Machine learning for the classification of unexploded ordnance (UXO) from electromagnetic data

Lindsey J. Heagy[1], Douglas W. Oldenburg[2], Fernando Pérez[1], Laurens Beran[3]

[1]University of California Berkeley, [2]University of British Columbia, [3]Black Tusk Geophysics

Unexploded ordnance (UXO): A global problem

Definition: a munition that was armed, fired and remains unexploded

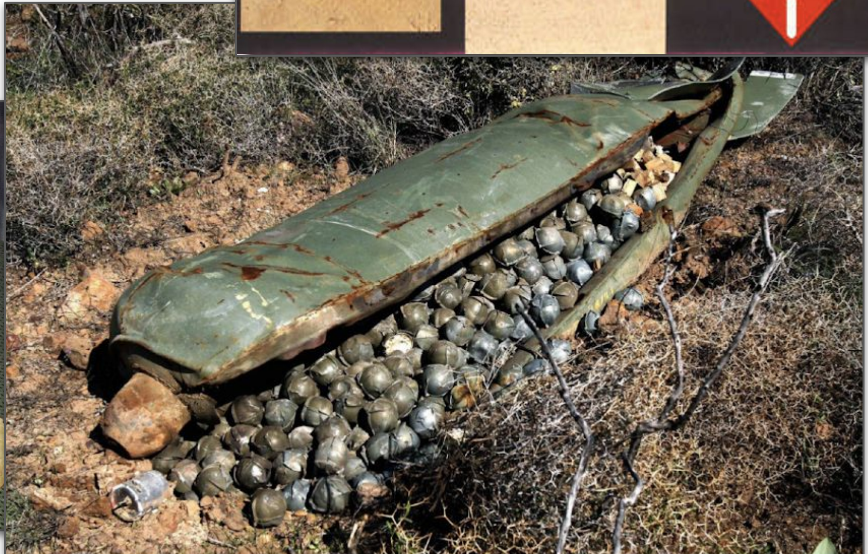
Sources:

- Regions of military conflict
- Munitions and bombing ranges
- Avalanche control

Countries significantly impacted by UXO

Various types of UXO

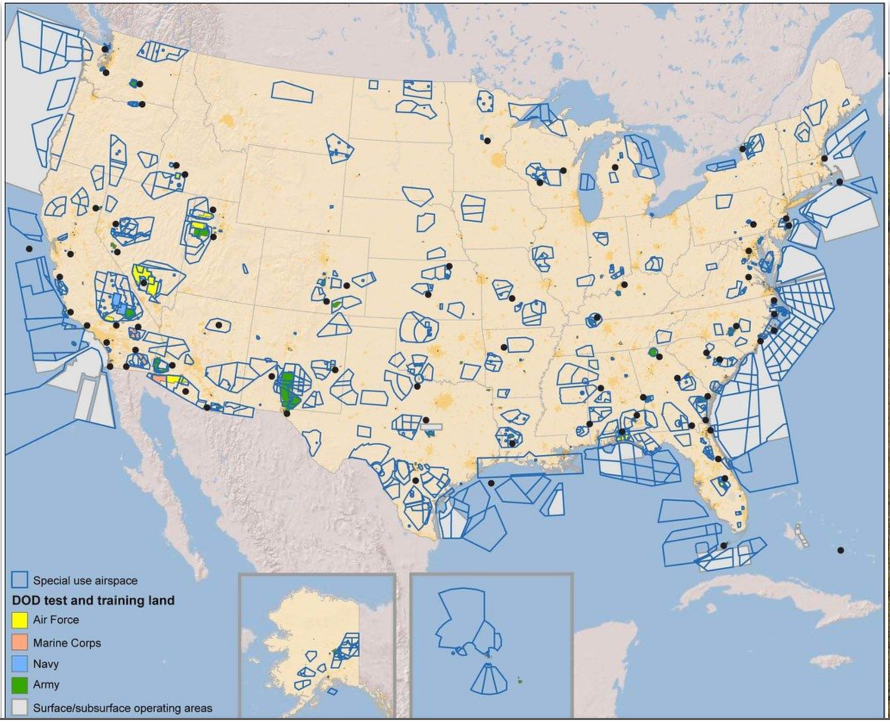
- Landmines
- Bombs
- Bombies (from cluster bombs)
- Rocket-propelled grenades (RPG)
- Hand-held grenades
- Mortars

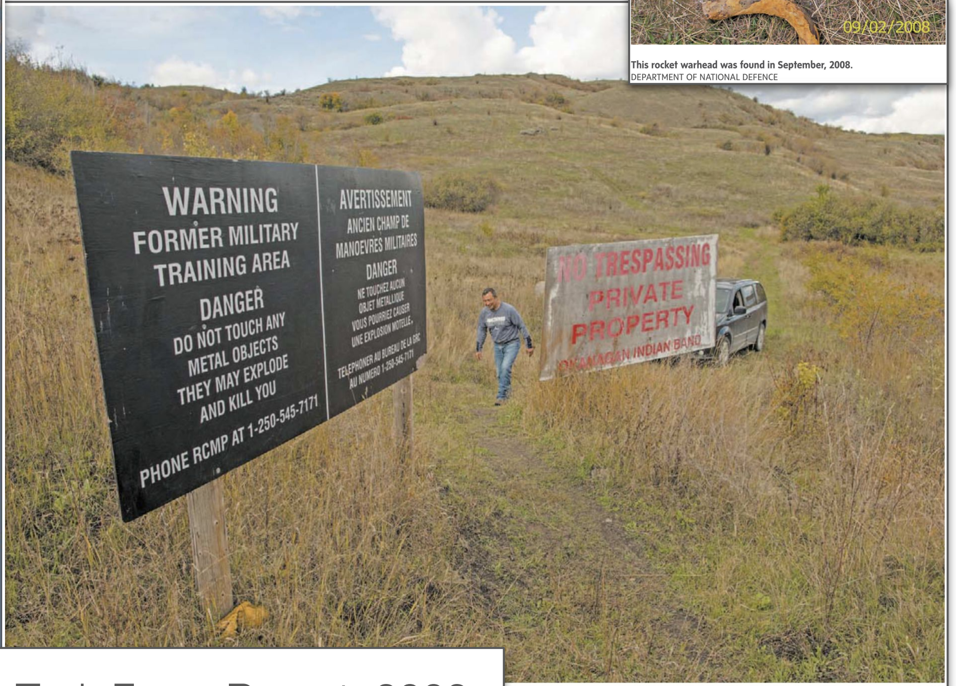


In the USA

?

