On-Sensor Machine Learning for Future Pixel Detectors: Regression under a Realistic Frontend Response

Harshul Gupta, on behalf of the smartpixels consortium
University of Illinois Chicago

CPAD 2026

21 October 2026

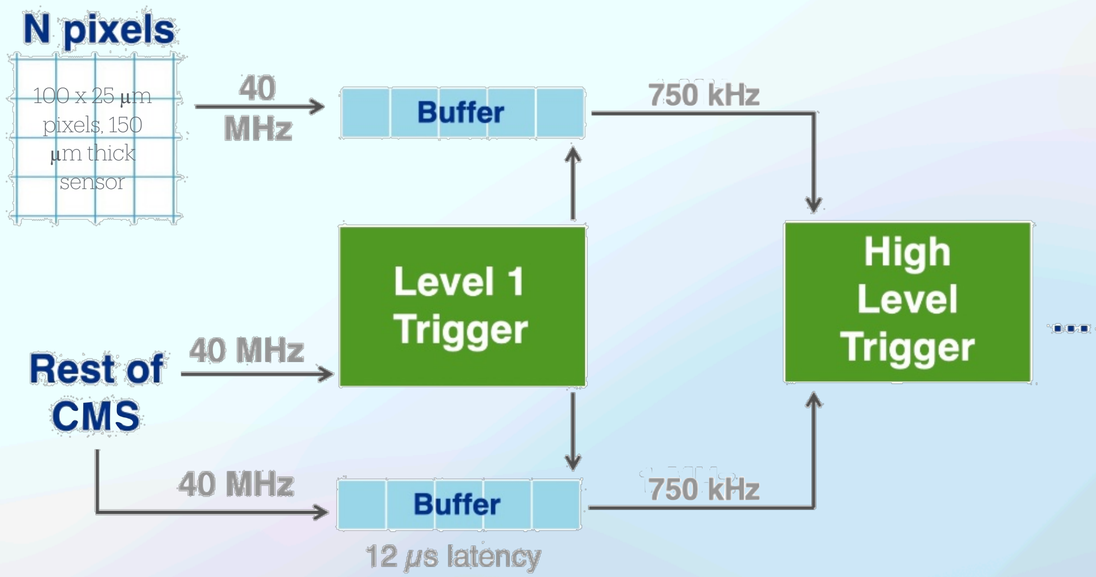
2026 Meenakshi Narain Graduate Scholar at the LPC, Fermilab

FERMILAB-SLIDES-26-0238-CMS-CSAID-ETD

Motivation

Ever Increasing Data Rates

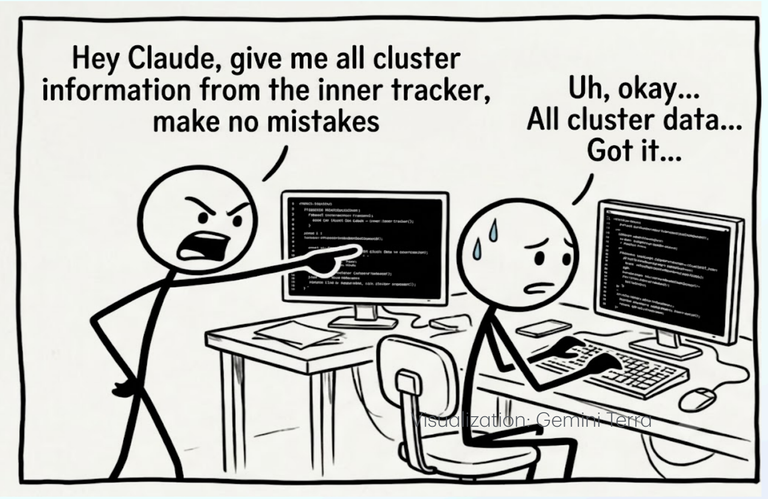
- At the HL-LHC, collisions happen at 40 MHz → **far too fast** for any detector to read out and store every event.
- The **CMS's Level-1 (L1) trigger** makes a **real-time keep/discard decision** on each collision within microseconds before the data leaves the detector.
- In CMS, **L1 is implemented in custom hardware** (FPGAs/ASICs) with a **fixed ~12 µs latency budget**, reducing the 40 MHz collision rate to a 750 kHz output rate for further (High-Level Trigger) processing.
- The inner tracker (**closest to the beam, highest data rates**) generates more data than the L1 pipeline can carry → so today it's **excluded from the L1 decision** entirely, and that physics that requires the pixel detector in order to be found is lost forever.

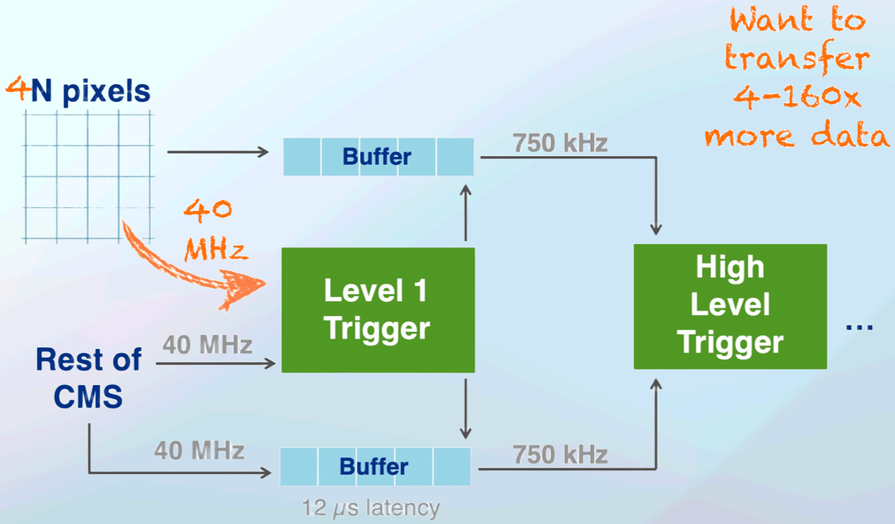


Smartpixels

Futuristic Pixel Detector: AI on Chip

- An innovative R&D concept! On-ASIC ML for data reduction!
- Identify & analyze local charge clusters in real time → Decide which ones to use in the L1 trigger.
- It opens up *new physics searches* by enabling access to **displaced tracking** and **exotic signatures** currently invisible to the L1 trigger.
- So we **retain more events** → enhance precision measurements analyses like **Higgs** and **di-Higgs**, **flavor tagging**, and **B-physics** by bringing their most discriminating information into the L1 trigger.
- Can be used at the **Muon Collider** and **FCCee** to *reduce data rates from beam backgrounds*.
- Our goal is to do a high-quality analysis of every collision at 40 MHz by adding pixels to the Level 1 Trigger.





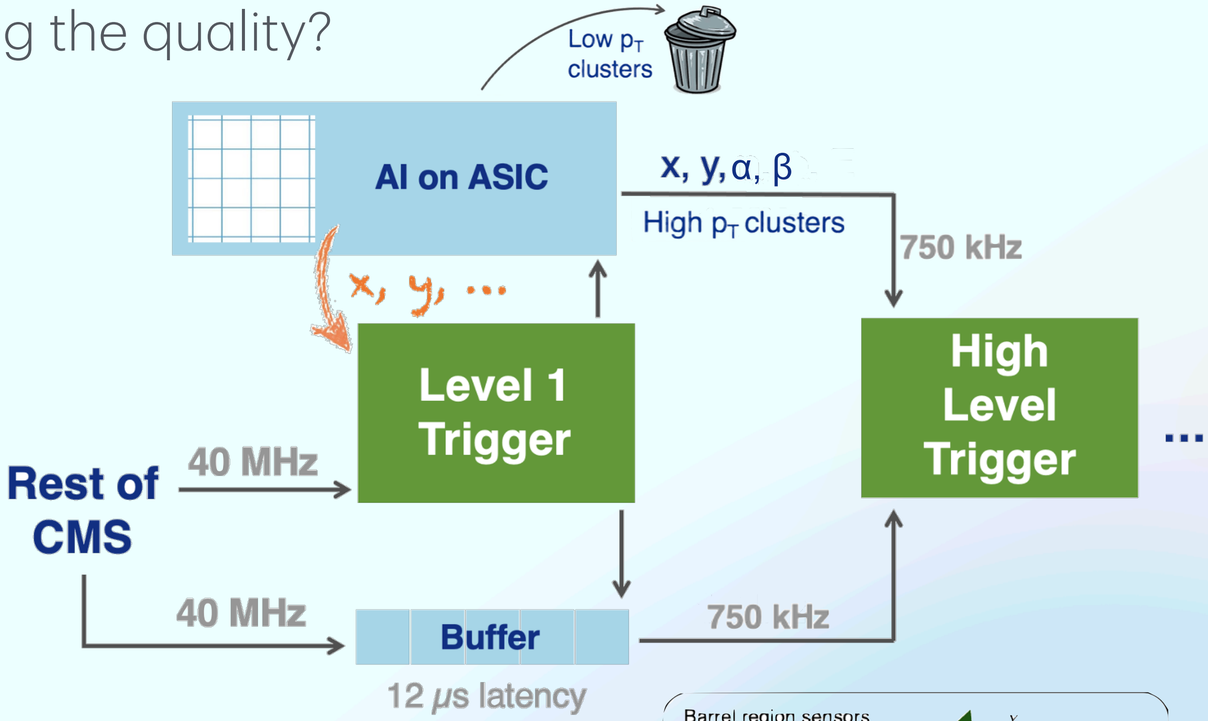
Too Much Data!!—Plan of Action

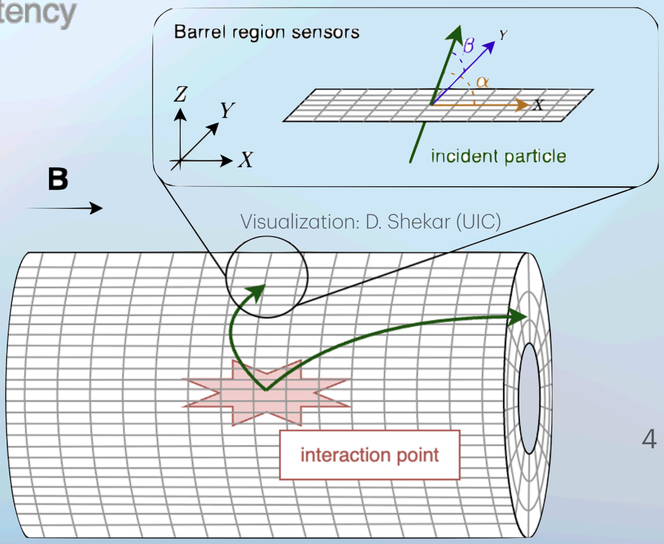
How to reduce the data read out while still retaining the quality?

- Faster data links, or
- Read out less data: Use a 2-bit ADC
  - Put an *ML On-ASIC* for data reduction:

1. **Filtering:** Train an algorithm to identify clusters that come from interesting tracks, e.g., high transverse momentum (> 0.2 GeV).

2. **Compression/Regression:** Train an algorithm to extract properties of the incident particle. Read this out instead of raw data. Enough reduction to send (some) features to Level 1! (>10x compression).

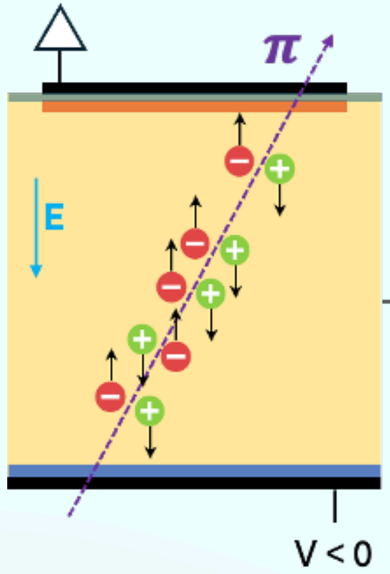




This Talk

- We have strong motivation for including **pixel information in L1 tracking**.
- Smartpixels aims to **regress track parameters on chip** and **filter out low-momentum tracks** using NNs.
- Promising results so far, but they are predicated on the **charge** directly **measured by the sensor** being exactly what is **fed to the NNs** (after digitization). In reality, the *analog frontend* processes that charge as it reads it out.
- Realistic effects considered here:
  - **Voltage shaping** by the amplifier
  - Realistic **noise** addition
  - ADC comparators' **Time-walk**
- Does the smartpixels algorithm's performance survive this realism?

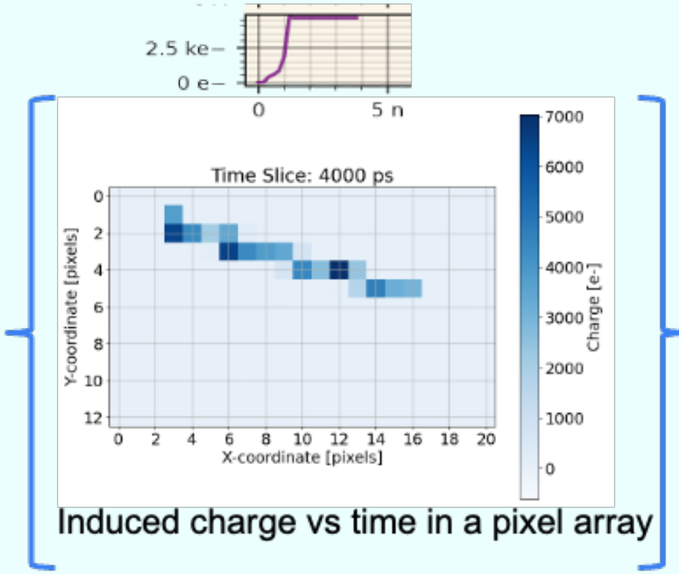
Algorithm Training Pipeline



Sensor information from TCAD

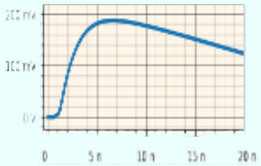
Incident particle tracks

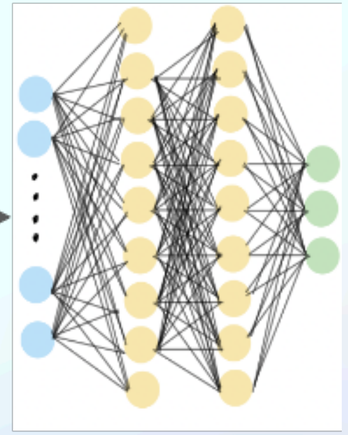
PixelAV



Induced charge vs time in a pixel array

Frontend Response





Neural network

<u>16x16xN input values</u>
Collected charge values from all 16x16 pixels at all N time steps

Regression CNN

<u>14 output values</u>
Incident positions (x,y), angles (α, β), for 2 time slices and uncertainties on all predictions!

Raw pixel data stays on-chip — only physics quantities are read out

On-sensor inference: cluster in, track parameters out.

