Machine Learning to Uncover the Oldest Light: U-Net Segmentation for Glitch Detection in Atacama Cosmology Telescope Data

The Final Report for KAUST Academy

25 - Aug - 2025

Anas Alshehri
U-Net Model implementation, simulation evaluation, real data pipeline
Electrical and Computer Engineering
King Abdulaziz University
Jeddah, Saudi Arabia
anas12alshehri12@gmail.com

Abdulaziz Alkharjy
simulation data generation, real data pipeline, clustering
Software Engineering
King Saud University
Riyadh, Saudi Arabia
abdulazizsaad1370@gmail.com

Mehtab Cheema
LayerCAM Model implementation, clustering
Computer science
University of Toronto
Ontario, Canada
https://www.linkedin.com/in/cheemamehtab/

Project Mentors: Adam Hincks, Renée Hložek, and Yilun Guan.

Project Objectives and Goals

The aim of this project is to develop a machine learning model capable of detecting and classifying glitches, such as point sources and cosmic rays, in time-ordered data (TODs) from telescopes like the Atacama Cosmology Telescope (ACT) and the Simons Observatory (SO). TODs are sequences of raw observational data collected over time by each detector in the telescope's focal plane. These data streams often contain glitches that contaminate the signal and must be flagged or removed before accurate analysis of the cosmic microwave background (CMB) is possible. Our goal is to automate this process

by using an image segmentation model to locate and classify each glitch on the focal plane image, which can then be mapped back to detector coordinates for further interpretation.

Methodology and Approach

Our methodology involves converting telescope timestream data into long exposure images of the focal plane. These images are then passed through the segmentation model, and the output masks are remapped into detector space to aid scientific interpretation.

Now we will dive deep into datasets and the model.

Datasets

- **Time-Ordered Data (TOD):** per-detector time series stored as a 2-D array with shape **[D, T]** = [number of detectors, number of time samples]. Each entry **TOD[d, t]** is the sample from detector **d** at time index **t**. These streams often contain transient glitches that should be detected/flagged for accurate CMB analysis.
- **Simulated Data**:
    - **Source/creation.** Synthetic **Time-Ordered Data (TOD)** with injected glitches (Point Sources, Cosmic Rays), controlled by **19 parameters** (as ahown in the table)
    - **Derived products.** TOD → **focal-plane images** on a fixed grid (e.g., **32×32**).
    - **Input representation.** We **use two channels** per image—**sum** and **max** exposure—which empirically gave the best performance and generalization on simulations.
    - **Labels.** Pixel-level masks (PS/CR) are generated analytically from ground-truth injections and focal-plane geometry.

All Simulation Parameters Table:

key	default value	units	explain / what it controls
FP_RADIUS	0.8	deg	Radius of the circular focal plane used to place detectors.
FP_NROWS	30	—	Number of detector rows (grid density proxy for focal-plane layout).

SIM_T0	1672531200	Unix s	Start epoch (Unix time) for the simulated scan.
SIM_AZ_START	45	deg	Starting azimuth angle of the telescope scan.
SIM_EL	60	deg	Fixed elevation angle maintained during the scan.
SIM_AZ_THROW	10	deg	Total azimuth throw (peak-to-peak sweep width).
SIM_V_AZ	2	deg/s	Azimuth scan speed (slew rate along azimuth).
SIM_SRATE	200	Hz	Sampling rate per detector (samples per second).
SIM_DURATION	120	s	Total scan duration used to generate the timestream.
WHITE_NOISE_LEVEL	0.2	arb.	RMS amplitude of injected white (instrumental) noise in TOD.
POINT_SRC_BEAM_SIGMA	4	arcmin	Gaussian beam σ for point-source rendering (FWHM ≈ 2.355×σ).
POINT_SRC_FLUX_LIMS	[1, 3]	arb.	Min/max point-source flux used when sampling source strengths.
CR_AMP_SCALE	5	arb.	Amplitude scale factor for simulated cosmic-ray hits.
CR_RADIUS_SCALE	2	arcmin	Characteristic radial extent of cosmic-ray footprints.
SEED	42	—	RNG seeds for reproducible dataset generation.
N_SRCS	5	—	Number of point sources per scene.
CR_RATE	0.01	—	Relative rate parameter controlling expected CR event frequency.
GRID_RESOLUTION	32	px	Output grid resolution per axis (produces 32×32 focal-plane images).
CHUNK_SEC	1	s	Time window per mapping chunk (frame length for video-style mapping).

With our generation script, a single run can produces 120 image–mask pairs plus their TOD excerpts in less than 1 minute. Full configuration and step-by-step logic are documented in the script code source.