This rocket warhead was found in September, 2008.
DEPARTMENT OF NATIONAL DEFENCE



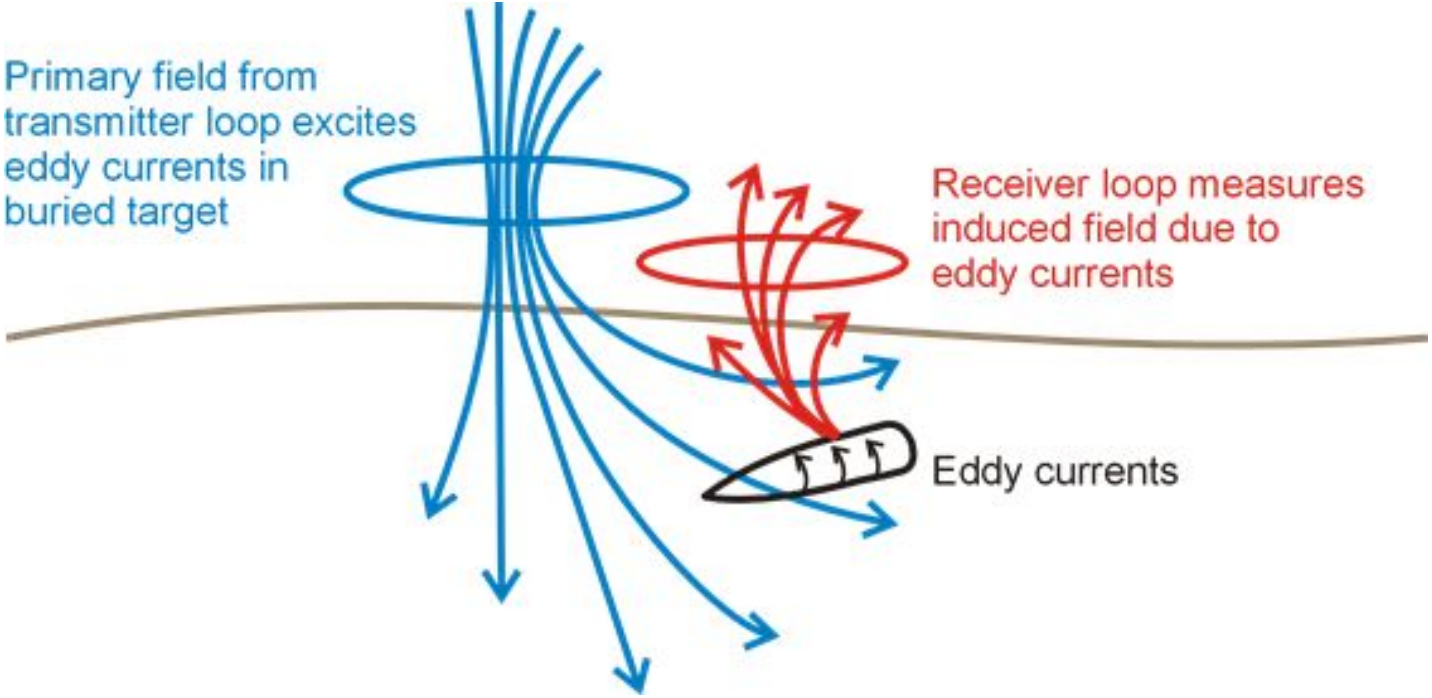


e presence of UXO. JEFF BASSETT/THE GLOBE AND MAIL

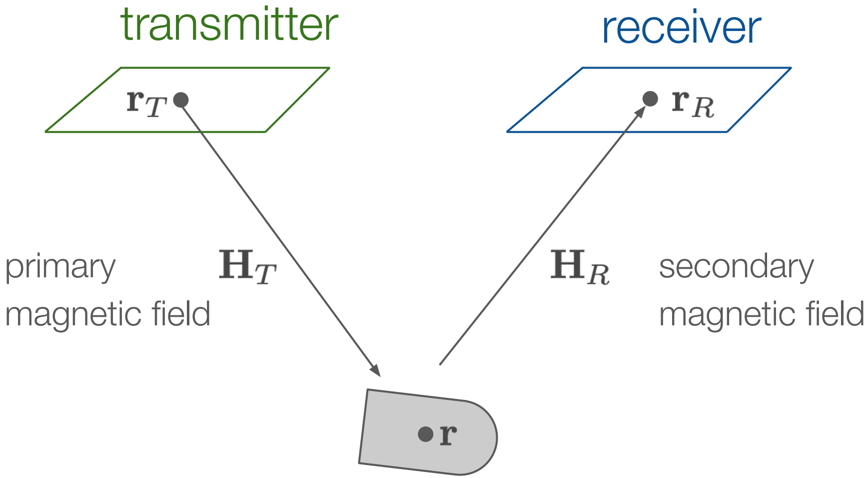
US Department of Defence UXO Task Force Report, 2003

The UXO cleanup problem is a very large-scale undertaking involving 10 million acres of land at some 1400 sites. Estimated clean-up cost of current UXOs is tens of billions of dollars.

