Multi-Scale Classification of Green Bank Telescope Signals

Jessica E. Liang[1,3], Ben Jacobson-Bell[2,3,4], Steve Croft[3,5,6]
[1]Department of Computer and Information Science, University of Pennsylvania
[2]Department of Astronomy, University of California, Berkeley
[3]Breakthrough Listen, University of California, Berkeley
[4]Department of Astronomy, Cornell University
[5]Breakthrough Listen, University of Oxford
[6]SETI Institute
[1]jeliang@seas.upenn.edu [2] benjacobsonbell@berkeley.edu [3]scroft@astro.berkeley.edu

Abstract

In this work, we propose a novel pipeline to enhance the detection and mitigation of wide-band Radio Frequency Interference (RFI) in mid-resolution (MR) spectrogram data from the Green Bank Telescope. Our approach integrates an unsupervised Mixture of Experts framework, combining the strengths of multiple edge detection algorithms—Sobel filters, structured forests, Canny edge detection, and the Hough transform—to robustly identify candidate signals in MR data. Leveraging these unsupervised labels, we fine-tune a YOLO-based supervised detection model, significantly enhancing detection efficiency and scalability. Additionally, we embed high-resolution (HR) signals detected by `turboSETI` into a latent representation space using the Vision Transformer (ViT-B16) model, enabling sophisticated matching between MR and HR signals. This approach substantially reduces false-positive technosignature candidates, improving the efficiency of extraterrestrial signal searches. Our method presents a significant advancement in automated signal detection, laying the groundwork for future large-scale technosignature exploration.

1 Introduction

Searching for technosignatures—observable indicators of extraterrestrial technology—requires analyzing massive radio astronomy datasets across multiple temporal and spectral scales. Breakthrough Listen (BL) provides data at multiple resolutions, each with trade-offs between time and frequency coverage [28]. Historically, searches at the Green Bank Telescope (GBT) have relied on High Spectral Resolution (HR) data, with $\sim$3 Hz channels and $\sim$18 s time bins ([10], [22]), ideal for detecting narrowband drifting signals via Doppler-drift analysis.

A limitation of this narrow focus is that many sources of radio-frequency interference (RFI), from terrestrial or satellite origins, span kilohertz to megahertz and show temporal variability. Relying solely on fine resolutions risks incomplete characterization or missed detection of such complex RFI, obscuring genuine technosignatures.

We propose a multi-scale classification framework for GBT data, extendable to HR products from other facilities such as the Allen Telescope Array and Parkes. Our approach combines HR detection with mid-resolution (MR) morphological analysis ($\sim$3 kHz, 1 s), balancing sensitivity to narrowband signals with broader contextual clarity. The pipeline proceeds in four stages: (1) candidate detection using HR data and Doppler-drift algorithms such as `turboSETI` [16]; (2) MR cutouts for

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Science.

morphological characterization (broadband vs. narrowband, impulsive vs. continuous); (3) clustering and association of HR and MR hits, demonstrated on the 97-galaxy survey of [10], with attention to the RFI-rich 1500–1650 MHz band; and (4) systematic exclusion of HR detections identified as wide-band RFI in MR data.

By integrating HR and MR through morphological analysis and machine learning—unsupervised edge detection and deep learning [31, 22]—our framework strengthens RFI discrimination and candidate reliability. This multi-resolution approach expands the diversity of observed phenomena and improves the robustness of SETI observations.

2 Related Work

The search for technosignatures has driven substantial advances in data collection, infrastructure, and analysis. The Breakthrough Listen initiative [28, 51] established standardized public datasets supporting reproducibility and large-scale collaboration, while the Exotica Catalog [27] broadened observational targets. Large-scale extragalactic surveys, such as the 97-galaxy project yielding over six million hits [10], underscore both scientific promise and the growing challenge of satellite interference.

With increasing data volume and complexity, machine learning has become central to technosignature detection and false-positive reduction. Deep networks improve sensitivity for narrowband searches [31], and transfer learning on CNNs (*FETCH*) enables real-time FRB classification [1]. Unsupervised methods [22] further aid anomaly and RFI detection without labeled data. Computer vision (CV) approaches—edge detection [8, 13] and the Mixture of Experts framework [45]—have been adapted to identify and classify diverse signal morphologies in SETI data.

Object detection models such as YOLO [44] have been repurposed for candidate identification in spectrograms [50], with multi-scale and specialized variants boosting accuracy [9, 49, 29]. Similar models benefit agriculture and small object detection [2, 21, 57], while CV adaptations to time–frequency data [35] enhance scalability for SETI.

Edge detection remains foundational. The Canny detector excels for faint features [24, 47, 53], with CUDA and shape-adaptive improvements [34, 38]. The Sobel operator has likewise been extended for denoising, acceleration, and 3D data [17, 23, 4].

Spectrogram-based representations are now standard across audio, biomedical, and radar domains [3]. Deep architectures achieve state-of-the-art performance [20, 56, 37, 25, 12], strengthened by multi-task learning and advanced feature extraction [54, 35, 11, 43, 26]. Transfer learning and specialized models [42, 40, 5] demonstrate flexibility across domains.

Finally, the Hough Transform remains indispensable for astronomical feature detection, extracting lines and curves from noisy data [6]. Applications span arcs [19], solar profiles [33], artifact removal [46], crescent detection [36], 3D tracking [55], and asteroid events [41, 30, 18].

Together, these developments highlight the value of treating radio data as images and applying modern CV and ML to SETI, improving scalability, efficiency, and scientific yield.

3 Proposed Method

While `turboSETI` is the de-facto engine for BL narrow-band searches, its sensitivity degrades on the mid-resolution (MR, $\sim$2.86 kHz) product compared to high-resolution (HR, $\sim$2.84 Hz) spectra [28]. A beacon confined to a few-hertz HR bin is spread over $\Delta f_{\rm mid}/\Delta f_{\rm fine} \approx 10^3$ MR channels, so per-channel S/N drops by the radiometer factor $\sqrt{\Delta f_{\rm mid}/\Delta f_{\rm fine}} \approx 32$. Thus the HR-tuned detection cut of $\rho_{\rm tseti} = 10$ misses many MR beacons. This setting corresponds to an effective $\mathrm{S/N} \simeq 33$ after correcting `turboSETI`'s noise estimate [10], so lowering the MR threshold would inflate false alarms.

We introduce a computer vision pipeline for RFI detection. Each scan is preprocessed by extracting one coarse channel, input to a YOLO model. To remove the DC spike at channel center, we replace the central column with the average of its neighbors, improving normalization and signal visibility. We then employ an unsupervised Mixture of Experts to identify signals; these outputs can also train YOLO for efficiency, though here we focus on the unsupervised stage.

Signals found in MR are cross-matched with HR hits from `turboSETI`. Spectrogram cutouts are embedded with a Vision Transformer (ViT-B16) [14]. When an HR hit lies inside an MR wideband detection, it is treated as substructure and discarded, reducing the candidate haystack. Figure 1 visualizes this pipeline. Experiments ran on two multi-GPU servers: one with dual Intel Xeon Silver 4210 CPUs (40 logical cores), 188 GB RAM, and four RTX A4000 GPUs (16 GB each); the other with dual AMD EPYC 7313 CPUs (32 cores), 504 GB RAM, and the same GPUs. Both used `CUDA 12.4` with driver 550.78.

3.1 Unsupervised RFI Detection

In order to successfully train a YOLO model, we begin by labeling each coarse channel image with boxes or regions of interest with RFI. In the Appendix, we evaluate a simple thresholding method to extract regions of interest. Due to poor performance, we shift to more complex methods of unsupervised edge detection. Common existing methods include Sobel filters, structured forests, Canny edge detection, and Hough transform.

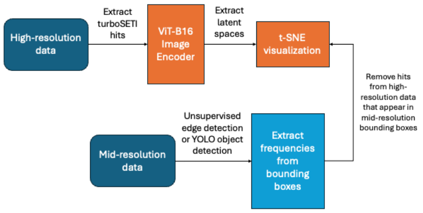

High-resolution data
Extract turboSETI hits
ViT-B16 Image Encoder
Extract latent spaces
t-SNE visualization
Remove hits from high-resolution data that appear in mid-resolution bounding boxes
Mid-resolution data
Unsupervised edge detection or YOLO object detection
Extract frequencies from bounding boxes

Figure 1: Workflow to incorporate MR data to remove wide-band RFI

Sobel-based Unified Edge–Density (UED) Map. The Sobel operator uses 3 × 3 kernels

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = S_x^\top$$

to approximate first-order derivatives along time and frequency axes. For a grayscale spectrogram I, we compute

$$G_x = \texttt{Sobel}(I, \texttt{CV32F}, 1, 0, 3), \quad G_y = \texttt{Sobel}(I, \texttt{CV32F}, 0, 1, 3),$$

with 32-bit precision. The gradient magnitude

$$M_{\text{UED}}(x, y) = \sqrt{G_x^2 + G_y^2}$$

highlights rapid intensity changes.

After min–max normalization to $[0, 1]$, the result is the UED map used in fusion and blob detection. Sobel UED is fast and preserves edge strength, though contours may be thick and noise-sensitive. Computations used `OpenCV 4.9`/`NumPy 1.26`, finishing in $\sim$1 hr on a 40-core Xeon server.

Structured–Forest Edge Map (SF). Structured forests [13] employ a supervised random-forest that predicts small binary edge masks directly from image (or spectrogram) patches; averaging those patch-level votes yields a dense probability map that is remarkably tolerant of textured backgrounds and subtle RFI artifacts, yet still runs in real time. In our pipeline we apply the pre-trained model shipped with `OpenCV's ximgproc` module as follows. Given a raw spectrogram block $B \in \mathbb{R}^{H \times W}$, we scale it to $[0, 1]$

$$I = \frac{B - \min B}{\text{ptp}(B) + 10^{-6}},$$

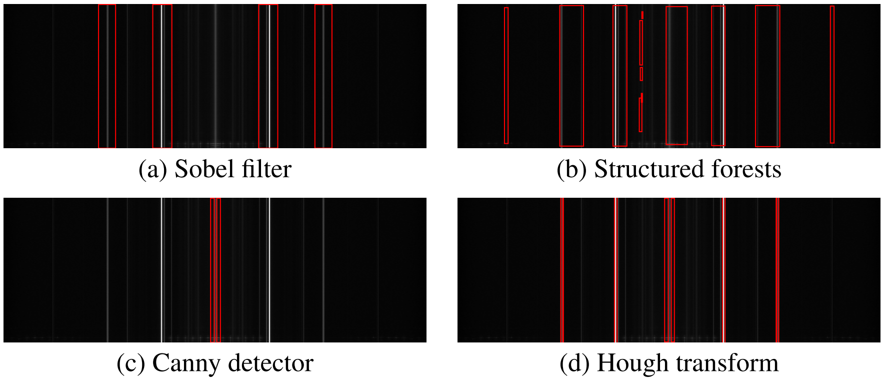
replicate the single channel to form a float-32 pseudo-RGB tensor $I_{\text{RGB}} \in [0, 1]^{H \times W \times 3}$, and obtain the SF edge probability map via

$$M_{\text{SF}} = \texttt{detectEdges}(I_{\text{RGB}}) \in [0, 1]^{H \times W}.$$

Experiments finished in 1.5 hours on a 40-core Intel Xeon Silver 4210 CPU with 188 GB RAM.

Canny Edge Detector. To obtain thin yet contiguous edge regions, we employ the Canny detector followed by contour filling, implemented in Python with `opencv-python` (`cv2`) and `NumPy`. Starting from an 8-bit spectrogram frame I_8, we first suppress impulsive noise with a 5×5 median filter, then apply Canny hysteresis with fixed thresholds $T_{\text{low}} = 50$ and $T_{\text{high}} = 150$:

$$E = \texttt{Canny}\big(\texttt{medianBlur}(I_8, 5), \, T_{\text{low}}, T_{\text{high}}\big)$$

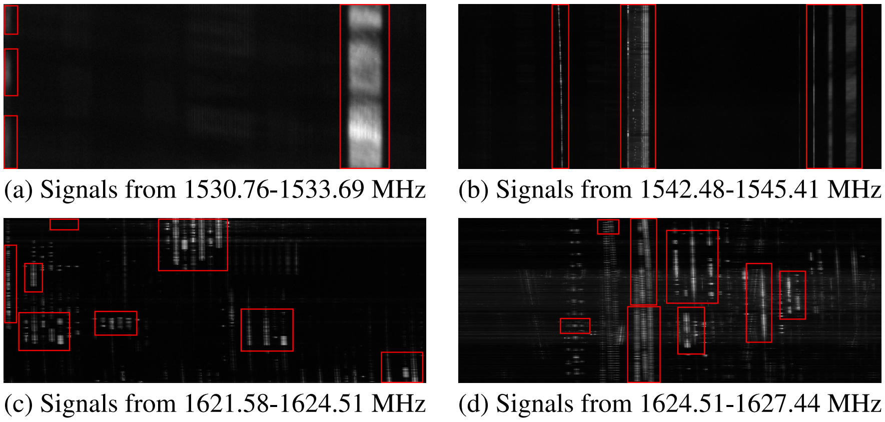

(a) Sobel filter

(b) Structured forests

(c) Canny detector

(d) Hough transform

Figure 2: Comparison of edge detection methods on the same spectrogram: (a) Sobel, (b) Canny, (c) Structured forests, (d) Hough transform. Red bounding boxes mark time–frequency regions that the method flagged as candidate signals.