- **Real Data (from ACT)** :
  - **Source.** Slices of **ACT** TOD: per-detector timestreams across the focal plane that can contain transient glitches contaminating the CMB signal.
  - **Derived products.** TOD → **focal-plane images** using our exposure pipeline.
  - **Input representation.** We **use a single channel (max exposure)**, which proved most robust on real ACT data given sampling variability and label characteristics.
  - **Labels.** Since pixel-level masks were not originally available, we generated them by projecting detector/time glitch labels into the 32×32 focal-plane grid using our preprocessing pipeline. Specifically, labeled slices were mapped to the focal-plane coordinates of affected detectors, and then converted into binary masks (0/1, saved as PNGs). These masks serve as ground truth for training and evaluation, but they are derived from existing glitch metadata rather than from manual pixel annotations.

Datasets summary

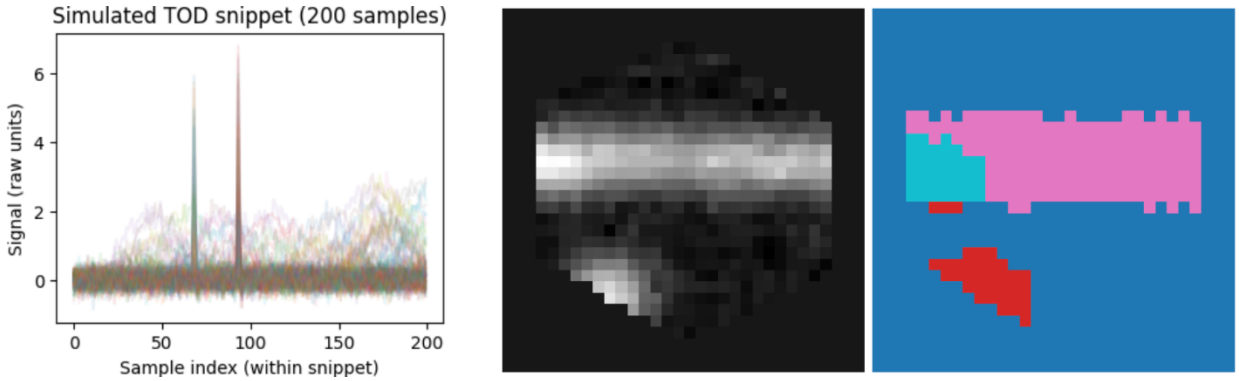
-	**Simulated**	**Real (ACT)**
Input channels for the model	**2** (sum, max)	**1** (max)
Labels	Exact pixel masks from injections	Masks derived from detector/time metadata
Glitches	PS, and CR	PS, CR, and EL.
Grid	Fixed (e.g., **32×32**)	Same grid

- **Mapping from TOD to focal-plane images (common to simulated & real data)**
  - **Load geometry & TOD.** Read detector positions on the focal plane and the corresponding per-detector timestreams.
  - **Preprocess TOD.** Apply calibration, de-meaning, de-trending, and a **high-pass filter at 2 Hz** (sample rate **400 Hz**) to suppress low-frequency drift.
  - **Select detectors.** Restrict to **uncut detectors** in the **f090** band.
  - **Chunk in time (exposures).** Define time windows start,stop per frame.
    Note (real data conversion): we materialized frames for slices listed in the annotation CSV (e.g., Point Source, Cosmic Ray).

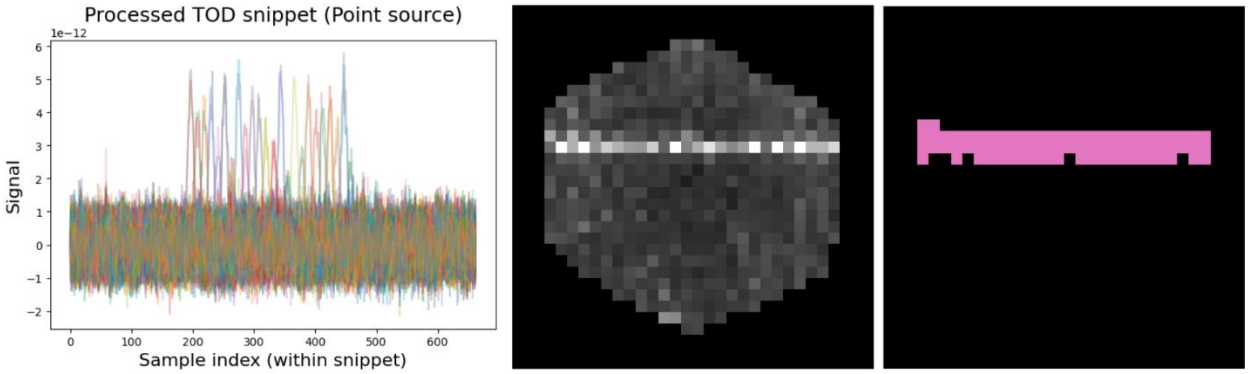
- **Map to a fixed grid.** Use the known detector sky coordinates to **map samples onto a fixed 32×32 focal-plane grid**, yielding a time×32×32 "video" for each TOD.
- **Aggregate per frame.** Within each exposure window, reduce the time axis to an image by taking the **maximum over time** (single-channel "max exposure").
  Simulated: two channels sum, max.
  Real (ACT): one channel, max (most robust on real data).
- **Normalize & save.** For real images, apply **per-frame percentile clipping** (0.5–99.5%) and rescale to 0,1, then save as 8-bit images. Filenames **encode the TOD ID and slice indices** for traceability.