Electromagnetics



Modelling the electromagnetic response: UXO



Approximate Physics Model

data

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

rotation matrix

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

polarization matrix

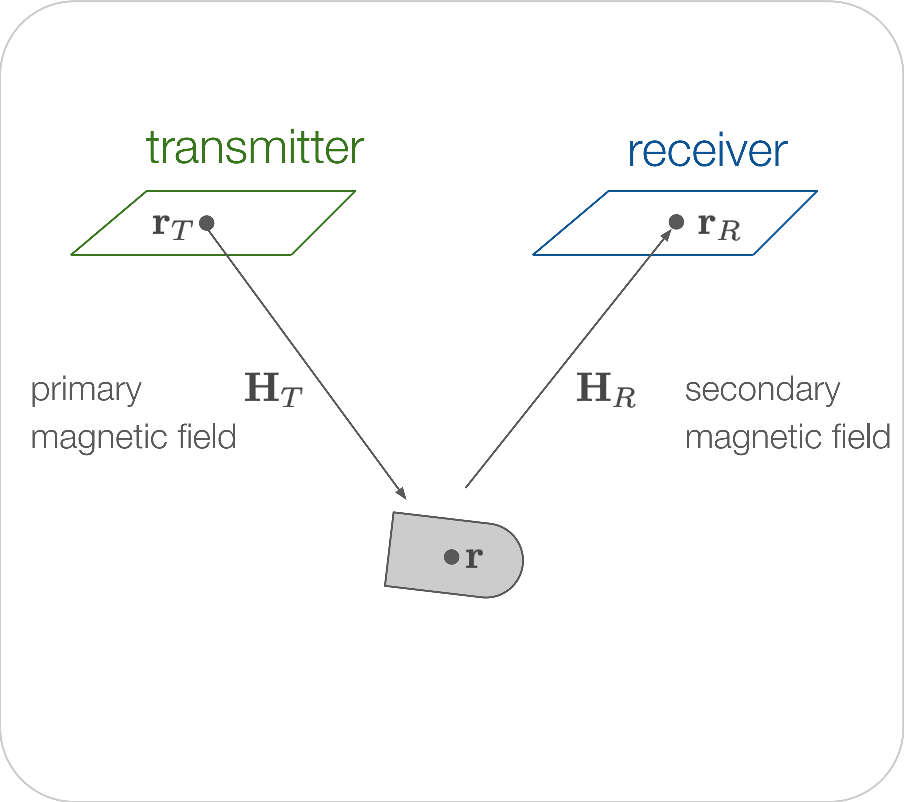
Euler angles

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

Unknowns

(x, y, z)	(ϕ, θ, ψ)	(L_1, L_2, L_3)
UXO location	orientation	polarizations

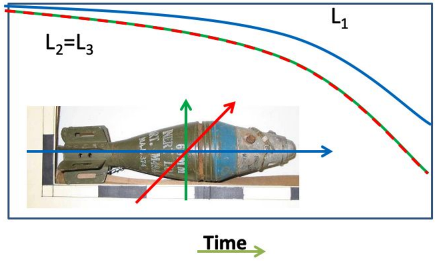
Modelling the electromagnetic response: UXO



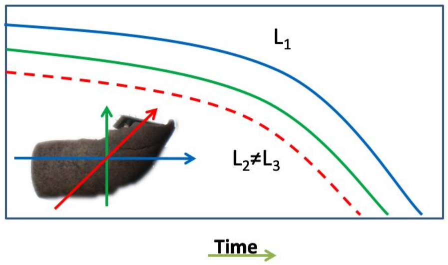
UXO generally distinguished by:

- large amplitude, slow-decaying primary (L1) polarizability
- equal secondary polarizabilities (L2=L3)

UXO



Not UXO



Survey and Data

UltraTEM system:

Black Tusk GEOPHYSICS

- 5 transmitters
- 11 receivers (3-component)
- 27 time channels

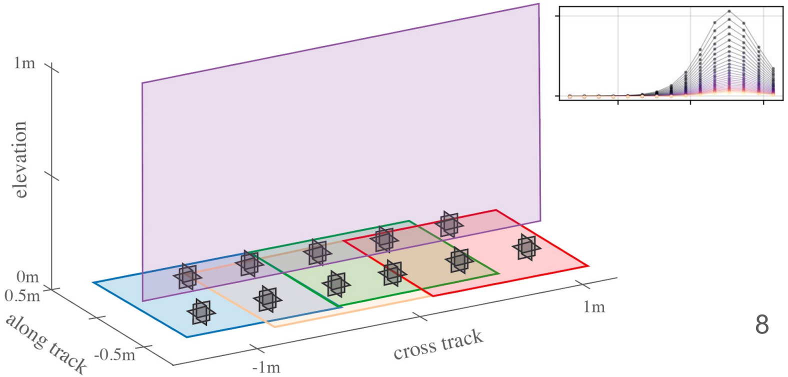
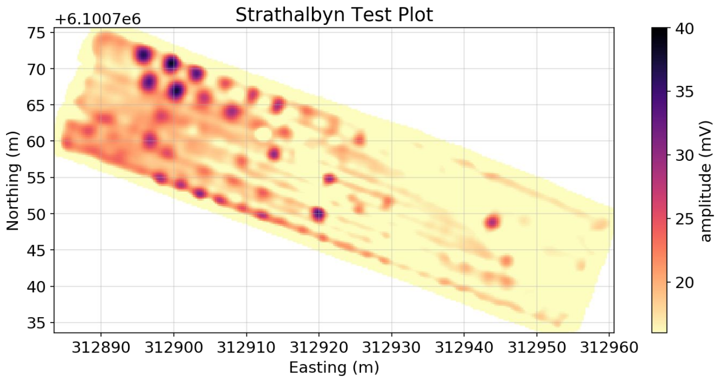
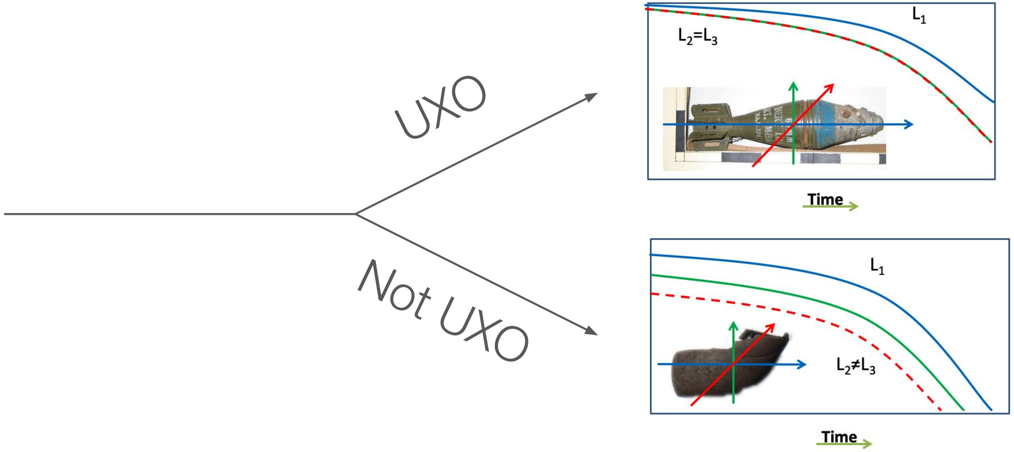