(a) Signals from 1530.76-1533.69 MHz

(b) Signals from 1542.48-1545.41 MHz

(c) Signals from 1621.58-1624.51 MHz

(d) Signals from 1624.51-1627.44 MHz

Figure 3: Qualitative behavior of the Mixture of Experts algorithm on four representative mid-resolution frames. Red bounding boxes mark time–frequency regions that the method flagged as candidate signals.

and $E \in \{0,1\}^{H\times W}$.

The internal Canny stages (Gaussian smoothing, Sobel gradients, non-maximum suppression, and double-threshold linkage) yield a sub-pixel-accurate binary edge map E. To convert these one-pixel traces into dense blobs we extract all external contours $\mathcal{C} = \texttt{findContours}(E, \texttt{RETR\_EXTERNAL})$ and rasterize them, filled, onto a blank mask $M_{\text{CM}} \in \{0,1\}^{H\times W}$ via $\texttt{drawContours}(M_{\text{CM}}, \mathcal{C}, -1, 1, \texttt{thickness} = -1)$. The resulting Canny–Contour Mask preserves complete signal tracks while excluding isolated noise fragments, making it a robust input for the subsequent multi-map fusion stage. Experiments were completed in 1 hour on a dual 40-core Intel Xeon Silver 4210 CPU system with 188 GB RAM.

Hough Transform. The Hough transform converts edge pixels (t, f) in the time–frequency plane into votes in a parametric space (drift rate, intercept); peaks in that space reveal straight, narrow-band tracks even when partially occluded. We implement its light-weight variant `probabilistic_hough_line` from `scikit-image` on the Canny edge mask E. Using a vote threshold of 5, a minimum segment length of 10 px, and a maximum gap of 2 px yields a set of line segments $\mathcal{L} = \{((y_0, x_0), (y_1, x_1))\}$. Each segment is rasterized at unit intensity onto a blank float-32 canvas M_{PH} via `cv2.line`, and a 3×3 Gaussian blur smooths the accumulator:

$$M_{\text{PH}} \leftarrow \texttt{GaussianBlur}(M_{\text{PH}}, 3).$$

The resulting Probabilistic Hough map highlights constant-drift or Doppler-corrected signals while suppressing isolated noise. Experiments finished in 2 hours on a 40-core Intel Xeon Silver 4210 CPU with 188 GB RAM.

Figure 2 (with additional cases in the Appendix) compares four edge maps on mid-resolution frames. Expert blind review shows trends consistent with each detector's design. Sobel (UED) highlights strong local gradients, excelling on large high-contrast events but amplifying noise into many false positives. Structured Forest (SF) integrates patch context, recovering thin straight drifts, but fragments curved or hopping tracks. Canny-Contour Mask (CM) produces razor-thin edges and suppresses noise via hysteresis, though meandering signals may self-intersect, splitting into multiple blobs. The Probabilistic Hough (PH) transform is highly specific for constant-drift lines, but frequency hops or curvature break tracks into disjoint detections, lowering recall. These complementary error modes motivate their weighted fusion.

3.2 Mixture of Experts

Since each of our four unsupervised edge detection methods have different strengths and weaknesses, we propose mixing these methods in order to minimize both false positives and negatives in our identification process. Let $M_{\mathrm{SF}}, M_{\mathrm{UED}}, M_{\mathrm{CM}}, M_{\mathrm{PH}} \in [0,1]^{H\times W}$ be the four edge maps described earlier. We first form a weighted linear fusion

$$M_{\text{fuse}} \;=\; w_{\mathrm{SF}}M_{\mathrm{SF}} + w_{\mathrm{UED}}M_{\mathrm{UED}} + w_{\mathrm{CM}}M_{\mathrm{CM}} + w_{\mathrm{PH}}M_{\mathrm{PH}},$$

where uniform weights $(w_{\mathrm{SF}}, w_{\mathrm{UED}}, w_{\mathrm{CM}}, w_{\mathrm{PH}}) = (0.25, 0.25, 0.25, 0.25)$ are used. The fused map is standardized pixel-wise, $Z = (M_{\text{fuse}} - \mu)/(\sigma + 10^{-8})$, with μ, σ the global mean and standard deviation, and thresholded at $Z > 2$ to produce a binary mask B.

Connected components in B are labeled (`ndi.label`) and all regions smaller than 50 pixels are discarded. To merge neighboring blobs representing the same physical track we perform a separable dilation with horizontal and vertical structuring elements of length $k = 20$ (i.e. gaps ≤ 2 px are bridged), take the union of the two dilations, and apply a 3×3 binary closing to refill small holes:

$$B \xrightarrow{\text{dilate }(1\times k,\ k\times 1)} B' \xrightarrow{\text{close }3\times 3} B_{\text{filled}}$$

Axis-aligned YOLO-style bounding boxes are then extracted from B_{filled} and overlapping boxes are merged. Finally, for each candidate box we compute the mean brightness of its corresponding patch in the original spectrogram block; a box is kept only if that patch is no more than 10% darker than the global mean, i.e. $\bar{I}_{\text{patch}} \geq 0.9\,\bar{I}_{\text{global}}$. This brightness gate suppresses spurious detections on exceptionally dark background regions. Experiments finished in 110 hours on a 40-core Intel Xeon Silver 4210 CPU with 188 GB RAM.

3.3 Evaluation of Edge Detection

To ensure accuracy of bounding boxes, we only evaluate coarse channels with kurtosis values of at least 7.0; channels with a smaller kurtosis value are likely to be purely noise, so we disregard these channels for our study. Figure 3 shows a subsample of images used for qualitative analysis.

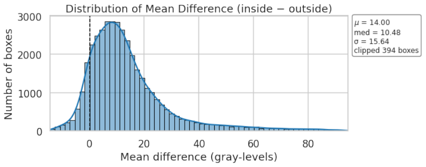


Figure 4: Histogram showing mean difference between the inside and outside of bounding boxes created by our Mixture of Experts algorithm.

To quantify detection quality, we plot the pixel-intensity difference between each bounding box and its surrounding region. A well-placed box should yield a positive distance from zero. Figure 4 shows the resulting histogram after clipping the outer 0.5% of values for legibility. The distribution is skewed to the right with a mean of $\mu = 14.0$ and a median of 10.48, indicating that most boxes indeed enclose brighter signal energy than their background. Negative tails remain, caused by occasional false positives (dimmer interiors), missed faint signals (brighter exteriors), or very weak detections that reduce the inside–outside contrast. Based on our qualitative and quantitative analysis, we use this algorithm for all signal extraction from mid-resolution data. These labels can also be used as input to a YOLO model for improved generalization and speed in future work.

3.4 YOLO Signal Detection

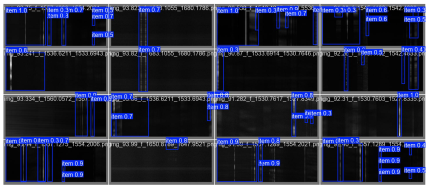


Figure 5: Validation batch with predicted signals labeled

The automatically labeled coarse–channel crops are used to fine-tune the *Nano* variant of Ultralytics **YOLOv11** (`yolo11n.pt`, COCO-pretrained). We create a random 80-20 training and validation split from our dataset of 8,037 images and corresponding labels extracted from our unsupervised Mixture of Experts model. For deterministic behavior we fix the global seed to 42 in Python, NumPy, PyTorch, and pass the same value to YOLO's internal RNG. Training runs for 100 epochs with SGD at an initial learning-rate of 10^{-3}, a 5-epoch linear warm-up, momentum 0.937, and weight-decay $5{\times}10^{-4}$. Mini-batches contain 32 rectangular images (`rect=True`); an out-of-memory back-off loop halves the batch size on the fly if the GPU exhausts memory. Data augmentation is kept minimal to preserve temporal structure: mosaic is disabled, horizontal flips are turned off (`mosaic=False`, `fliplr=0`), and training proceeds in single-class mode (`single_cls=True`). The classification, distribution-focal, and IoU loss terms are each weighted 0.3, and early stopping triggers after 20 stagnant epochs. Figure 5 shows one of the validation batches output with labels for identified signals. The YOLO model was trained in under 1 hour on a dual AMD EPYC 7313 system with 32 CPU cores, 504 GB of memory, and four NVIDIA RTX A4000 GPUs (16 GB each).

4 Extracting High-Resolution Hits

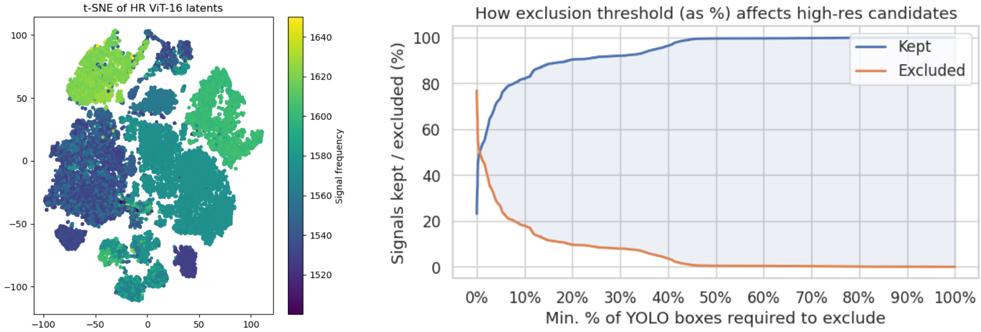
In order to match hits across resolutions, we use hits output from `turboSETI` on HR data that fall in our range of interest (1500-1650 MHz). To improve visualization, each row is upsampled such that every image of a hit has 279 time bins, matching the time dimension of mid-resolution files. This resampling is purely for human interpretability, as all images are later resized to a fixed size during vision-model pre-processing. Hits are also extracted with a buffer of 4 frequency bins on each side, ensuring the full hit is extracted and providing additional spectral context and better image quality.

Extracting the latent space of hits, we adopt the **Vision Transformer Base** with 16×16 patches (**ViT-B/16**) [14] using the pretrained `ViT_B_16_Weights.IMAGENET1K_V1` checkpoint distributed by TorchVision. The network is kept frozen with no additional fine-tuning and serves purely as a feature extractor. ViT-B/16 contains 12 Transformer encoder layers, each with hidden dimension 768 and 12-head self-attention, for a total of $\sim$ 86 M parameters. The published weights obtain 81.1 % top-1 (95.3 % top-5) accuracy on ImageNet-1K. Before feeding a crop to ViT, we resize the shorter side to 256 px, center-crop to 224×224 px, replicate the single-channel spectrogram to RGB, convert the image to a float tensor in $[0, 1]$, and apply the standard ImageNet normalization ($\boldsymbol{\mu}{=}(0.485, 0.456, 0.406)$, $\boldsymbol{\sigma}{=}(0.229, 0.224, 0.225)$). This preprocessing conforms each crop to the input geometry and color-channel statistics expected by the pretrained ViT, ensuring full compatibility without retraining the early layers.

We reduce the $N \times D$ ViT feature matrix, where N is the number of time–frequency samples (spectrogram patches) and $D{=}768$ is the original dimensionality of each ViT feature vector, using principal–component analysis (PCA). Employing scikit-learn's `PCA` with `random_state=42` yields a deterministic 32-dimensional latent embedding $\mathbf{Z} \in \mathbb{R}^{N \times 32}$ that is used for all subsequent visualization and clustering steps.

The resulting vectors can be visualized using t-SNE. We row-wise ℓ_2-normalized the 32-dimensional ViT embeddings and projected them with Barnes–Hut t-SNE (scikit-learn $1.x$) using $n_{\text{components}}{=}2$, perplexity $=30$, early_exaggeration $=12$ (default), metric = Euclidean, and a fixed random seed of 42.

We evaluate the resulting plot shown in Figure 6a. Although the frequency of signals is not incorporated into the latent space, the color map is based on the start frequency of the signal which we extract from the corresponding file name. Signals that originate from the same transmitter tend to look alike and therefore group into color-consistent clusters on the t-SNE map. However, multiple services can share a band (e.g. Wi-Fi and Bluetooth at 2.4 GHz), and even a "single-allocation" band may have several emitters whose fine-resolution morphologies differ. A prominent example is the bright green cluster near 1620 MHz; this frequency range (1610–1626.5 MHz) is allocated to the Iridium satellite constellation [22], indicating that the clustered points capture visually similar Iridium signatures. Since the clusters in this plot are distinguished relatively well between colors, we see that the latent spaces are accurate representations of signals in HR spectrograms. We confirm this by visual inspection of a random sample of signals from each cluster. Experiments were completed in approximately five minutes using four NVIDIA RTX A4000 GPUs (16 GB each).

