Development & performance studies of PicoCal timing simulation for LHCb Upgrade II

Zifan Li (Peking University)

(on behalf of the LHCb ECAL U-II group)

CLHCP2026, Wuhan, 2026/10/31

CONTENTS

I. The LHCb ECAL in Upgrade II

II. Development of PicoCal Timing Simulation

III. Validation and Performance Studies

IV. Summary

The LHCb ECAL in Upgrade II

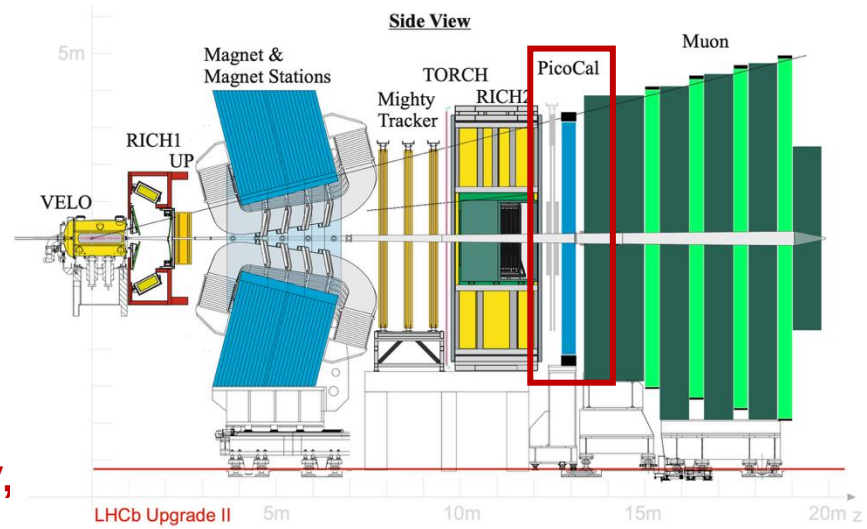
LHCb ECAL in Upgrade II

The LHCb ECAL in Upgrade II (PicoCal) will…

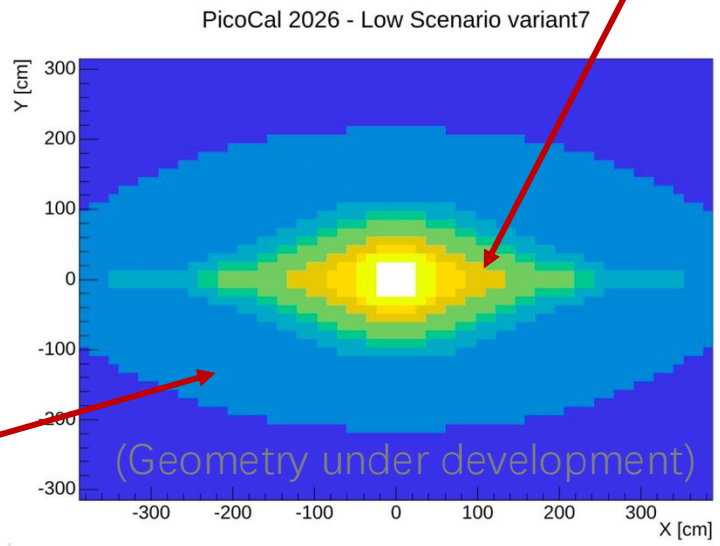
- **be capable of time measurement**
- utilize Spaghetti Calorimeter (SpaCal) technology
- have higher spatial granularity

… to adapt the significant higher …

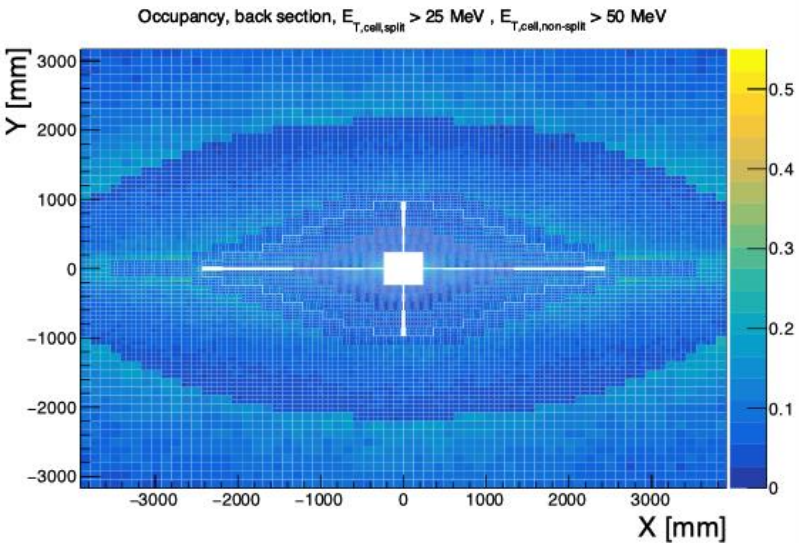
- luminosity
- radiation dose
- occupancy
- pile-up



Inner regions:
SpaCal modules
(W+GAGG, W+Poly,
Pb+Poly etc.)



Outer regions:
Shashlik modules



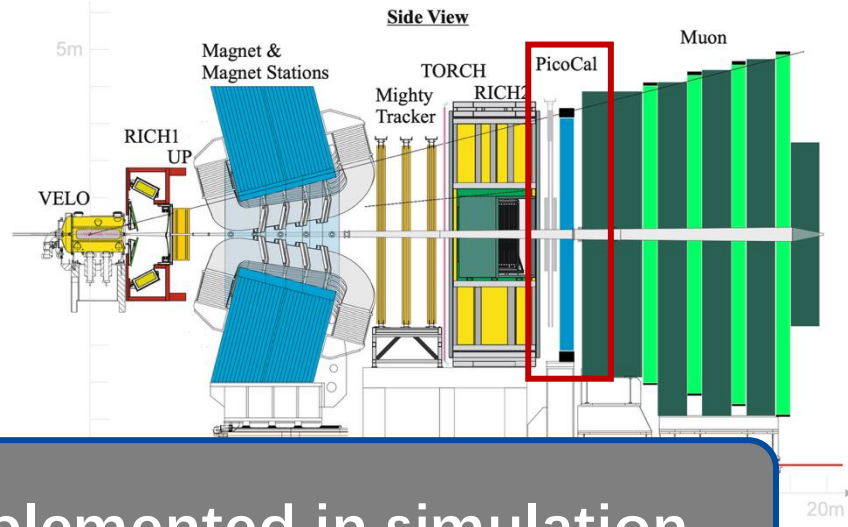
LHCb ECAL in Upgrade II

The LHCb ECAL in Upgrade II (PicoCal) will…

- **be capable of time measurement**
- utilize Spaghetti Calorimeter (SpaCal) technology
- have higher spatial granularity

… to adapt the significant higher …

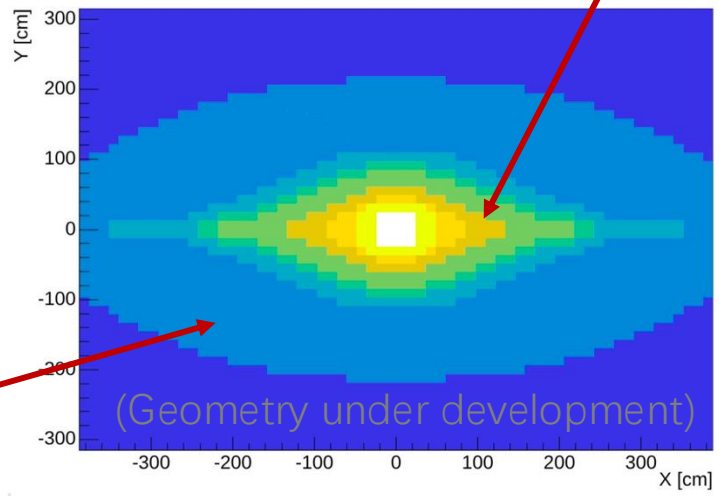
- radiation dose
- occupancy
- pile-up

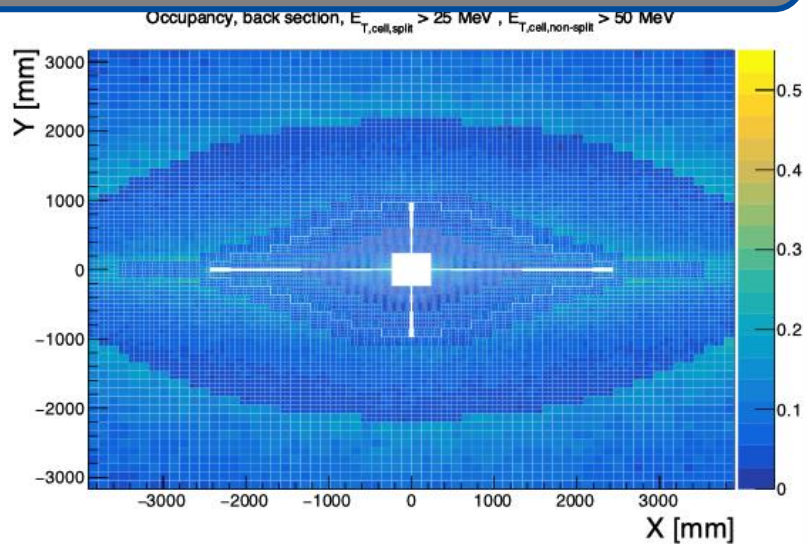


Inner regions:
SpaCal modules

The timing ability of ECAL is a new feature to be implemented in simulation

Outer regions:
Shashlik modules





Target of Timing Simulation

The target of PicoCal timing simulation is to **implement ECAL timing ability in the simulation framework**, including …

- generate timing response for each PicoCal cell hit (Digitization)
- use cell-level time info for further processing (Reconstruction etc.)

… while …

- providing a good description of PicoCal timing capability (e.g. time resolution)
- balancing computing resource usage and granularity of physical process description

Target of Timing Simulation

The target of PicoCal timing simulation is to **implement ECAL timing ability in the simulation framework**, including …

- generate timing response for each PicoCal cell hit (Digitization)
- use cell-level time info for further processing (Reconstruction etc.)

