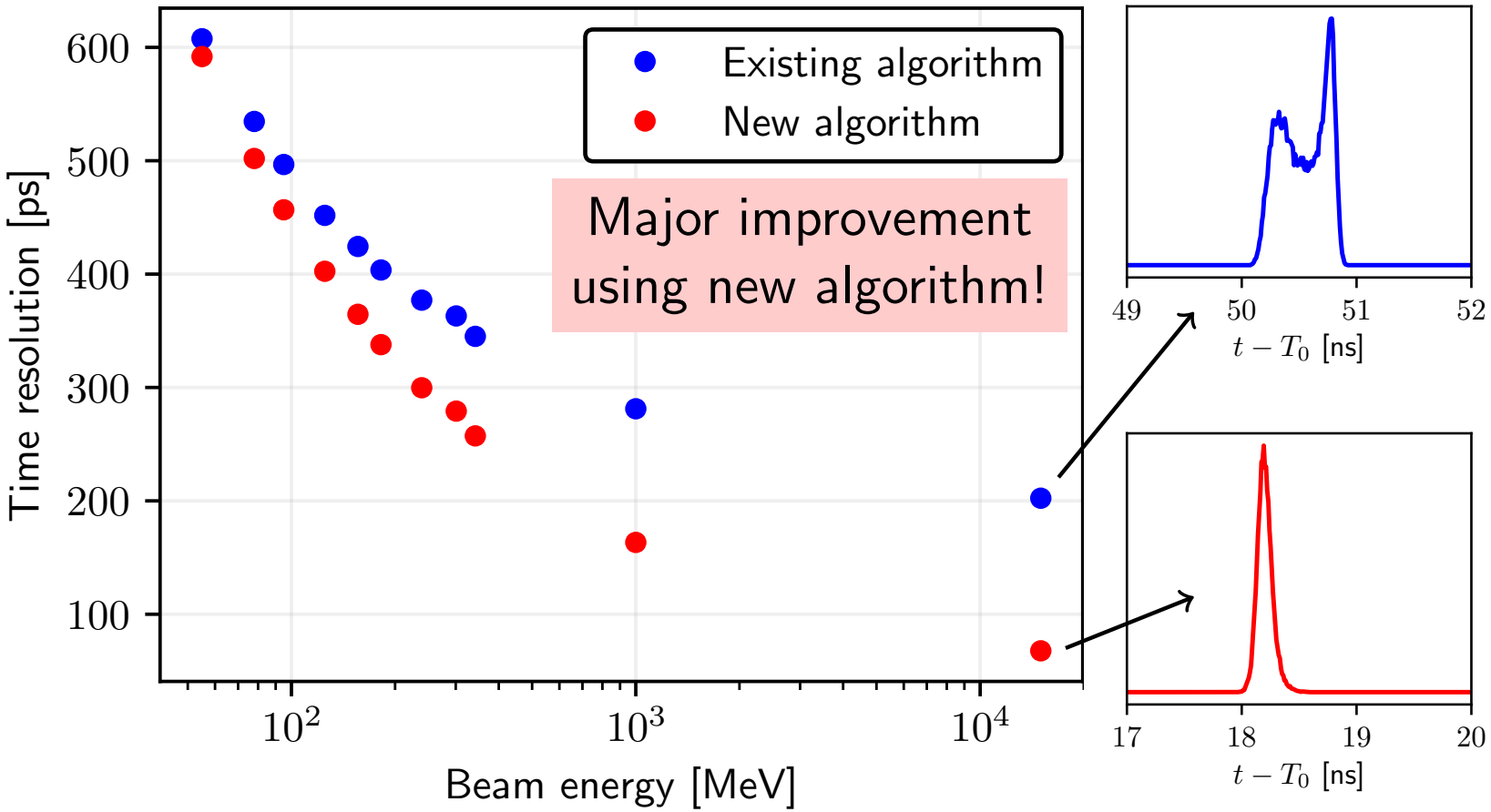
Relative energy resolution of single cell determined by fitting Novosibirsk distribution to pulse-height (amplitude A) spectra.