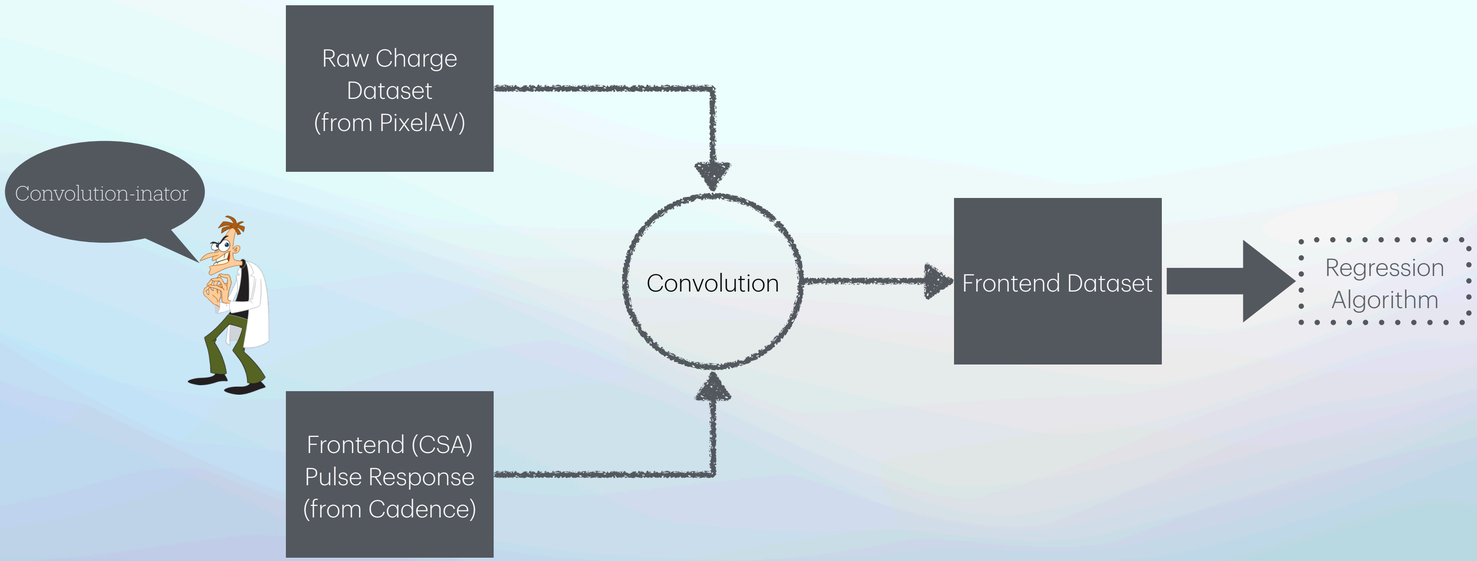
Simulated MIP interactions for this futuristic pixel detector (modeled for Phase 3 CMS, but the results are generic).

- **Located at a radius of 30 mm**
- **In a 3.8 T magnetic field**
- **For 50 x 12.5 micron pixel sensors, 100 μm thickness**
- **At time steps of 10 ps and 200 ps**
- **A flat-pT distribution with cot α ∈ [-10, 10]; cot β ∈ [-2, 2].**
- **300 K and 243K temperatures**
- **Over a pixel readout window (e.g., 16 × 16, 13 × 21)**

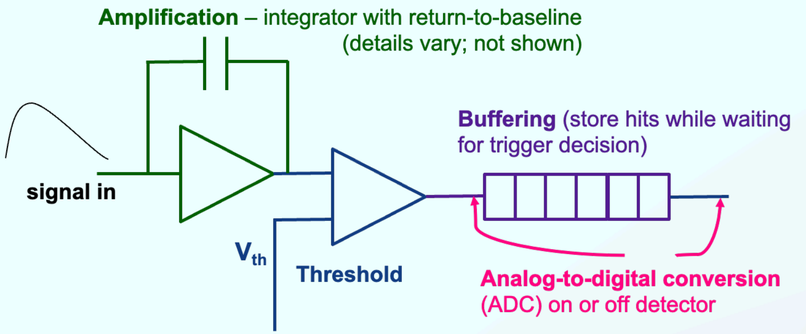
How is the frontend simulated?

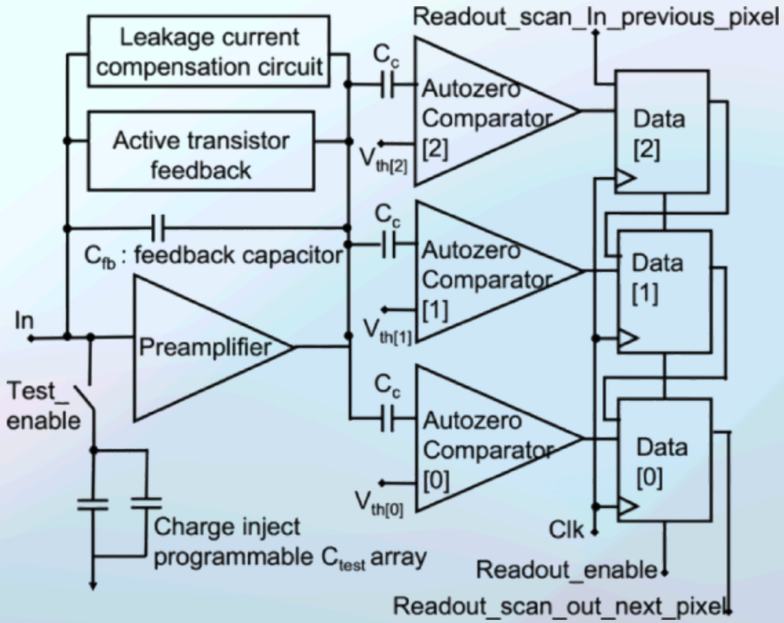
Effect of Readout Electronics

- Our Engineering team implemented the Analog Frontend in the *Cadence Circuit Design*.
- Extracted **pulse response of the CSA** (Charge Sensitive Amplifier) stage.
- So we feed our raw pixel charge to the CSA to get voltages, which are then passed to the ADC (whose comparator delays are added using an LUT) and finally fed to the regression network.
- In this talk, the final dataset is referred to as the "*Convolutional Dataset*" or the "**Frontend (or Post-Processed) Dataset**".



Visualization: C. Mills (UIC)

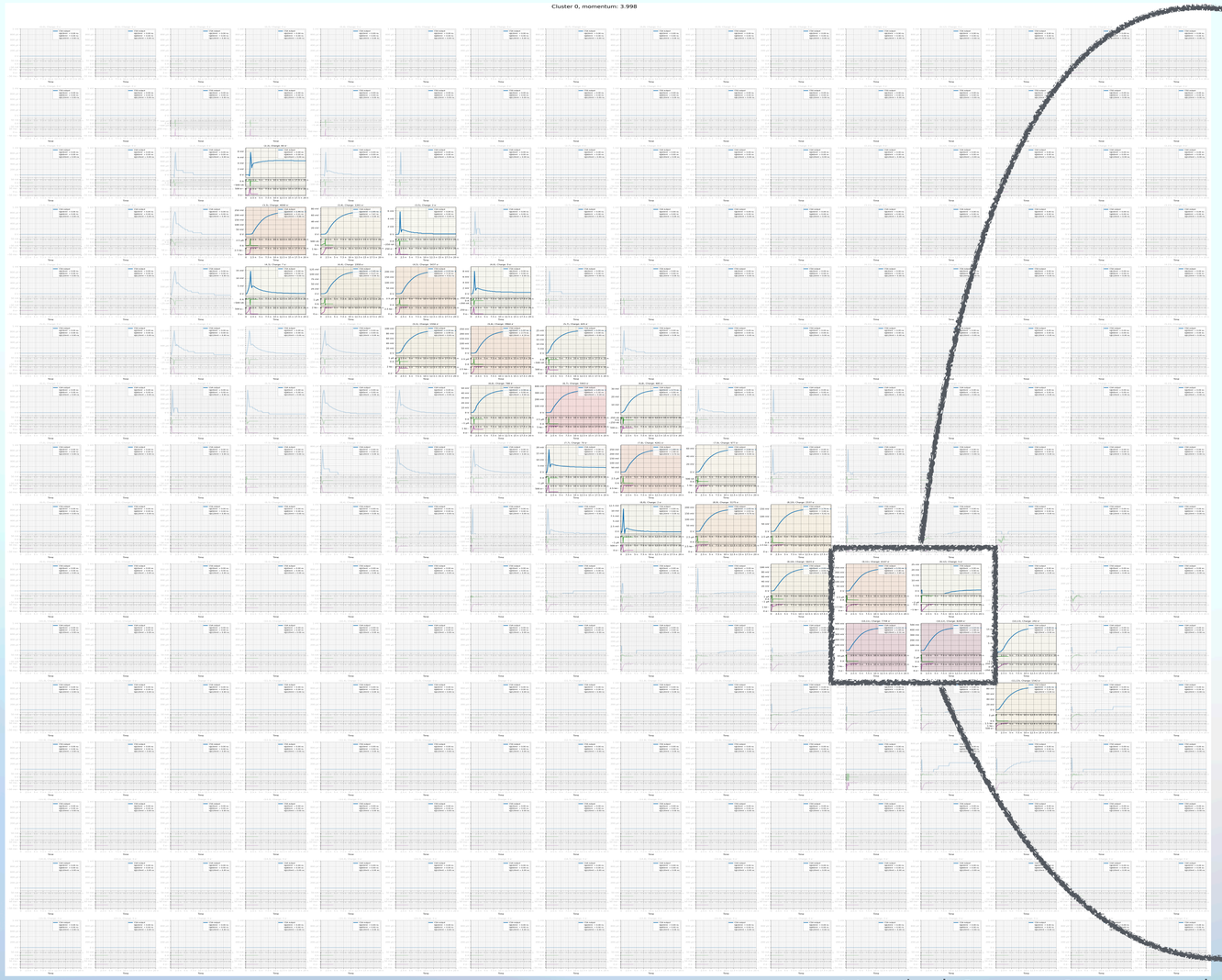


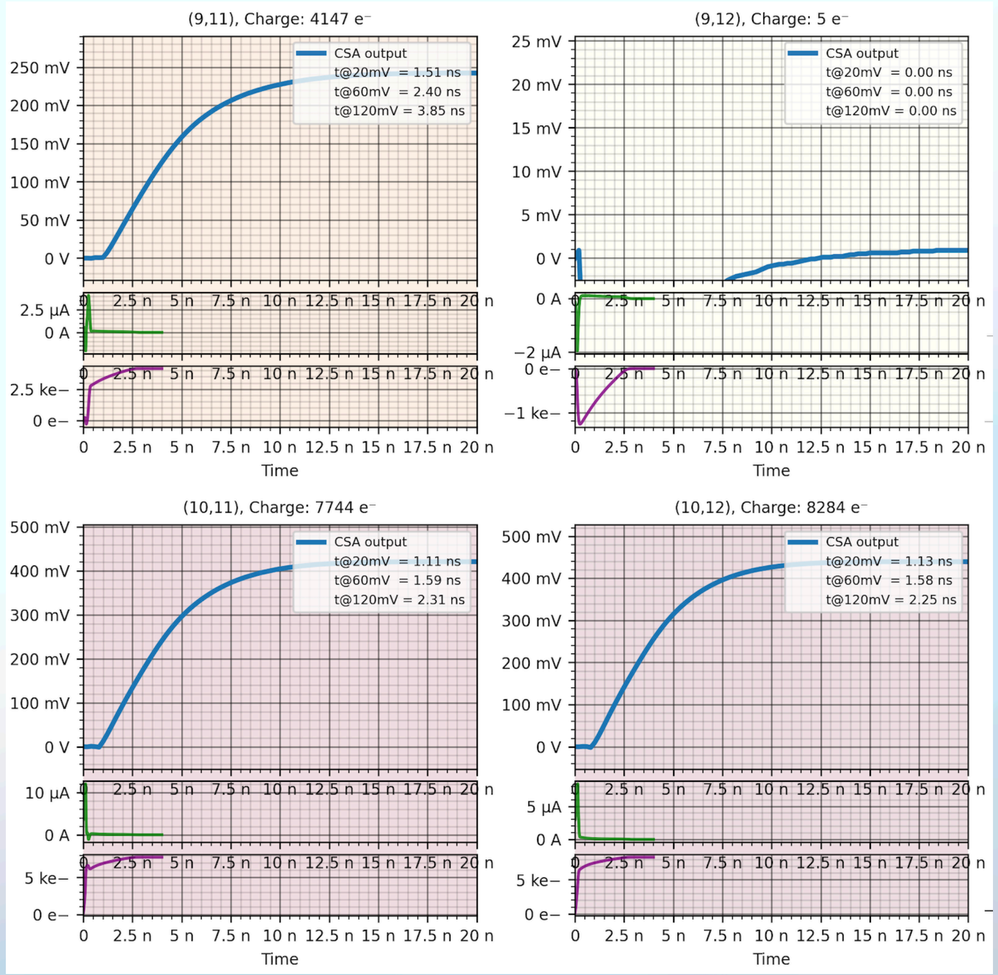


AFE, Source: B. Parpillon et. al.

Post-processed Dataset

Example event from post-processed dataset: Charge, Current, and Voltage waveforms vs Time"





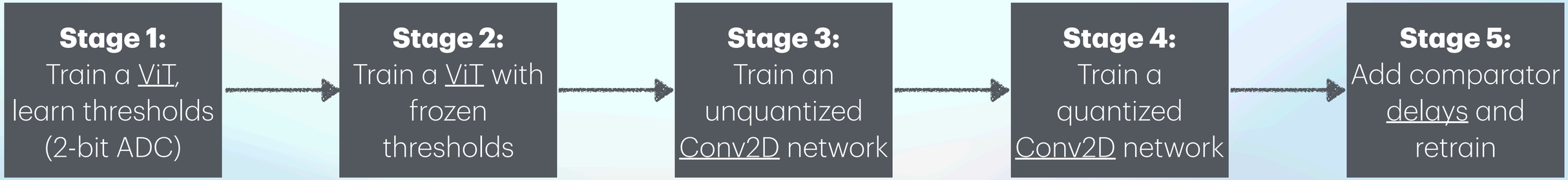
Harshul Gupta, Smartpixels Consortium

Visualization: H. Gupta (UIC), M. Ronchi (Polimi)

Training Setup & Constraints

From Threshold Search to Chip-Deployable Model

- **2 time slices** used per event, sampled before the pulse peak → captures the rising edge without waiting for the full pulse.
- A **time-correlated noise** model is applied across pixels rather than independent per-pixel noise to achieve a more realistic detector simulation.
- **Contained-cluster selection**: only *tracks* **fully within the pixel readout window** are used, discarding edge-truncated clusters that have an *incorrect correlation between truth label and cluster shape*.