- **Mask construction**
  - **Simulated data:** Rasterize the **known PS/CR injection geometry** onto the same 32×32 grid to obtain **exact pixel masks**.
  - **Real data:** Convert **detector/time glitch labels** from the metadata into **binary pixel masks** by mapping the flagged detectors within each exposure window onto the 32×32 grid. These masks are **derived from metadata**, not hand-drawn pixel annotations and not predicted by the model.

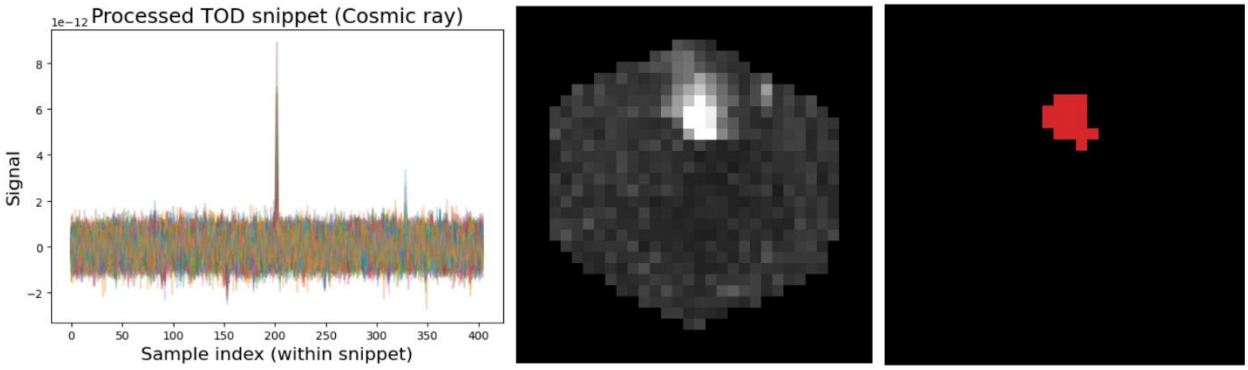
- **Transition to real data (what changed)**
  - Restricted to **uncut f090 detectors**.
  - Added a **2 Hz high-pass** step appropriate for the 400 Hz sampling.
  - Chose **single-channel max** aggregation for robustness on real data.
  - Generated masks **from metadata labels** rather than from exact injection definitions.



Simulated TOD slice → focal plane → mask



*Real **Point Source** TOD (ACT) slice → focal plane → mask*



*Real **Cosmic Ray** TOD (ACT) slice → focal plane → mask*

Model

We adopt a compact **U-Net–style convolutional neural network** for pixel-level segmentation of glitches in focal-plane images. The encoder–decoder structure captures both local spatial detail and global context, while skip connections ensure that fine-scale information is preserved throughout the network.

Architecture.