Image source: http://www.btgeophysics.com/UltraTEMBasicDescriptionv2.pdf

Current methodology

Digital Geophysical Map → Target picking → Cued Interrogation → Feature estimation → Quality Control → Classification

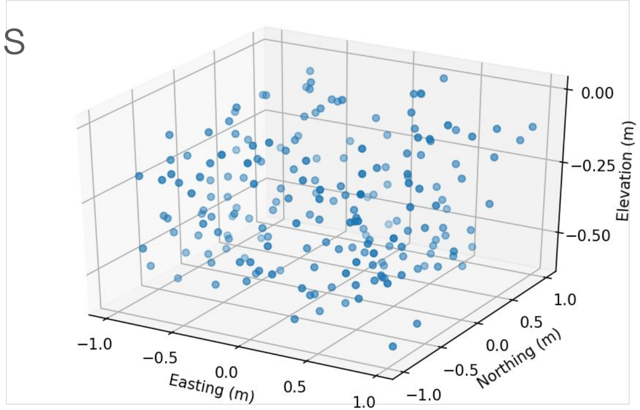




Current methodology

Digital Geophysical Map → Target picking → Cued Interrogation → Feature estimation → Quality Control → Classification

(1) Populate region of interest with random seed locations



(2) For a each seeded location:

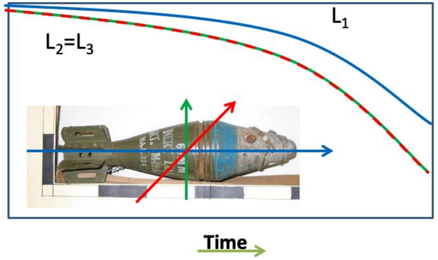
- Inversion 1: estimate

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

- Evaluate data fit, remove points with poor fit
- Inversion 2: decompose and isolate $\mathbf{L}(t)$

(3) For each $\mathbf{L}(t)$: Fingerprint with library of known ordnance

(4) Make a dig list



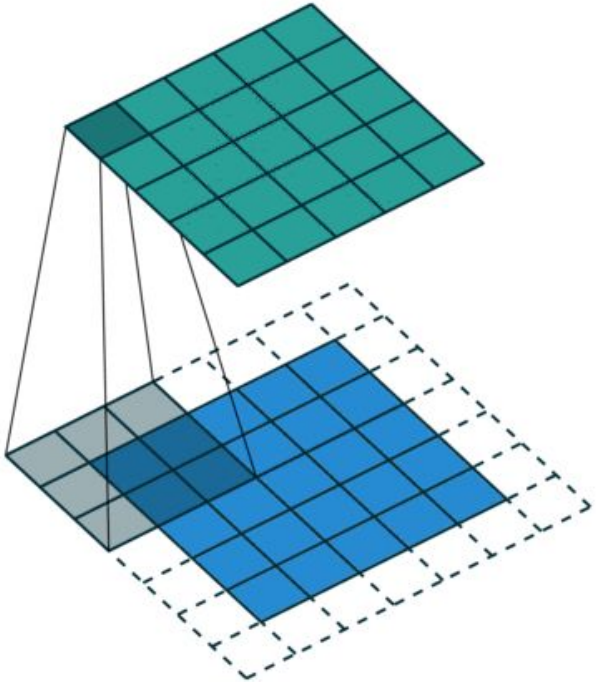
Candidate for convolutional neural networks?

Features of EM data for UXO detection / classification:

- Available library of ordnance objects with polarizations
- Access to labeled field data sets

Convolutional neural networks

- Workhorse in deep learning
- Convolutional filters look at spatial / temporal features in the data



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Neural networks: a brief overview

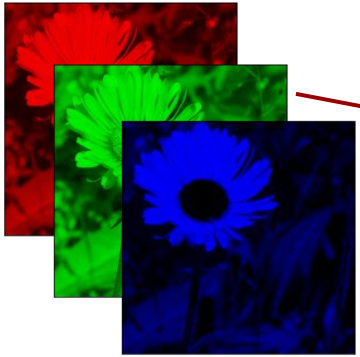
Supervised classification problem

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

Input

Features

$\mathbf{X}$



$(nx \times ny \times 3)$

Neural network

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

Convolutional layers

$$\begin{aligned}
\mathbf{Y}_1 &= \sigma(\mathbf{K}_0 \mathbf{X} + b_0) \\
\mathbf{Y}_2 &= \sigma(\mathbf{K}_1 \mathbf{Y}_1 + b_1) \\
&\vdots \\
\mathbf{Y}_N &= \sigma(\mathbf{K}_{N-1} \mathbf{Y}_{N-1} + b_{N-1})
\end{aligned}$$

Linear classifier

$$\mathbf{s} = \mathbf{W}\mathbf{Y}_N + \mathbf{b}$$

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

Class probabilities

p_0	p_1	p_2	p_3
flower	...	...	dog

$(n \text{ inputs} \times n \text{ classes})$

$$p(j|\mathbf{s}) = \frac{e^{s_j}}{\sum_k e^{s_k}}$$

class

all classes

Follows notation of Haber and Ruthotto

Neural networks: a brief overview

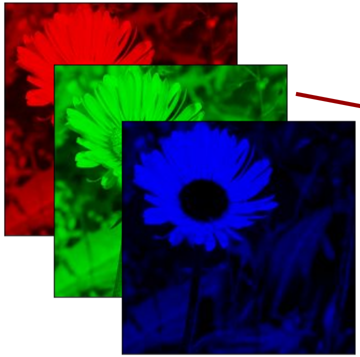
Training

define an optimization problem to estimate network parameters

Input

Features

$\mathbf{X}$



$(nx \times ny \times 3)$

Neural network

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

all trainable parameters

Convolutional layers

$$
\begin{aligned}
\mathbf{Y}_1 &= \sigma(\underline{\mathbf{K}_0}\mathbf{X} + \underline{b_0}) \\
\mathbf{Y}_2 &= \sigma(\underline{\mathbf{K}_1}\mathbf{Y}_1 + \underline{b_1}) \\
&\vdots \\
\mathbf{Y}_N &= \sigma(\underline{\mathbf{K}_{N-1}}\mathbf{Y}_{N-1} + \underline{b_{N-1}})
\end{aligned}
$$

Linear classifier

$$\mathbf{s} = \underline{\mathbf{W}}\mathbf{Y}_N + \underline{\mathbf{b}}$$

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

Class probabilities

predicted

p_0	p_1	p_2	p_3

true

q_0	q_1	q_2	q_3

Measure: cross entropy loss

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

Follows notation of Haber and Ruthotto

Neural networks: a brief overview

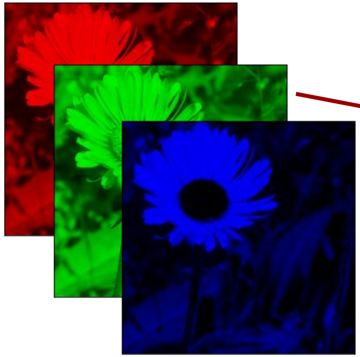
Translating to the UXO problem

What are the inputs?

