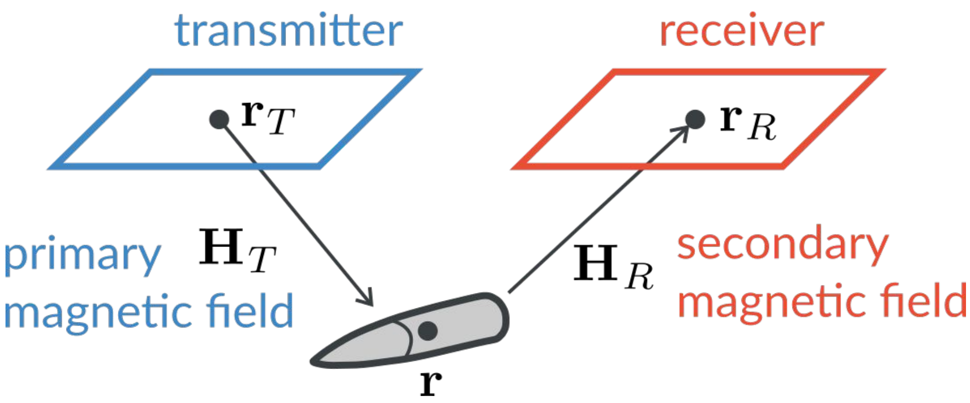
A convolutional neural network for the classification of UXO in marine settings

Jorge Lopez-Alvis[1], Lindsey J. Heagy[1], Douglas W. Oldenburg[1], Stephen Billings[2], Lin-Ping Song[2]

[1]University of British Columbia, [2]Black Tusk Geophysics

This work is supported by SERDP project MR22-3487

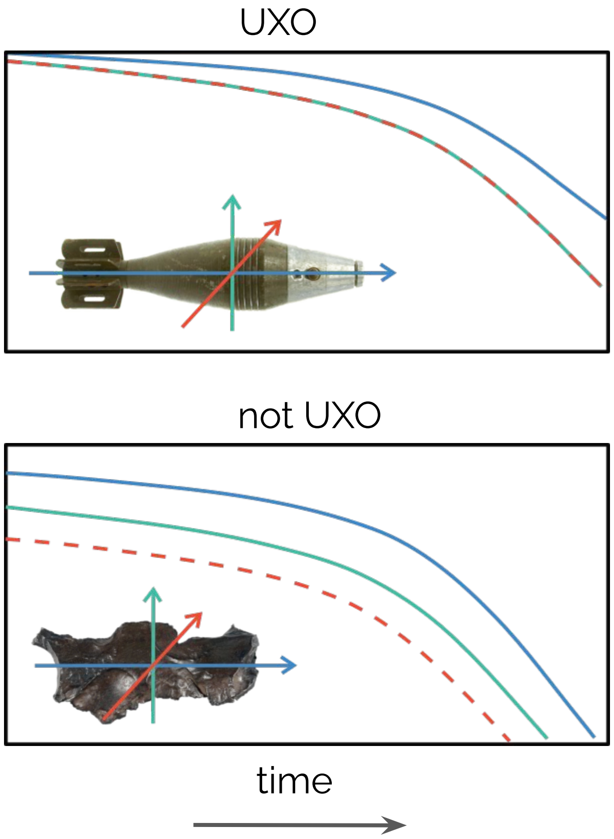
Time-domain EM response of a UXO



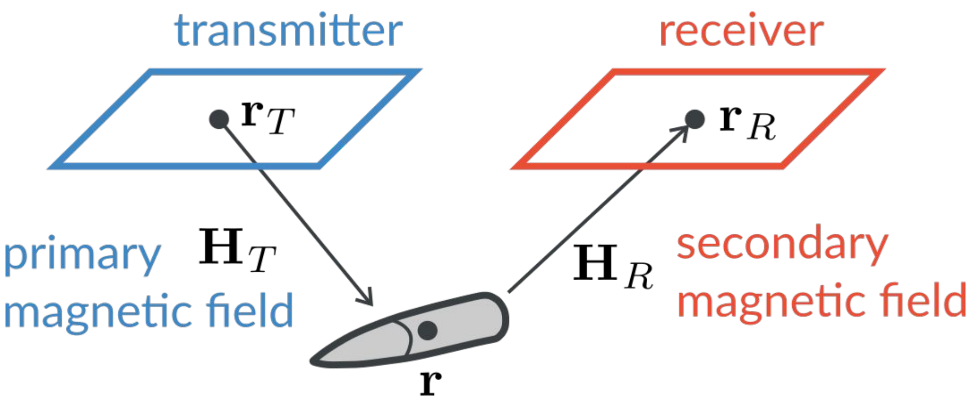
$$d(\mathbf{r}_R, t) = \mathbf{H}_R(\mathbf{r}, \mathbf{r}_R) \cdot \mathbf{P}(t) \cdot \mathbf{H}_T(\mathbf{r}, \mathbf{r}_T)$$

$$\mathbf{P}(t) = \mathbf{A}(\phi, \theta, \psi) \cdot \mathbf{L}(t) \cdot \mathbf{A}^\top(\phi, \theta, \psi)$$

$$\mathbf{L}(t) = \begin{pmatrix} L_1 & & \\ & L_2 & \\ & & L_3 \end{pmatrix}$$



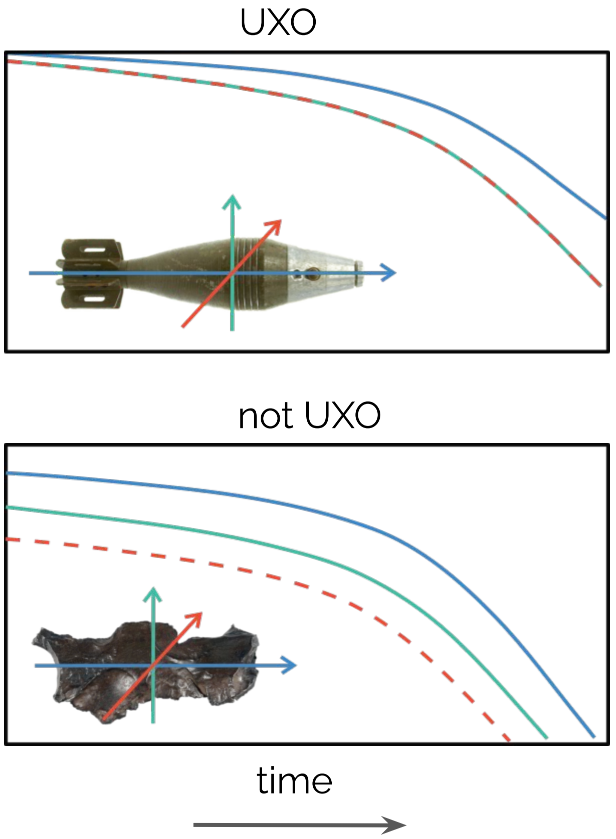
Time-domain EM response of a UXO



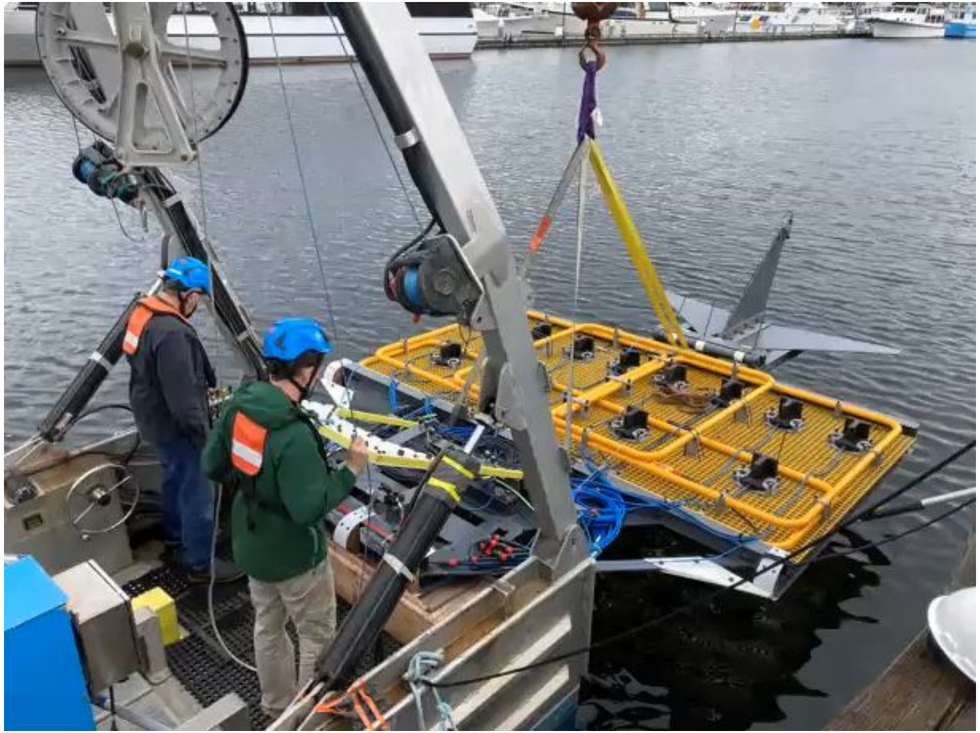
$$d(\mathbf{r}_R, t) = \mathbf{H}_R(\underbrace{\mathbf{r}}, \mathbf{r}_R) \cdot \mathbf{P}(t) \cdot \mathbf{H}_T(\mathbf{r}, \mathbf{r}_T)$$

$$\mathbf{P}(t) = \underbrace{\mathbf{A}(\phi, \theta, \psi)} \cdot \mathbf{L}(t) \cdot \mathbf{A}^\top(\phi, \theta, \psi) \qquad \mathbf{L}(t) = \underbrace{\begin{pmatrix} L_1 & & \\ & L_2 & \\ & & L_3 \end{pmatrix}}$$

traditional approach: use inversion to get these and then classify based on **L**(t)



Survey and system



UltraTEMA-4 system:

4 transmitters

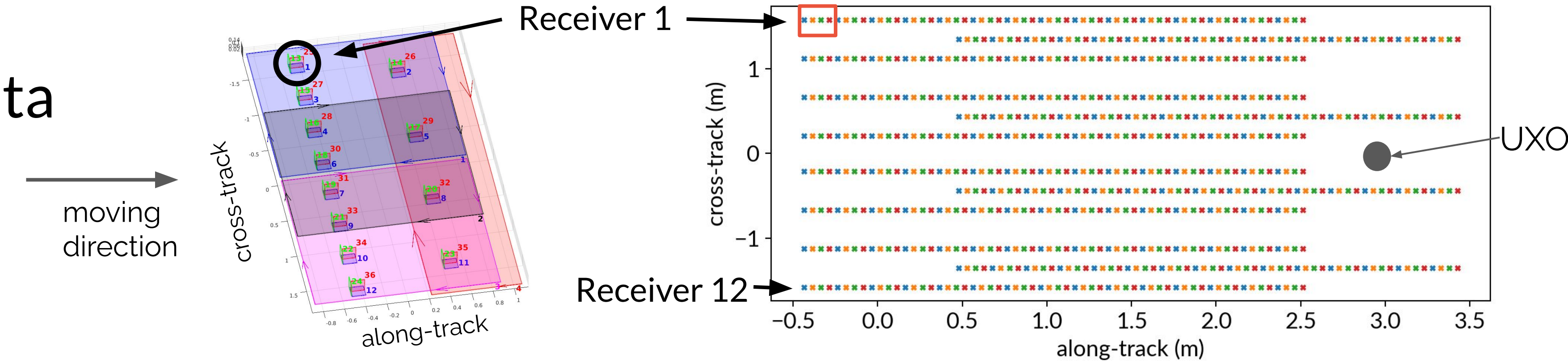
12 receivers (3-component)

Height above sea-bottom: ~1 m

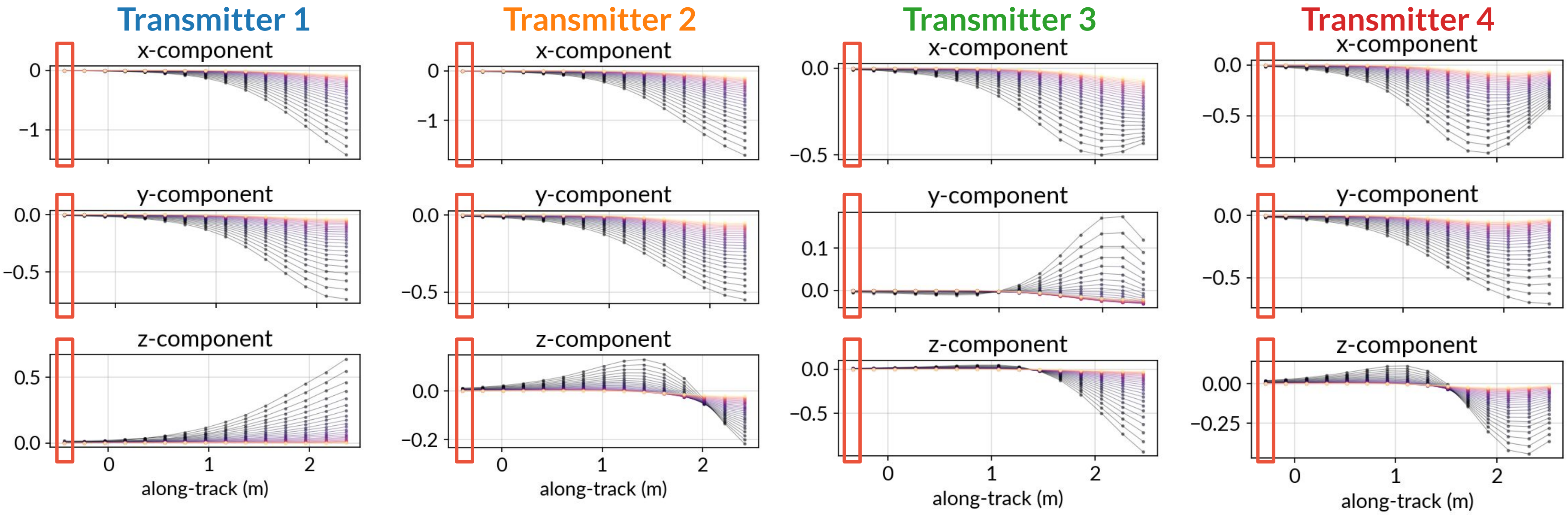
Challenges:

- Accuracy in location
- EM response of seawater

Data

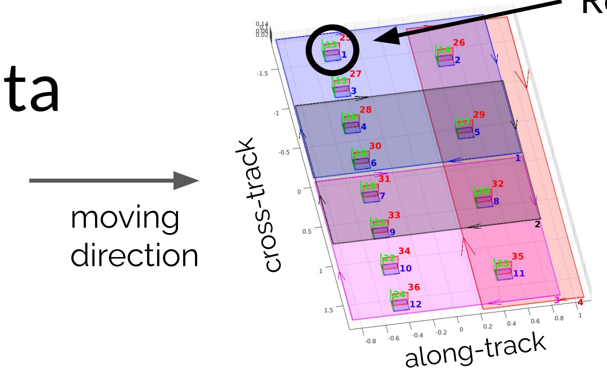


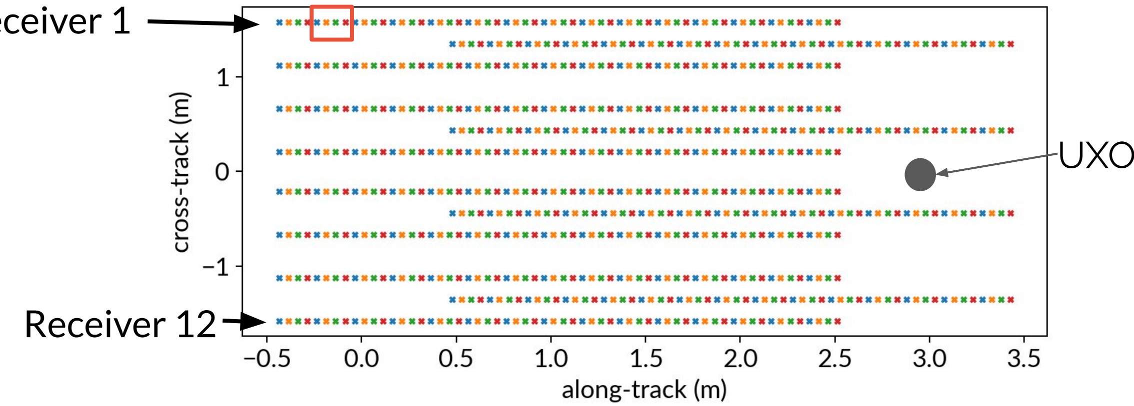
moving direction
cross-track
along-track
Receiver 1
Receiver 12
UXO
cross-track (m)
along-track (m)

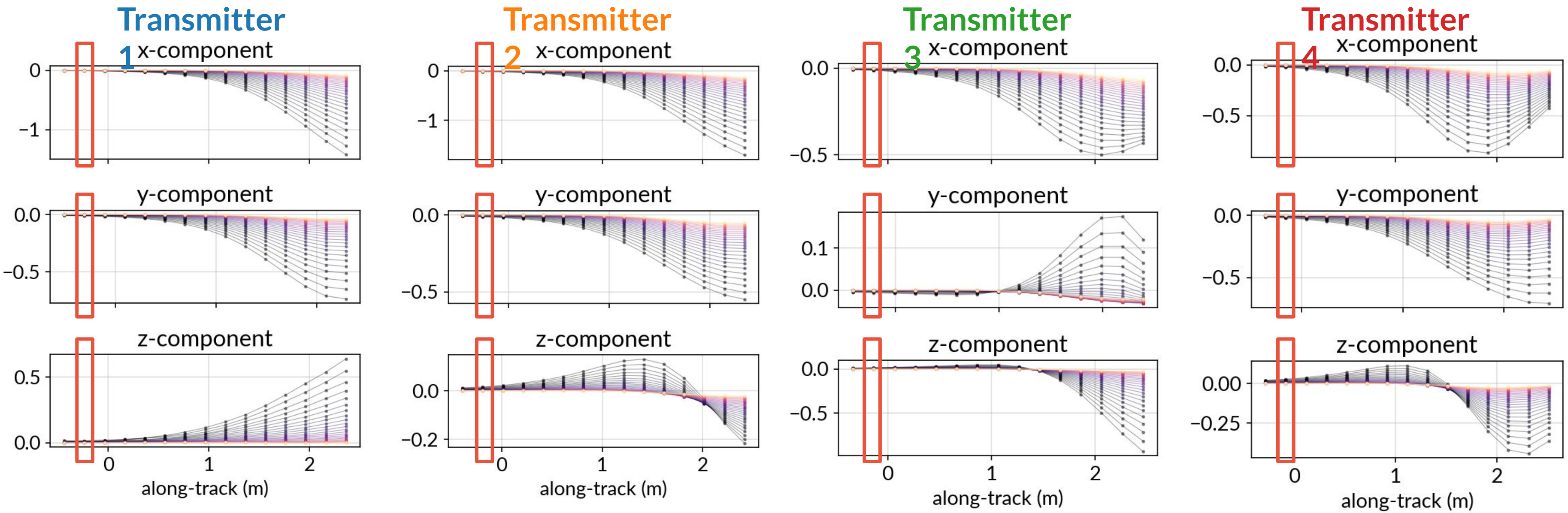


Transmitter 1
Transmitter 2
Transmitter 3
Transmitter 4
x-component
y-component
z-component
along-track (m)

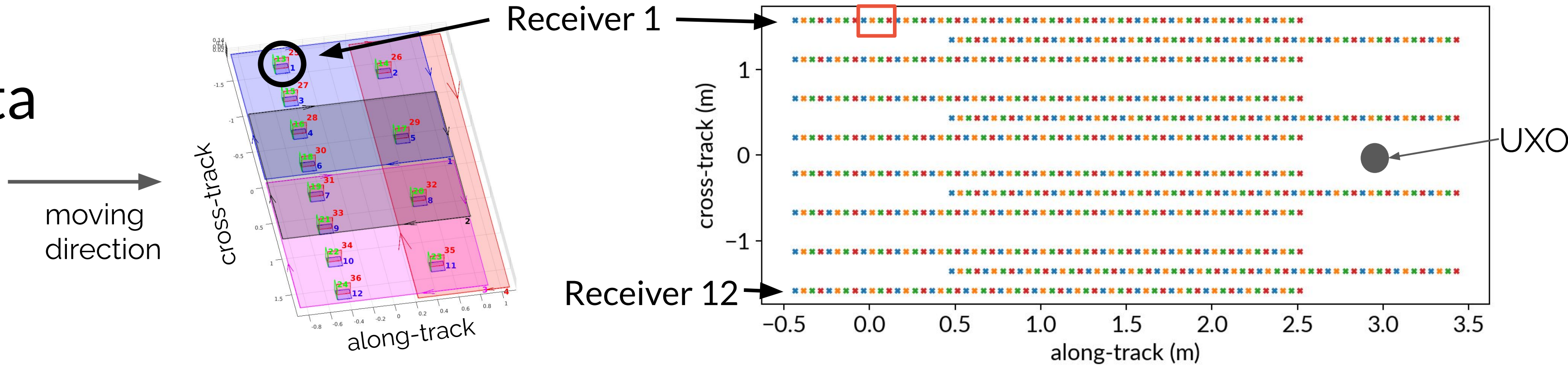
Data

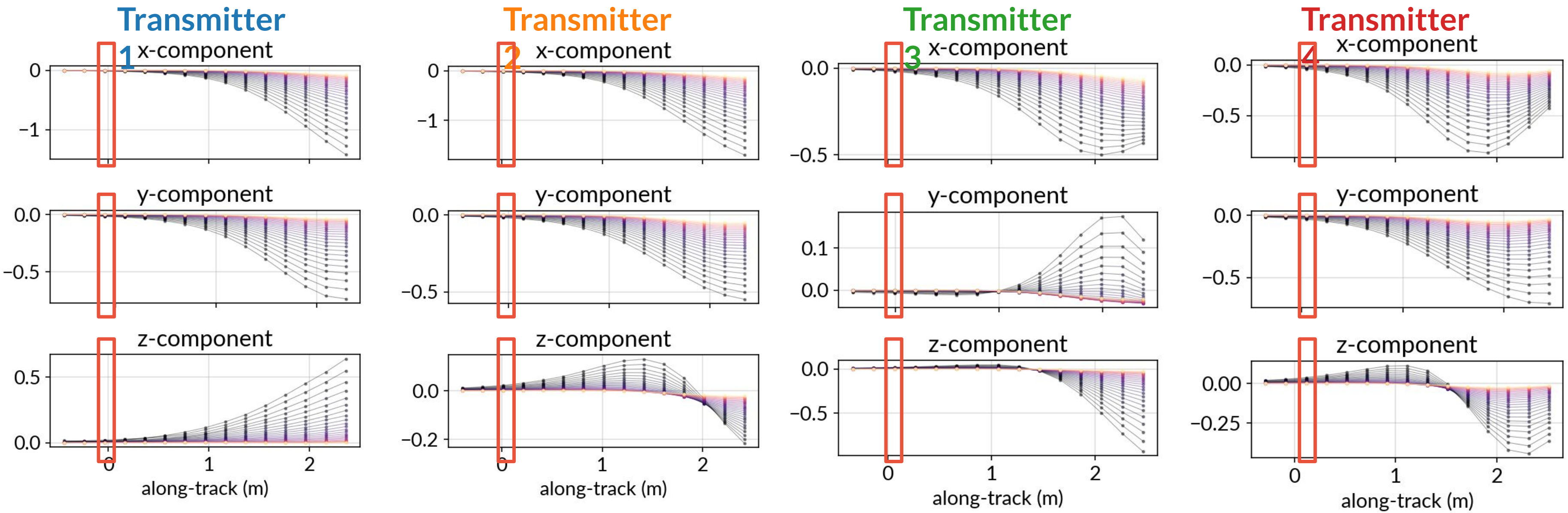

moving direction
cross-track
along-track


Receiver 1
Receiver 12
UXO
cross-track (m)
along-track (m)
1
0
−1
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

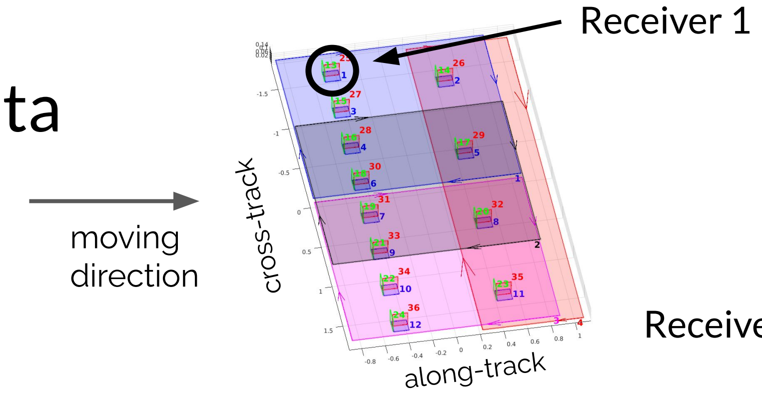

Transmitter 1
Transmitter 2
Transmitter 3
Transmitter 4
x-component
y-component
z-component
along-track (m)

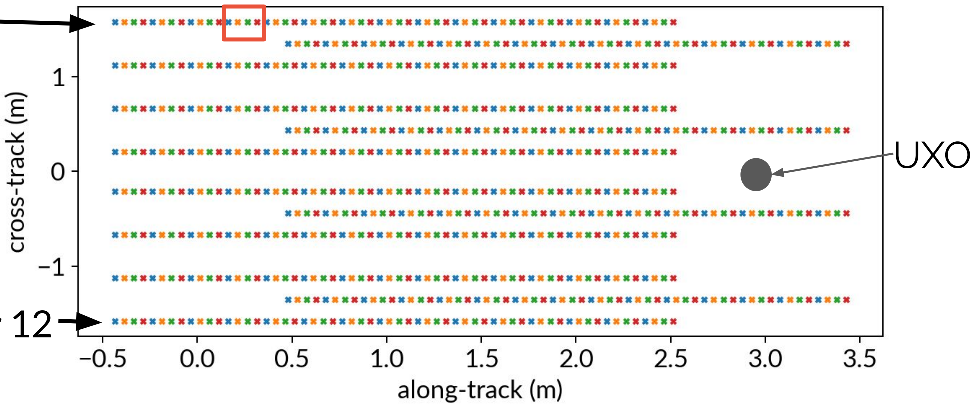
Data

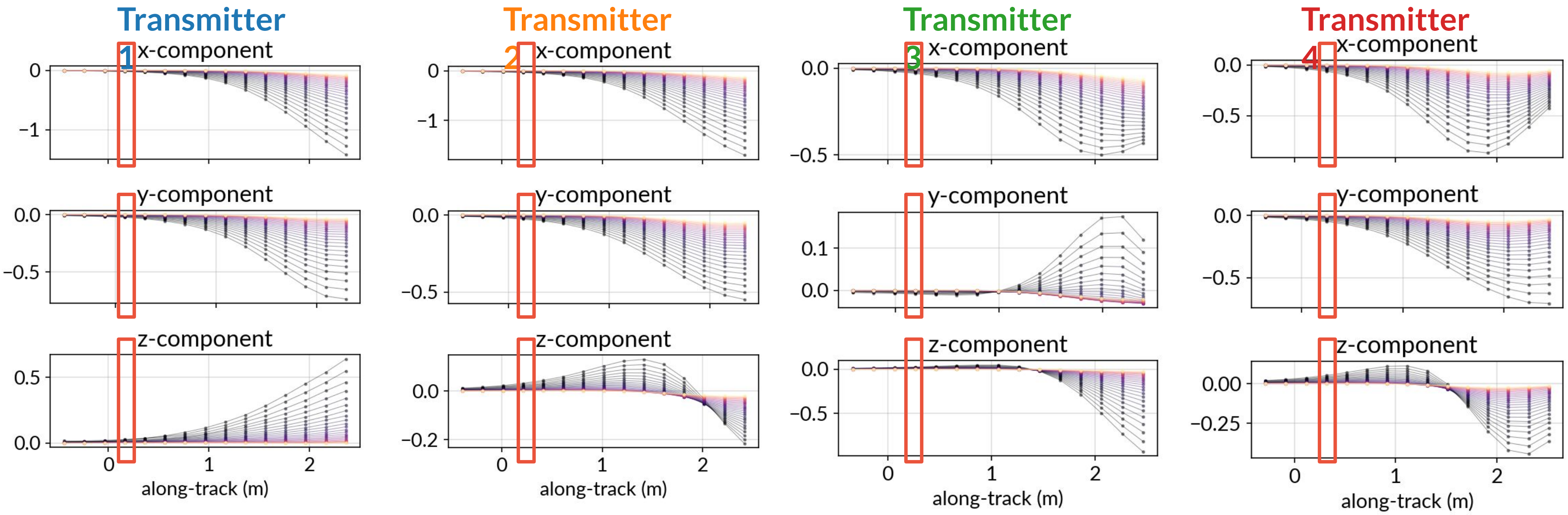

Receiver 1
Receiver 12
moving
direction
cross-track
along-track
cross-track (m)
along-track (m)
UXO