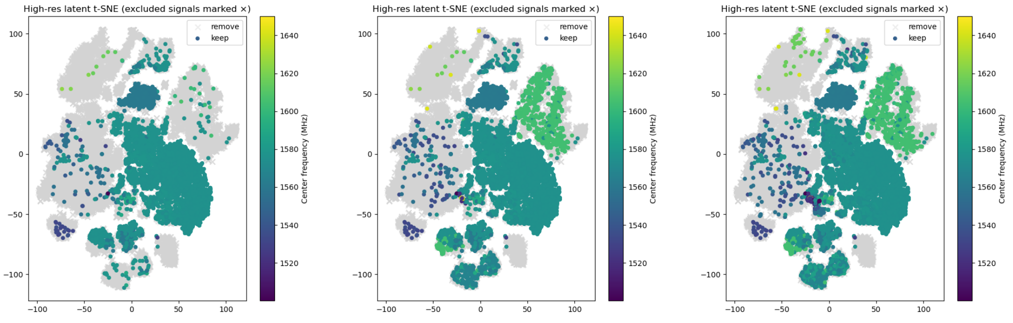


(a) (b)

Figure 6: (a) t-SNE visualization of latent space for hits in HR spectrograms found using `turboSETI`. (b) Relation between exclusion threshold (percentage) and proportion of signals kept or excluded.

5 Reducing the Haystack

Using the hits extracted from MR data, we can use their frequency ranges in order to find where broadband RFI is located. Based on the bounding boxes found from our Mixture of Experts model, we can extract the exact frequencies of these signals based on the start and end frequencies of the coarse channel. Each hit from HR data found using `turboSETI` has a center frequency associated, which can be used to match to signals at MR. We can adjust the threshold at which a signal is considered RFI and removed from the haystack. Figure 7 demonstrates these results side-by-side. With a strict overlap criterion, discarding any HR hit whose frequency range intersects at least one MR bounding box, we remove 29,212 of the 37,996 HR candidates (77%). Relaxing the requirement so that a hit is discarded only if it overlaps five or ten MR boxes lowers the rejected set to 27,605 (73%) and 26,214 (69%) candidates, respectively. Figure 6b plots this inverse relationship between the overlap threshold and the number of discarded signals, allowing users to tune the threshold to balance completeness against false-positive suppression.

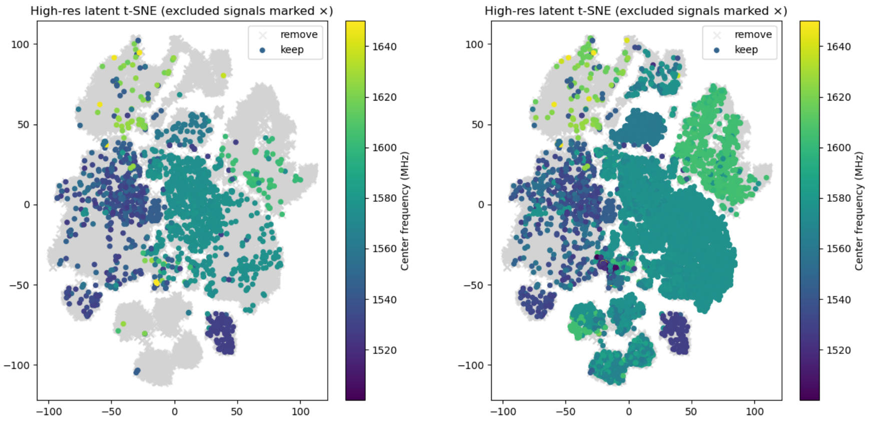


(a) (b) (c)

Figure 7: Comparison of varying overlap thresholds for signal removal. (a) At least 1 MR signal overlap: removes 6163 signals, (b) At least 5 MR signals overlap: removes 5457 signals, (c) At least 10 MR signals overlap: removes 4376 signals.

We can further analyze the original latent space (Figure 6a) in comparison with the plots in Figure 7. The original bright green cluster belonging to frequencies at around 1620 MHz, most likely Iridium signals, are excluded in all three plots in Figure 7, meaning that almost all of them overlap with at least 10 signals found in MR data. Since Iridium signals are a common form of broadband RFI, we see that this methodology is able to remove a large amount of hits from HR data that are most likely attributed to RFI. This allows us to reduce the number of signals examined in our search for technosignatures.

In addition to counting MR–HR overlaps, we incorporate latent-space densities to preserve outliers that may represent rare or transient phenomena. Gaussian Kernel Density Estimation (KDE) estimates point densities in the latent space, enabling identification of low-density signals [39]. We train a Gaussian KDE (`sklearn.neighbors.KernelDensity`) on a random subsample of up to $20{,}000$ points, with bandwidth chosen by `GridSearchCV` over $[0.05, 1.5]$ using three-fold cross-validation. The fitted KDE scores all points, yielding log-densities exponentiated to density-like values. An anomaly threshold is set at the 5% log-density quantile (`numpy.quantile`), flagging the lowest-density 5% as sparse or anomalous. In the reported run, the best bandwidth was 0.05, identifying $\sim$1,900 low-density points. Figure 8a shows that preserving only these points removes 36,096 signals.



(a)

(b)

Figure 8: Signal removal process. (a) Conditioning on density: 36,096 signals removed, (b) Final plot: 25,015 signals removed.

We can combine these conditions such that signals that appear in at least 10 MR bounding boxes and are not flagged as low density get excluded. Figure 8b visualizes this final result, and we remove 25,015 of the 37,996 candidates. In the Appendix we examine matching latent spaces and clusters of signals in high-resolution and mid-resolution data, which can help us further understand hits found for high-resolution spectrograms.

6 Conclusions and Future Work

In this paper, we presented a novel pipeline that leverages unsupervised edge detection and advanced computer vision techniques to enhance the detection and rejection of wide-band Radio Frequency Interference (RFI) in mid-resolution (MR) spectrogram data from the GBT. We developed a Mixture of Experts (MoE) framework integrating Sobel filters, structured forests, Canny edge detection, and the Hough transform to robustly identify candidate signals. These unsupervised labels were validated both qualitatively and quantitatively, demonstrating accurate localization of signals within spectrograms. Using these labels, we fine-tuned a YOLO-based supervised model for efficient, generalizable detection. We then extracted high-resolution (HR) hits from corresponding `turboSETI` data. Embedding HR signals into a ViT-B16 latent space enabled cross-resolution matching, allowing the removal of HR hits likely originating from wide-band RFI and significantly reducing the number of candidate technosignatures requiring further scrutiny.

Future work will focus on several directions to improve robustness and scalability. First, we will also expand YOLO training to larger and more diverse datasets, including observations under varied noise and instrumental conditions, to enhance generalization across telescope configurations. Finally, we aim to incorporate temporal coherence and spatial context through advanced modeling techniques, such as recurrent networks or transformer-based architectures, to better capture complex time–frequency dynamics in signal evolution.

Acknowledgments

The Breakthrough Prize Foundation funds the Breakthrough Initiatives, which manages Breakthrough Listen. The Green Bank Observatory facility is supported by the National Science Foundation, and is operated by Associated Universities, Inc., under a cooperative agreement. JEL was funded as a participant in the Berkeley SETI Research Center Research Experience for Undergraduates Site, supported by the National Science Foundation under Grant No. 2244242.

References

[1] Devansh Agarwal, Kshitij Aggarwal, Sarah Burke-Spolaor, Duncan R Lorimer, and Nathaniel Garver-Daniels. Fetch: A deep-learning based classifier for fast transient classification. *Monthly Notices of the Royal Astronomical Society*, 497(2):1661–1674, 06 2020.

[2] Yousef Alhwaiti, Muntazir Khan, Muhammad Asim, Muhammad Hameed Siddiqi, Muhammad Ishaq, and Madallah Alruwaili. Leveraging yolo deep learning models to enhance plant disease identification. *Scientific Reports*, 15(1):7969, 2025.

[3] Richard A Altes. Detection, estimation, and classification with spectrograms. *The Journal of the Acoustical Society of America*, 67(4):1232–1246, 1980.

[4] Haroon Ashraf, Wail A Mousa, and Saleh Al Dossary. Sobel filter for edge detection of hexagonally sampled 3d seismic data. *Geophysics*, 81(6):N41–N51, 2016.

[5] Ahmad Azab and Mahmoud Khasawneh. Msic: Malware spectrogram image classification. *IEEE access*, 8(102):007–102, 2020.

[6] Pascal Ballester. Hough transform and astronomical data analysis. *Vistas in Astronomy*, 40(4):479–485, 1996.

[7] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[8] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 2009.

[9] Yuming Chen, Xinbin Yuan, Jiabao Wang, Ruiqi Wu, Xiang Li, Qibin Hou, and Ming-Ming Cheng. Yolo-ms: Rethinking multi-scale representation learning for real-time object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[10] Carmen Choza, Daniel Bautista, Steve Croft, Andrew PV Siemion, Bryan Brzycki, Krishnakumar Bhattaram, Daniel Czech, Imke de Pater, Vishal Gajjar, Howard Isaacson, et al. The breakthrough listen search for intelligent life: Technosignature search of 97 nearby galaxies. *The Astronomical Journal*, 167(1):10, 2023.

[11] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.

[12] Jonathan Dennis, Huy Dat Tran, and Haizhou Li. Spectrogram image feature for sound event classification in mismatched conditions. *IEEE signal processing letters*, 18(2):130–133, 2010.

[13] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1841–1848, 2013.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[16] Vishal Gajjar, Andrew Siemion, Steve Croft, Bryan Brzycki, Marta Burgay, Tobia Carozzi, Raimondo Concu, Daniel Czech, David Deboer, Julia DeMarines, Jamie Drew, J. Enriquez, James Fawcett, Peter Gallagher, Michael Gerret, Nectaria Gizani, Greg Hellbourg, Jamie Holder, Howard Isaacson, and Yunfan Zhang. The breakthrough listen search for extraterrestrial intelligence. *10.48550/arXiv.1907.05519*, 07 2019.

[17] Wenshuo Gao, Xiaoguang Zhang, Lei Yang, and Huizhong Liu. An improved sobel edge detection. In *2010 3rd International conference on computer science and information technology*, volume 5, pages 67–71. IEEE, 2010.

[18] Boliang He and Yongheng Zhao. Determining the observation epochs of star catalogs from ancient china using the generalized hough transform method. *arXiv preprint arXiv:2504.02182*, 2025.

[19] Christopher Hollitt and Melanie Johnston-Hollitt. Feature detection in radio astronomy using the circle hough transform. *Publications of the Astronomical Society of Australia*, 29(3):309–317, 2012.

[20] Jingshan Huang, Binqiang Chen, Bin Yao, and Wangpeng He. Ecg arrhythmia classification using stft-based spectrogram and convolutional neural network. *IEEE access*, 7:92871–92880, 2019.

[21] Yiqi Huang, Zhenhao Liu, Hehua Zhao, Chao Tang, Bo Liu, Zaiyuan Li, Fanghao Wan, Wanqiang Qian, and Xi Qiao. Yolo-ysts: An improved yolov10n-based method for real-time field pest detection. *Agronomy*, 15(3):575, 2025.

[22] Ben Jacobson-Bell, Steve Croft, Carmen Choza, Alex Andersson, Daniel Bautista, Vishal Gajjar, Matthew Lebofsky, David HE MacMahon, Caleb Painter, and Andrew PV Siemion. Anomaly detection and radio-frequency interference classification with unsupervised learning in narrowband radio technosignature searches. *The Astronomical Journal*, 169(4):206, 2025.

[23] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[24] Sergii V Khlamov, Iryna Tabakova, Tetiana O Trunova, and Zhanna V Deineko. Machine vision for astronomical images using the canny edge detector. In *IT&I Workshops*, pages 1–10, 2022.

[25] Peerapol Khunarsal, Chidchanok Lursinsap, and Thanapant Raicharoen. Very short time environmental sound classification based on spectrogram pattern matching. *Information Sciences*, 243:57–74, 2013.

[26] Elly C Knight, Sergio Poo Hernandez, Erin M Bayne, Vadim Bulitko, and Benjamin V Tucker. Pre-processing spectrogram parameters improve the accuracy of bioacoustic classification using convolutional neural networks. *Bioacoustics*, 29(3):337–355, 2020.

[27] Brian C Lacki, Bryan Brzycki, Steve Croft, Daniel Czech, David DeBoer, Julia DeMarines, Vishal Gajjar, Howard Isaacson, Matt Lebofsky, David HE MacMahon, et al. One of everything: The breakthrough listen exotica catalog. *The Astrophysical Journal Supplement Series*, 257(2):42, 2021.

