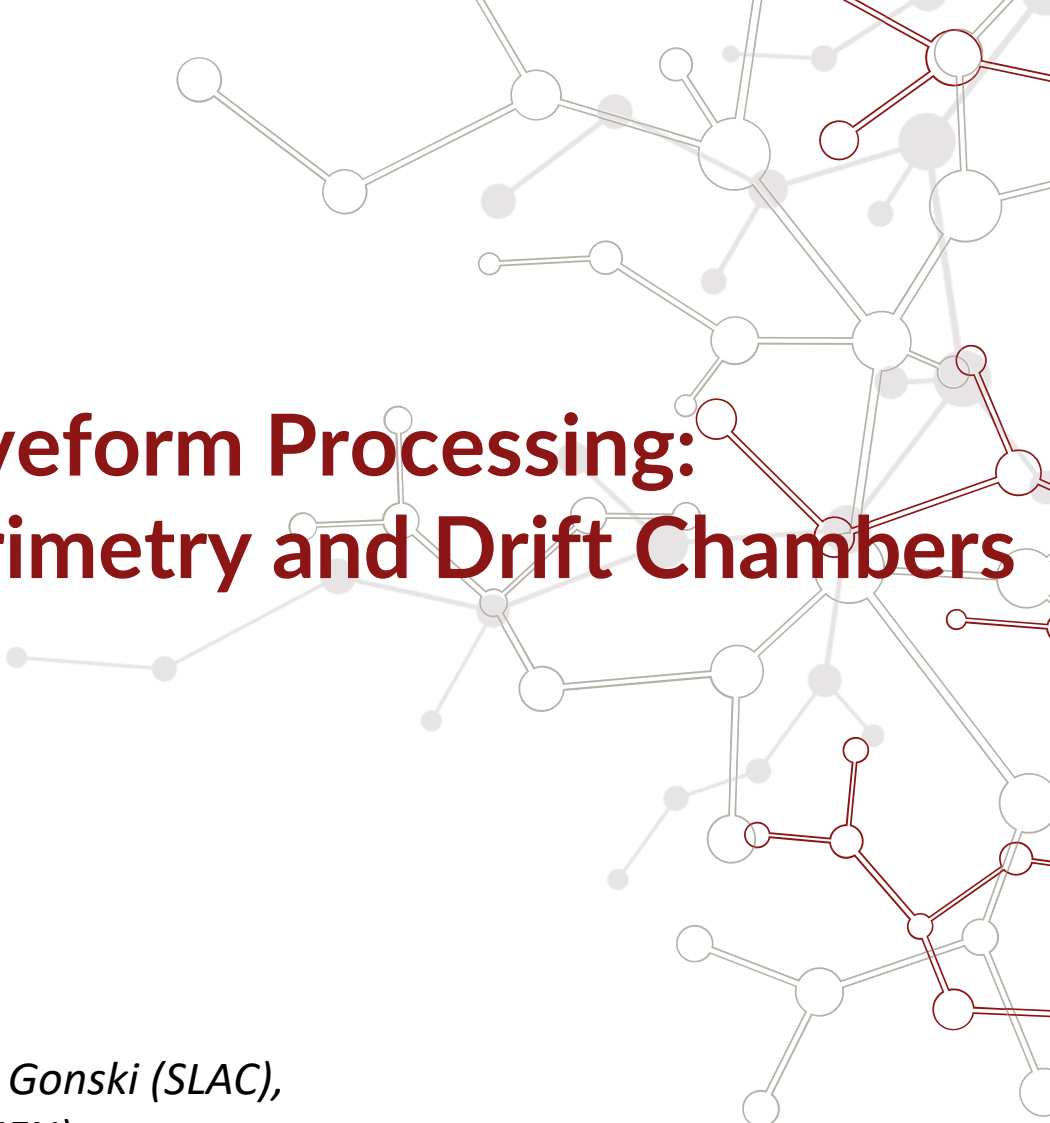




On-detector ML for Real-Time Waveform Processing: Applications to Dual-Readout Calorimetry and Drift Chambers

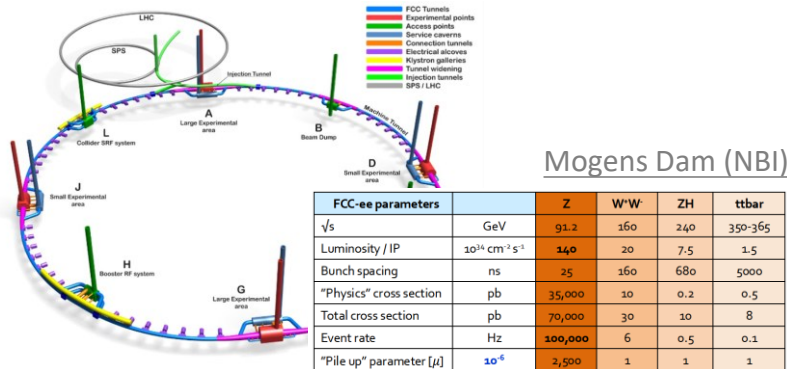
Speaker: Qibin LIU (SLAC)

CPAD2026 @ UW Oct 2026

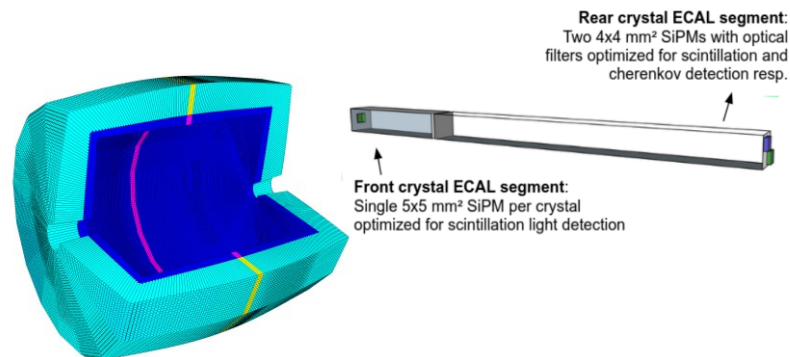
*Collaboration work with Liangyu Wu, Gia Ancone (Stanford) and Julia Gonski (SLAC),
Marco Toliman Lucchini, Marcello Campajola and Stefano Moneta (INFN),
Larry Ruckman, Aseem Gupta (SLAC)*

Readout Challenge For Modern Detector

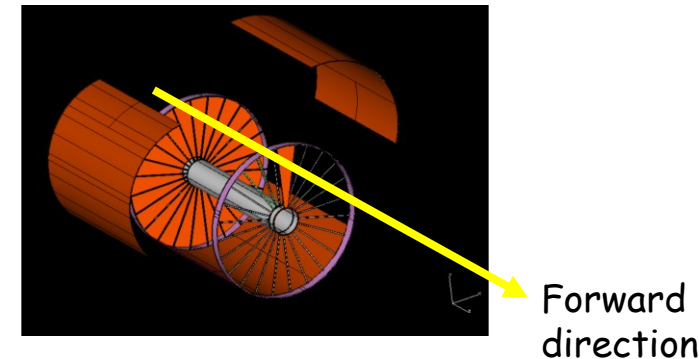
High event rates, high granularity, and strict material budget



FCC-ee luminosity up to $10^{36} \text{ cm}^{-2} \text{ s}^{-1}$
 Physics events at **200 kHz**
Trigger-less readout design under discussion



Dual-readout calorimeter @ IDEA FCC [2502.21223](#)
 High granularity up to **1.6M** readout channels
 Waveform readout, sampling rate up to **GSa/s**



Drift chamber @ IDEA FCC [2502.21223](#)
 High transparency – low materials
 5.0% X_0 forward, **75% from electronics**

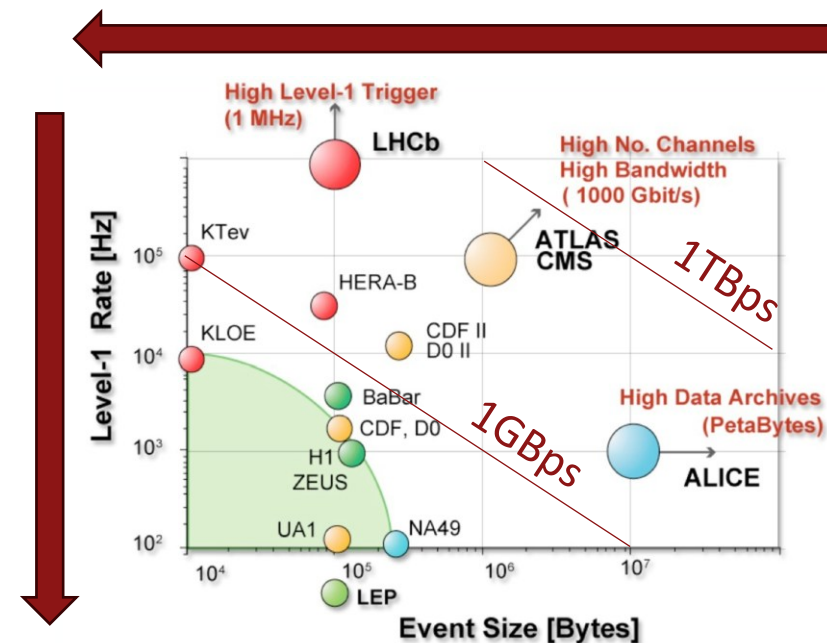
Data Reduction At Source: On-detector Real-Time ML

Max reduction when data processed BEFORE transmission

- Reduce event rate: trigger and streaming filter
- Reduce event size: data compression
- Real-time algo needed, limited resources → highly customized

Machine Learning (ML) provides general algorithm

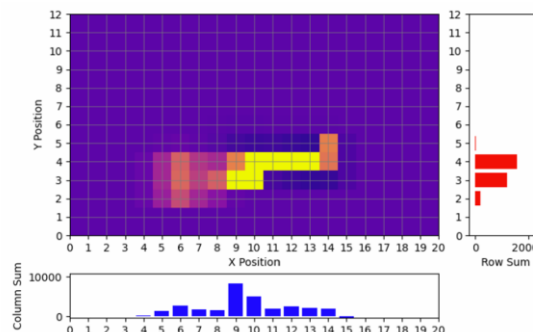
- Generic and reusable architecture for different application
- Learn from data as a universal function approximator
- Resources usage and timing predicable and constant
- Scale up with resources: continue to improve with more training data.



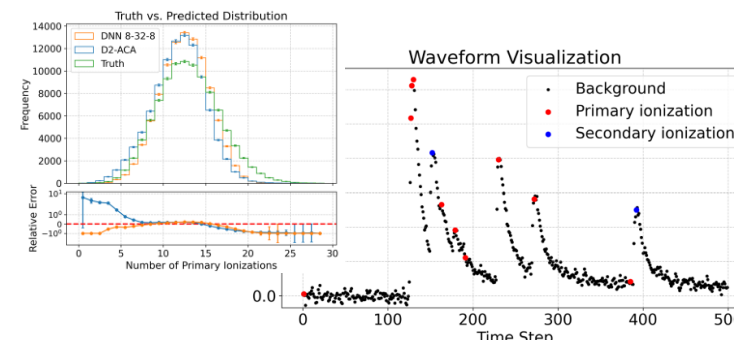
Examples of real-time ML algorithm for future HEP detector