… while …

- providing a good description of PicoCal timing capability (e.g. time resolution)
- balancing computing resource usage and granularity of physical process description

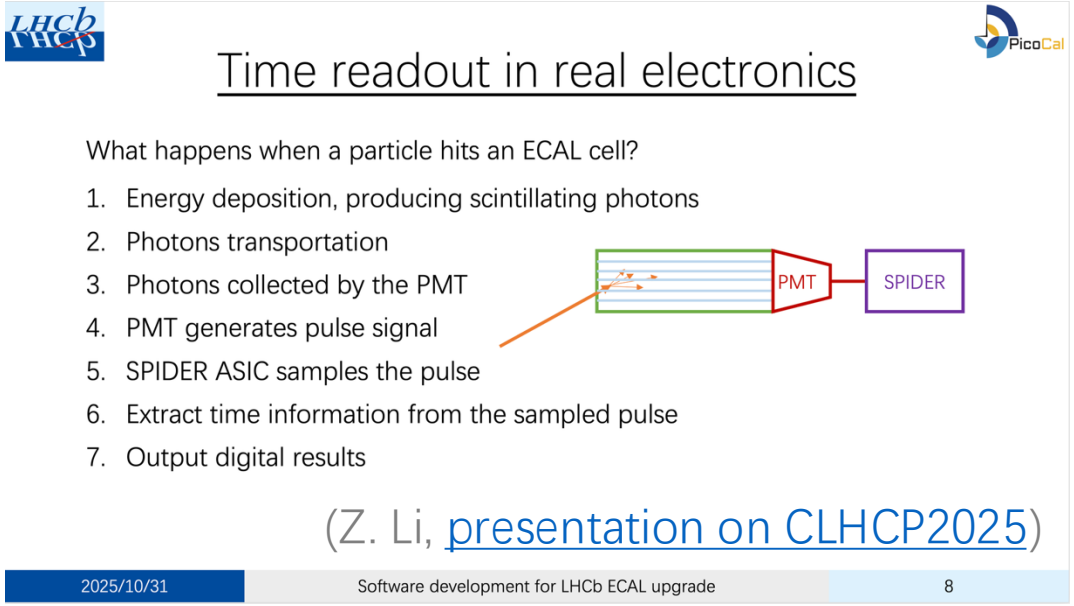
This presentation will focus on the recent progress on digitization process

Development of PicoCal Timing Simulation

Simplify the Simulation

Particle detection in an optical ECAL is consisted of many physical process …

- **Full simulation** takes unacceptable time
  - HybridMC (see backup) is a large progress, but not enough for mass production
- **Time smearing** is fast (no simulation), but …
  - Not suitable for studying other effects, e.g. pile-up and spillover in one cell



Simulation we want: suitable for studying pile-up effect etc., and fast

Full, ray-tracing simulation (no simplification)

?

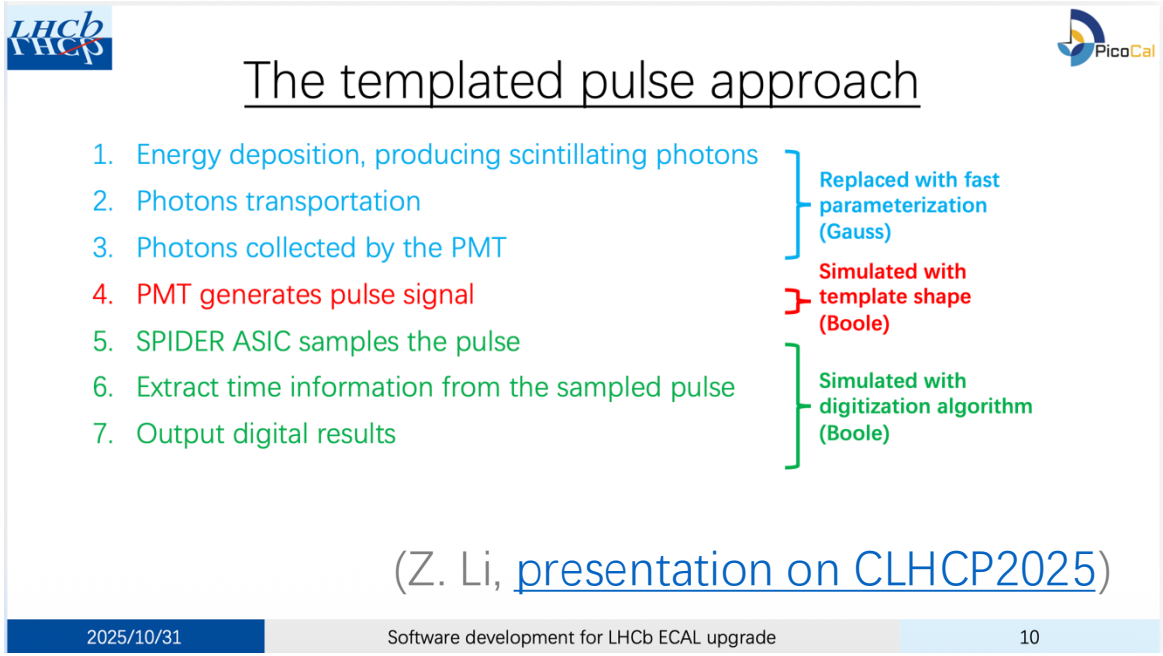
Time smearing (most simplified)

Simplicity

A Templated Pulse Approach

The templated pulse approach:

- No ray-tracing
- Calculate timestamp from simulated PMT pulses (instead of simple smearing)
- PMT pulses generated with parameterized template functions
- Template parameter distributions are extracted from reference datasets, reflecting the characteristic of ECAL module and PMT



Full, ray-tracing simulation (no simplification)

Templated pulse generation (simplify optical transportation)

Time smearing (most simplified)

Simplicity

Prog. 1: Re-designing the Template Function

Template function should provide good description to the actual pulse shape

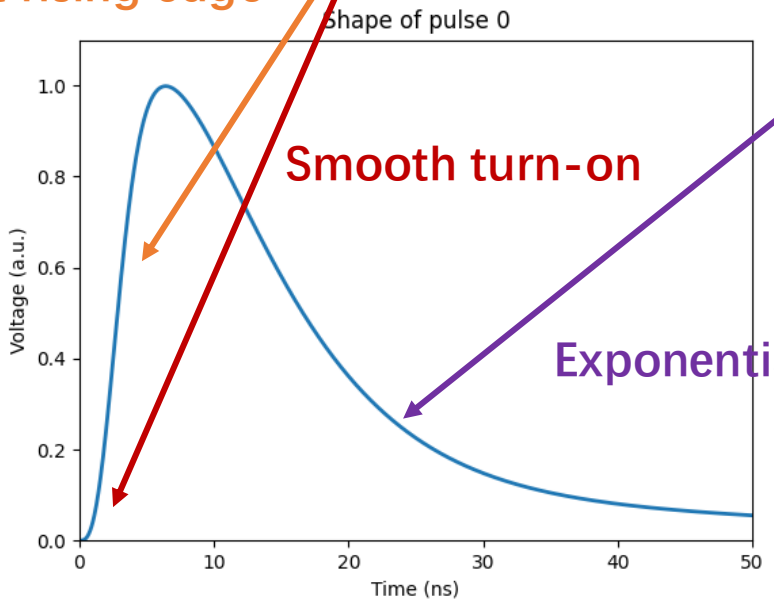
The following function was used; it was fine

$$F(t; A, t_0, \tau_0, f_i, \tau_i) = A \cdot (t - t_0) \cdot \left[1 - e^{-((t-t_0)/\tau_0)^2}\right] \cdot \sum_{i=1}^{2} f_i \, e^{-(t-t_0)/\tau_i} \quad \left(\sum_{i=1}^{2} f_i = 1\right)$$

Fast rising edge

Smooth turn-on

Exponential-like tail



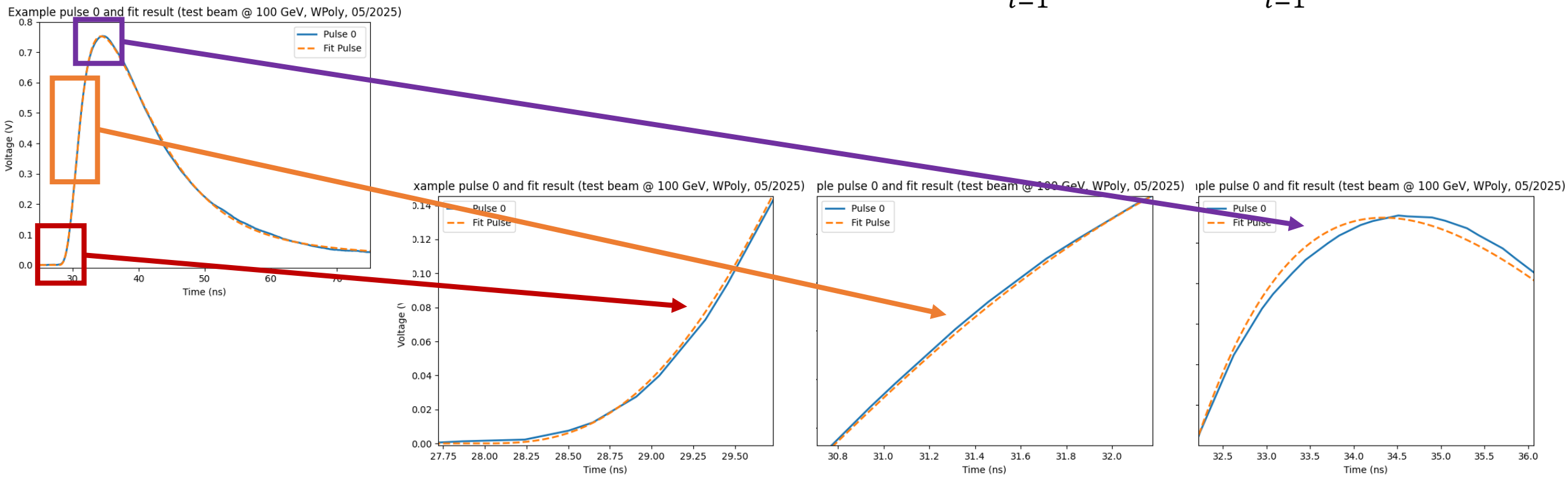
(Z. Li, presentation on CLHCP2025)