[28] Matthew Lebofsky, Steve Croft, Andrew PV Siemion, Danny C Price, J Emilio Enriquez, Howard Isaacson, David HE MacMahon, David Anderson, Bryan Brzycki, Jeff Cobb, et al. The breakthrough listen search for intelligent life: public data, formats, reduction, and archiving. *Publications of the Astronomical Society of the Pacific*, 131(1006):124505, 2019.

[29] Yuan Liu, Yilong Liu, Xiaoyan Guo, Xi Ling, and Qingyi Geng. Metal surface defect detection using slf-yolo enhanced yolov8 model. *Scientific Reports*, 15(1):11105, 2025.

[30] Kai-Jie Lo, Chan-Kao Chang, Hsing-Wen Lin, Meng-Feng Tsai, Wing-Huen Ip, Wen-Ping Chen, Ting-Shuo Yeh, KC Chambers, EA Magnier, ME Huber, et al. Asteroid discovery and light curve extraction using the hough transform: A rotation period study for subkilometer main-belt asteroids. *The Astronomical Journal*, 159(1):25, 2019.

[31] Peter Xiangyuan Ma, Cherry Ng, Leandro Rizk, Steve Croft, Andrew PV Siemion, Bryan Brzycki, Daniel Czech, Jamie Drew, Vishal Gajjar, John Hoang, et al. A deep-learning search for technosignatures from 820 nearby stars. *Nature Astronomy*, 7(4):492–502, 2023.

[32] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.

[33] Anna Maria Massone, Annalisa Perasso, Cristina Campi, and Mauro C Beltrametti. Profile detection in medical and astronomical images by means of the hough transform of special classes of curves. *Journal of Mathematical Imaging and Vision*, 51(2):296–310, 2015.

[34] Lesia I Mochurad. Canny edge detection analysis based on parallel algorithm, constructed complexity scale and cuda. *Computing and Informatics*, 41(4):957–980, 2022.

[35] Loris Nanni, Andrea Rigo, Alessandra Lumini, and Sheryl Brahnam. Spectrogram classification using dissimilarity space. *Applied Sciences*, 10(12):4176, 2020.

[36] ALAM Nasir, R Umar, WNJW Yussof, N Ahmad, AN Zulkeflee, NH Sabri, WMYW Chik, NA Mahiddin, AH Bely, and AIS Izdihar. New crescent moon detection using circular hough transform (cht). *Astronomy and Computing*, 51:100902, 2025.

[37] Zakaria Neili and Kenneth Sundaraj. A comparative study of the spectrogram, scalogram, melspectrogram and gammatonegram time-frequency representations for the classification of lung sounds using the icbhi database based on cnns. *Biomedical Engineering/Biomedizinische Technik*, 67(5):367–390, 2022.

[38] Karen A Panetta, Sos S Agaian, Shahan C Nercessian, and Ali A Almunstashri. Shape-dependent canny edge detector. *Optical Engineering*, 50(8):087008–087008, 2011.

[39] Snir Pardo, Dovi Poznanski, Steve Croft, Andrew P. V. Siemion, and Matthew Lebofsky. Using anomaly detection to search for technosignatures in breakthrough listen observations. *The Astronomical Journal*, 170(1):12, jun 2025.

[40] Dongsuk Park, Seungeui Lee, SeongUk Park, and Nojun Kwak. Radar-spectrogram-based uav classification using convolutional neural networks. *Sensors*, 21(1):210, 2020.

[41] Ritesh Patel, Vaibhav Pant, Priyanka Iyer, Dipankar Banerjee, Marilena Mierla, and Matthew J West. Automated detection of accelerating solar eruptions using parabolic hough transform. *Solar Physics*, 296(2):31, 2021.

[42] Sai Samarth R Phaye, Emmanouil Benetos, and Ye Wang. Subspectralnet–using sub-spectrogram based convolutional neural networks for acoustic scene classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 825–829. IEEE, 2019.

[43] Priyanshu Rawat, Madhvan Bajaj, Satvik Vats, and Vikrant Sharma. A comprehensive study based on mfcc and spectrogram for audio classification. *Journal of Information and Optimization Sciences*, 44(6):1057–1074, 2023.

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[45] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[46] AJ Storkey, NC Hambly, CKI Williams, and Robert G Mann. Cleaning sky survey data bases using hough transform and renewal string approaches. *Monthly Notices of the Royal Astronomical Society*, 347(1):36–51, 2004.

[47] Wei Tang, Li-Chun Zhu, and Qi-Ming Wang. Recognition of fast reflector nodes based on canny operator. *Research in Astronomy and Astrophysics*, 20(8):126, 2020.

[48] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on neural networks*, 11(3):586–600, 2000.

[49] Dingran Wang, Jiasheng Tan, Hong Wang, Lingjie Kong, Chi Zhang, Dongxu Pan, Tan Li, and Jingbo Liu. Sds-yolo: An improved vibratory position detection algorithm based on yolov11. *Measurement*, 244:116518, 2025.

[50] Luke Wood, Kevin Anderson, Peter Gerstoft, Richard Bell, Raghab Subbaraman, and Dinesh Bharadia. Deep learning object detection approaches to signal identification, 2022.

[51] S. Pete Worden, Jamie Drew, Andrew Siemion, Dan Werthimer, David DeBoer, Steve Croft, David MacMahon, Matt Lebofsky, Howard Isaacson, Jack Hickish, Danny Price, Vishal Gajjar, and Jason T. Wright. Breakthrough listen – a new search for life in the universe. *Acta Astronautica*, 139:98–101, 2017.

[52] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[53] Xiqi Yang, Qingfeng Zhang, Xiyu Yang, Qingyu Peng, Zhan Li, and Na Wang. Edge detection in cassini astronomy image using extreme learning machine. In *MATEC Web of Conferences*, volume 189, page 06007. EDP Sciences, 2018.

[54] Yuni Zeng, Hua Mao, Dezhong Peng, and Zhang Yi. Spectrogram based multi-task audio classification. *Multimedia Tools and Applications*, 78(3):3705–3722, 2019.

[55] Bo Zhang, ShaoMing Hu, Junju Du, Xu Yang, Xu Chen, Hai Jiang, Hai Cao, and Shuai Feng. Detecting moving objects in photometric images using 3d hough transform. *Publications of the Astronomical Society of the Pacific*, 136(5):054502, 2024.

[56] Wenjie Zhang, Jiqing Han, and Shiwen Deng. Heart sound classification based on scaled spectrogram and tensor decomposition. *Expert Systems with Applications*, 84:220–231, 2017.

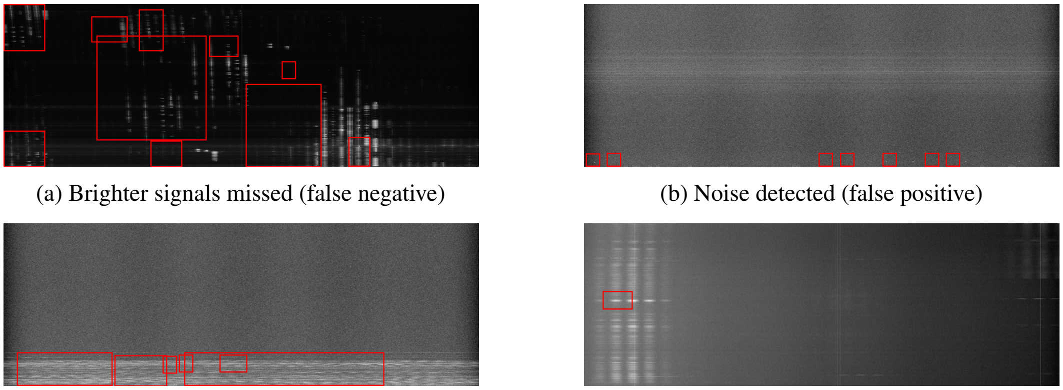
[57] Shilong Zhou, Haijin Zhou, and Lei Qian. A multi-scale small object detection algorithm sma-yolo for uav remote sensing images. *Scientific Reports*, 15(1):9255, 2025.

A Unsupervised Edge Detection Outputs

A.1 NumPy Masking

A straightforward and highly interpretable approach for unsupervised edge detection is to apply a simple NumPy-based thresholding mask, where we retain only pixels with intensities greater than the mean plus a multiple of the standard deviation. Specifically, we compute the mean μ and standard deviation σ of the coarse-channel spectrogram and keep pixels satisfying $I \geq \mu + k\sigma$, with $k \in \{5, 6, 7\}$. This method is attractive for its simplicity and speed.

Figure 9 shows example outputs using a strict threshold of $\mu + 7\sigma$. Panel (a) shows that brighter signals can still be missed entirely, resulting in false negatives, while panel (b) demonstrates that random background noise often exceeds the threshold, producing false positives. Panel (c) captures a horizontal track but fragments it into multiple disjoint boxes, and panel (d) shows only partial detection of a vertical signal. Overall, these examples highlight that simple thresholding struggles to robustly bound signals while minimizing both false negatives and false positives. These shortcomings motivate the exploration of more sophisticated edge detection strategies that can account for noise patterns, signal morphology, and continuity, ultimately providing more reliable candidate extractions. Experiments were conducted on a system with dual Intel Xeon Silver 4210 CPUs (40 logical cores), 188 GB RAM, and four NVIDIA RTX A4000 GPUs (16 GB each). These experiments completed in approximately 1 hours.



(a) Brighter signals missed (false negative)

(b) Noise detected (false positive)

(c) Horizontal signal fragmented

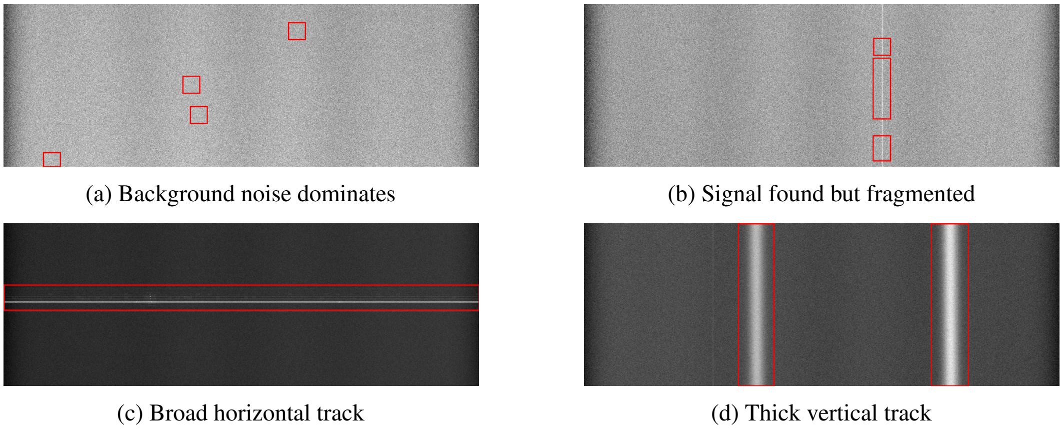
(d) Partial signal detected

Figure 9: Qualitative behavior of the Numpy mask ($\mu \pm 7\sigma$) on four representative mid-resolution frames.

A.2 Sobel-based Unified Edge-Density Map

Figure 10 illustrates the qualitative behavior of the Sobel-based Unified Edge–Density (UED) method [4] on a set of representative mid-resolution spectrogram frames. This classical gradient-based detector responds strongly to high-contrast transitions, allowing it to detect both narrow vertical tracks associated with drifting narrowband signals and broad horizontal tracks corresponding to broadband or impulsive interference. Panel (c) demonstrates a broad horizontal drift that is captured clearly and continuously, while panel (d) shows strong vertical tracks that are also successfully identified, albeit with edges slightly thicker than the true signal due to the gradient magnitude response.

However, Sobel filtering also exhibits several limitations. In low-signal or noise-dominated regions, small random intensity variations are often enhanced, producing isolated speckles that are misclassified as candidate signals, as seen in panel (a). In addition, weak or partially occluded vertical tracks may fragment into multiple disconnected bounding boxes rather than a single continuous detection, as shown in panel (b). These behaviors highlight Sobel's sensitivity to background noise and its tendency to over-segment faint signals, motivating the later fusion of multiple edge detection methods to improve robustness against false positives and signal fragmentation. Experiments were performed on a system with dual Intel Xeon Silver 4210 CPUs (40 logical cores), 188 GB of RAM, and four NVIDIA RTX A4000 GPUs (16 GB each), completing in roughly 1 hour.