Waveform Applications

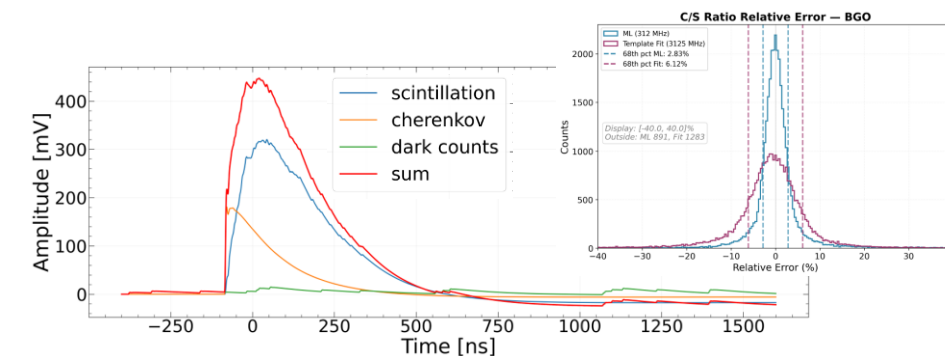
Smart-Pixel Signal Disc. with BDT



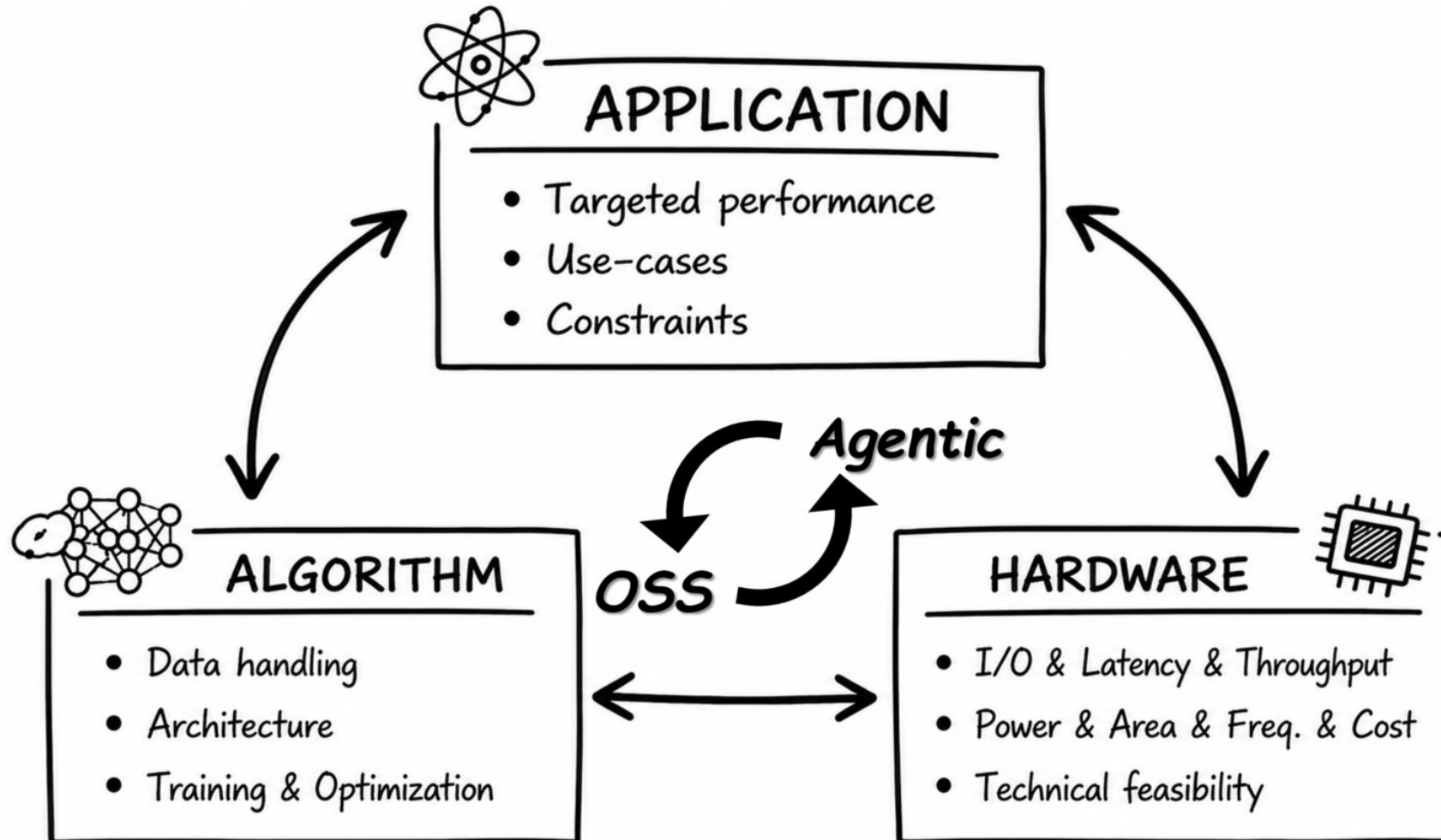
Drift Chamber dN/dX PID with NN



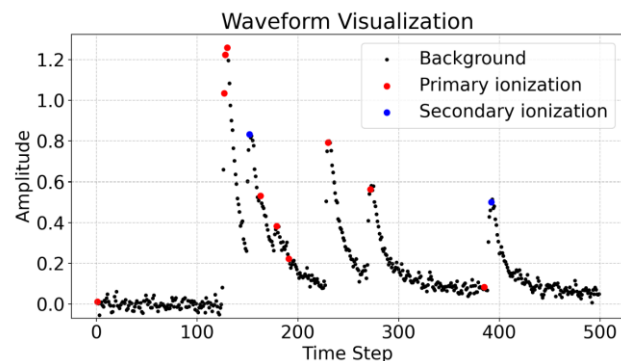
Dual-readout Calo. C/S separation with NN



Co-design of On-detector Real-Time ML Algorithm



Application: Waveform Processing for Detector Readout



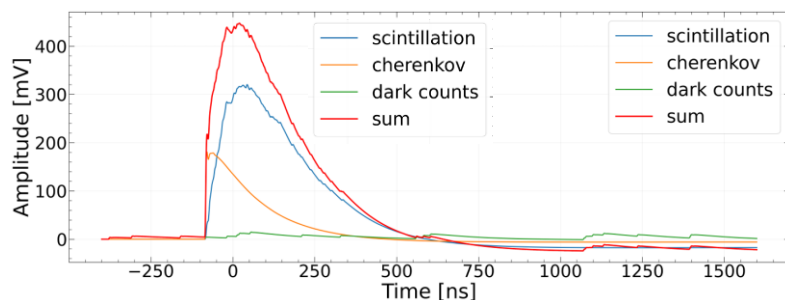
Peak counting



$$\sum_i^N P(A_i, T_i) + \sum_i^{N'} P'(A'_i, T'_i) + \text{nonlinear}$$

Primary Peak Pulse (points to $P(A_i, T_i)$) Secondary peak (points to $P'(A'_i, T'_i)$)

Amplitude (points to A_i) Time (points to T_i)



Component deconvolution



$$\sum_i^C P(T_i) + \sum_i^S P(T'_i) + \text{nonlinear}$$

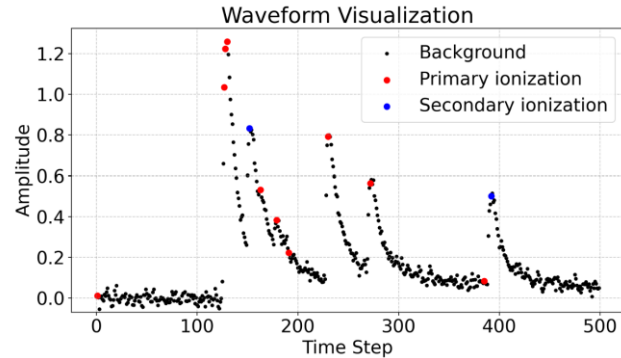
Single Photon Pulse (points to $P(T_i)$)

T_i and T'_i follow different templates

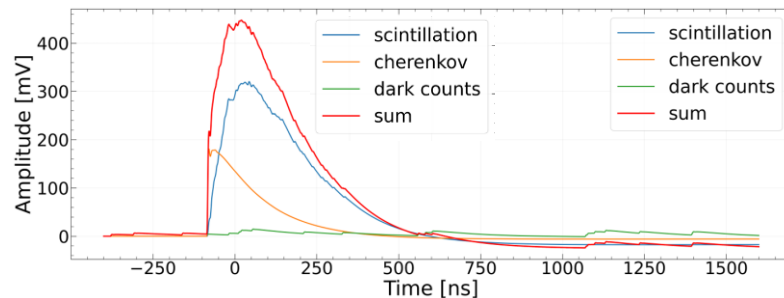
Much of the raw detector data in HEP comes as waveforms

- Sparse information: Mainly interested in charge (integral), amplitude, timing, etc.
- Complex signals: Noise, overlapping pulses, cross-talk, and nonlinear effects.
- Online processing: Greatly reduces data by transmitting physics information instead of raw waveforms.
- Real-time constraints: Conventional fitting is often too slow; ML offers an efficient alternative.

Algorithm: Design of Architecture



Peak counting



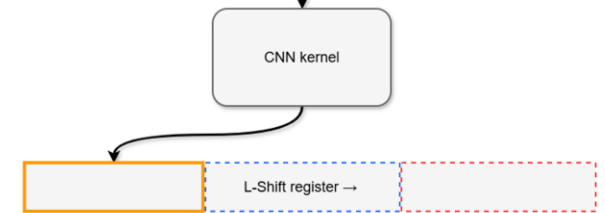
Component deconvolution



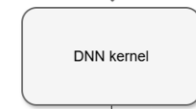
Input
Waveform



Shift Registered
Buffer



DNN kernel



Output

General hardware-aware ML design:

- CNN+DNN architecture: CNN captures time translation feature and DNN for non-linear correction
- Hardware-friendly buffer: shift register
- Streaming I/O: match the requirement for readout

Algorithm: Training and Quantization

ML algorithm trained and optimized using High Granularity Quantization (HGQ) tool [Chang2025, 2026]

- Quantization Aware Training (QAT) and trainable quantization: optimized for compact on-detector ML
- EBOPs/LUTs estimation during training time: performance, resources and multiple objectives
- Quantized model is essentially compact integer calculation (bit operation) and zero suppressed

Algo: Differentiable Quantization

1: Let f be the trainable parameter for quantization bits.

$$\begin{aligned}\delta_f &\equiv x - f^q(x) \\ &= x - \left[x \cdot 2^f \right] \cdot 2^{-f}\end{aligned}$$

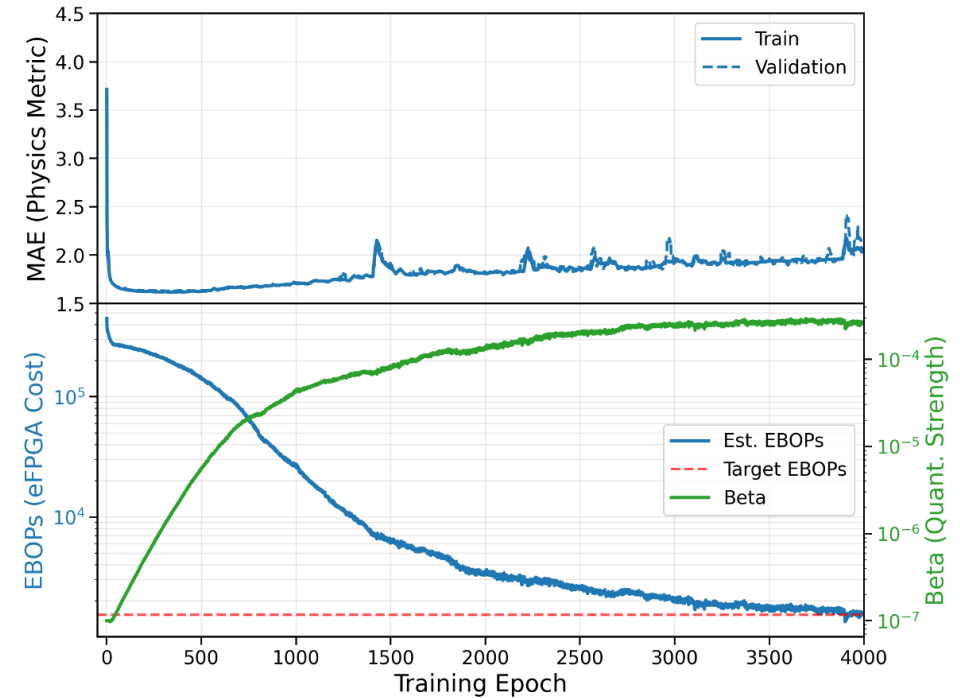
2: The gradient with respect to f is given by

$$\frac{\partial \mathcal{L}}{\partial f} = \frac{\partial \mathcal{L}}{\partial \delta_f} \cdot \frac{\partial \delta_f}{\partial f}$$

3: The first term is obtained from back-propagation.

4: The second term is approximated a surrogate gradient,

$$\frac{\partial \delta_f}{\partial f} \leftarrow -\log 2 \cdot \delta_f$$



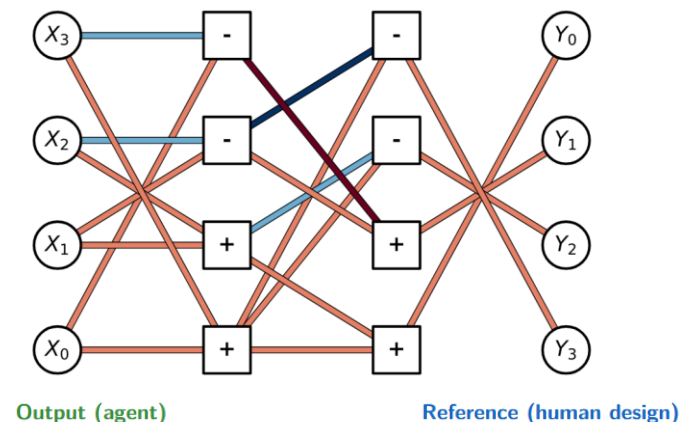
Algorithm: RTL Generation

RTL generation of ML kernel using the DA4ML [Chang2026]

- HLS-free, bit-exact mapping of ML model to RTL via symbolic tracing
- Optimized using distributed arithmetic: reuse the intermediate calculation
- LUT-only and combinational implementation (pipelining optional)
- Fully open-source toolchain including simulation

System integration using agentic AI-assisted method

- Interface integration, AXI-S handling and kernel scheduling
- Open-source agentic framework [Liu2026] utilized
 - Commercial solution widely available as well (e.g. Claude Code)
- Specification (RTL skeleton) to Codes with closed validation loop
 - Different RTLs achieve same functionality and timing are equivalent



```
...
localparam
    KERNEL_OUTPUT_BIT = 90;
kernel_wrapper u_kernel(...);
localparam SHIFT_REG_WIDTH
    = 20 * KERNEL_OUTPUT_BIT;
reg [W-1:0] shift_reg;
always @(posedge clk)
    if (inp_valid) begin
        shift_reg[count*90+: 90]
            <= kernel_out;
    end
    if (count==19) begin
        out_valid<=1; count<=0;
    end else begin
        count<=count+1;
        out_valid<=0;
    end
end end
dense_wrapper u_dense(...);
assign model_out={18'b0,dense_out};
...
```

```
...
localparam
    KERNEL_OUTPUT_BIT = 90;
localparam II = 20;
localparam SHIFT_REG_WIDTH
    = II * KERNEL_OUTPUT_BIT;
reg [W-1:0] shift_reg;
always @(posedge clk)
    if (inp_valid)
        shift_reg <= {kernel_out,
            shift_reg[W-1:90]};
reg [4:0] cnt = 0;
always @(posedge clk)
    if (inp_valid)
        cnt <= (cnt==19) ? 0 : cnt+1;
always @(posedge clk)
    out_valid <=
        (inp_valid && cnt==19);
assign model_out =
    {{18{1'b0}}, dense_out};
...
```

Hardware: embedded FPGA Target

On-detector applications impose special requirements:

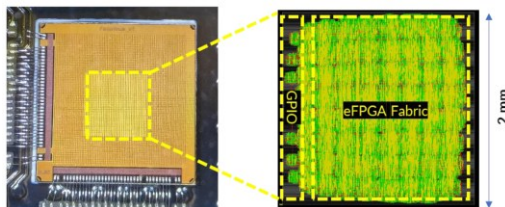
- Harsh environments → radiation tolerance, extreme temperatures, ...
- Tight resource and footprint constraints → low material budget and low power consumption
- Configurability and programmability → enable in-situ upgrades and flexible deployment

Embedded FPGA (eFPGA) provides a unique and powerful solution:

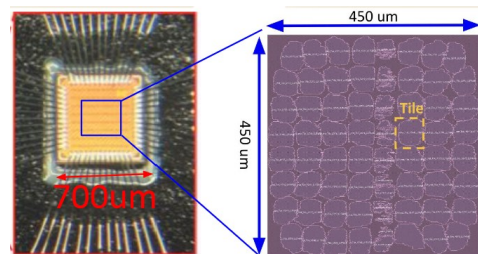
- Gate-level flexibility integrated into on-detector chips
- Enable customized algorithms optimized for available resources
- A promising platform for real-time on-detector ML

eFPGA R&D @ SLAC

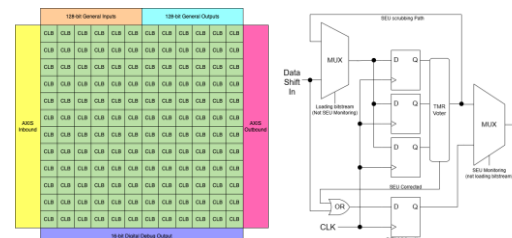
Larry Ruckman



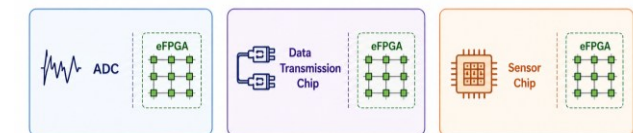
eFPGA prototype v1
130 nm CMOS 5 mm x 5 mm



eFPGA prototype v2
28 nm CMOS 1 mm x 1 mm



eFPGA prototype v3
Work in Progress



eFPGA next generation
Real Embedded **FPGA**

Application

Cluster Counting for Drift Chamber on-detector Readout

Readout for Drift Chamber Detector @ IDEA FCC-ee

Tracking detector with precise spatial resolution and very low material budget

Mature, cost-effective technology:

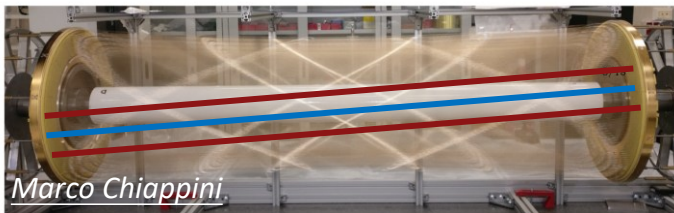
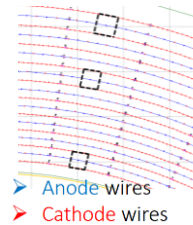
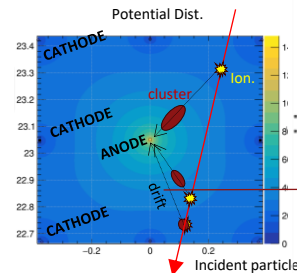
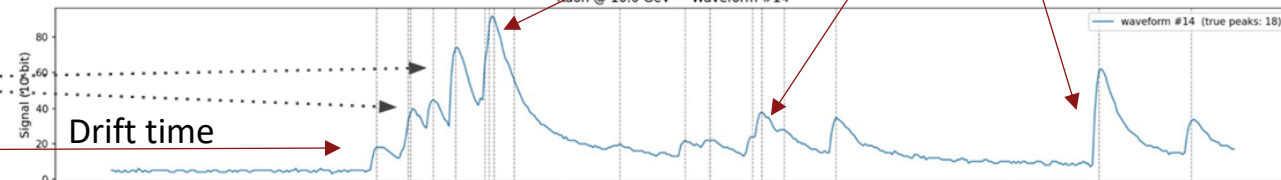
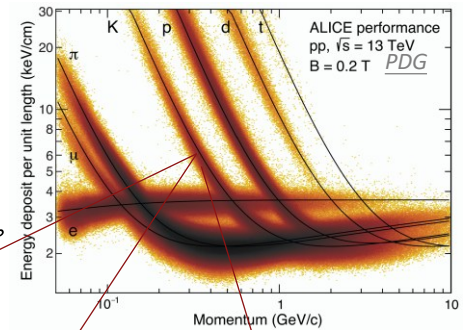
- Particle ionization in gas, followed by drift, amplification, and signal collection under an EM field

Readout design: precise timing measurement combined with pulse-shape analysis

- **Leading-edge timing:** measures drift time and reconstructs the **incident particle position**
- **Pulse shape and peak structure:** encode ionization information, enabling **particle-identification (P-ID)**

Full waveform readout from all channels generates enormous data rates

- **Real-time waveform processing is motivated and necessary**
- **Highly reduced transmission: N sampling $\rightarrow$ 1 output**

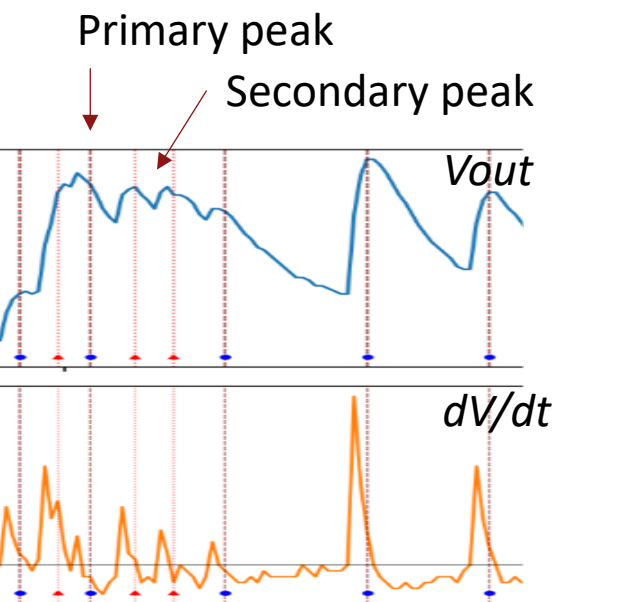


ML-based Real-Time Cluster Counting

12

Principles of P-ID: ionization power ~ charge or number of clusters (peaks)

- Derivative-based methods: large uncertainty from secondary peaks.
- Offline ML [Tian2025]: improved performance, GPU-based.
- Online ML [Yilmaz2025]: FPGA-based, but resource-heavy (~30k LUT6 + DSPs).



ALGORITHM I BUFFERED STREAMING CNN ARCHITECTURE FOR REAL-TIME PRIMARY-CLUSTER COUNTING.	
Input: $\mathbf{x}_m = (x_{m,0}, \dots, x_{m,N-1}) \in \mathbb{Z}^N, x_{m,i} \in \{0, 1\}^{B_{\text{ADC}}}$	
Output: $\hat{y} \in \mathbb{R}$	
Parameters: $N, B_{\text{ADC}} = 10, F = 5$ $W_{\text{in}} = NB_{\text{ADC}}, L = 500$ $II = L/N, B_{\text{kern}}, W_{\text{sr}} = FB_{\text{kern}} II$	
# Operation (quantization omitted)	Dimension
<i>Conv1D with kernel size $N = \text{stride } N$</i>	
1 $k_{m,f} = \sigma\left(\sum_{i=0}^{N-1} w_{f,i}^{(c)} x_{m,i} + b_f^{(c)}\right), \quad f = 1, \dots, F$	$\mathbf{k}_m \in \mathbb{R}^F$
<i>Shift-register buffer</i>	
2 $\mathbf{S}_m \leftarrow \{\mathbf{k}_m, \mathbf{S}_{m-1}[0 : F(II - 1) - 1]\}$	$\mathbf{S}_m \in \mathbb{R}^{FII}$
<i>MLP prediction head</i>	
3 $\mathbf{h}_1 = \sigma(\mathbf{W}_1 \text{vec}(\mathbf{S}) + \mathbf{b}_1)$	$\mathbf{W}_1 \in \mathbb{R}^{16 \times FII}$
4 $\mathbf{h}_2 = \sigma(\mathbf{W}_2 \mathbf{h}_1 + \mathbf{b}_2)$	$\mathbf{W}_2 \in \mathbb{R}^{8 \times 16}$
5 $\hat{y} = \mathbf{W}_3 \mathbf{h}_2 + b_3$	$\mathbf{W}_3 \in \mathbb{R}^{1 \times 8}$
<i>Sequential control</i>	
6 $\text{cnt} \leftarrow \text{cnt} + 1$	
7 if $\text{cnt} = II$: $y \leftarrow \hat{y}, \text{cnt} \leftarrow 0$	

Ultra-small ML algorithm co-designed for on-detector chip

- Input: 500 ADC samples (10-bit, 1.5 GHz), demultiplexed N samples/cycle.
- Conv1D: 5 shared-weight filters (kernel = stride = 25).
- Buffer: Accumulate 5 features/cycle over 500/N cycles (5*N features).
- MLP: 5*N → 16 → 8 → 1; predict one primary-cluster count per waveform.

Performance Evaluation and Optimization

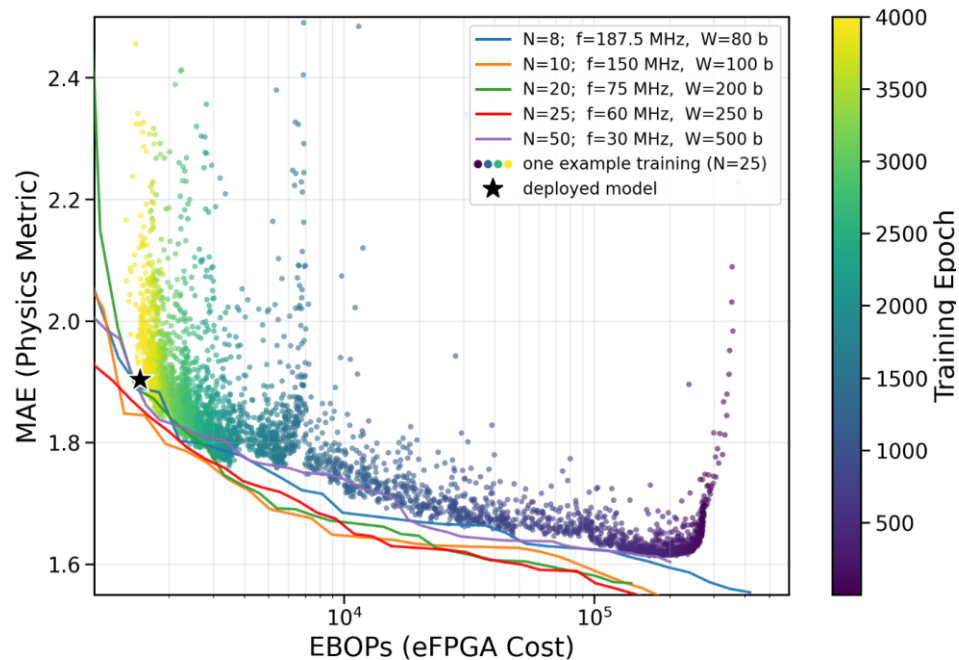
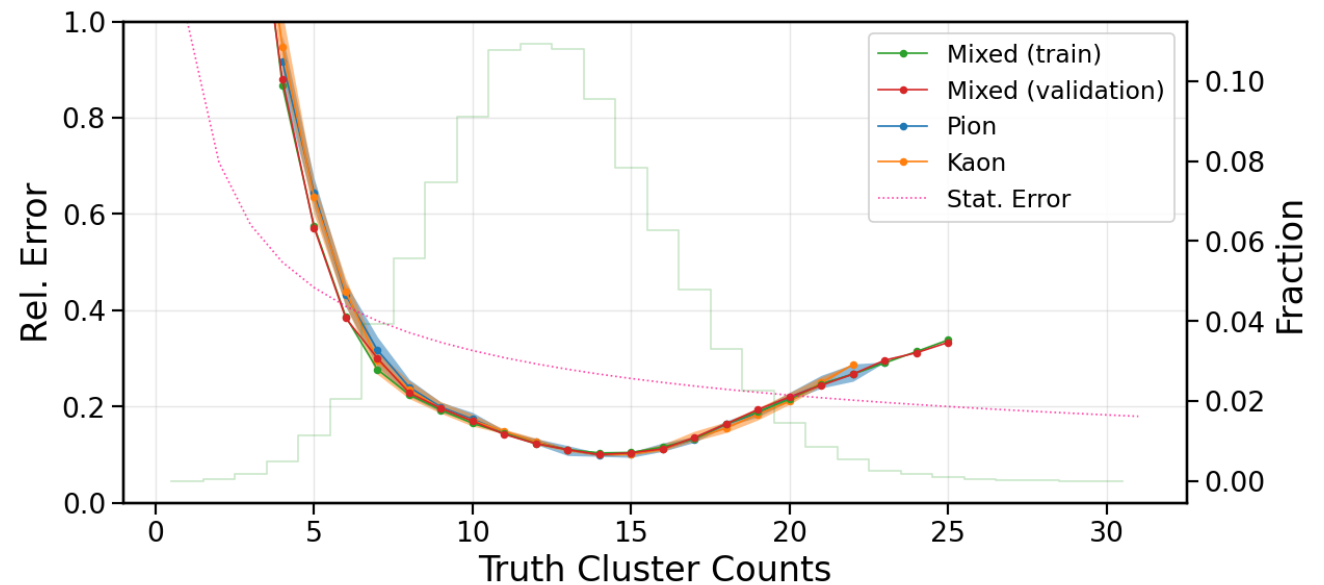
13

Algorithm is optimized with multiple objectives: error of the predicted count, hardware utilization and latency

- Approach the pareto front: at given resources and latency, this algorithm provides optimal physics performance

Cluster-counting performance evaluated for pions and kaons across particle momenta (0.2–20 GeV).

- Rel. error as low as 10% (on top of intrinsic stat. unc.) with no significant bias across particle types or momenta.