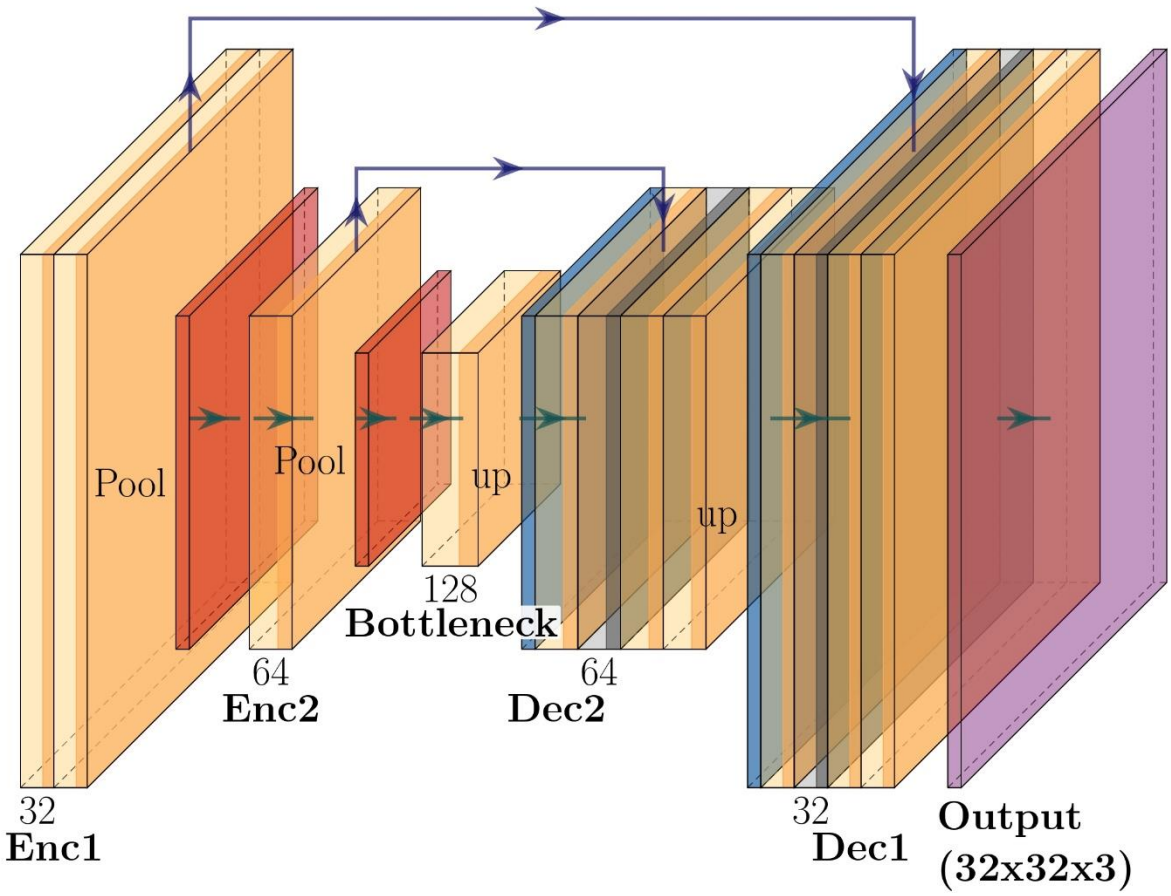
The model consists of two **downsampling stages** (encoder) followed by a **bottleneck** and two **upsampling stages** (decoder). Each downsampling step halves the spatial resolution and doubles the number of feature channels, allowing the network to progressively extract higher-level representations. The bottleneck layer encodes global features across the focal plane. In the decoder, transpose convolutions restore spatial resolution, and feature maps are concatenated with their encoder counterparts through skip connections, enabling precise localization of predicted regions. Dropout layers are inserted in the decoder to improve generalization. The final convolution produces a multi-channel output mask, one channel per glitch class.

Training setup.

The network is trained with a **hybrid loss** combining *binary cross-entropy* (penalizing pixel-wise misclassifications) and *Dice loss* (directly optimizing overlap between predicted and ground-truth masks). This combination balances per-pixel accuracy with robustness to class imbalance, which is important since glitches are sparse compared to background pixels.

Dataset used for training.

The training set includes **486 point source examples**, **2173 cosmic ray examples**, and **1384 electronic glitch examples**, each paired with a ground-truth mask. The dataset supports **multilabel segmentation**, meaning a pixel can belong to multiple classes simultaneously. This enables the model to correctly handle overlapping glitches in the same focal-plane region, a failure case for many existing glitch-finding approaches.



U-Net Model Architecture.

Current Progress and Milestones Achieved

Below we outline the key stages of the work—from early simulation to the final real-data results—so the reader can see how the pieces fit together.

- **Simulated data generation:** Generated **simulated TODs** with injected glitches (PS, CR), then projected them to **focal-plane images** (also made short "video-style" renderings of detector signals). The simulator exposes **19 parameters** (as shown in Table X). *(More details in the Datasets section.)*
- **Simulated-data training:** Trained a lightweight **2-down / 2-up U-Net** on the simulated images; ran **cross-training / cross-testing** across distinct simulated

sets (different seeds/noise/exposure configs) to probe generalization. On simulations, achieved **IoU = 0.94 (Point Sources)** and **0.79 (Cosmic Rays)**.

