Illuminating Neural Networks: A Cycle of Explainable AI for Gravitational Waves

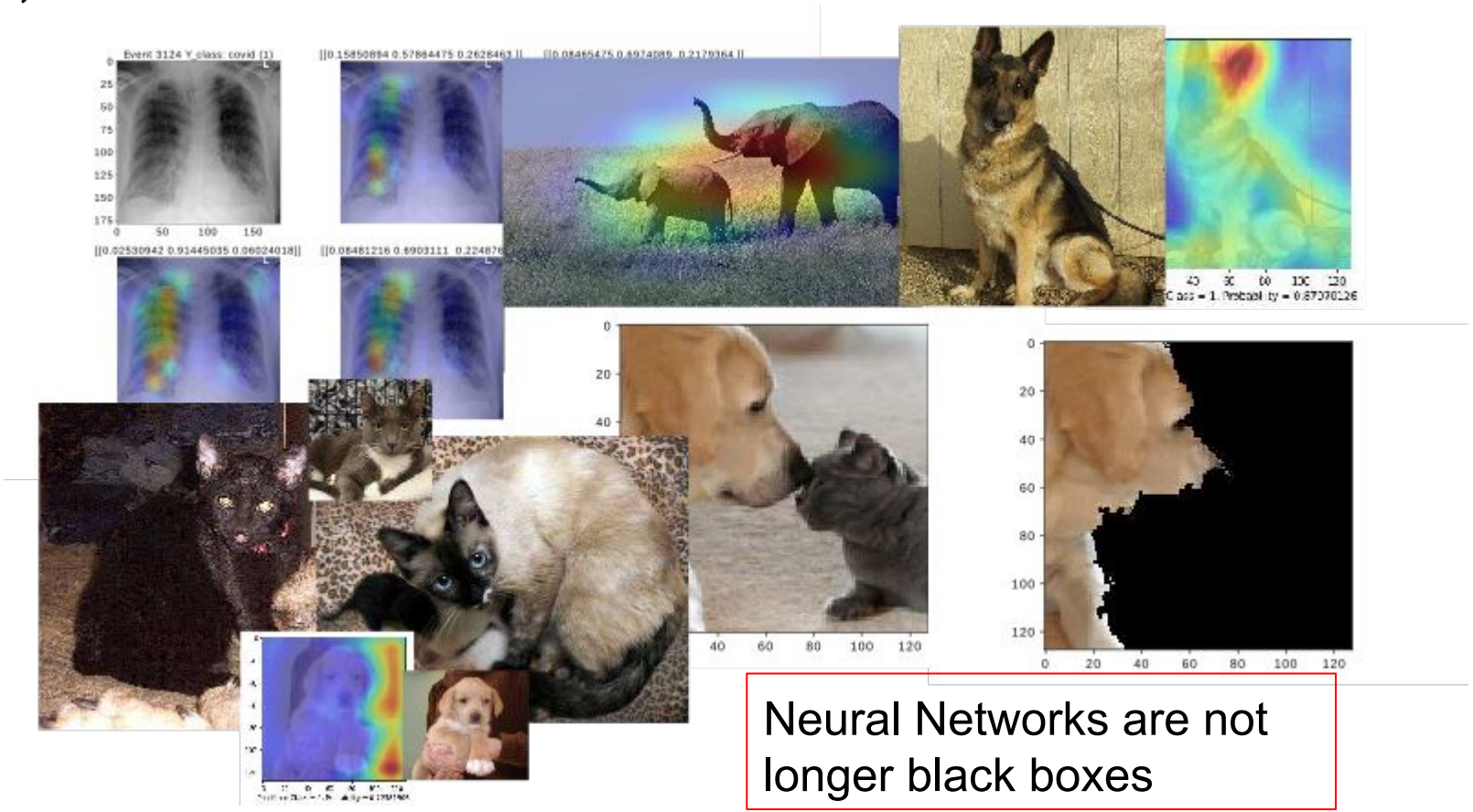
Miguel Cárdenas-Montes
CIEMAT (Spain)

Neural Networks

- **Neural Networks** has become an integral component of data analyses pipelines in contemporary experiments within Particle Physics and Astrophysics, including Gravitational Waves analysis.
- Their application emerges as a promising solution to address **the escalating volume, speed and complexity of experimental data**.

- Within this context, **Explainable Artificial Intelligence** (XAI) assumes a crucial role in deciphering the intricate decisions of Deep Learning, uncovering biases, and facilitating iterative improvements.
- **XAI** algorithms serve as **interpreters**, offering researchers and practitioners a **glimpse into the mechanisms guiding the models' decisions for later improvements**.
- Not more deeper and deeper NN for improving the performance!

XAI, intuition



Neural Networks are not longer black boxes

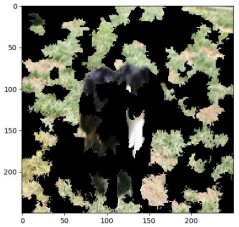
XAI, intuition and biased learning

Dog correctly classified as dog (class 1): [4.8e-11 (cat), 1.0e+00 (dog)]

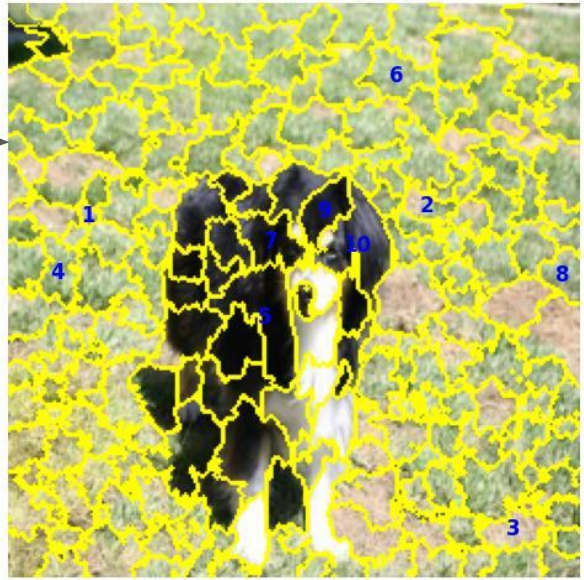
The 10 most important patches are mostly grass.

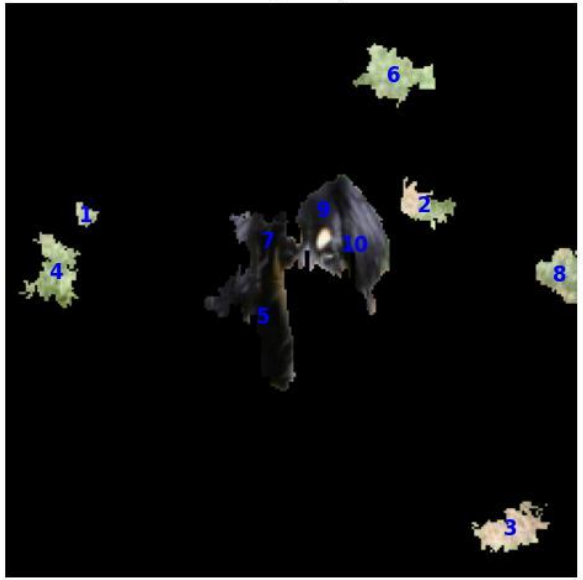
Image segmented in homogeneous patches (250).

Randomly patches are switched on/off and then predicted.