RTL Implementation for On-detector Target

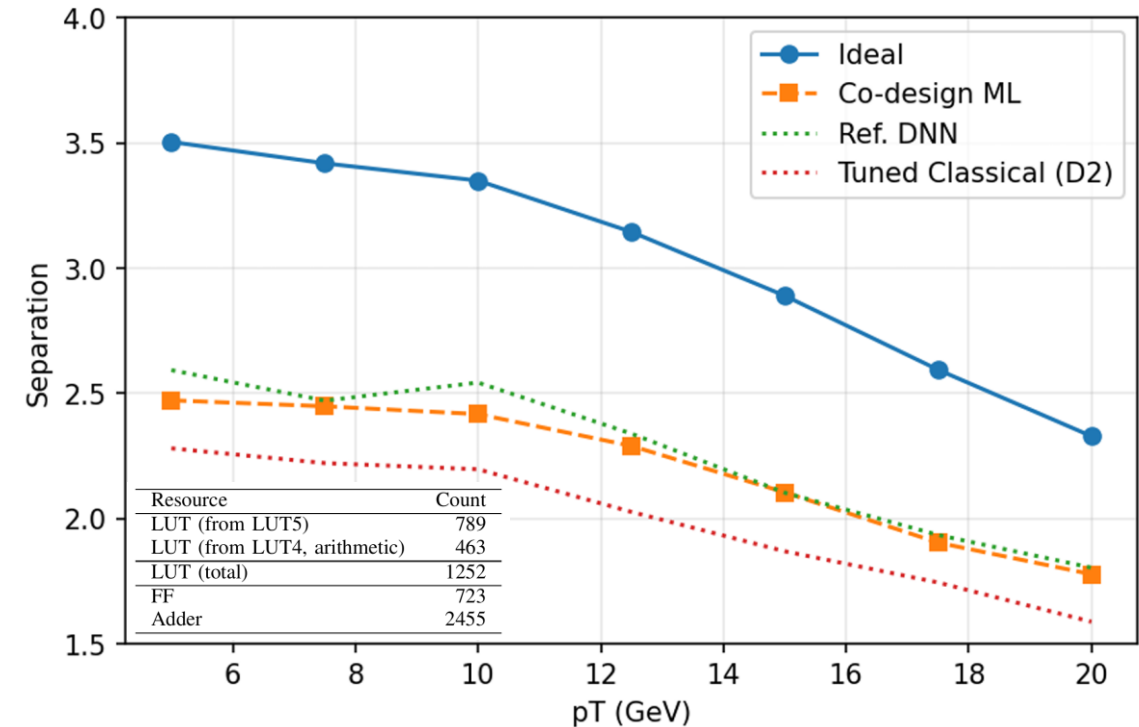
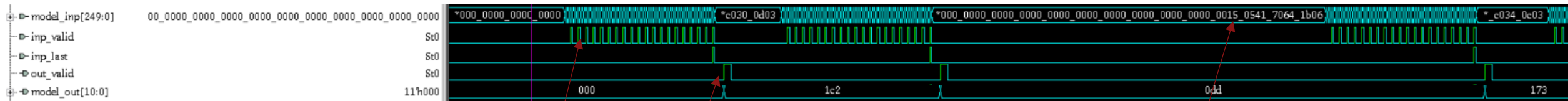
Kaon/pion separation evaluated as physics benchmark

- More accurate cluster counting → Better PID.
- Ground-truth cluster counts → Ideal PID separation.

Ultra-small ML achieved with ~1K LUTs

- Outperforms derivative-based (D2) peak finding.
- Comparable performance to reference FPGA ML (HLS).
- More than 20× lower resource usage than reference ML.

RTL fully validated in the system with AXI-S interface



Input Streaming
25 Sa/cycle

Output Burst
20 cycle/event

Emulation of Input Stall

Application

Waveform decomposition for Dual Readout Calorimeter

Dual-readout Calorimeter at FCC-ee

Different particles exhibit different electromagnetic-to-hadronic (EM/Had) shower fractions in the calorimeter

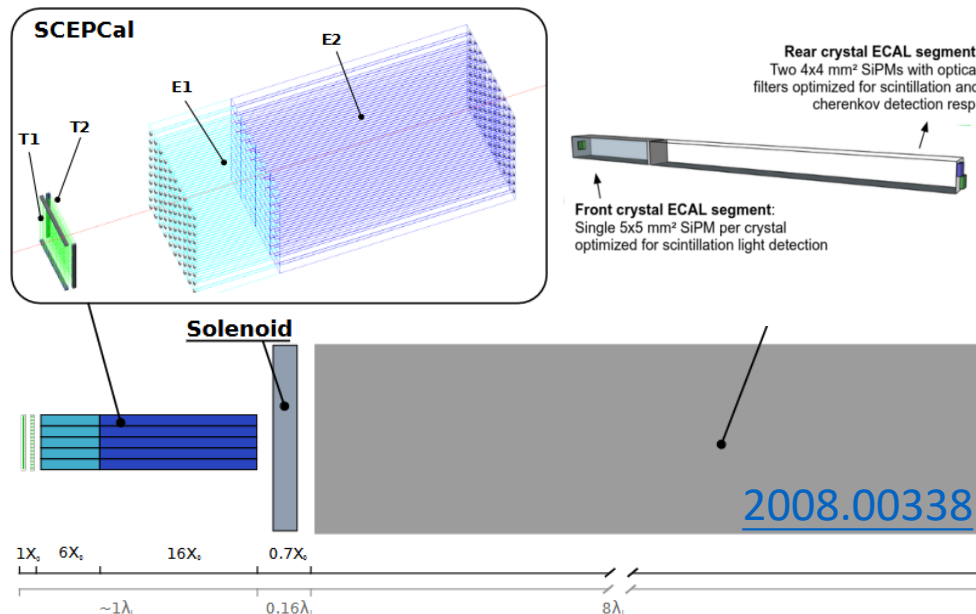
- Trouble for the **energy calibration** and highly downgrade the scale performance

Dual-readout calorimeter measures both scintillation and Cherenkov signals

- **Cherenkov process is primarily sensitive to relativistic particles and provides an additional measurement**

Combining and calibrating the two measurements, energy resolution can be improved

Challenge to readout: overlapping Cherenkov and scintillation in single SiPM -- **waveform processing is necessary to decompose C/S**



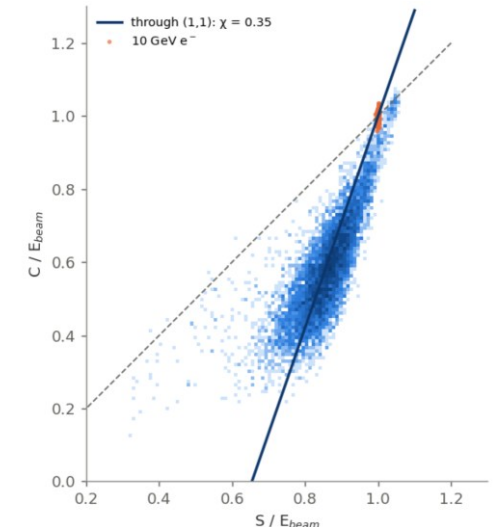
Energy dependent

$$S = E_{true}(f + (1 - f)h_S),$$

$$C = E_{true}(f + (1 - f)h_C).$$

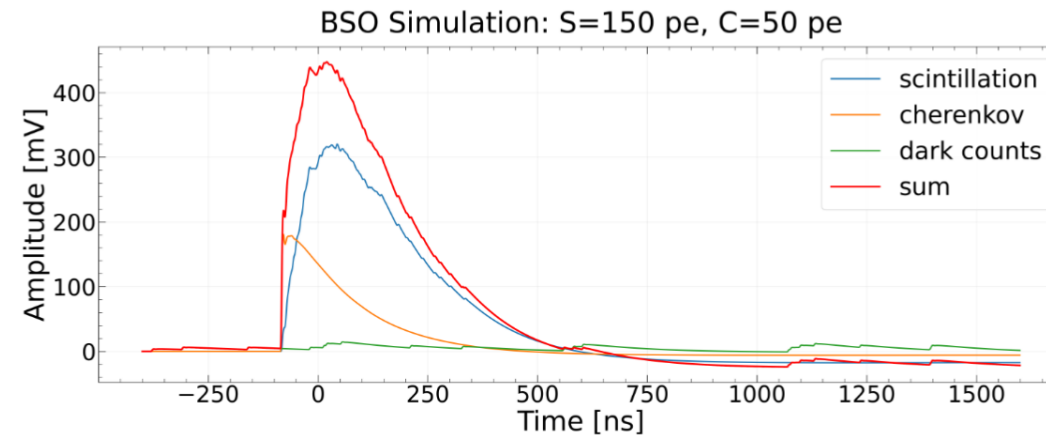
Energy independent
(Calibratable)

Simulation of
10 GeV Kaon, PWO
Fully contained (ideal)



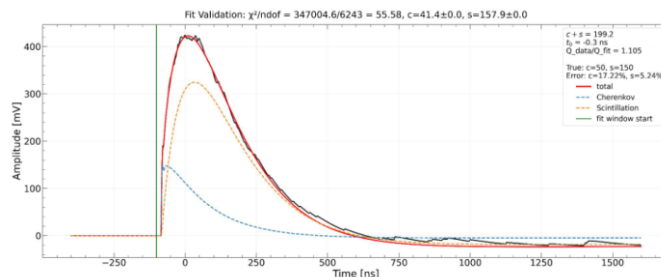
C/S Decomposition: Template Fitting v.s. ML Algorithm

17

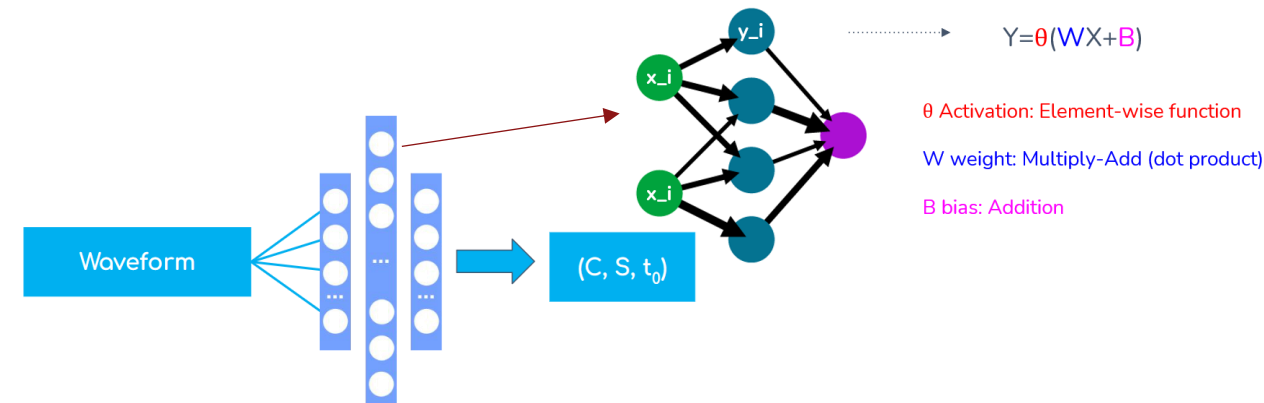


Template Fitting

$$WF(t) = c \cdot C_{\text{template}}(t - t_0) + s \cdot S_{\text{template}}(t - t_0) + \text{offset}$$



ML-based Algorithm

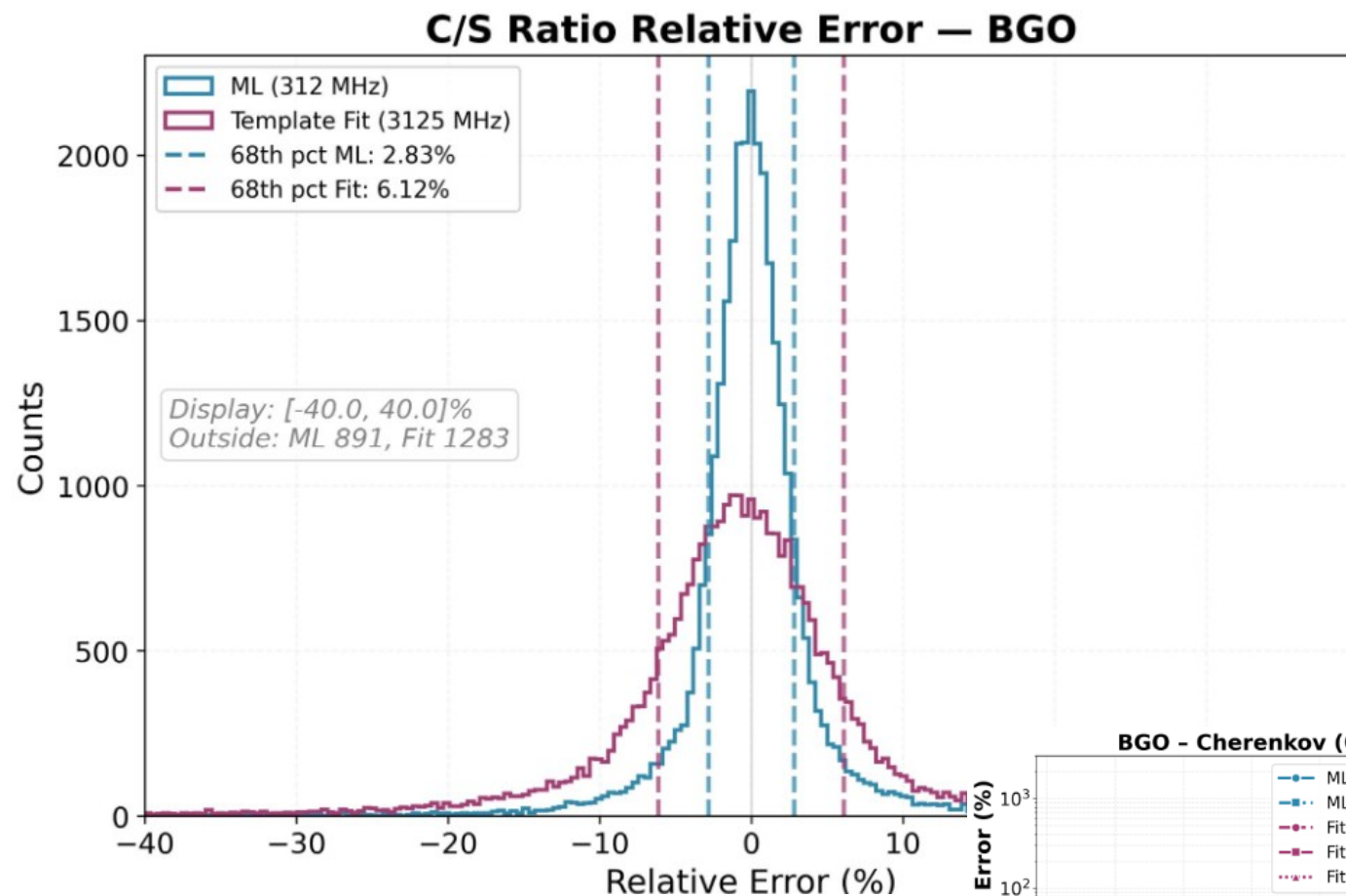


- Assuming linear scaling of the pulse
- Initial value and fitting range need tuning
- Fit the C and S component according to template
- Only possible offline due to the iterative minimization