Results — time resolution



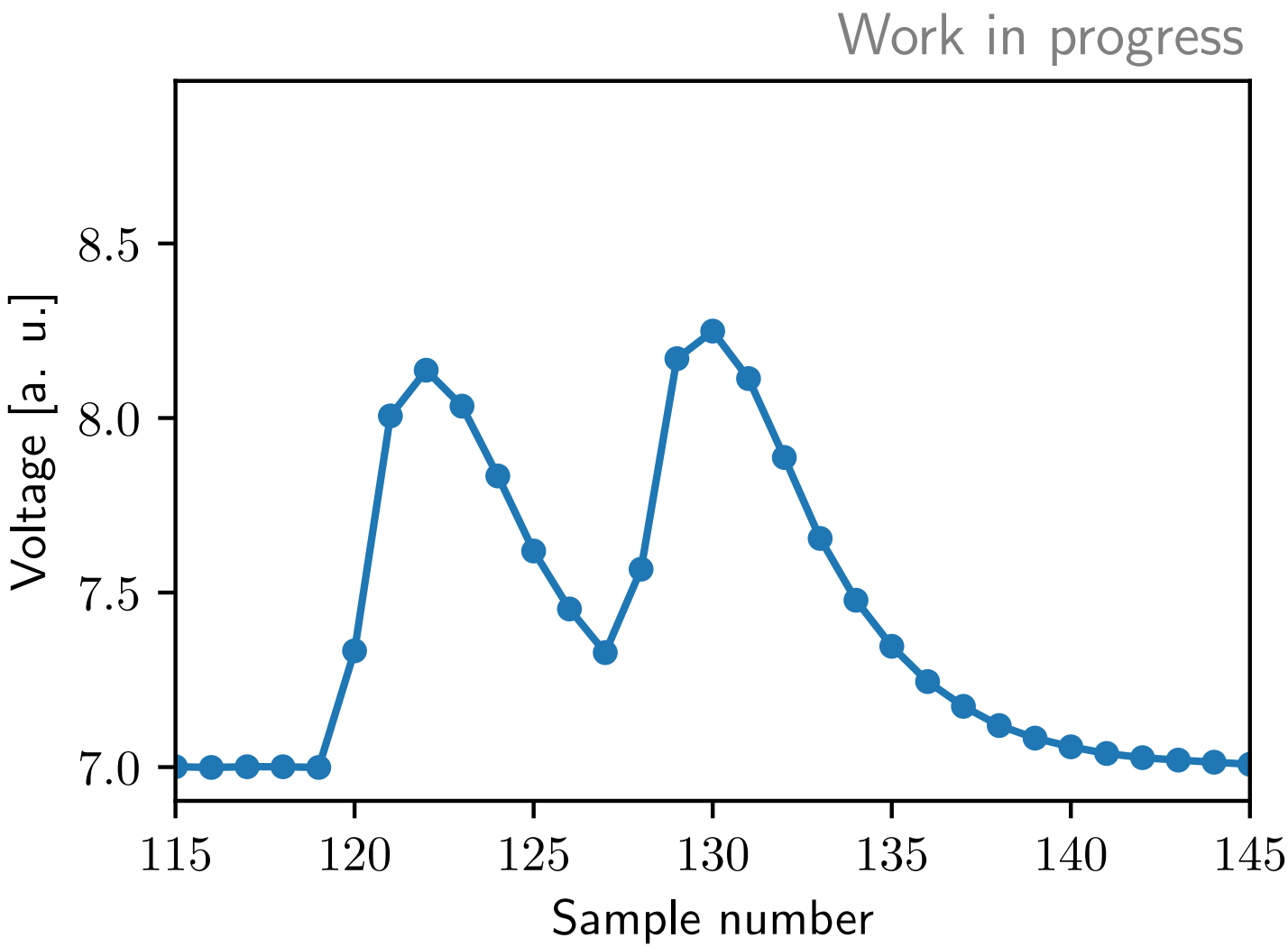
Time resolution determined from $\sigma(t - T_0)$.
T_0: time when photon hits front of detector.