Degradation of prediction illuminates the importance of the patches. And repeat …

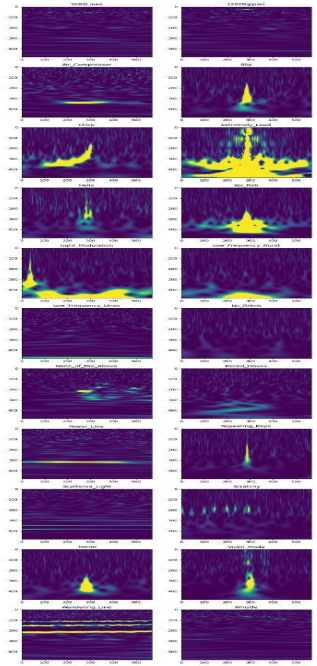




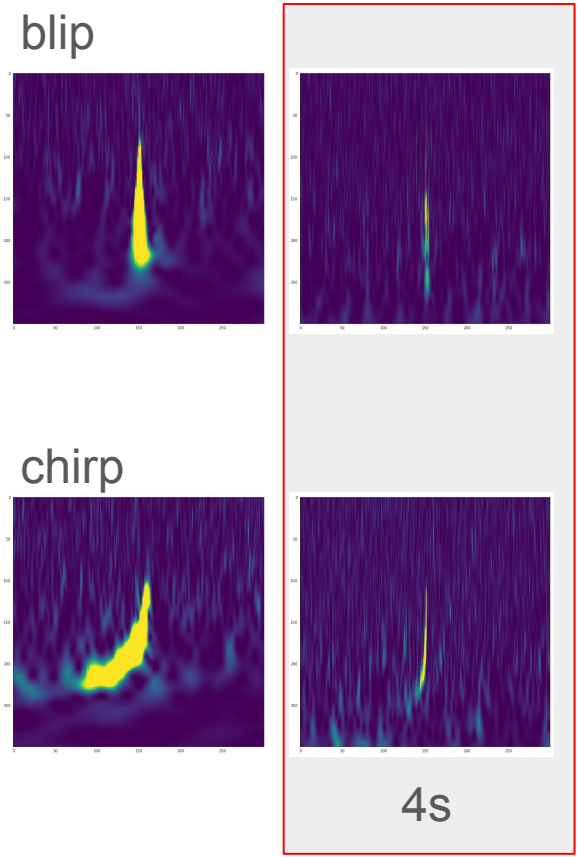
Linear regressor with patches on/off as input and the performance degradation as output.

Gravity Spy (Kaggle)

- time-frequency images (Q-transform),
- 22 labels (chirp + 21 glitches),
- 31.9k files: train, test, validation;
- 4 images per event, time window of 0.5, 1, 2, and 4 seconds,
- strongly unbalanced labels,
- public data set.



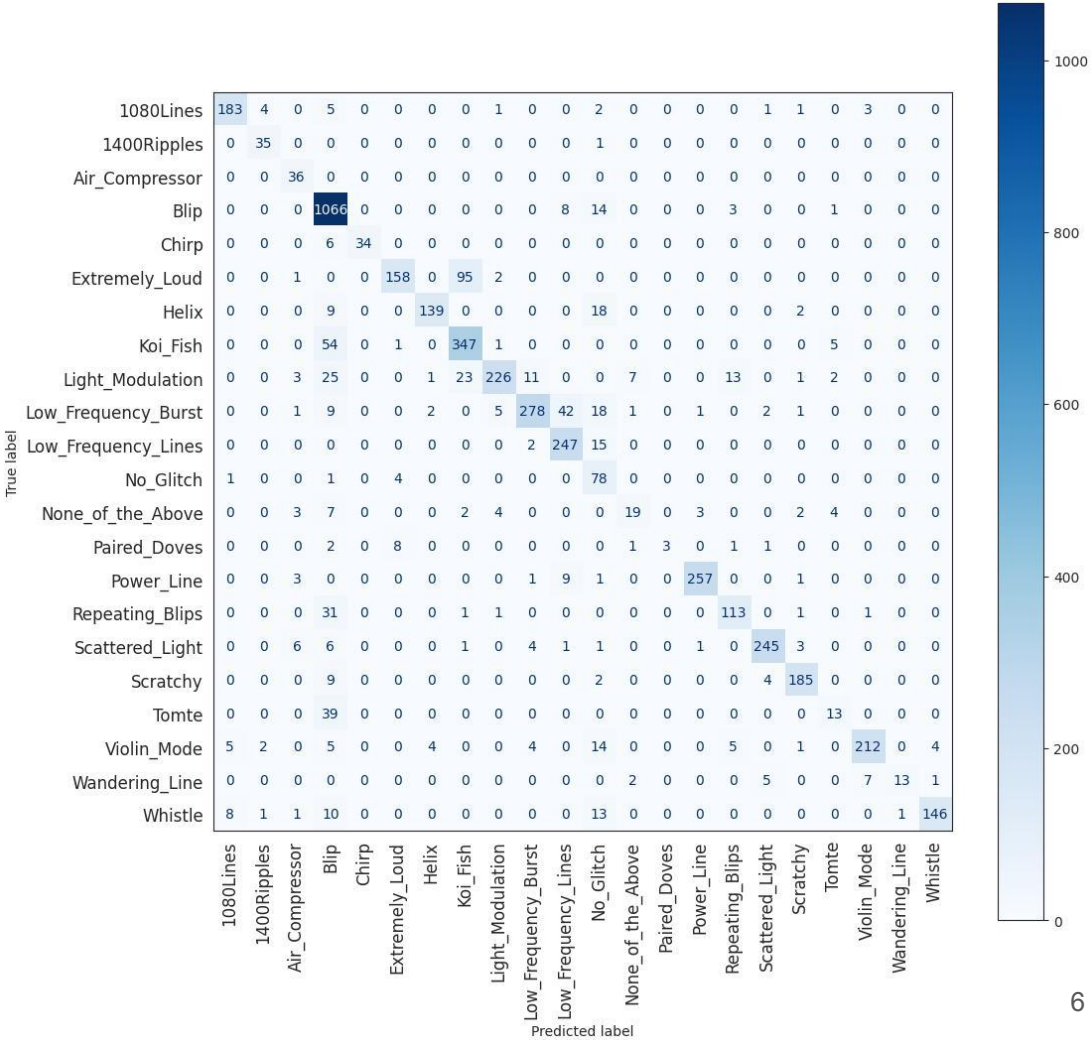
- Ideal for testing ideas.
- Images ideal for Convolutional NN (CNN).
- Classification per image or per event?



Gravity Spy (Kaggle)

- (2 or 3)X(Conv2D+MaxPooling) + output layer (from 6 to 2 [irreducible] chirp errors).
- Trainable parameters: 1M-400k
- ~190 per epoch
- EarlyStopping patience=3

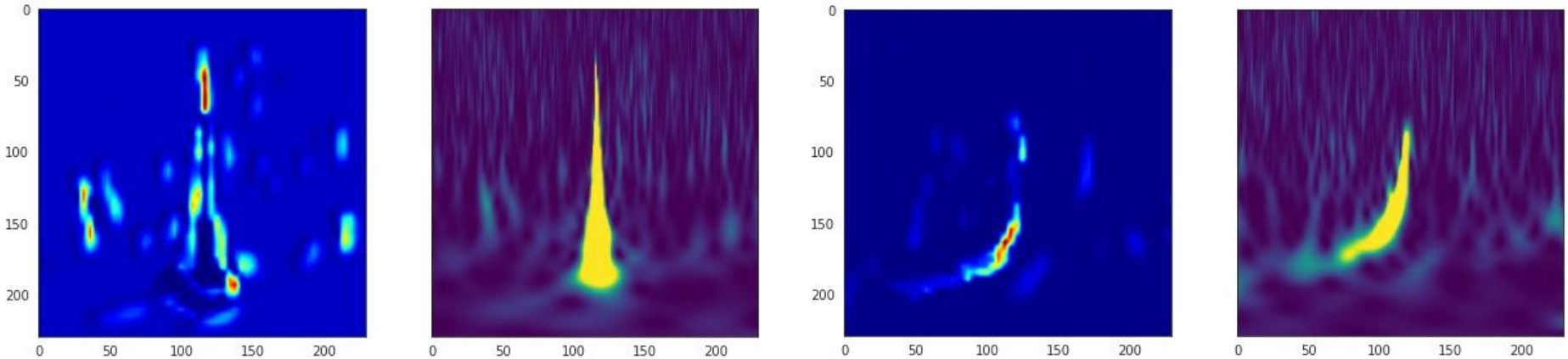
- Most of the chirp errors go to blip label.
- All the chirp errors are for 4 s time window.