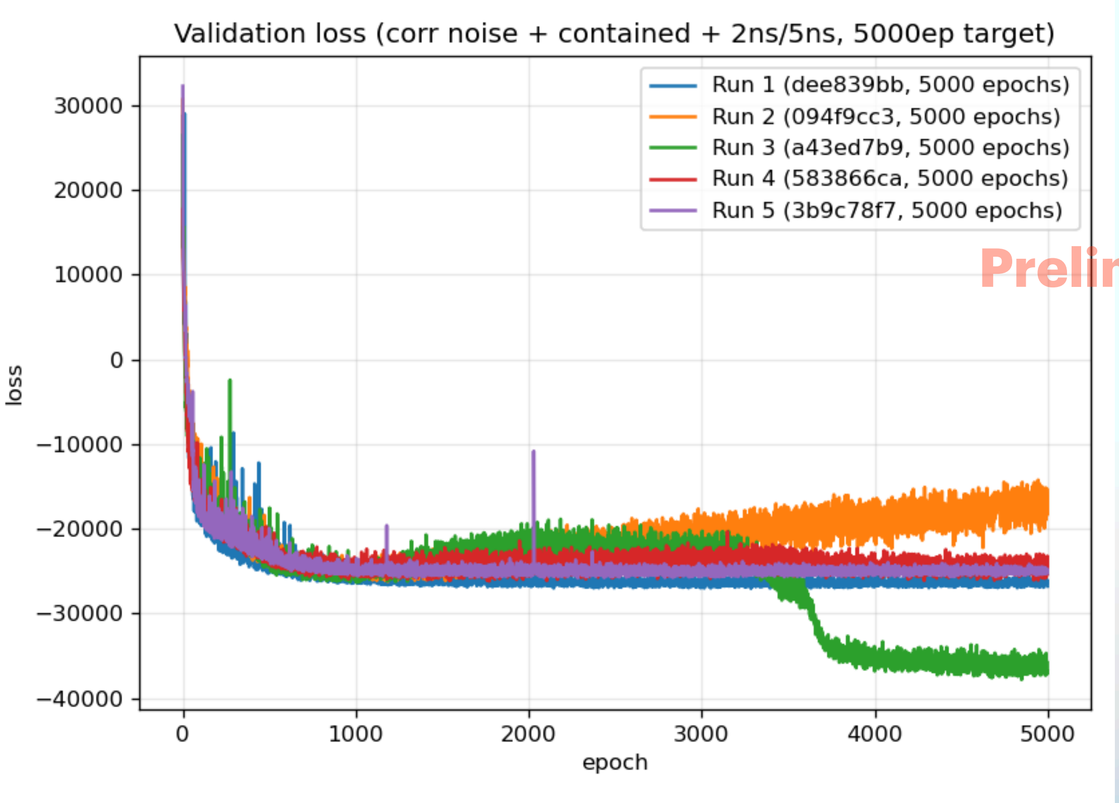


*ViT: Vision Transformer; Conv2D: Convolutional 2D Network

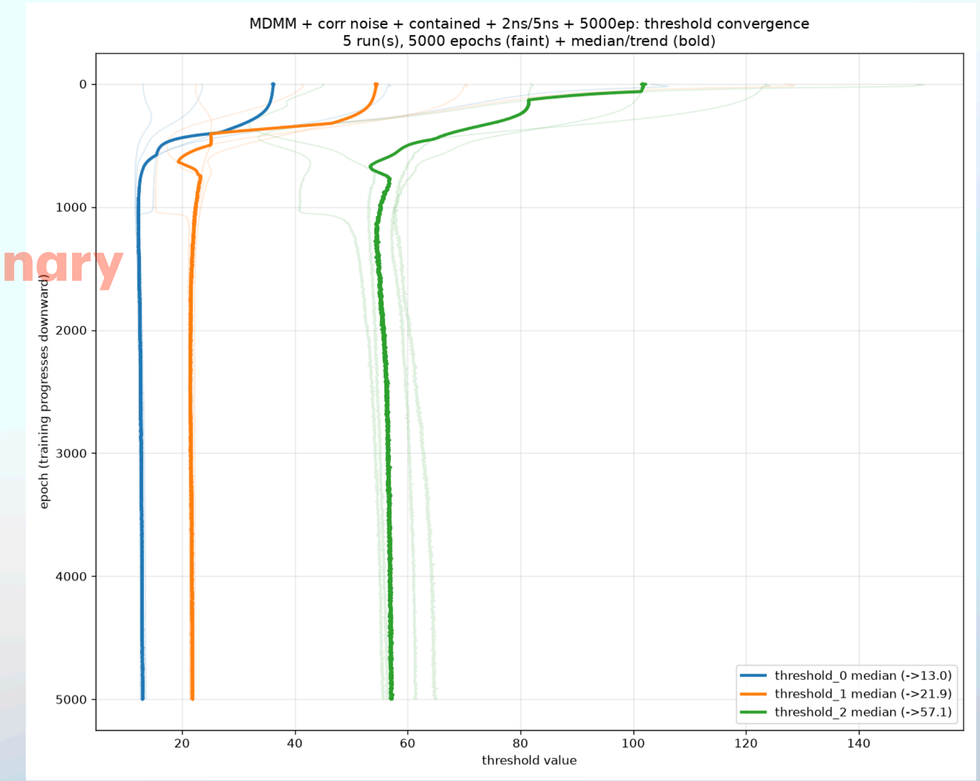
Each stage adds exactly one more real-world constraint on top of the previous one, so the accuracy drop at each step can be attributed to a single specific cause rather than several tangled together.

Stage 1: Training Transformer Model Searching Thresholds

- 2 and 5 ns time slices, 5 runs of 5000 epochs each, random initial thresholds.
- **Thresholds are learned**, not chosen: a soft, differentiable stand-in for the ADC (**SoftDigitization**) lets gradients reach the threshold values, annealed toward hard digitization as training proceeds.
- Thresholds **comparable** to those found in the **raw charge dataset** cases (with 58 μV/e gain conversion factor).
- Correlated Noise added; NLL Loss performance stuck at a local minima for most runs.

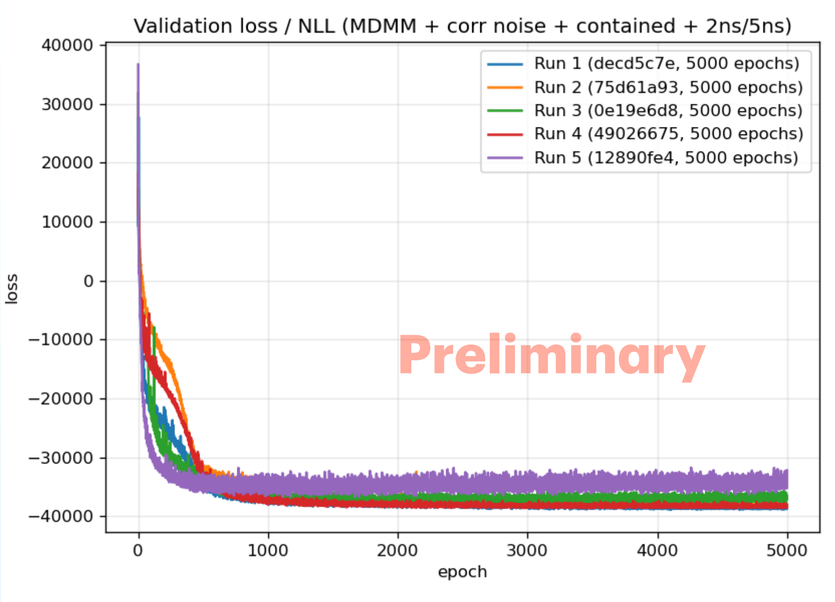


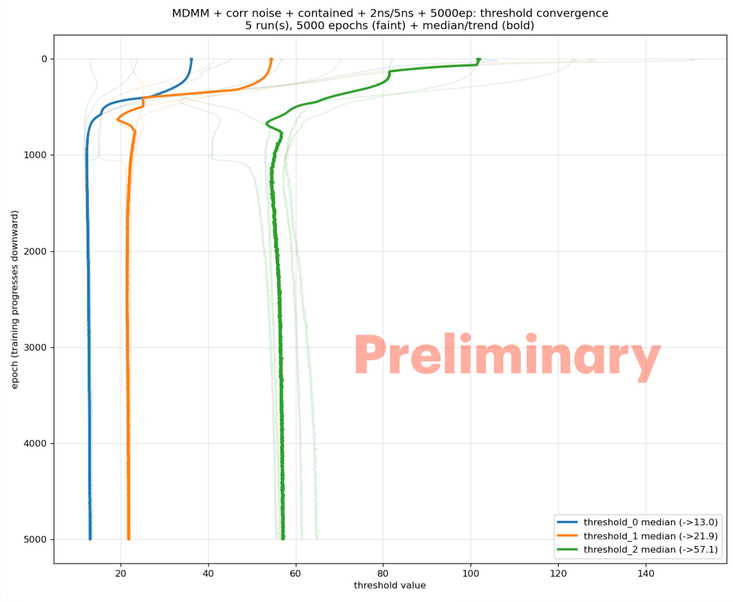
Preliminary

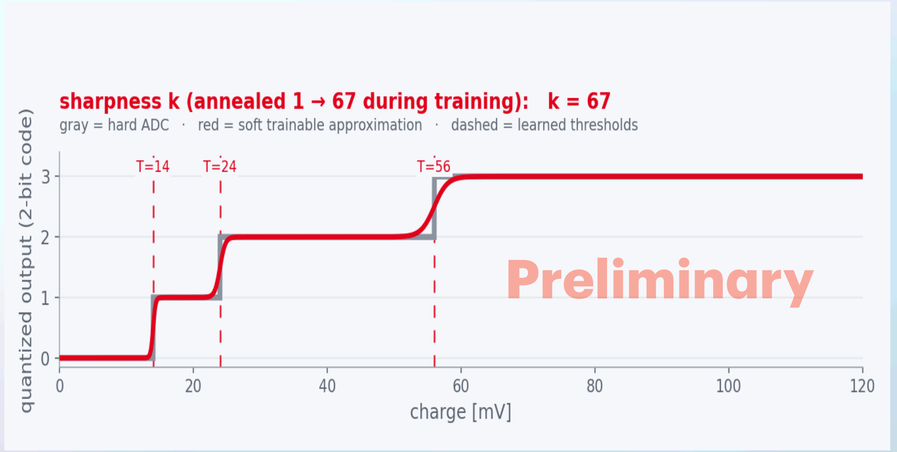


Truth Correlated MDMM

- Angle predictions collapse to a near-constant value → minimizes NLL loss without actually tracking the true cotα/cotβ signal → Getting out of Local minima via MDMM (Modified Differential Multiplier Method).
- MDMM adds a Lagrangian penalty enforcing a minimum correlation between predictions and truth → Self-tuned → Applied identically across every stage, from the full ViT down to the compact Conv2D.
- Training-time only: Lagrange multipliers are discarded after training → deployed architecture and weights are unaffected



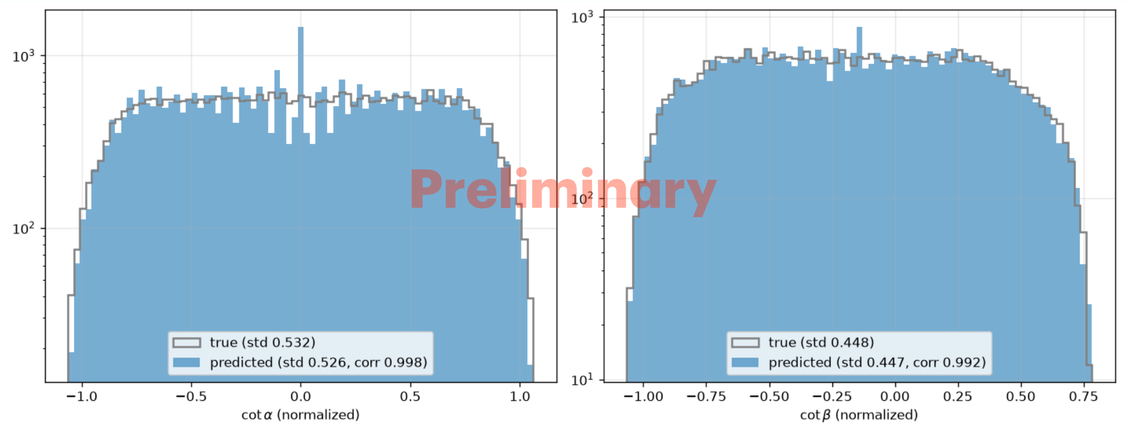




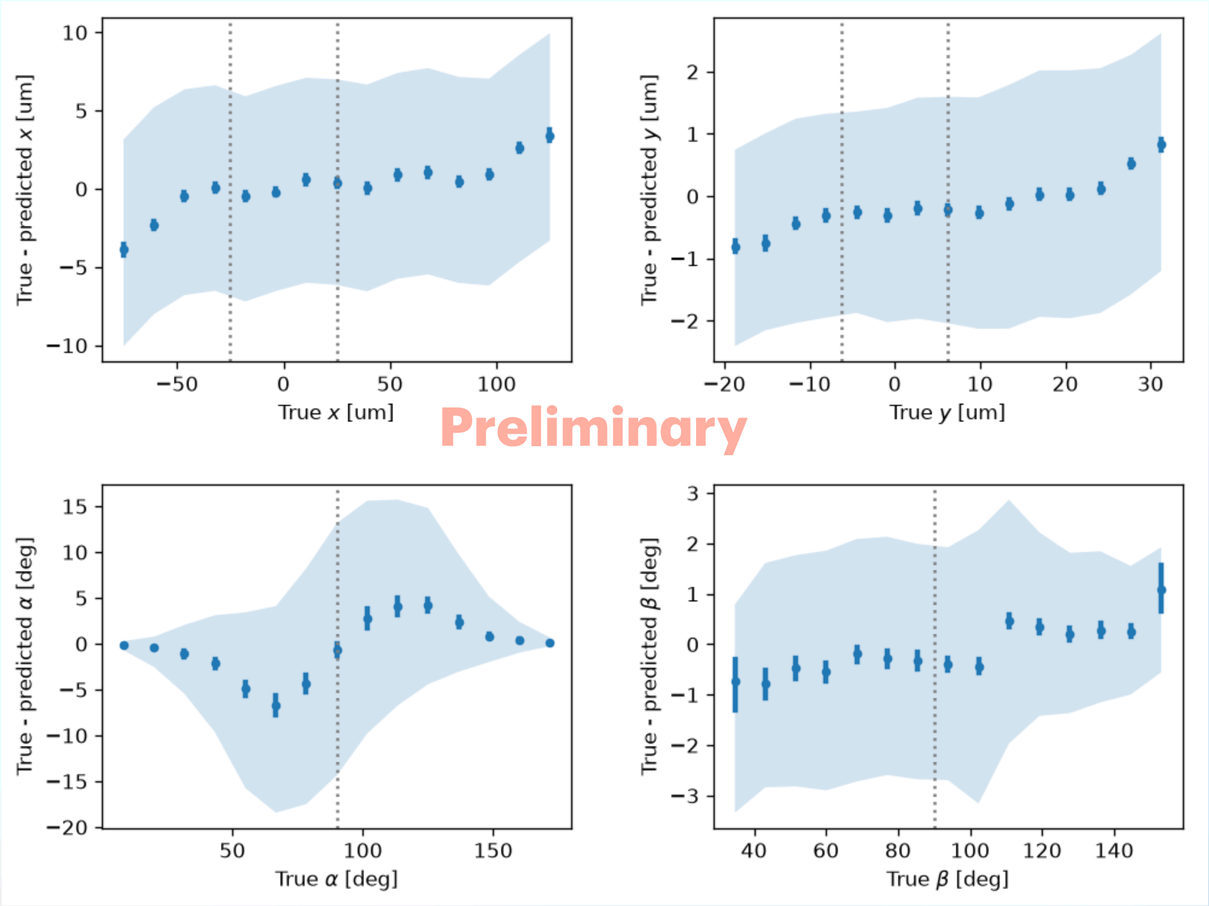
Stage 2: Transformer Model With Fixed Thresholds

- ViT is run again, this time with **frozen thresholds** (derived in stage 1), but **no noise**.
- NLL Loss around -36k.
- Good truth correlation, performance **persists**!

Predicted vs True Angle Dustributions: Stage 2—no-noise (ViT, MDMM), 2/5ns



Summary: Stage 2—no-noise (ViT, frozen hard-digitized thresholds, MDMM), 2/5ns



For a good performance here, we would like the residual spread to be minimal.

