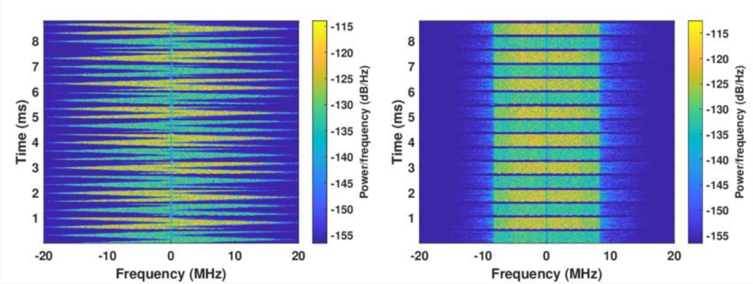
# *AI Enabled RF Strategies*

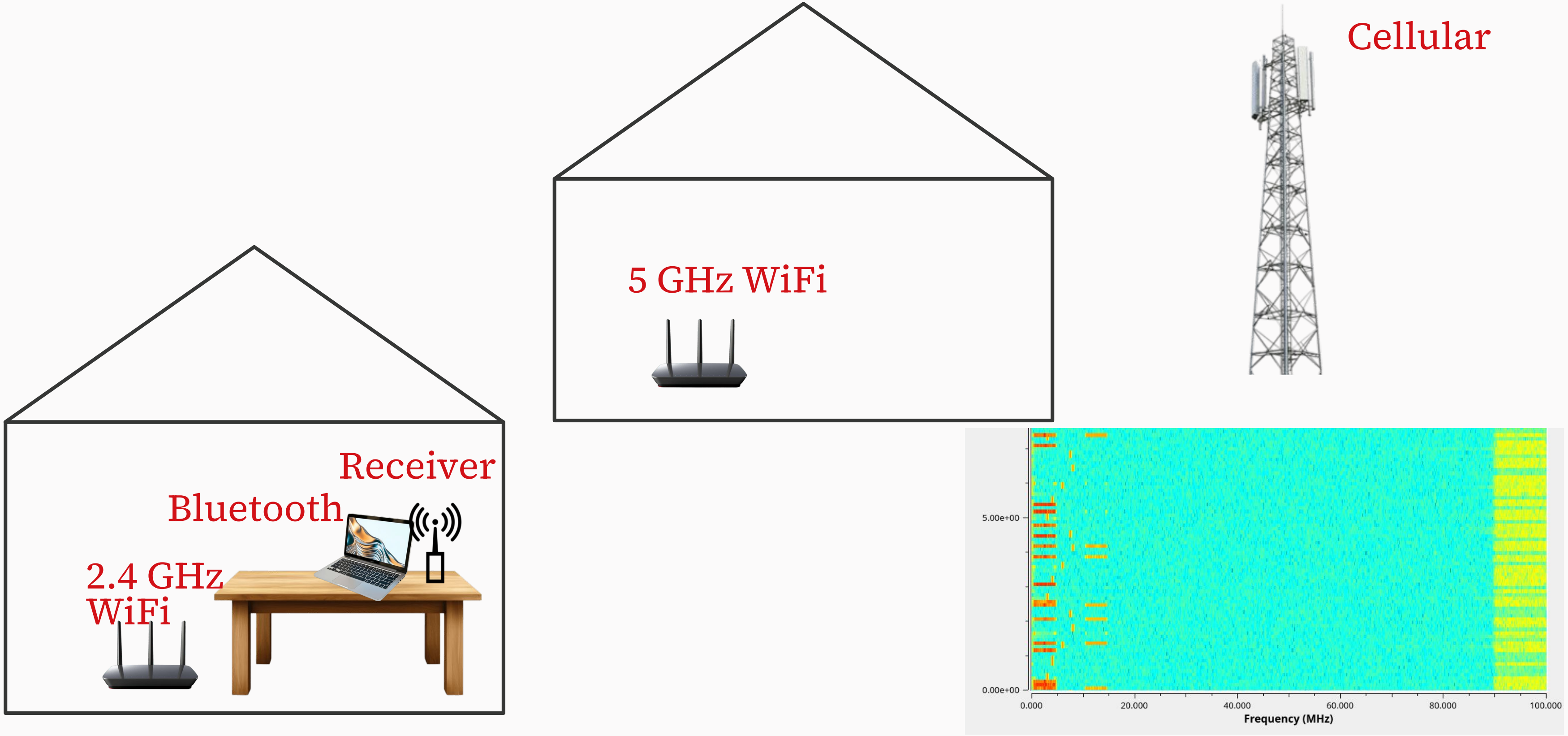



A new channel-blind approach to analyzing packet-based communications

Atharv Thakur, Anant Chary

Advisor: Predrag Spasojevic

# *Simulation Setup*

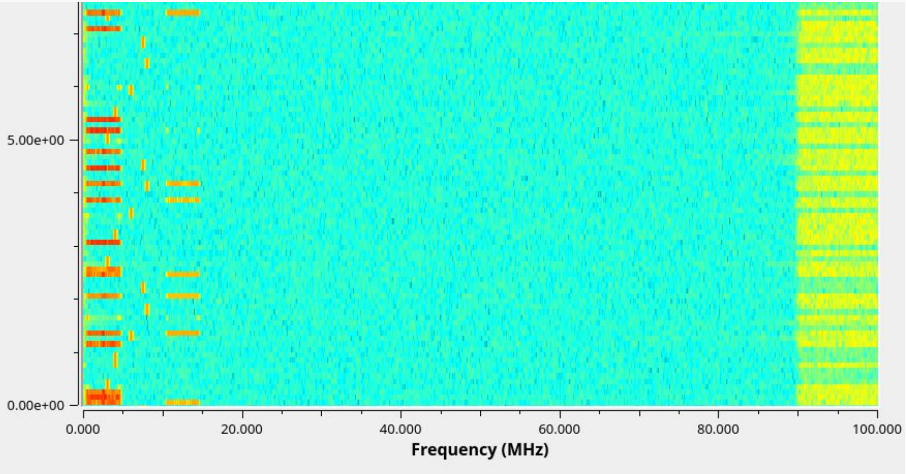


# *Project Overview*

- Our goal is to efficiently localize communication packets within a spectrogram

- Inputs: Protocol Info, i.e.
  - Frequency band(s)
  - Minimal Packet size