XAI (GradCAM), correctly classified events

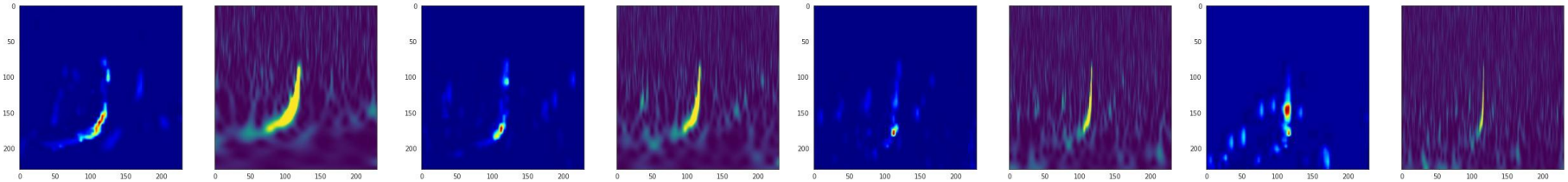
Blip

Chirp



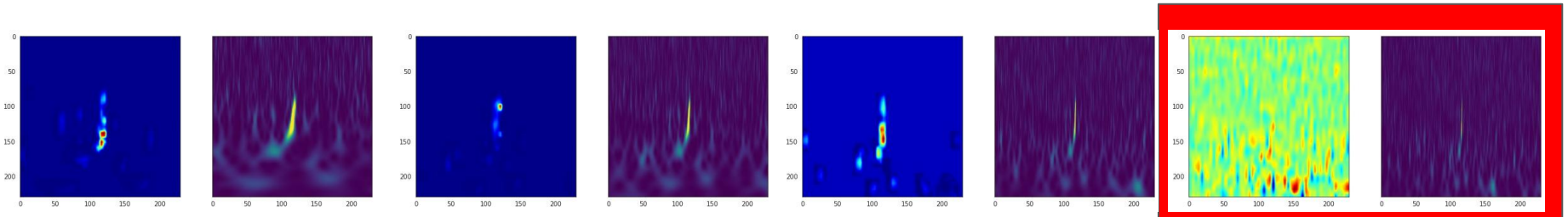
Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization https://doi.org/10.48550/arXiv.1610.02391

XAI (GradCAM), correctly classified events



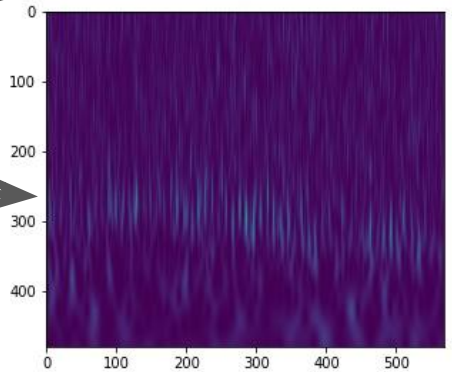
Chirp event (correctly classified) from 0.5 s to 4 s.
As the window is larger, the image is more stretched vertically.
Horizontal part of the chirp becomes less relevant, and more prone to misclassification.

XAI (GradCAM), incorrectly classified events

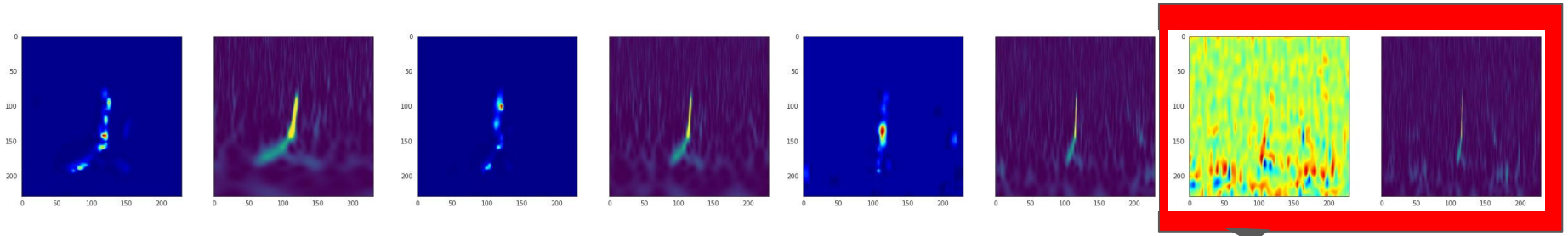


Chirp (correctly (mis)classified) from 0.5 s to 4 s.
For 4s event misclassified as “scratchy”.

How modify the input for avoiding this misclassification?



XAI (GradCAM), incorrectly classified events



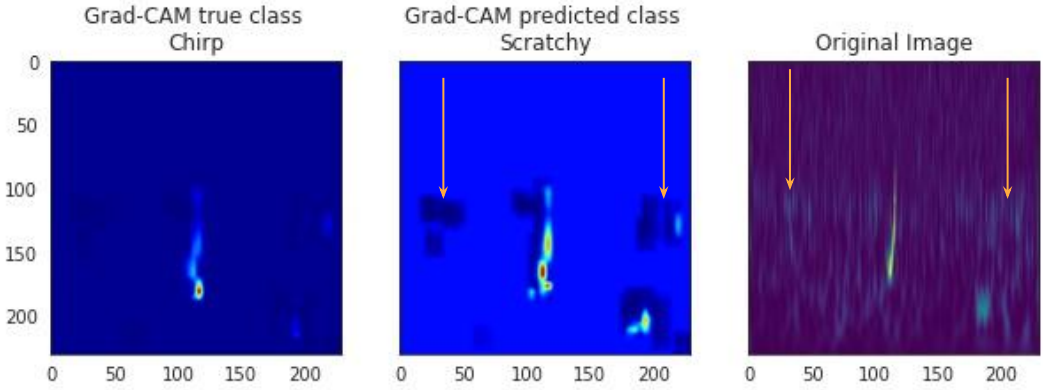
Chirp (correctly (mis)classified) from 0.5 s to 4 s.
For 4s event misclassified as “blip”.

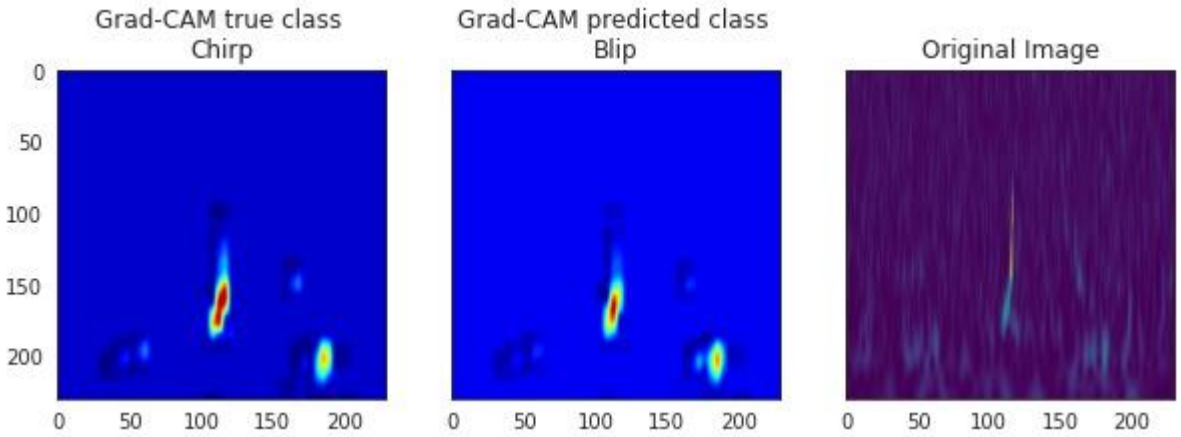
How modify the input for avoiding this misclassification?

New ideas for improving the classification.

XAI (GradCAM), incorrectly classified events

How to modify the input for remove confusing features (pixels)?
XAI teaches us how.





