Wavelet-Based Extraction of Transient Noise in Gravitational-Wave Interferometers using a Learning Architecture Guided by Manifold Clustering

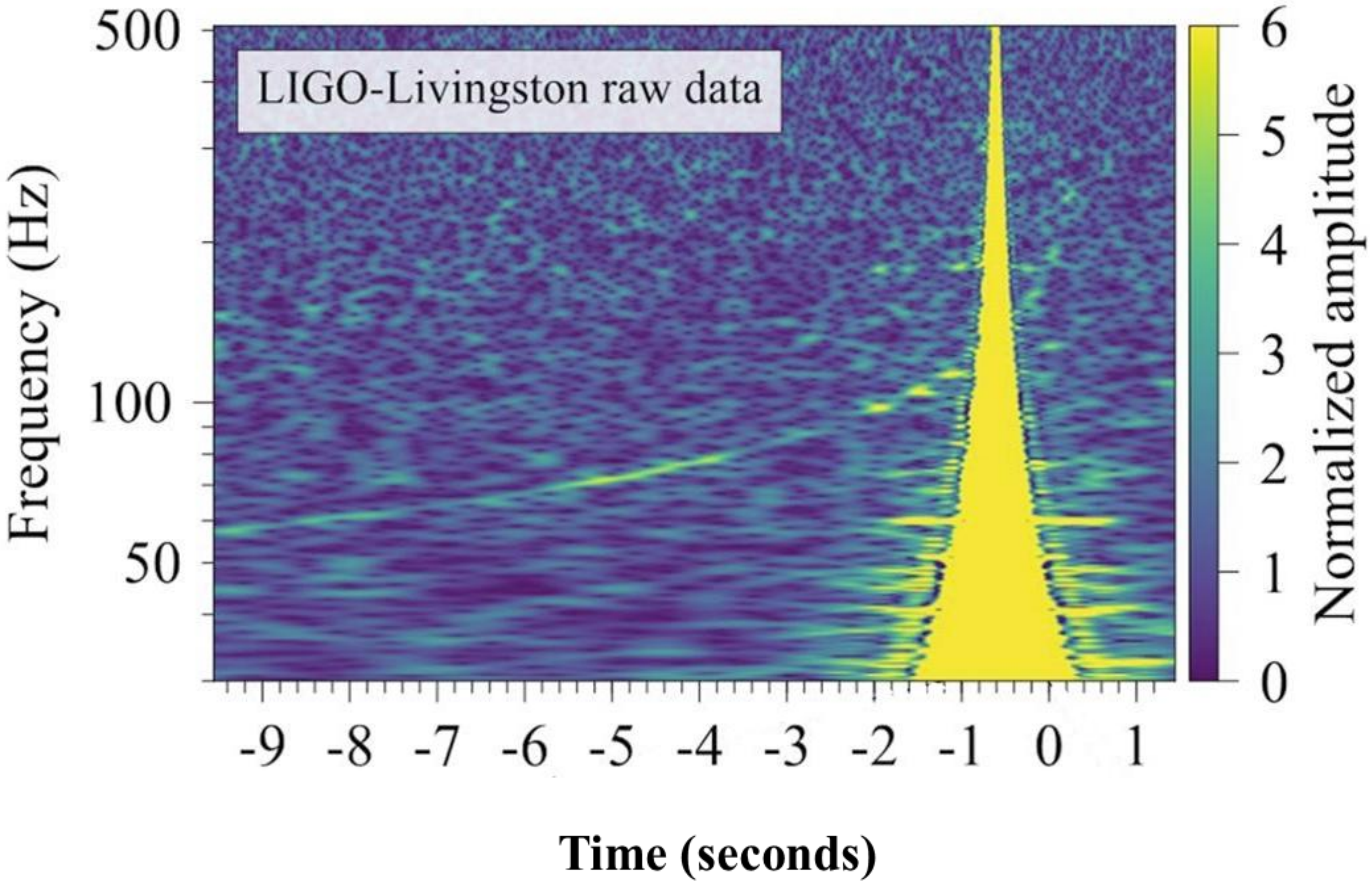
Christopher Alléné,

January the 22nd, 2026

東京都市大学
TOKYO CITY UNIVERSITY

JAPAN SOCIETY FOR THE PROMOTION OF SCIENCE • FOUNDED 1932 •
JSPS

Glitch



Glitch:

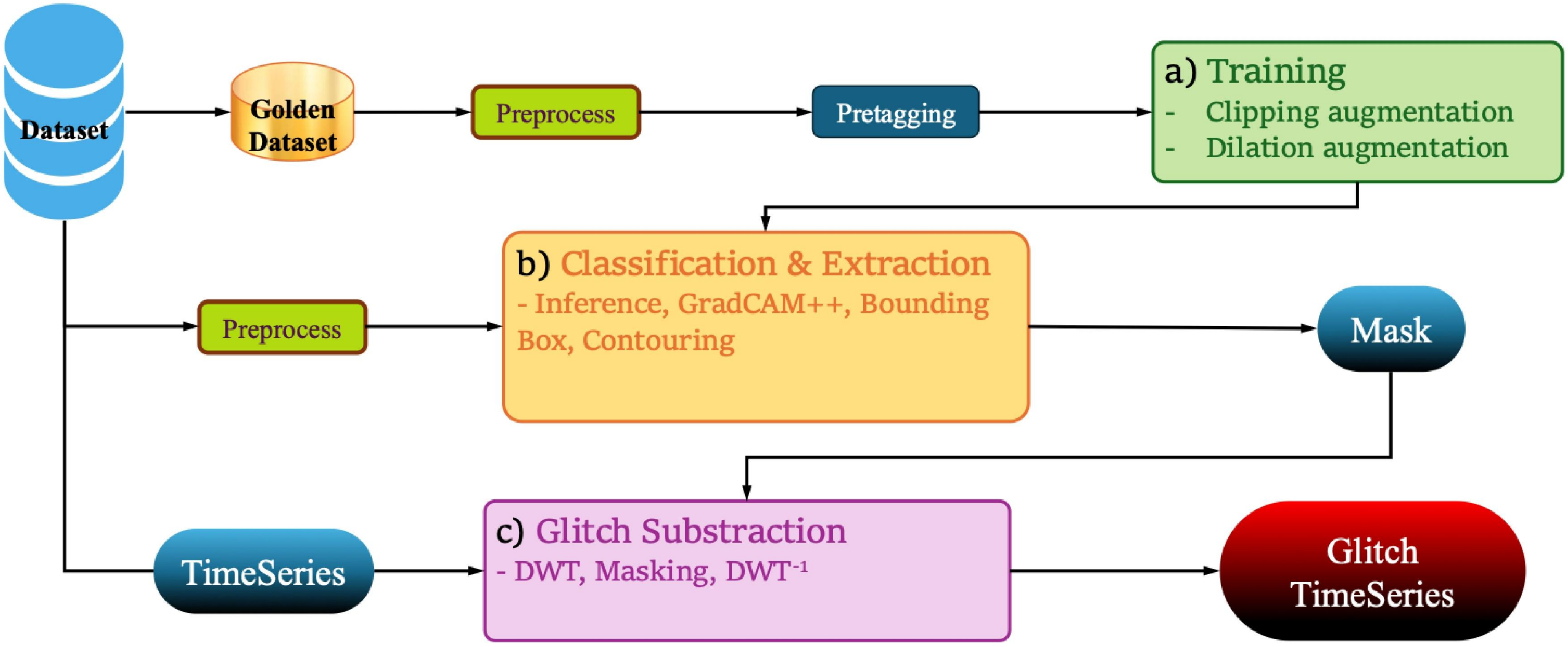
- Short-duration transient noise;
- Can be superimposed on astrophysical signal;
- Can be very loud;
- Can last until ~4 s;

Motivation

- To isolate glitches;
- Better understanding of the noise (non-gaussian one especially);
- Be able to produce bank of glitch to simulate noise with non-gaussianities;

⇒ We need then to be able to **segment** and to **subtract** glitches

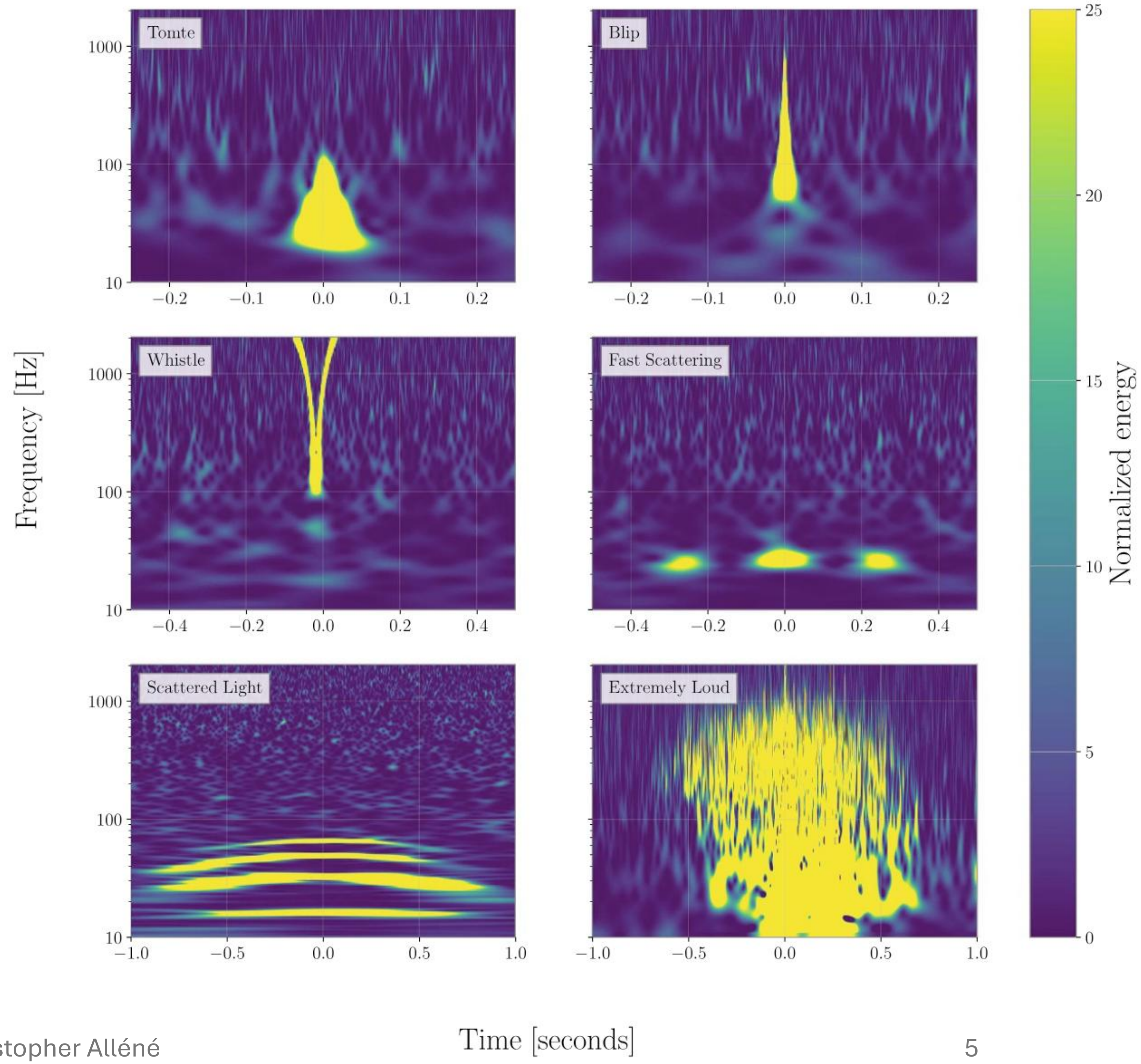
Workflow


Dataset
Golden Dataset
Preprocess
Pretagging
a) Training
- Clipping augmentation
- Dilation augmentation
Preprocess
b) Classification & Extraction
- Inference, GradCAM++, Bounding Box, Contouring
Mask
TimeSeries
c) Glitch Substraction
- DWT, Masking, DWT-1
Glitch TimeSeries

Golden Dataset Sele

- Among 26 classes* focus on Blip, Whistle or Scattered Light;
- Equipartition of event type
- GravitySpy CL> 95%
- No siblings or neighbour events
  - No events in a ±2 s window

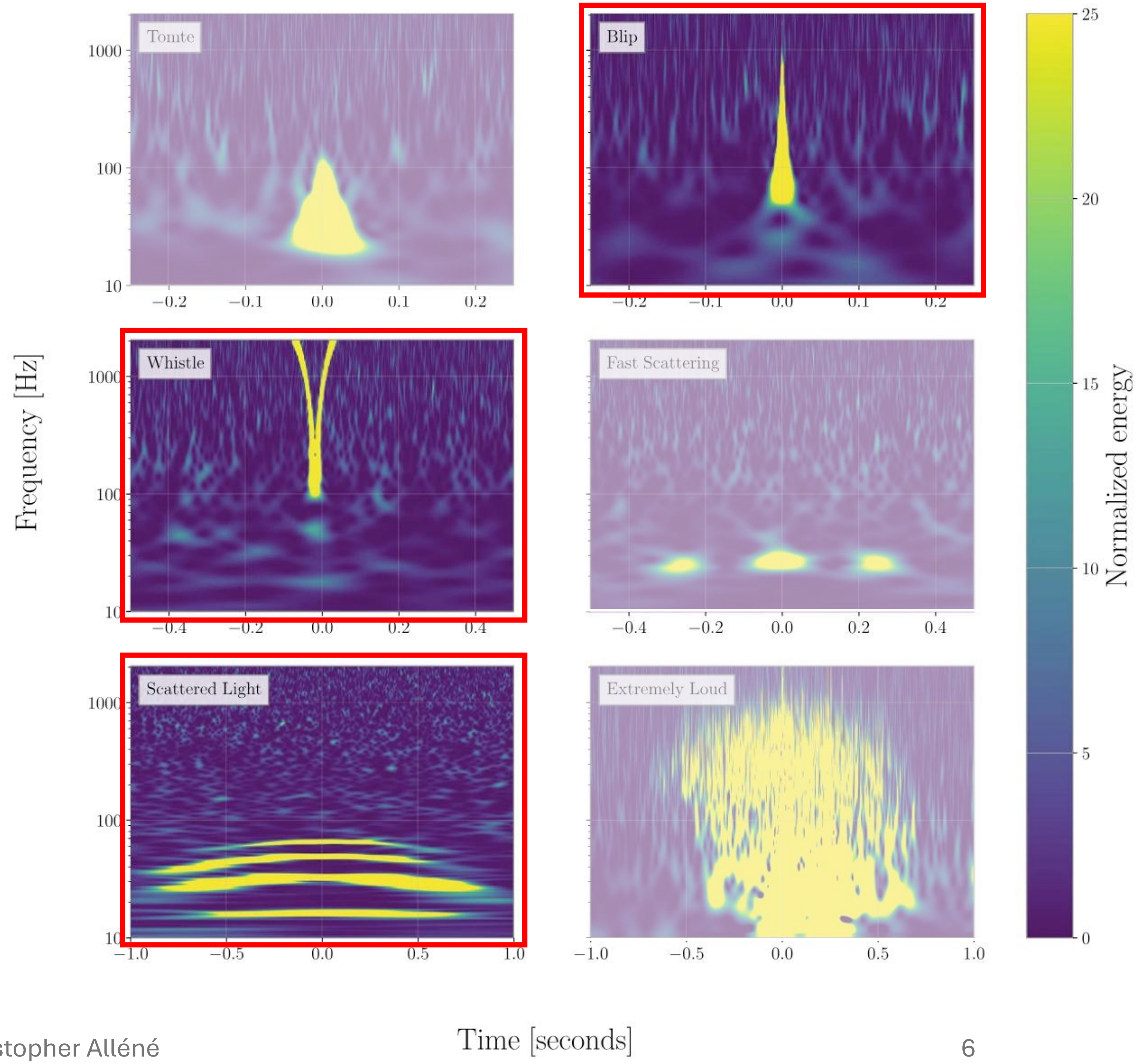
*according the GravitySpy classification (Zevin et al. 2017)