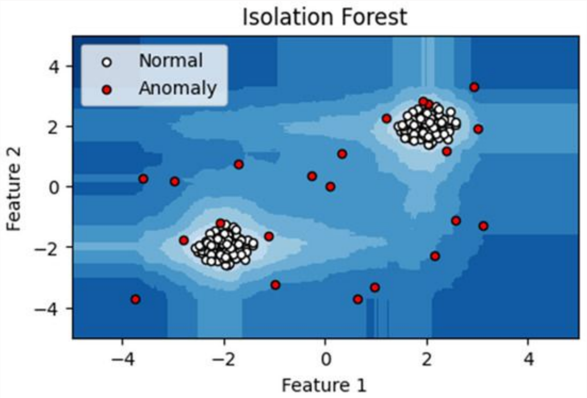
- Outputs: Bounding Boxes

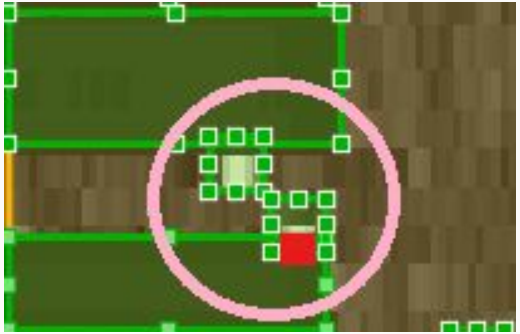


# Motivation

Precise localizations can help us find:

- Denial of service attacks
- Unexpected transmissions
- Interference

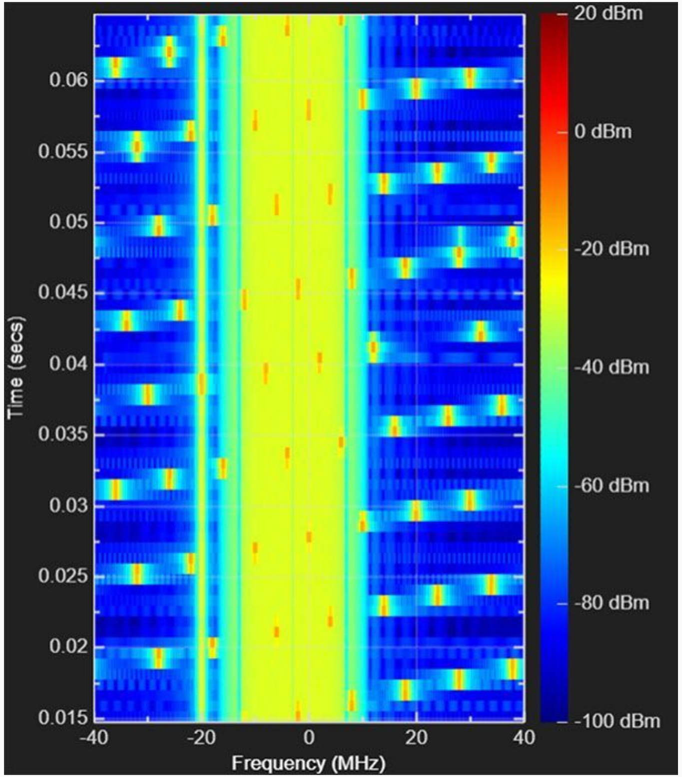


# *Previous Works*

## Previous works use:

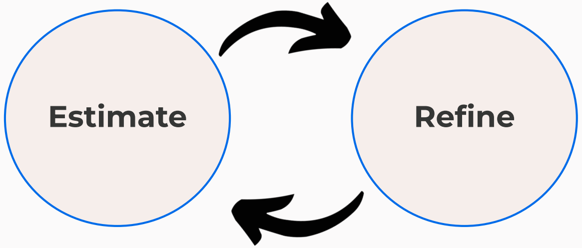
- Thresholding[1]
- Hypothesis testing
    - Neural networks[2]
- Iterative methods designed for images with arbitrary boundaries[3]