XAI (LIME), correctly classified events

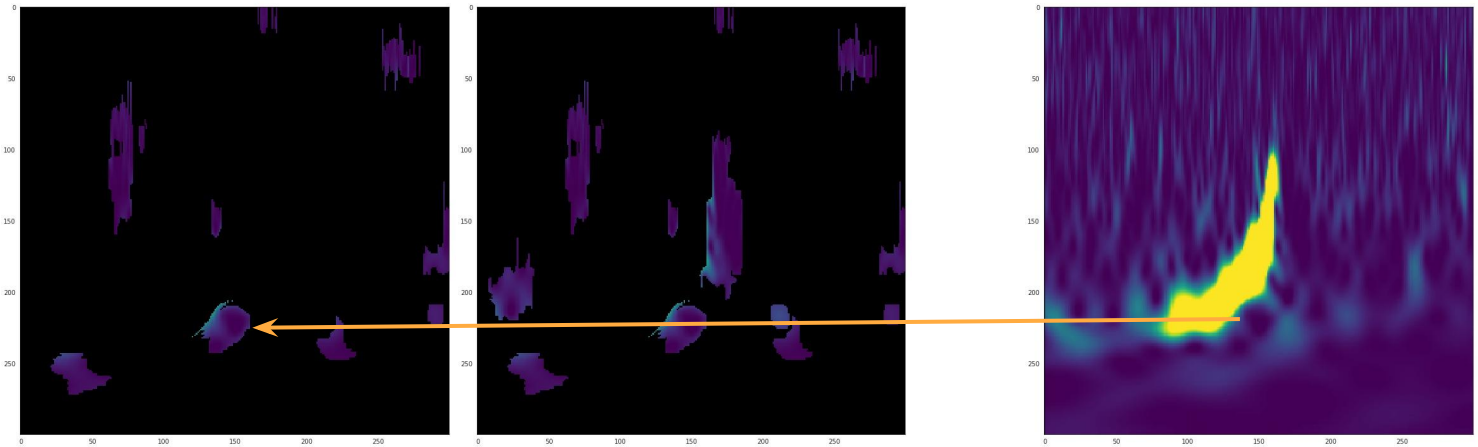
Most relevant patches for correctly classify.

In chirps, ramp-up in frequency is relevant (asymmetric horizontal).

5 parches

10 parches

Chirp



XAI (LIME), correctly classified events

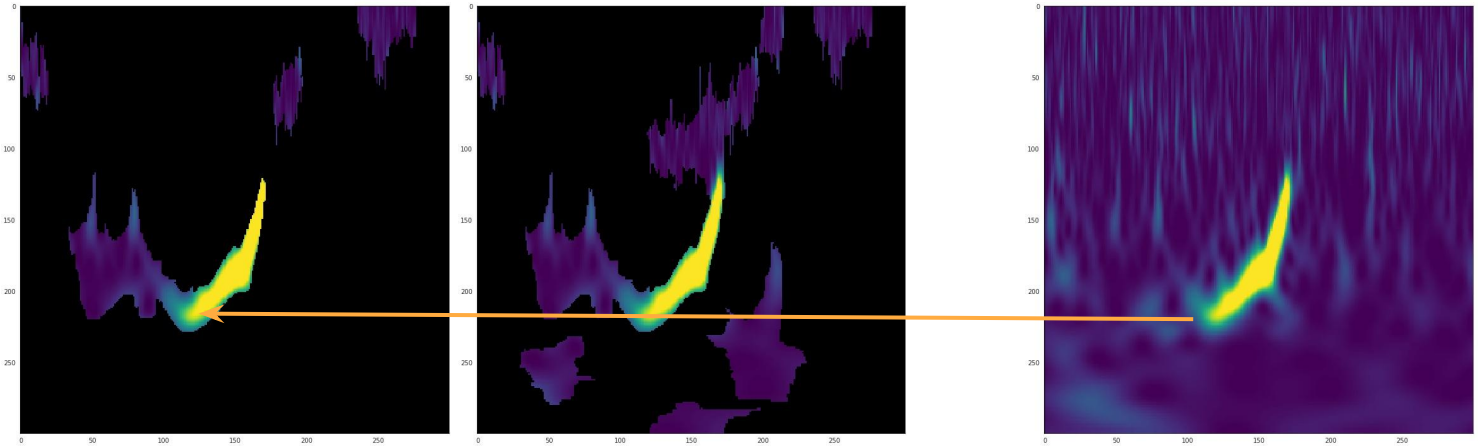
Most relevant patches for correctly classify.

In chirps, ramp-up in frequency is relevant (asymmetric horizontal).

5 parches

10 parches

Chirp



"Why Should I Trust You?": Explaining the Predictions of Any Classifier: https://doi.org/10.48550/arXiv.1602.04938

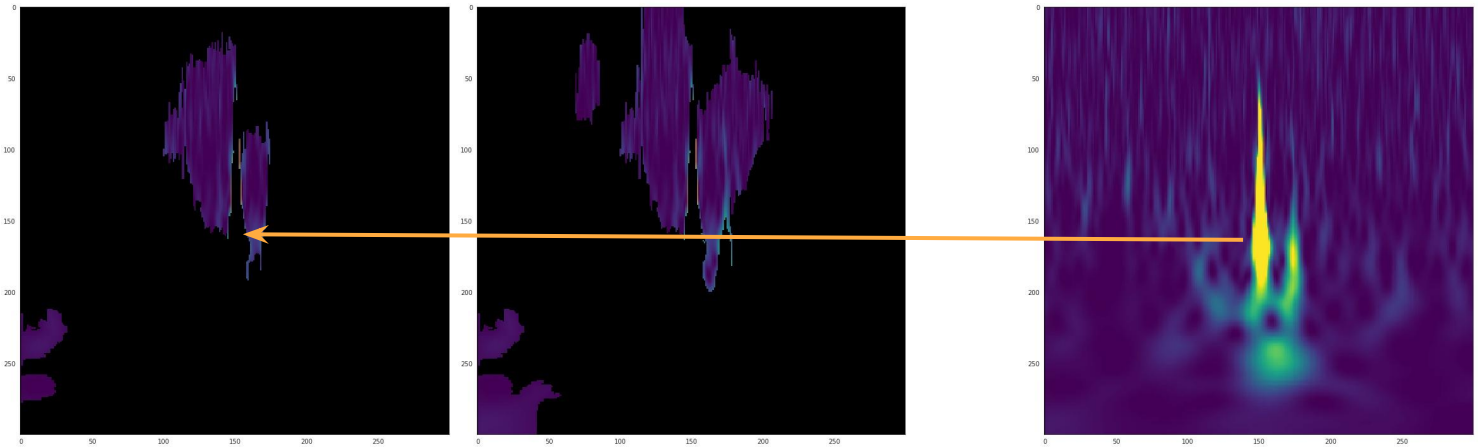
XAI (LIME), correctly classified events

In blips, vertical-lateral to stronger part of the signal is relevant.

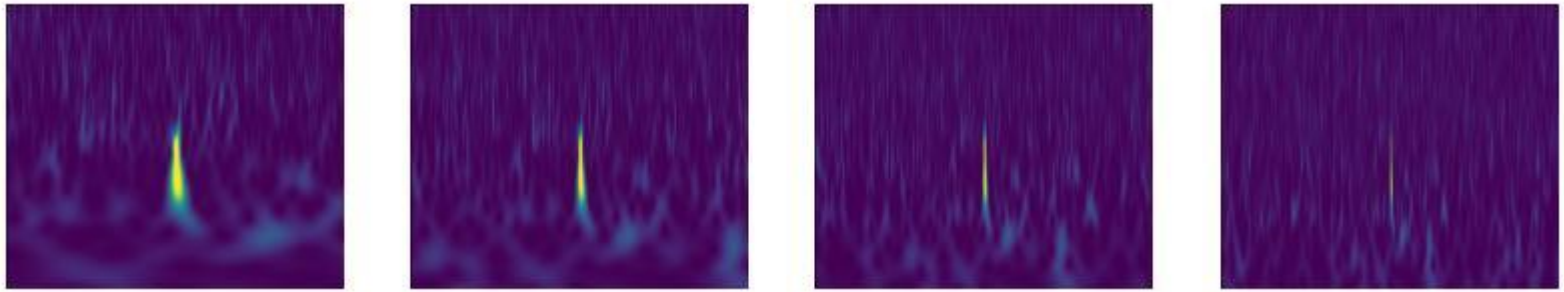
5 parches

10 parches

Blip



Events as sequences of images (time windows)



Convert 4 images into a sequence (event) -> event classifier

input shape = (4,479,569,3)

model.add(layers.Conv3D(8, kernel_size=(1, 3, 3), activation='relu'))

instead of:

model.add(layers.Conv2D(filters=128, kernel_size=(3,3), activation = 'relu'))

Input: multitime tensor data

- No errors on chirps or 1 single error.