Prog. 1: Re-designing the Template Function

Template function should provide good description to the actual pulse shape

The following function was used; it was fine ⋯ **but not perfect!**

$$F(t; A, t_0, \tau_0, f_i, \tau_i) = A \cdot (t - t_0) \cdot \left[1 - e^{-((t-t_0)/\tau_0)^2}\right] \cdot \sum_{i=1}^{2} f_i \, e^{-(t-t_0)/\tau_i} \quad \left(\sum_{i=1}^{2} f_i = 1\right)$$

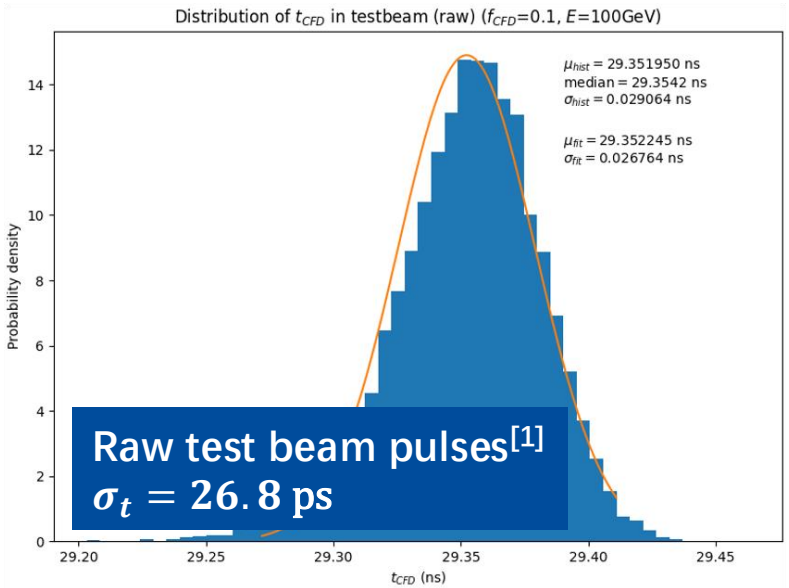


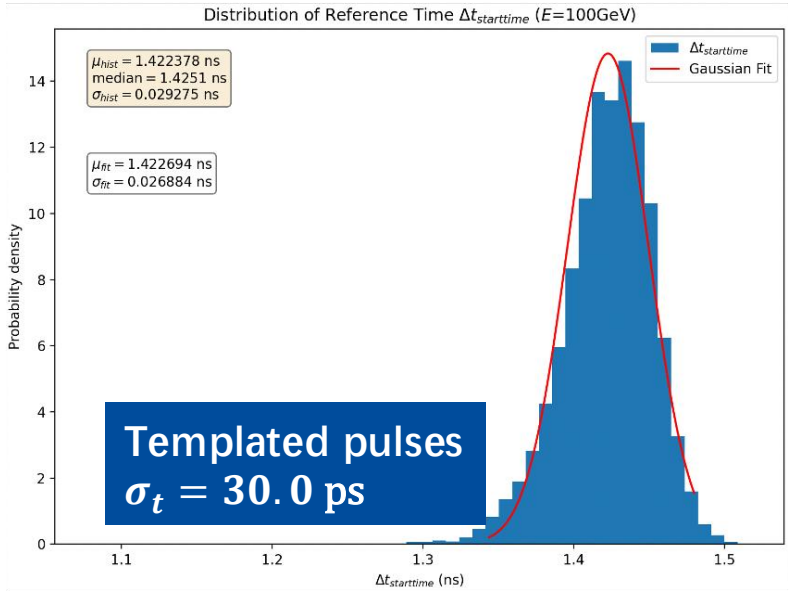
Prog. 1: Re-designing the Template Function

Template function should provide good description to the actual pulse shape

The following function was used; it was fine ⋯ **but not perfect!**

$$F(t; A, t_0, \tau_0, f_i, \tau_i) = A \cdot (t - t_0) \cdot \left[1 - e^{-((t-t_0)/\tau_0)^2}\right] \cdot \sum_{i=1}^{2} f_i \, e^{-(t-t_0)/\tau_i} \quad \left(\sum_{i=1}^{2} f_i = 1\right)$$





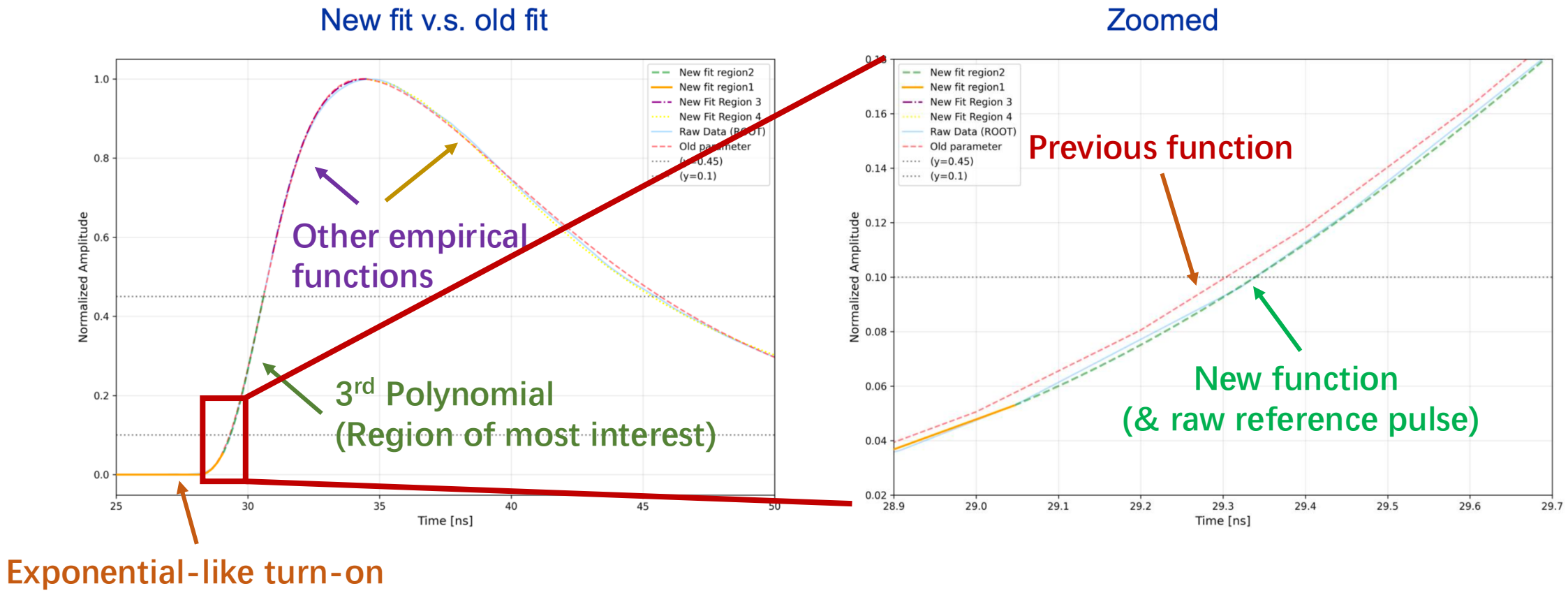
Time resolution discrepancy between raw pulses and template functions fitted to them

(M. Zhao et al., 2026)

[1]: W+GAGG SpaCal, electron, 100GeV

Prog. 1: Re-designing the Template Function

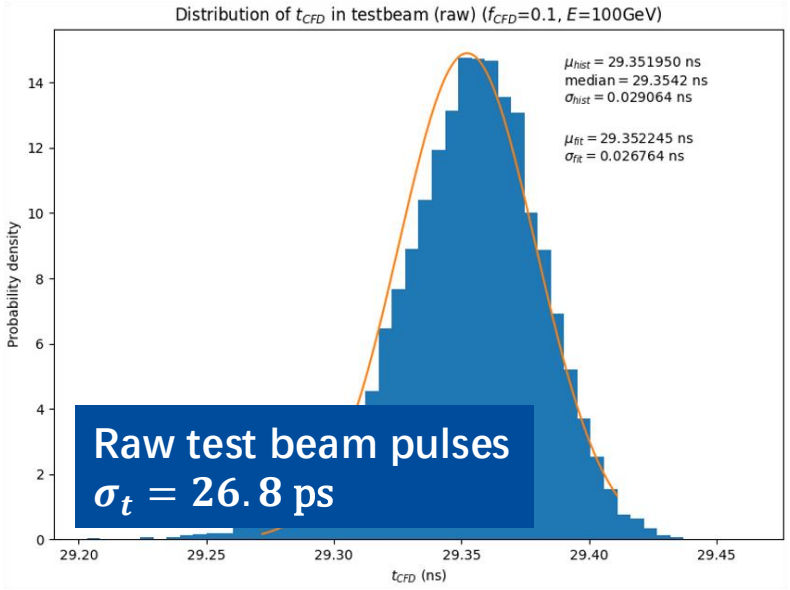
The description is improved with a new, segmented function

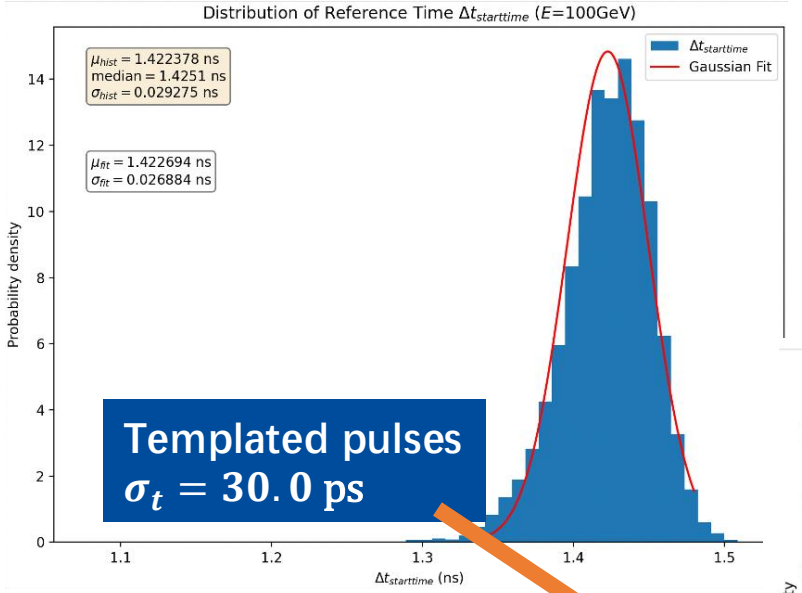


(M. Zhao et al., 2026)