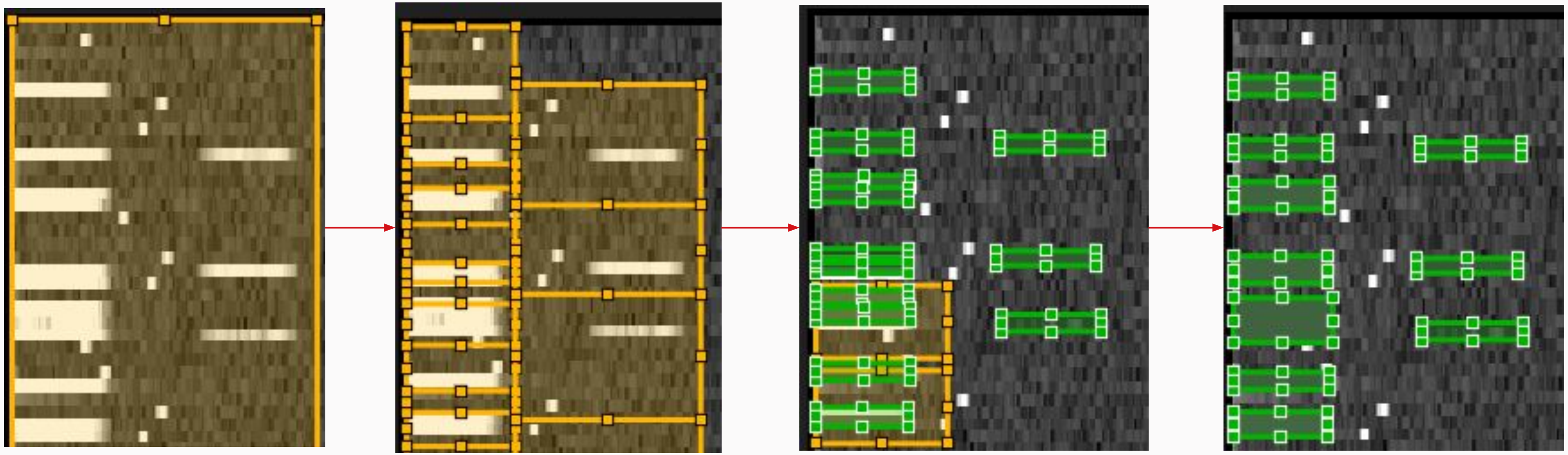




1. Neymotin et al. — Detecting Spontaneous Neural Oscillation Events in Primate Auditory Cortex.
2. Kang et al. — Joint Detection and Classification of Communication and Radar Signals in Congested RF Environments Using YOLOv8
3. Wali et al. — Enhanced medical image segmentation using novel level set evolution and efficient optimization, Comanciu et al. — Mean Shift: A Robust Approach Toward Feature Space Analysis

# Intro to Methodology

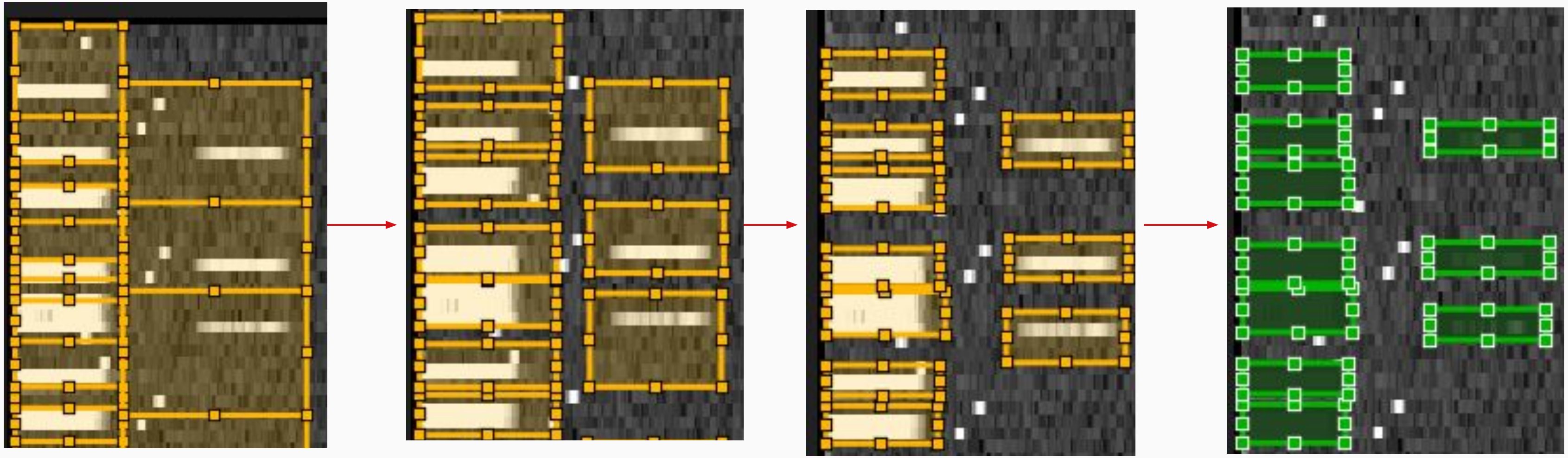


# *Intro to Methodology*

If you can quantify, you can optimize:

- An "energy" function quantifies how good a bounding box is
- We move/resize boxes to optimize an energy function $E(\mathbf{x}_i)$