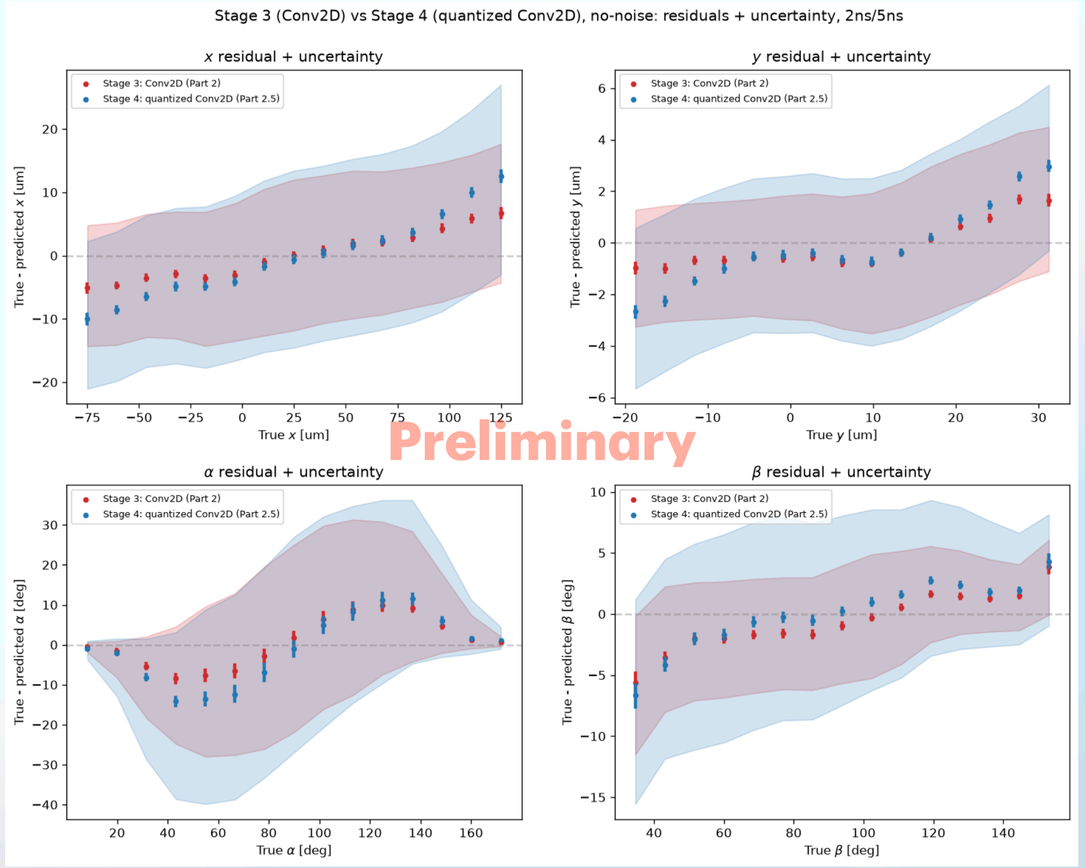
The **flatter the line** (distribution), the better the model predicts; a **slope** of **one** indicates *"no information"* is extracted.

The **band** is the **resolution**, so a **narrower band is better.**

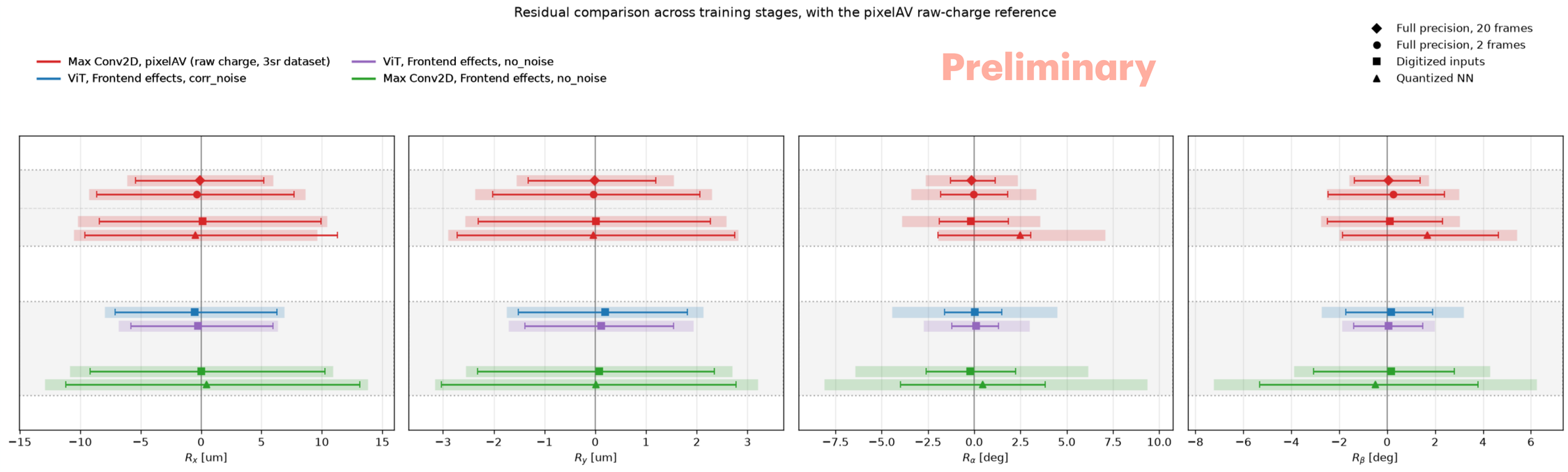
Stage 3 & 4: Compact Conv2D NN

Shrinking to a chip-sized architecture

- In stage 3, we have the same frozen thresholds, the same MDMM, and the same dataset → **swapped the full ViT for a compact Conv2D (1,869 params)** matching the target chip's architecture shape.
- Only one change—the architecture—so the drop in accuracy is attributable to model size alone.
- For stage 4, weights constrained to low precision (**quantized**), 4 bits in the convolutions and 8 in the dense head → adding further realism → closer to the chip-deployable version.
- Accuracy degrades, <u>monotonically</u>, **ViT → Conv2D → quantized Conv2D**, exactly as the staging intends.



What does Frontend Truly Cost?



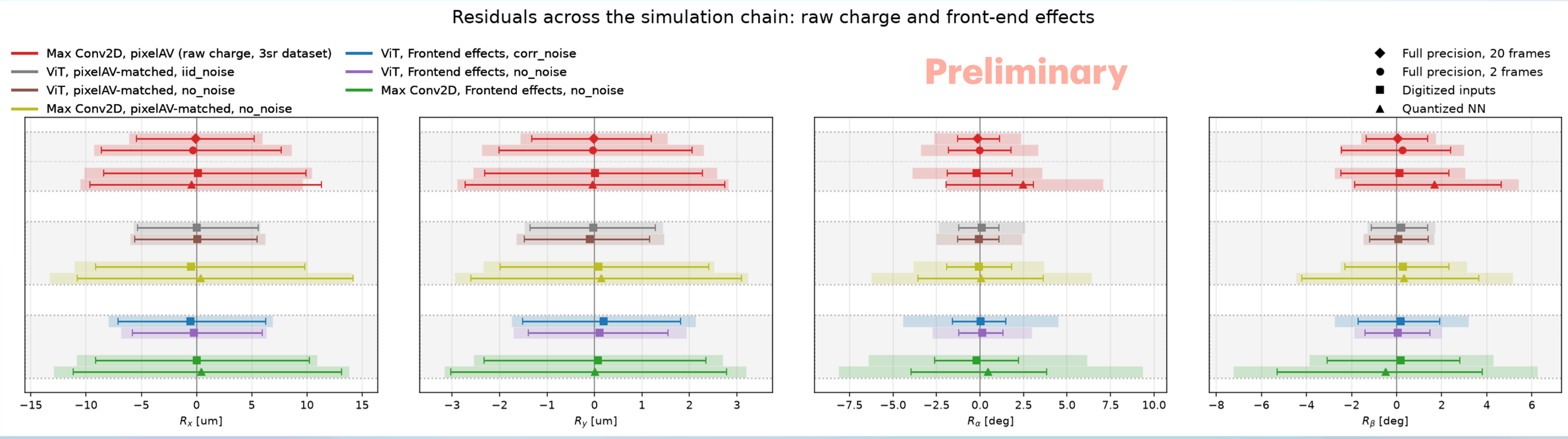
- Any gap in performance could have been the frontend or just the sample.
- **Raw-charge dataset** → Huge!! Had more ranges in angles | **Frontend dataset** → Small!! Had a restricted angle range.
- *Like-for-like control* → matched in everything except the frontend. Subsampled pixelAV datasets to match the statistics of the Frontend Dataset.
- Ran the identical four-stage pipeline on it, so each stage has a raw-charge twin.

Thin Statistics Test

The Cost of the Physical Frontend Lies Almost Entirely in Reduced Angular Resolution

On comparing the **frontend dataset results with stat-matched PixelAV results**:

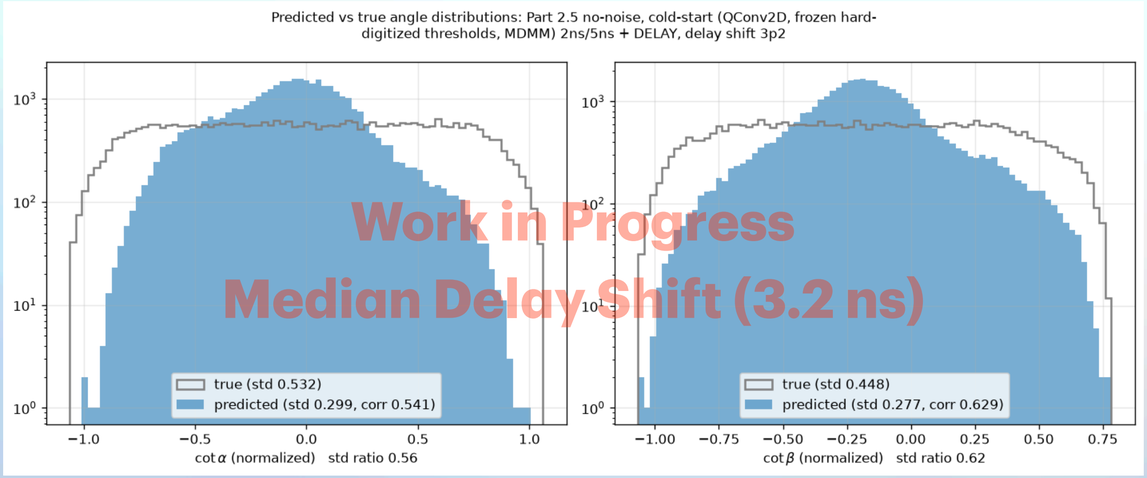
- Position resolution barely changes. **x** and **y** 1-σ residuals (half-width) agree to within a few % at every stage — **x**: 5.53 → 5.89, **y**: 1.32 → 1.47 μm at ViT; **x**: 9.49 → 9.69, **y**: 2.19 → 2.33 μm at Conv2D.
- Angle resolution degrades by about 30% at Conv2D. 1-σ half-widths: **α**: 1.87° → 2.42°, **β**: 2.31° → 2.94°.
- Statistics matched by construction, so the comparison is fair, and what remains is not due to the sample size.

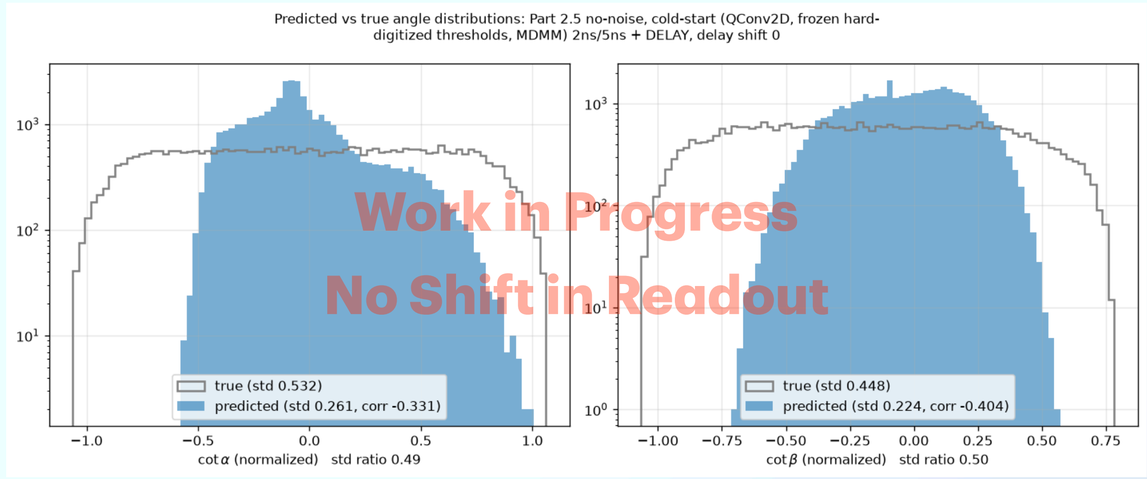


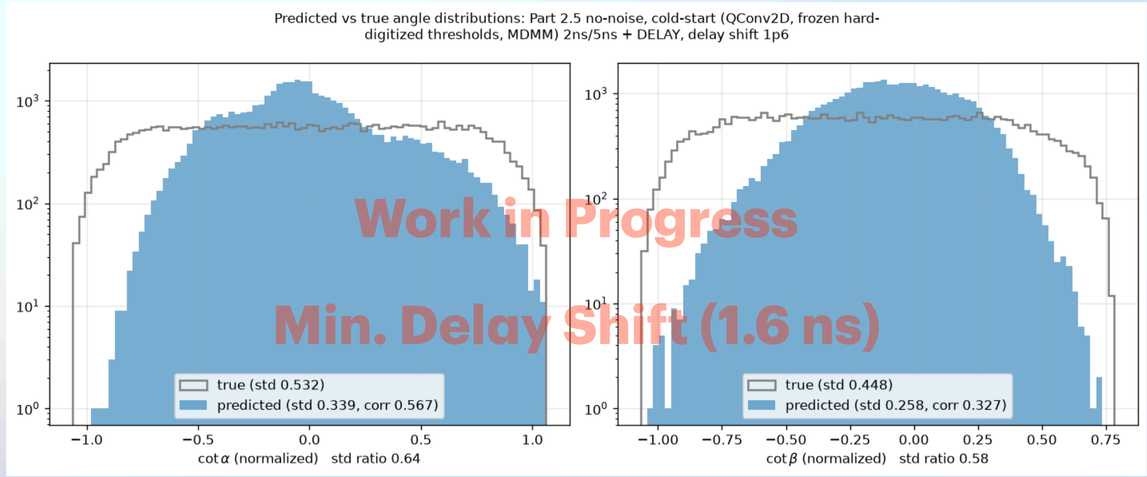
Adding Further Realism: Comparator Delay

Compartor Walks in Time

- **Per-pixel delay**, an LUT maps **input charge and threshold** to a crossing delay, taken **at the highest threshold** the pixel reaches.
- A **fixed 12.5 ns auto-zero window** limits how late a crossing can still register: a crossing that lands outside it is **floored to the previous level**.
- Pass-through only! As of now, the trained Stage 3 and 4 networks were evaluated on delayed data with 2 ns and 5 ns time slices, and then advanced by the minimum (1.6 ns) and median (3.2 ns) delays.
- Position survives and re-times (Conv2D **x** corr 0.975 (no delay) → 0.864 → 0.950 → 0.960). The angle prediction collapses to 29–50% of the true spread, and re-timing recovers it only partially, to 59–69%







Takeaways

- Biggest Takeaway: We can now test the on-chip NN against a realistic frontend, end-to-end. CSA shaping, 2-bit digitization, and comparator time-walk, rather than idealized charge alone.
- **Network performance persists even** when moving to a realistic frontend model → accuracy doesn't fully collapse under real hardware constraints (at least positions).
- We can quantify the **cost of dropping each idealization**: one change per stage and statistics matching. That is how we separated the effect of sample size and found the real effect of frontend hardware.
- We can now also say frontend effects **do not cost all observables equally** → re-timing brings position back → the angle stays collapsed, because the delay is per-pixel and no single shift fits every pixel.
- Plan: For a **SoftRouter** that learns which two slices to read, as **SoftDigitization** does for the thresholds → already implemented and trained in separate studies → next, train it on delayed data: can a learned placement of the two samples recover the angle?
- As we approach more and more realism in our model training, we expect to achieve parity with real-device performance. Stay tuned!