Golden Dataset Sele

- Among 26 classes* focus on Blip, Whistle or Scattered Light;
- Equipartition of event type
- GravitySpy CL> 95%
- No siblings or neighbour events
  - No events in a ±2 s window

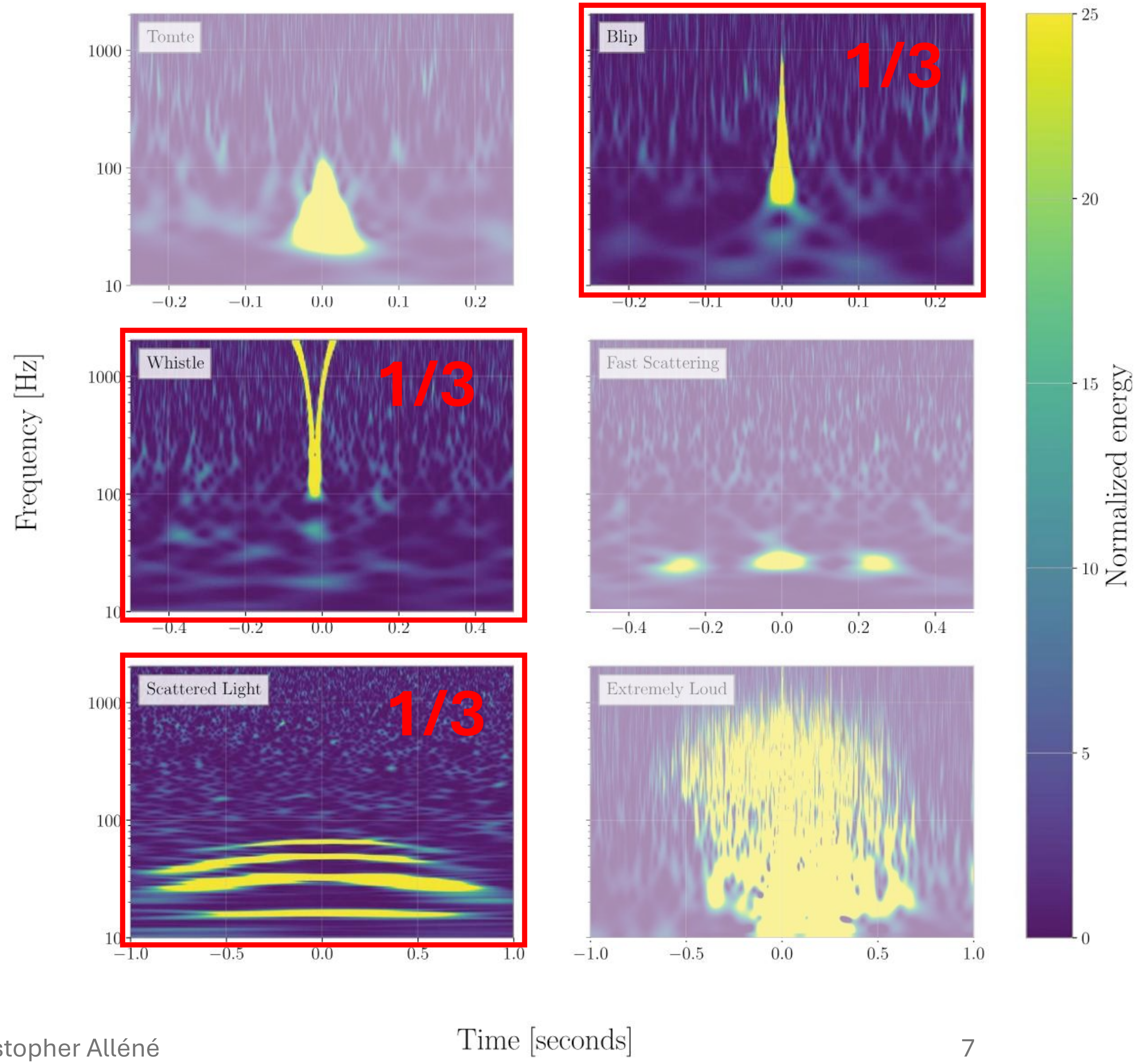
*according the GravitySpy classification (Zevin et al. 2017)



Golden Dataset Sele

- Among 26 classes* focus on Blip, Whistle or Scattered Light;
- Equipartition of event type
- GravitySpy CL> 95%
- No siblings or neighbour events
  - No events in a ±2 s window

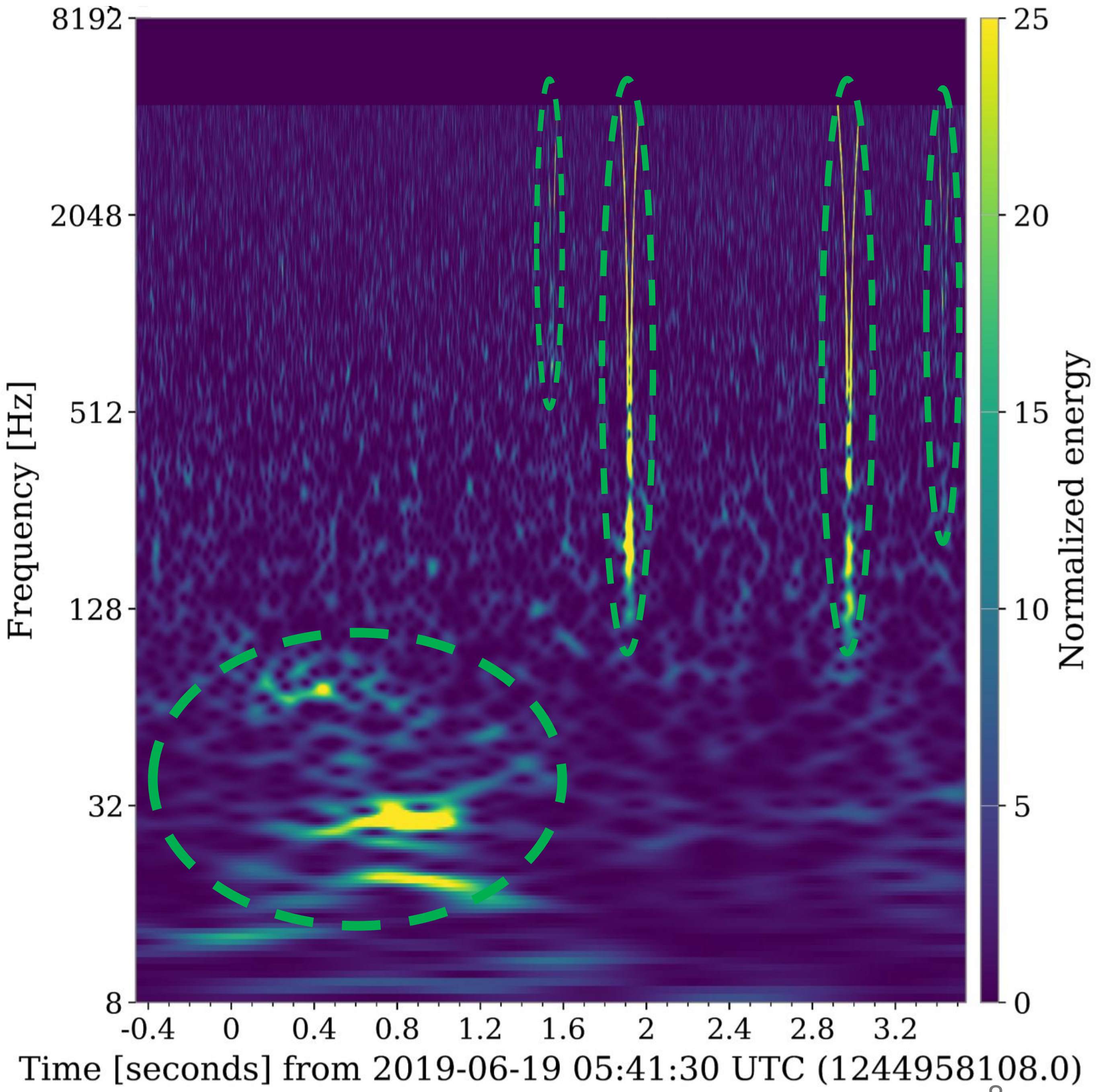
*according the GravitySpy classification (Zevin et al. 2017)



Golden Dataset Selection

- Among 26 classes* focus on Blip, Whistle or Scattered Light;
- Equipartition of event type
- GravitySpy CL> 95%
- No siblings or neighbour events
  - No events in a ±2 s window

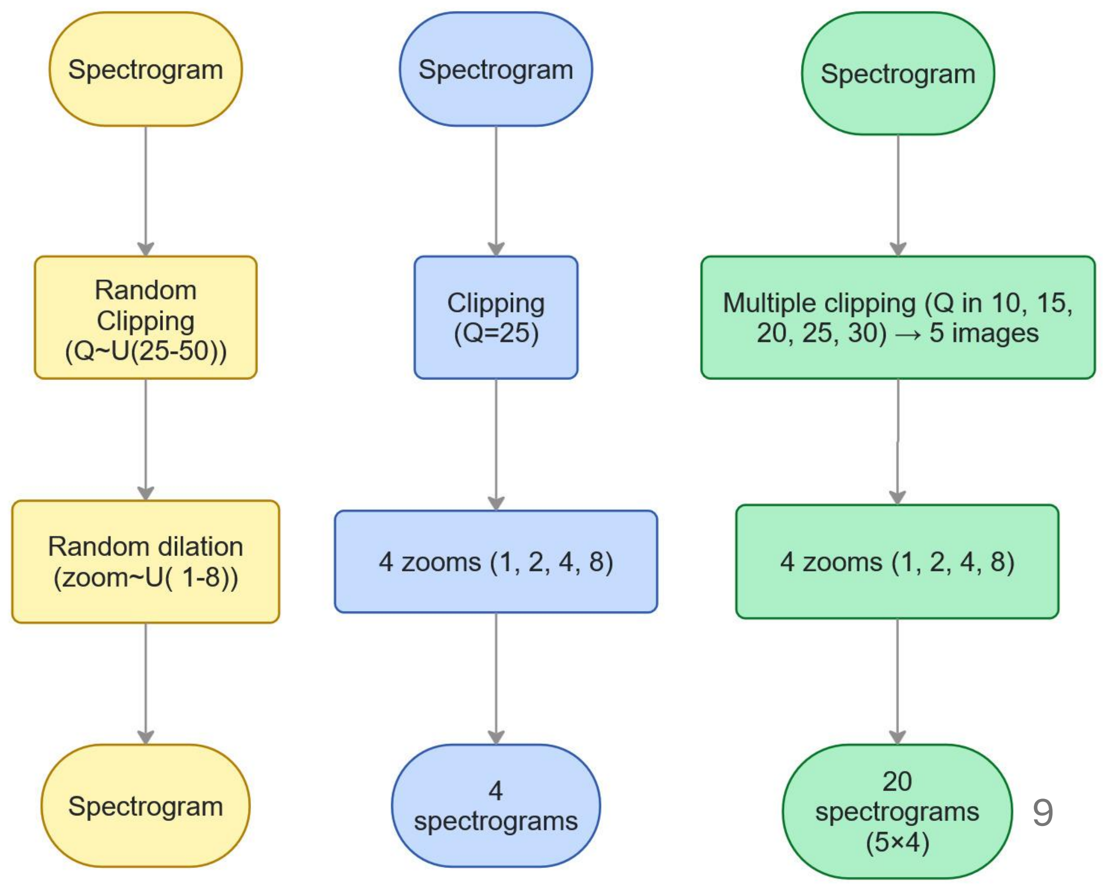
*according the GravitySpy classification (Zevin et al. 2017)



Preprocessing and data augmentation

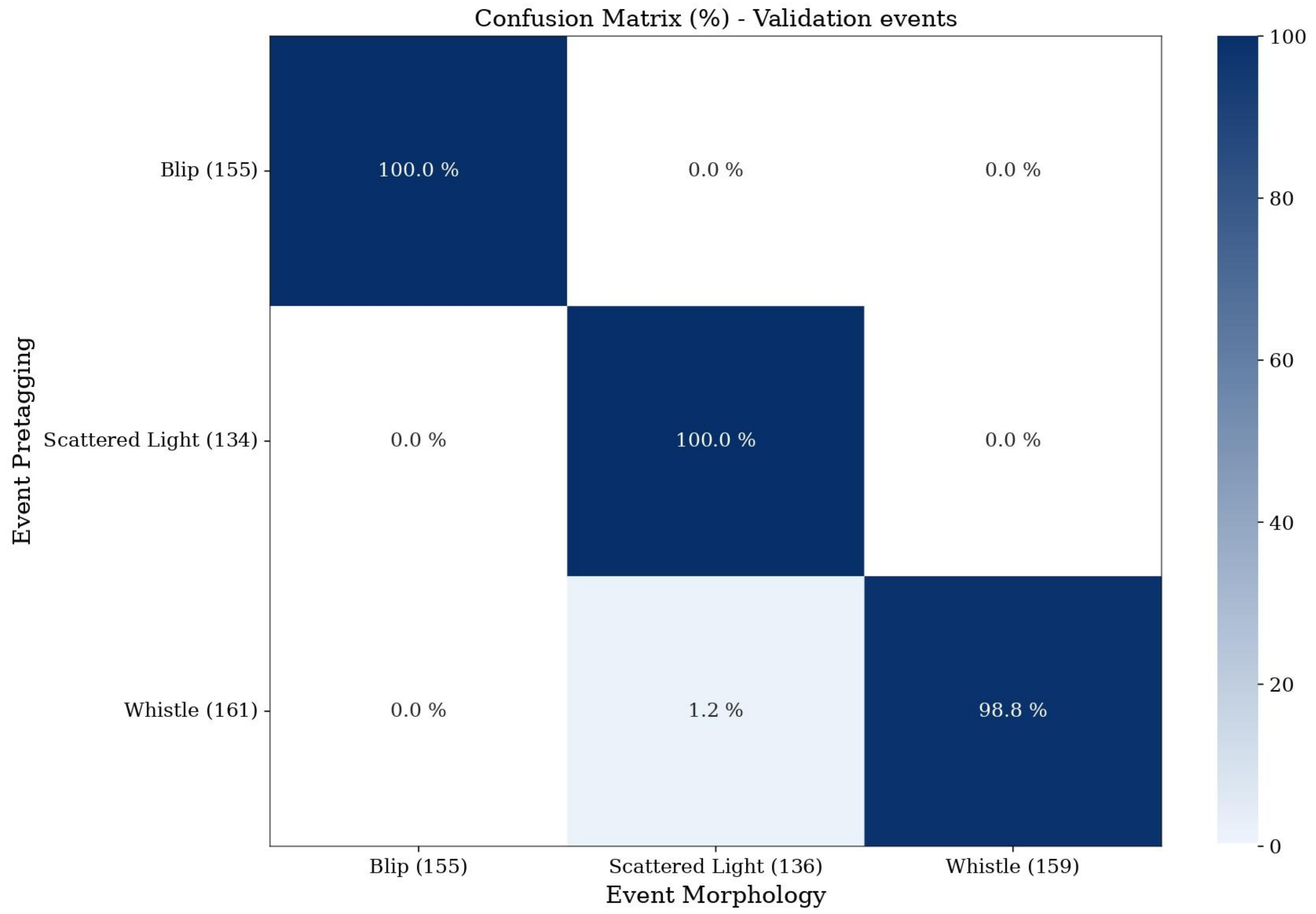
- Q-transform the Timeseries signal;
- Keep frequencies between 8 Hz and 8192 Hz;
- Keep only a 4 second window centered on the trigger;
- Put the frequency axis in log-scale;
- Apply the data augmentation;
- Add padding on the 4 borders;
- Resize to have a 224×224 image;