(a) Background noise dominates

(b) Signal found but fragmented

(c) Broad horizontal track

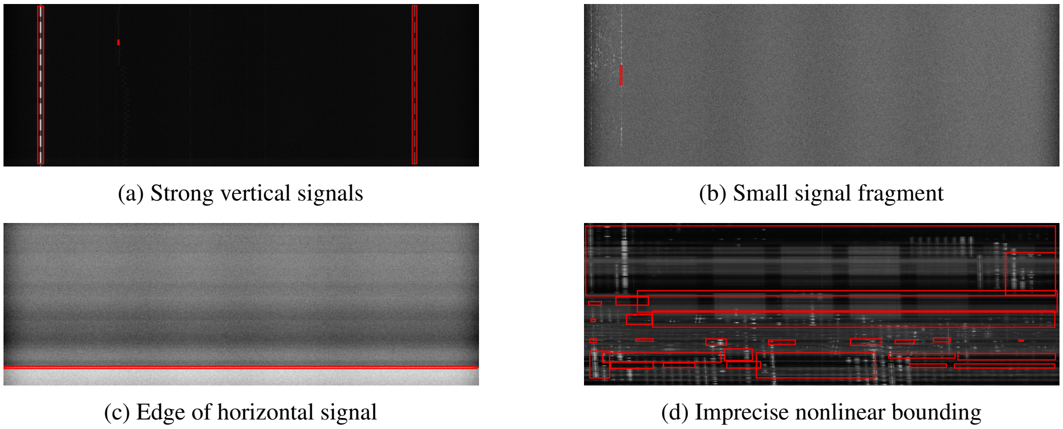
(d) Thick vertical track

Figure 10: Qualitative behavior of the Sobel-based Unified Edge–Density (UED) map on four representative mid-resolution frames.

A.3 Structured Forests

Structured Forest (SF) edge detection [13] provides a more context-aware behavior than simple gradient-based methods such as Sobel. By leveraging a supervised random-forest model trained on small image patches, SF predicts local edge probabilities that capture both intensity transitions and short-range structural cues.

Figure 11 illustrates its qualitative performance for signal detection. Panel (a) shows that SF effectively traces narrow, continuous vertical signal tracks and produces clean edges, However, panel (b) and panel (d) demonstrates its poorer performance in precisely bounding nonlinear or amorphous signals. SF strikes a more favorable balance between noise robustness and edge sharpness, reducing the false positives of Sobel filtering. Experiments were performed on a system with dual Intel Xeon Silver 4210 CPUs (40 logical cores), 188 GB of RAM, and four NVIDIA RTX A4000 GPUs (16 GB each), completing in roughly 1.5 hours.



(a) Strong vertical signals

(b) Small signal fragment

(c) Edge of horizontal signal

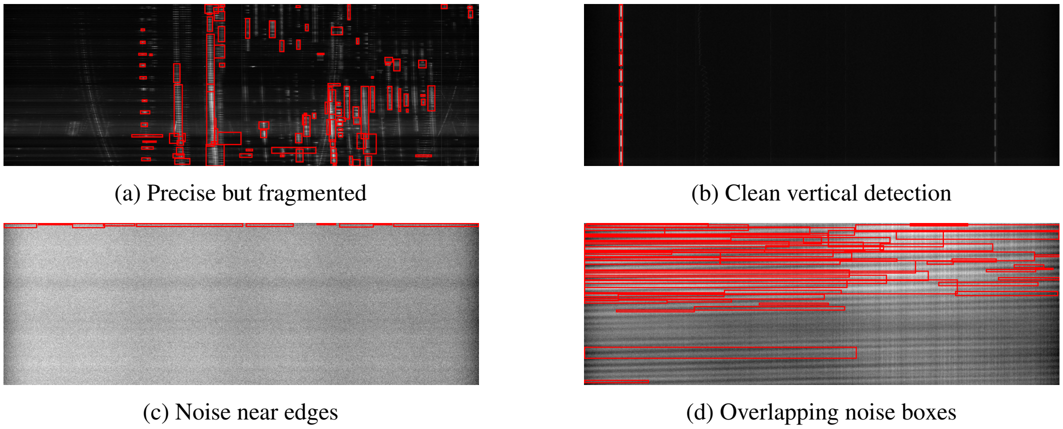
(d) Imprecise nonlinear bounding

Figure 11: Qualitative behavior of Structured Forest (SF) on four representative mid-resolution frames.

A.4 Canny Edge

The Canny edge detector [8][24] provides sharp, well-defined edges that are advantageous for isolating signal boundaries in mid-resolution spectrograms. By combining Gaussian smoothing, gradient-based detection, non-maximum suppression, and hysteresis thresholding, it produces thin and continuous contours that suppress most isolated noise. Figure 12 demonstrates its qualitative performance. Panel (b) shows a clean detection of a strong vertical signal with minimal edge thickening, and panel (a) illustrates Canny's ability to produce precise bounding boxes; however, these edges can become overly fragmented when signals are weak or discontinuous.

Panel (c) highlights its sensitivity to noise, particularly near coarse-channel edges. Panel (d) shows how noisy regions can lead to many overlapping bounding boxes, complicating downstream labeling. Overall, Canny provides accurate edge localization and thin contours, but it has tendencies to fragment faint signals and produce multiple small false positives. Experiments were performed on a system with dual Intel Xeon Silver 4210 CPUs (40 logical cores), 188 GB of RAM, and four NVIDIA RTX A4000 GPUs (16 GB each), completing in approximately 1 hour.


(a) Precise but fragmented

(b) Clean vertical detection

(c) Noise near edges

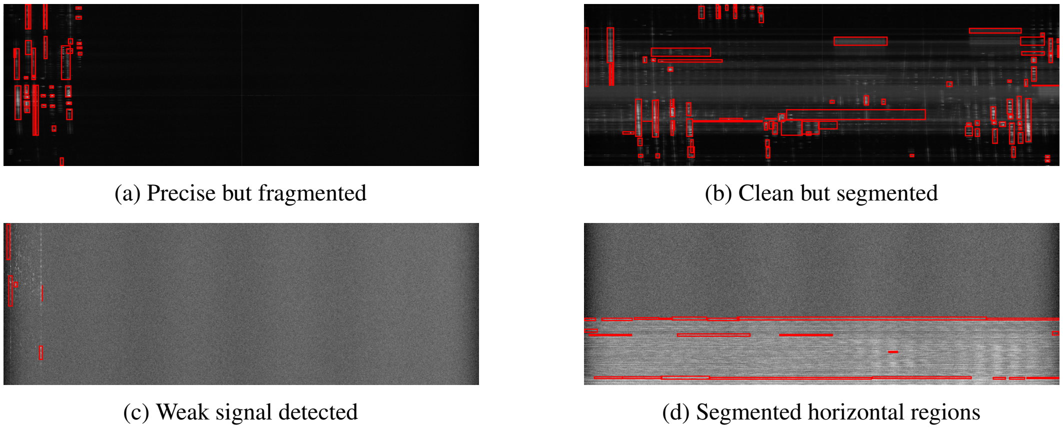
(d) Overlapping noise boxes

Figure 12: Qualitative behavior of Canny edge detection on four representative mid-resolution frames.

A.5 Hough Transform

The Hough transform [6] specializes in detecting straight and parametric signal tracks, making it particularly effective for constant-drift narrowband signals in mid-resolution spectrograms. By voting in a parameter space defined by line slope and intercept, it aggregates evidence across pixels to highlight coherent linear structures even when partially occluded. Figure 13 illustrates its qualitative performance. Panels (a) and (b) shows the clean detection of stronger signals, though some are slightly segmented due to drift variations or when signals are weak or discontinuous.

Panel (c) captures limited portions of a weak signal, indicating that Hough retains some sensitivity but struggles to form complete tracks under low SNR conditions. Panel (d) highlights its ability to capture horizontal signal regions, though detections are highly segmented, reflecting the method's bias toward straight-line continuity. Overall, the Hough transform excels at isolating linear and constant-drift signals but exhibits reduced recall for nonlinear or very weak signals. Experiments finished in 2 hours on a 40-core Intel Xeon Silver 4210 CPU with 188 GB RAM.


(a) Precise but fragmented

(b) Clean but segmented

(c) Weak signal detected

(d) Segmented horizontal regions

Figure 13: Qualitative behavior of Hough transform on four representative mid-resolution frames.

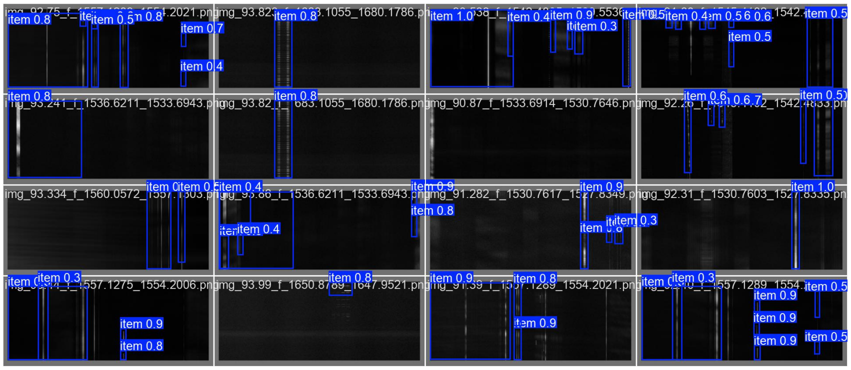


Figure 14: Validation batch from YOLOv8 with predicted signals labeled

B YOLOv8 Experimentation

In addition to the YOLOv11 experiments presented in the main text, we also fine-tune the *Nano* version of Ultralytics **YOLOv8** (`yolov8n.pt`, pretrained on COCO) [44]. Our dataset consists of the same 8,037 spectrogram crops with bounding boxes generated by the unsupervised Mixture-of-Experts model, which we randomly divide into an 80% training set and a 20% validation set. To ensure reproducibility, we fix the global random seed to 42 across Python, NumPy, and PyTorch, and provide the same seed to YOLO's internal random number generator.

Training is performed for 100 epochs using stochastic gradient descent with an initial learning rate of 10^{-3}, a 5-epoch linear warm-up, momentum of 0.937, and weight decay of 5×10^{-4}. Batches contain 32 rectangular images (`rect=True`), and if GPU memory is exceeded, an automatic back-off routine reduces the batch size by half. Data augmentation is deliberately minimal to preserve the temporal integrity of signals: mosaic augmentation is disabled, horizontal flips are off (`mosaic=False`, `fliplr=0`), and the network is trained in single-class mode (`single_cls=True`). The classification, distribution-focal, and IoU loss components each carry a weight of 0.3, and early stopping is applied after 20 consecutive epochs without improvement. This YOLOv8 model completed training in less than one hour, similar to YOLOv11 model, on a dual AMD EPYC 7313 workstation equipped with 32 CPU cores, 504 GB of RAM, and four NVIDIA RTX A4000 GPUs (16 GB each).

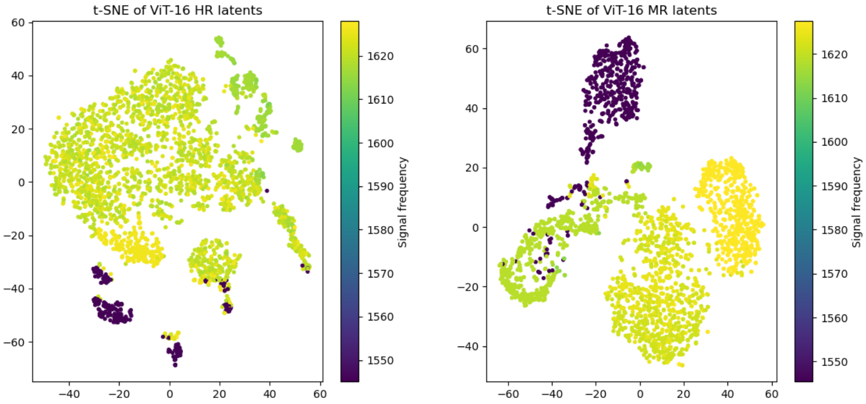
Figure 14 displays a sample validation batch with predicted signal labels. We see that YOLOv8 has comparable performance to YOLOv11 on the validation set. However, YOLOv11 is better suited for spectrogram analysis due to its incorporation of a refined backbone and neck with improved feature fusion, enabling better detection of small and thin objects such as drifting signal tracks. The YOLOv11 *Nano* variant also uses fewer parameters, making it more computationally efficient for scaling to larger datasets and incorporation into a larger pipeline.

C Subset Analysis: Iridium (1610-1629 MHz) and 1545 MHz Signals

We investigate how signals map across resolutions to better understand how mid-resolution (MR) data products can aid in interpreting detections in high-resolution (HR) data, particularly for reducing the high hit rate in technosignature searches. In this subset analysis, we focus on distinguishing signals from the Iridium satellite constellation (typically observed between 1610–1626.5 MHz) from those found near 1545 MHz, a region of the L-band dominated by terrestrial RFI. We sample signals from both the MR and HR datasets whose start frequencies fall within either [1610, 1630) MHz or [1545, 1546) MHz. Because of differing detection procedures—and the fact that multiple HR detections often correspond to a single MR event—the number of sampled signals is not directly balanced between resolutions for either the Iridium or 1545 MHz bands.