Transmitter 1
Transmitter 2
Transmitter 3
Transmitter 4
x-component
y-component
z-component
along-track (m)

Data

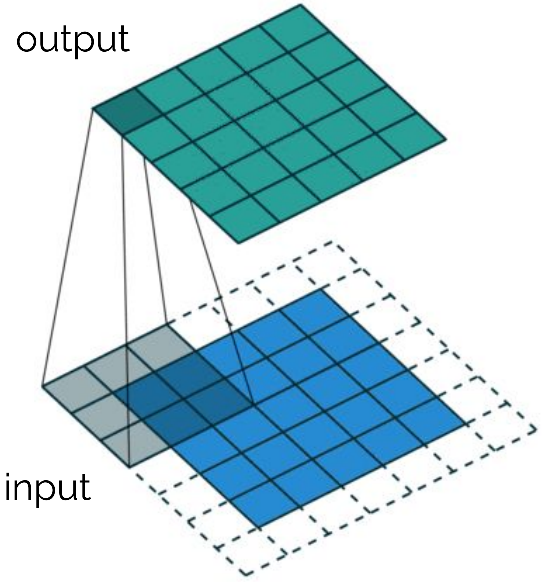

moving
direction
cross-track
along-track
Receiver 1

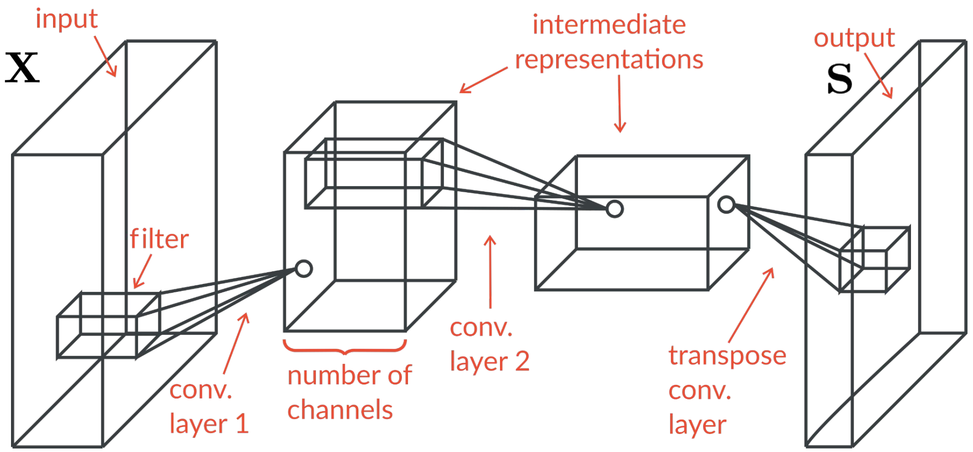

Receiver 1
Receiver 12
UXO
cross-track (m)
along-track (m)


Transmitter 1
Transmitter 2
Transmitter 3
Transmitter 4
x-component
y-component
z-component
along-track (m)

Can we classify directly from data?

Densely sampled data and correlated in space and time: a good candidate for convolutional neural networks.





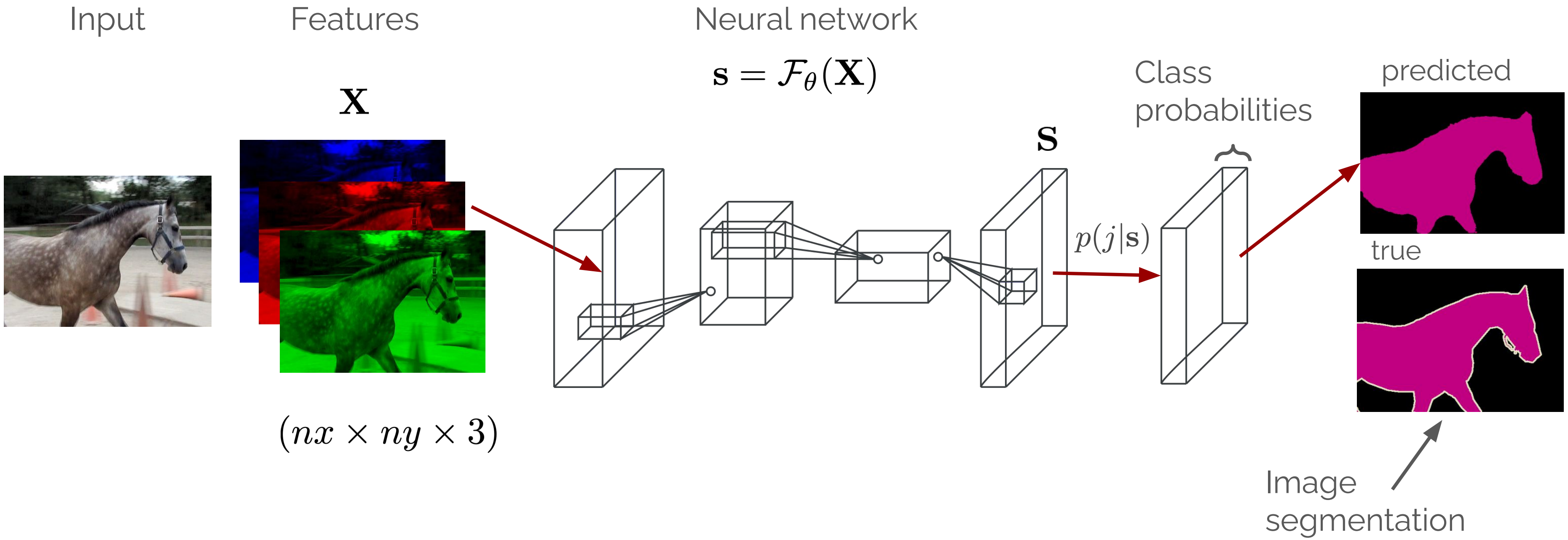
$$\mathbf{s} = \mathcal{F}_\theta(\mathbf{X})$$

https://github.com/vdumoulin/conv_arithmetic/blob/master/README.md

Convolutional Neural Networks

Supervised classification problem

provided data with labels, construct a function (network) that outputs labels given input data



Convolutional Neural Networks

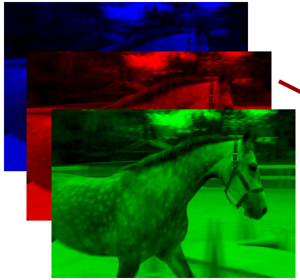
Training

define an optimization problem to estimate network parameters

Input

Features

$\mathbf{X}$



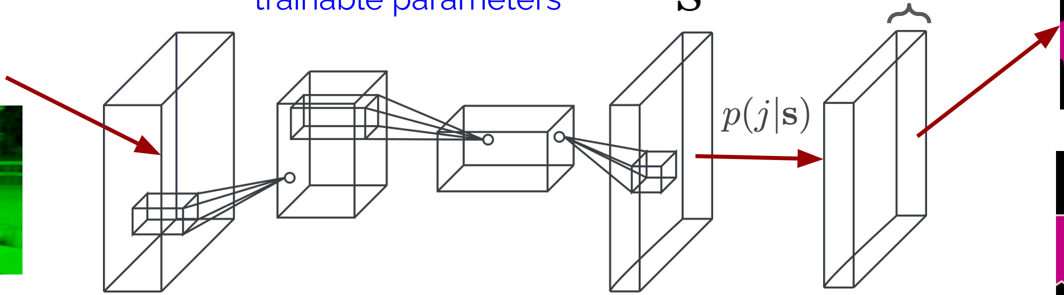
$(nx \times ny \times 3)$

Neural network

$\mathbf{s} = \mathcal{F}_{\underline{\theta}}(\mathbf{X})$

trainable parameters

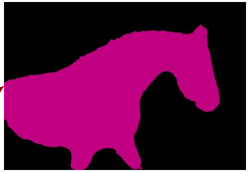
$\mathbf{S}$



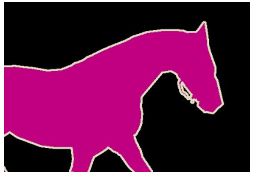
$p(j|\mathbf{s})$

Class probabilities

predicted



true

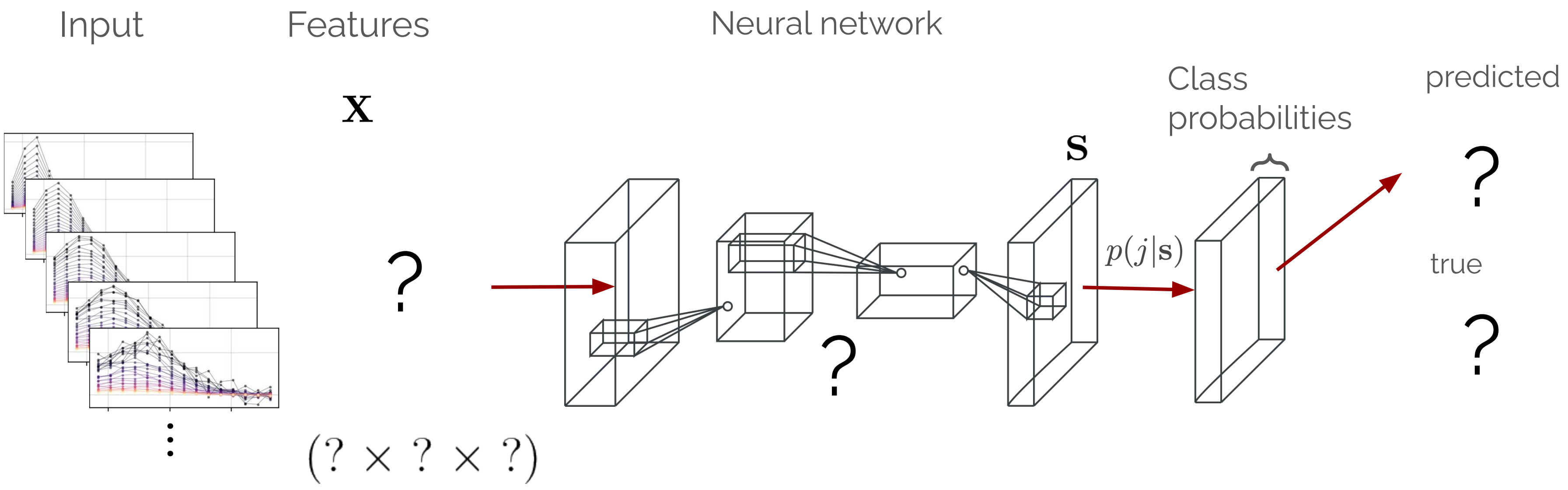


Measure: cross entropy loss

$$\min_{\theta} \phi = -\sum q_j \log(p_j)$$

Convolutional Neural Networks

How do we translate these things to the UXO classification problem?



Convolutional Neural Networks

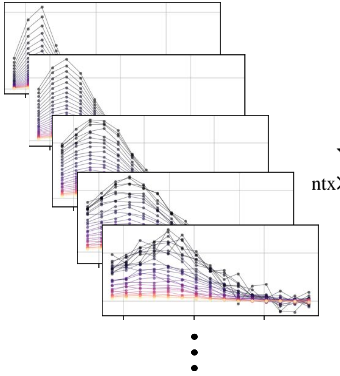
How do we translate these things to the UXO classification problem?