- Forward pass-through non-linear layers
- Parameters fits on labeled data (simulation)
- Predict directly the number C-photon and S-photon
- Hardware friendly and streaming computation

Study of Physics Performance

18

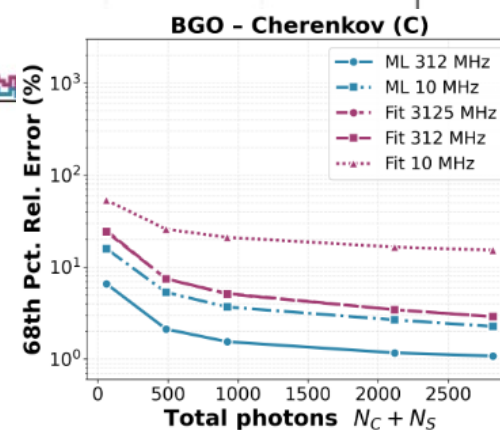
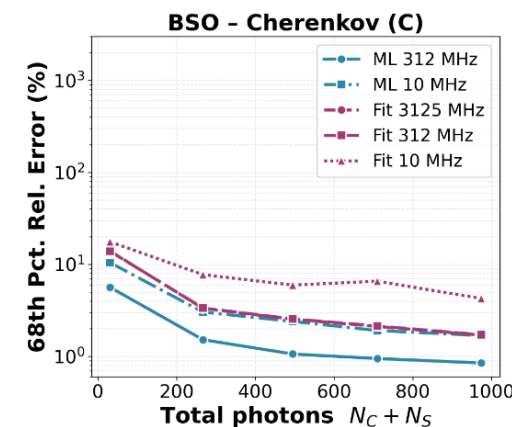
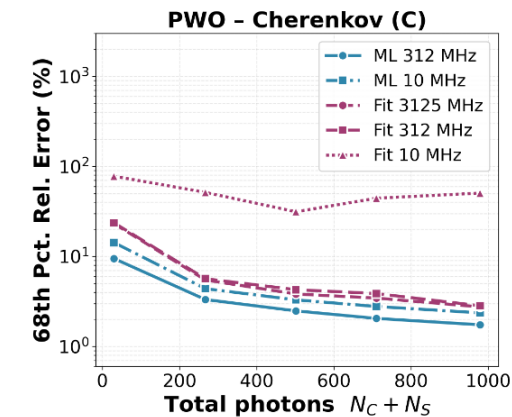


Evaluation across Particle Types and Energies

- 50:50 electrons and kaons, at 1, 5, 10, and 30 GeV.
- Single model generalizes across energies and broad C/S range

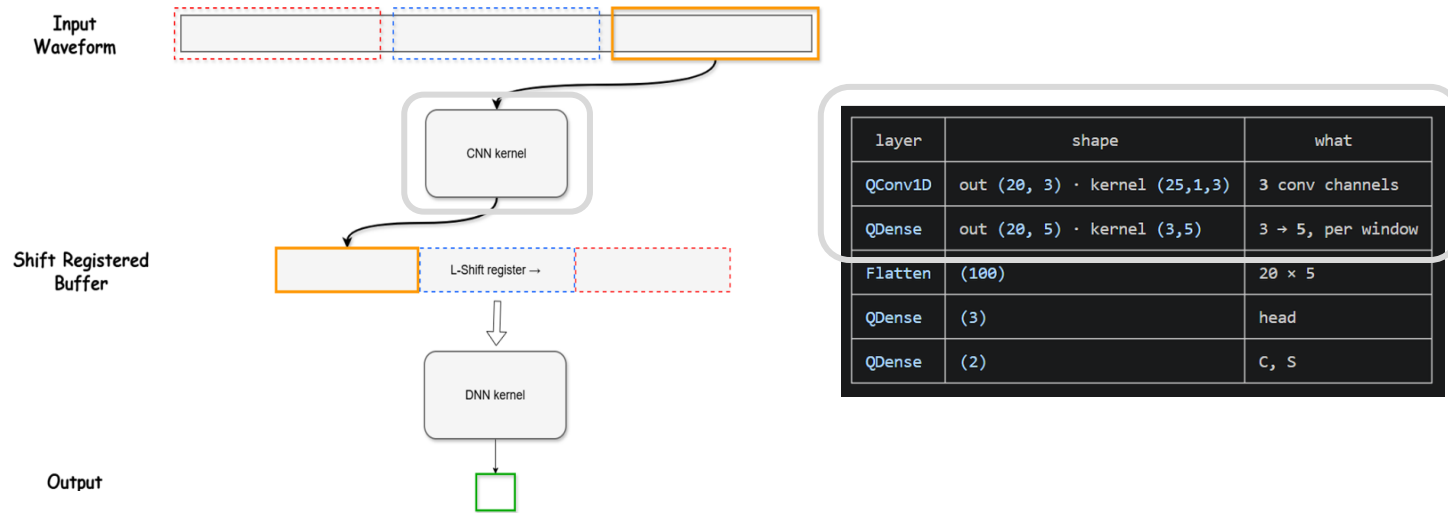
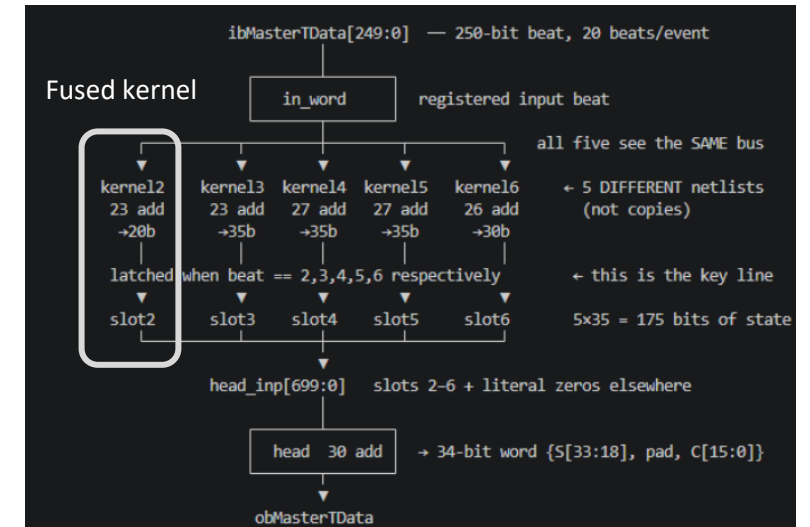
Performance across Crystals and Sampling Rates

- BGO, BSO, PWO; sampling rates from 10 to 312 MHz.
- C/S ratio rel. error: 3.1–7.4% (312 MHz); 5.8–11.5% (10 MHz).
- Comparable to or better than template fitting at 3.125 GHz



RTL Implementation for On-detector Target

19

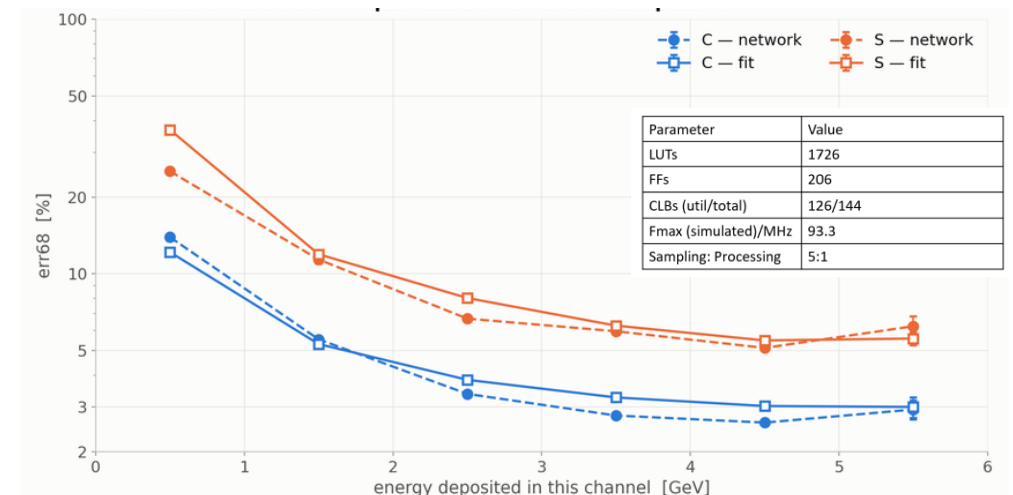


Compression and Re-architecture for On-Detector Deployment

- Sliding CNN-DNN for filtering and nonlinear correction.
- Kernel fusion for compact RTL and routing.

Hardware-aware scheduling for resource efficiency.

- Performance Maintained at only 1.7k LUTs
- 30x resource reduction vs. plain MLP and HLS (C++) implementation.
- FPGA emulation at 93.3 MHz, supporting ADC rates up to 460 MHz.



Outlook

Unlocking the Scaling Potential of On-Detector ML

Unlocking the Scaling Potential of On-Detector ML

Key Strength of ML: Generality and Scalability

- Performance scales with data and training, potentially approaching statistical limits across tasks.

On-Detector ML Cannot Fully Exploit This Advantage

- Limited task-specific labeled/calibration data.
- Restricted model capacity due to hardware constraints.

Can Foundation Models and Knowledge Distillation Break These Limits?

- Leverage large-scale pretraining and distill into compact on-detector models.

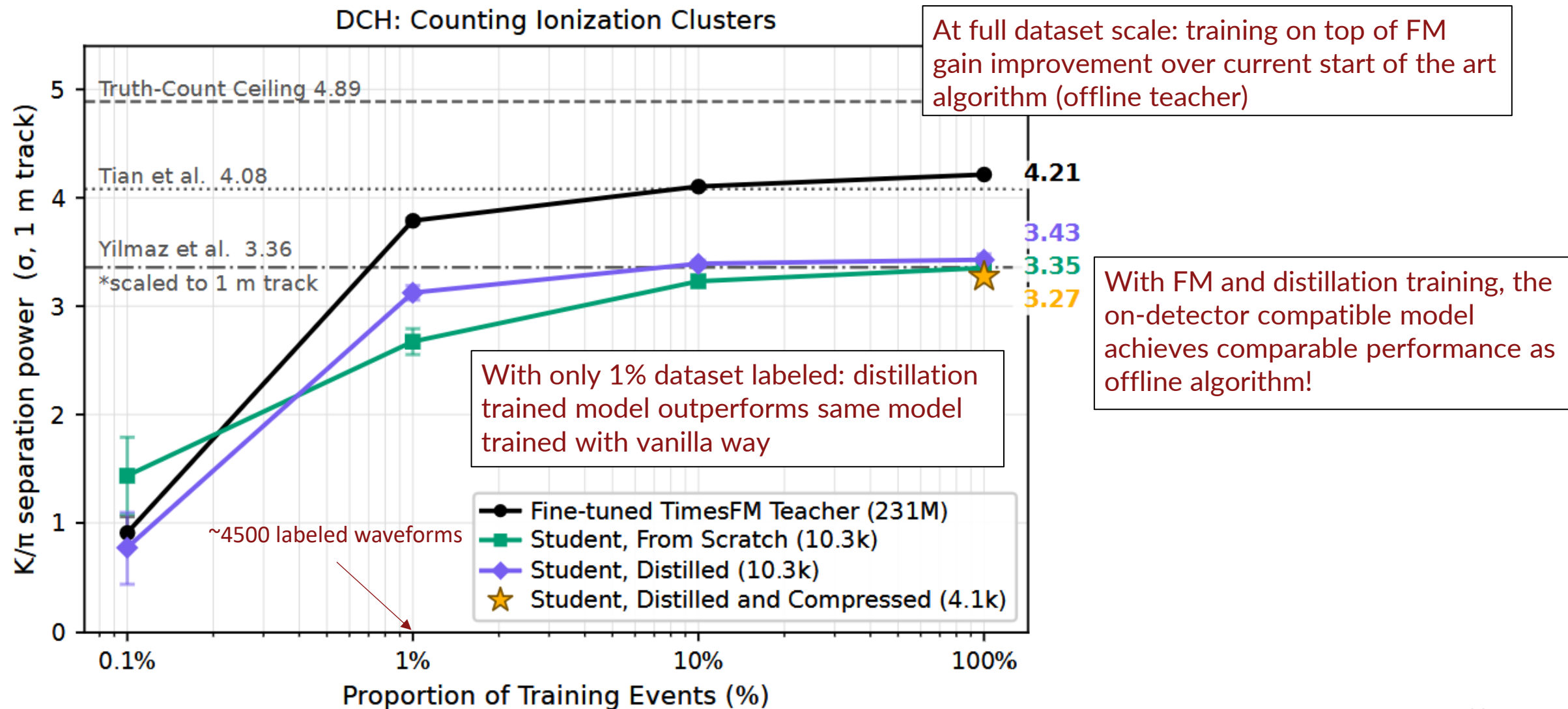
PRIMED – PRetrained Industrial Models for the Edge of Detectors

- Fine-tune Google's time-series foundation model (TimesFM) and distill for waveform readout.
- Preliminary result: task-specific performance benefits from general large-scale pretraining.

Bridge the Online–Offline Divide

- Bring large-scale offline learning to real-time on-detector ML.

PRetrained Industrial Models for the Edge of Detectors



Summary

High Granularity and Event Rates Make Data Movement a Key Bottleneck in HEP

- Real-time on-detector waveform processing reduces data at the source.

End-to-End Co-Design for Ultra-Small On-Detector ML

- Hardware-friendly architectures, quantization, and hardware-aware optimization.
- Direct RTL and agentic-assisted implementation, validated for embedded FPGA (eFPGA) applications.

ML Applications for Real-Time On-Detector Processing

- Ultra-small ML for primary-cluster counting in drift chambers.
- ML-based C/S waveform decomposition in dual-readout calorimeters.

Outlook: Unlocking the Scaling Potential of On-Detector ML

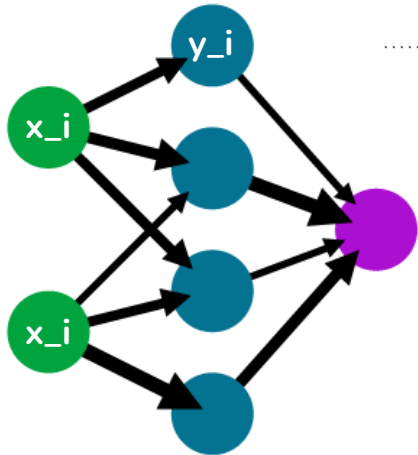
- Leverage foundation models and knowledge distillation to bridge offline learning and real-time on-detector processing.