C.1 Extracting Latent Spaces

We begin our analysis by first extracting the latent space of each signal. As used in our original pipeline, we adopt the **Vision Transformer Base** with 16×16 patches (**ViT-B/16**) [14]. We use the pretrained `ViT_B_16_Weights.IMAGENET1K_V1` checkpoint distributed by TorchVision, once again keeping the network frozen with no additional fine-tuning. We then apply principal component analysis (PCA) to the resulting $N \times D$ ViT feature matrix—where N is the number of spectrogram patches and $D = 768$ is the original embedding size—to obtain a lower-dimensional representation. Using scikit-learn's `PCA` with `random_state=42`, we project into a deterministic 32-dimensional latent space, yielding $\mathbf{Z} \in \mathbb{R}^{N \times 32}$ for all subsequent visualization and clustering tasks.



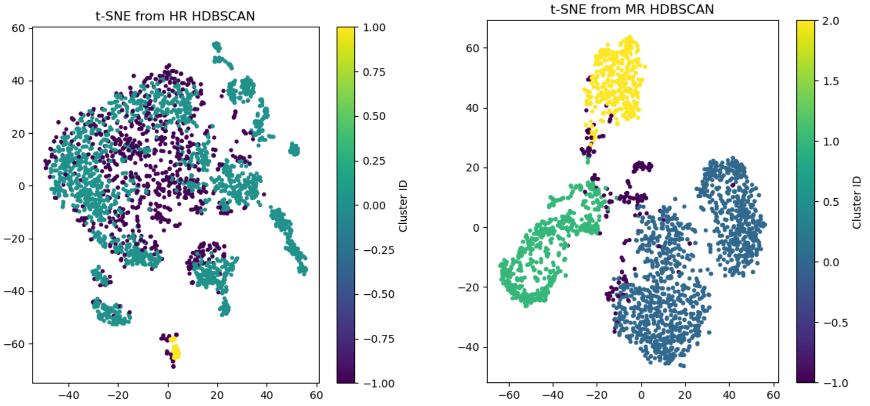
(a) Signals from HR data

(b) Signals from MR data

Figure 15: Latent spaces extracted from ViT-B16 for 1545 MHz and Iridium (1610-1629 MHz) signals

We again visualized the resulting vectors with Barnes–Hut t-SNE. Before projection, the 32-dimensional ViT embeddings were row-wise ℓ_2-normalized, and t-SNE (scikit-learn $1x$) was applied with $n_{\text{components}} = 2$, a perplexity of 30, early_exaggeration set to its default value of 12, Euclidean distance as the metric, and a fixed random seed of 42. The resulting embeddings are shown in Figure 15. Experiments were completed in approximately five minutes using four NVIDIA RTX A4000 GPUs (16 GB each).

Although frequency was not incorporated into the latent spaces, we see that signals separate relatively clearly based on color, which represents the start frequency of the signal. Plot (a) of the HR latent spaces shows that although clusters are not very tightly separated by color, we can still differentiate signals at 1545 MHz (purple points) from Iridium signals (green and yellow points). Plot (b) has clearer differentiation of signals based on frequency. Signals at 1545 MHz exist in a cluster clearly separated from Iridium signals. We also notice that signals from 1610-1629 MHz are separated into more than one cluster. This implies that in MR, some Iridium signals from 1610-1620 MHz may look different from Iridium signals greater than 1620 MHz.



(a) Clusters for HR signals

(b) Clusters for MR signals

Figure 16: HDBSCAN output for 1545 MHz and Iridium (1600-1629 MHz) signals

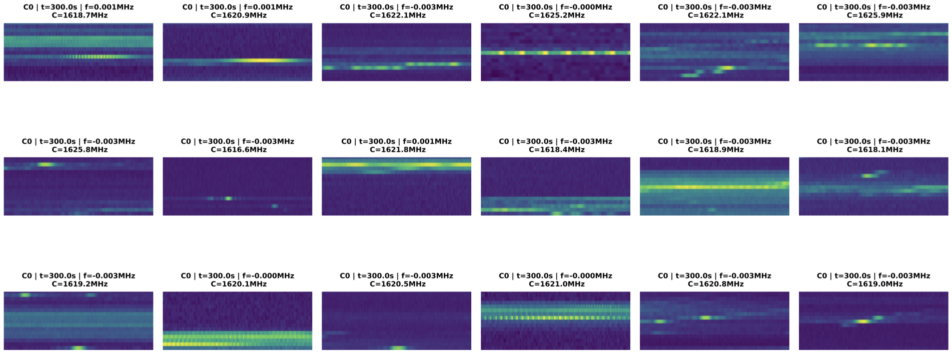


Figure 17: High-resolution cluster 0, primarily aligned with frequencies greater than 1610 MHz

C.2 Clustering with HDBSCAN

To cluster the 32-dimensional latent embeddings, we employ **HDBSCAN** (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [7]. HDBSCAN extends the DBSCAN framework [15] by building a hierarchy of density-based clusters and then condensing it to select the most stable structures, automatically determining the number of clusters without requiring a fixed k. Unlike k-means or spectral methods, HDBSCAN can identify clusters of varying densities and shapes while labeling low-density points as noise rather than forcing an assignment. This property is particularly useful for our spectrogram embeddings, where meaningful signal clusters may coexist with sparse or spurious points caused by RFI or weak detections. By using HDBSCAN, we obtain a set of robust clusters and an explicit outlier class that can be filtered or analyzed separately in downstream steps.

We cluster the normalized MR embeddings $\mathbf{Z}_{\text{mid}}$ using **HDBSCAN**, setting $\texttt{min\_cluster\_size} = 100$, $\texttt{min\_samples} = 50$, and use the cosine distance metric with the excess-of-mass (eom) cluster selection method. We also cluster the normalized HR embeddings $\mathbf{Z}_{\text{high}}$ with HDBSCAN with parameters $\texttt{min\_cluster\_size} = 20$, $\texttt{min\_samples} = 15$, and again use the cosine distance metric with the excess-of-mass (eom) cluster selection method. We performed a simple hyperparameter search for both clusterers, testing values $\texttt{min\_cluster\_size} \in \{15, 20, 50, 75, 100\}$ and $\texttt{min\_samples} \in \{15, 20, 50, 75, 100\}$. We found that other hyperparameter combinations led to less homogeneous clusters and misalignment with the frequency values shown in Figure 15. All experiments were completed in under ten minutes using four NVIDIA RTX A4000 GPUs (16 GB each).

Figure 16 show t-SNE plots of the resulting clusters. Comparing these plots to Figure 15, we notice that the clusters are relatively well aligned with frequencies as expected. We further evaluate the homogeneity of signals in each cluster. Figure 17 shows a sample of HR signals in cluster 0, and Figure 18 shows a sample of MR signals in cluster 0. We notice that each cluster appears internally homogeneous, and they both align with frequencies greater than 1610 MHz (most likely Iridium signals). After thorough evaluation, the clusters appear sufficiently homogeneous, so we proceed with aligning clusters across resolutions.

C.3 Matching HR and MR Clusters

To align HR clusters with their MR counterparts, we compute the ℓ_2-normalized centroid of each cluster in the latent space and solve a one-to-one assignment that maximizes the sum of cross-resolution cosine similarities. A formal proof of the optimality of this mapping is provided in Section D. In practice, this approach can also be extended to a many-to-one setting, since multiple HR clusters may correspond to the same MR cluster.

We begin by jointly reducing the mid- and fine-resolution latent vectors into a common low-dimensional space via PCA. Let

$$\mathbf{L}_{\text{mid}} \in \mathbb{R}^{N_{\text{mid}} \times 768}, \quad \mathbf{L}_{\text{fine}} \in \mathbb{R}^{N_{\text{fine}} \times 768}$$

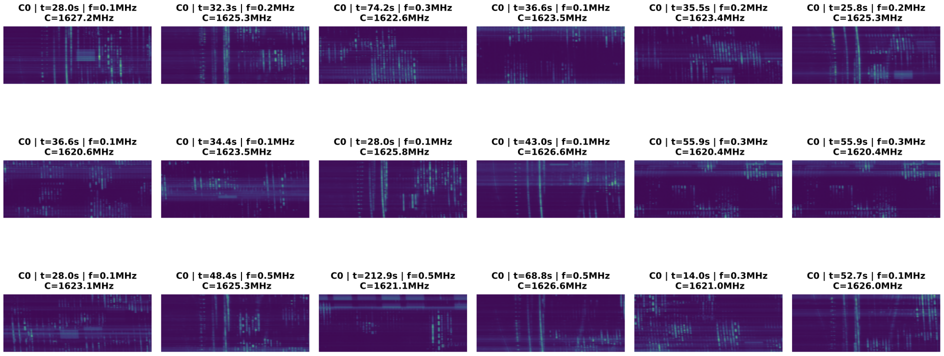


Figure 18: Mid-resolution cluster 0, primarily aligned with frequencies greater than 1610 MHz

be the original ViT embeddings. We stack them to form $\mathbf{L} = \begin{bmatrix}\mathbf{L}_{\text{mid}} \\ \mathbf{L}_{\text{fine}}\end{bmatrix} \in \mathbb{R}^{(N_{\text{mid}}+N_{\text{fine}})\times 768}$, and apply PCA with

$$k = \min\bigl(\text{target\_pcs},\ 768,\ N_{\text{mid}} + N_{\text{fine}}\bigr)$$

principal components (fixed random seed). Denote the projected embeddings by $\mathbf{M} = \mathrm{PCA}(\mathbf{L}_{\text{mid}}) \in \mathbb{R}^{N_{\text{mid}}\times k}$ and $\mathbf{F} = \mathrm{PCA}(\mathbf{L}_{\text{fine}}) \in \mathbb{R}^{N_{\text{fine}}\times k}$.

Next, we compute one centroid per cluster in each space. HDBSCAN assigns a label of -1 to points considered noise, meaning they do not belong to any dense cluster and are treated as outliers. Accordingly, we compute centroids only for clusters with labels $c \neq -1$; for such a cluster c, let

$$\boldsymbol{\mu}_c^{(\text{mid})} = \frac{1}{|\mathcal{I}_c|} \sum_{i\in\mathcal{I}_c} \mathbf{M}_{i\,:}, \quad \boldsymbol{\mu}_c^{(\text{fine})} = \frac{1}{|\mathcal{J}_c|} \sum_{j\in\mathcal{J}_c} \mathbf{F}_{j\,:},$$

where $\mathcal{I}_c$ and $\mathcal{J}_c$ index points with label c. We then ℓ_2-normalize each centroid to unit length, producing $\mathbf{C}_{\text{mid}} \in \mathbb{R}^{M\times k}$ and $\mathbf{C}_{\text{fine}} \in \mathbb{R}^{M'\times k}$.

To seed an orthogonal alignment, we form the initial similarity matrix

$$S_{ij}^{(0)} = \cos\bigl(\boldsymbol{\mu}_i^{(\text{mid})},\ \boldsymbol{\mu}_j^{(\text{fine})}\bigr)$$

and greedily pair each mid centroid i with the fine centroid $\arg\max_j S_{ij}^{(0)}$. Let $\{(i_\ell, j_\ell)\}_{\ell=1}^{K}$ be these seed pairs. We solve the Procrustes problem

$$R^* = \arg\min_{R\in O(k)} \bigl\| \mathbf{C}_{\text{fine}}^{(\text{seed})} R - \mathbf{C}_{\text{mid}}^{(\text{seed})} \bigr\|_F$$

via the closed-form SVD solution, and rotate all fine centroids as $\widetilde{\mathbf{C}}_{\text{fine}} = \mathbf{C}_{\text{fine}} R^*$.

We then recompute the post-alignment cosine similarity matrix

$$S_{ij} = \cos\bigl(\boldsymbol{\mu}_i^{(\text{mid})},\ \widetilde{\boldsymbol{\mu}}_j^{(\text{fine})}\bigr),$$

and apply the Hungarian algorithm to the cost matrix $-S$ to find the one-to-one assignment Π^* that maximizes total cosine similarity. This yields an optimal mapping $\text{mid} \to \text{fine}$. Finally, we optionally prune any matched pair whose similarity S_{ij} falls below a threshold (e.g. 0.30), and report the surviving correspondences along with any unmatched clusters. Experiments were completed in under five minutes using four NVIDIA RTX A4000 GPUs (16 GB each).