Thank You!!

Smartpixels Team

- **Cornell University:** Jennet Dickinson (professor), Ben Weiss (grad student)
- **Fermi National Accelerator Laboratory:** Douglas Berry (scientist), Giuseppe Di Guglielmo (senior engineer), Farah Fahim, Abhijith Gandrakota (scientist), Lindsey Gray (scientist), James Hirschauer (scientist), Ron Lipton (scientist), Benjamin Parpillon, Chinar Syal, Nhan Tran (scientist), Vasilije Perovic (postdoc)
- **Johns Hopkins University:** Petra Maksimovic (professor), Morris Swartz (professor)
- **Northeastern University:** Nick Manganelli (postdoc)
- **Purdue University:** Miaoyuan Liu (professor), Arghya Ranjan Das (grad student), Shiqi Kuang (grad student)
- **University of Chicago:** Karri DiPetrillo (professor), Benjamin Roberts (postdoc), Eliza Howard (grad student), Daniel Abadjiev (grad student), Eric You (undergrad student)
- **University of Colorado Boulder:** Keith Ulmer (professor), Jannicke Pearkes (postdoc)
- **University of Illinois Chicago:** Corrinne Mills (professor), Amit Trivedi (professor), Mohammad Abrar Wadud (postdoc), Harshul Gupta (graduate student), Danush Shekar (grad student)
- **University of Illinois Urbana-Champaign:** Mark S. Neubauer (professor), David Jiang (grad student)

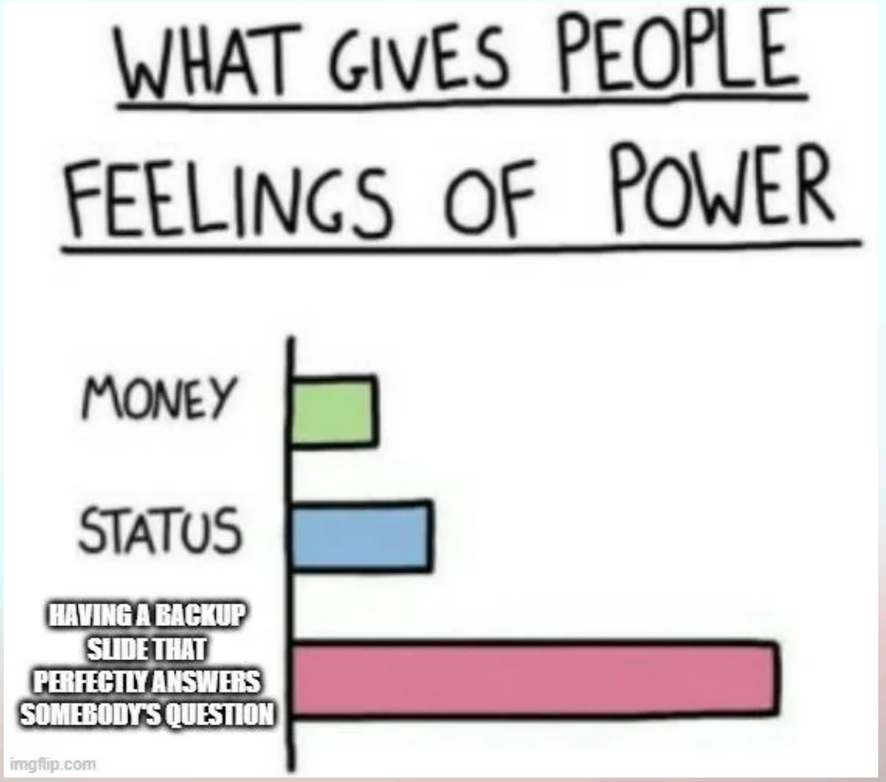
References

- Arghya Ranjan Das et al, On-chip probabilistic inference for charged-particle tracking at the sensor edge. Submitted to PRX Intelligence
- Jieun Yoo et al, Smart pixel sensors: towards on-sensor filtering of pixel clusters with deep learning, Mach. Learn.: Sci. Technol. 5 035047
- Benjamin Parpillon, A. R. Trivedi and F. Fahim, "Readout IC with 40 MSPS in-pixel ADC for future vertex detector upgrades of Large Hadron Collider," 2023 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, USA, 2023, pp. 1-5, doi: 10.1109/ISCAS46773.2023.10182033.
- Shruti R. Kulkarni et al, "On-Sensor Data Filtering using Neuromorphic Computing for High Energy Physics Experiments", ICONS 23 Proceedings of the 2023 International Conference on Neuromorphic Systems, Aug. 2023, https://doi.org/10.1145/3589737.3605976
- Narasinga Rao Miniskar et. al, A 3D Implementation of Convolutional Neural Network for Fast Inference", 2023 IEEE International Symposium on Circuits and Systems (ISCAS)
- L. Gray (FNAL), Wine & Cheese seminar talk: https://indico.fnal.gov/event/71335/
- J. Dickinson (Cornell), CPAD 2022: https://indico.bnl.gov/event/17072/contributions/70204/
- D. Shekar (UIC), CPAD 2024: https://indico.phy.ornl.gov/event/510/contributions/2260/
- C. Mills (UIC), Vertex 2025: https://indico.phy.ornl.gov/event/677/contributions/2754/
- D. Shekar (UIC), EPIGRAPHY 2026: https://indico.cern.ch/event/1608897/contributions/6814461/

Acknowledgements

- U.S. Department of Energy award DE-SC0023715
- Meenakshi Narain Graduate Scholar Fellowship by LHC Physics Center (LPC), Fermilab
- Kudos to Arghya Ranjan Das, Jennet Dickinson, Benjamin Parpillon, Lindsey Gray and Corrinne Mills for their constant help and support for these studies.

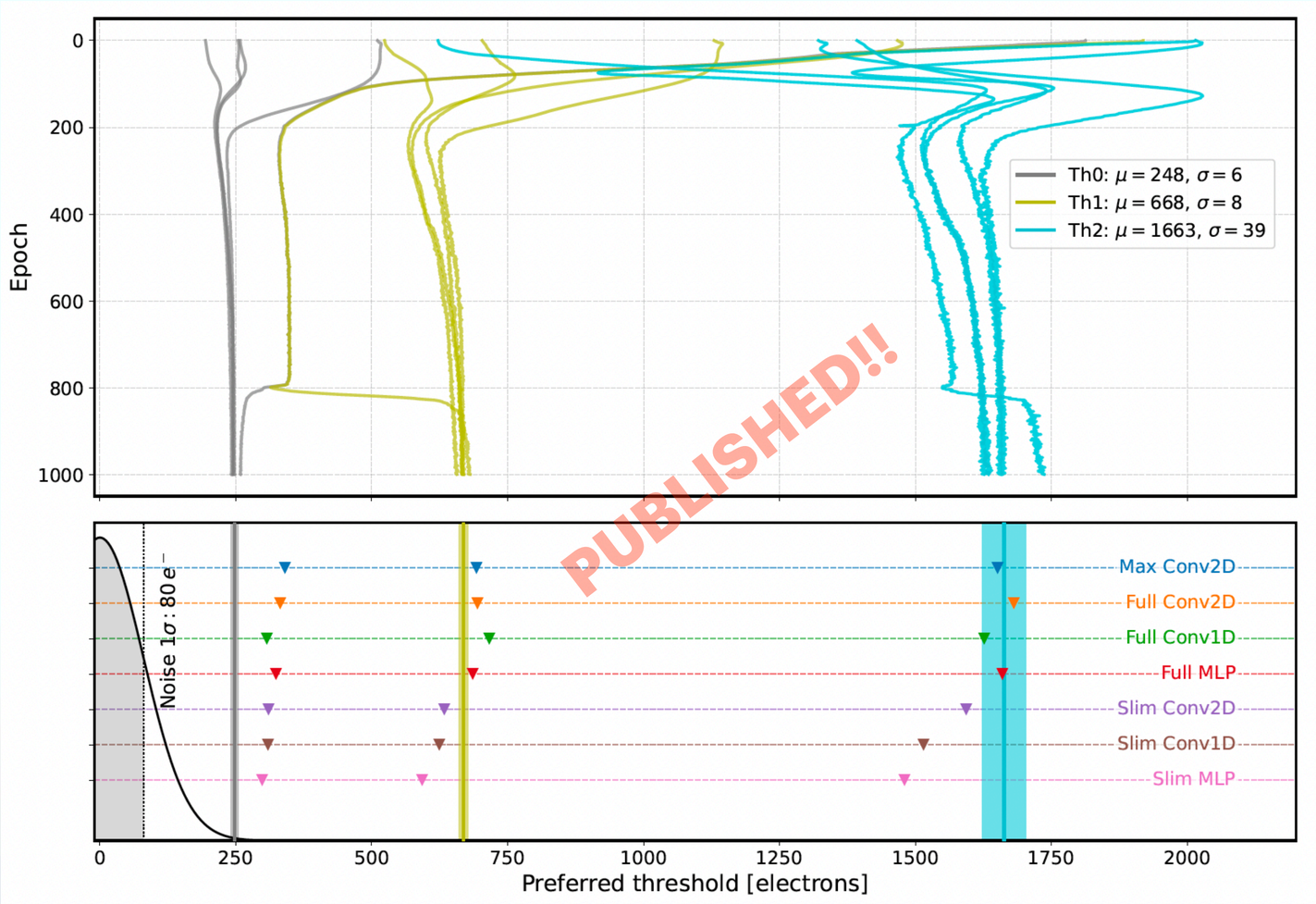
Backup


WHAT GIVES PEOPLE
FEELINGS OF POWER
MONEY
STATUS
HAVING A BACKUP SLIDE THAT PERFECTLY ANSWERS SOMEBODY'S QUESTION
imgflip.com

Charge Dataset and Training results

Regression studies with the raw electron charge dataset were done and published.

- Top: Threshold values as a function of epoch in the training of the Max transformer with SoftQuantize. Bottom: Points indicate the preferred charge thresholds for each regression model.
- The vertical lines and shaded bands represent, respectively, the means and standard deviations of the thresholds obtained by the transformer.



Credits: A. Das (Purdue), and smartpixels team

More Results: here

Model Outputs

What do we get out of it?

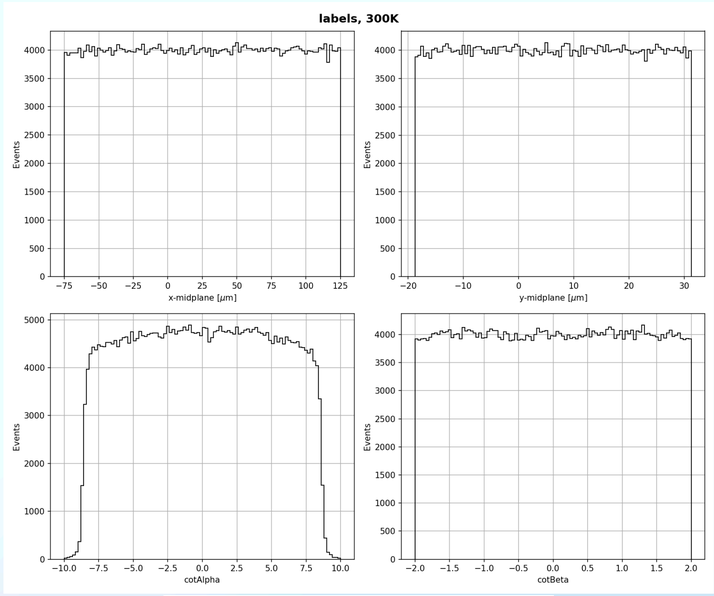
- The 14 outputs, in the order the model actually produces them [x, M11, y, M22, cot α, M33, cot β, M44, M21, M31, M32, M41, M42, M43]:
- 4 means (the predicted quadruple): x, y, cot α, cot β
- 10 uncertainty terms — a lower-triangular 4×4 Cholesky factor L (so the full covariance $\Sigma = L \cdot L^{T}$ is guaranteed valid/positive-semi-definite, rather than the net outputting raw covariance entries directly):
- Row 1: M11
- Row 2: M21, M22
- Row 3: M31, M32, M33
- Row 4: M41, M42, M43, M44

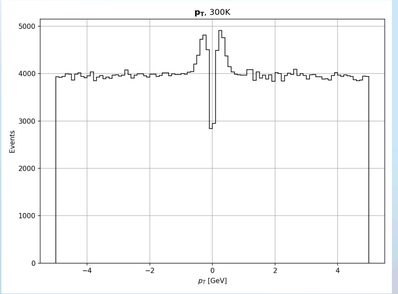
Constructing a dataset for analysis

The ML is only as good as the training data!

- State-of-the-art datasets for developing algorithms for on-ASIC implementation are generated using TCAD and PixelAV simulations with realistic detector geometry and operating conditions.

- We simulated MIP interactions for this futuristic pixel detector (Phase 3 CMS)
  - Located at a radius of 30 mm
  - In a 3.8 T magnetic field
  - For 50 x 12.5 micron pixel sensors
  - With 100 μm thickness
  - At time steps of 10 ps and 200 ps
  - Up to 4 ns (or further)
  - A flat-pT distribution with cot α ∈ [-10, 10]; cot β ∈ [-2, 2].
  - 300 K and 243K temperatures
  - Over a pixel readout window (e.g., 16 × 16, 13 × 21)



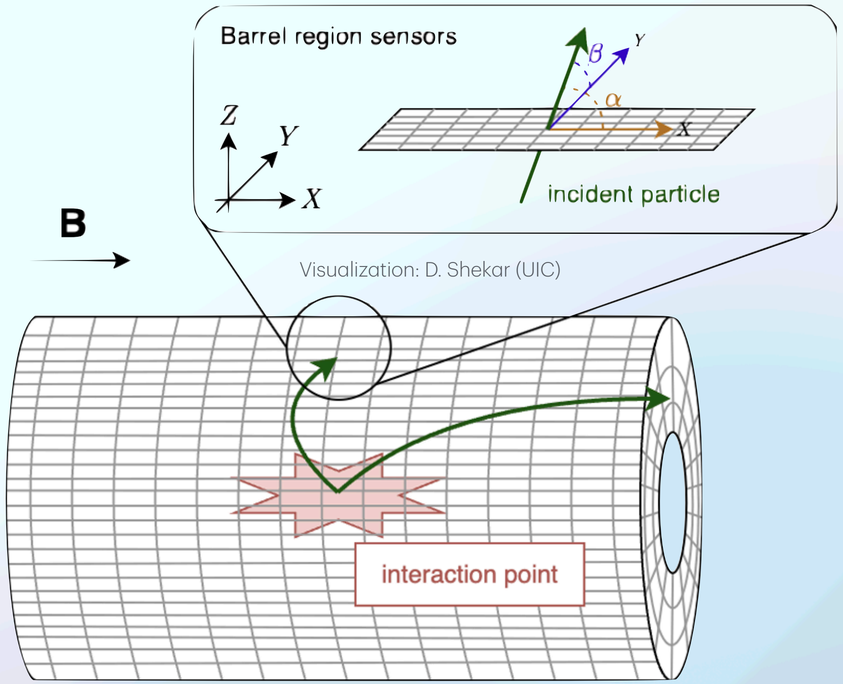


Some older datasets are available at: Zenodo

How are Clusters Identified?