Data augmentation



Training

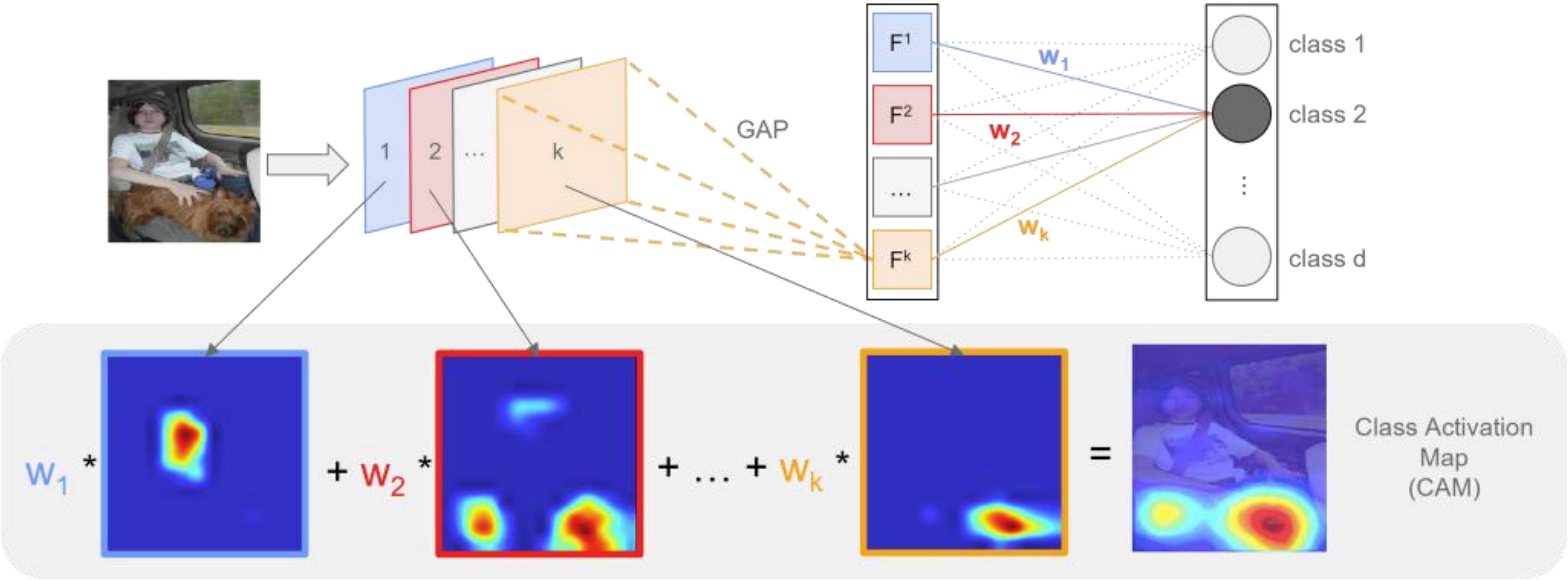
- ResNet50 Model basis;
- Trained on Golden Dataset;
- Pre-processed events;
- Curiculum Learning (based on triggers SNR level;
- 4500 events dataset (1500 of each class);
  - 70% training set
  - 20% validation set
  - 10% test set
- 50 epochs;



⇒ (a) Glitch events can be classified

Class Activation Map

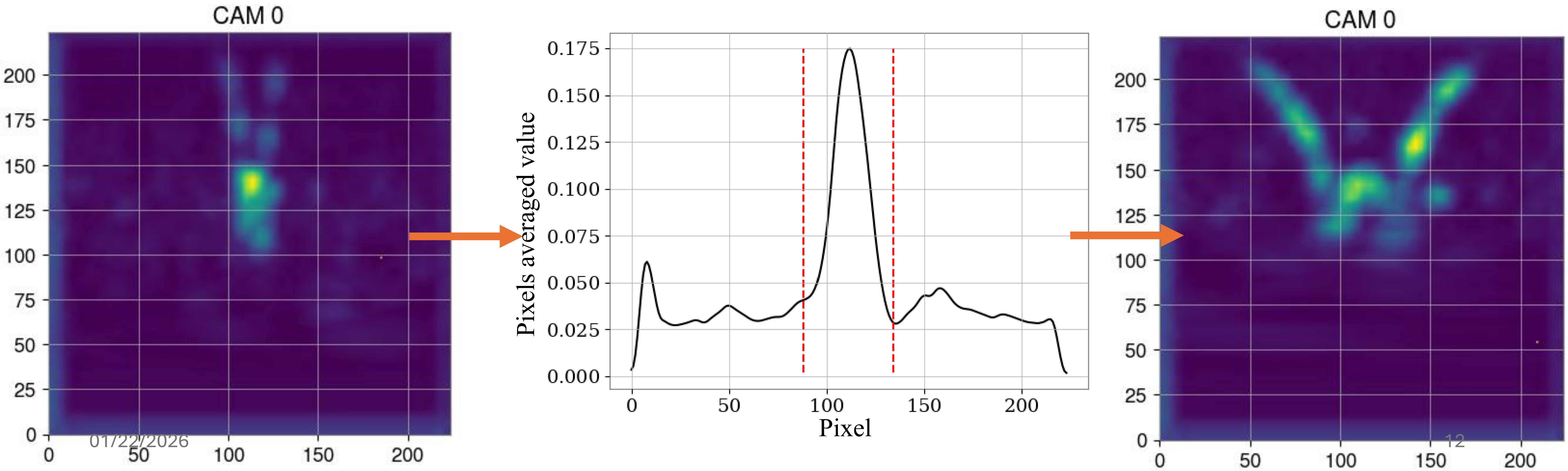
Having a heatmap of the activate neurons on a given layer:
➢Tells where is the relevant information on the input image.



*Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016

From CAM to zoom selection

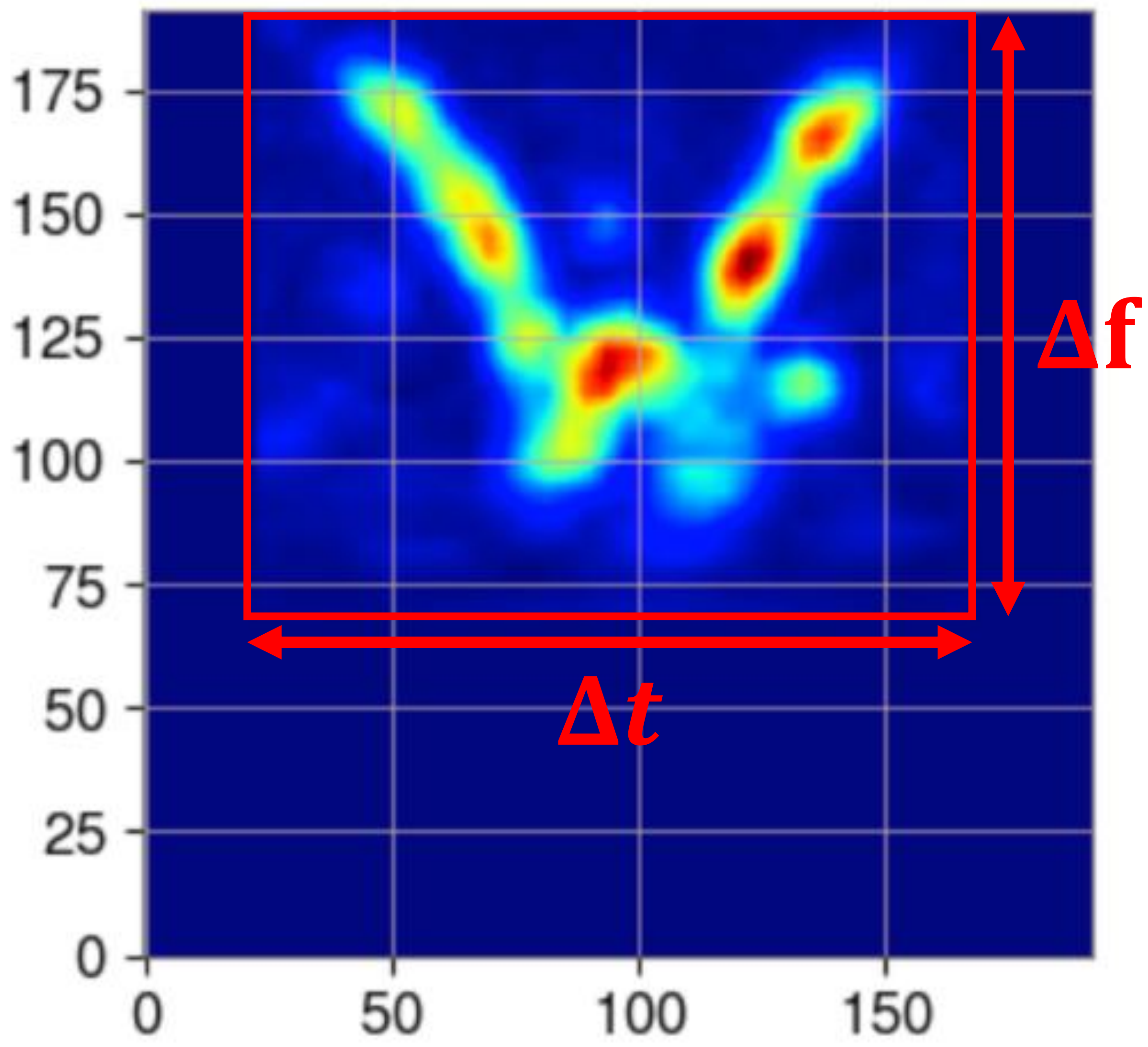
- Layer of the first bottleneck;
- Integrate over the frequency axis;
- Refine the zoom level by estimating the event duration;



Glitch Emboxing

Do once more the integrations of the CAM;

- First on frequencies;
  - Gives time duration Δt;
- Second over time;
  - Gives frequency bandwidth Δf;



⇒ (b) First glitch extraction

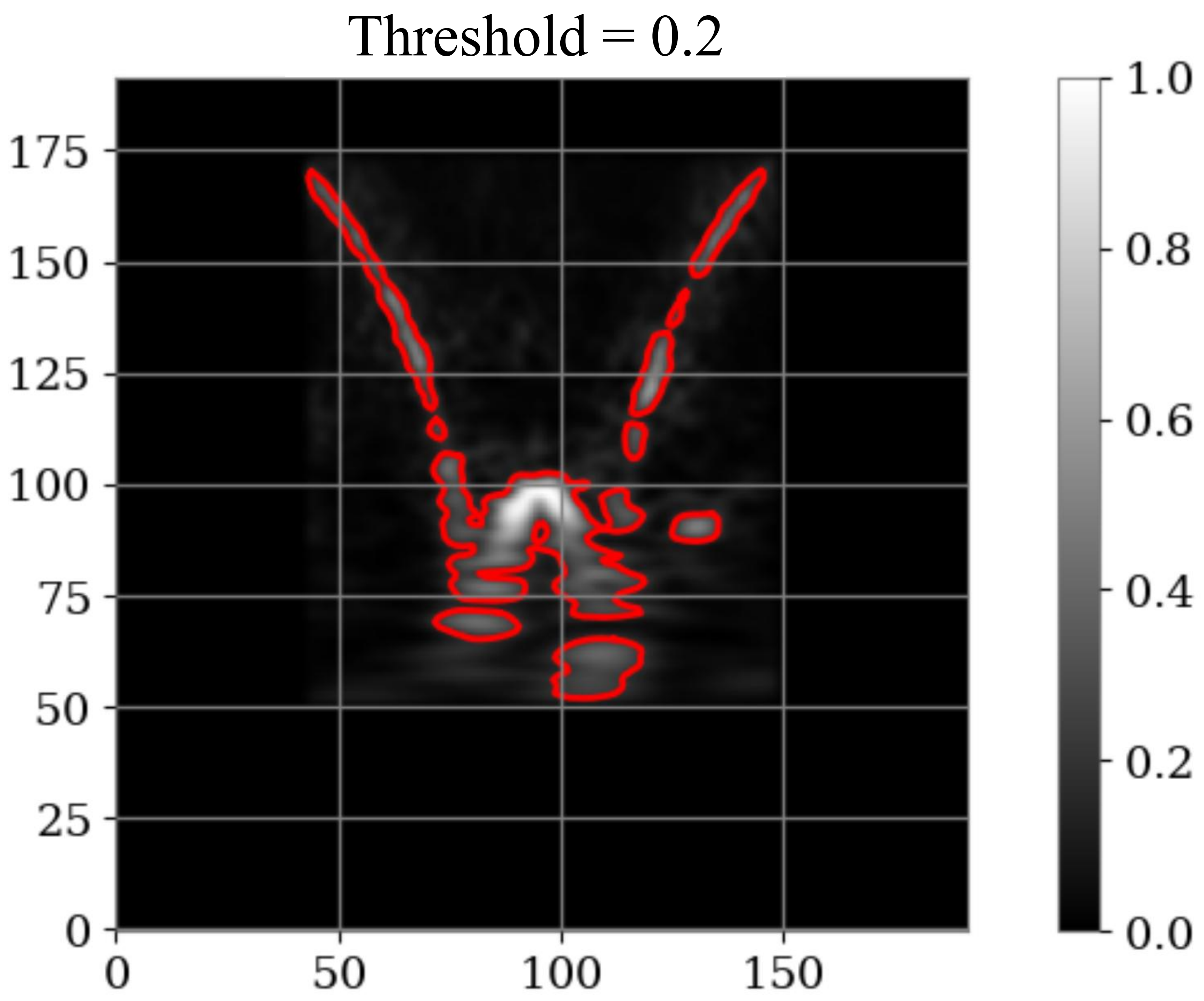
Contouring & Mask

Preprocessing the contouring:

- Make Rectangular window (of size $\Delta t \times \Delta f$);
- Apply the rectangular window to input and CAM;
- Contouring for {5, 10, … , 50}% thresholds on:

$$\mathrm{w_{ij}} = \sqrt{input_{ij} \times CAM_{ij}}$$

- Fill the contour with ones to make binary mask base.