- 2D
  - 3 errors
  - 190 s /epoch, ~24 epochs, patience 3
  - 1M-400k trainable parameters.
- 3D
  - 0-2 errors (depends on 2D or 3D SpaAtt)
  - 15 s /epoch, ~4 epochs, patience 3
  - ~ 150k par. Lower carbon footprint
- Spatial Attention layer for interpretability.

Spatial Attention

```
class SpatialAttention(Layer):
        .
        .
        .
        self.kernel = self.add_weight(name='kernel',
                                      shape = (1,1,1,input_shape[-1],1),
                                      initializer='uniform',
                                      trainable=True)class SpatialAttention(Layer):
        .
        .
        .
        self.kernel = self.add_weight(name='kernel',
                                      shape = (1,1,1,input_shape[-1],1),
                                      initializer='uniform',
                                      trainable=True)
        .
        .
        .
        attention = tf.nn.sigmoid(tf.nn.conv3d(x, self.kernel,
strides=[1,1,1,1,1], padding='SAME'))
        return x * attention
```

Spatial attention 3D

```
inputs = keras.Input(shape=sample_shape)

# smartly remove this segment if not spatial attention
SpaAtt = SpatialAttention()(inputs)

conv1 = layers.Conv3D(8, kernel_size=(1, 3, 3), activation='relu')(SpaAtt)
maxpool1 = layers.MaxPooling3D(pool_size=(1, 3, 3))(conv1)
```

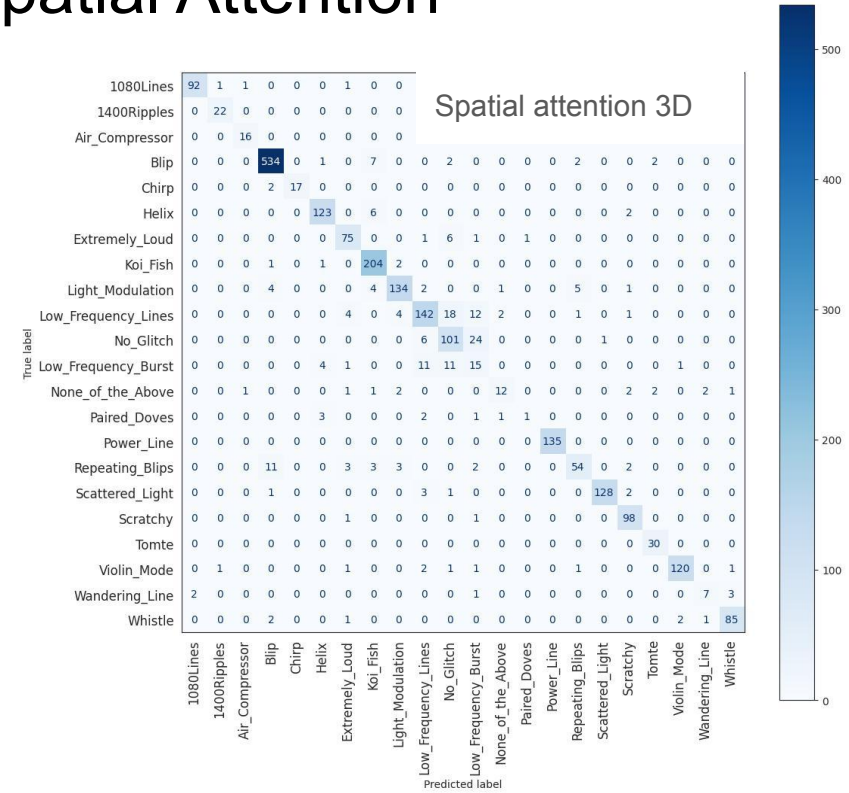
```
self.kernel = self.add_weight(name='kernel',
                              shape = (1,1,input_shape[-1],1),
                              initializer='uniform',
                              trainable=True)cla
```

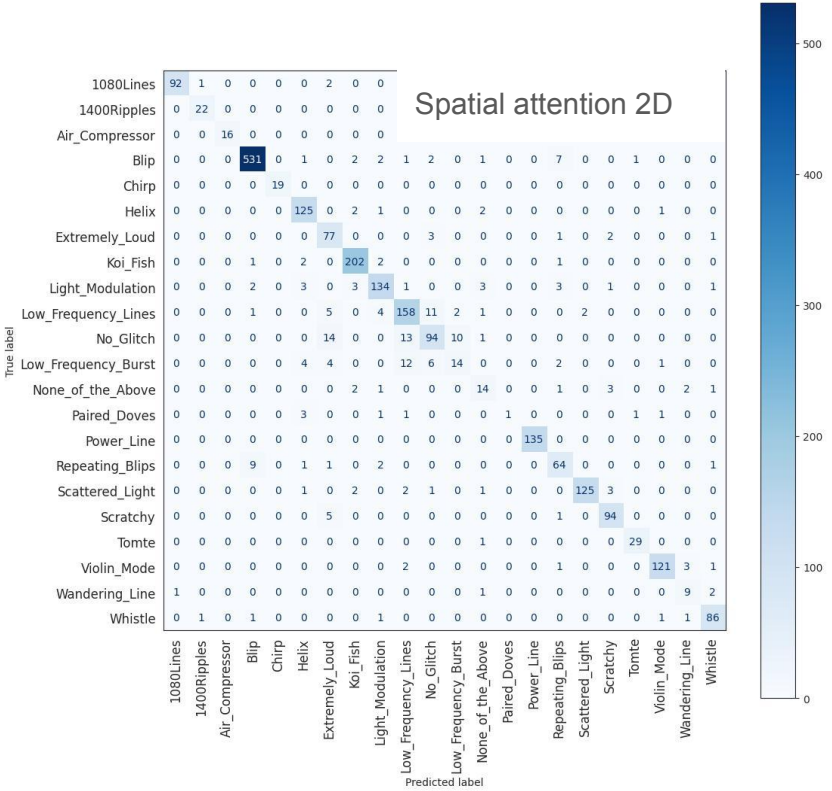
Spatial attention 2D

Spatial attention per time layer.

It learns to amplify the pixels of interest, minimizing others.