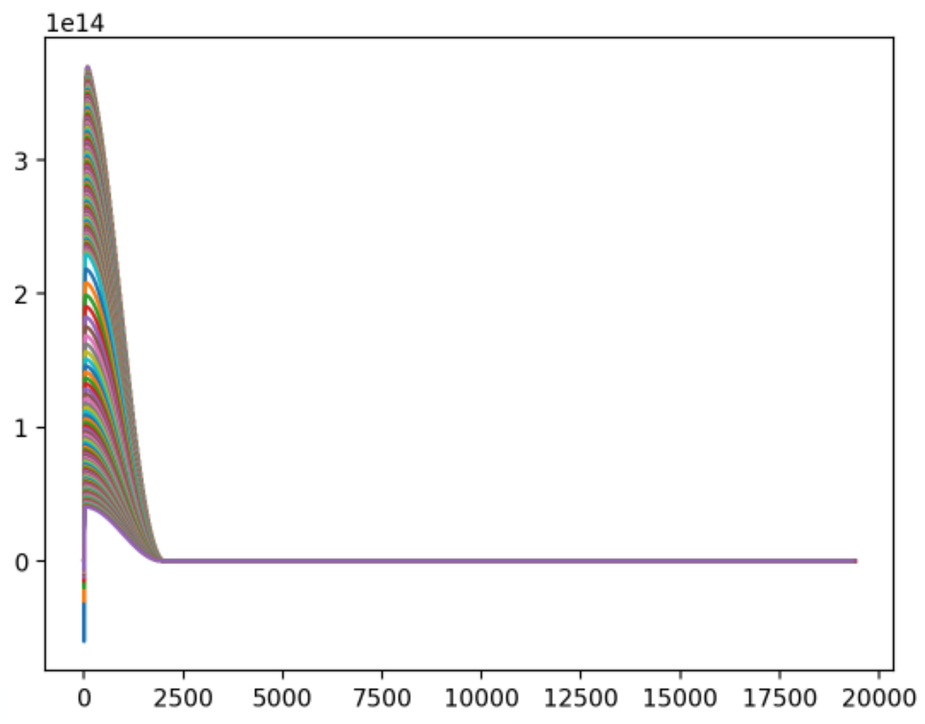
What to look for?

- Our Models aim to estimate the **x- and y-positions**, as well as the **cotangents** of the **angles α & β**.
- x, y = the track's position extrapolated to the **sensor midplane** (z = 50 μm), not where it physically enters the sensor.
- X-midplane/Y-midplane is a pure simulation-truth quantity, computed directly from the particle's true track geometry (entry point + trajectory slope), independent of how charge actually spreads across pixels:
- **α** - the incident angle in the x – z plane
- **β** - the angle in the y – z plane
- x-midplane = x-entry + cot α * (thickness/2 - z-entry)
- y-midplane = y-entry + cot β * (thickness/2 - z-entry)
- Each simulated MIP crossing produces exactly **one** entry point, **one** trajectory, and hence **one** (x-midplane, y-midplane, cot α, cot β) **quadruple**.
- This single-truth-value quadruple is the regression target.

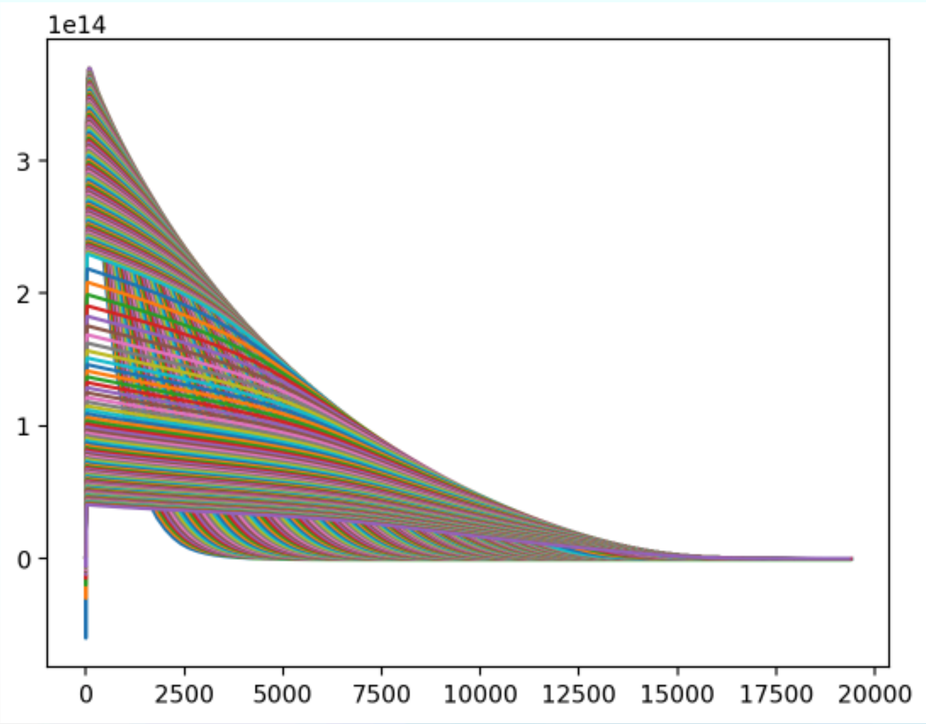


Pulse Response

Tapered



Non-Tapered



Visualization: J. Dickinson (Cornell), L. Gray (FNAL)

From Threshold Search to Chip-Deployable Model

Training Stages Details

- **Stage 1:** ViT + trainable soft 2-bit ADC → thresholds are learned jointly with the position/angle regressor, producing the median thresholds carried forward into every later stage. Noise is added in this stage
- **Stage 2:** Same ViT, thresholds now frozen and hard-digitized (real ADC-style bucketize, not the soft trainable version) → isolates the cost of no longer being able to optimize thresholds during training. Noise is removed, but thresholds are kept.
- **Stage 3:** Shrunk to a compact, chip-sized Conv2D architecture (unquantized) → isolates the cost of moving from the full ViT down to a hardware-sized model.
- **Stage 4:** Same compact architecture, now weight-quantized (QKeras, 4-bit conv / 8-bit dense) → the actual chip-deployable model → isolates the cost of quantization-aware training.
- **Stage 5:** Add the pixel-level comparator (threshold) delays from a lookup table and perform a pass-through (or retraining).
- Each stage adds exactly one more real-world constraint on top of the previous one, so the accuracy drop at each step can be attributed to a single specific cause rather than several tangled together.

Noise Modeling

- CSA (Charge Sensitive Amplifier) bandwidth limits the noise spectrum, turning white noise into a band-limited process, and adjacent time samples are correlated.

  Noise parameters

  ENC (Equivalent Noise Charge) → 80 e^- RMS

  Charge-to-voltage gain (CvG) → 58 μV / e^-

  Noise amplitude σ → CvG × ENC × 1000 = 4.64 mV

  Mean μ → 0 mV

  CSA bandwidth f_c → 55 MHz

  Time constant $\tau = 1/(2\pi f_c)$ → 2.894 ns

- Correlation model

  $n_1 \sim \mathcal{N}(\mu, \sigma)$

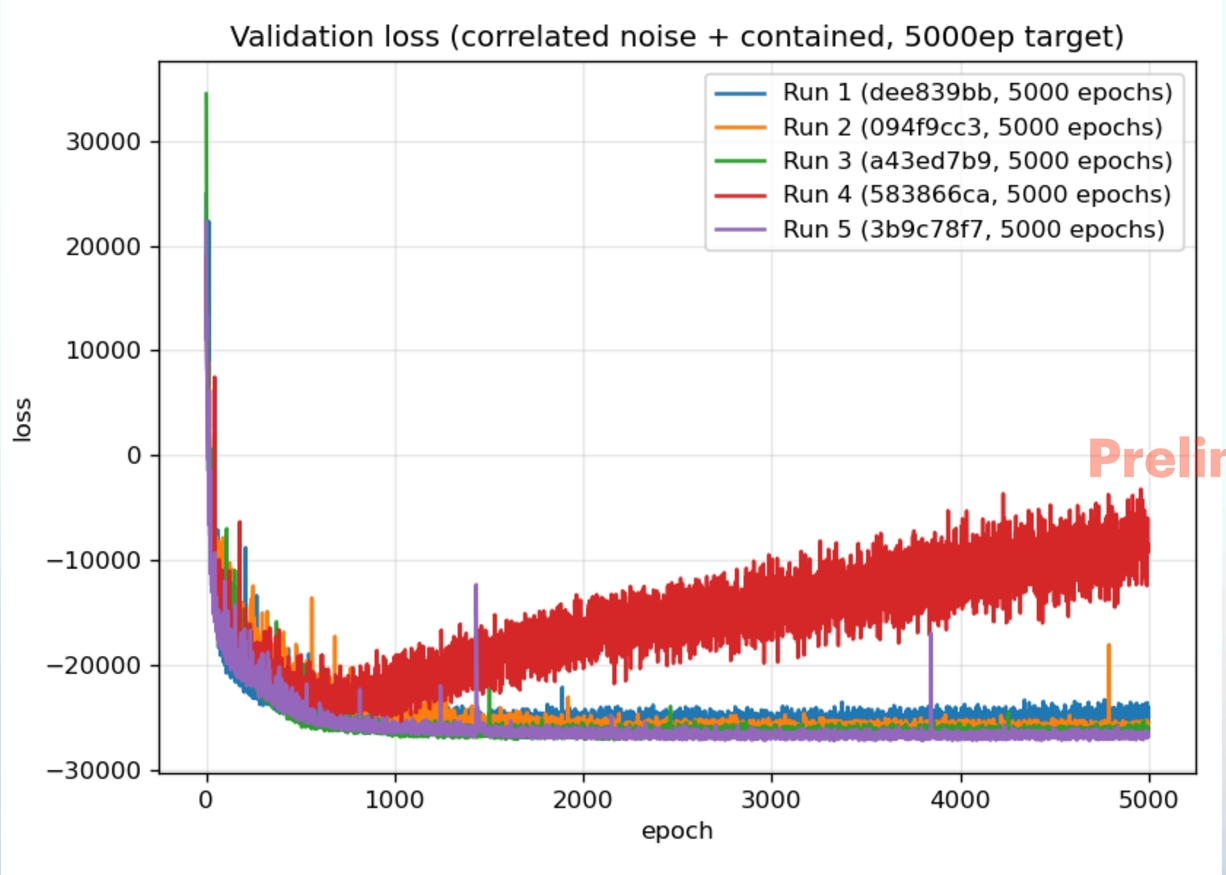
  $n_i = \mu + \rho\,(n_{i-1} - \mu) + \sqrt{(1 - \rho^2)} \cdot \mathcal{N}(0, \sigma)$ for $i > 1$

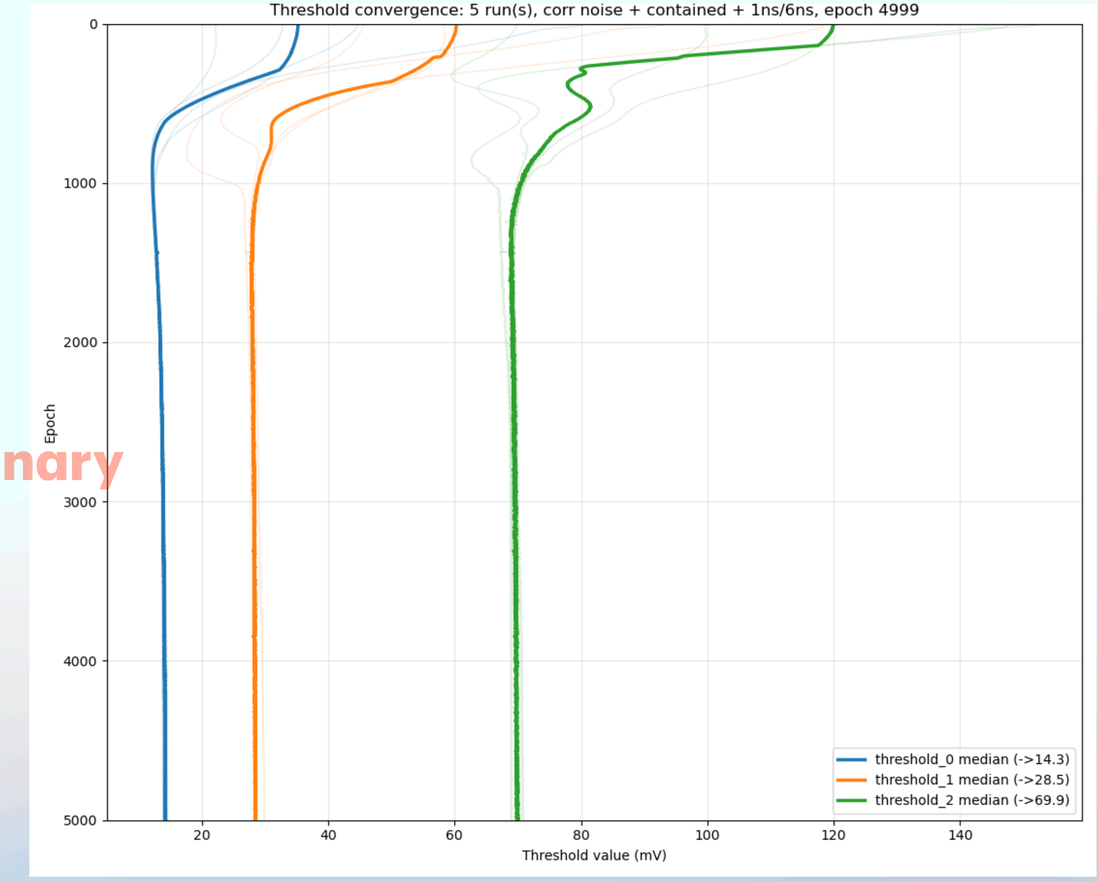
  where $\rho = \exp(-\Delta t / \tau)$

- Adjacent-slice correlation = ρ, two-apart = ρ^2, k-apart = ρ^k
- Noise is baked into TFRecords at generation time — not re-applied during training
- σ is identical for both experiments; only ρ changes with the time gap

Case: Contained Clusters, Time-Correlated Noise, 1/6 ns Slices

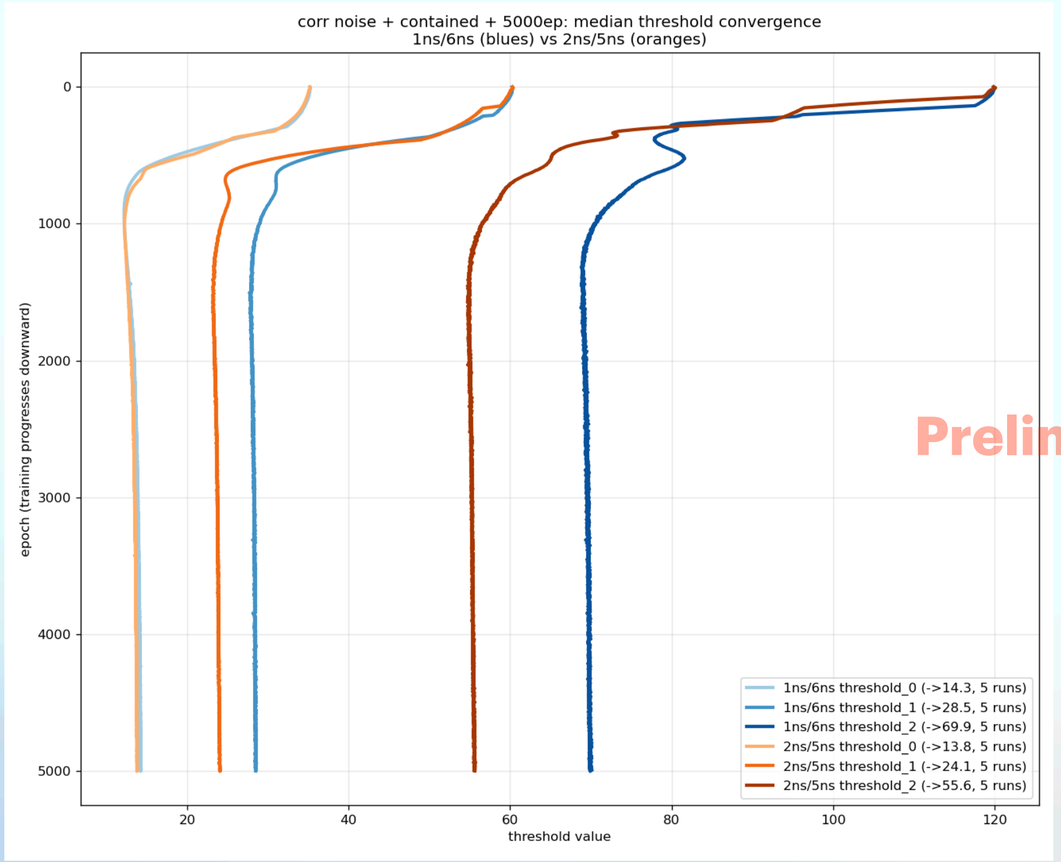
- 5 runs of 5000 epochs each, random initial thresholds.
- Thresholds comparable to raw charge datasets cases (with 58 μV/e gain conversion factor).
- Correlated Noise added, NLL Loss performance stuck at local minima.