Input

Features

$\mathbf{X}$



$(nx \times ny \times 3)$

Neural network

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

Convolutional layers

$$\mathbf{Y}_1 = \sigma(\underline{\mathbf{K}_0}\mathbf{X} + \underline{b_0})$$
$$\mathbf{Y}_2 = \sigma(\underline{\mathbf{K}_1}\mathbf{Y}_1 + \underline{b_1})$$
$$\vdots$$
$$\mathbf{Y}_N = \sigma(\underline{\mathbf{K}_{N-1}}\mathbf{Y}_{N-1} + \underline{b_{N-1}})$$

Linear classifier

$$\mathbf{s} = \underline{\mathbf{W}}\mathbf{Y}_N + \underline{\mathbf{b}}$$

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

Class probabilities

predicted

p_0	p_1	p_2	p_3

true

q_0	q_1	q_2	q_3

Measure: cross entropy loss

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

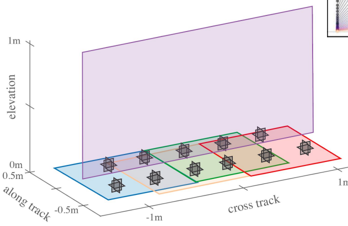
Follows notation of Haber and Ruthotto

Neural networks: a brief overview

Translating to the UXO problem

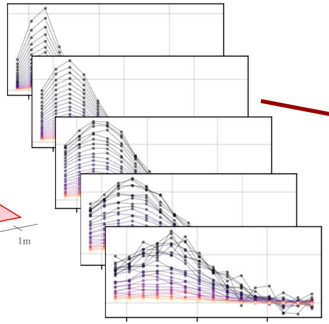
What are the inputs?

Input



Features

$\mathbf{X}$



$\vdots$

$(nx \times nt \times 165)$

Neural network

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

Convolutional layers

$$\mathbf{Y}_1 = \sigma(\underline{\mathbf{K}_0}\mathbf{X} + \underline{b_0})$$
$$\mathbf{Y}_2 = \sigma(\underline{\mathbf{K}_1}\mathbf{Y}_1 + \underline{b_1})$$
$$\vdots$$
$$\mathbf{Y}_N = \sigma(\underline{\mathbf{K}_{N-1}}\mathbf{Y}_{N-1} + \underline{b_{N-1}})$$

Linear classifier

$$\mathbf{s} = \underline{\mathbf{W}}\mathbf{Y}_N + \underline{\mathbf{b}}$$

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

Class probabilities

predicted

p_0	p_1	p_2	p_3

true

q_0	q_1	q_2	q_3

Measure: cross entropy loss

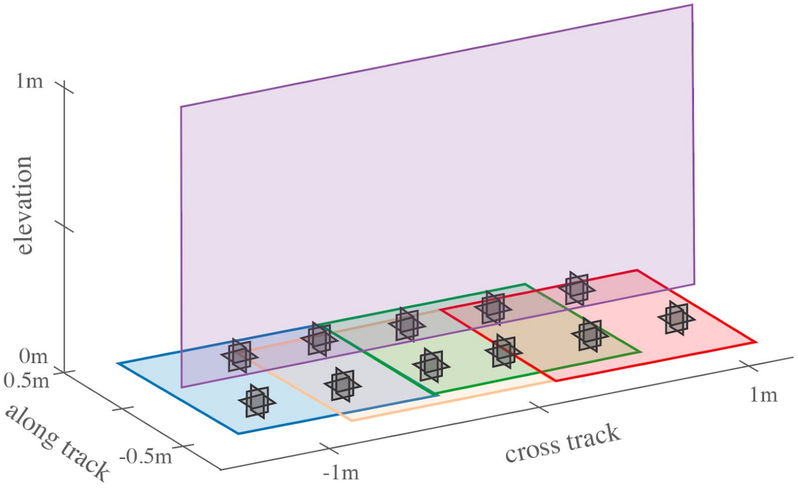
$$\min_\theta \phi = -\sum q_j \log(p_j)$$