- Real data mapping to the focal plane. After receiving real ACT TODs, we built preprocessing (demean/detrend/high-pass) and converted timestreams to long-exposure focal-plane images.
- Real-data mask generation. Because pixel-level masks were not available, we generated them from metadata. Specifically, detector/time glitch labels from ACT TOD slices were mapped onto the focal-plane grid, producing binary 32×32 masks for each slice.
- Real-data training. We trained the same U-Net architecture **from scratch** on the real focal-plane images using these metadata-derived masks. This ensured the model learned directly from ACT data, independent of prior training on simulations.
- End-to-end pipeline. We implemented a production-style pipeline:
  Real TODs → preprocessing → focal-plane exposure images → U-Net segmentation → class-specific detector lists.
- Real-data performance. The model demonstrates strong separation of overlapping events (e.g., PS+CR+EL within the same region) on ACT images. Detailed metrics and examples are provided in *Preliminary results or findings*.
- LayerCAM explainability. We integrated LayerCAM to verify that predictions focus on physically meaningful regions in 32×32 inputs, using these maps for QA and targeted model refinements.
- Parallel clustering study (separate from U-Net; different dataset)
  – Data: SAT telescope (not ACT); arrays WS0–WS6 (WS5 absent).
  – Aim: Unsupervised grouping of instrumental-glitch families.
  – Features/space: 7 per-detector stats; clustering in 7-D; PCA 7→2 for visualization only.
  – Methods: DBSCAN vs. HDBSCAN; swept min_cluster_size {5,10,15,20,25}, min_samples {1–5}.
  – Result: HDBSCAN > DBSCAN under varying density—more distinct, stable clusters with fewer noise labels (cf. Figs. 15–16).
  – Artifacts: Cluster IDs/counts, PCA loadings, per-array summaries (for QA/heuristics).
  – Compute: Princeton University HPC.
  – Note: We can't share additional details (data, settings, plots) here for now due to data-use/NDA restrictions.

Preliminary Results or Findings

- **Effect of exposure aggregation (simulation):** We compared different ways of forming focal-plane images from TODs: **sum**, **max**, and **sum+max (mixed)**. The U-Net performs best when trained and tested on the same aggregation method (mean IoU ≈0.84–0.86), while cross-exposure generalization is poor. Mixed-channel training provides the most robust compromise across setups. This motivated our choice of **max exposure** for real ACT data, where only a single channel is feasible.

train/test	sum	max	max and sum (mixed)
sum	IoU cosmic-ray : 0.787 IoU point-src : 0.938 Mean IoU : 0.862	IoU cosmic-ray : 0.271 IoU point-src : 0.359 Mean IoU : 0.315	IoU cosmic-ray : 0.388 IoU point-src : 0.649 Mean IoU : 0.519
max	IoU cosmic-ray : 0.193 IoU point-src : 0.285 Mean IoU : 0.239	IoU cosmic-ray : 0.904 IoU point-src : 0.811 Mean IoU : 0.858	IoU cosmic-ray : 0.430 IoU point-src : 0.441 Mean IoU : 0.436
max and sum (mixed)	IoU cosmic-ray : 0.759 IoU point-src : 0.920 Mean IoU : 0.839	IoU cosmic-ray : 0.898 IoU point-src : 0.804 Mean IoU : 0.851	IoU cosmic-ray : 0.823 IoU point-src : 0.861 Mean IoU : 0.842

IoU performance of the U-Net under different exposure aggregation methods (sum, max, mixed). Rows indicate the training exposure type, and columns indicate the testing exposure type. Models perform best when train and test exposures match (mean IoU ≈0.84–0.86), while cross-exposure transfer leads to large drops. Mixed exposures provide balanced performance across setups.