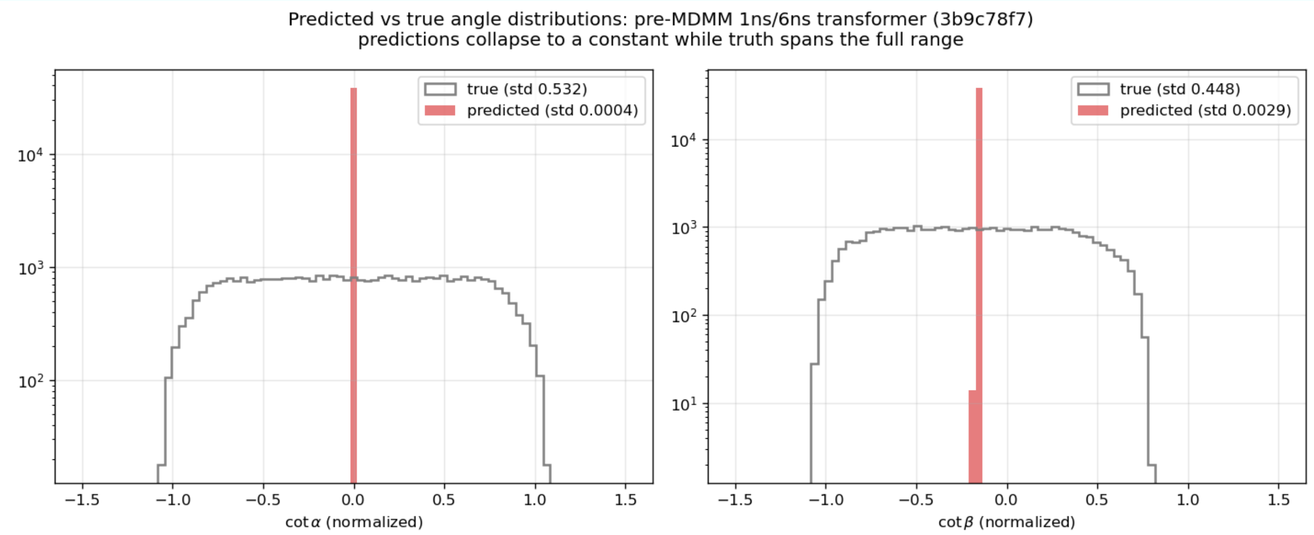




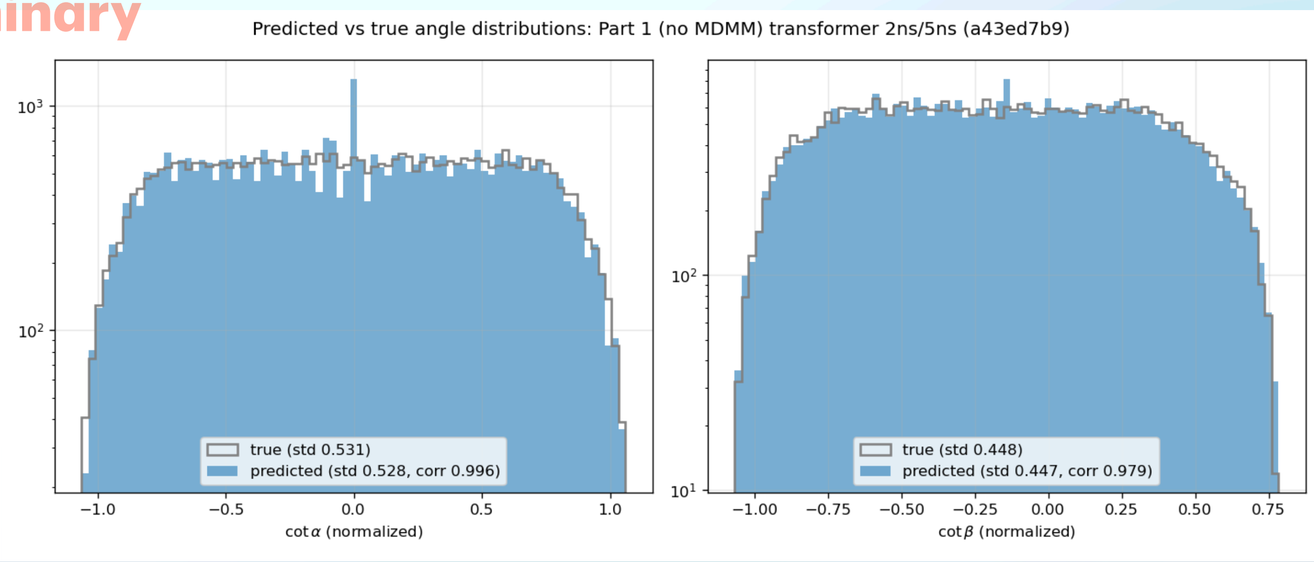
Preliminary

Comparison for 1/6 ns & 2/5 ns cases (with & w/o MDMM)



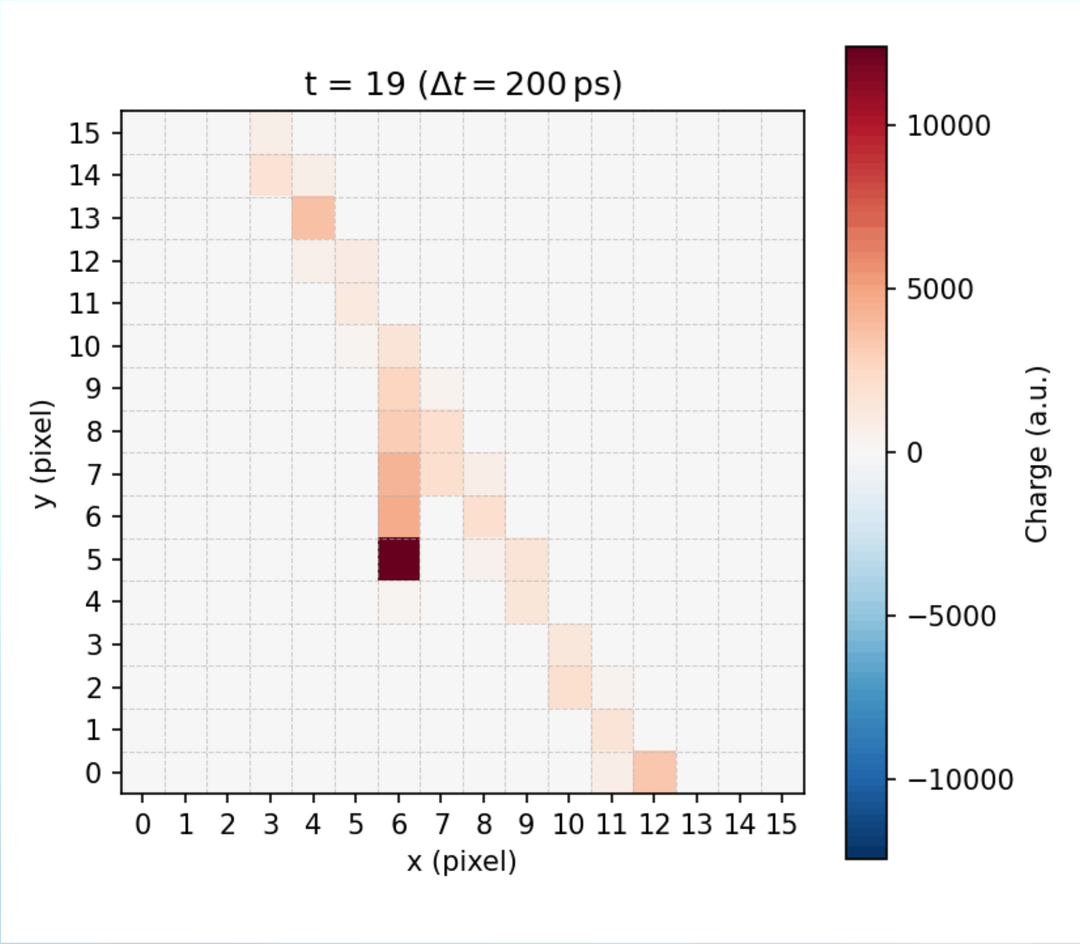


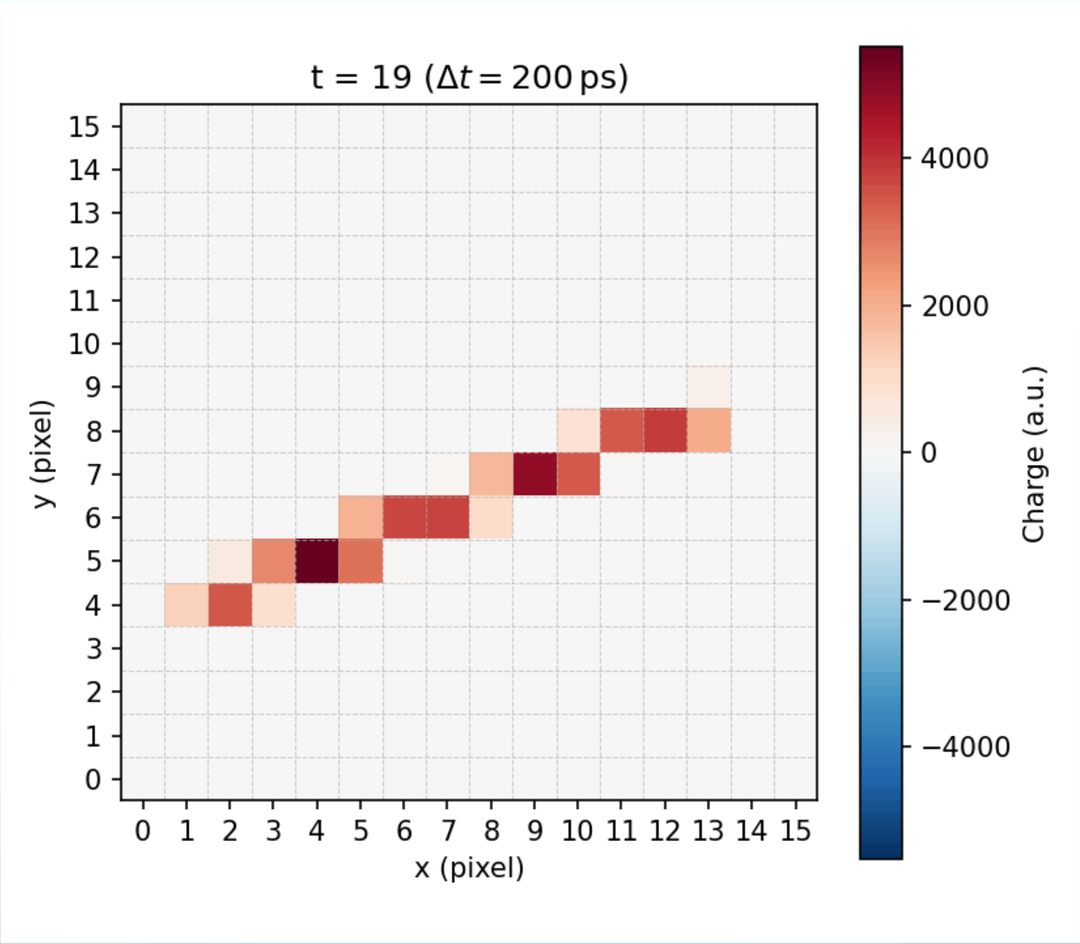
Preliminary



Charge Tracks

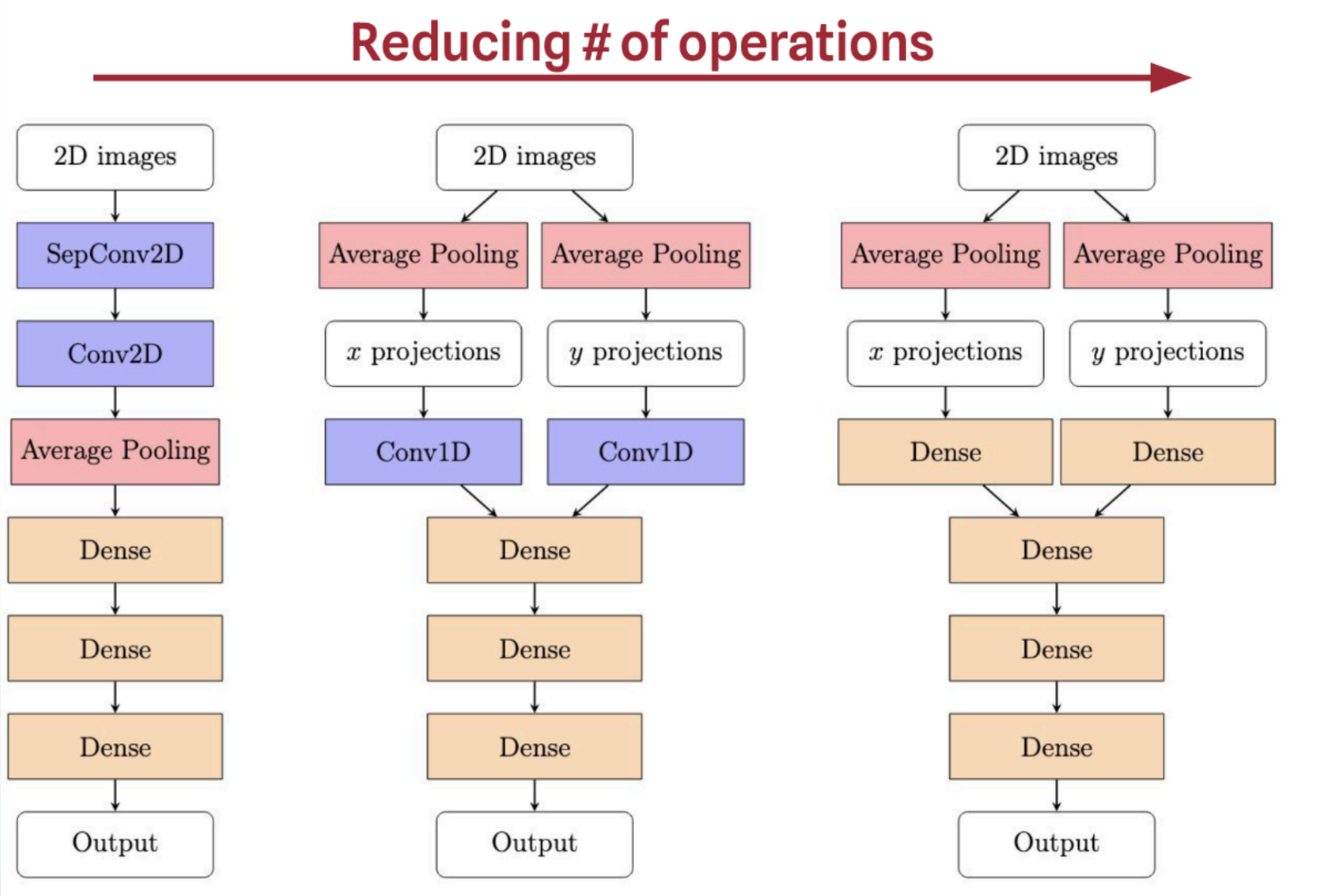
What do they look like?