We visualize this final mapping using a heatmap, as shown in Figure 19. Based on the output from the Hungarian algorithm, MR cluster 1 (aligned with frequency range centered around 1610 MHz, as shown in Figure 15) is left unmatched. MR cluster 2 with center frequency 1545.4 MHz has been matched with HR cluster 1 with center frequency 1567.5 MHz. We can conclude that each of these clusters align with signals at 1545 MHz, and the cosine similarity of 0.94 indicates their correct mapping. Similarly, MR cluster 0 with center frequency 1624.7 MHz has been matched with HR cluster 0 with center frequency 1614.3 MHz with a cosine similarity score of 0.44. Each of these

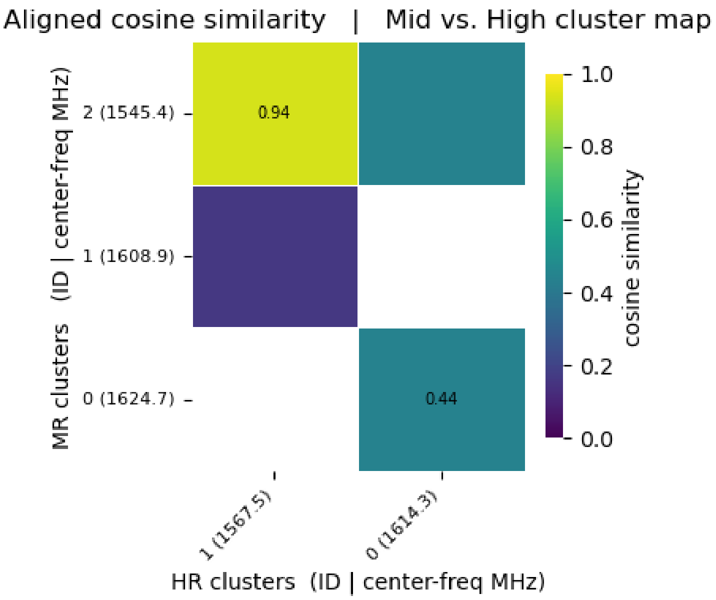


Figure 19: Heatmap showing Hungarian algorithm matching across resolutions

clusters most likely represents Iridium signals, indicating they have also been correctly matched. Since MR cluster 1 with center frequency 1608.9 MHz is left mismatched, we presume that Iridium signals in the lower range appear different or have different structure at mid-resolution compared to high-resolution. Since clusters of signals at 1545 MHz and 1610-1629 MHz have been matched in the way we expect, we attempt to extend this analysis to all signals from 1500-1650 MHz.

D From Dual Latent Spaces to a Globally–Optimal Cluster Mapping

We denote

$$\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \mathbb{R}^D, \tag{1}$$

$$\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\} \subset \mathbb{R}^D \tag{2}$$

as the "mid–resolution" and "fine–resolution" latent clouds obtained after a shared k–dimensional PCA projection [32]. Following HDBSCAN, we remove label -1 (noise) and retain $\mathcal{X}_1, \dots, \mathcal{X}_{k_X}$ and $\mathcal{Y}_1, \dots, \mathcal{Y}_{k_Y}$, the genuine clusters.

D.1 Notation and basic preprocessing

1. **PCA reduction**
   We stack both clouds, run PCA, keep $k = \min(8, D, m+n)$ principal axes, and write $\mathbf{z}_i := \mathrm{PCA}_k(\mathbf{x}_i)$, $\mathbf{w}_j := \mathrm{PCA}_k(\mathbf{y}_j)$.
2. **Unit–length centroids.**
   For each non-noise cluster we form

$$\boldsymbol{\mu}_i := \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{z} \in \mathcal{X}_i} \mathbf{z}, \tag{3}$$

$$\boldsymbol{\nu}_j := \frac{1}{|\mathcal{Y}_j|} \sum_{\mathbf{w} \in \mathcal{Y}_j} \mathbf{w}, \tag{4}$$

$$\widetilde{\boldsymbol{\mu}}_i := \frac{\boldsymbol{\mu}_i}{\|\boldsymbol{\mu}_i\|_2}, \quad \widetilde{\boldsymbol{\nu}}_j := \frac{\boldsymbol{\nu}_j}{\|\boldsymbol{\nu}_j\|_2}. \tag{5}$$

   This keeps the cosine similarity equal to the Euclidean inner product.

3. **Stacking.**
Define

$$A := \left[\widetilde{\boldsymbol{\mu}}_1^\top, \ldots, \widetilde{\boldsymbol{\mu}}_{k_X}^\top\right]^\top \in \mathbb{R}^{k_X \times k}, \tag{6}$$

$$B := \left[\widetilde{\boldsymbol{\nu}}_1^\top, \ldots, \widetilde{\boldsymbol{\nu}}_{k_Y}^\top\right]^\top \in \mathbb{R}^{k_Y \times k}. \tag{7}$$

Rows of A and B are unit vectors.

D.2 Step 1: Seed correspondences

We obtain an initial many–to–one map $g : \{1, \ldots, k_X\} \to \{1, \ldots, k_Y\}$ by greedy row–wise maximization of

$$S^{(0)} := A\,B^\top, \tag{8}$$

$$C := \{(i, g(i)) \mid i = 1, \ldots, k_X\}. \tag{9}$$

Let $A_C \in \mathbb{R}^{|C| \times k}$ and $B_C \in \mathbb{R}^{|C| \times k}$ be the sub-matrices whose rows correspond to C.

D.3 Step 2: Orthogonal Procrustes alignment

We wish to find the orthogonal matrix $R \in O(k)$ that best aligns the “fine” centroid matrix $B_C \in \mathbb{R}^{n \times k}$ to the “mid” centroid matrix $A_C \in \mathbb{R}^{n \times k}$, in the least–squares sense:

$$R^\star = \underset{R \in O(k)}{\operatorname{argmin}} \left\| B_C\,R - A_C \right\|_F^2. \tag{10}$$

1. Expand the Frobenius norm as a trace Using $\|X - Y\|_F^2 = \operatorname{tr}((X - Y)^\top (X - Y))$ with $X = B_C R$, $Y = A_C$, we get

$$\begin{aligned}\|B_C R - A_C\|_F^2 &= \operatorname{tr}\big((B_C R)^\top (B_C R)\big) + \operatorname{tr}(A_C^\top A_C) \\ &\quad - 2\operatorname{tr}\big(R^\top B_C^\top A_C\big).\end{aligned} \tag{11}$$

Since $RR^\top = I$, the first trace reduces to $\operatorname{tr}(B_C^\top B_C)$, which is constant. Hence

$$R^\star = \max_{R^\top R = I} \operatorname{tr}\big(R^\top B_C^\top A_C\big). \tag{12}$$

Define the cross–covariance

$$M := B_C^\top A_C \in \mathbb{R}^{k \times k}. \tag{13}$$

2. Singular–value decomposition Compute

$$M = U\,\Sigma\,V^\top, \tag{14}$$

where $U, V \in O(k)$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_k)$, and $\sigma_i \geq 0$.

3. Change of variables Set

$$\begin{aligned}\operatorname{tr}(R^\top M) &= \operatorname{tr}(R^\top U\,\Sigma\,V^\top) = \operatorname{tr}(V^\top R^\top U\,\Sigma) \\ &= \operatorname{tr}(Q\,\Sigma), \quad Q := V^\top R^\top U \in O(k),\end{aligned} \tag{15}$$

$$\operatorname{tr}(Q\,\Sigma) = \sum_{i=1}^{k} Q_{ii}\,\sigma_i. \tag{16}$$

Since $\sigma_i \geq 0$, the maximum over Q occurs at $Q = I$, implying

$$V^\top (R^\star)^\top U = I \quad \Longrightarrow \quad R^\star = U\,V^\top. \tag{17}$$

4. Conclusion Therefore the unique orthogonal solution is

$$R^\star = U\,V^\top, \quad \text{where } B_C^\top A_C = U\,\Sigma\,V^\top. \tag{18}$$

Applying $R^\star$ to the full fine-centroid matrix B yields $\widetilde{B} = B\,R^\star$, which preserves lengths and therefore $\langle \widetilde{\nu}_i, \widetilde{\nu}_j \rangle$ remains the aligned cosine similarity.

D.4 Step 3: Global assignment as a linear program

Define the similarity matrix $S := A\,\widetilde{B}^{\top} \in \mathbb{R}^{k_X \times k_Y}$. Our goal is a one-to-one map that *maximizes total similarity*.

$$\max_{\Pi \in \mathcal{P}} \operatorname{tr}\bigl(S^{\top}\Pi\bigr) \tag{19}$$

where

$$\mathcal{P} := \bigl\{\Pi \in \{0,1\}^{k_X \times k_Y} : \Pi\mathbf{1} \le \mathbf{1},\ \Pi^{\top}\mathbf{1} \le \mathbf{1}\bigr\}. \tag{20}$$

This is the linear assignment problem.

Theorem 1 (Optimal cluster matching)**.** *Let* $(\hat{\mathbf{r}}, \hat{\mathbf{c}}) =$ *Hungarian*$(-S)$ *be the index pairs returned by the Hungarian algorithm applied to the* cost *matrix* $-S$*. The permutation matrix* $\Pi^{\star}$ *subject to* $\Pi^{\star}_{\hat{r}_i, \hat{c}_i} = 1$ *(and zeros elsewhere) maximizes* $\operatorname{tr}(S^{\top}\Pi)$ *over* $\mathcal{P}$*.*

Proof. Equivalently, define the cost matrix $C = -S$ and rewrite as $\min_{\Pi \in \mathcal{P}} \operatorname{tr}(C^{\top}\Pi)$. Its primal and dual linear programs are:

$$\begin{aligned}
\text{(Primal)} \quad & \min_{\Pi \ge 0} \sum_{i=1}^{k_X}\sum_{j=1}^{k_Y} C_{ij}\,\Pi_{ij} && (21)\\
\text{s.t.} \quad & \sum_{j=1}^{k_Y} \Pi_{ij} = 1, \quad i = 1, \dots, k_X, && (22)\\
& \sum_{i=1}^{k_X} \Pi_{ij} = 1, \quad j = 1, \dots, k_Y.
\end{aligned}$$

$$\begin{aligned}
\text{(Dual)} \quad \max_{\mathbf{u},\mathbf{v}} \quad & \sum_{i=1}^{k_X} u_i \ + \ \sum_{j=1}^{k_Y} v_j && (23)\\
\text{s.t.} \quad & u_i + v_j \le C_{ij}, \quad \forall\, i = 1, \dots, k_X,\ j = 1, \dots, k_Y.
\end{aligned}$$

The Hungarian method maintains at each iteration:

$$u_i + v_j \ \le \ C_{ij}, \quad \bar{C}_{ij} := C_{ij} - u_i - v_j \ge 0,$$

and grows a matching using only the zero–reduced–cost edges $\bar{C}_{ij} = 0$. When no augmenting path exists, the potentials $(\mathbf{u}, \mathbf{v})$ are updated by the minimal slack $\min_{(i,j):\bar{C}_{ij}>0} \bar{C}_{ij}$, preserving dual feasibility and creating at least one new tight edge. Upon termination with a perfect matching, complementary slackness $\Pi_{ij} > 0 \implies u_i + v_j = C_{ij}$ holds, and by strong duality the returned $\Pi^{\star}$ minimises the cost in (21), equivalently maximises similarity. □

D.5 Step 4: Resulting mapping

$$\boxed{i \longleftrightarrow j \quad \text{iff} \quad \Pi^{\star}_{ij} = 1}$$

is a bijection between a subset of mid clusters and a subset of fine clusters. Any cluster left unmatched by $\Pi^{\star}$ has no partner with non–negative similarity higher than the Hungarian optimum.

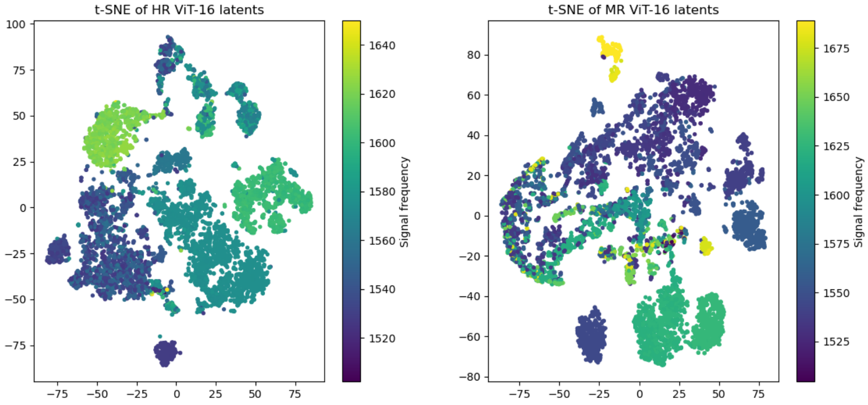
Corollary 1. *The above four-step pipeline (PCA* → *unit centroids* → *Procrustes alignment* → *Hungarian assignment) yields the* unique *mapping that maximises the sum of pairwise aligned cosine similarities.*

D.6 Complexity summary

PCA projection	:	$O\big((m+n)Dk\big)$
Centroid computation	:	$O\big((m+n)k\big)$
Procrustes (SVD)	:	$O(k^3)$
Similarity matrix	:	$O(k_X k_Y k)$
Hungarian assignment	:	$O\big(\max\{k_X, k_Y\}^3\big)$

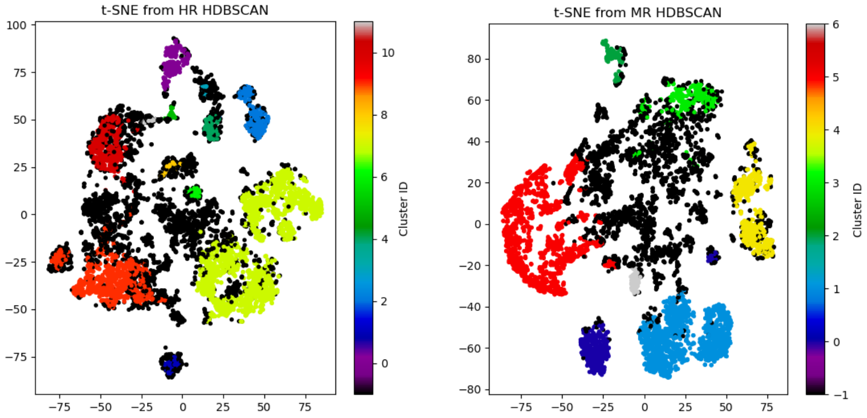
For typical SETI clustering workloads $k_X, k_Y < 10^3$, the runtime is negligible compared with the upstream neural-network inference, while providing a provably optimal cross-resolution correspondence.