Finding optimal contour threshold

Protocol to statistically test the threshold:

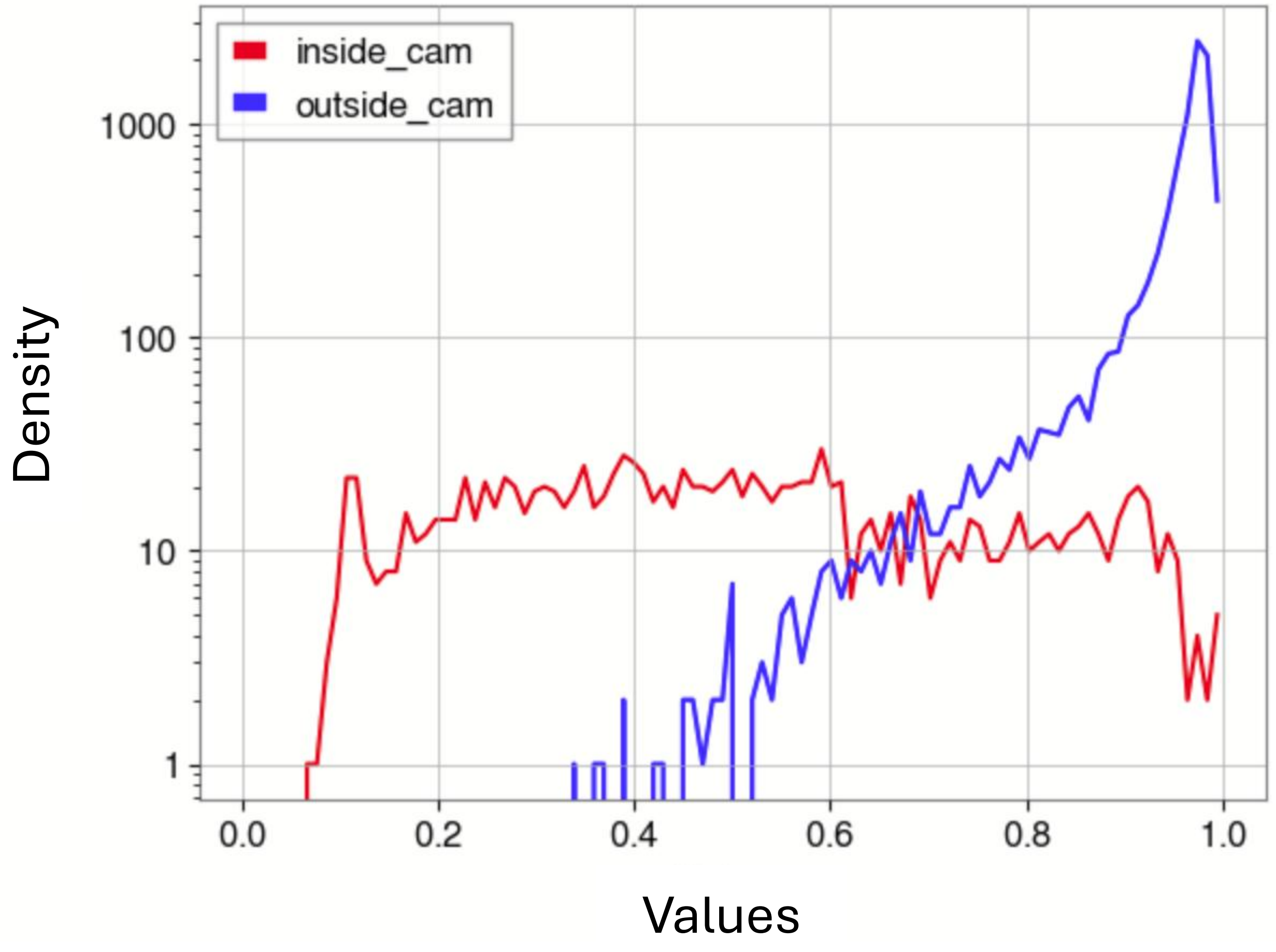
- Normalize the CAM
- Apply the binary mask (contouring result) on CAM :

$$inside_{cam,ij} = mask_{\mathrm{ij}} \times CAM_{ij}$$

- Apply the inverse mask on CAM, then invert it again :

$$outside_{cam,ij} = 1 - \left(1 - mask_{ij}\right) \times CAM_{\mathrm{ij}}$$

- Make the associated distributions : (p_{in}, p_{out})
- Get a statistic of it



Statistic to optimize

Maximize the common surface under both of the histograms :

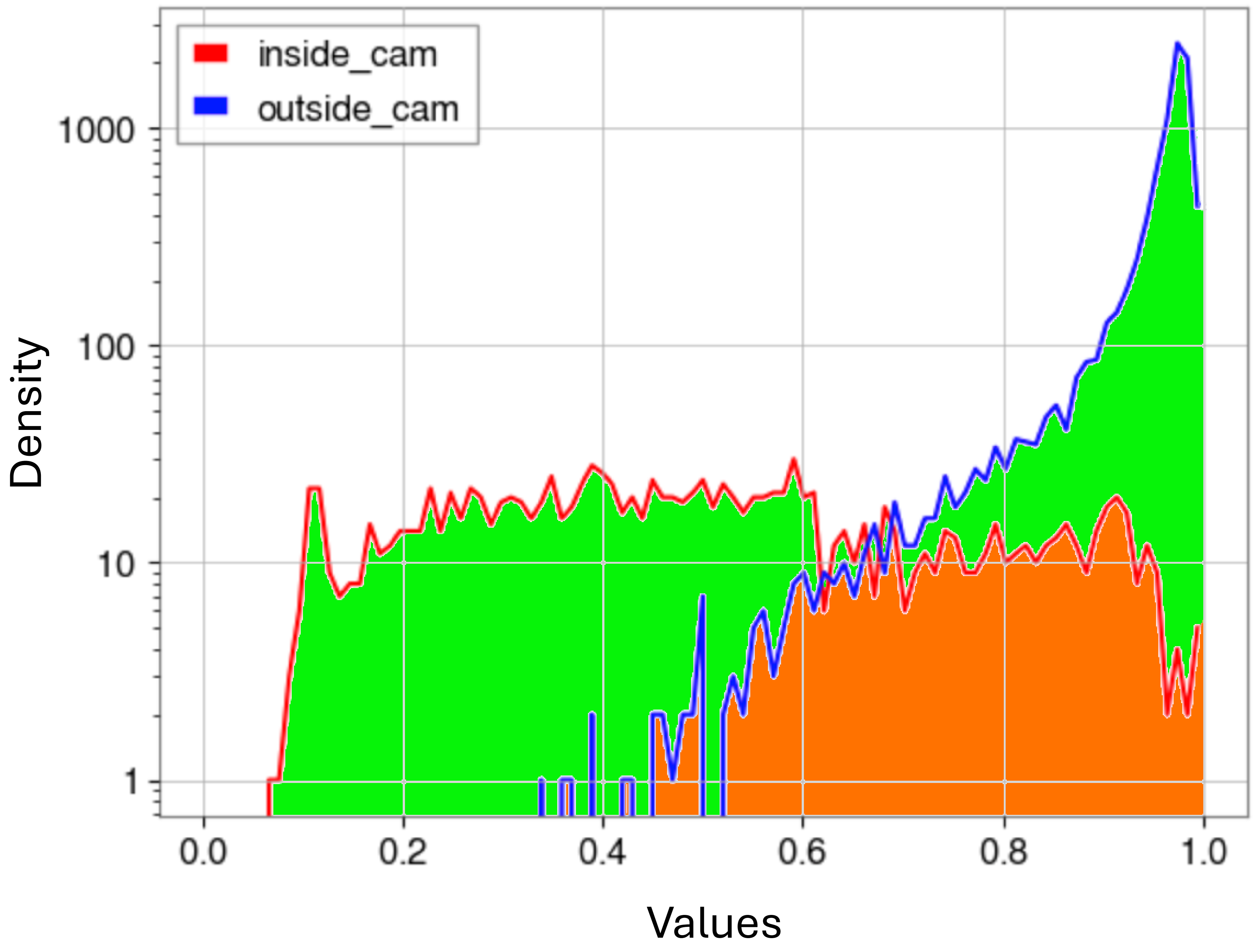
$$\eta_{min} = \sum_{i}^{N} \min(p_{in,i}, p_{out,i})$$

Minimize the difference between the 2 histograms :

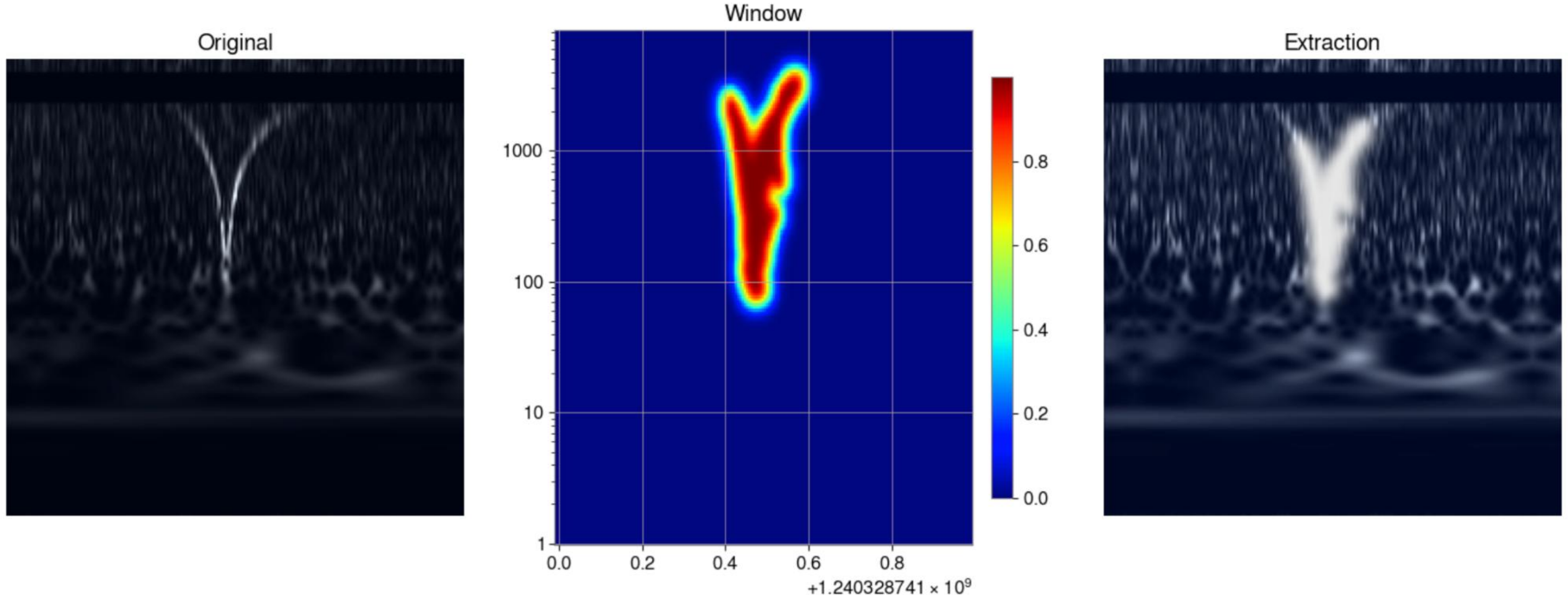
$$\eta_{diff} = \sum_{i}^{N} |p_{in,i} - p_{out,i}|$$

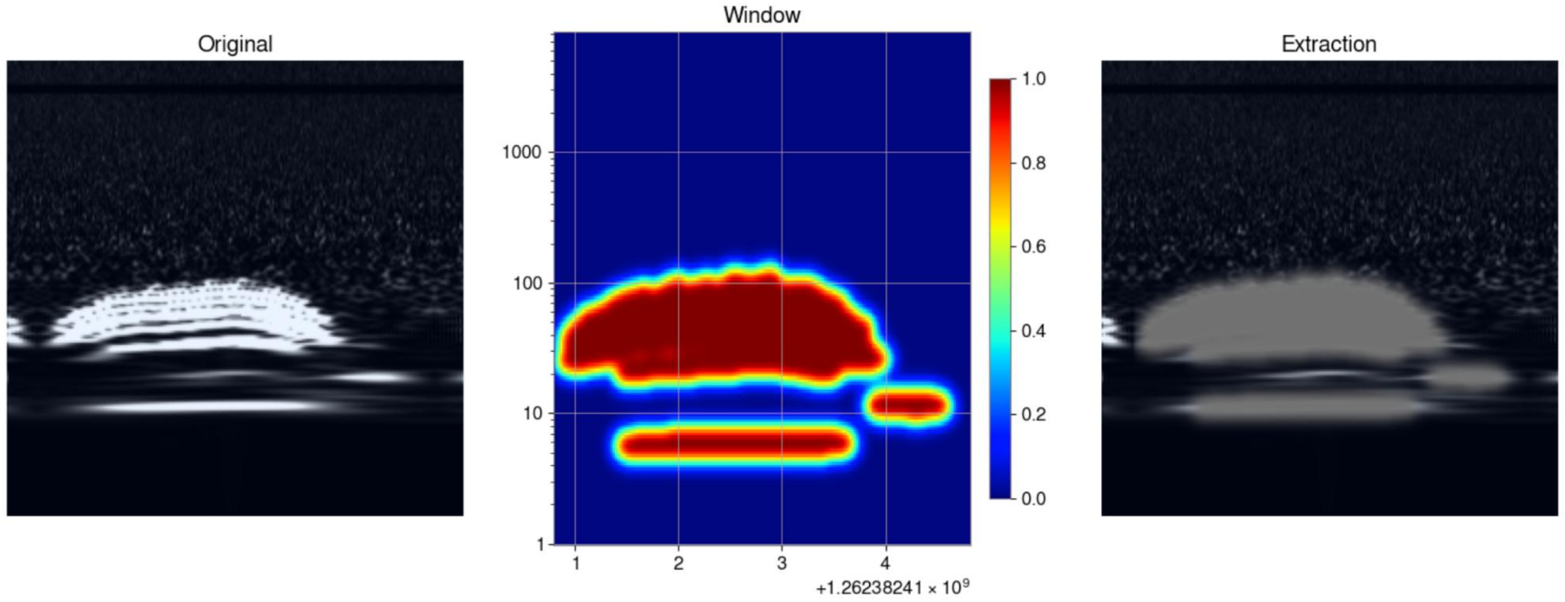
Find the threshold maximizing :