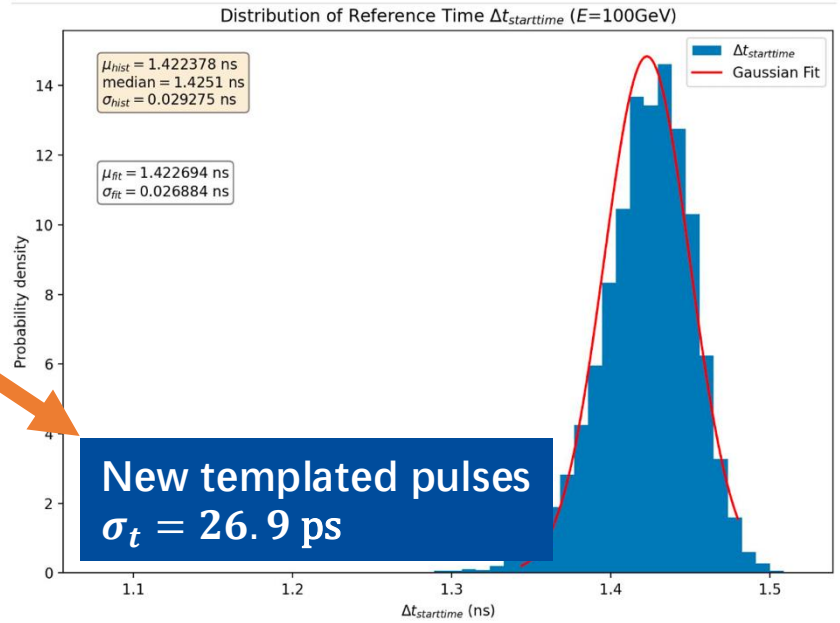
Prog. 1: Re-designing the Template Function

The description is improved with a new, segmented function





(M. Zhao et al., 2026)



Raw reference pulses and template pulses
now have the same time resolution

Prog. 2: Handling Different Modules/Particles

In the early R&D phase, parameter distributions were extracted from only **one reference dataset** (W+GAGG SpaCal, electron @ 100GeV)

These distributions were **applied to all module and particle types** in simulation

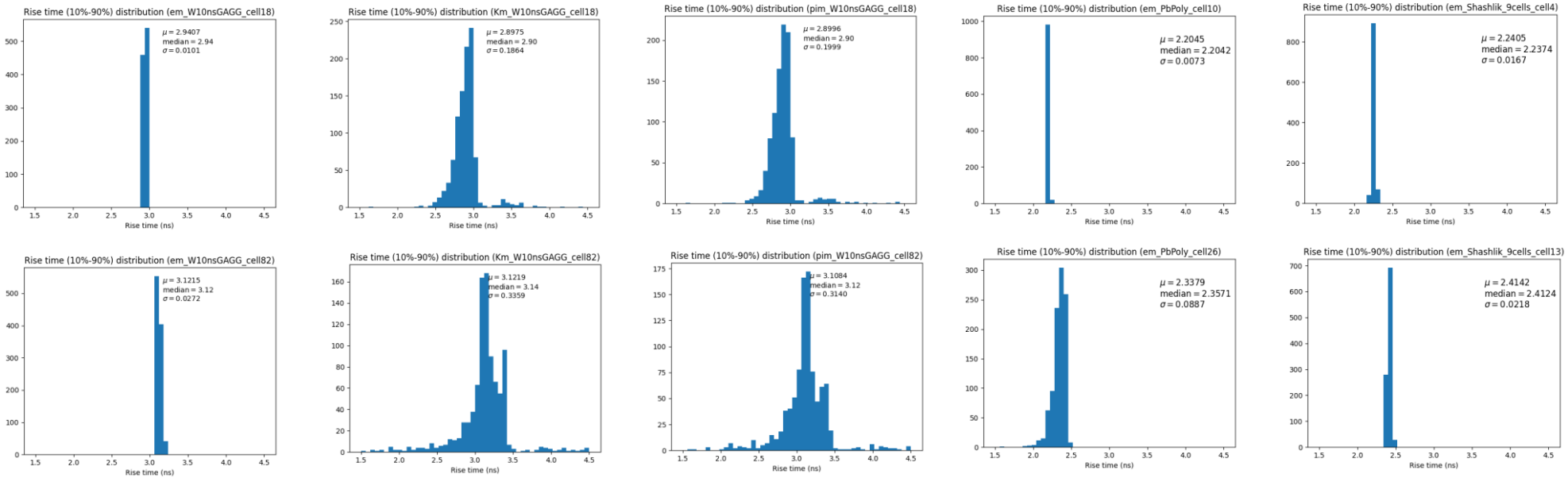
Prog. 2: Handling Different Modules/Particles

In the early R&D phase, parameter distributions were extracted from only one reference dataset (W+GAGG SpaCal, electron @ 100GeV)

These distributions were applied to all module and particle types in simulation

Studies showed this not accurate!

Rise time distribution, E=100GeV, single module



Different combinations have different pulse characteristic, e.g. rise time

(Z. Li et al., 2025)

Prog. 2: Handling Different Modules/Particles

In the early R&D phase, parameter distributions were extracted from only one reference dataset (W+GAGG SpaCal, electron @ 100GeV)

These distributions were applied to all module and particle types in simulation

Studies showed this not accurate!

Currently, **parameter distributions for different module / particle types** are extracted from corresponding reference datasets, then **applied correspondingly during pulse generation**

Prog. 3: Runtime Parameter Generation

Pulse generation ≈ generate a set of template function parameters

At first, parameters were **drawn from a template pool**, which comes from fitting template function to reference datasets

This implies …

- Large template pools (therefore reference datasets), for statistical independence
- No ‘interpolation’ between pools (will explain later)

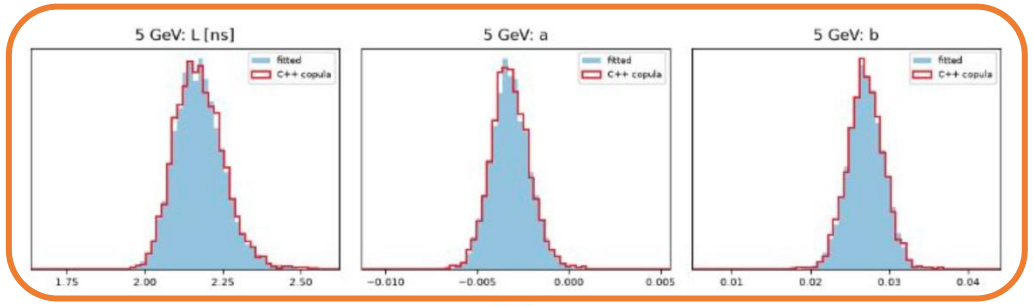
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	A_fit	t0_fit	tau0_fit	a	b	c	d	A3	t3	tau3a	tau3b	A4	t4	tau4a	tau4b	tb12	tb23	tb34
2	0.679188746	28.18332961	9.988767356	-0.0293919	2.691684931	-81.848989	826.8000956	0.433682579	28.07304818	2.580581064	6.282247810	15.2395409	27.59248736	7.394074913	27.59447832	29.04644	30.940094	34.51065
3	0.686467972	28.22475710	9.988222873	-0.0308808	2.837440495	-86.571173	877.4726032	0.450366540	28.24704283	2.390857171	6.04470175	15.78482696	27.34104651	7.589032890	27.32389249	29.19484	30.808472	34.398598
4	0.733643819	28.16949085	9.987009062	-0.0288902	2.637269151	-79.950137	805.2676513	0.380524210	27.85764321	2.755339611	7.151232906	12.42652136	28.89598036	6.388219709	30.81476784	29.122528	30.87014	35.106613
5	0.673354900	28.20938412	9.988684491	-0.0349569	3.195952622	-97.075003	979.9890853	0.412178832	28.04222268	2.568371363	6.602931010	14.35706501	28.06177732	7.071406541	28.06457989	29.135834	31.029716	34.74446
6	0.812595709	28.27016945	9.98807731	-0.0395747	3.611603543	-109.53825	1104.489499	0.413748157	27.99256764	2.588008685	6.579061923	14.37177871	27.93675808	7.114052655	28.83684258	29.342133	30.971802	34.682037

Template pool
(each line = a template = a set of parameters)

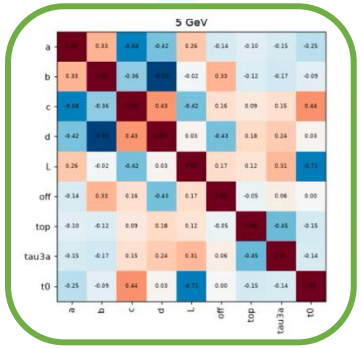
Prog. 3: Runtime Parameter Generation

Now, parameters are **generated by runtime sampling**

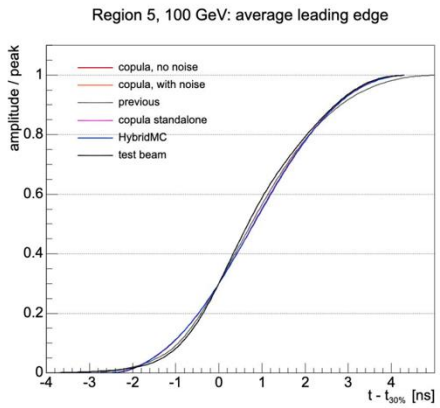
Gaussian ‘kernal’ sampling



‘Correlations’



(E. Caire et al., 2026)