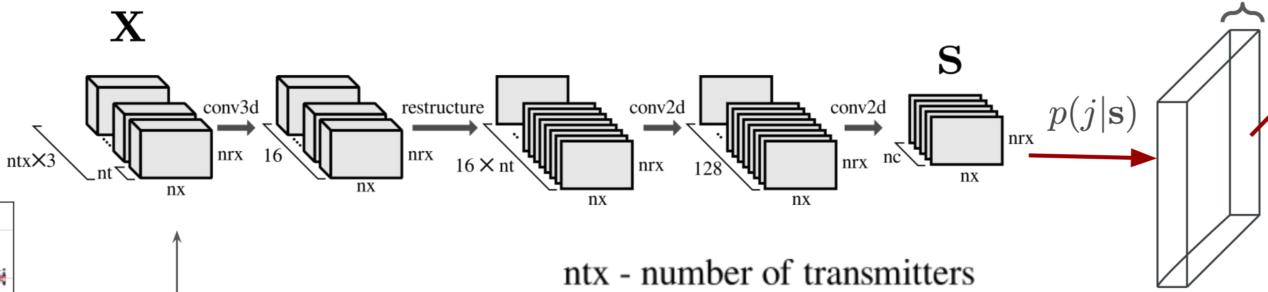
Input

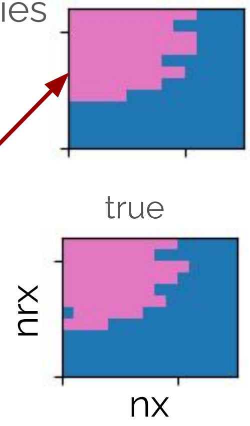
Features

Neural network

Class probabilities







$$(nx \times nrx \times nt \times (ntx \times 3))$$

ntx - number of transmitters

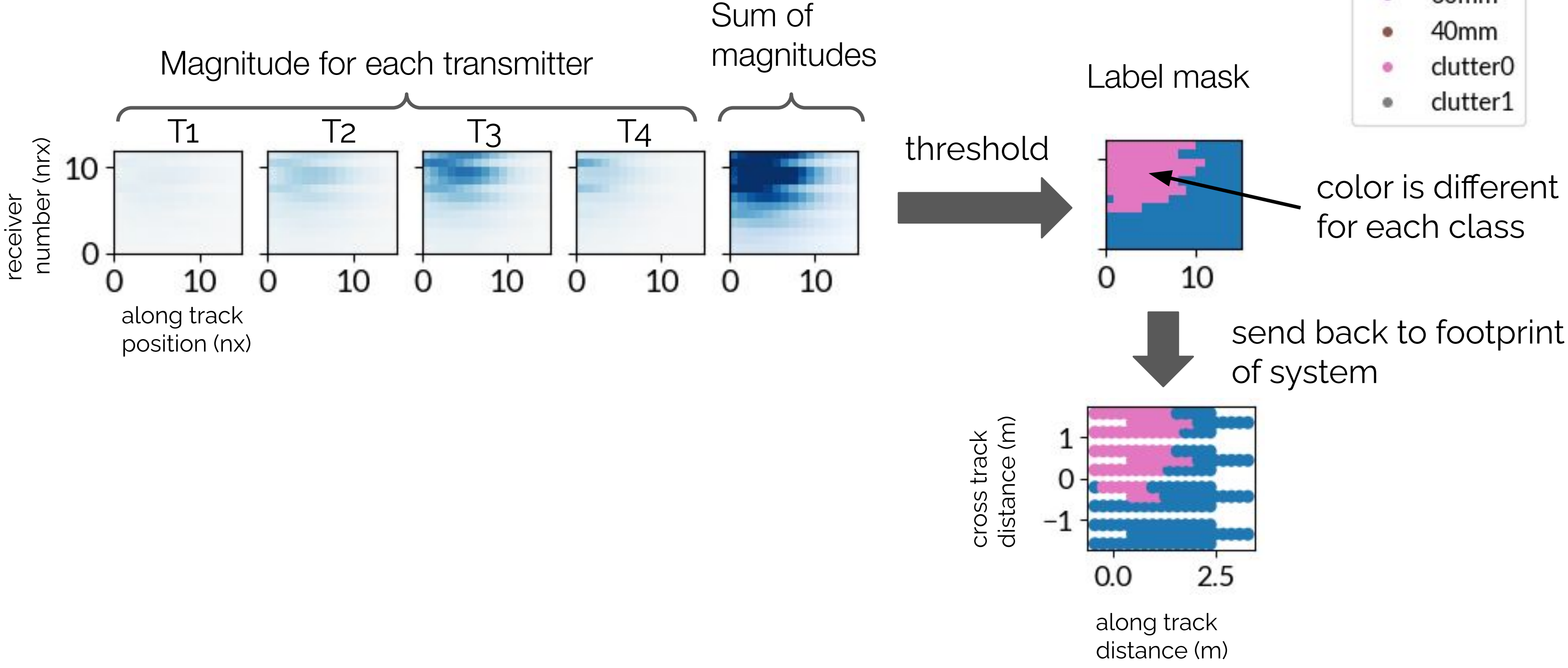
nrx - number of receiver cubes

nt - number of time channels

nx - number of positions in spatial window (along track)

nc - number of classes

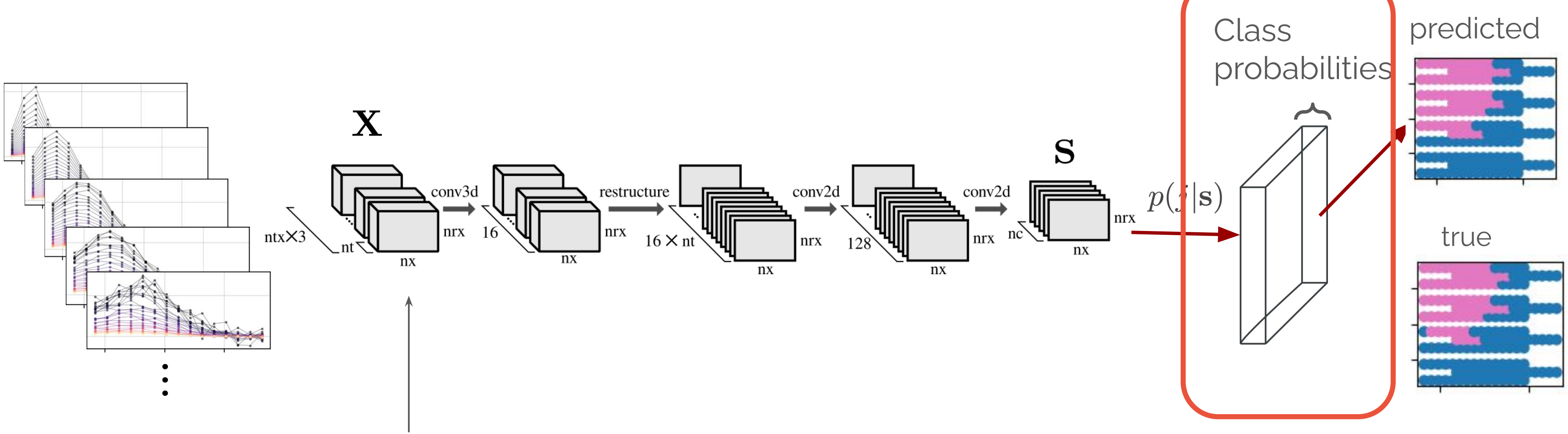
Defining label masks



Convolutional Neural Networks

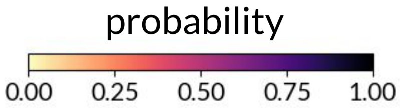
How do we translate these things to the UXO classification problem?

Input | Features | Neural network

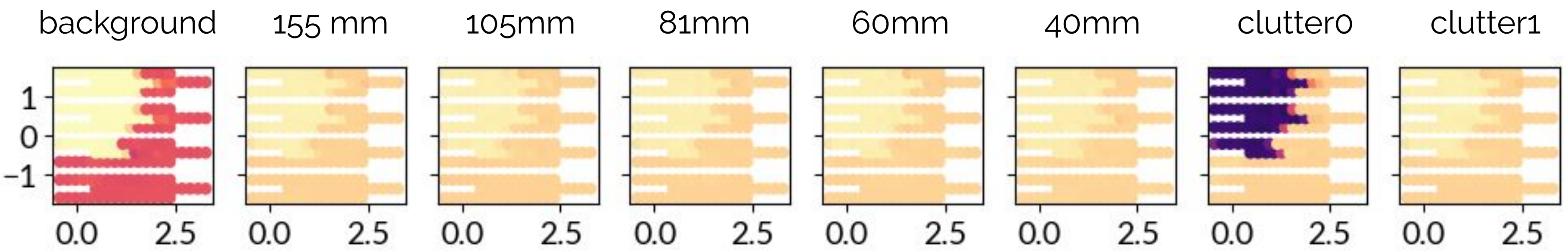


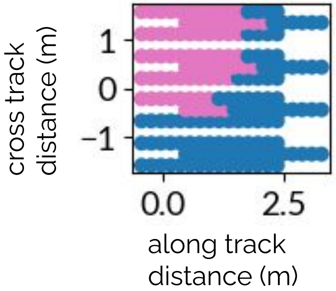
$$(\boldsymbol{nx} \times \boldsymbol{nrx} \times \boldsymbol{nt} \times (\boldsymbol{ntx} \times \mathbf{3}))$$

Probability layer and classification



eight different classes:

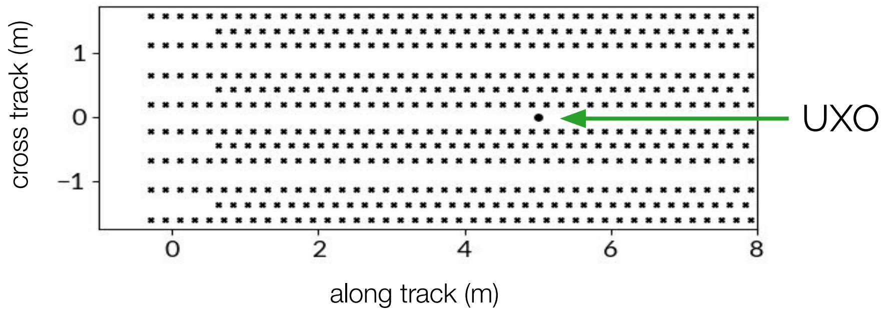




point-wise classification according to max probability

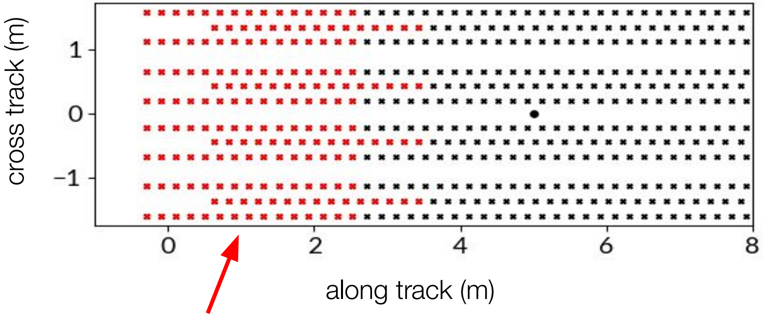
Application to a line of data

Input features are created by using a sliding window:



Application to a line of data

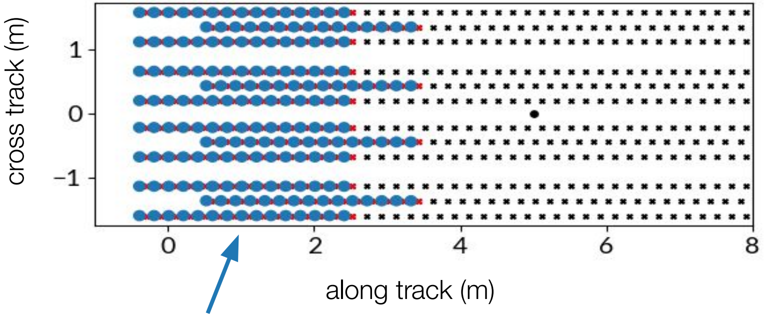
Input features are created by using a sliding window:



sliding window

Application to a line of data

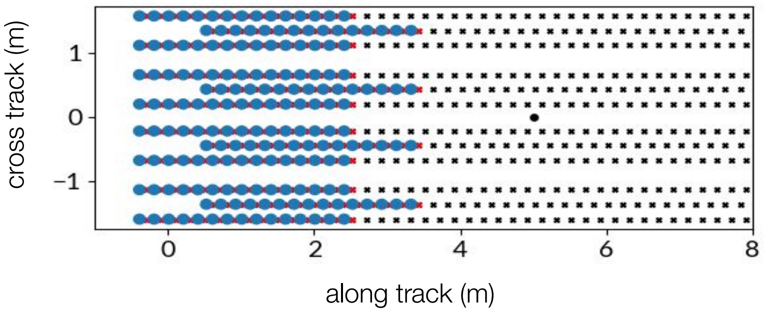
Input features are created by using a sliding window:



Neural network output (class)

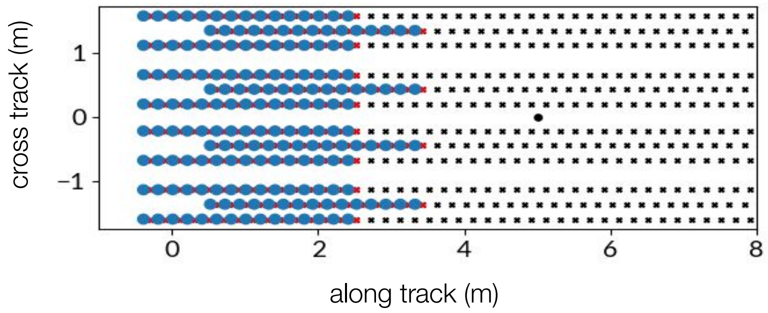
Application to a line of data

Input features are created by using a sliding window:

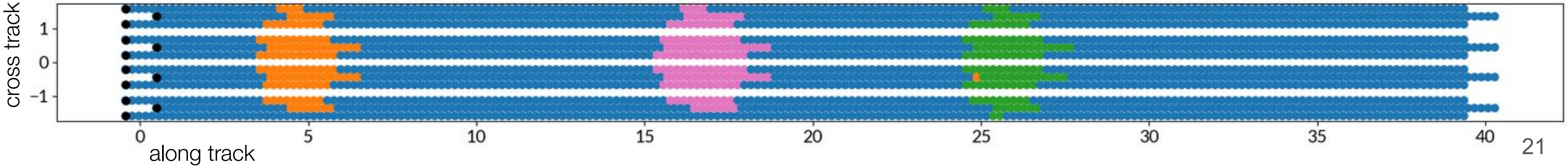


Application to a line of data

Input features are created by using a sliding window:



Single acquisition line with three objects (classification results)



Training for marine data

8 classes:

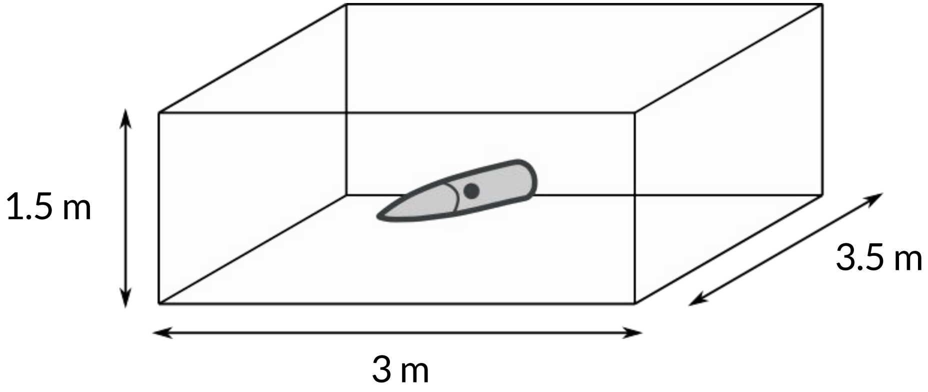
- background
- 155 mm
- 105 mm
- 81 mm
- 60 mm
- 40 mm
- Clutter0 (spheres and disks)
- Clutter1 (rods)

# of realizations:

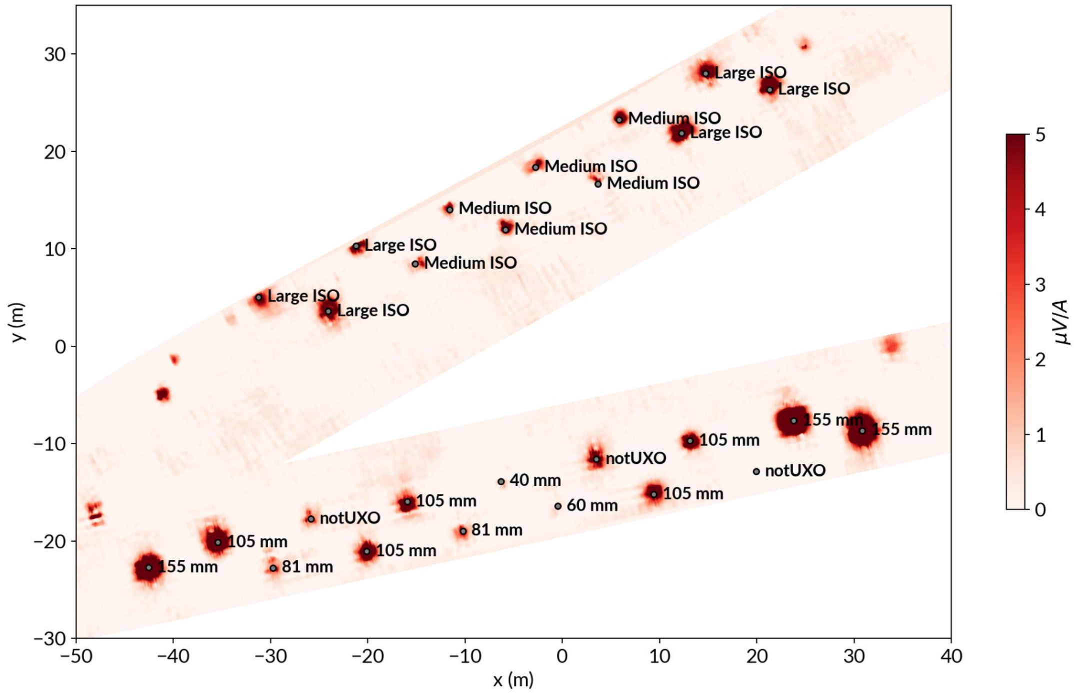
- Training: 8192, 81920
- Test and validation: 1024, 1024

Randomly assign:

- Target class
- Location (x, y, z)
- Orientation (ϕ, θ, ψ)
- Noise level: approximate from background areas in the field data

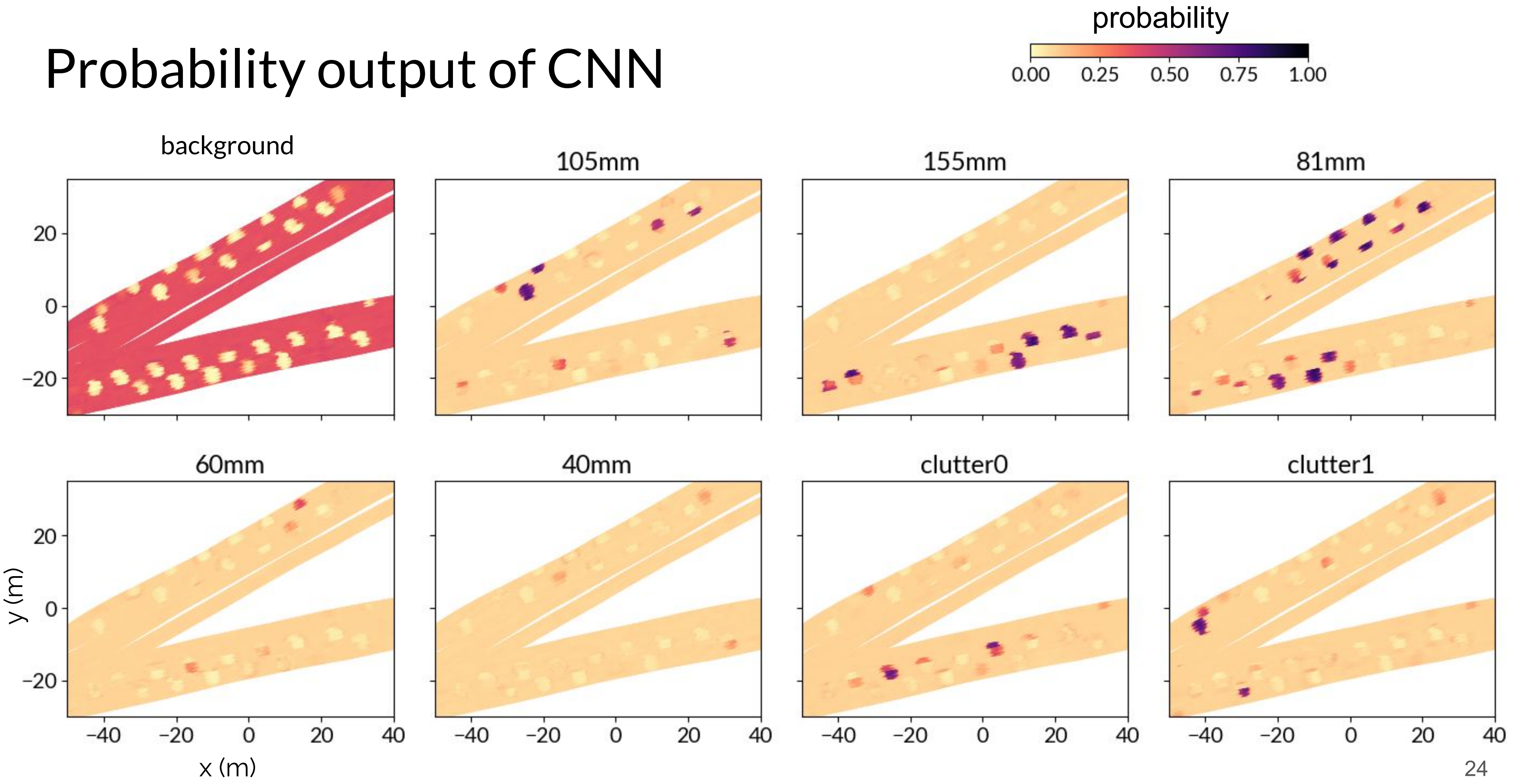


Calibration line Sequim Bay 2021

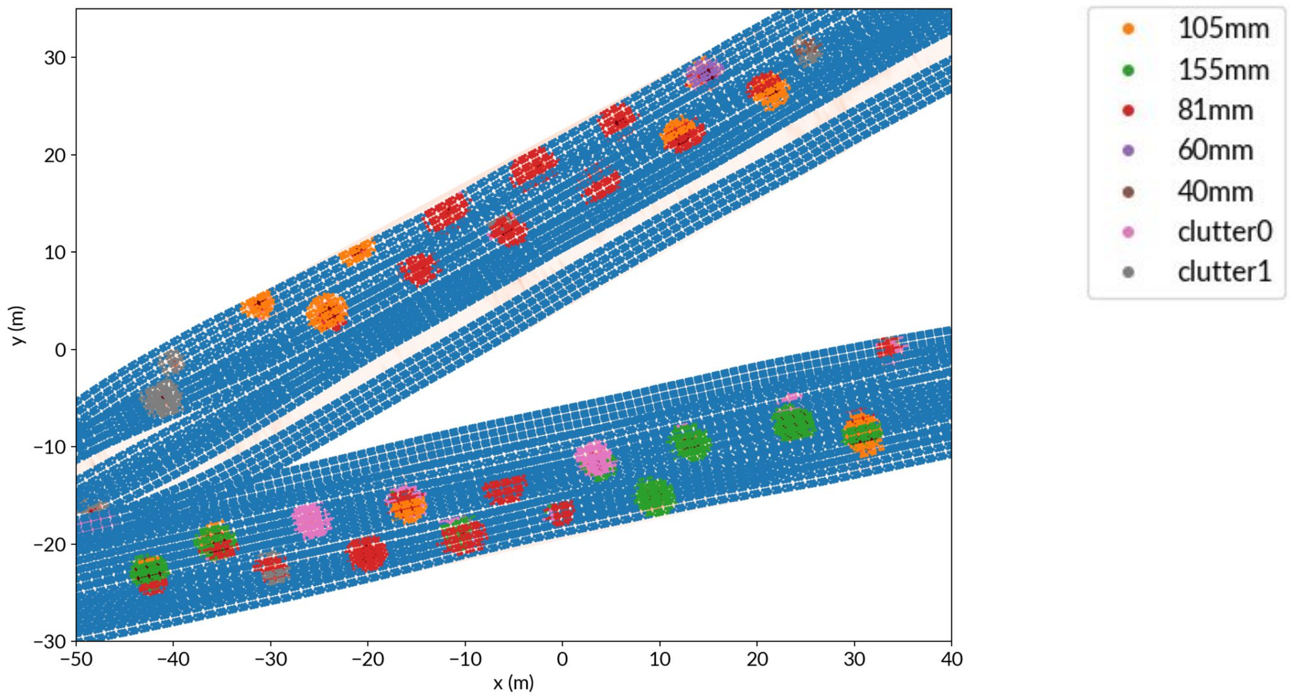


- 12 acquisition lines
- Current CNN requires seawater response removed
- Some ISOs present, we used only UXO objects to train (e.g. medium ISO ~ 61mm)

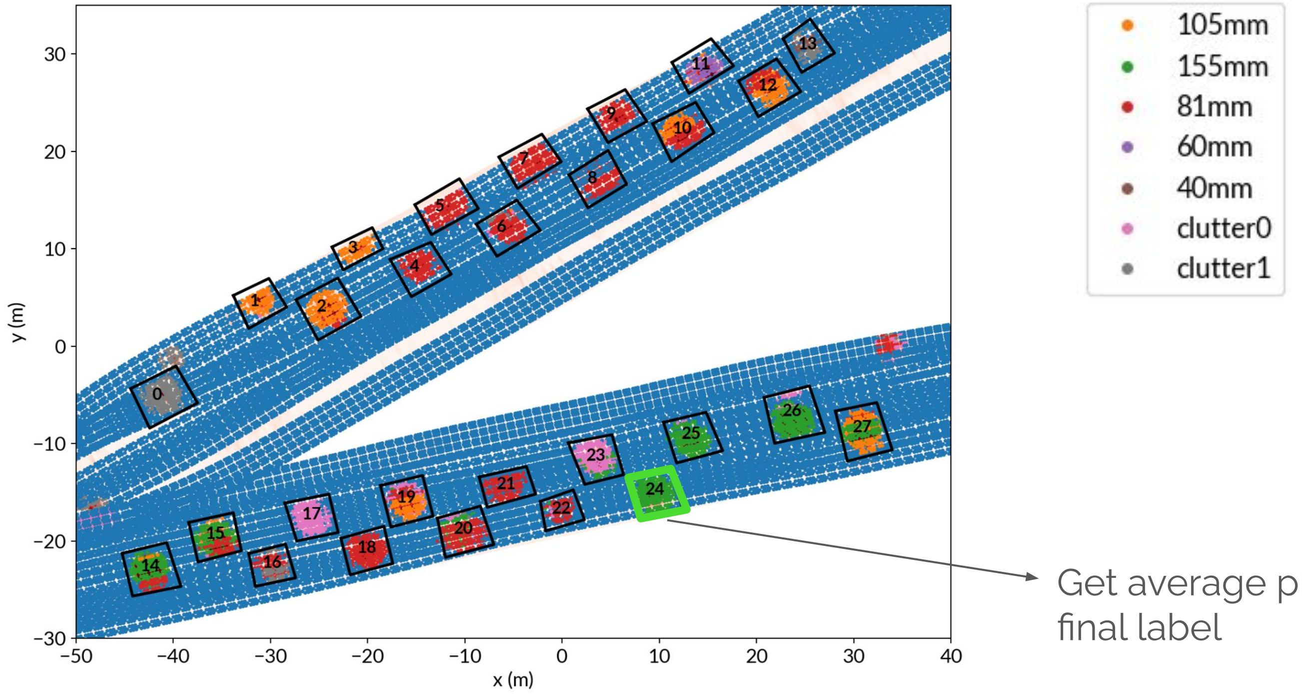
Probability output of CNN



Classification output of CNN - calibration line 2021

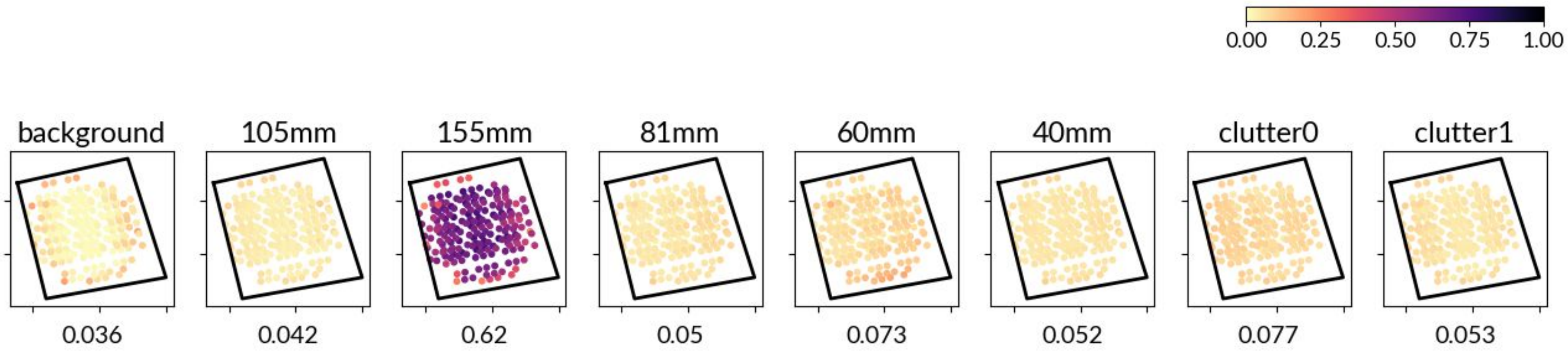


Divide in cells to get a single probability value per cell:



Get average probability for cell and assign final label

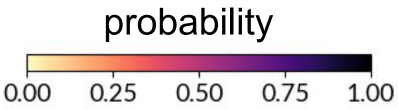
Average probability values for one cell:

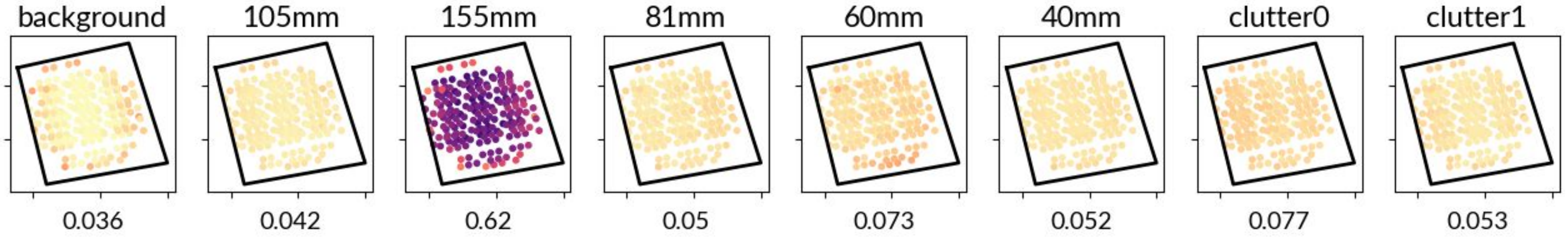


average probabilities (using all points not classified as background)

Assign label with highest probability: “155 mm”

Average probability values for one cell:





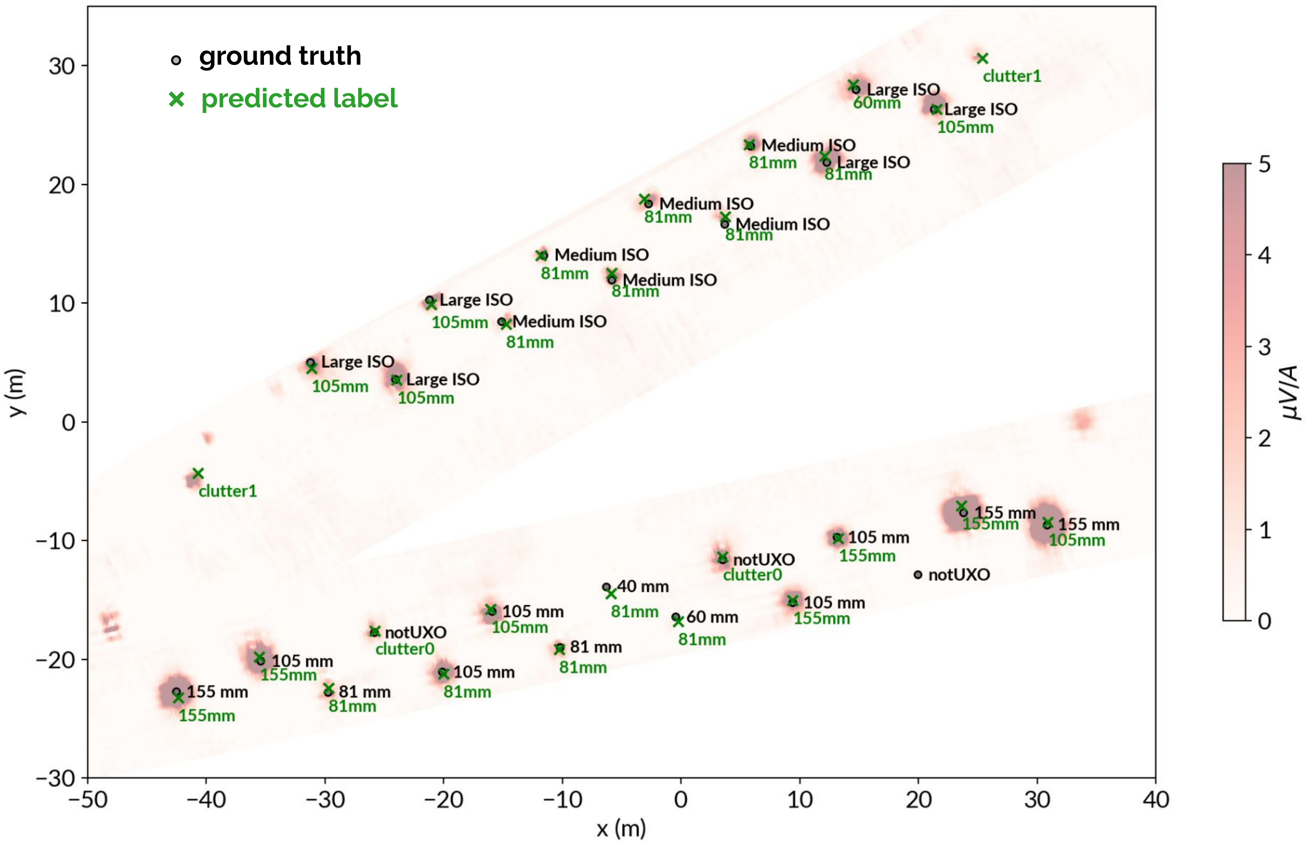
average probabilities (using all points not classified as background)

Assign label with highest probability: "155 mm"

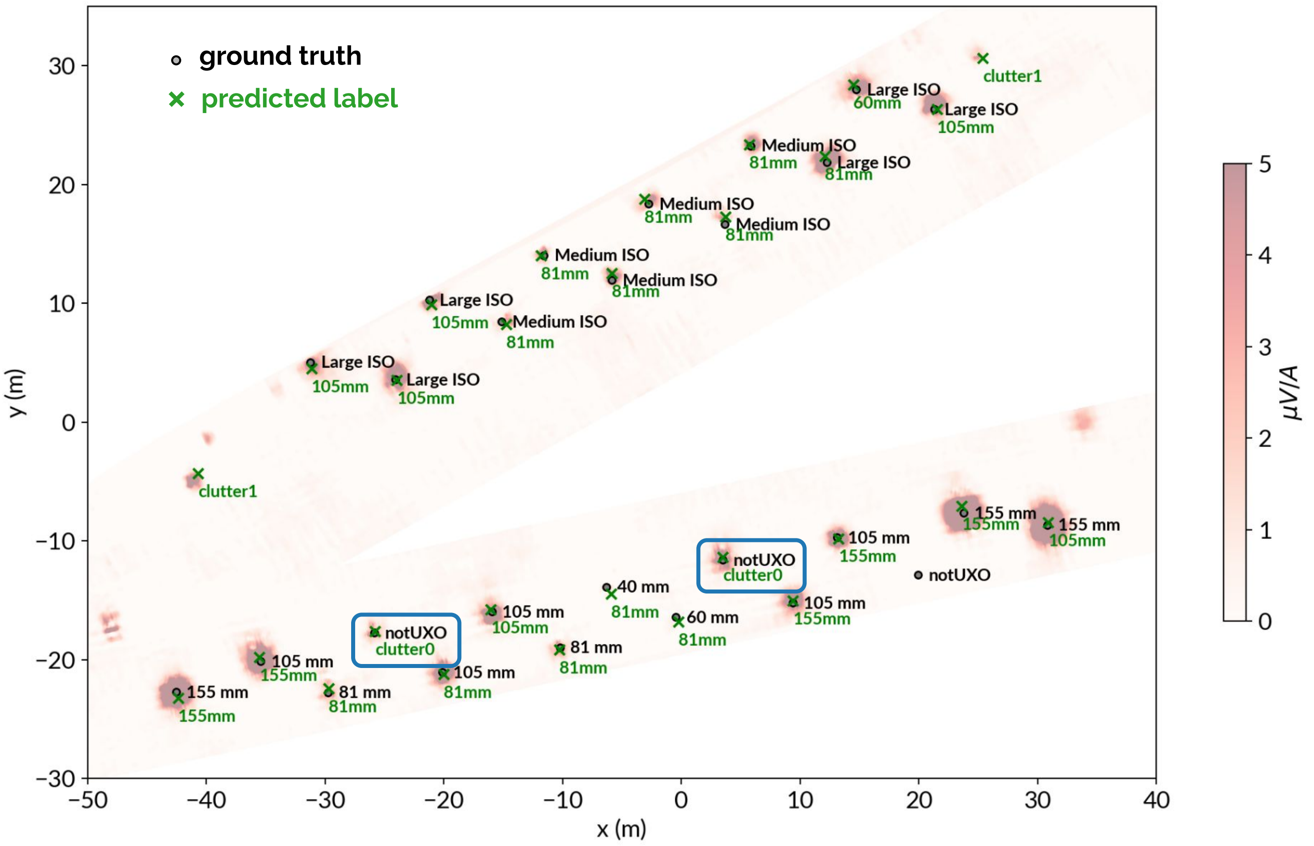
If object is classified as "clutter", apply "safety" rule.