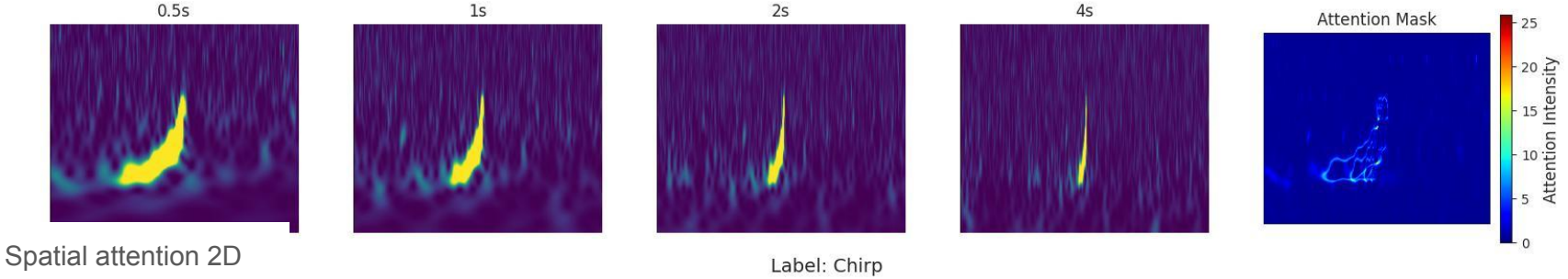
Spatial Attention





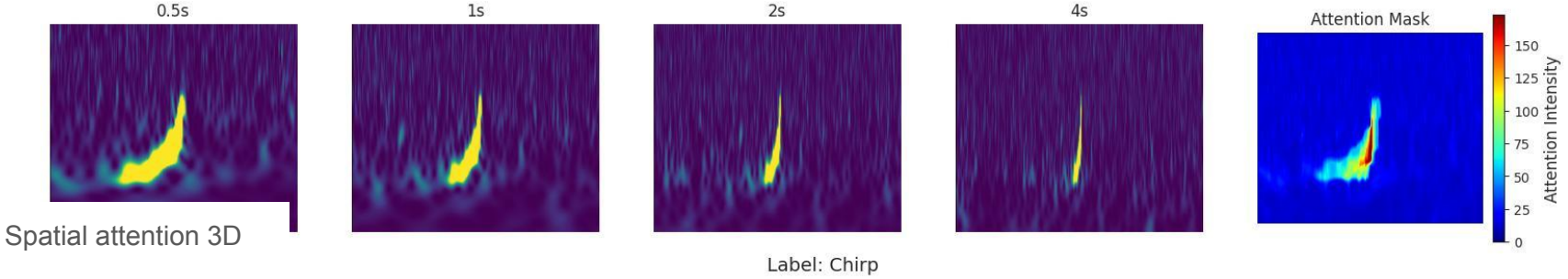
Spatial Attention 2D vs. 3D

Preliminary results



Spatial attention 2D

Label: Chirp

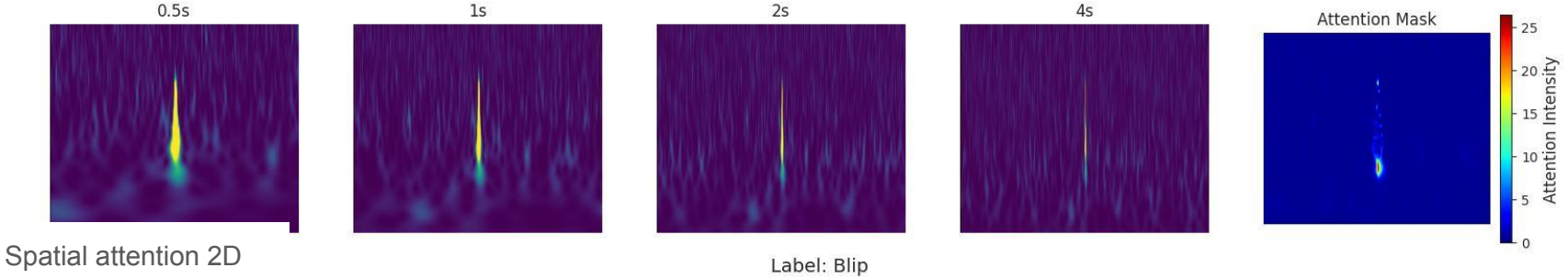


Spatial attention 3D

Label: Chirp

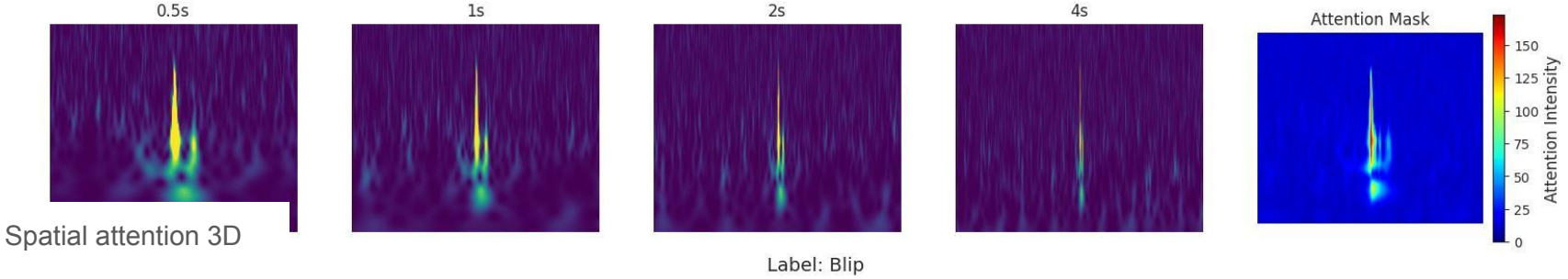
Spatial Attention 2D vs. 3D

Preliminary results



Spatial attention 2D

Label: Blip

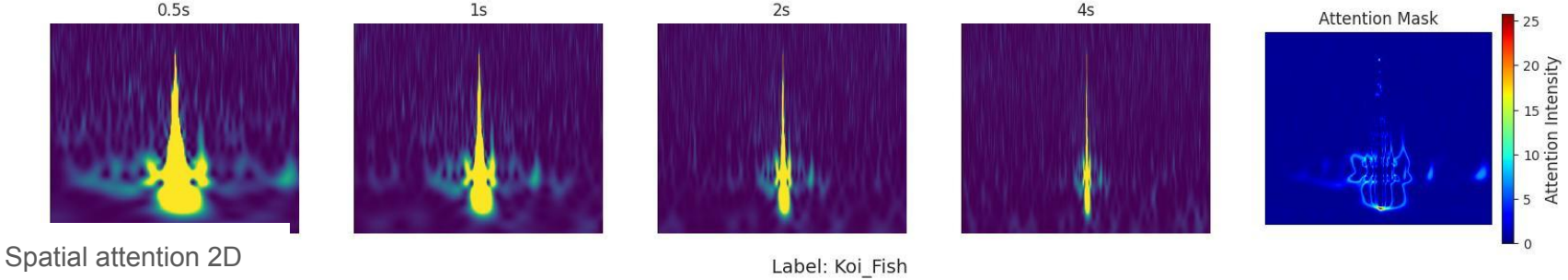


Spatial attention 3D

Label: Blip

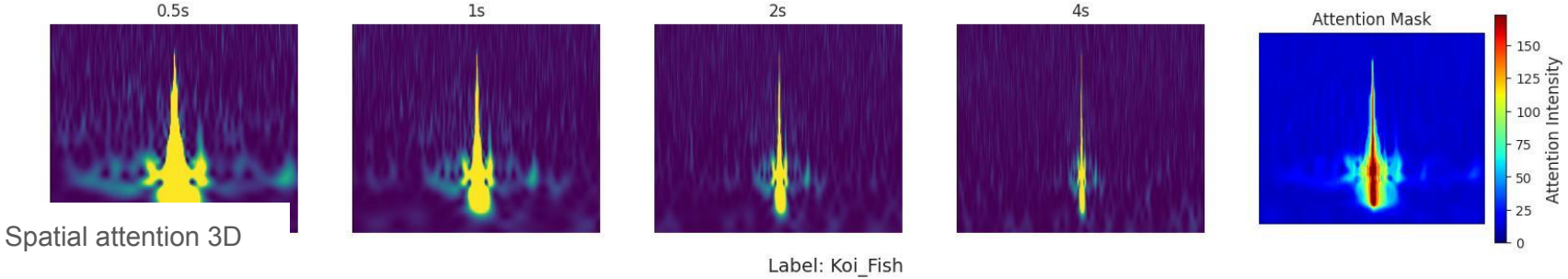
Spatial Attention 2D vs. 3D

Preliminary results



Spatial attention 2D

Label: Koi_Fish