Thanks!

How the ML actually runs on FPGA

A simple neural network

input layer hidden layer output layer



$$Y = \theta(WX + B)$$

θ Activation: Element-wise function

W weight: Multiply-Add (dot product)

B bias: Addition

```
mac_kernel(...){
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    y[idx] = w[idx] * x[idx] + b[idx]
}
// ...perform on N matrices
mac_kernel<<<blocks, threads>>>(X, W, B, Y, N);
```

Matrix/Tensor Op.

A $\vec{Y} = \theta(W \cdot \vec{X} + \vec{B})$

```
for (int i = 0; i < N; i++) {
    z += w[i] * x[i];
}
```

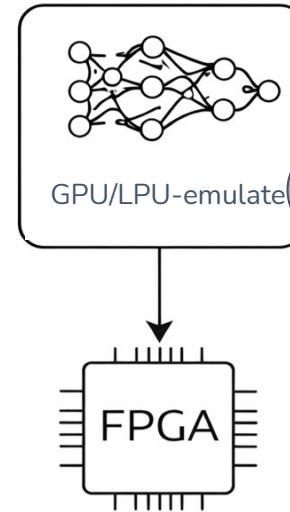
Scalar Op. + Loop

B $Y_i = \sum_j \theta(w_{i,j}X_j + b_i)$

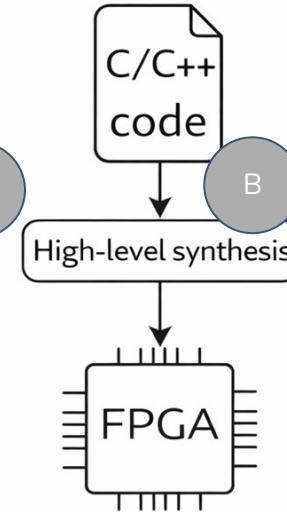
```
module mac2u(input [1:0] W, X,
             input [2:0] B,
             output [4:0] Y);
    wire w0=W[0], w1=W[1], x0=X[0], x1=X[1];
    wire t0=w0&x0, t1=w1&x0, t2=w0&x1, t3=w1&x1;
    wire p0=t0, p1=t1^t2, c1=t1&t2, p2=t3^c1, p3=t3&c1;
    assign S = {1'x0,p3,p2,p1,p0} + {2'x00,B};
endmodule
```