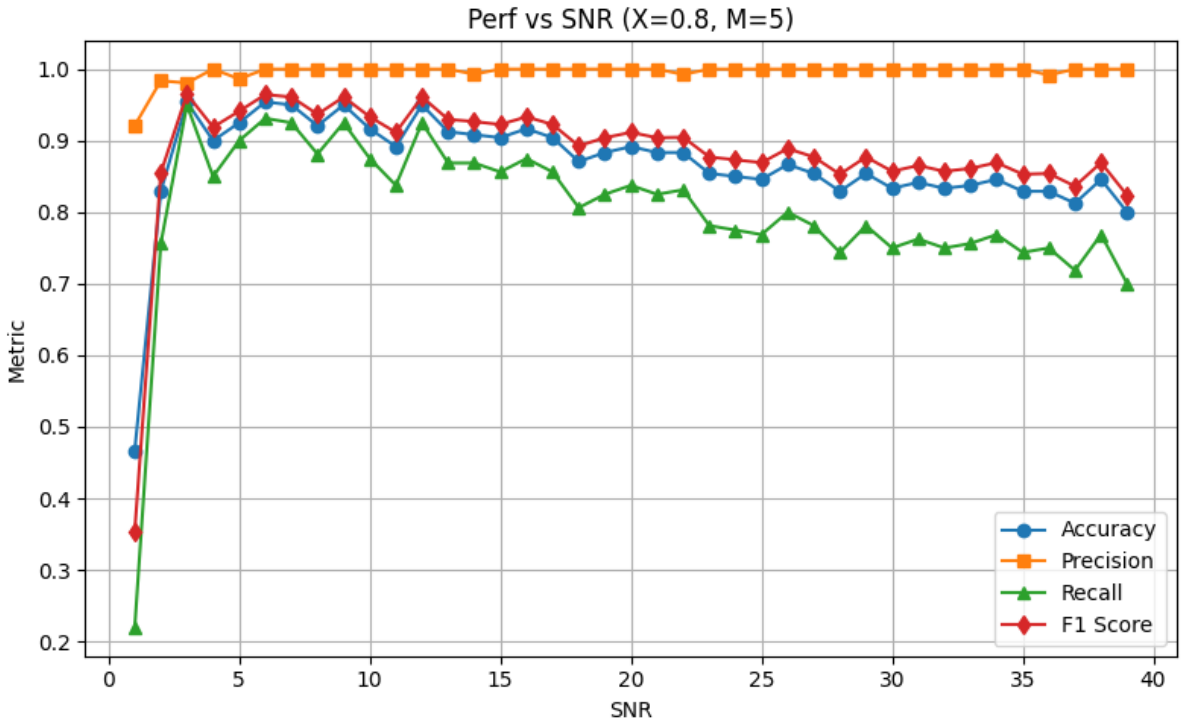
- **Overlap separation (new capability)**: The multilabel U-Net resolves overlapping glitches within the same focal-plane region (e.g., PS + CR), producing separate masks where the existing heuristic glitch finder cannot reliably disambiguate co-located events. Qualitative side-by-sides show clearer separation and fewer "single-label" failures in overlap cases.

- **Pixel-level performance on real ACT data (metadata-derived masks):** Using binary masks generated from detector/time metadata (rather than hand-drawn annotations), the model achieves:
  - **Point Sources (PS):** IoU = 0.93
  - **Cosmic Rays (CR):** IoU = 0.79

- **Electronic Glitches (EL):** IoU = 0.65

Because these masks are metadata-derived, the IoUs are conservative: true signal outside the metadata is penalized as false positives.

Cross-pipeline and SNR robustness with image-level evaluation: On real ACT images generated by an independent TOD→image pipeline, the model maintained strong performance, with trends remaining stable across varying signal-to-noise ratios (SNR). To rigorously evaluate this robustness on Maxwell Bridgewater's real dataset—which only provided text labels per image at different SNR levels—we developed a quantitative image-level evaluation method. Model predictions were post-processed by thresholding pixel probabilities and requiring a minimum number of activated pixels per class. An event was marked as "present" if both thresholds were exceeded and compared against the ground-truth label, enabling computation of accuracy, precision, recall, and F1-score even without pixel-level masks.