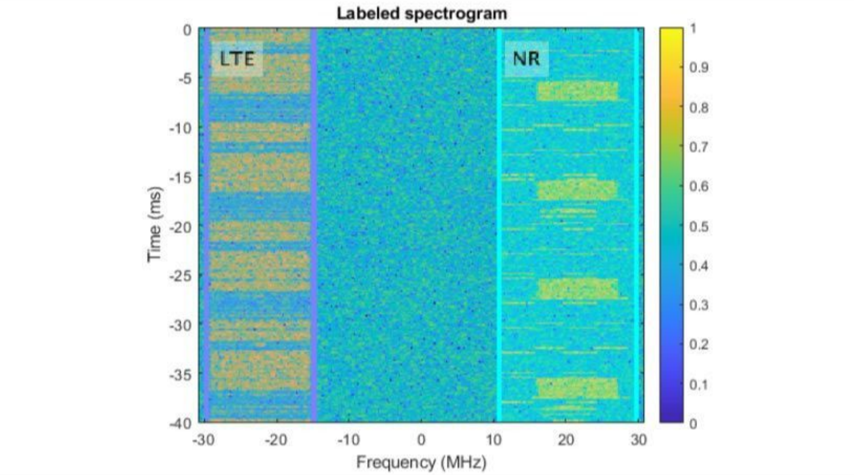
# *Formulation - Guiding Properties*

Assume we know:

- An estimate of the packet sizes
- An estimate of the noise threshold

Optimize bounding boxes for:

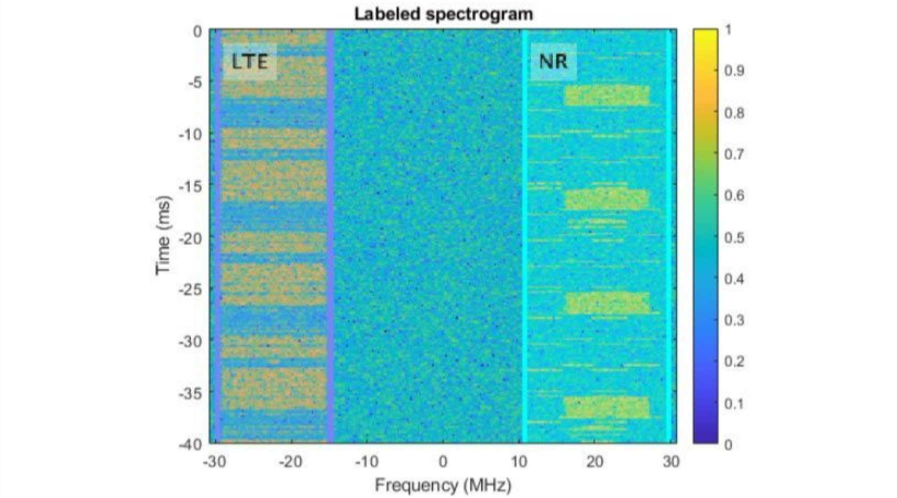
- Size similar to the expected packets
- Location near areas of high power
- Contents entirely filled with power



# *Formulation - Guiding Properties*

An optimal box:

- Matches the approximate size of a protocol packet
  - How do we quantify this?



# *Formulation - Size Constraint Term*

$$F_s(\mathbf{x}_i) = (\mathbf{s}(\mathbf{x}_i) - \mathbf{c_s})^T \mathbf{C}_\gamma (\mathbf{s}(\mathbf{x}_i) - \mathbf{c_s})$$

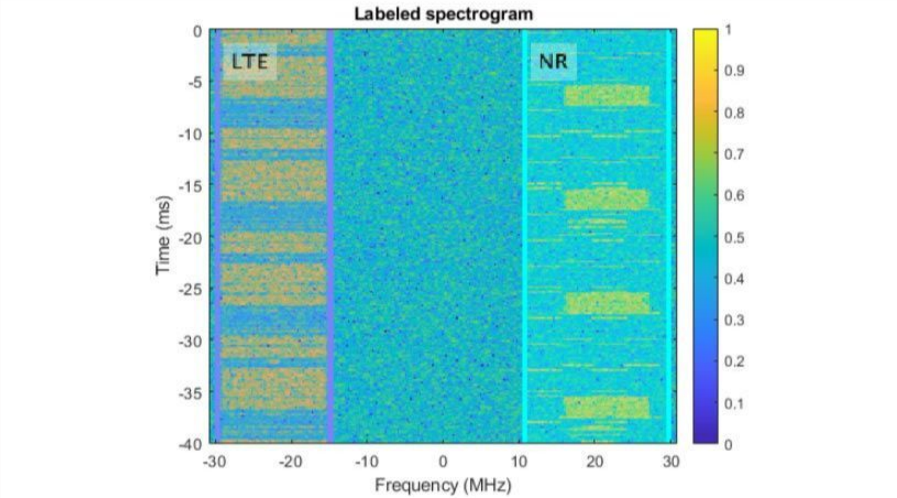
$F_s$ is a quadratic penalty based on <u>size</u>

$\mathbf{x_i}$ - rectangle coordinates
$\mathbf{C_\gamma}$ - weight matrix
$\mathbf{c_s}$ - smallest packet size

# *Formulation - Guiding Properties*

An optimal box:

- <u>Finds regions of high power</u>
  - How do we quantify this?