Safety rule:
if prob(clutter) < 0.3 then change label to next likely UXO

Predicted labels vs truth labels - calibration line 2021

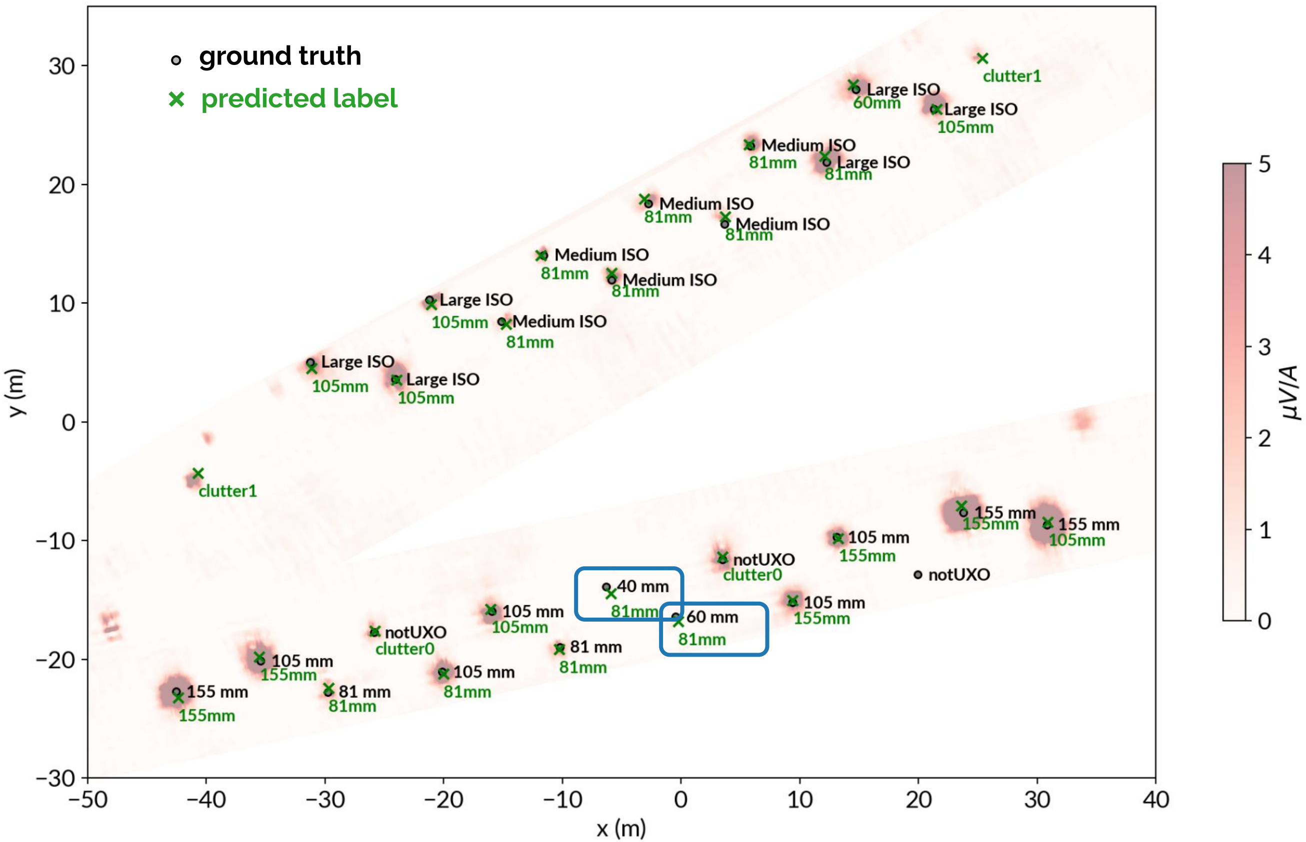


Predicted labels vs truth labels - calibration line 2021



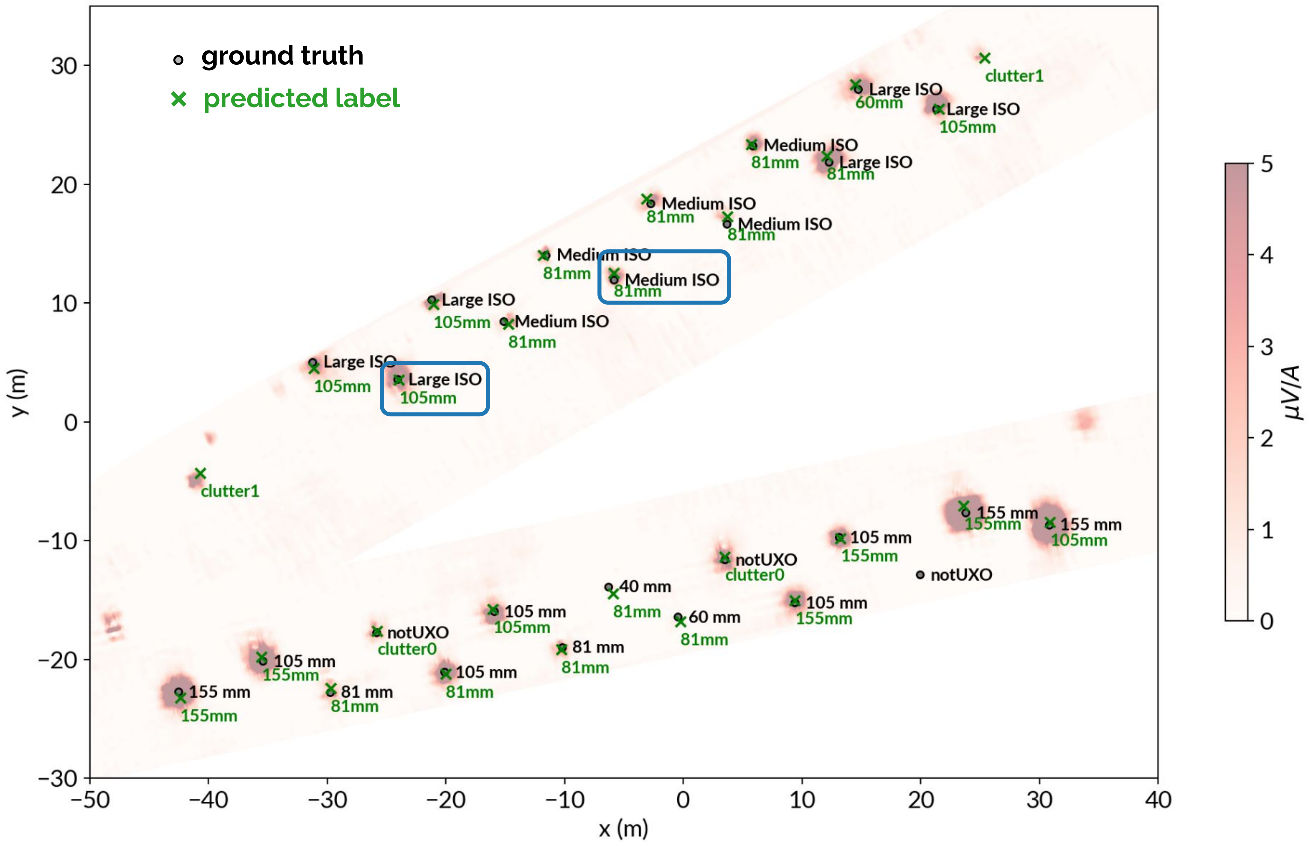
- Correctly predicted clutter

Predicted labels vs truth labels - calibration line 2021



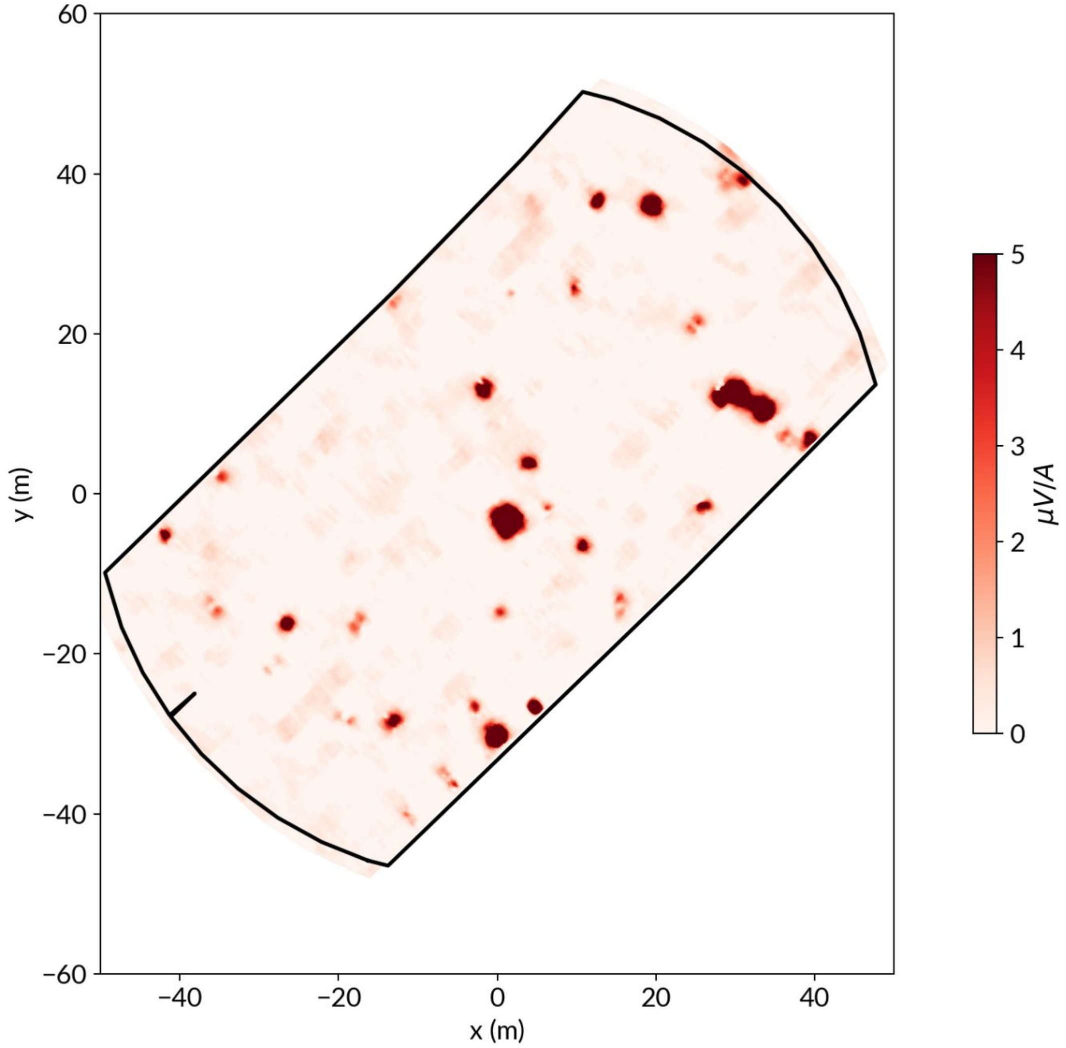
- Correctly predicted clutter
- Did not miss any UXO

Predicted labels vs truth labels - calibration line 2021



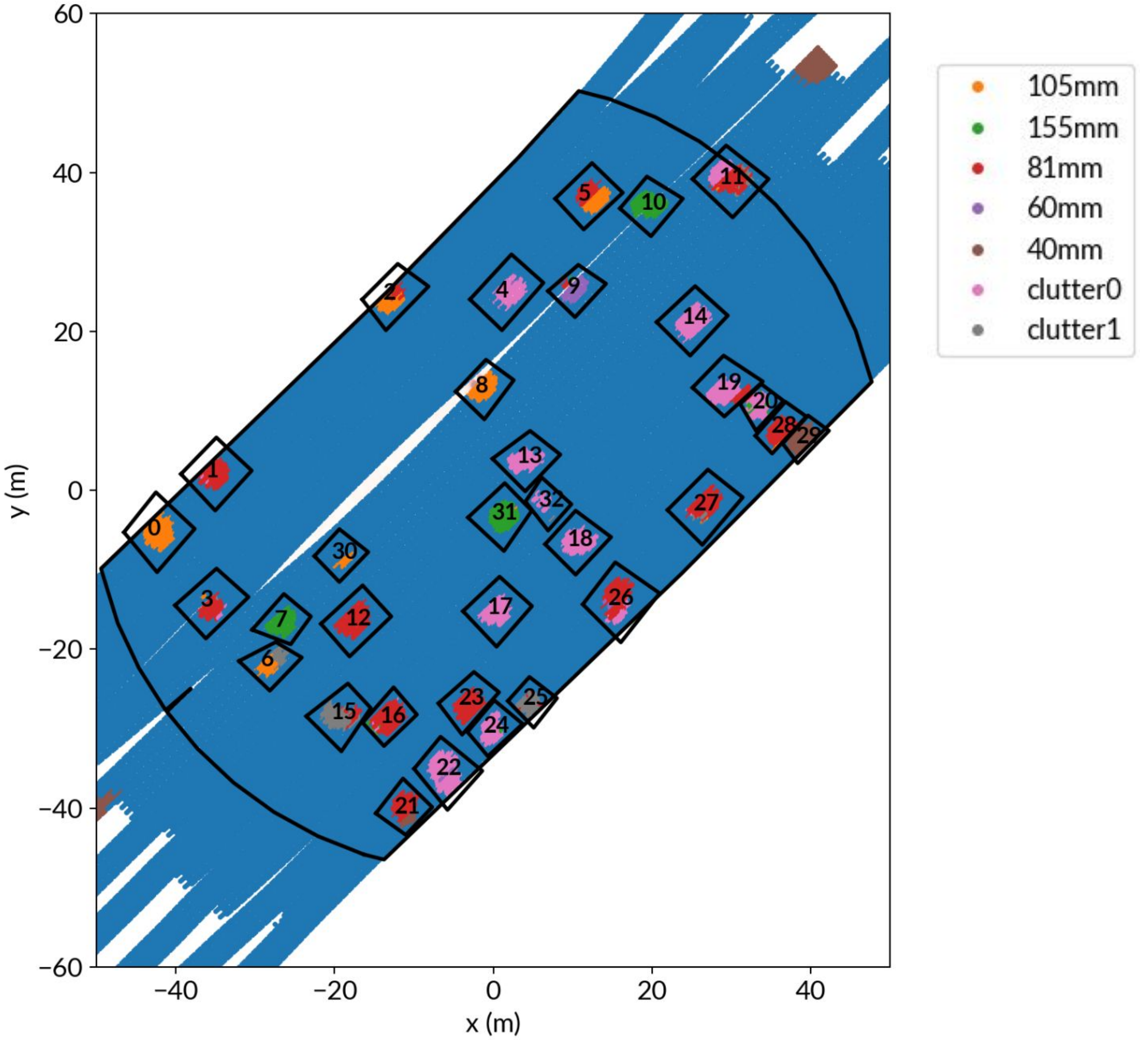
- Correctly predicted clutter
- Did not miss any UXO
- Classified to closest object included in training set

Blindgrid 2021
Sequim Bay



Blindgrid 2021
Sequim Bay

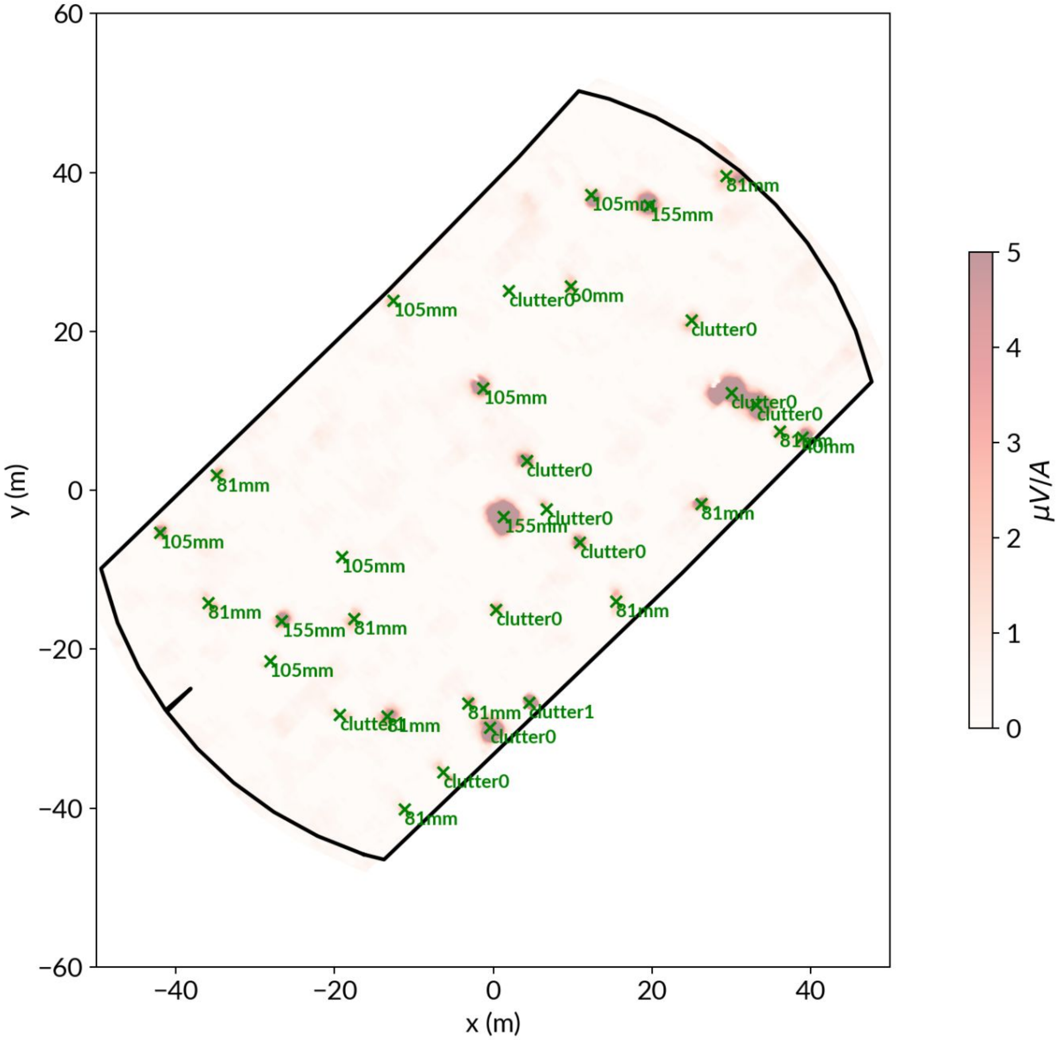
CNN classification output



Blindgrid 2021
Sequim Bay

Predicted labels

rank	label	prob.	dig
1	40mm	0.74	1
2	105mm	0.66	1
3	81mm	0.60	1
⋮			
32	clutter0	0.55	0
33	clutter0	0.61	0