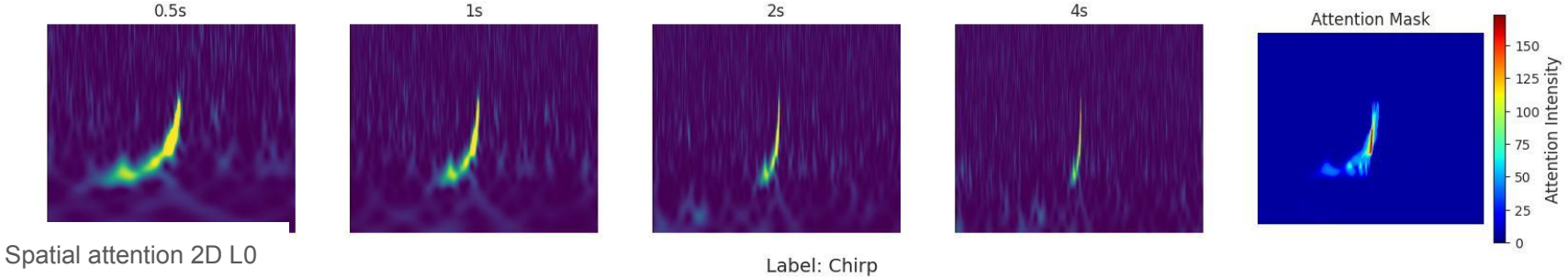


Spatial attention 3D

Label: Koi_Fish

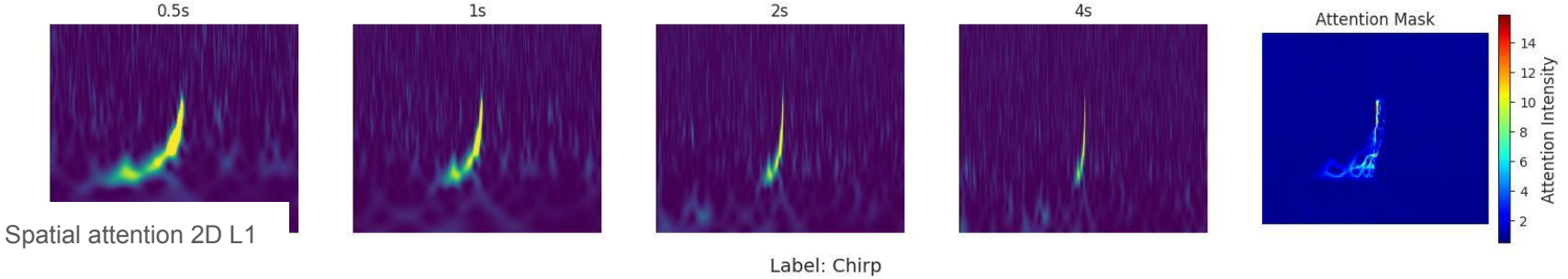
Spatial Attention in the two first layers

Preliminary results



Spatial attention 2D L0

Label: Chirp

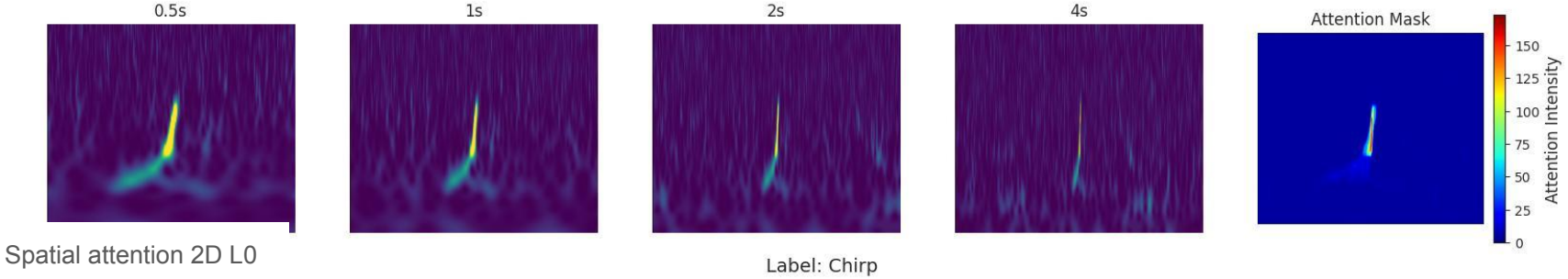


Spatial attention 2D L1

Label: Chirp

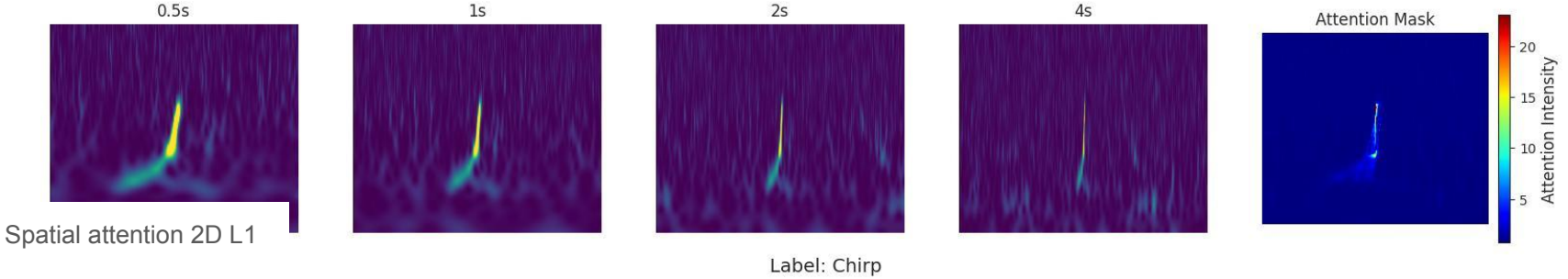
Spatial Attention in the two first layers

Preliminary results



Spatial attention 2D L0

Label: Chirp

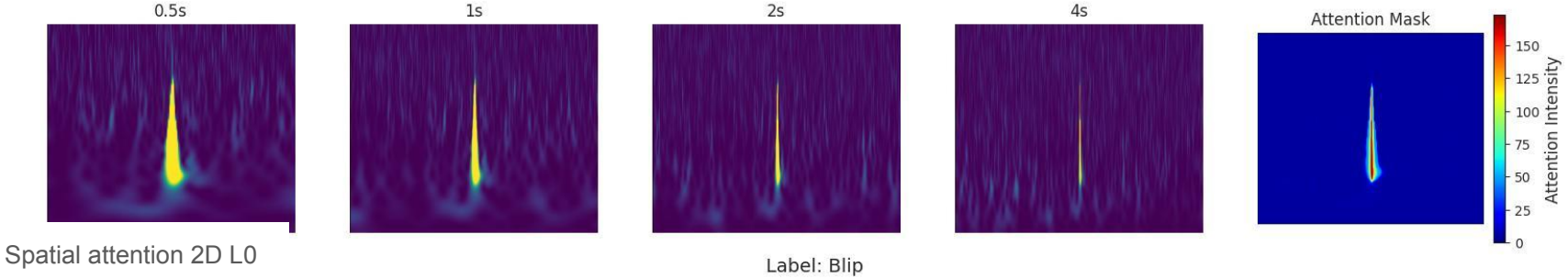


Spatial attention 2D L1

Label: Chirp

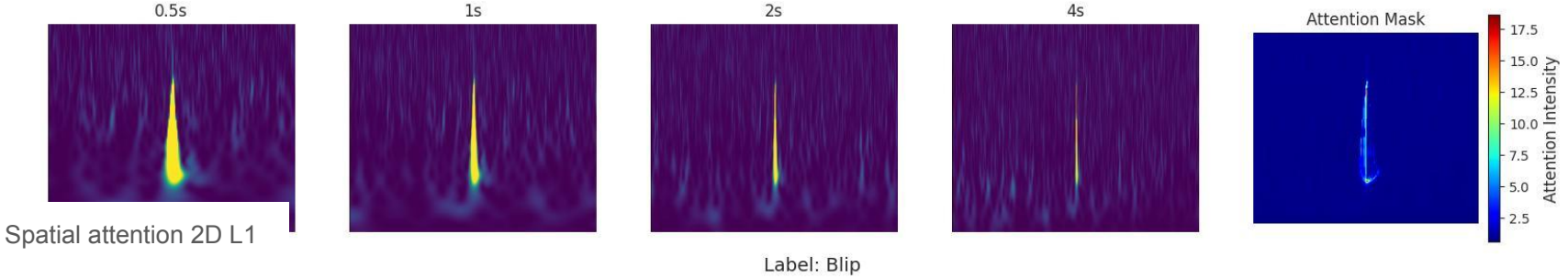
Spatial Attention in the two first layers