E Matching Signals Across Resolutions



(a) Signals from HR data (b) Signals from MR data

Figure 20: Latent spaces extracted from ViT-B16 for signals from 1500-1650 MHz

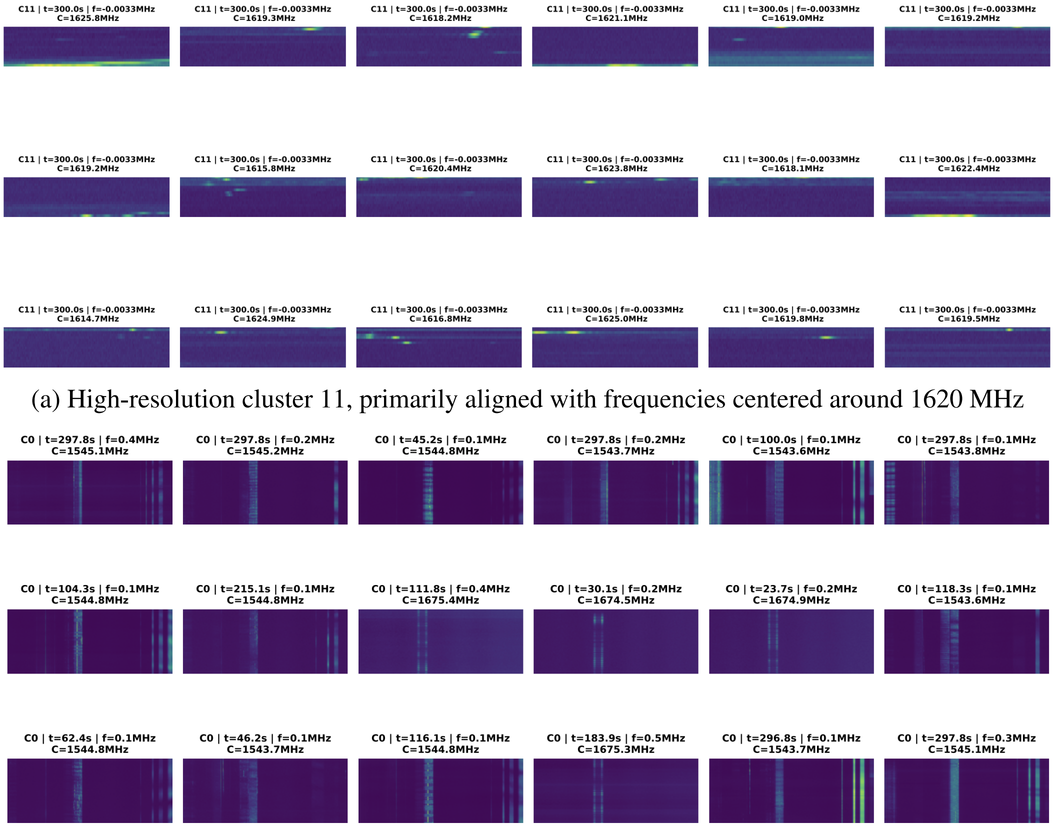


(a) Clusters for HR data (b) Clusters for MR data

Figure 21: HDBSCAN output for signals from 1500-1650 MHz

E.1 Extracting Latent Spaces

Expanding on our analysis in section C, we investigate all signals found from 1500-1650 MHz in L-Band. We use our original dataset of MR and HR signals in this range, once again extracting the latent space of each image with **ViT-B/16**) [14]. We use scikit-learn's `PCA` with `random_state=42` to project into a 32-dimensional latent space, yielding $\mathbf{Z} \in \mathbb{R}^{N \times 32}$ for all subsequent visualization and clustering tasks.



(a) High-resolution cluster 11, primarily aligned with frequencies centered around 1620 MHz

(b) Mid-resolution cluster 0, primarily aligned with frequencies centered around 1545 MHz

Figure 22: Evaluation of HR and MR cluster quality

We visualized the resulting vectors with Barnes–Hut t-SNE after row-wise ℓ_2-normalization. t-SNE (scikit-learn $1x$) was applied with $n_{\text{components}} = 2$, a perplexity of 30, early_exaggeration set to its default value of 12, Euclidean distance as the metric, and a fixed random seed of 42. The resulting embeddings are shown in Figure 20.

Similar to section C.1, we see that signals separate relatively clearly based on start frequency of the signal (represented by the color). In both plots (a) and (b), although not all clusters are very tightly separated by color, we see relatively clear differentiation. For example, we once again the green points with frequencies around 1620 MHz tend to cluster together, representing Iridium. Since these latent representations match our expectations of how signals at similar frequencies also appear visually similar, we proceed with using these vectors for clustering. All experiments were completed in under five minutes using four NVIDIA RTX A4000 GPUs (16 GB each).

E.2 Clustering with HDBSCAN

We once again cluster [52][48] the normalized mid-resolution embeddings $\mathbf{Z}_{\text{mid}}$ using **HDBSCAN**, with `min_cluster_size` $= 100$, `min_samples` $= 50$, the cosine distance metric, and the excess-of-mass (`eom`) cluster selection method. Similarly, the normalized high-resolution embeddings $\mathbf{Z}_{\text{high}}$ are clustered with HDBSCAN using `min_cluster_size` $= 20$, `min_samples` $= 20$, and the same cosine-distance `eom` configuration. For both clusterers, we performed a simple hyperparameter sweep over `min_cluster_size`, `min_samples` $\in \{15, 20, 50, 75, 100\}$ and found that other combinations produced less homogeneous clusters and poorer alignment with the frequency structure in Figure 20.

Figure 21 shows the resulting t-SNE visualizations of the clusters. Compared to Figure 20, the clusters are well aligned with frequency as expected, and we further assess the homogeneity of signals within each cluster. Figure 22 plot (a) represents a sample of signals from cluster 11, primarily centered around 1620 MHz. Plot (b) represents a sample of signals from cluster 0, primarily centered around 1545 MHz. Although some outliers exist in every cluster, analysis of each cluster reveals strong visual homogeneity. Some signals in each cluster may be from outlier frequency ranges, but their morphologies appear similar to the rest of the cluster. Since the clusters are strongly homogeneous, we proceed with matching clusters across different resolutions. All experiments were completed in under five minutes using four NVIDIA RTX A4000 GPUs (16 GB each).

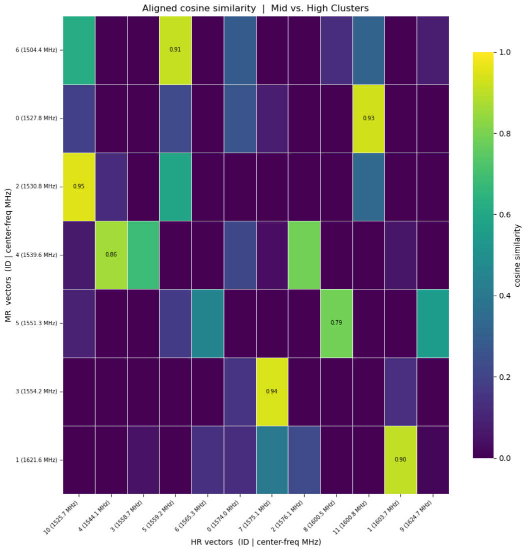


Figure 23: Heatmap visualizing Hungarian algorithm applied to clusters of all 1500-1650 MHz signals

E.3 Matching HR and MR clusters

We repeat the method detailed in section C.3 extended to a subset of 8,037 signals ranging from 1500-1650 MHz. After performing PCA on the stacked original embeddings to reduce our dimensionality to 8, we compute one centroid per cluster and perform Procrustes alignment. The final matching is found using the Hungarian algorithm and visualized in Figure 23. HR clusters 0, 2, 3, 6, and 9 are left unmatched.

However, we notice that the cosine similarities do not align with the frequency ranges of signals as expected. Each axis is sorted by the mean frequency calculated for each centroid, meaning that we expect strong matches along the diagonal. Our results show that clusters with little to no overlap in frequencies are matched with high cosine similarity values. For example, HR cluster 11 with center frequency 1600.8 MHz (visualized in Figure 22a) is matched to MR cluster 0 with center frequency 1527.8 MHz (visualized in Figure 22b) with cosine similarity of 0.93. We notice that the center frequencies and ranges shown in their corresponding figures have little to no overlap. This pattern is apparent in many other pairs, such as MR cluster 5 (center frequency 1551.3 MHz) and HR cluster 8 (center frequency 1600.5 MHz) with cosine similarity of 0.79, as well as MR cluster 6 (center frequency 1504.4 MHz) and HR cluster 5 (center frequency 1559.2 MHz) with cosine similarity of 0.91. Since our previous analysis showed that each cluster was relatively homogeneous, these results imply that cosine similarity is not an indicative metric of similarity for clusters across resolutions.

F Multi-Resolution Latent Space Analysis

For further analysis, we first concatenate the HR and MR feature matrices into one combined dataset, and fit a two-component PCA model on this merged set to ensure both resolutions share the same projection axes. We then project each resolution's features separately into the resulting two-dimensional space. Figure 24 shows a side-by-side scatterplot, where the top panel displays the high-resolution points in the 2D PCA embedding, colored by their start frequencies; the bottom panel shows the mid-resolution points using the same axes and color mapping.

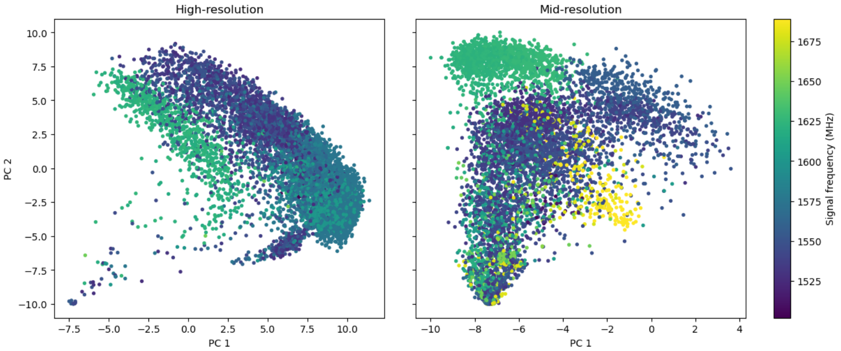


Figure 24: Resulting projections of HR and MR signals after combined PCA

By comparing each of these plots, we clearly see that the points and clusters are not aligned by frequency across resolutions. In other words, a signal at HR found at 1625 MHz (represented by a green point) does not map to the same space as a signal at MR found at the same frequency range. Our results indicate that the latent spaces derived from high- and mid-resolution features do not preserve a simple correspondence based on physical frequency. Cosine similarity, which reflects angular proximity in the embedding space, is therefore not always a reliable proxy for frequency identity. Signals observed at the same MHz can appear dissimilar if resolution-specific artifacts or noise push their embeddings apart, while signals at different frequencies may appear close if their morphological features dominate the representation. In this context, cosine similarity captures geometric relationships between embeddings but does not guarantee alignment with the true spectral properties of the signals.

Although high- and mid-resolution latent spaces do not always align by frequency, this analysis reinforces the utility of our multi-resolution pipeline in our main paper for reducing the high-resolution candidate haystack. By leveraging unsupervised edge detection and a Mixture of Experts framework to identify broadband RFI in MR data, and embedding HR hits in a ViT-B16 latent space, we can remove HR signals that likely correspond to RFI before downstream analysis. Integrating these approaches more tightly, such as by refining cross-resolution matching, promises to further shrink the candidate set and accelerate the identification of promising technosignatures in large surveys. In particular, improving cross-resolution matching could involve incorporating frequency as a conditioning variable to guide associations, or applying cluster-aware matching that links multiple HR detections to their corresponding MR broadband events. Such refinements would reduce duplicate or spurious candidates while preserving rare or isolated signals of interest.