Concluding remarks:

- A CNN with image segmentation architecture was successfully used to classify UXOs from marine EM data
- Some limitations:
    - CNN is relatively sensitive to effectiveness of seawater response removal
    - Objects used to generate synthetic data should be close to the objects on the field (CNNs perform poorly when extrapolating)
    - Full inputs needed (if one receiver or transmitter is missing, we skip that window)
- Future work:
    - Try different seawater response removal or training with seawater response
    - Explore ways to share information between different acquisition lines

Concluding remarks:

- A CNN with image segmentation architecture was successfully used to classify UXOs from marine EM data
- Some limitations:
  - CNN is relatively sensitive to effectiveness of seawater response removal
  - Objects used to generate synthetic data should be close to the objects on the field (CNNs perform poorly when extrapolating)
  - Full inputs needed (if one receiver or transmitter is missing, we skip that window)
- Future work:
  - Try different seawater response removal or training with seawater response
  - Explore ways to share information between different acquisition lines

Thank you!

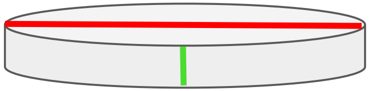
Jorge Lopez-Alvis jlalvis@eoas.ubc.ca

Clutter design

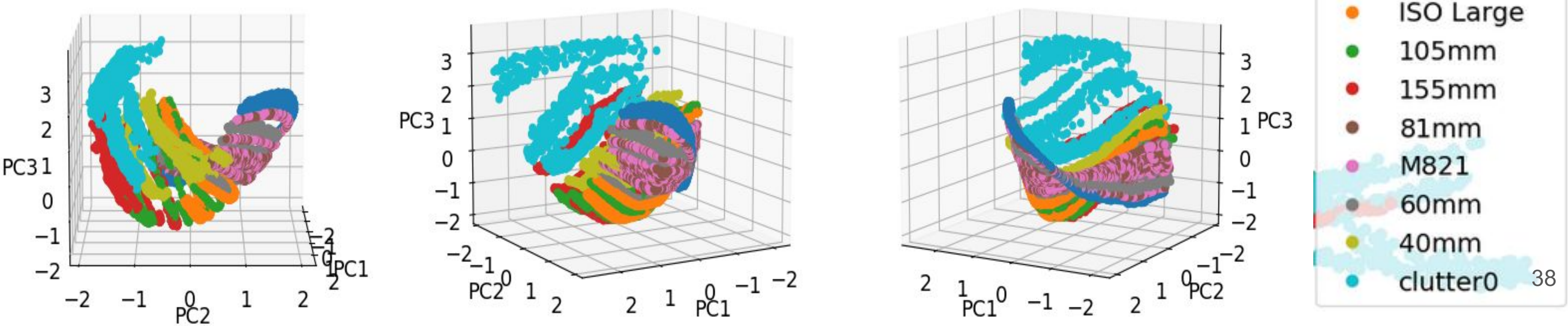
L1 and L2

L3

disk

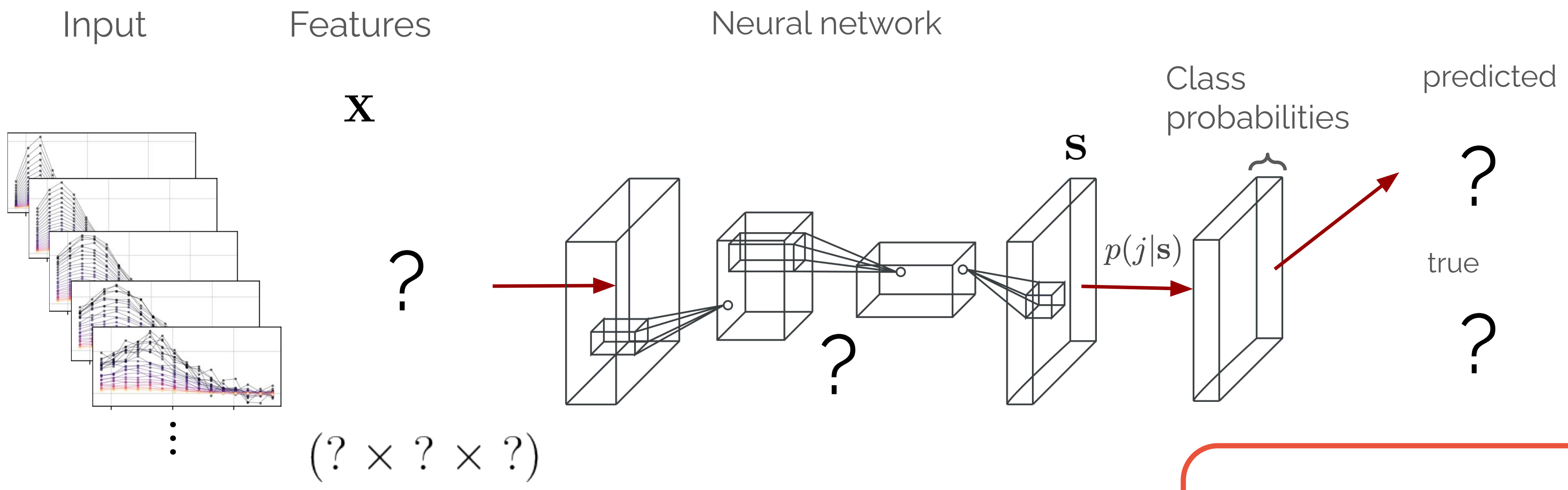


PCA was helpful to decide whether clutter objects are very close to UXOs:



Convolutional Neural Networks

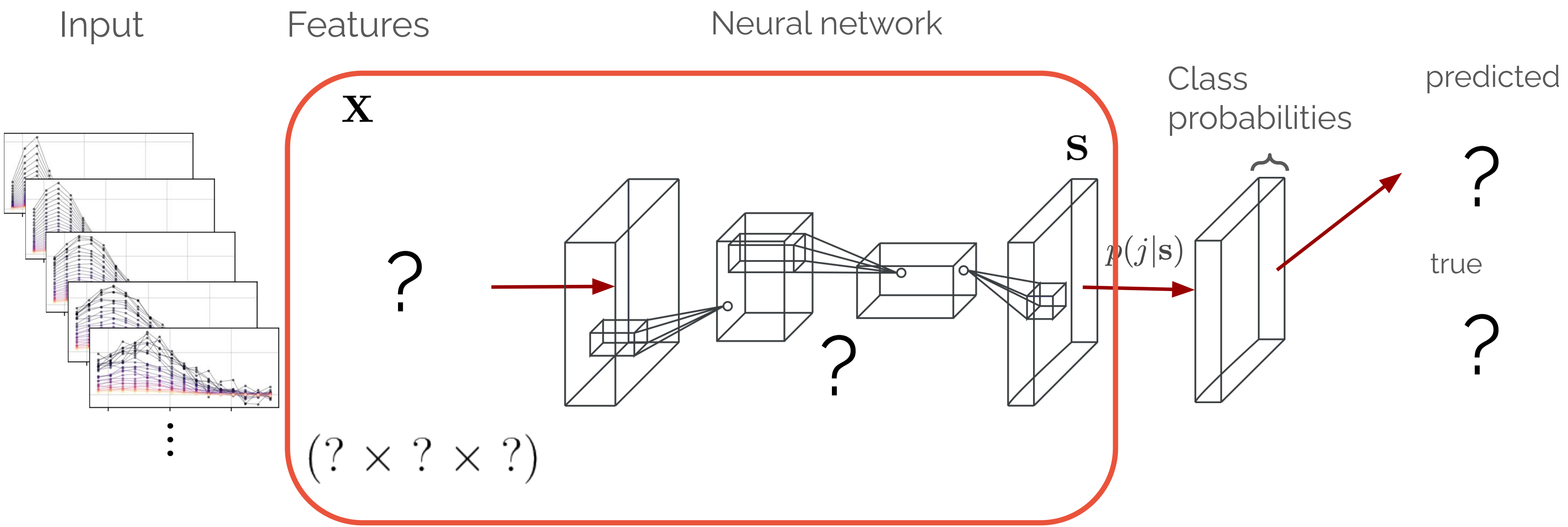
How do we translate these things to the UXO classification problem?



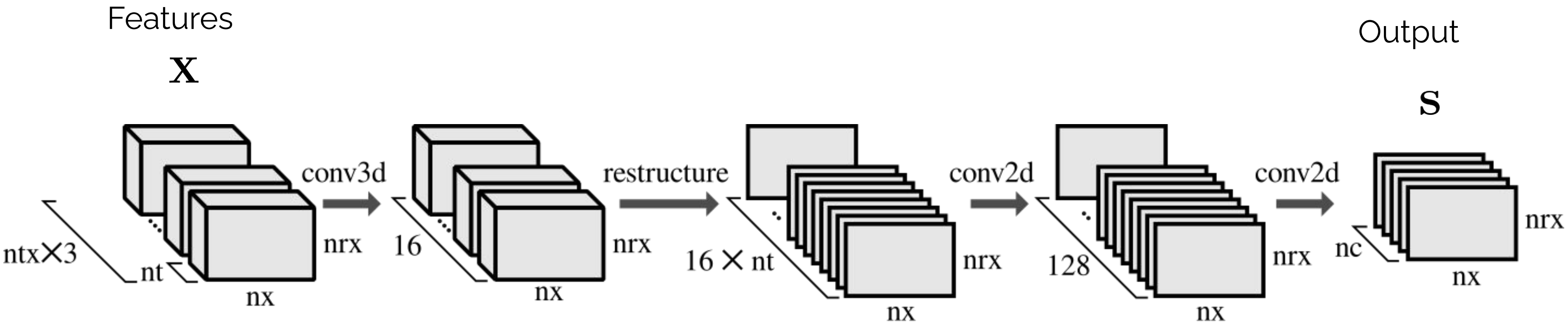
Training with synthetic data (using UXO library)

Convolutional Neural Networks

How do we translate these things to the UXO classification problem?



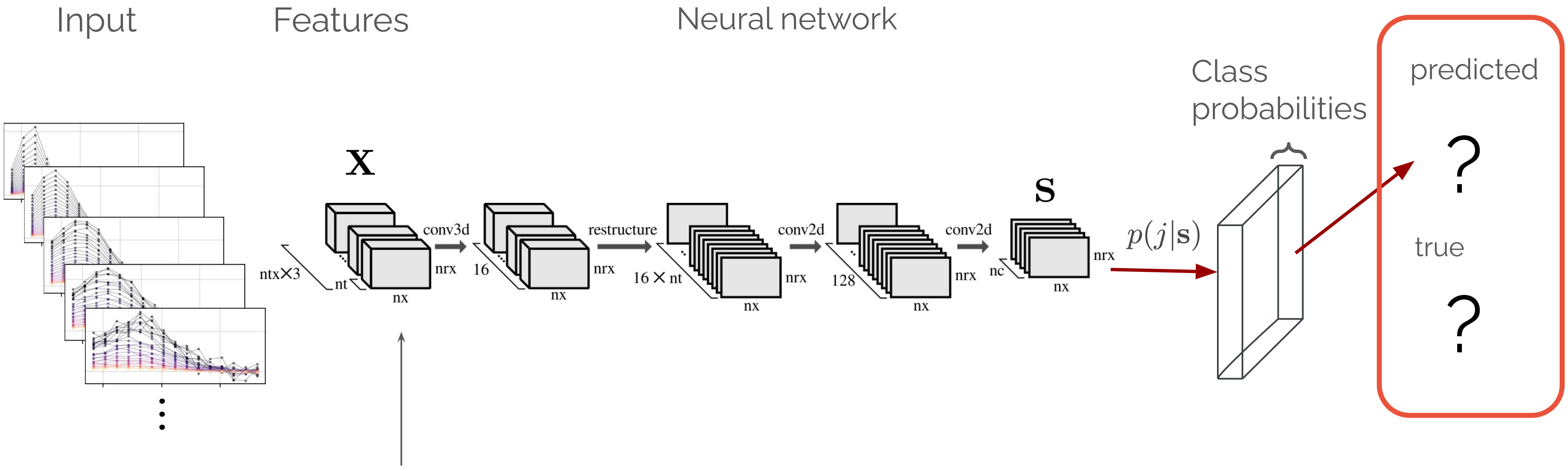
CNN - image segmentation architecture



ntx - number of transmitters
nrx - number of receiver cubes
nt - number of time channels
nx - number of positions in window
nc - number of classes

Convolutional Neural Networks

How do we translate these things to the UXO classification problem?



$$(nx \times nrx \times nt \times (ntx \times 3))$$