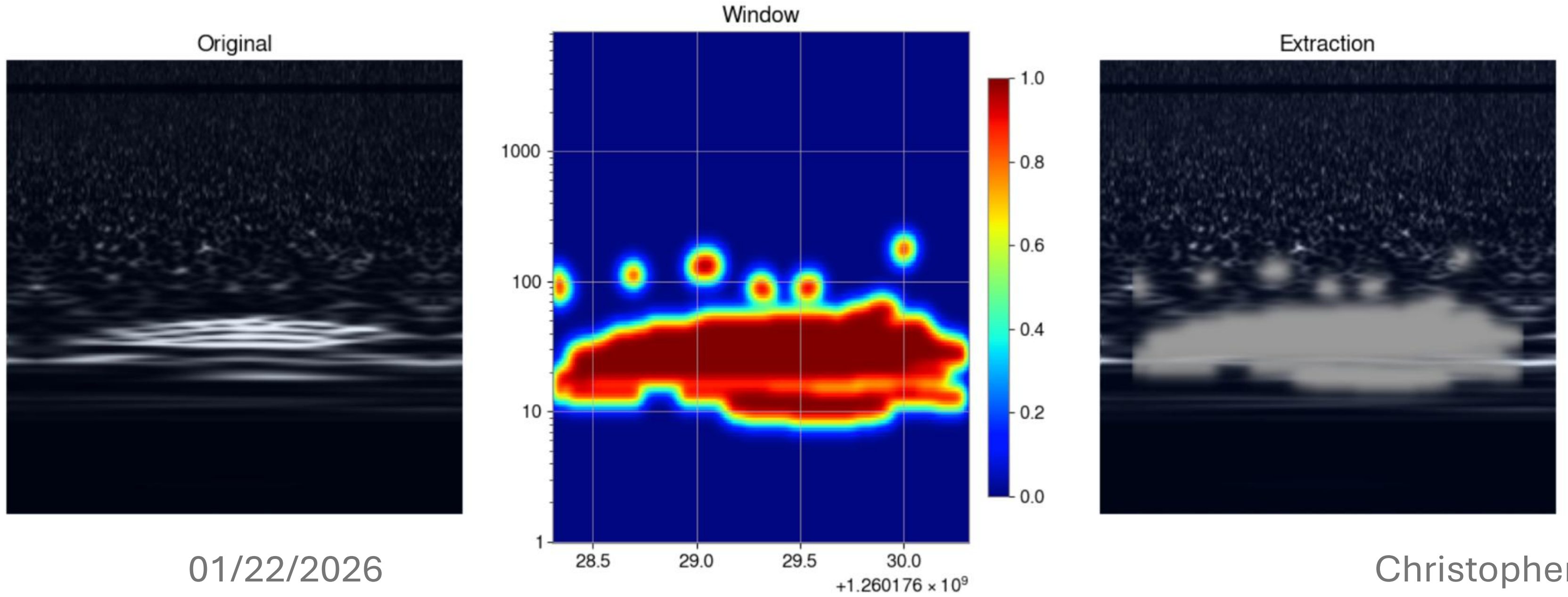
$$\eta_{tot} = \eta_{min} - \eta_{diff}$$