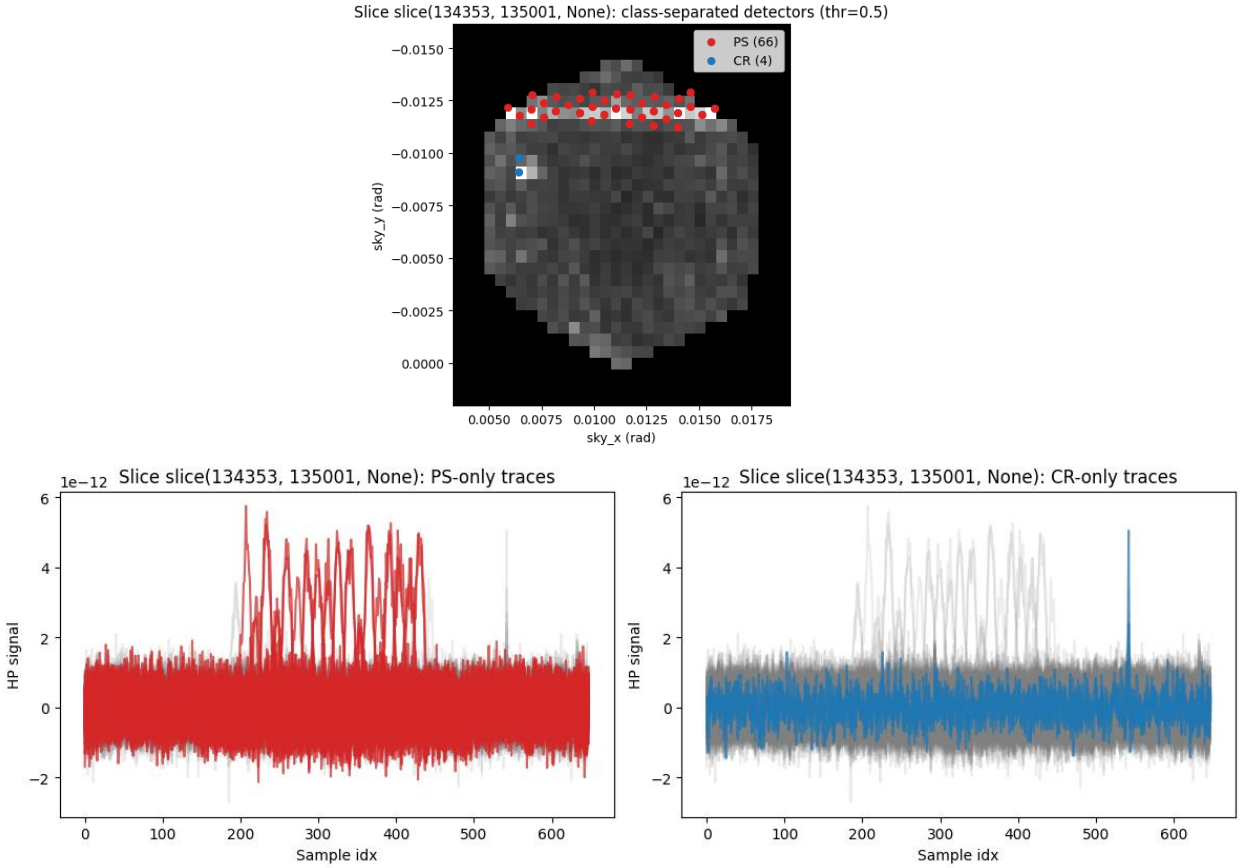


performance metrics for multilabel classification on Max's dataset.

- **Readiness for integration:** Given its ability to separate overlapping events and its robustness on multiple real-data pipelines, the approach is a strong candidate for

integration into production pipelines such as the Simons Observatory data-processing chain.

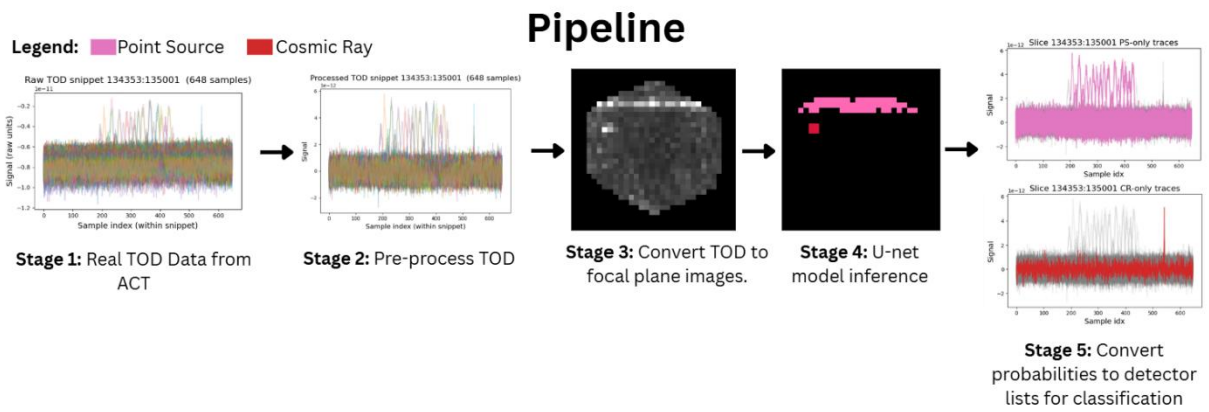
Additionally, we completed the implementation of the detector mapping pipeline, which converts predicted segmentation masks back to individual detectors, including their unique IDs and sky coordinates.



An example of predicted mask mapped back to detector layout.

Pipeline

We use the following real-data pipeline on ACT observations. Starting from synchronous time-ordered data (TOD), we apply standard preprocessing (demean, detrend, calibration, high-pass), window the streams, and rasterize samples onto the focal-plane grid to form single-channel max-exposure images. A lightweight U-Net—trained end-to-end from scratch on real images using pixel masks generated from detector/time metadata, produces per-pixel probabilities for Point Sources (PS), Cosmic Rays (CR), and Electronic Glitches (EL). Predictions are then thresholded, grouped, and mapped back to detector IDs and time indices to yield class-specific detector lists. The figure and steps below walk through these stages at a glance.