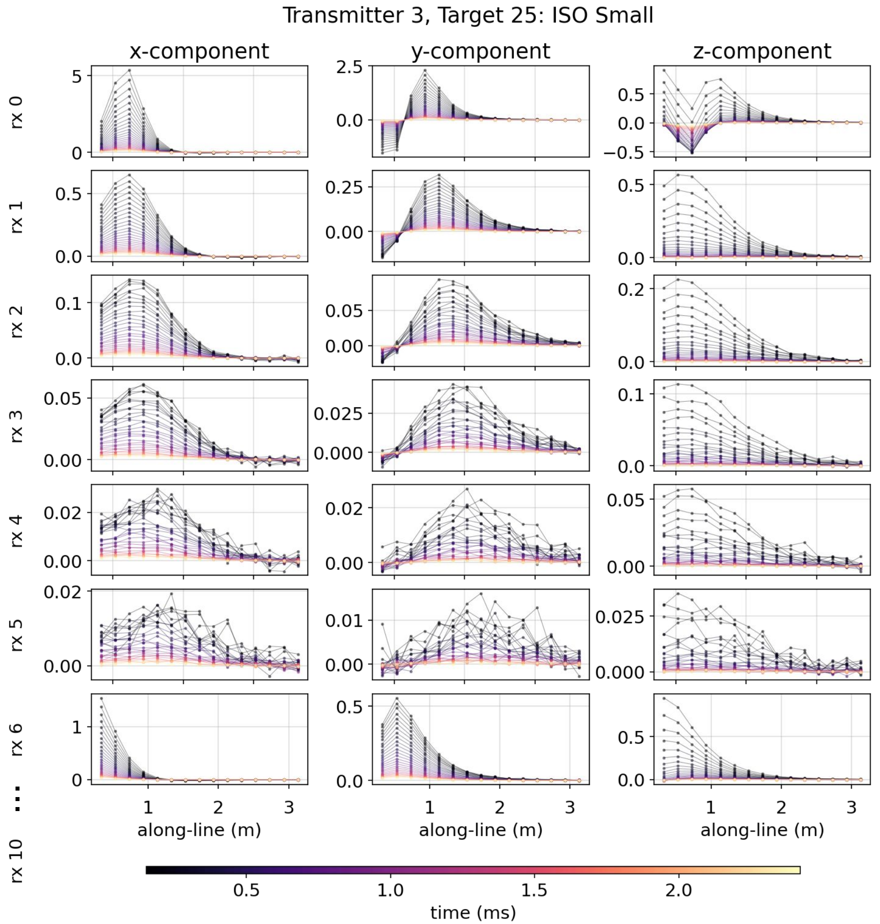
Follows notation of Haber and Ruthotto

CNN Setup: What are the data

UltraTEM data:

- 165 transmitter - receiver combinations
- 27 time-channels
- # soundings in a given along-line distance (3m window, 15 locations)

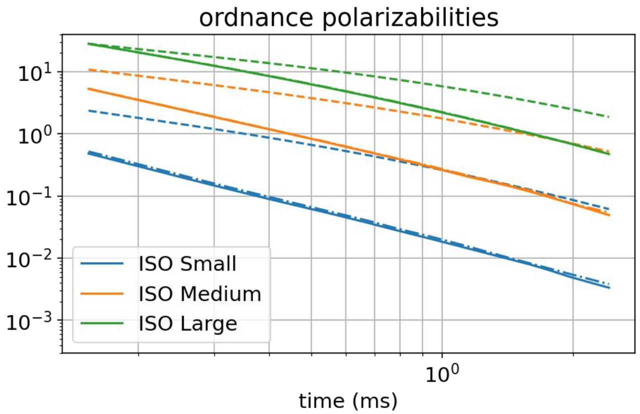




Training data

Target Classes:

- Background
- Small ISO
- Medium ISO
- Large ISO
- Clutter (20mm, spherical objects)

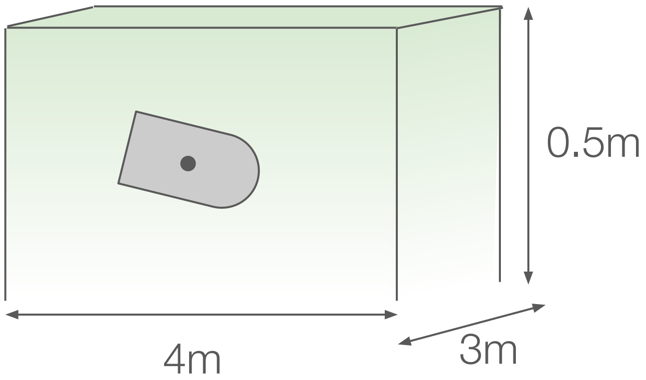


Thousands of realizations

- Training: ~8k
- Test & Validation: ~1k

For each realization, randomly assign:

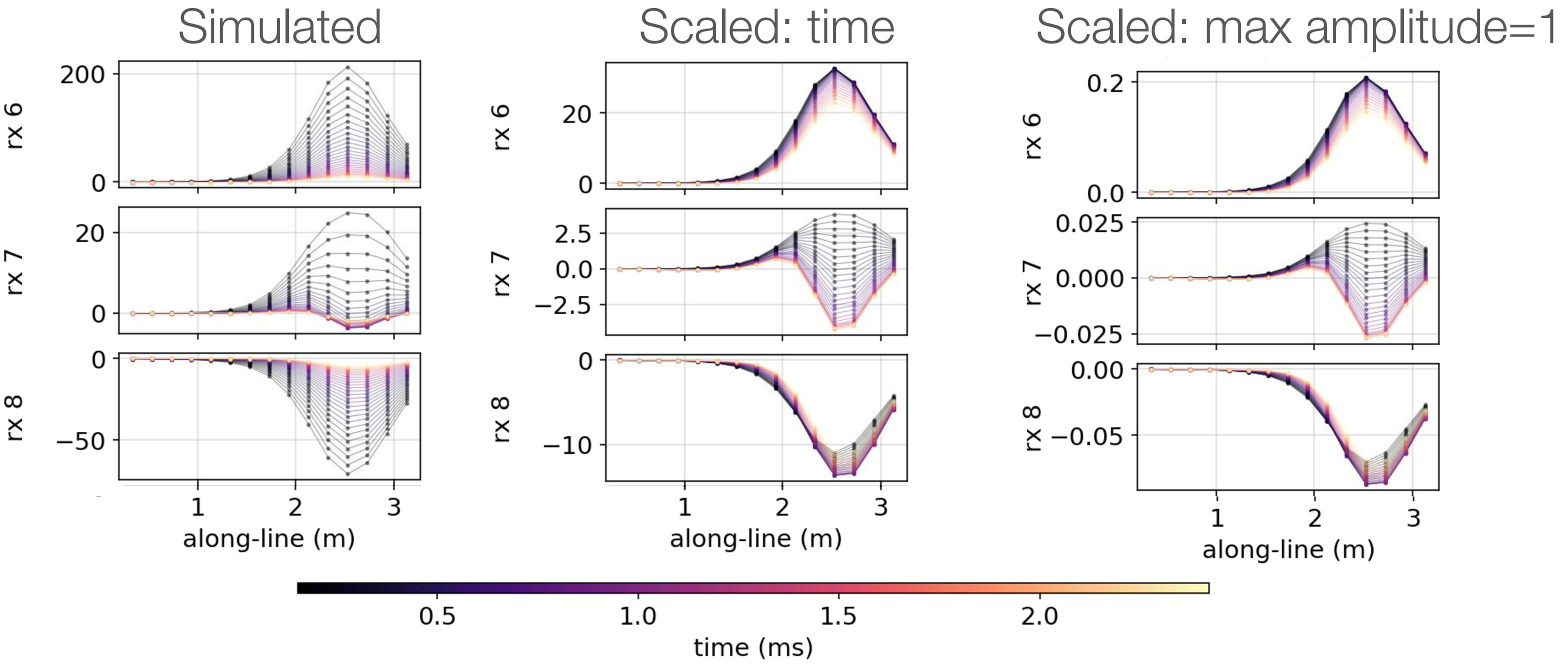
- Target class
- Location (x, y, z)
- Orientation (ϕ, θ, ψ)
- Noise level $\varepsilon = a\frac{1}{t}\mathcal{N}(0, 1)$