Generated template

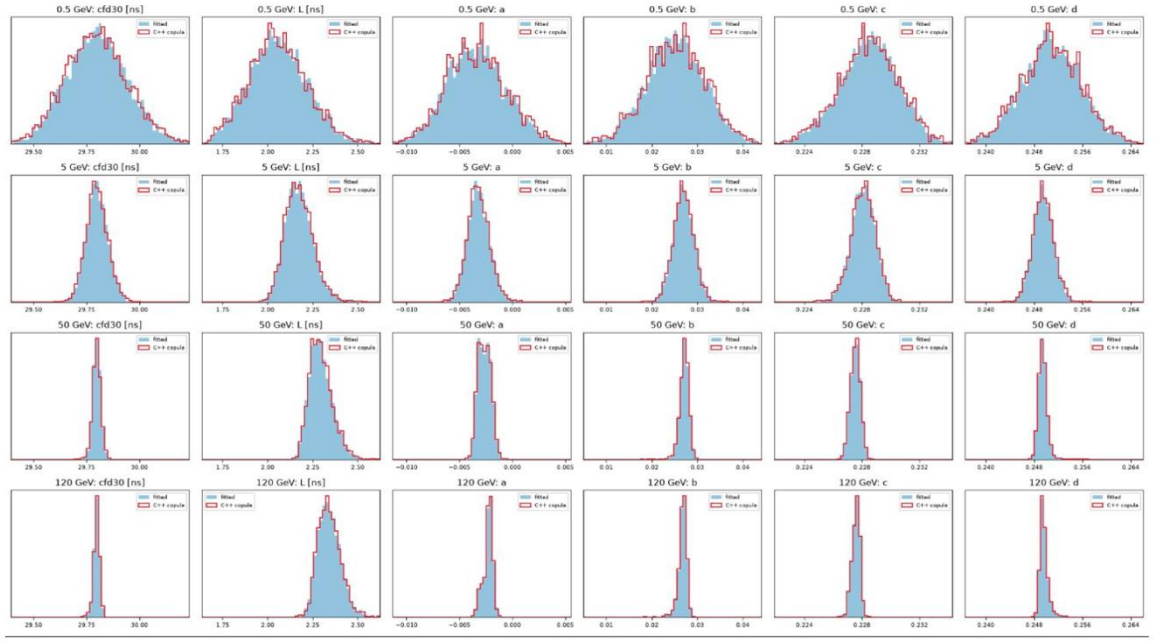
Runtime-resampling with Gaussian Copula method
(Implemented in Boole !887)

Prog. 4: Energy-dependent Templates

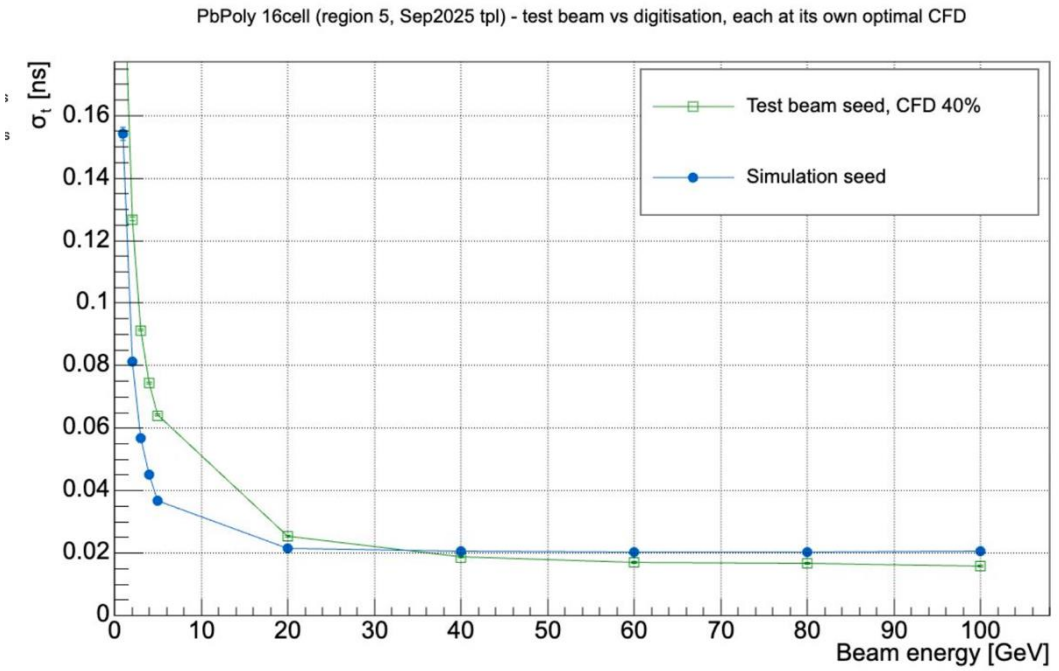
Parameter distribution is affected by particle energy (due to photo-statistics etc.)

Template pool at 1 energy point (100GeV) is not enough

E



(←Template Parameters→)



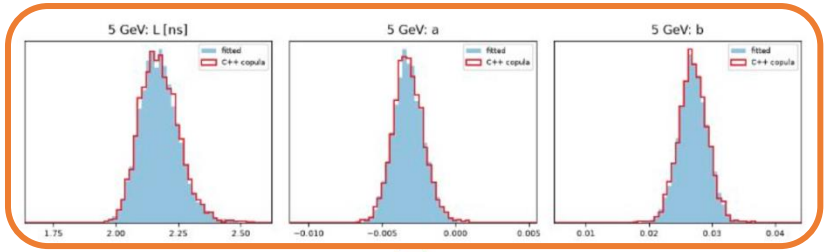
(E. Caire et al., 2026)

Prog. 4: Energy-dependent Templates

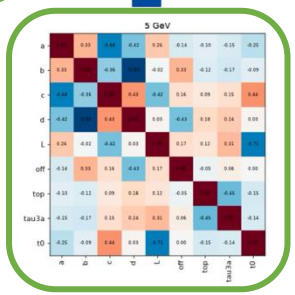
Energy-dependent distributions are now extracted and used in pulse generation

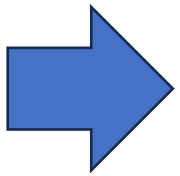
The runtime sampling method also enables **'interpolation' of distributions** between energy points

Gaussian 'kernal' sampling



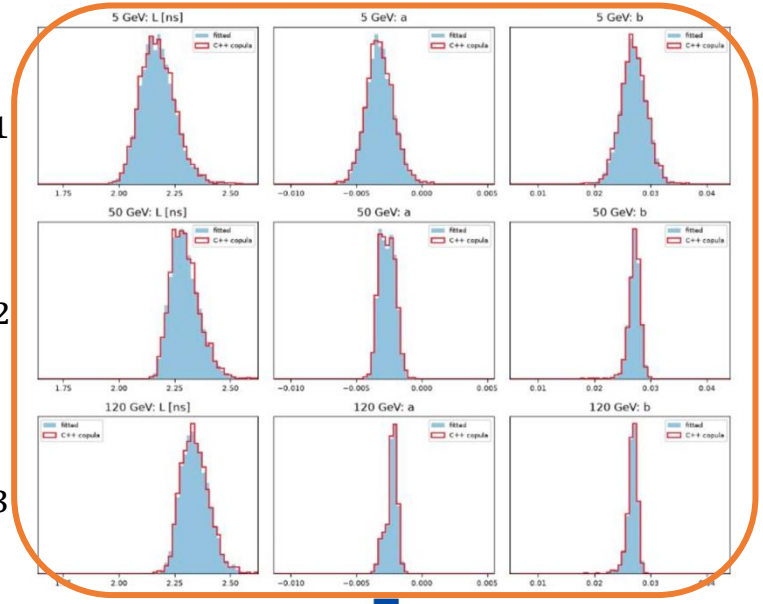
'Correlations'



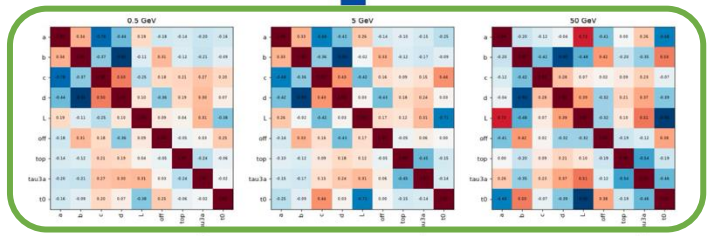


Gaussian 'kernal' sampling

E_1
E_2
E_3



'Correlations'