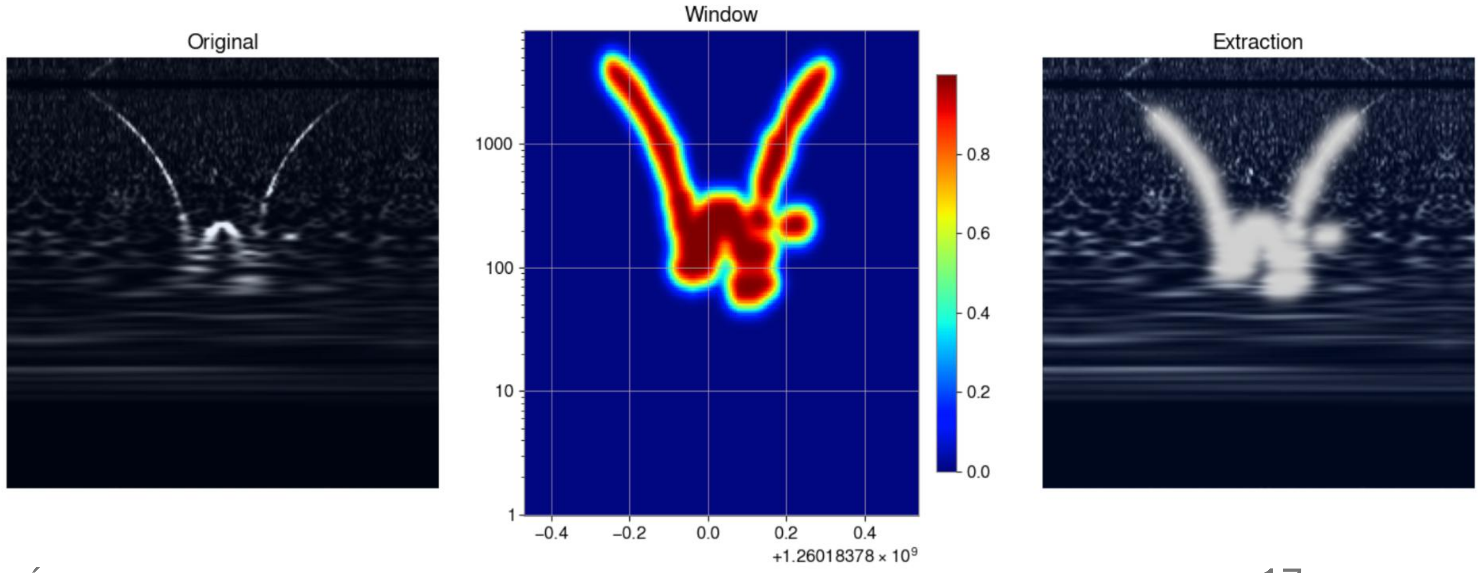


Example of masking

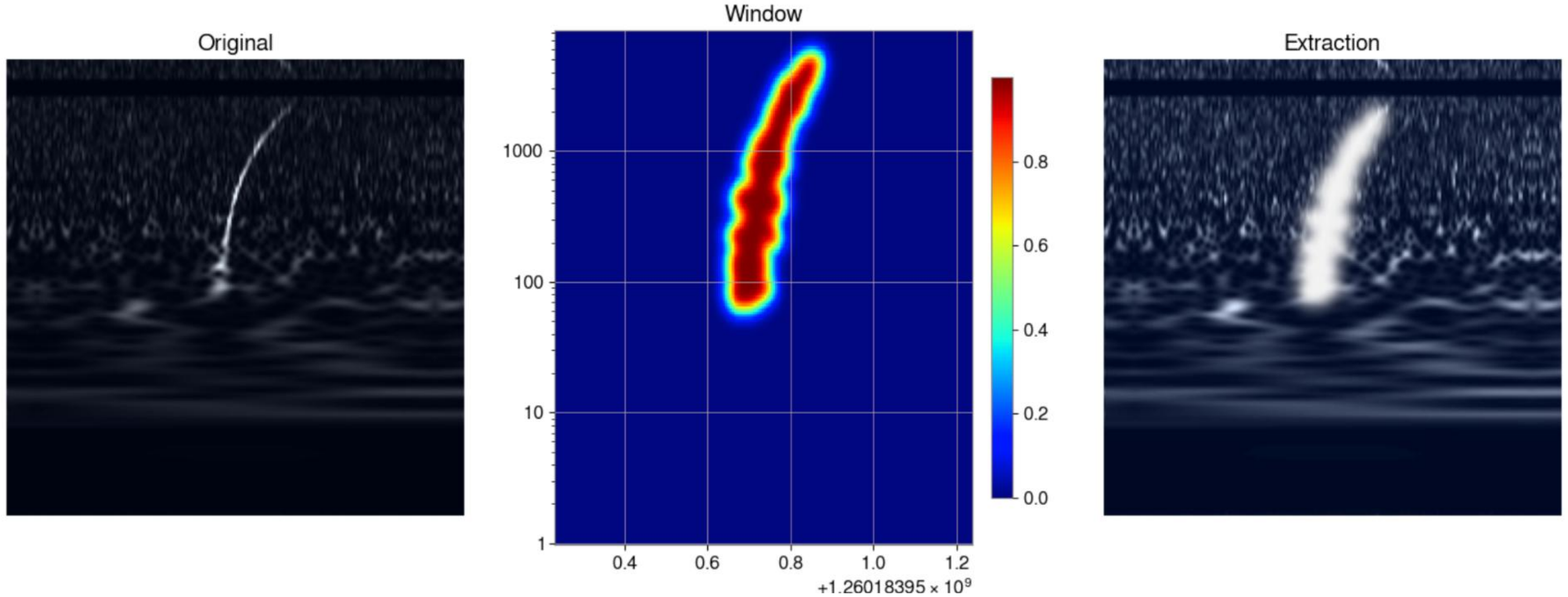


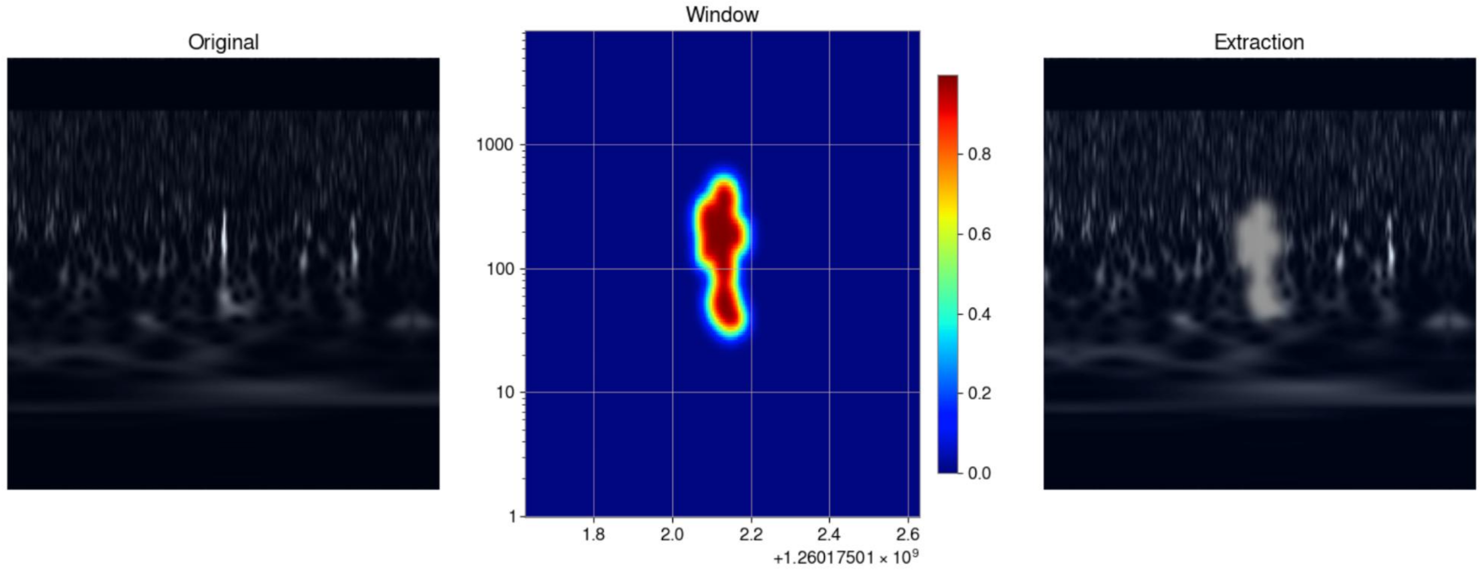


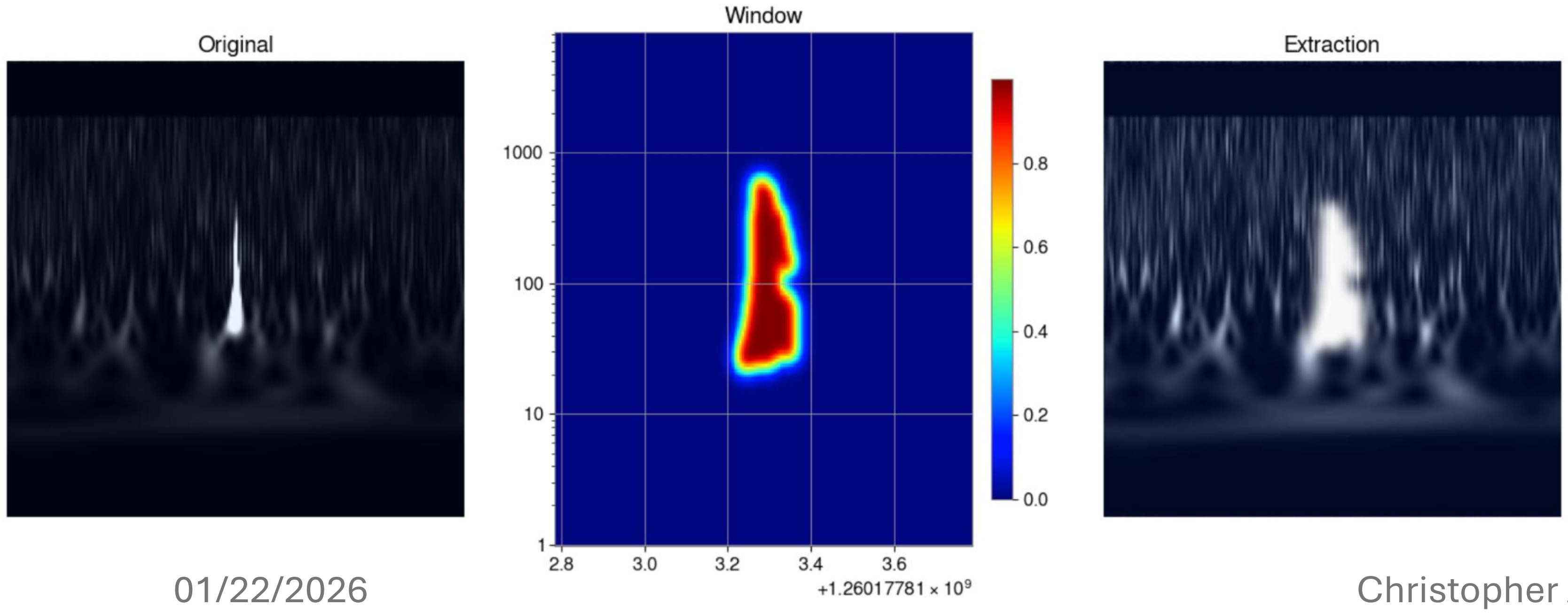




Example of masking

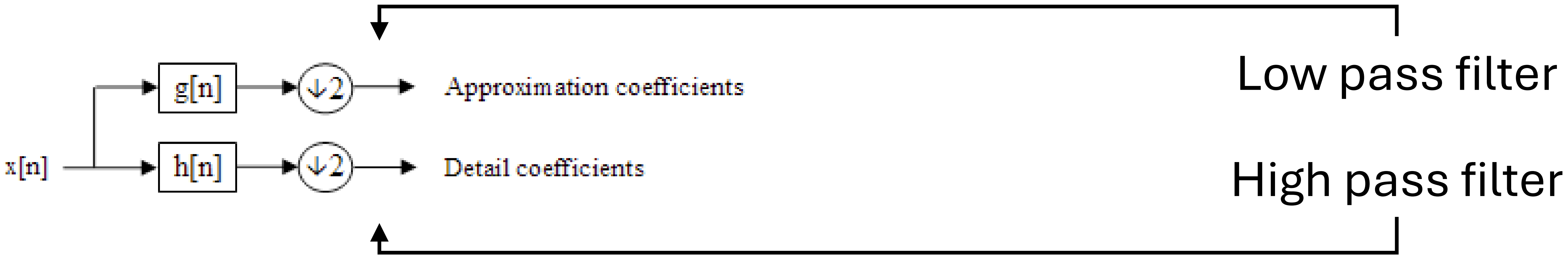




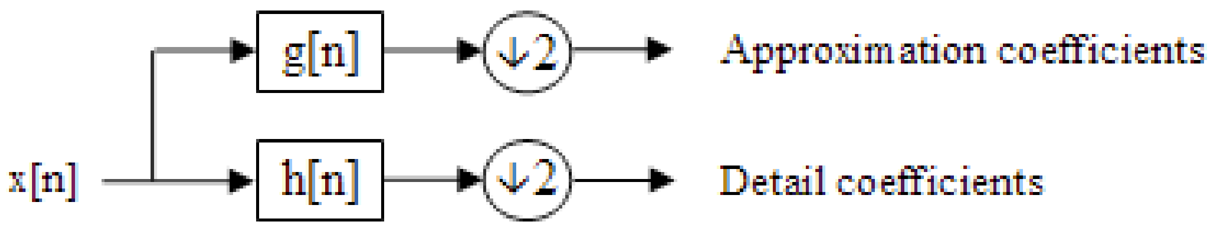


⇒ (b) Second glitch extraction

Discrete Wavelet Transform



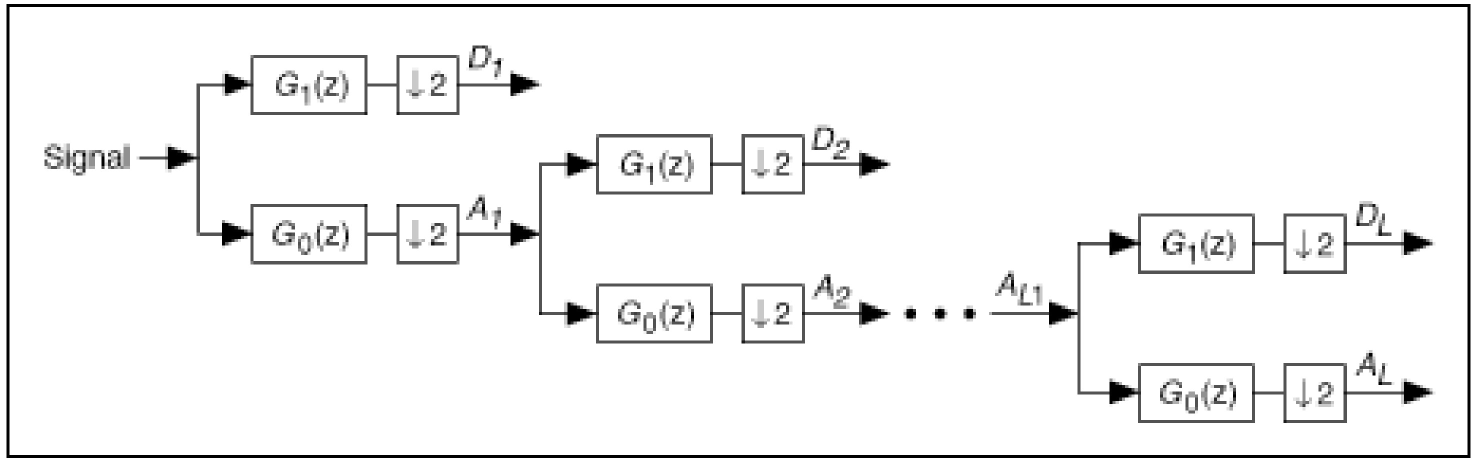
Discrete Wavelet Transform



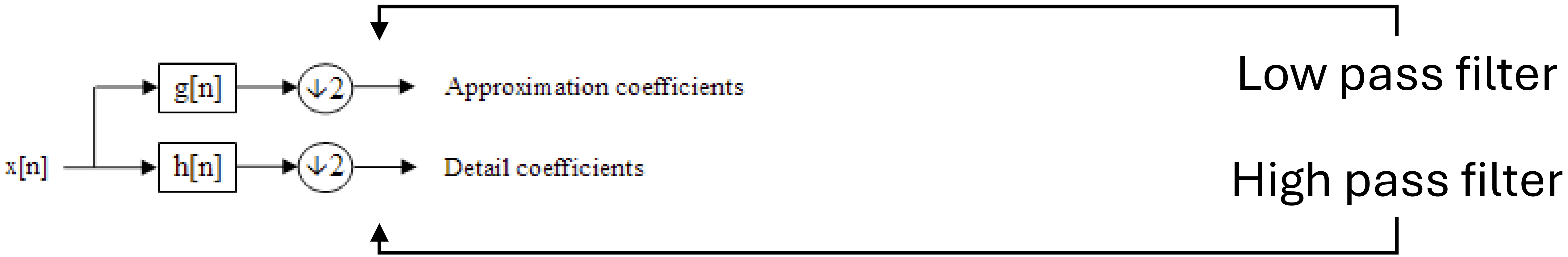
Low pass filter

High pass filter

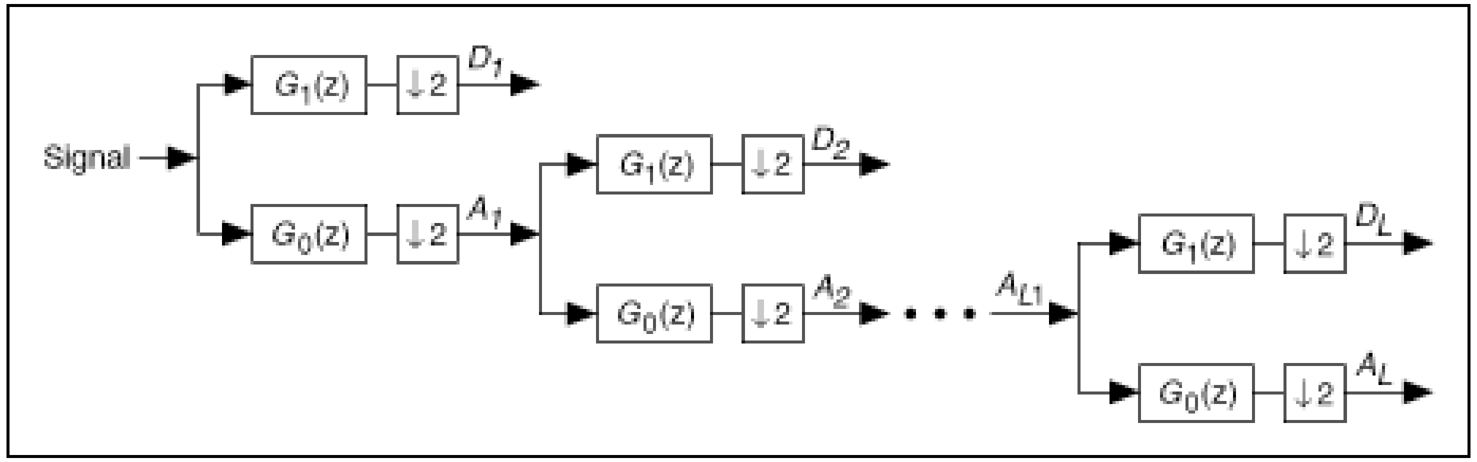
Transformation



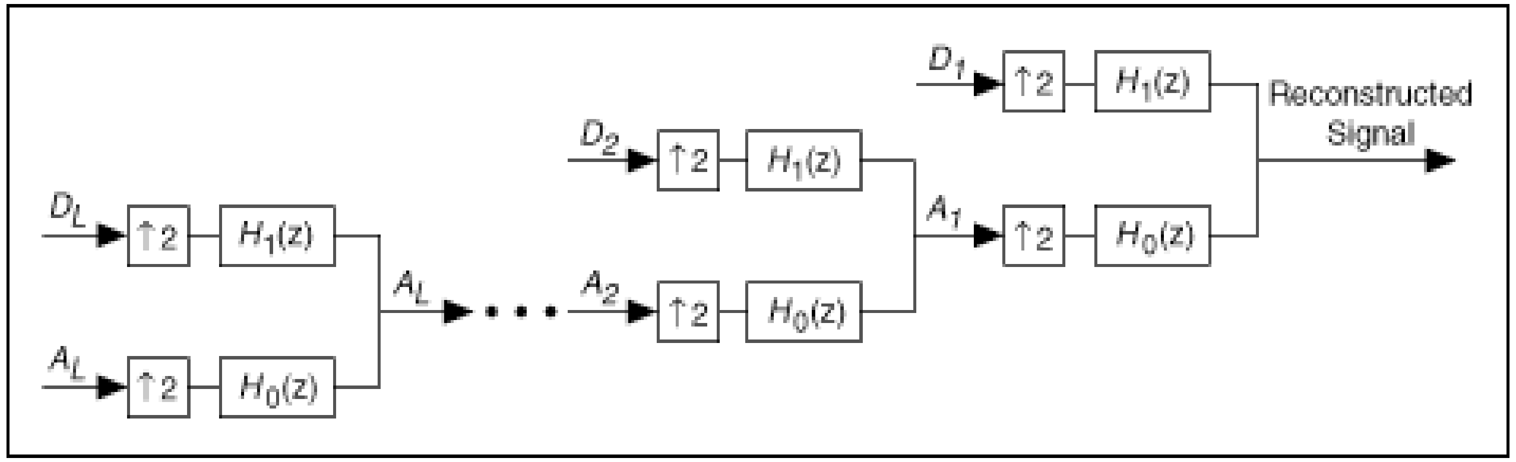
Discrete Wavelet Transform



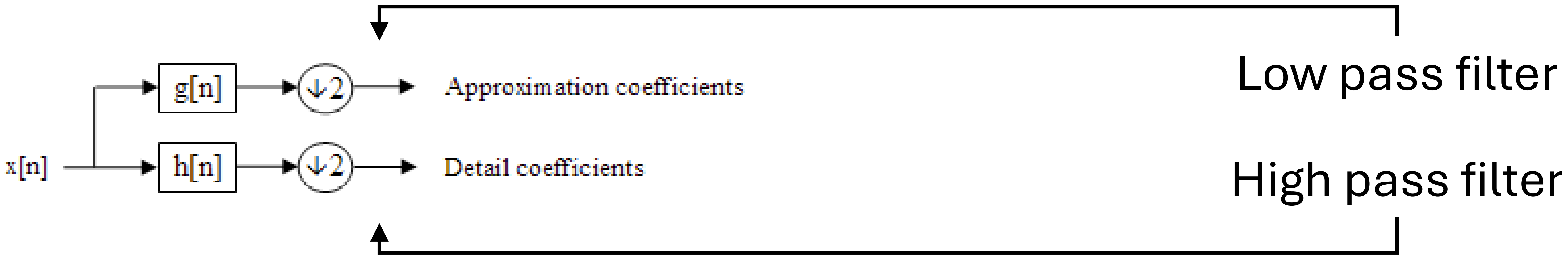
Transformation



Reconstruction

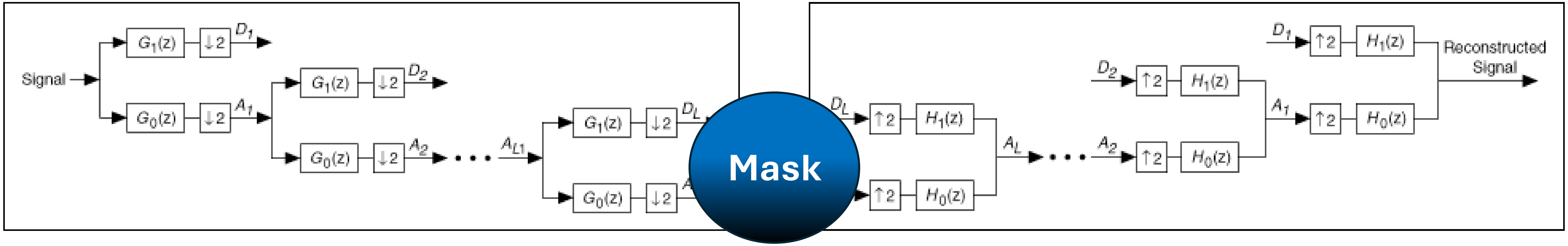


Discrete Wavelet Transform



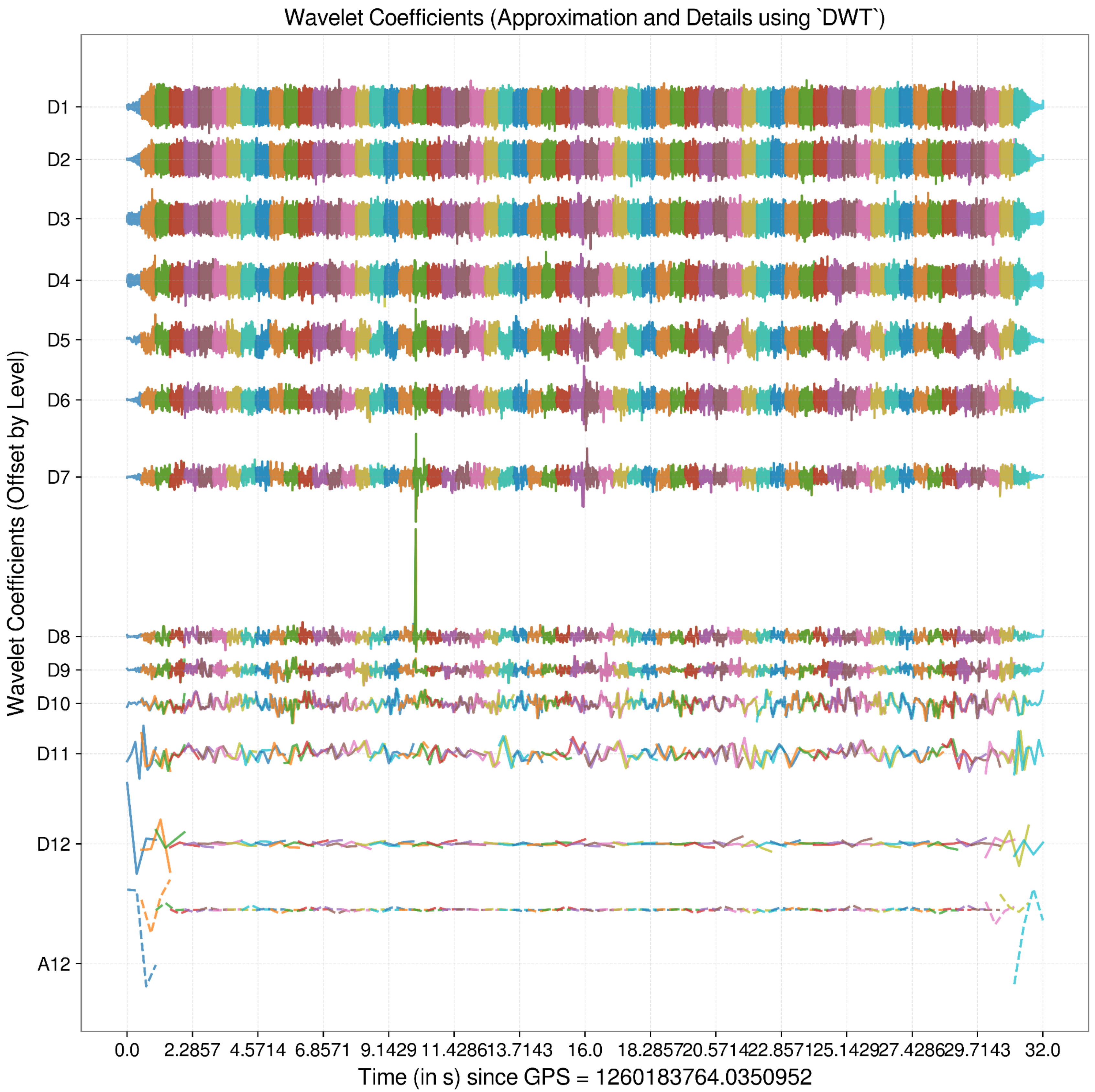
Transformation

Reconstruction



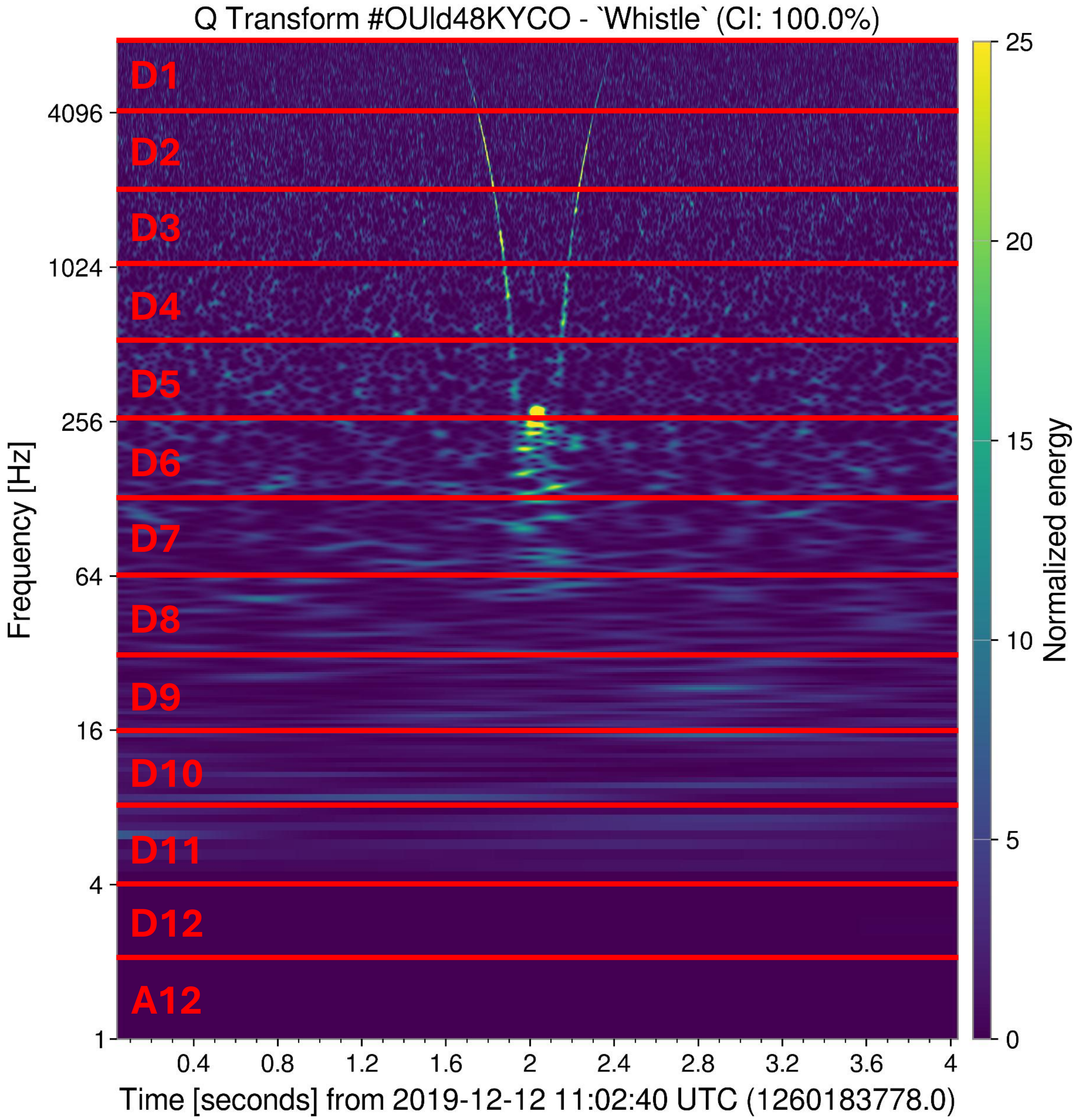