Results — time resolution

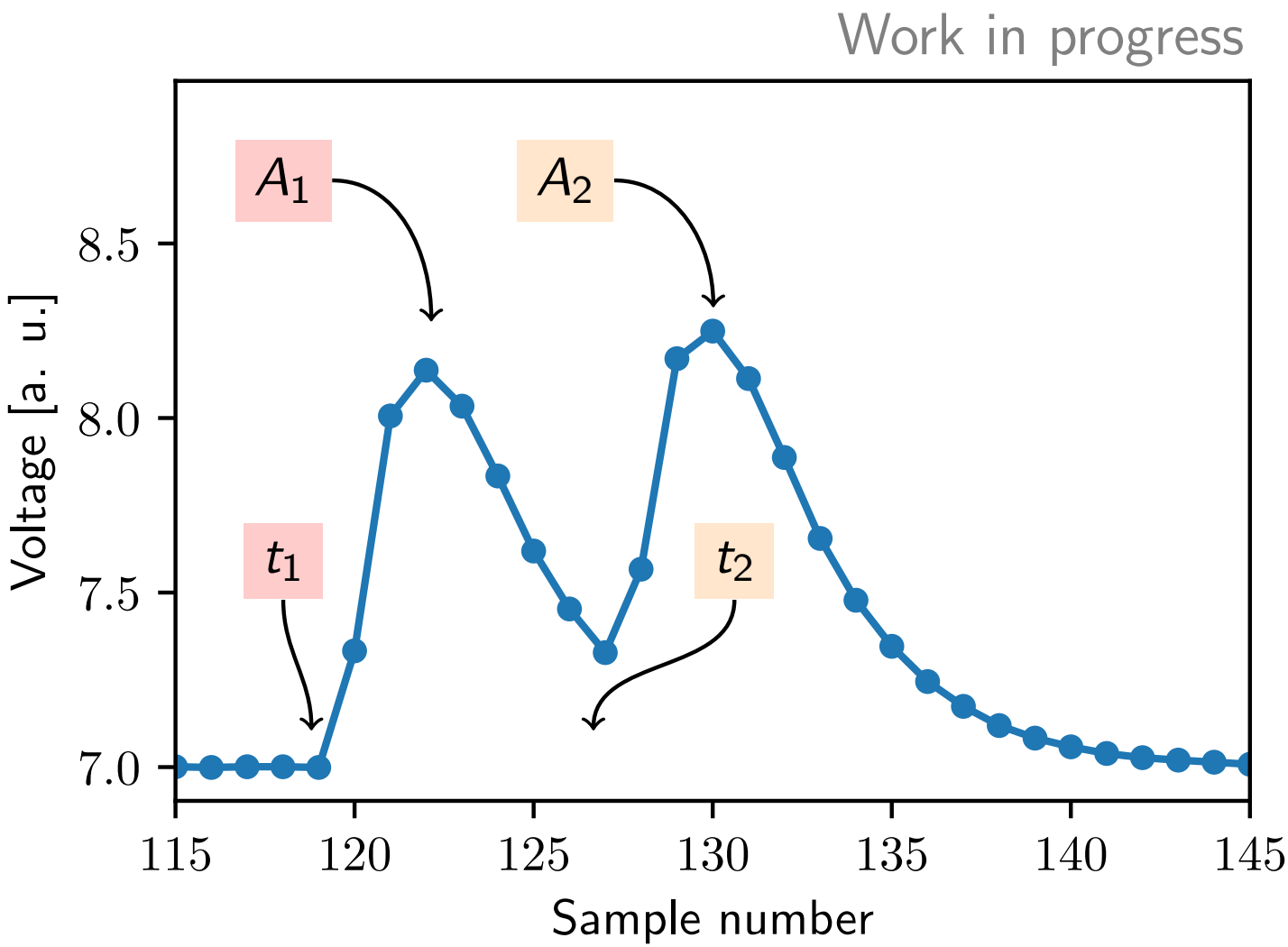


Time resolution determined from $\sigma(t - T_0)$.
T_0: time when photon hits front of detector.