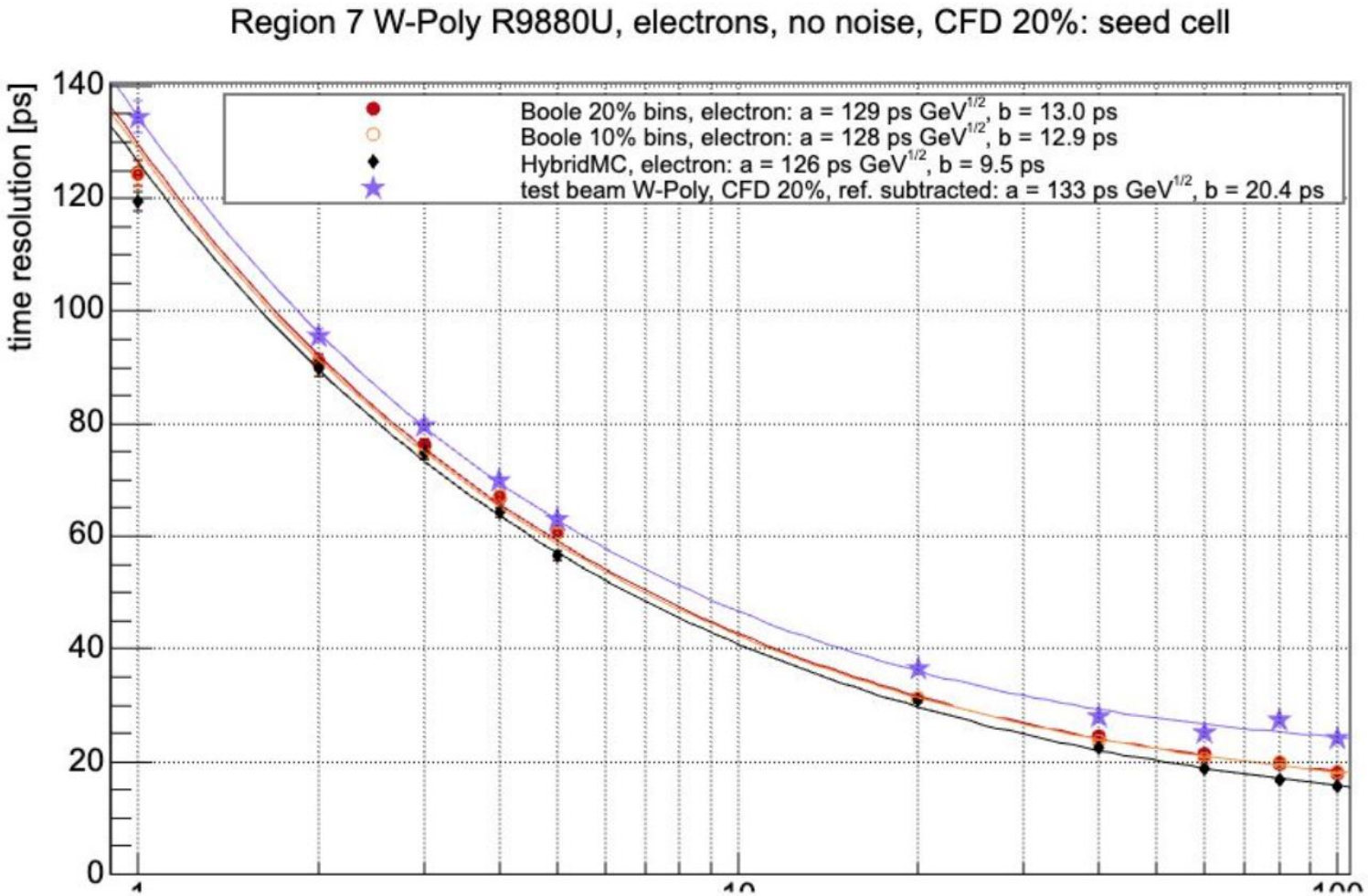


E_1 E_2 E_3

Validation and Performance Studies

Resolution – ParticleGun, No Noise

An example: $\sigma_t(E)$ curve of electrons[1] in W+Poly SpaCal modules

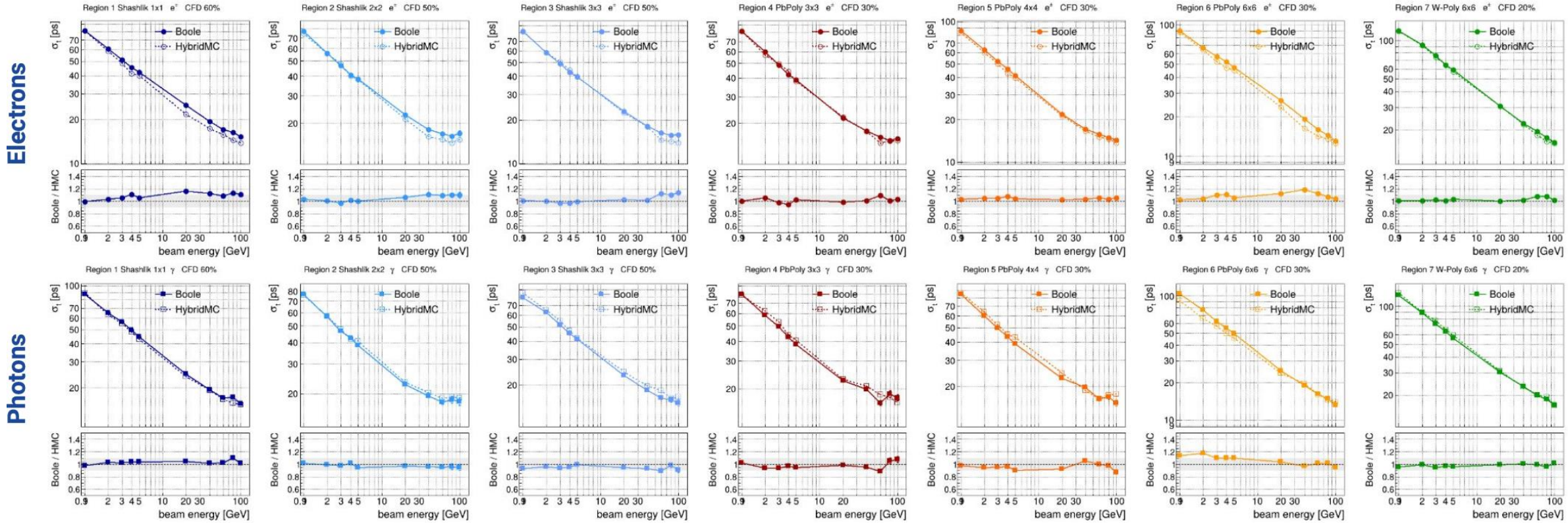


(E. Caire et al., 2026)

[1]: Seed cell only

Resolution – ParticleGun, No Noise

Good agreements for electrons and photons, across all module type and energies

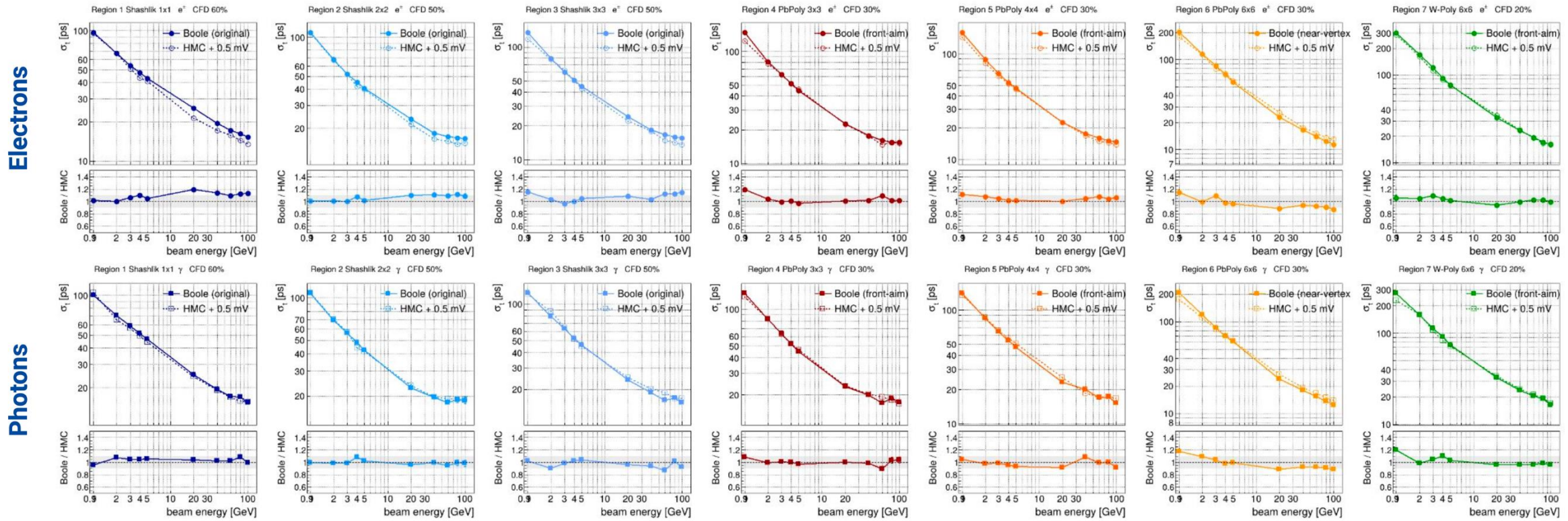


(←ECAL Regions / module types→)

(E. Caire et al., 2026)

Resolution – ParticleGun, With Noise

Good agreements with electronic noise added



(←ECAL Regions / module types→)

(E. Caire et al., 2026)

Time Consumption

Time per event	Single-row templates	Final copula
Pulse algorithm, 1 GeV	0.18 ms	0.23 ms
Pulse algorithm, 100 GeV	1.03 ms	1.38 ms
Whole digitisation, 1 GeV	1.6 ms	1.9 ms
Whole digitisation, 100 GeV	7.0 ms	7.1 ms

The runtime generation provides a better agreement in time resolution, while costing only ~10% more total digitization time

(E. Caire et al., 2026)

Summary

Summary

Timing simulation for LHCb's PicoCal detector has been actively developed:

- **Continued with the templated pulse approach** established since last year
- **Extensive works are done to improve the description**, e.g. re-designing template function, add module/particle dependency …
- **New features are implemented** (e.g. runtime parameter generation) to suit the need of future mass production

Various validation and performance studies are made:

- **The ParticleGun simulation shows good agreement** with reference datasets
- (Validation with physics channels is also ongoing, see Yuhang's presentation later this session)

The timing simulation for LHCb Upgrade II TDR study is nearly converged

- Other efforts to improve it for future study are ongoing

THANK YOU FOR LISTENING!

BACKUPS

Two simulation systems

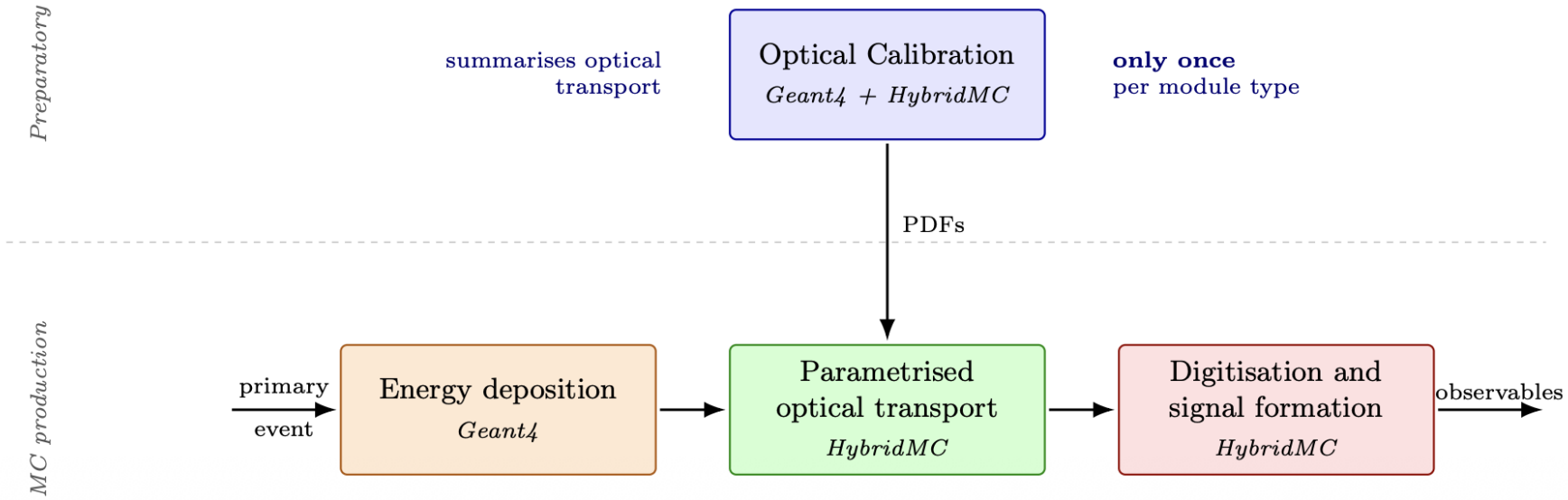
2 independent simulation systems are involved in the PicoCal R&D:

- **The LHCb software stack**
  - The official simulation framework to be used for future physics analysis
  - Aims for centralized, mass-production
  - **Needs 'usable' throughput, while maintaining good agreements**
- **HybridMC**
  - A framework for fast detailed simulation of optical calorimeter
  - Aims for fast reference sample production in R&D process
  - **Prioritize precision, while being much faster than full, ray-tracing simulation**

The concept of HybridMC

The idea: factorise out the optical transport

1. **Once per module type**: a full ray-tracing *optical calibration* encodes the light-transport physics into probability density functions (PDFs)
2. **Every event**: Geant4 produces the energy-deposition map with *optical photons switched off*; a parametrised *summary Monte Carlo* step then draws the detected photons from the PDFs



Timing simulation - reconstruction

For the time being, the reconstruction simply uses the **timestamp of a cluster's seed cell** as the timestamp of the cluster

Various efforts are ongoing to further utilize time information, including using Machine Learning technics

Better Reference Dataset Coverage

Pulses from test beam experiments were used as reference datasets

- **Pros:** Realistic, reflects the concurrent design
- **Cons:**
  - Can't generate on demand
  - No datasets for hadrons (e.g. charged pions)
  - Dataset granularity limited by test beam schedule

Currently, HybridMC samples are used as reference datasets instead

New Template Function

$$f(t) = \begin{cases} A_1 \left(e^{\frac{t-t_0}{\tau_0}} - 1 \right) (t - t_0) & \text{if } t_0 \leq t < tb_{1,2} \\ \sum_{n=0}^{3} c_n T_n(\hat{t}) & \text{if } tb_{1,2} \leq t \leq tb_{2,3} \\ A_3(t - t_3) \left(1 - e^{-\left(\frac{t-t_3}{\tau_{3a}}\right)^2} \right) e^{-\frac{t-t_3}{\tau_{3b}}} & \text{if } tb_{2,3} < t < tb_{3,4} \\ \frac{A_4}{t-t_4} \left(1 - e^{-\left(\frac{t-t_4}{\tau_{4a}}\right)^2} \right) e^{-\frac{t-t_4}{\tau_{4b}}} & \text{if } t \geq tb_{3,4} \end{cases}$$

Where:

$$\hat{t} = \frac{2(t - t_{\min})}{t_{\max} - t_{\min}} - 1 \quad \text{(Scaled time for Chebyshev)}$$

$T_n(\hat{t})$ = Chebyshev polynomial of the first kind of degree n

c_n = Coefficients corresponding to $[T_0, T_1, T_2, T_3]$

- Parameters:
  - Region1: A1, t0, tau0
  - Region2: c0, c1, c2, c3
    (or named d,c,b,a)
  - Region3: A3, t3, tau3, tau3b
  - Region4, A4, t4, tau4, tau3b
  - Margin: tb12, tb23, tb34

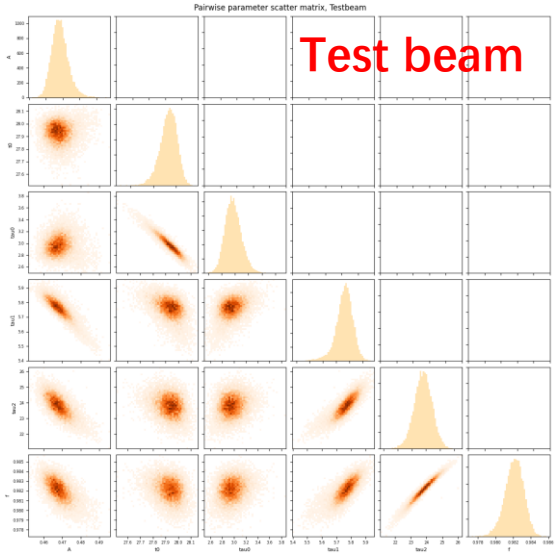
Fitting method

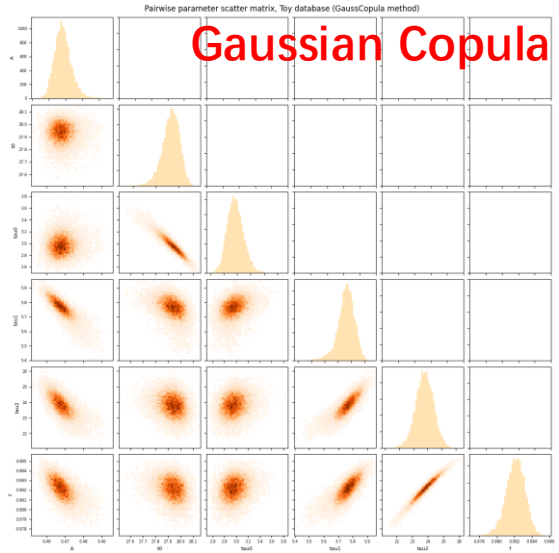
- Constraints: Region 1-2 and Region 3-4 are C^1 connected, Region 2-3 are C^0 connected (avoid over constraint)

1. Fit Region 2 first
2. Fit Region 1 and Region 3 with boundary constraints
3. Fit Region4 with boundary constraints

Gaussian Copula Method

- Conduct a specific transformation T to the parameter sets
  - T regards only the marginal distributions
- Estimate the resulted joint distribution with n-d Gaussian distribution
  - The estimation have same covariant matrix with the joint distribution
- Sample the n-d Gaussian distribution, do the inverse transformation T^{-1}

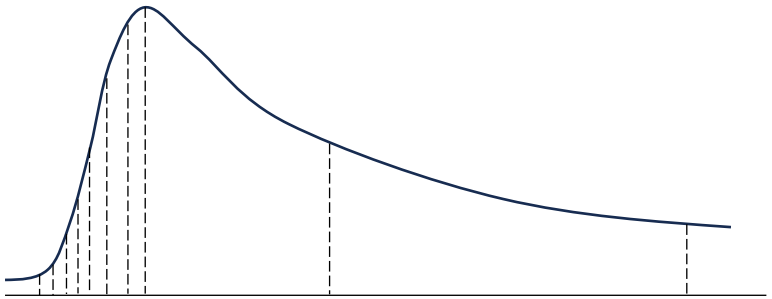




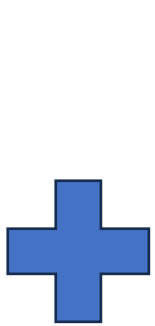
Prog 5: Pulse Generation via Transformation

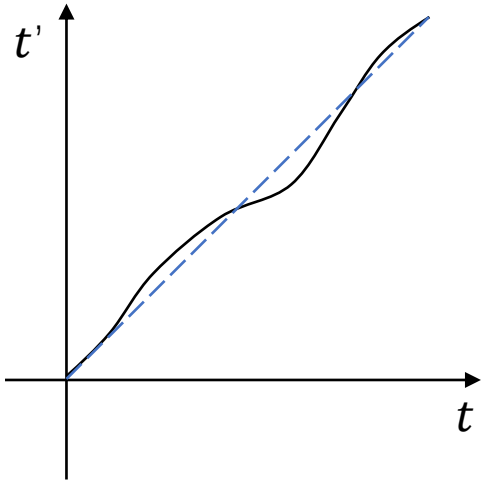
A waveform-transformation method is proposed for the possible …

- Better description of pulse shape variation (therefore, time resolution) w.r.t particle energy
- Faster pulse generation