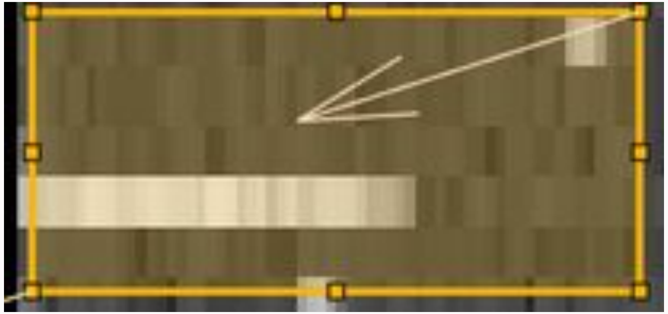


# *Formulation - Centroid Term*

$$F_c(\mathbf{x}_i) = \frac{1}{2}\tau\,||\bar{\mathbf{z}}(\mathbf{x}_i) - \mathbf{m}(\mathbf{x}_i)||_2^2$$

$F_c(\mathbf{x})$ is a quadratic penalty based on the distance between:

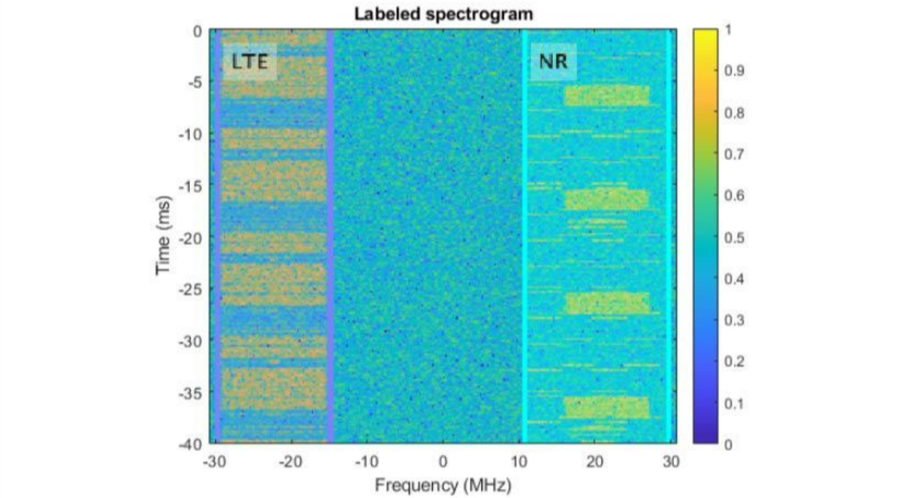
$\bar{\mathbf{z}}(\mathbf{x}_i)$ - <u>centroid</u>
$\mathbf{m}(\mathbf{x})$ - center



# *Formulation - Intro to Methodology*

An optimal box:

- Encompasses entire packet regions
    - How do quantify this?



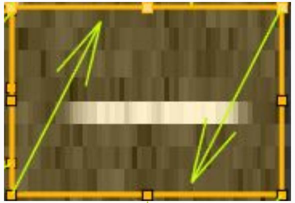
# *Formulation - Centroid Term*

$$G(\mathbf{x}_i) = 1 - \frac{\int_{R_i} [P > \nu] \, d\mu}{|R_i|}$$

$G(\mathbf{x})$: proportion of P below the noise threshold $\nu$.

Multiplying by $G(\mathbf{x})$ provides a new way of reaching minimal energy: be full of power.

# *Formulation - Energy Functional*

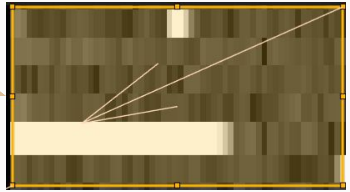


$F_s$ causes shrinking up to a minimum size

$$E(\mathbf{x}_i) = G(\mathbf{x}_i)(F_s(\mathbf{x}_i) + F_c(\mathbf{x}_i))$$

G acts as a counterbalance; full boxes are also happy boxes.

$F_c$ moves the geometric center toward the centroid



# *Formulation - Gradient*

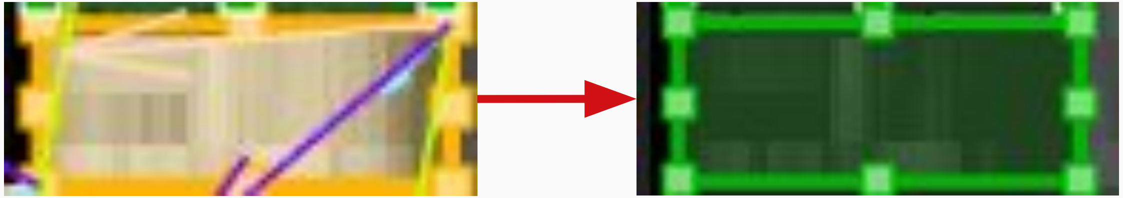
$$dE = \nabla E \cdot d\mathbf{x}_i$$

Iteratively deforming boxes by $\nabla E(\mathbf{x}_i)$ leads to optimal bounding boxes as measured by E.

# *Formulation - Hiding*

Once a box has mostly stopped moving, its contents are hidden from the spectrogram.