C Bit Operation

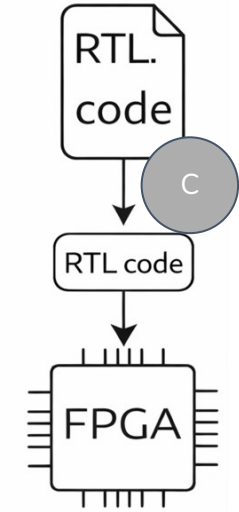
Accelerator
based



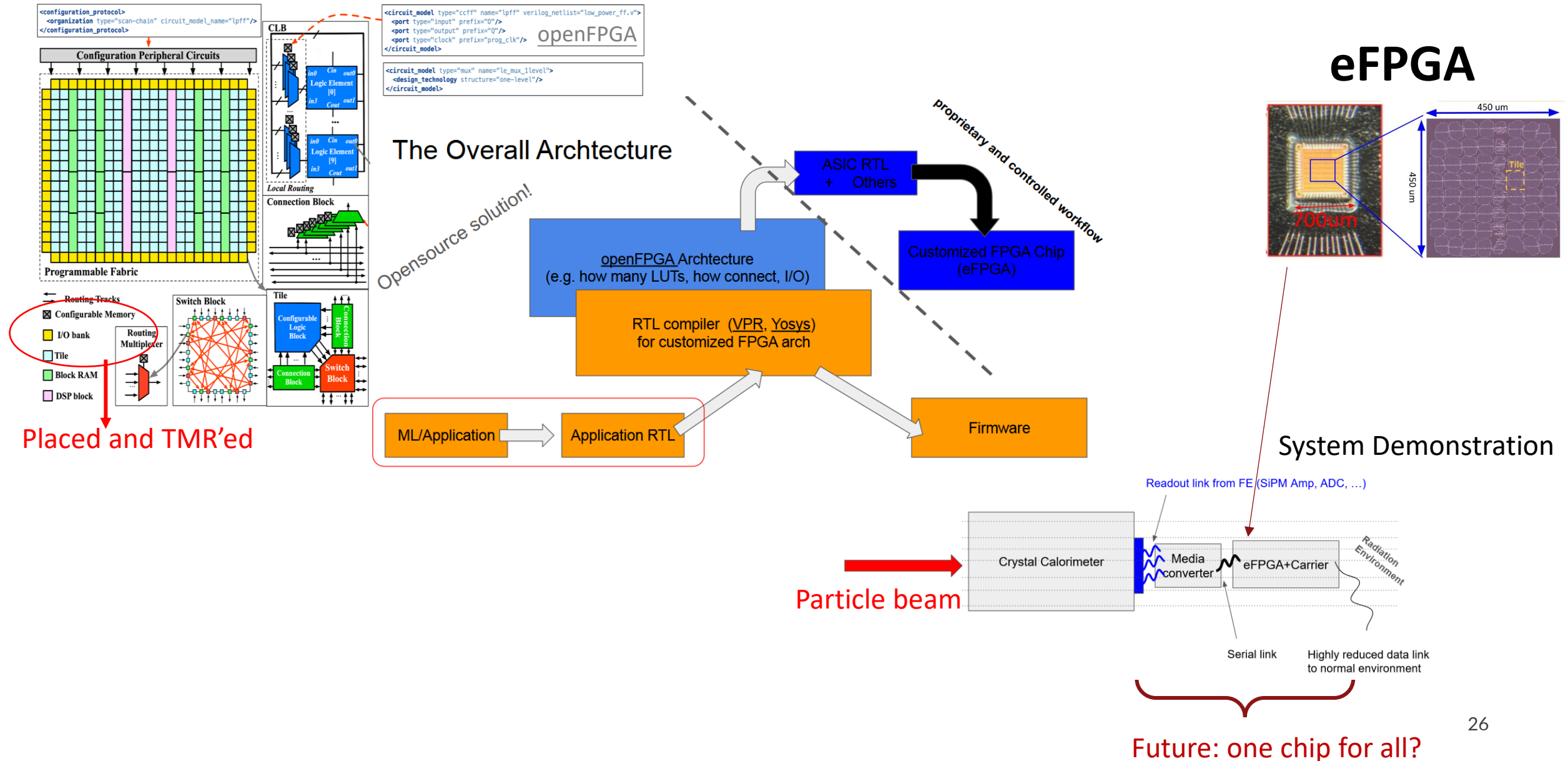
High-level
synthesis based



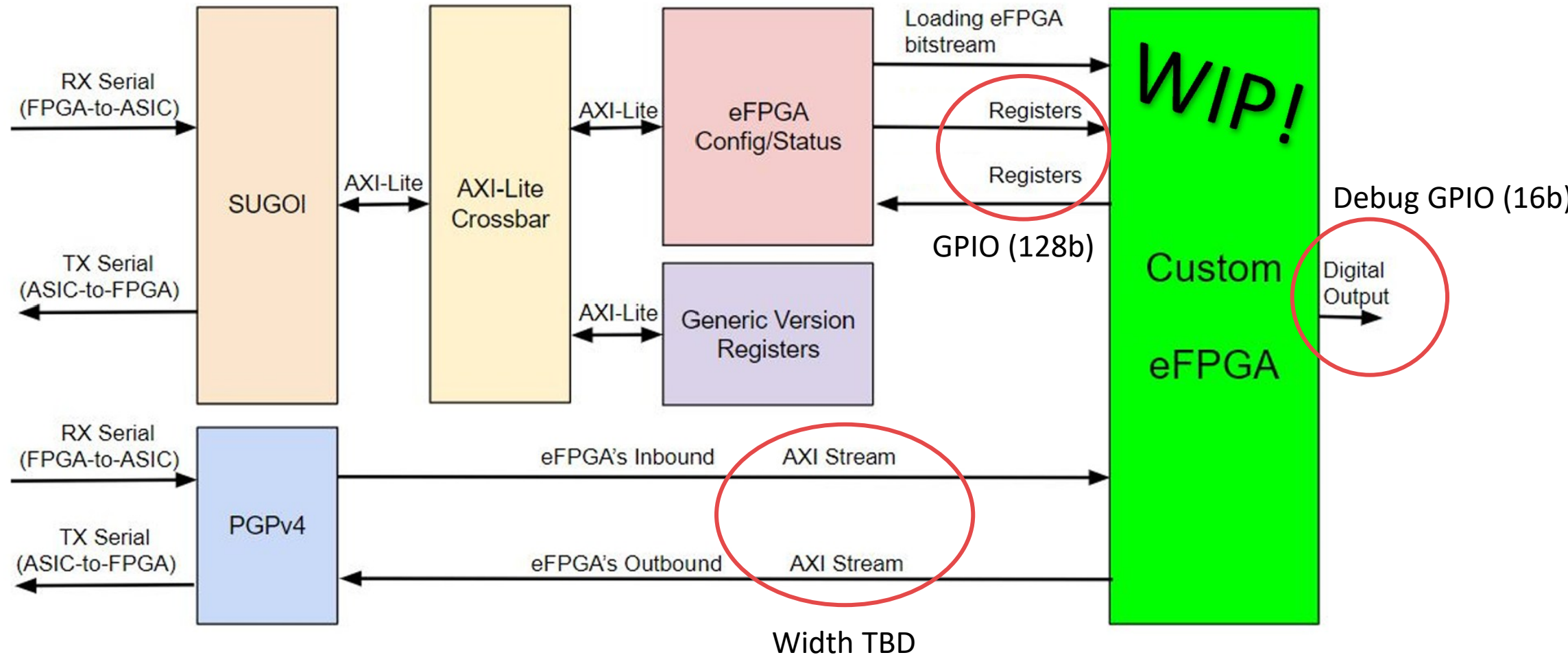
Bit-level
Synthesis



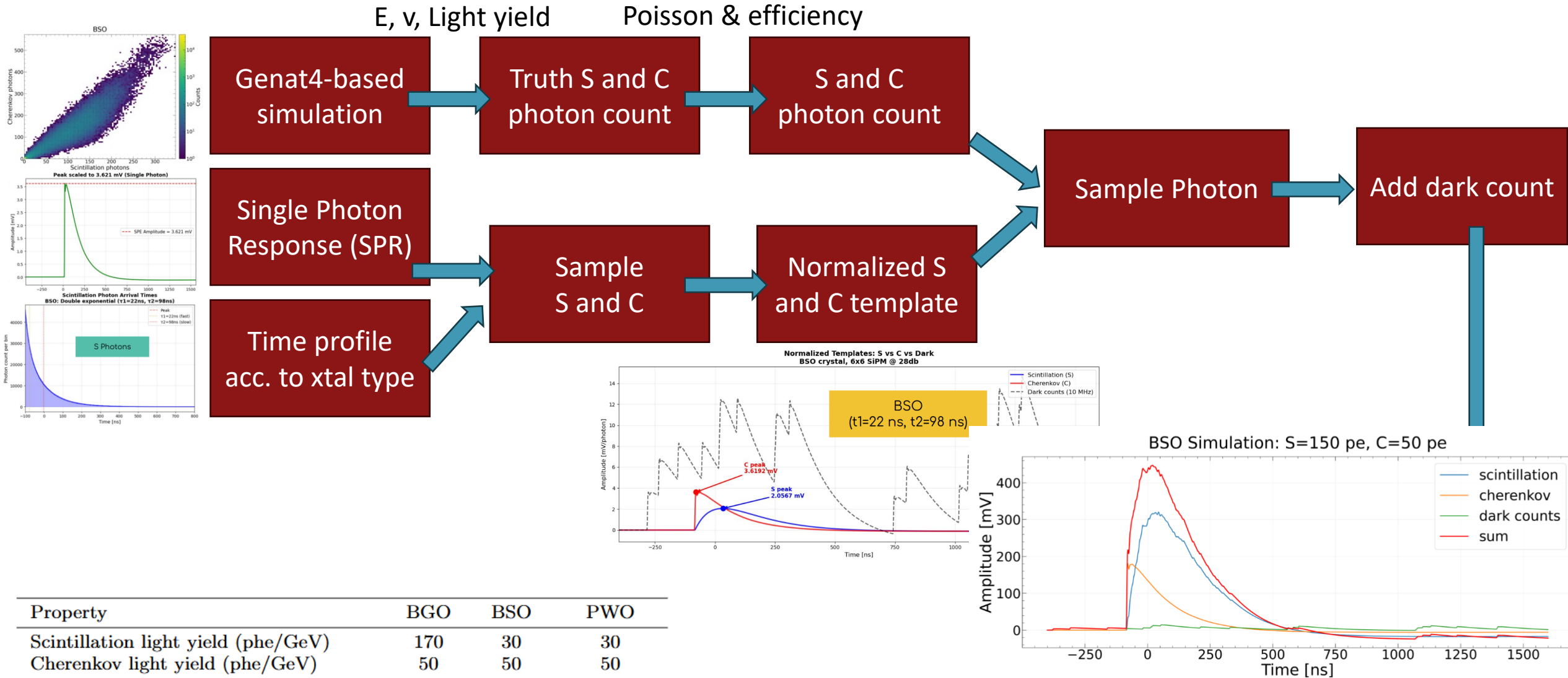
Hardware Implementation



eFPGA: System Integration

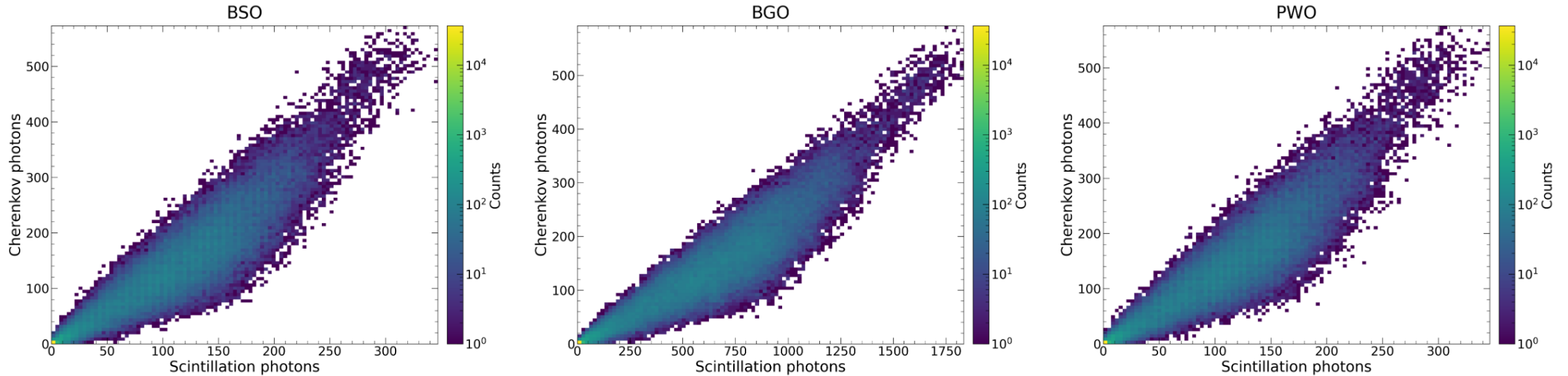


Digitization/Waveform Simulation



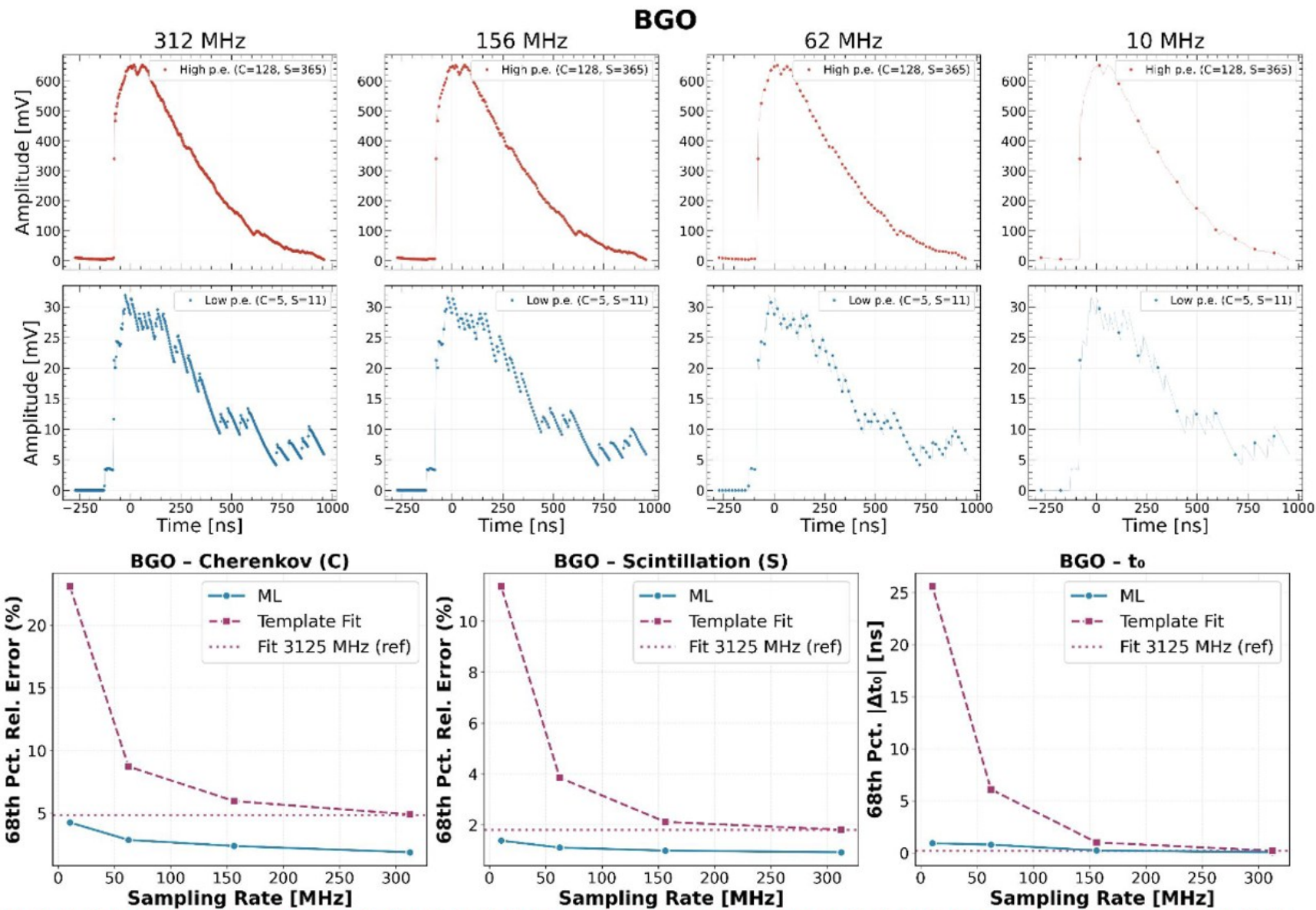
Property	BGO	BSO	PWO
Scintillation light yield (phe/GeV)	170	30	30
Cherenkov light yield (phe/GeV)	50	50	50
Decay model	Bi-exp	Bi-exp	Single-exp
Fast decay constant τ_1 (ns)	50	22	7.5
Slow decay constant τ_2 (ns)	320	98	—

Digitization/Waveform Simulation

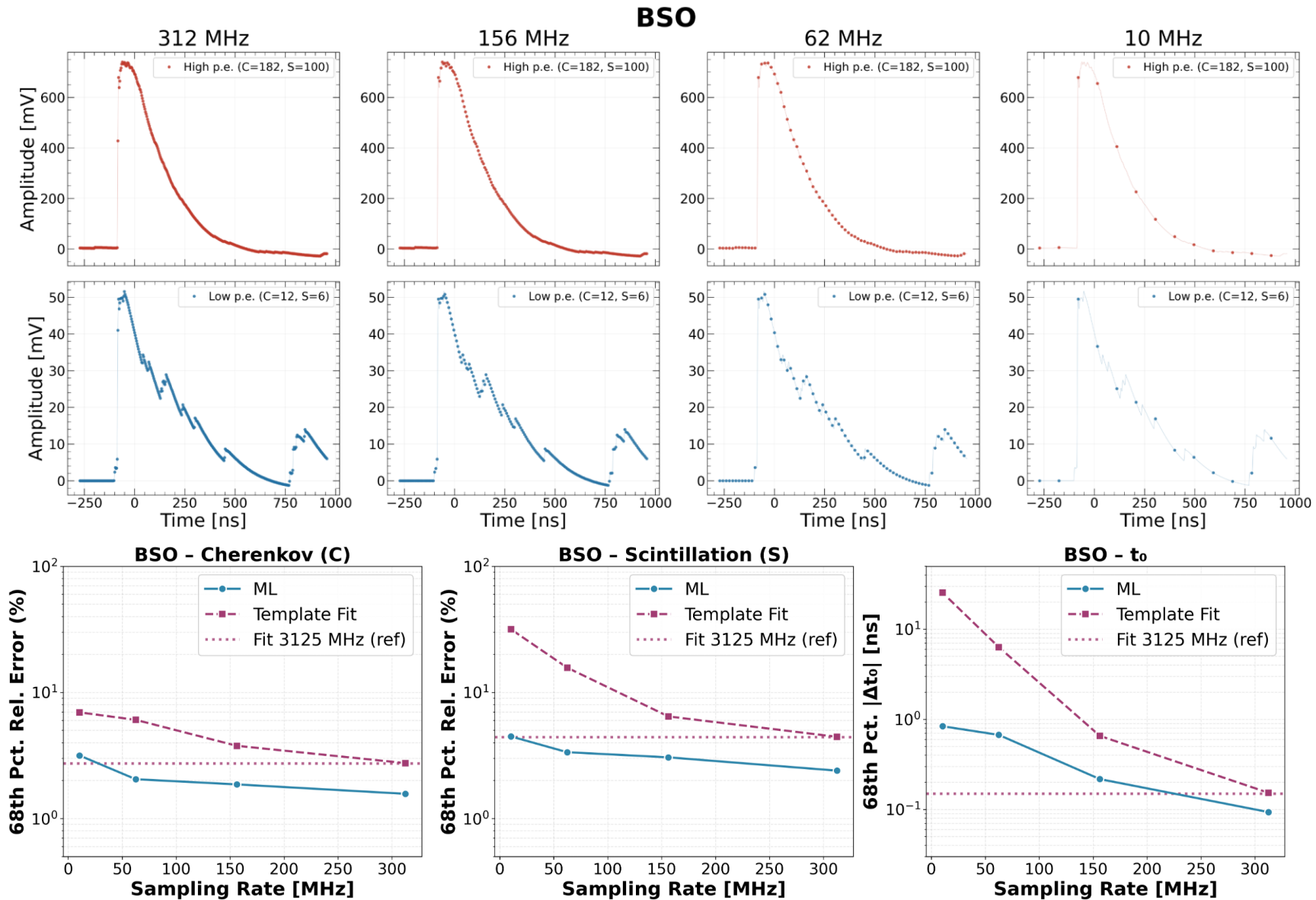


Property	BGO	BSO	PWO
Scintillation light yield (phe/GeV)	170	30	30
Cherenkov light yield (phe/GeV)	50	50	50
Decay model	Bi-exp	Bi-exp	Single-exp
Fast decay constant τ_1 (ns)	50	22	7.5
Slow decay constant τ_2 (ns)	320	98	—