Masking the DWT of the signal

- Each details coefficients follows:
  - $D_n \in \left[\frac{f_s}{2^{n+1}}; \frac{f_s}{2^n}\right]$;
  - D_1 from 8192 to 4096;
  - D_2 from 4096 to 2048;
  - D_3 from 2048 to 1024;
  - …
- And the approximant :
  - $A_n \in \left[0; \frac{f_s}{2^{n+1}}\right]$;



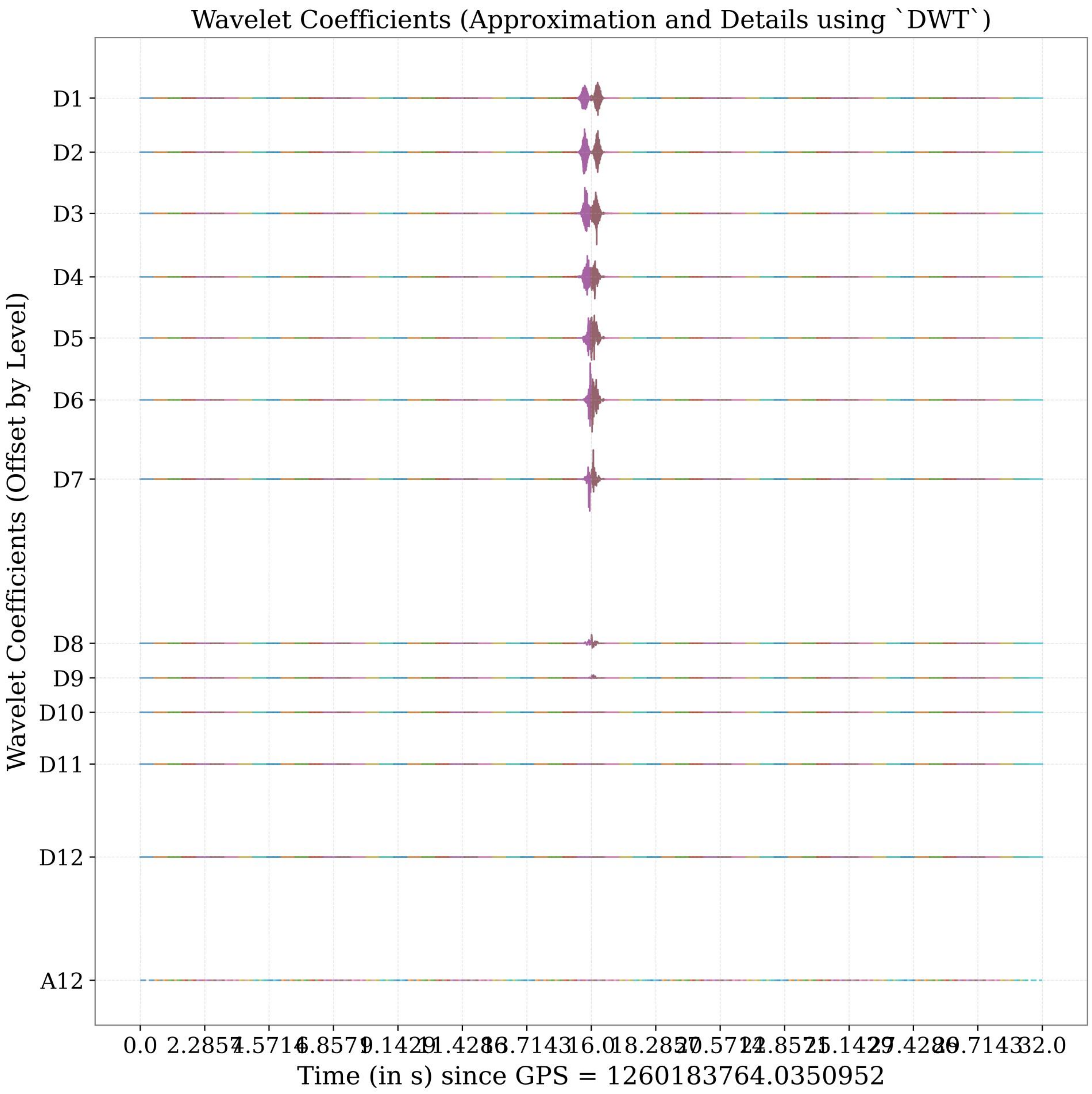
Masking the DWT of the signal

1. Split the Q-Transform space according DWT splitting;
2. Integrate the mask along frequencies;
3. Resample to fit the DWT sampling;
4. Multiply the DWT by the translate mask;
5. Transform back to a time series;



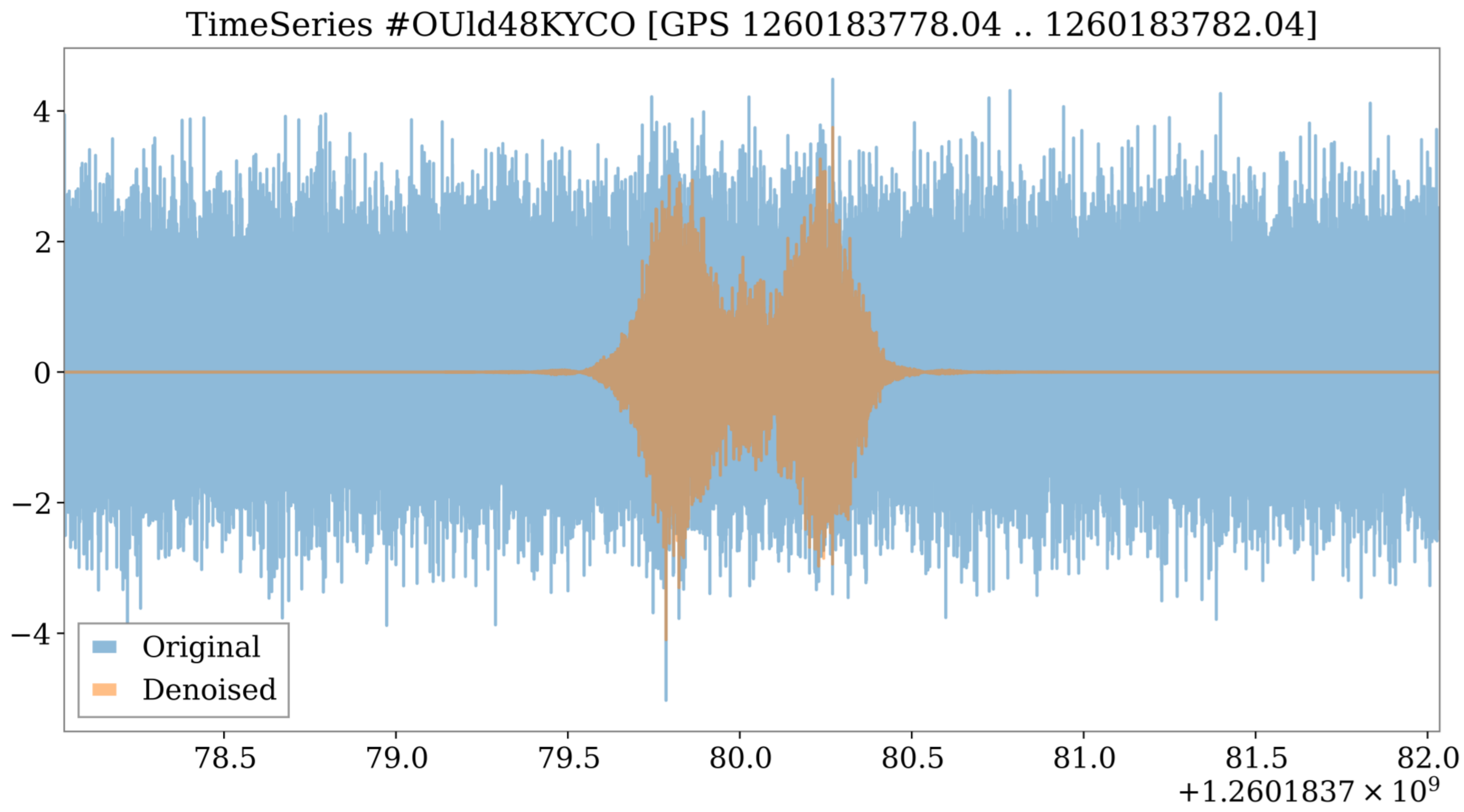
Masking the DWT of the signal

1. Split the Q-Transform space according DWT splitting;
2. Integrate the mask along frequencies;
3. Resample to fit the DWT sampling;
4. Multiply the DWT by the translate mask;
5. Transform back to a time series;