This prevents:
- Repeat boxes
- Obstacles in identifying smaller packets



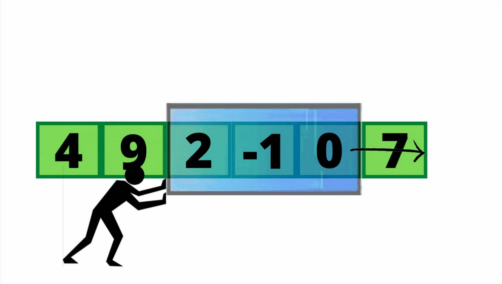
# *Estimation / Splitting*

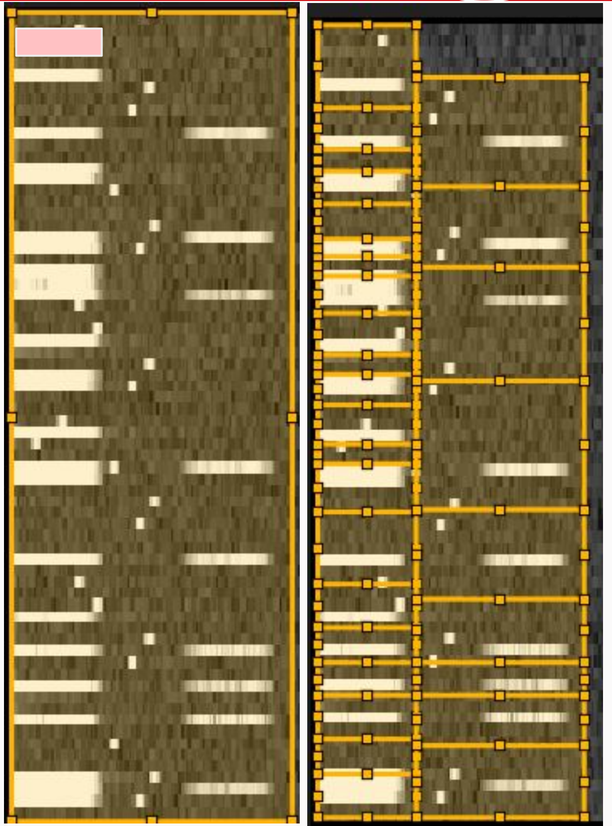
Now that we know how to refine, we need to figure out how to estimate:

1. Slide a packet-shaped shadow across dimension reduced spectrograms

2. Peaks ≈ centers

3. Reconstruct edges

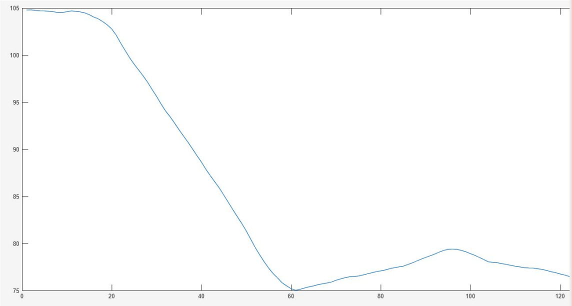


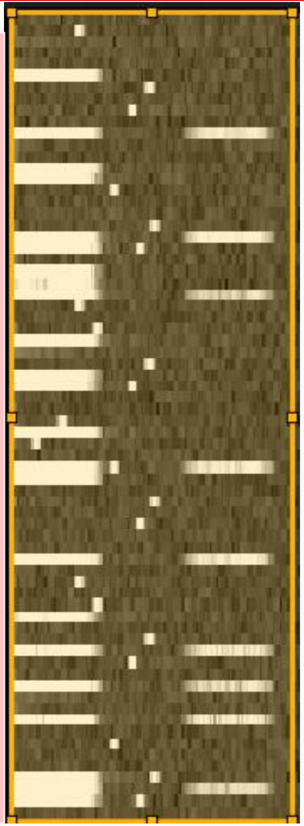



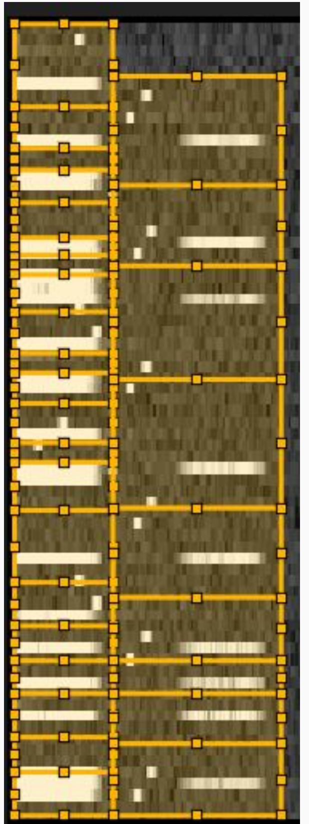
# *Algorithm Improvements - Splitting*

Benefits:

- Resistant to noise
- Ignores packets of different protocols
- Very fast — based on convolution





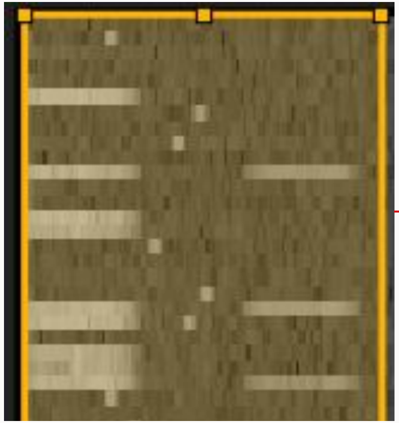
# *Algorithm Improvements - Otsu's Method*

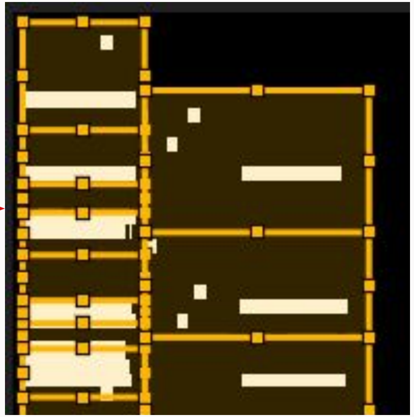
What happens if a bounding box is wrong?