Existing template: $f(t)$



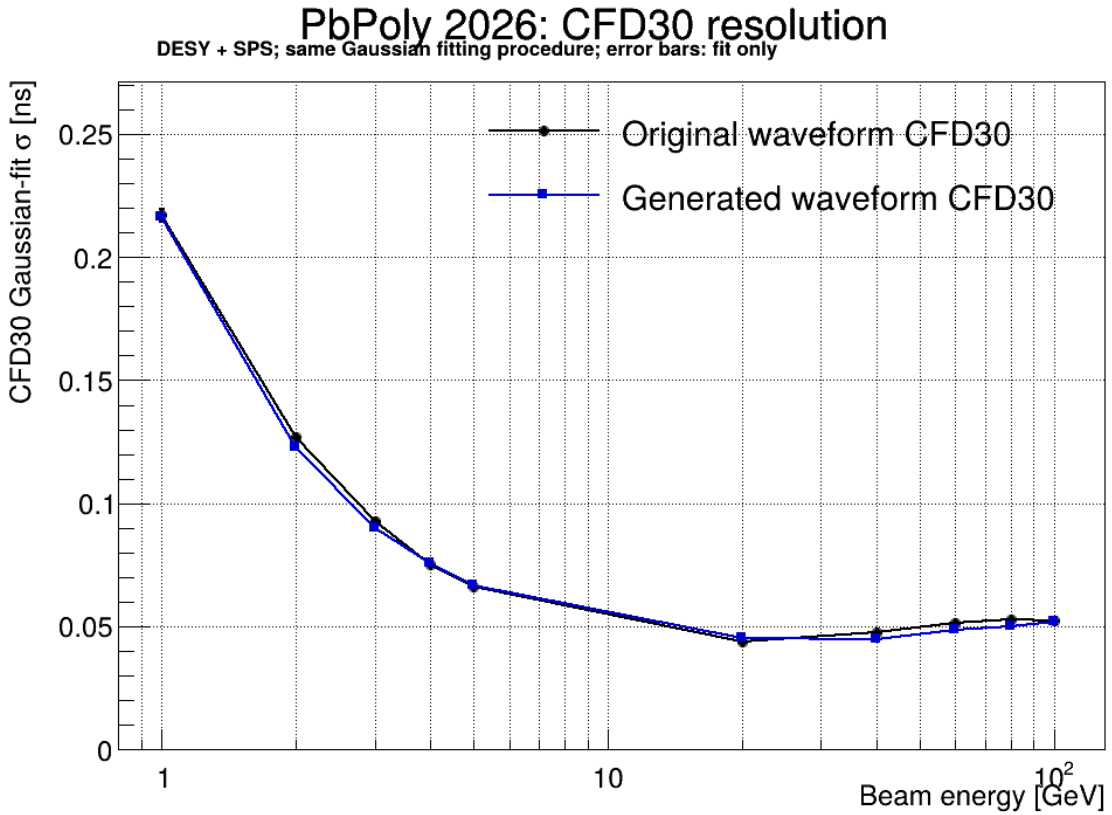


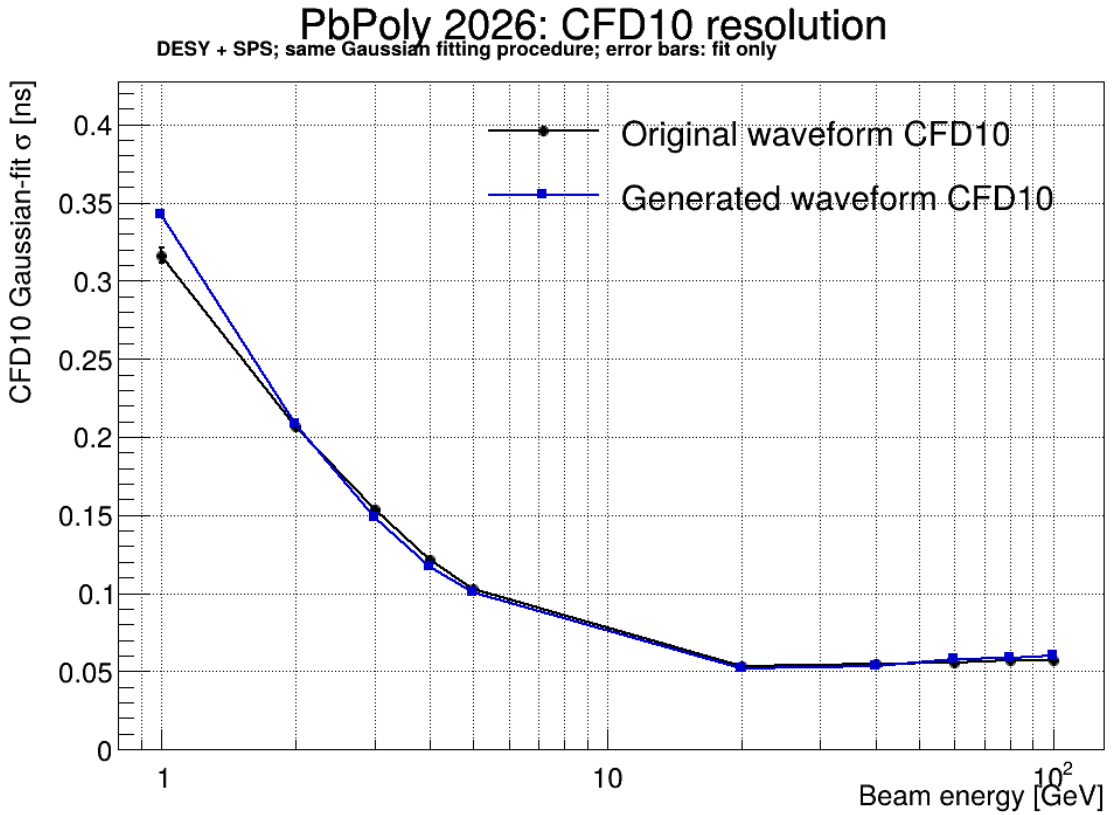
Generated "Nudge": $g(t;E)$

Generated template
$f(g^{-1}(t;E))$

(M. Zhao et al., 2026)

Prog 5: Pulse Generation via Transformation





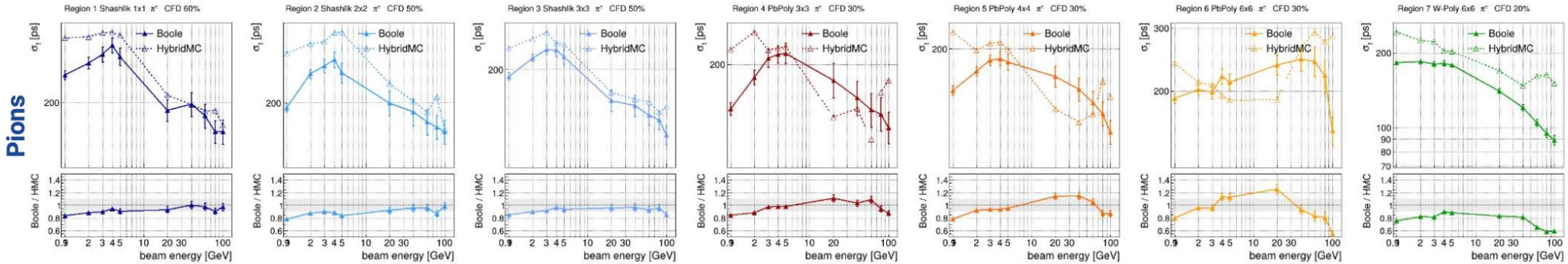
Note 1: Results from standalone scripts, not using Boole
Note 2: Electronic noise is not added (as in Boole) to the generated templates, because TB pulses already contains them

Resolution - Pions

Digitization validation - pions

Using the same principle as the electrons and photons we produce sets of charged pions and sample their waveforms for all regions

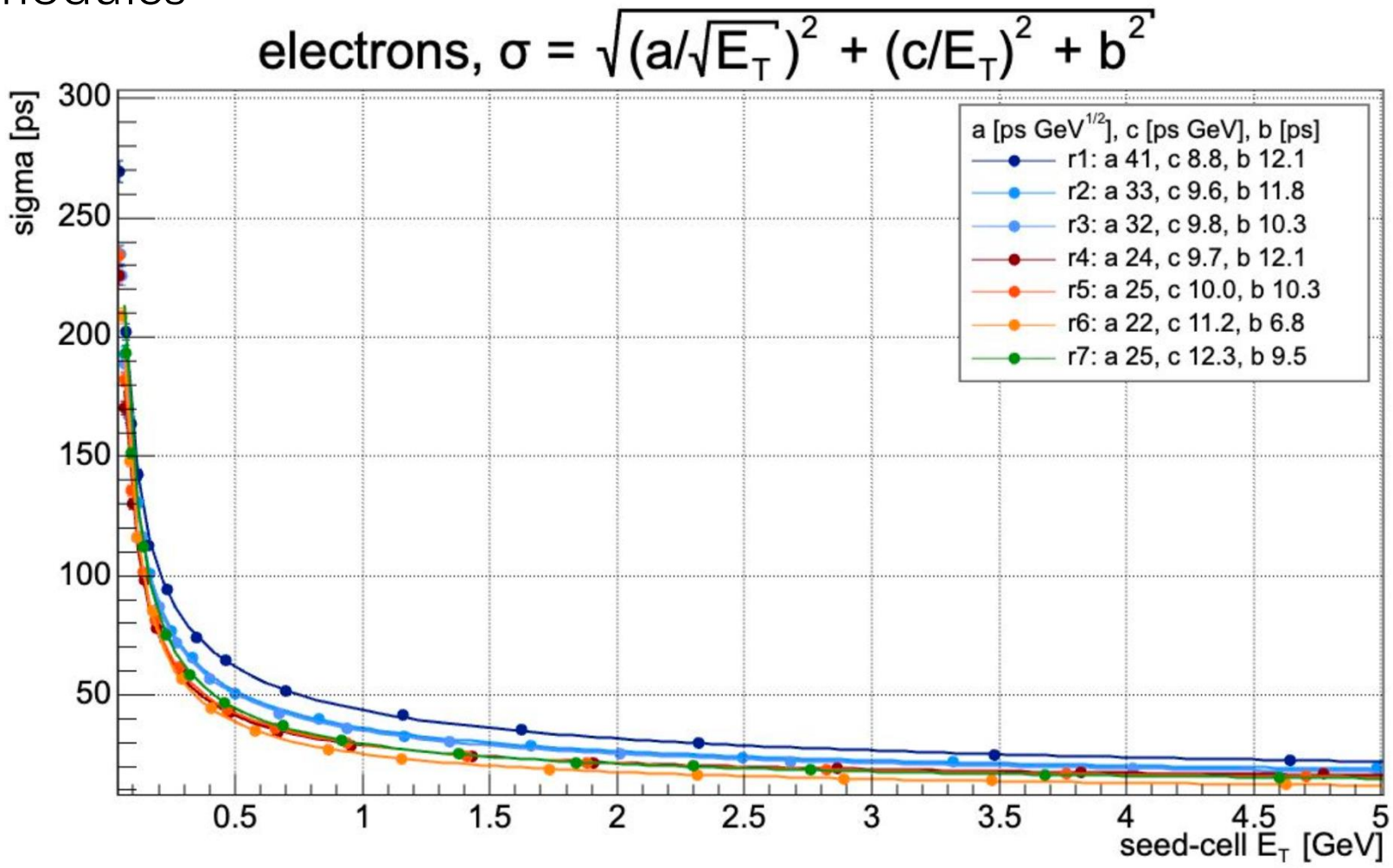
We see that the match between hybridMC and Boole is not there yet but it can be explained !



Problem: 100 GeV pion leaves anything from 0.3 to 30 GeV in the seed cell, and the mean time drifts across that range. HybridMC keeps that drift in its resolution; Boole doesn't take that into account in the current templates i.e the mean time of arrival is not taken into the parameters, only the pulse shape

This could be solved by either adapting the templates for the pions or fixing the definition of time of arrival of the particles

(E. Caire et al., 2026)

Resolution vs. E_T – ParticleGun, With Noise

Good agreement with the empirical formula, with different parameter values for different modules