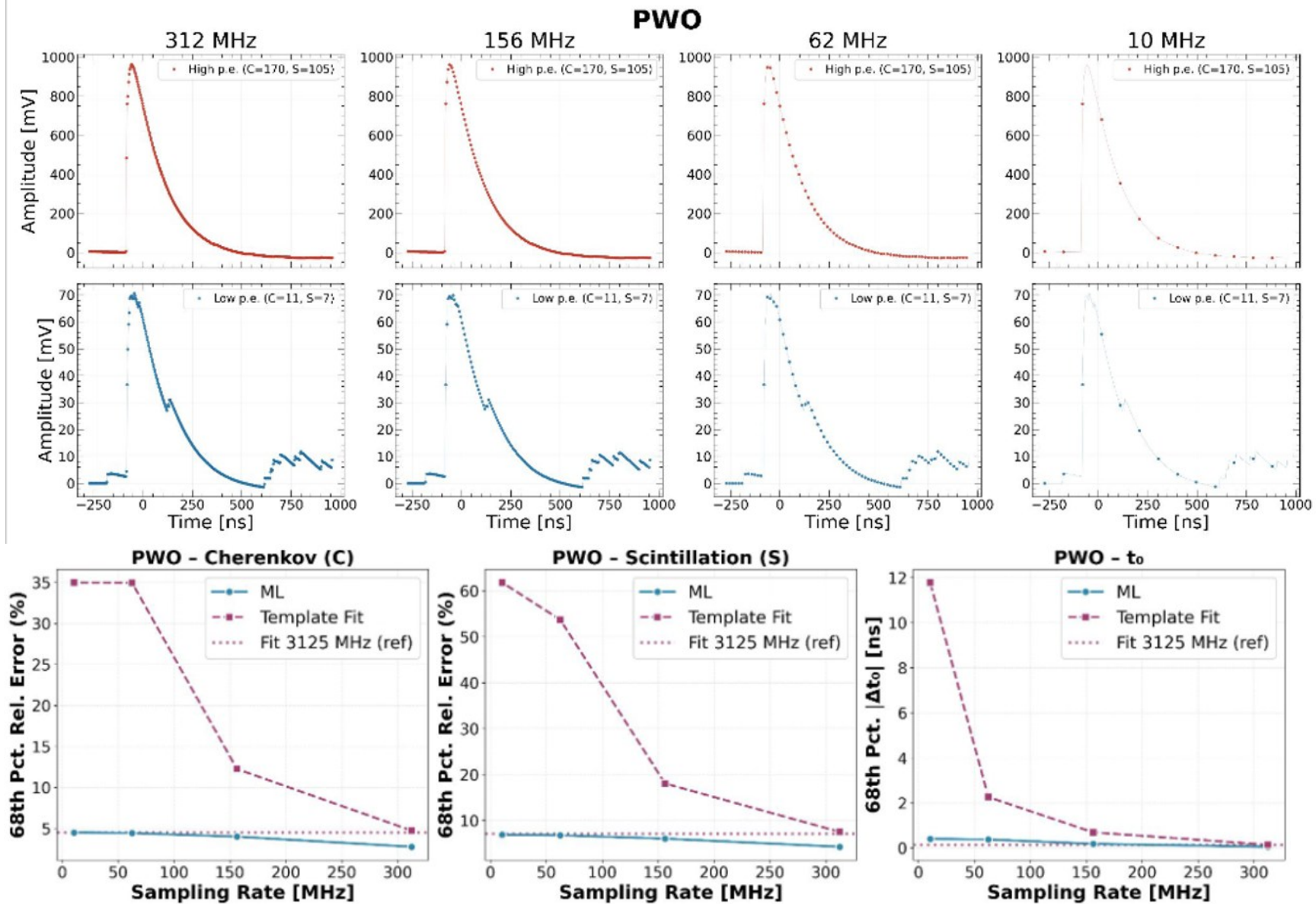
Sampling Rate and Performance



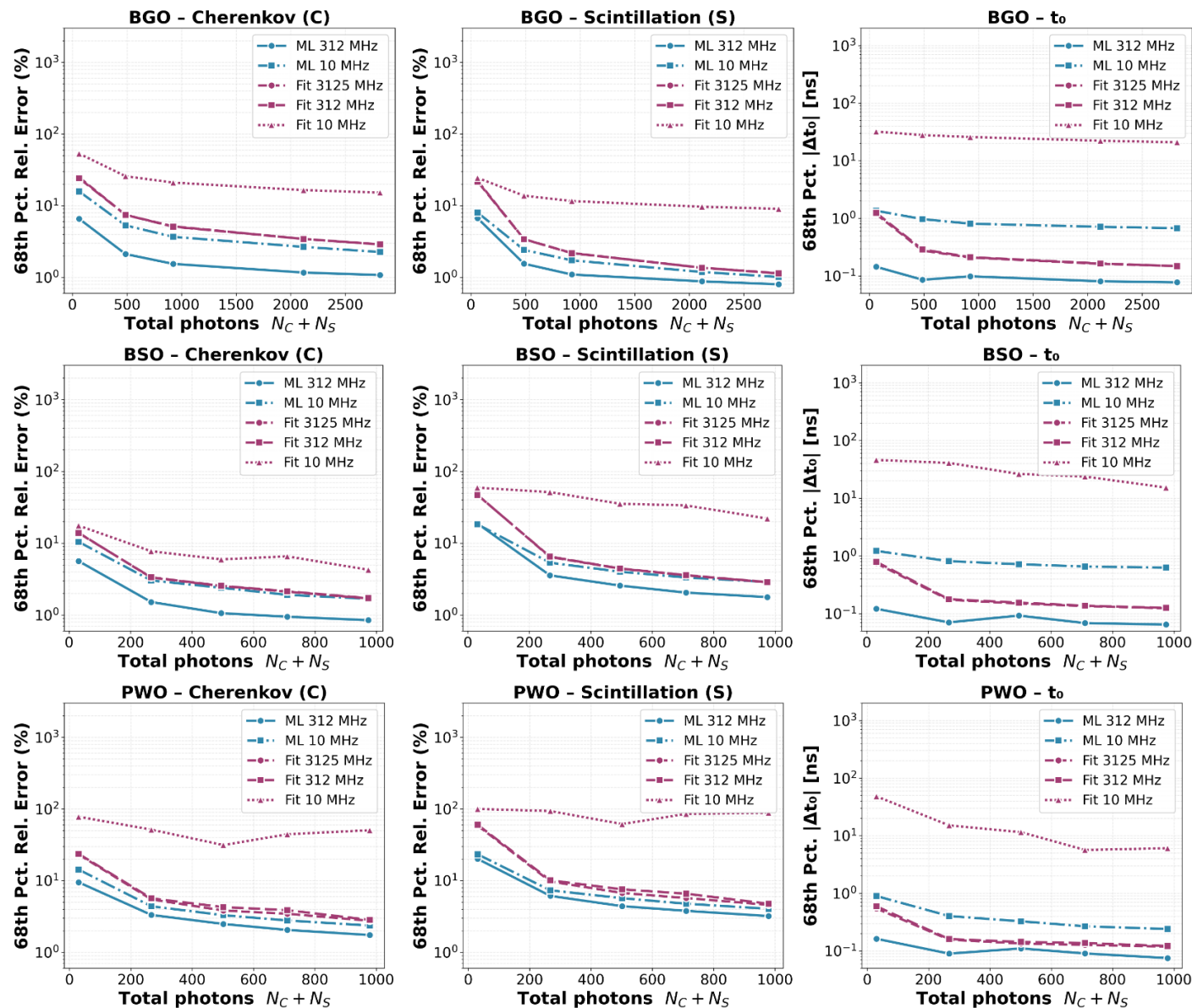
Sampling Rate and Performance



Down-sampled Sampling Rate and Performance

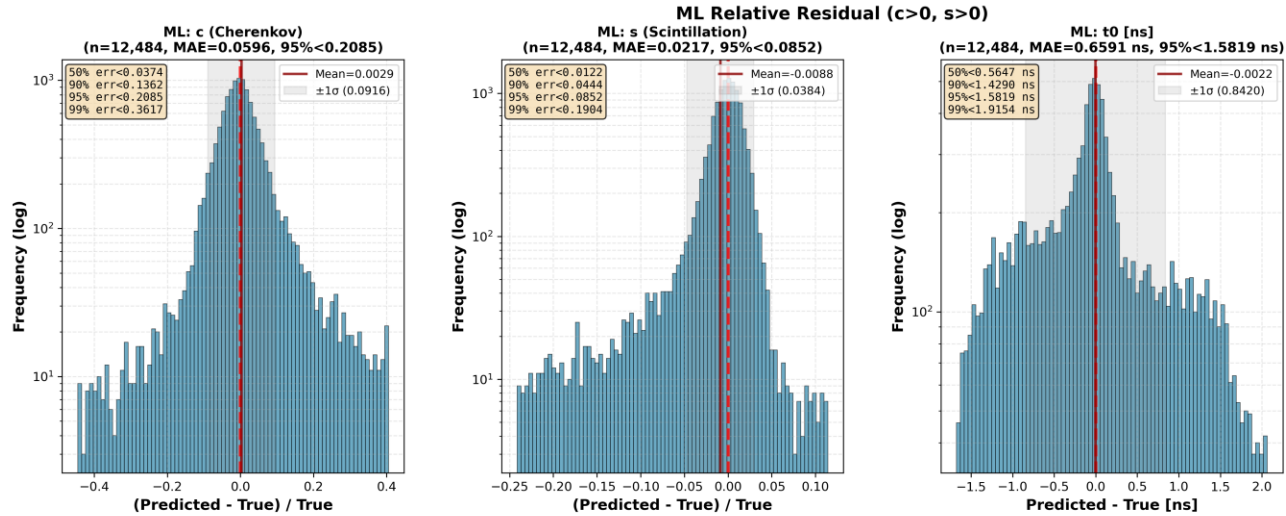


Performance Summary of ML-based DRO Algorithm



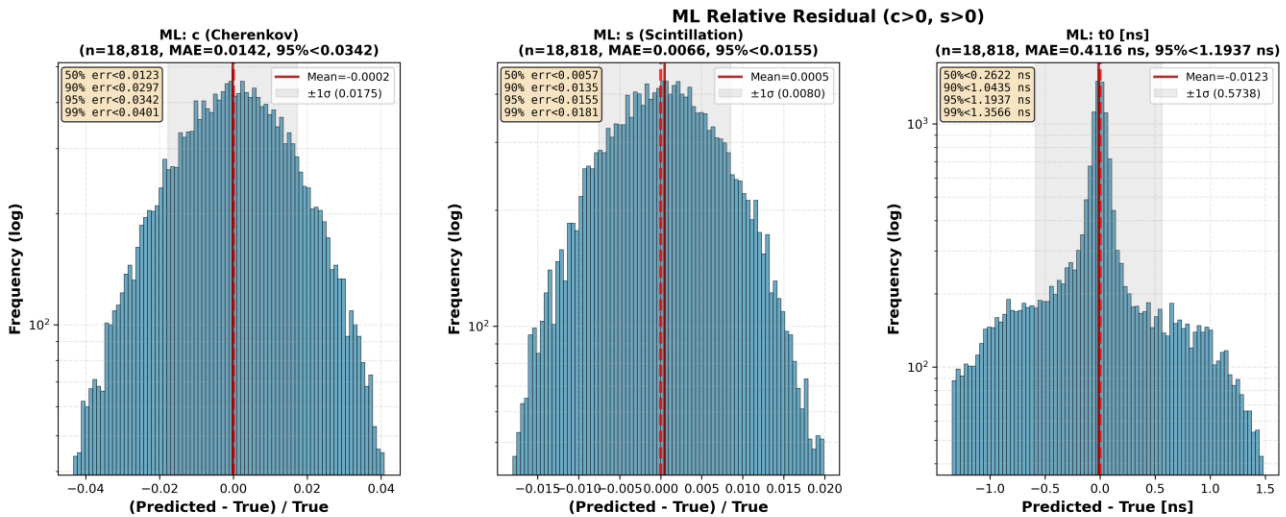
Electron/Kaon separated evaluation

Kaon



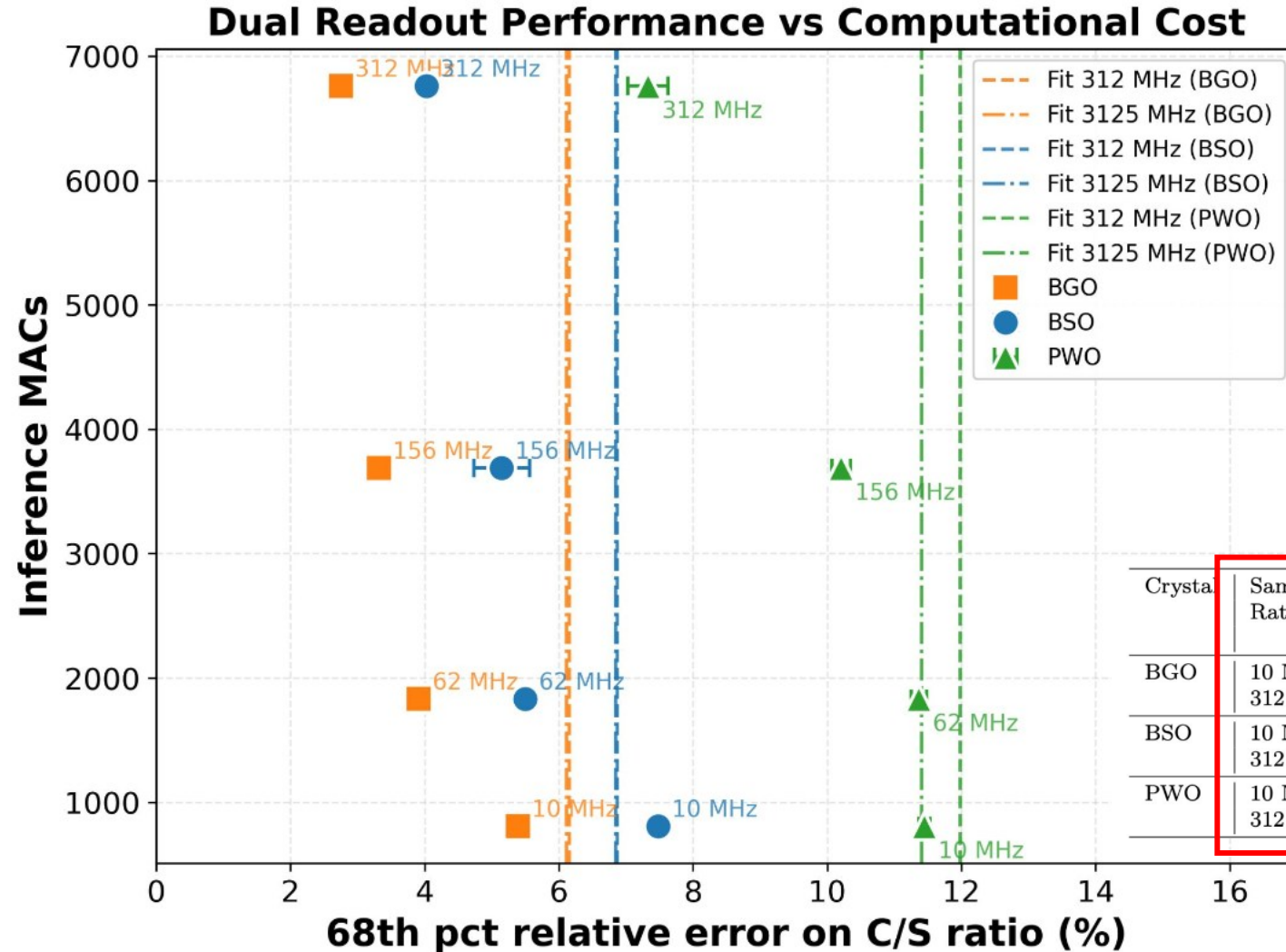
ML Statistics	
Filter: c>0, s>0	
Events: 13,142/20,000	
c (Cherenkov):	
Events:	12,484 (clip 658)
Mean:	0.002884
Std:	0.091633
MAE:	0.059624
50% err:	0.037365
90% err:	0.136175
95% err:	0.208521
99% err:	0.361734
s (Scintillation):	
Events:	12,484 (clip 658)
Mean:	-0.008797
Std:	0.038372
MAE:	0.021747
50% err:	0.012167
90% err:	0.044375
95% err:	0.085177
99% err:	0.190416
t0 [ns]:	
Events:	12,484 (clip 658)
Mean:	-0.002170
Std:	0.842015
MAE:	0.659111
50% err:	0.564746
90% err:	1.429998
95% err:	1.581906
99% err:	1.915381

Electron



ML Statistics	
Filter: c>0, s>0	
Events: 19,810/20,000	
c (Cherenkov):	
Events:	18,818 (clip 992)
Mean:	-0.000193
Std:	0.017545
MAE:	0.014238
50% err:	0.012309
90% err:	0.029664
95% err:	0.034232
99% err:	0.040109
s (Scintillation):	
Events:	18,818 (clip 992)
Mean:	0.000495
Std:	0.008040
MAE:	0.006574
50% err:	0.005738
90% err:	0.013516
95% err:	0.015508
99% err:	0.018104
t0 [ns]:	
Events:	18,818 (clip 992)
Mean:	-0.012294
Std:	0.573823
MAE:	0.411621
50% err:	0.262182
90% err:	1.043543
95% err:	1.193723
99% err:	1.356556

Resources Utilization (HLS)



ML-based method outperform offline fitting

- Achieved with much slower sampling rate
- Low latency, low power and less requirement for ADC

Small model enables the promising online deployment

- 6% error at 30K LUTs and no DSP
- Purely combinational and simple for integration

Crystal	Sampling Rate	Compression Strategy	Est. Freq [MHz]	LUT	FF	DSP	ML err68	Fit err68 @3GHz
BGO	10 MHz	40% sparsity, $\langle 10, 1 \rangle$ QAT	57.5	36,879	213	0	5.84%	6.12%
	312 MHz	30% sparsity, $\langle 10, 2 \rangle$ QAT	46.7	312,914	3,933	0	3.08%	6.12%
BSO	10 MHz	50% sparsity, $\langle 10, 2 \rangle$ QAT	57.7	30,471	213	0	8.14%	6.85%
	312 MHz	30% sparsity, $\langle 10, 4 \rangle$ QAT	46.6	328,953	3,933	0	4.18%	6.85%
PWO	10 MHz	50% sparsity, $\langle 10, 2 \rangle$ QAT	57.7	30,321	213	0	11.47%	11.41%
	312 MHz	30% sparsity, $\langle 10, 4 \rangle$ QAT	46.6	311,282	3,933	0	7.40%	11.41%