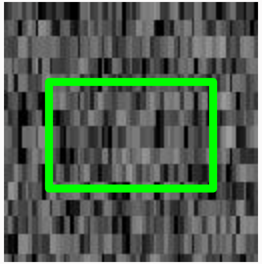
1. Calculate noise threshold with Otsu's method (maximize intraclass variance of pixels)

2. Calculate fullness, i,e. % above noise

3. Remove boxes with fullness below a threshold







# *Demo from MATLAB*

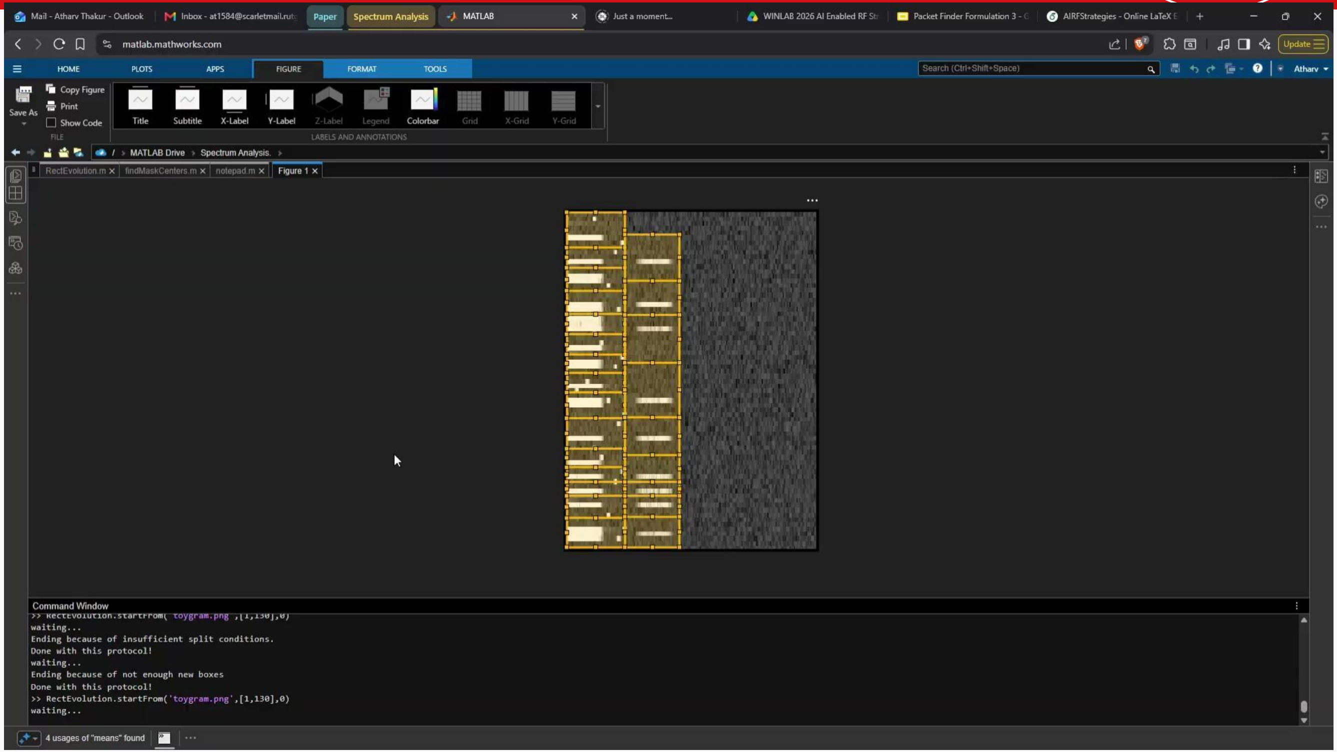
# Demo

The arrows represent different contributions to box motion

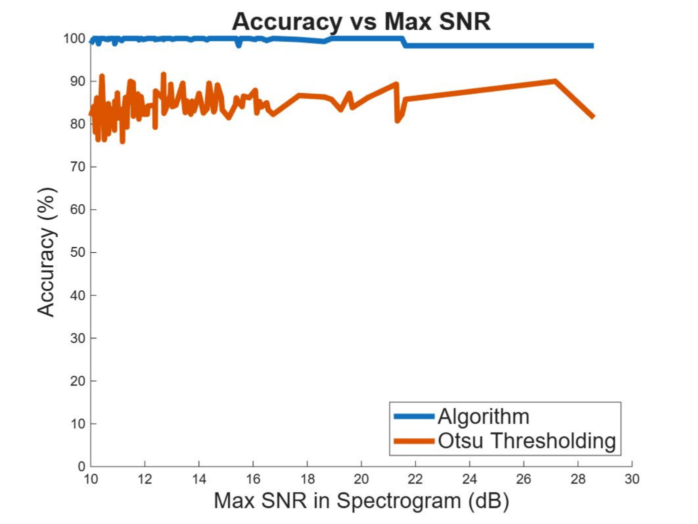
Green - match size of minimal element

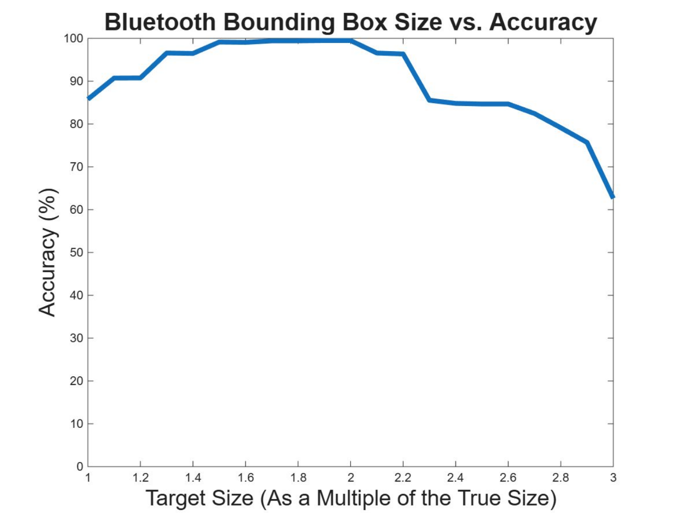
Tan - match geometric center to power centroid