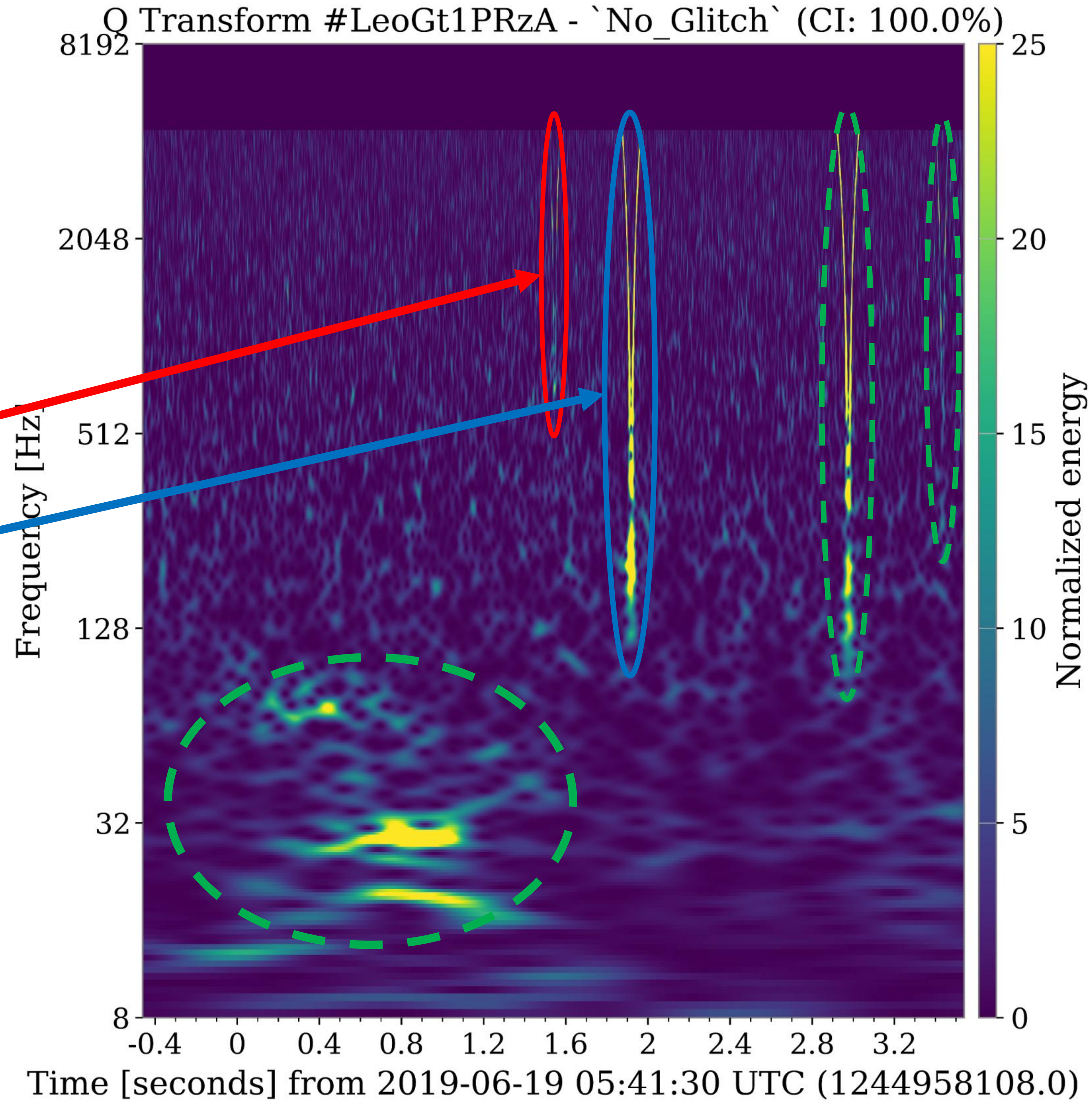
Masking the DWT of the signal

1. Split the Q-Transform space according DWT splitting;
2. Integrate the mask along frequencies;
3. Resample to fit the DWT sampling;
4. Multiply the DWT by the translate mask;
5. Transform back to a time series;



Complex Conditions Glitch

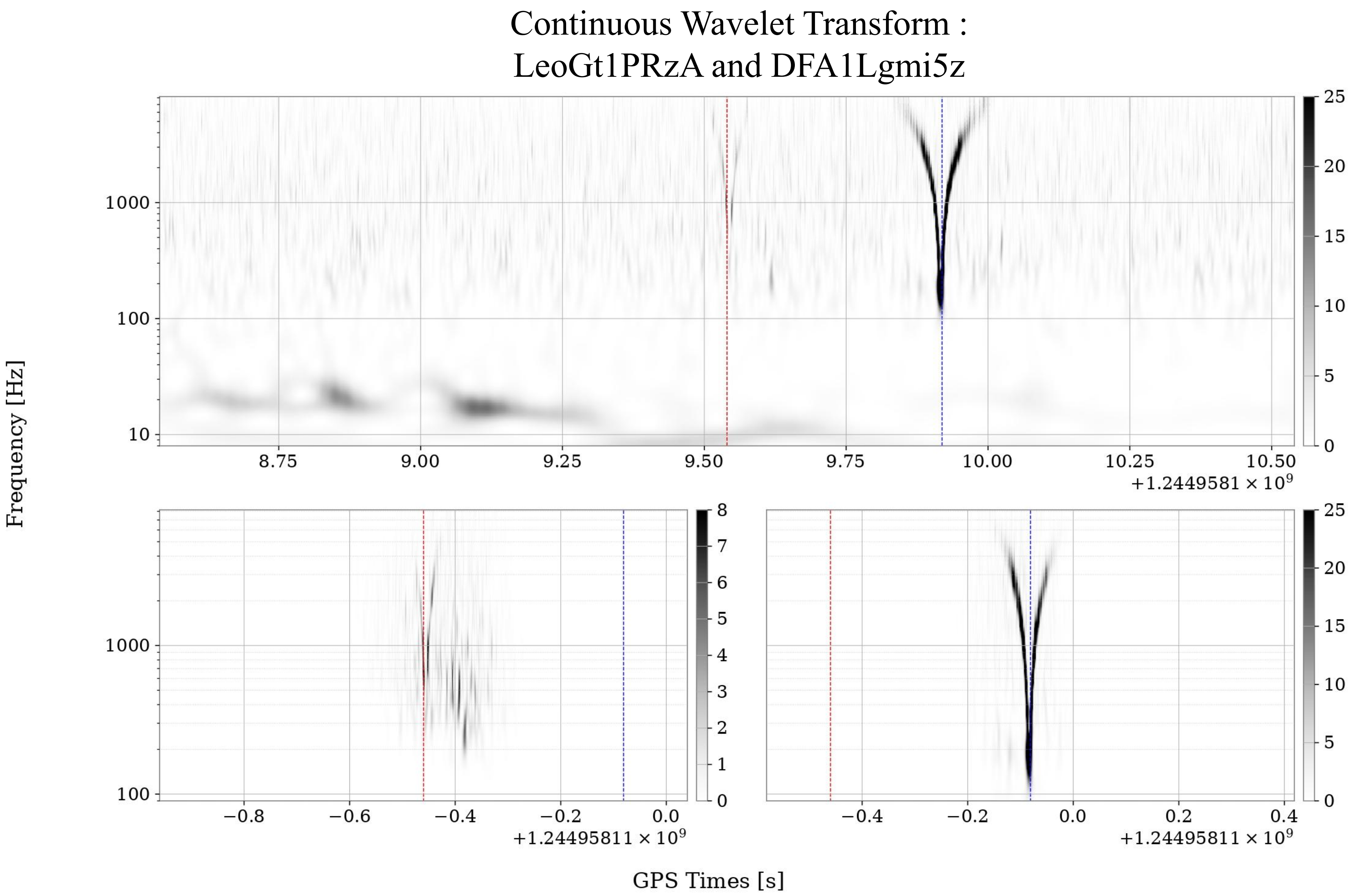
1. Multiples glitches:
   - LeoGt1PRzA
   - DFA1Lgmi5z
   - ...
2. Very fainted glitch



Complex Conditions
Glitch

1. Multiples glitches:
   - LeoGt1PRzA (red line)
   - DFA1Lgmi5z (blue line)
   - ...
2. Very Fainted glitch

⇒ (c) Glitch subtracted



Conclusion

- We have a complete framework to (a) classify, (b) segment and (c) subtract glitches from noise.

- The segmentation and subtractions apply on fainted glitch

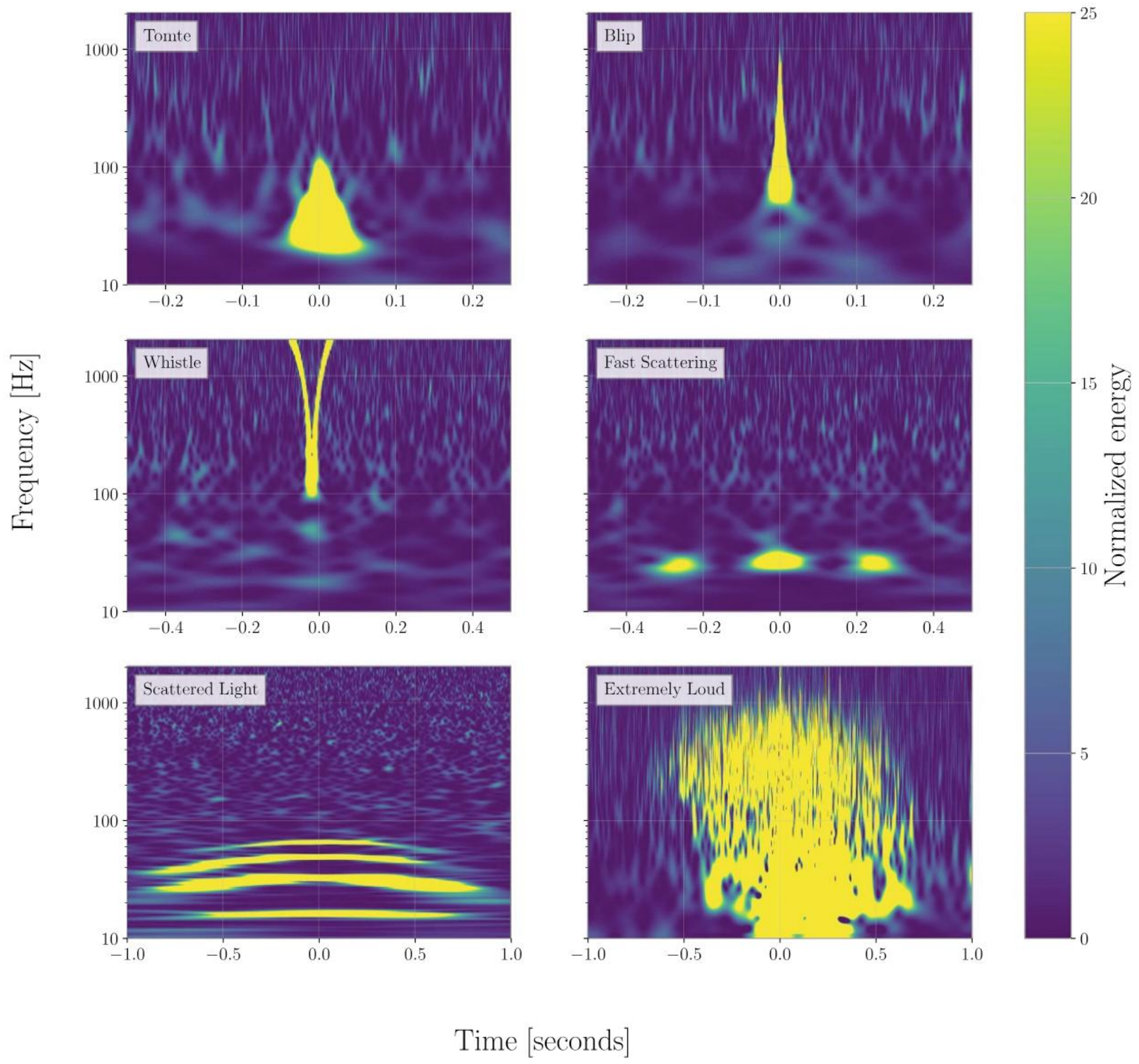
- A whole procedure to deal with multiple glitch in a narrow neighbouring allows to subtract them separately

Thanks for your attention

Glitch

GravitySpy classifies Glitches on events (Omicron triggers) among 26 classes.
Focus on :

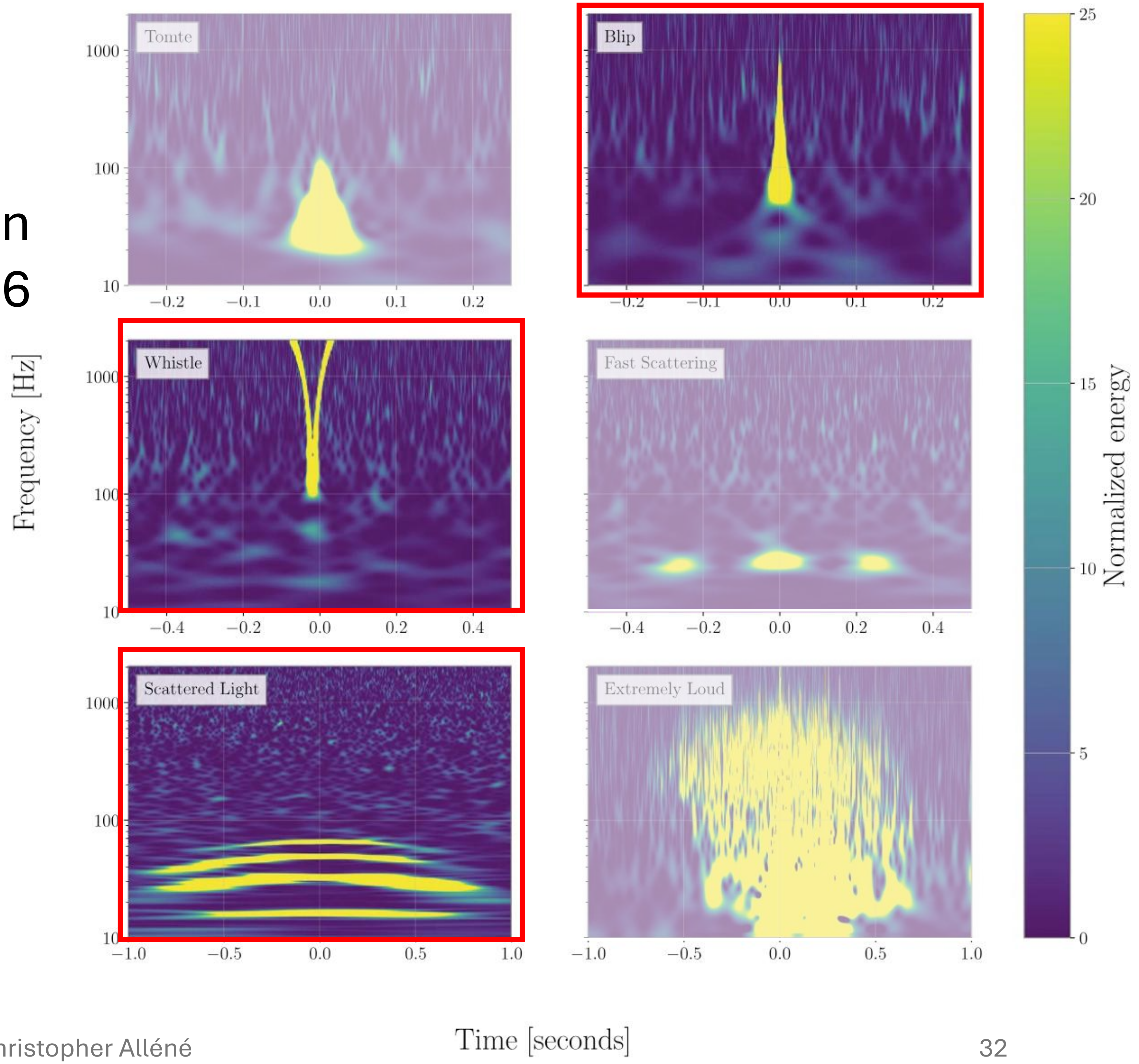
- Whistle
- Scattered Light
- Blip



Glitch

GravitySpy classifies Glitches on events (Omicron triggers) among 26 classes.
Focus on :

- Whistle
- Scattered Light
- Blip



Data augmentation