Preliminary results



Spatial attention 2D L0

Label: Blip

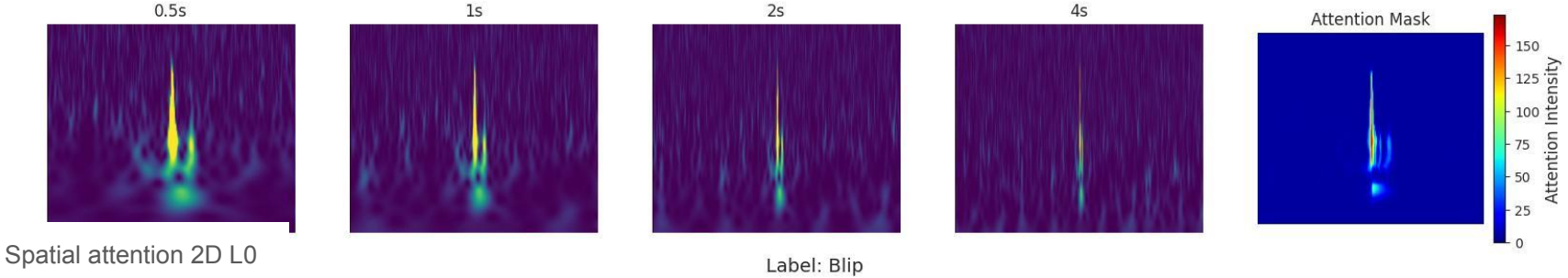


Spatial attention 2D L1

Label: Blip

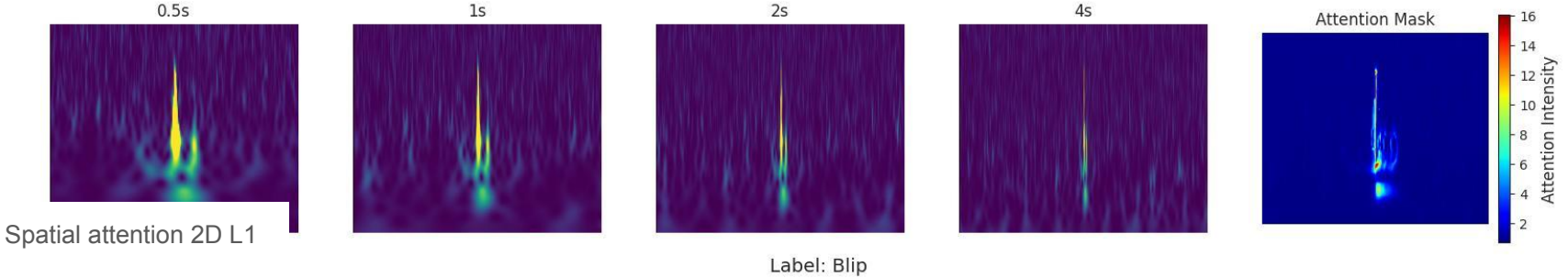
Spatial Attention in the two first layers

Preliminary results



Spatial attention 2D L0

Label: Blip

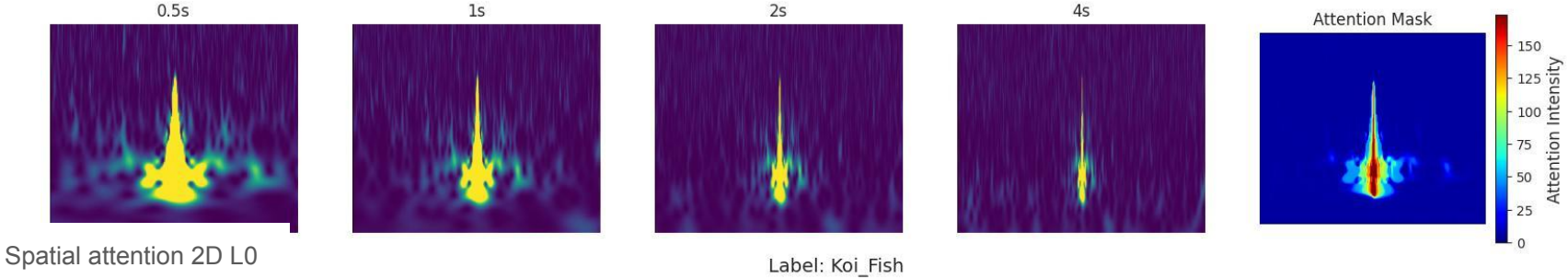


Spatial attention 2D L1

Label: Blip

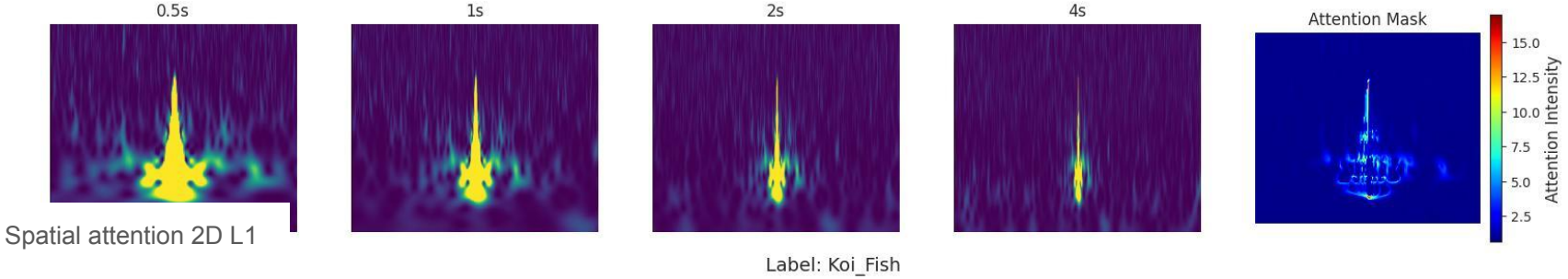
Spatial Attention in the two first layers

Preliminary results



Spatial attention 2D L0

Label: Koi_Fish

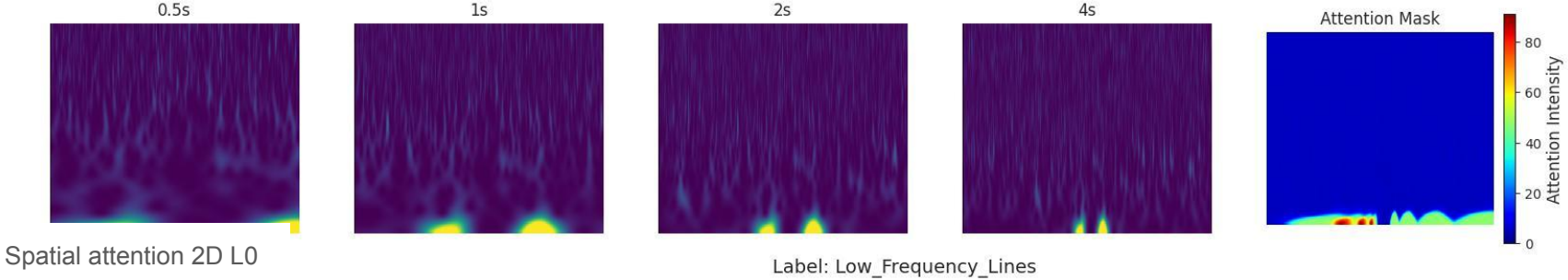


Spatial attention 2D L1

Label: Koi_Fish

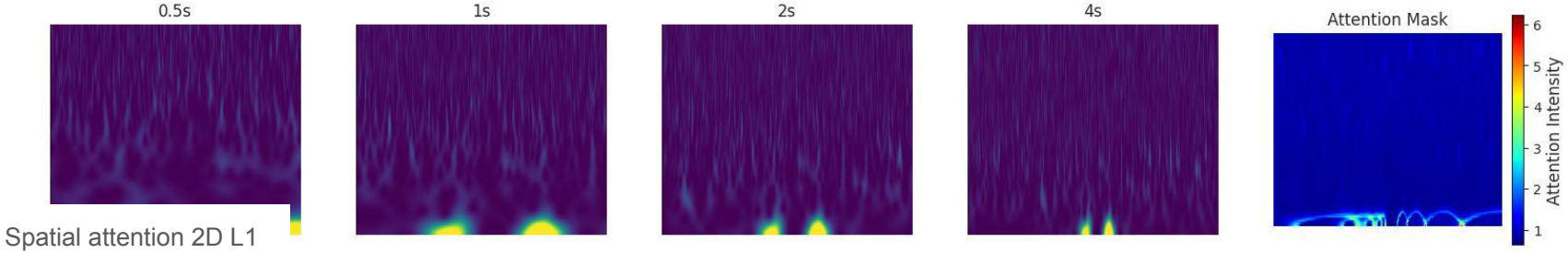
Spatial Attention in the two first layers

Preliminary results



Spatial attention 2D L0

Label: Low_Frequency_Lines



Spatial attention 2D L1

Label: Low_Frequency_Lines

Recap

- Neural Networks are **not inherently black boxes**.
- Explainable AI (XAI) enables the visualization of key input features.
- This helps practitioners address misclassifications without solely relying on deeper networks to enhance performance.
- XAI is the **connection** between AI practitioners and physicists. XAI is a dictionary between AI and another discipline!