Blue - fullness pushback



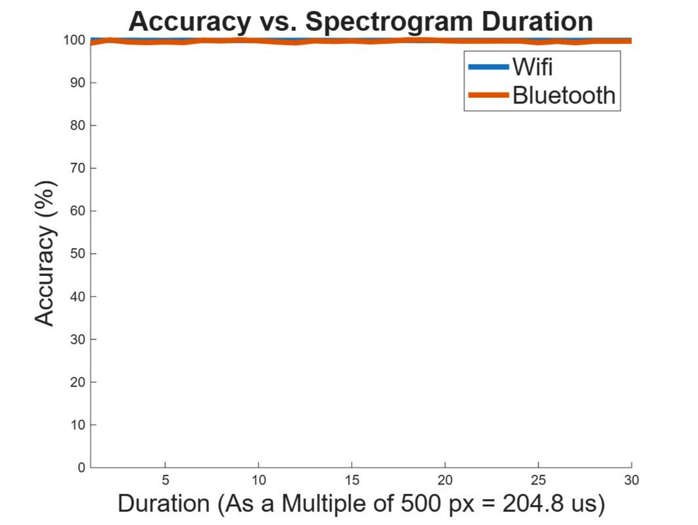
# *Benchmarking*

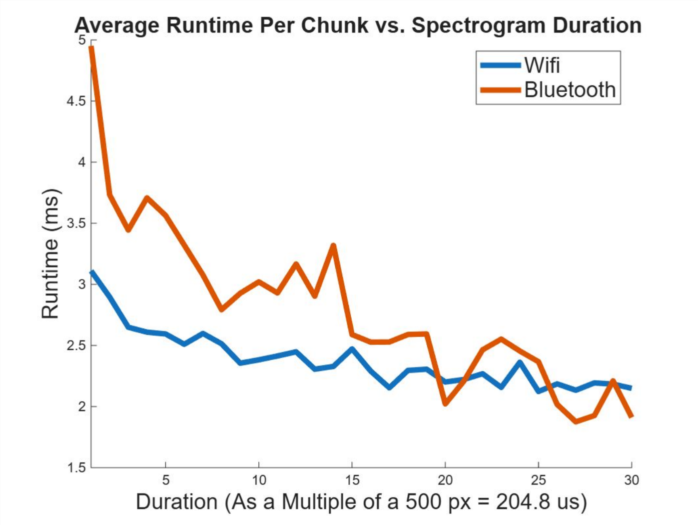



- Tested on 100 labeled GNU radio simulations of 204.8 µs each with spectrogram image size 100x500 px and max SNR $\in$ [10dB, 30dB]
- Algorithm achieves ~99.9% accuracy with manual parameters

# *Benchmarking*

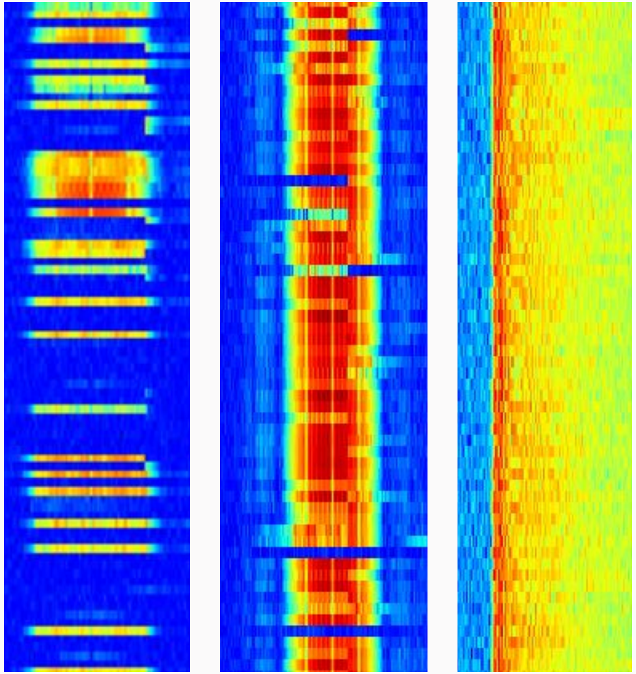



- Benchmarked in MATLAB Online (reproducible, no parallelization)
- Greedy algorithm, more efficient as input size increases

# *Benchmarking*

- Also tested on public dataset of over-the-air drone transmissions
- Preliminary results with manual rules:
    - Normal — (99%, ~14 ms)
    - Denial of Service — (94%, ~23 ms)
    - Jamming — (100%, ~17 ms)
- This is a lower bound, systematic methods would perform much better
- Original study uses neural networks, (98.9% accuracy, ~550 ms), requires a GPU.




1. Cengizhan Yapıcıoğlu, Sedef Demirci, Mehmet Demirci,DeepSpect: An RF spectrogram-based deep learning approach for near-real-time attack detection in FANETs,

# *Thanks! Any Questions?*