The Pipeline

Stage 1: TOD Data

- **Input format.** Each observation is a time-ordered data (TOD) file containing synchronous timestreams for thousands of detectors.
- **Glitches.** Transient events (point sources, cosmic rays, electronic spikes) appear as sharp excursions or step-like features in these per-detector signals.
- **Selection.** We restrict analysis to detectors in the **f090 band** that pass quality cuts (uncut detectors).

Stage 2: Preprocessing

- **Demean / detrend.** Remove constant offsets and linear drifts from each timestream.

- **Calibration.** Apply per-detector calibration factors to bring signals to consistent physical units.
- **High-pass filter.** Apply a digital high-pass filter with cutoff **fc = 2 Hz** (sampling rate **400 Hz**) to suppress slow drifts and preserve short transients (glitches).
- **Detector restriction.** After preprocessing, only uncut f090 detectors are retained for image formation.

Stage 3: Focal-plane images

- **Projection.** Each detector has known sky coordinates on the focal plane. Samples are projected onto a fixed **32×32 grid** that approximates the array geometry.
- **Time windowing.** For each labeled slice start,stop, data are restricted to that interval.
- **Aggregation.** The time axis is collapsed by taking the **maximum value across the window** ("max exposure"), producing a single-channel image where transient responses dominate.
- **Normalization.** Each image is percentile-clipped (0.5–99.5%) and rescaled to 0,1, then stored as an 8-bit grayscale image. Metadata (TOD ID, slice bounds, glitch type) are logged alongside.
-

Stage 4: U-Net Segmentation

- **Architecture.** A compact 2-down / bottleneck / 2-up U-Net processes the 32×32 input image.
- **Output.** The model produces **per-pixel probability maps** for each glitch class: Point Sources (PS), Cosmic Rays (CR), and Electronic Glitches (EL).
- **Multilabel capability.** Each pixel can be assigned to multiple classes simultaneously, enabling the network to capture overlaps (e.g., PS + CR in the same region).

Stage 5: Detector lists

- **Thresholding.** Probability maps are binarized using class-specific thresholds.
- **Grouping.** Contiguous activated pixels are grouped into candidate glitch regions.
- **Mapping back.** Regions are converted from image space to detector IDs and time indices using the focal-plane geometry and slice metadata.

- **Output.** The final product is a per-class list of affected detectors and their corresponding time windows, suitable for downstream quality control.

Challenges and Obstacles Encountered

- **Real data had no pixel masks:** We didn't receive pixel-level annotations for ACT. We generated masks from detector/time metadata by mapping flagged detectors within each slice onto the 32×32 focal-plane grid. These masks enable training but can be incomplete/noisy.
- **Image exposure design (sim vs. real):** How we aggregate time into images matters. On simulations we used two channels [SUM, MAX]; on real ACT we used a single MAX channel with a 2 Hz high-pass (400 Hz sampling) and 0.5–99.5% percentile clipping. This choice improved robustness on real data.
- **Domain gap between simulations and ACT:** Noise/systematics differ across domains. To avoid transferring simulation artifacts, we trained the same lightweight U-Net **from scratch** on real focal-plane images paired with metadata-derived masks.
- **Evaluation without perfect ground truth:** On simulations (with exact masks), we report pixel metrics (e.g., IoU). On real data (with metadata-derived masks), we combine pixel-level scores vs. those masks with event-level checks (compare predicted detector lists to metadata lists). When only image-level labels exist, we fall back to presence/absence via probability+area thresholds.
- **Geometry and resolution constraints:** We restrict to uncut f090 detectors and project to a fixed 32×32 grid for both images and masks. Coarse resolution and missing detectors can blur small/edge events; we mitigate this with consistent mapping, class-specific thresholds, connected-component grouping, and mapping predictions back to detector IDs and times.

Next Steps and Future Work

- The U-Net model is currently trained on 90 GHz data from one detector array.
    - Training on more frequencies and arrays will improve the model's robustness to instrument variations and real-world applications.

- LayerCAM can be used to identify frequency and array-specific features that influence predictions, to improve performance across all SO frequencies and detector arrays.
- Explore video segmentation capabilities.

Resources and Tools Used

- Python, PyTorch
- U-Net segmentation architecture
- Simulated data from the ACT pipeline
- Real ACT focal plane data (converted from TOD)
- Compute resources provided by UofT servers for the segmentation.
- Compute resources provided by Princeton University HPC for the clustering.