Data normalizations

2 steps:

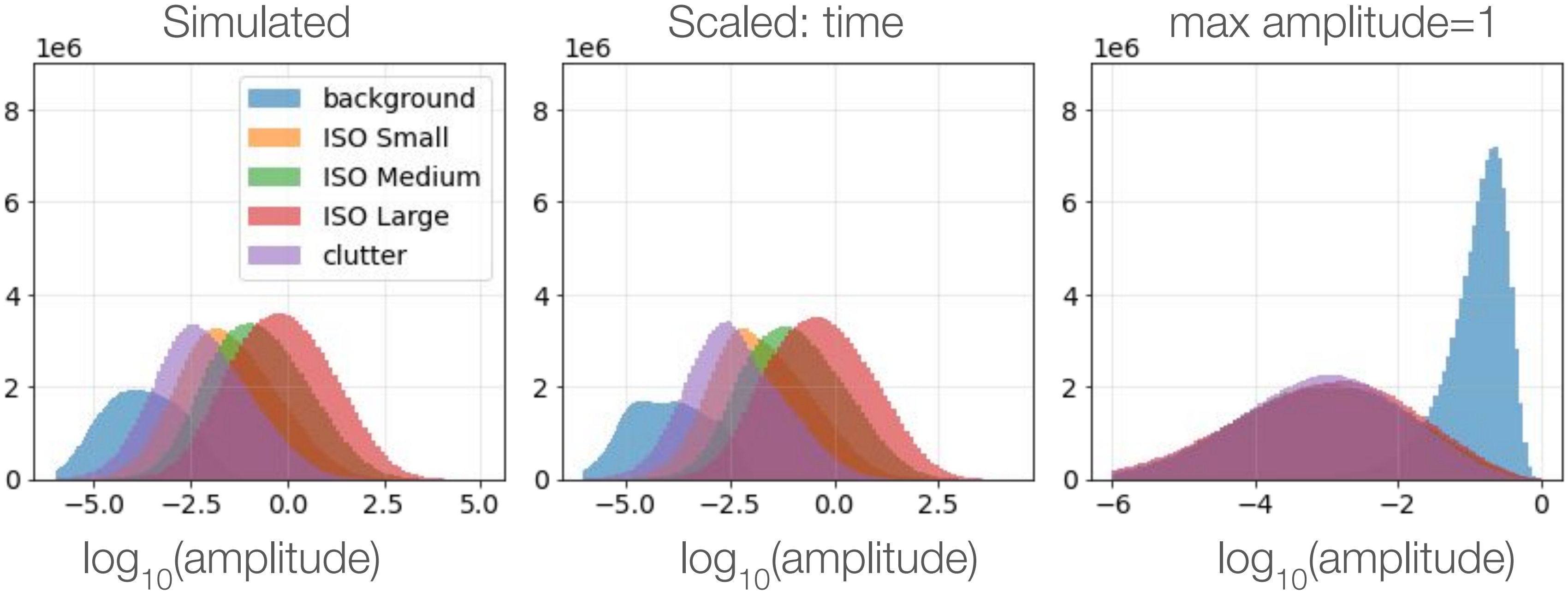
- Scale each data as a function of time for each channel (multiply by t)
- Normalize amplitude across all channels to have maximum amplitude 1



Data normalizations

2 steps:

- Scale each data as a function of time for each channel (multiply by t)
- Normalize amplitude across all channels to have maximum amplitude 1