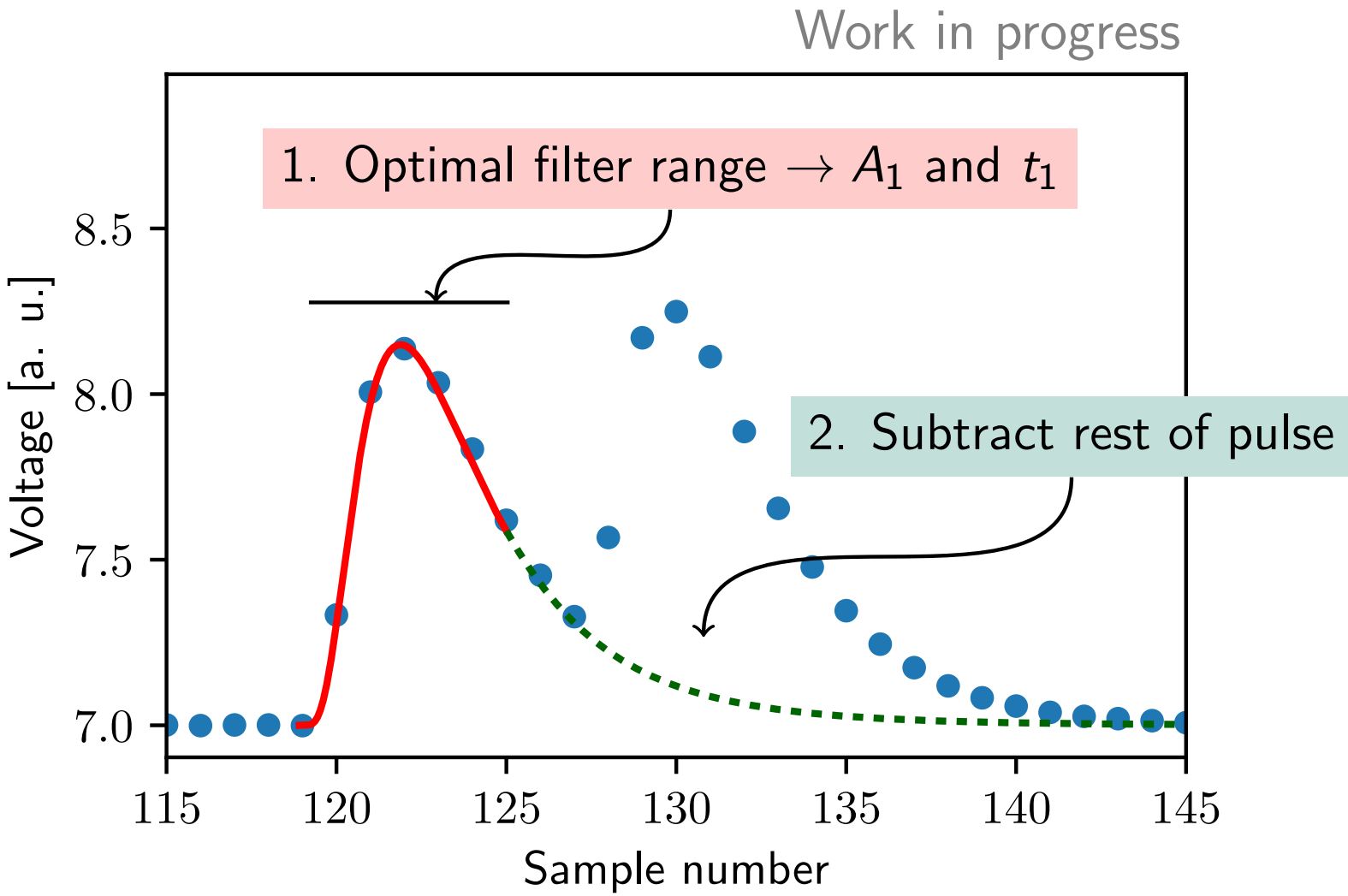
What about pileup reconstruction?



What about pileup reconstruction?



Pileup reconstruction



Summary & Outlook

- Shashlyk Monte-Carlo model has been developed and validated

Summary & Outlook

- Shashlyk Monte-Carlo model has been developed and validated
- A new method for FPGA-based feature extraction has been developed
- Based on method of *optimal filtering*

Summary & Outlook

- Shashlyk Monte-Carlo model has been developed and validated
- A new method for FPGA-based feature extraction has been developed
- Based on method of *optimal filtering*
- Slight improvement in energy resolution
- Big improvement in time resolution (relative to linear Constant Fraction)

Summary & Outlook

- Shashlyk Monte-Carlo model has been developed and validated
- A new method for FPGA-based feature extraction has been developed
- Based on method of *optimal filtering*
- Slight improvement in energy resolution
- Big improvement in time resolution (relative to linear Constant Fraction)
- Method can be used to reconstruct pile-up events
- Optimisations needed for final shashlyk digitiser hardware (e.g. different shaping time and sampling rate)

Thank you for your attention!