Visualization: J. Dickinson (Cornell), L. Gray (FNAL)

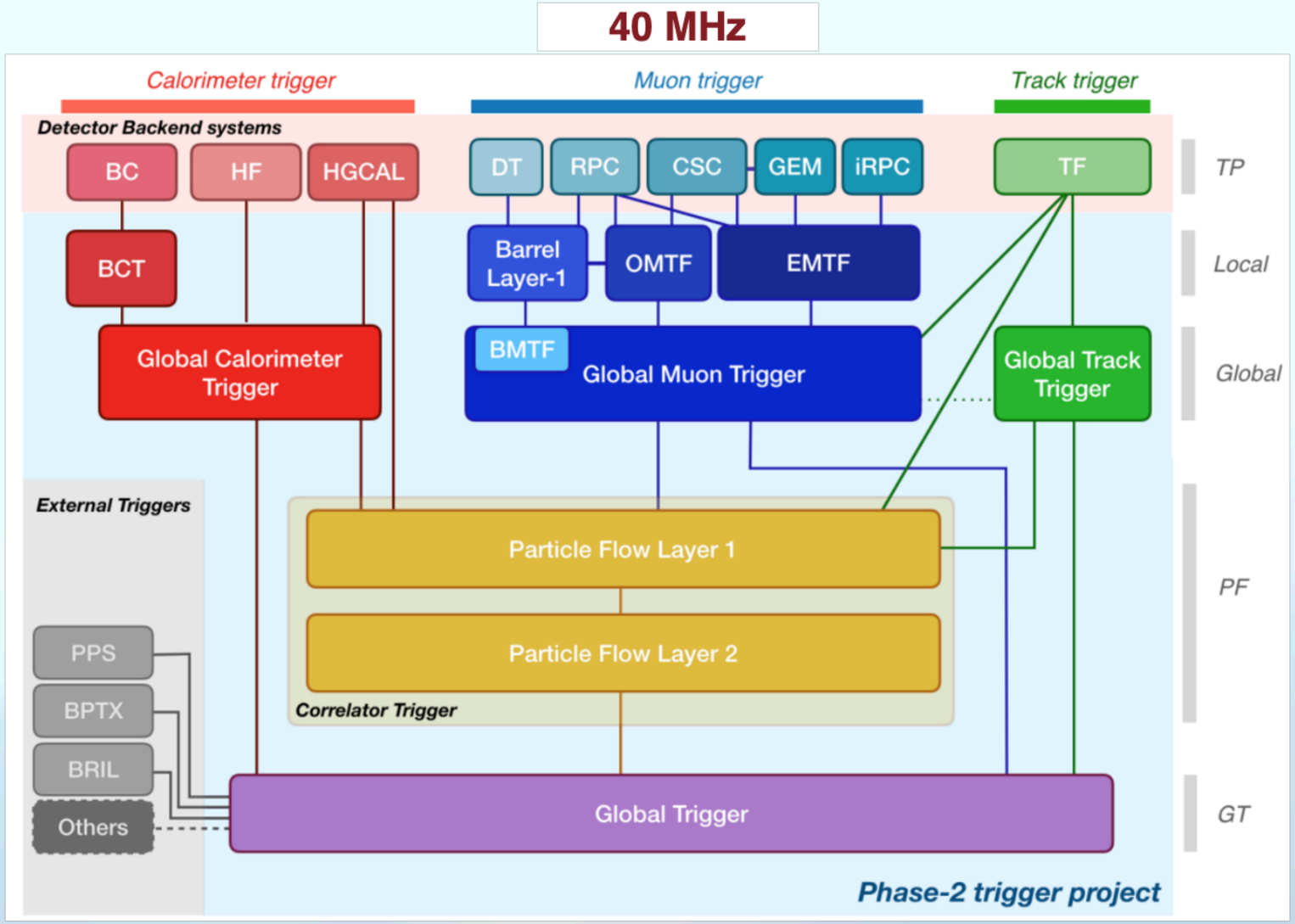
Regression Model



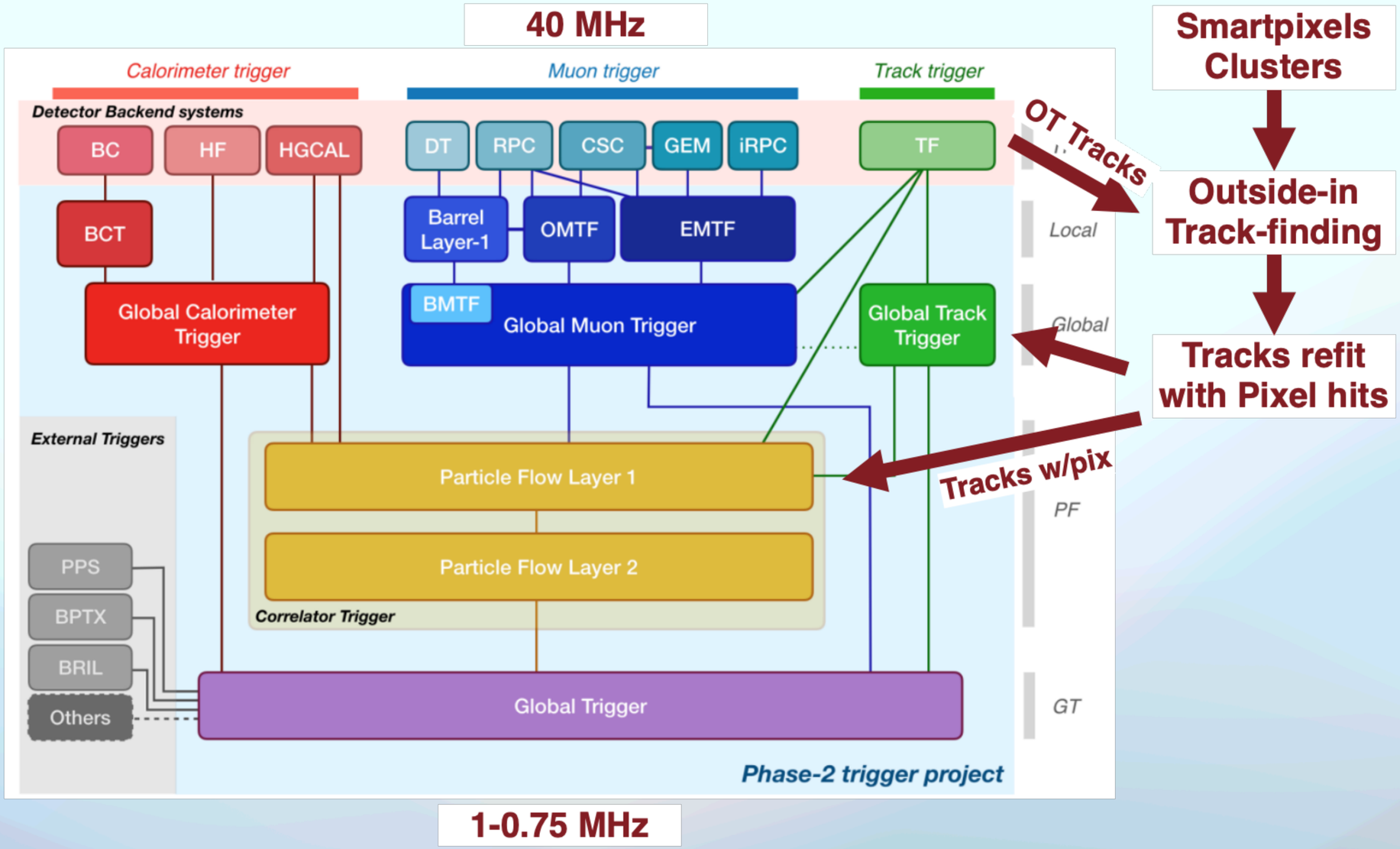
Harshul Gupta, Smartpixels Consortium

Visualization: J. Dickinson (Cornell)

The Phase 2 Level-One Trigger Upgrade



A possible L1 implementation



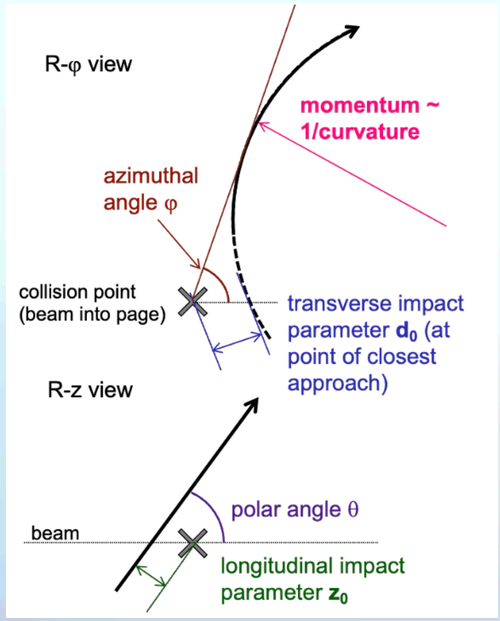
Harshul Gupta, Smartpixels Consortium

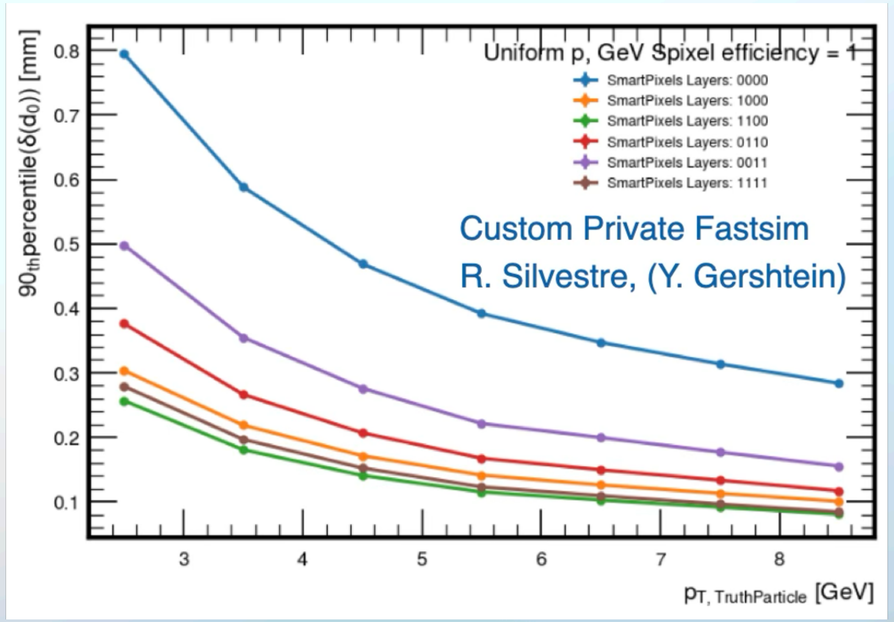
Visualization: L. Gray (FNAL)

Physics Cases

The Clearest Usecase: Impact Parameter

- The transverse impact parameter allows us to search for displaced vertices
  - Indicative of B-meson production, long-lived particles
- Longitudinal impact parameter helps separate proton collisions from one-another

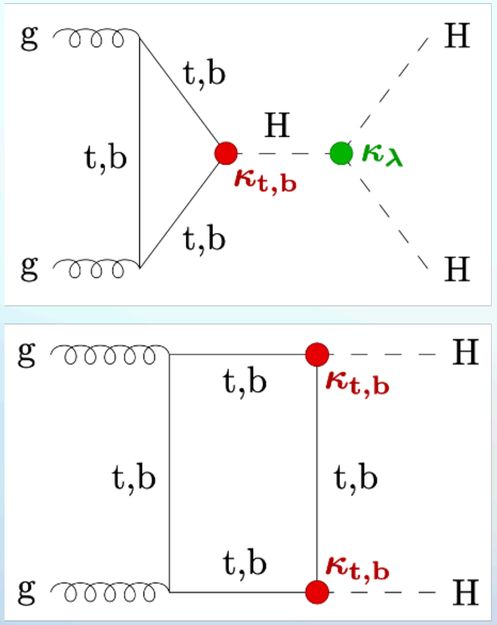


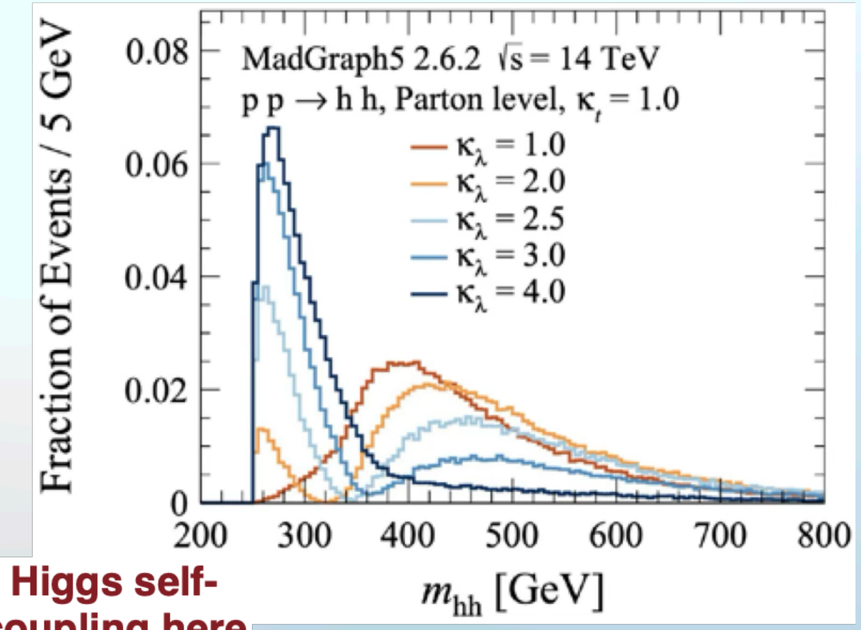


Physics Cases

What could precise impact parameters yield?

- Di-Higgs is the clearest near-term physics use case for this technology in collider HE
- The current outer tracker is only somewhat sensitive to displaced vertices
- The most critical part of Di-Higgs kinematic phase space is hardest to trigger on





Higgs self-coupling here