Training the CNN

Convolutional Network

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

outputs

network parameters

input features

Class probabilities

$$p(j|\mathbf{s}) = \frac{e^{s_j}}{\sum_k e^{s_k}}$$

class

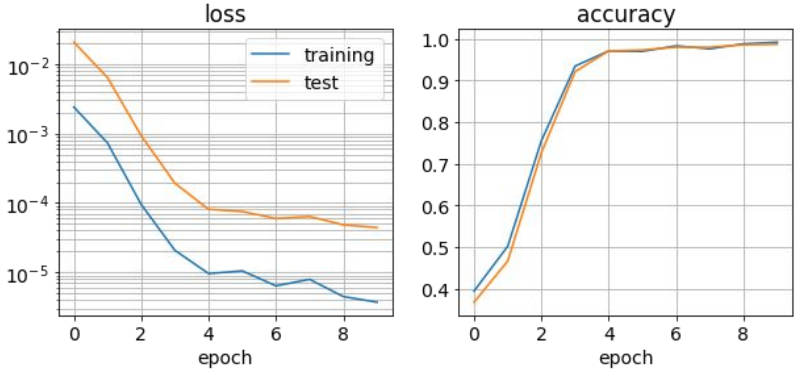
all classes

Classification

$$c_{\text{pred}} = \underset{j}{\operatorname{argmax}}\, p(j|\mathbf{s})$$

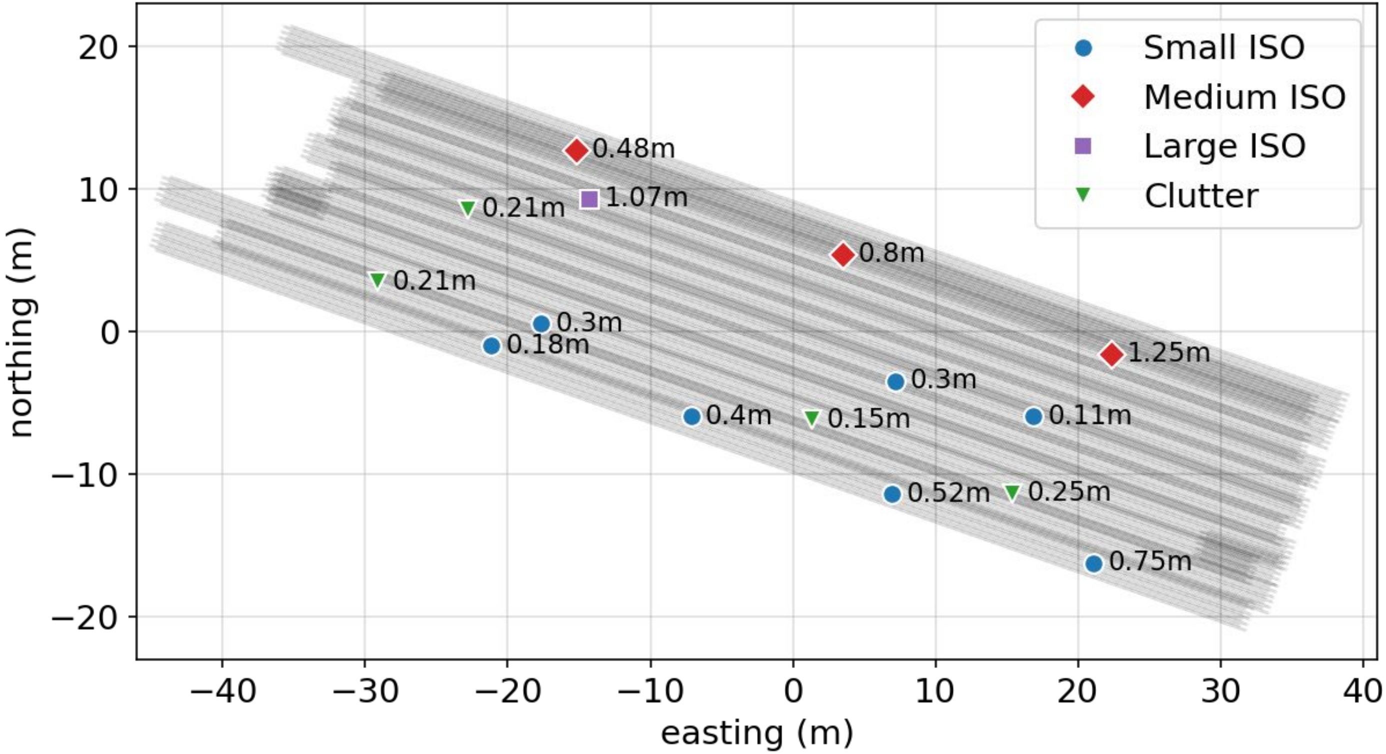
Training problem (use stochastic gradient descent)

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

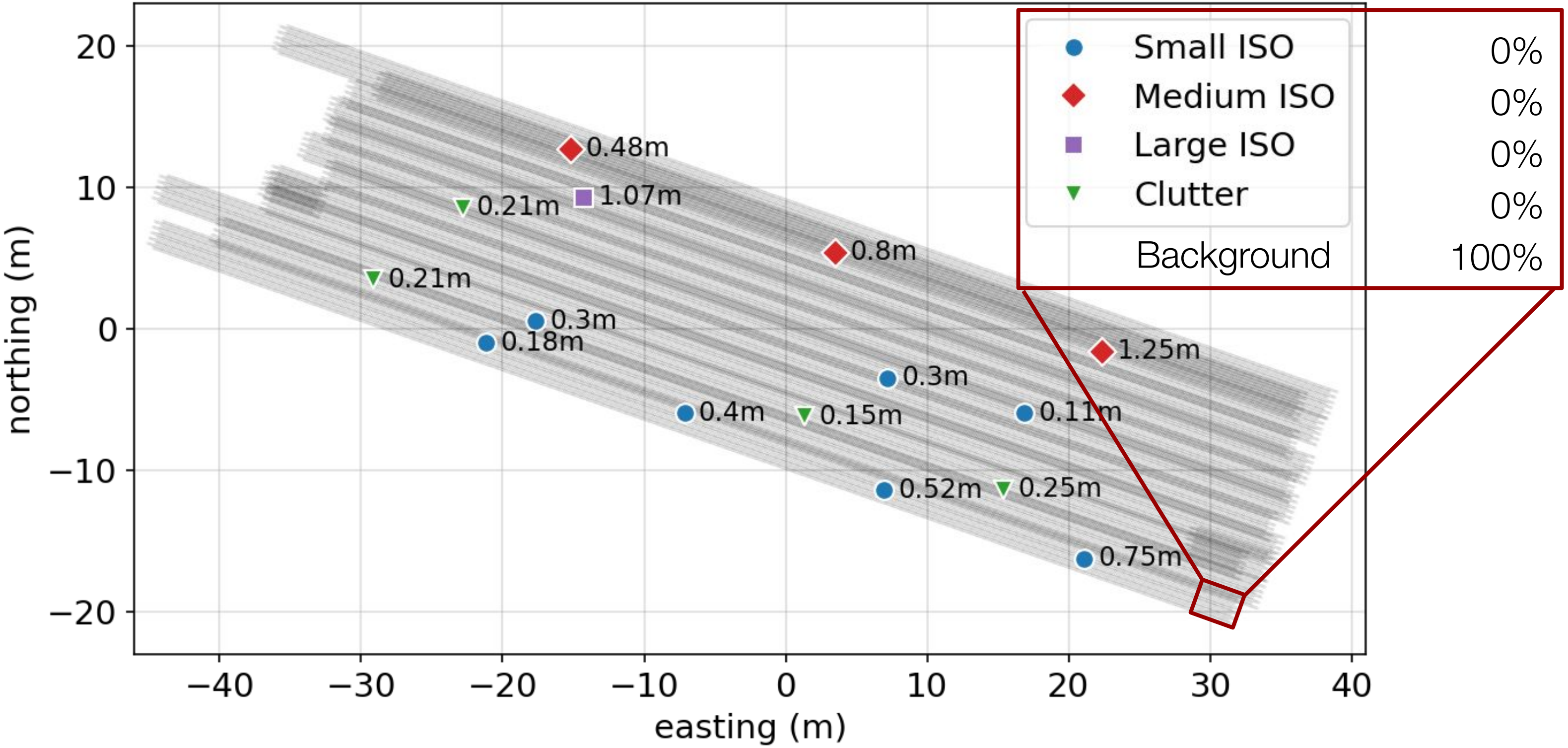


PyTorch

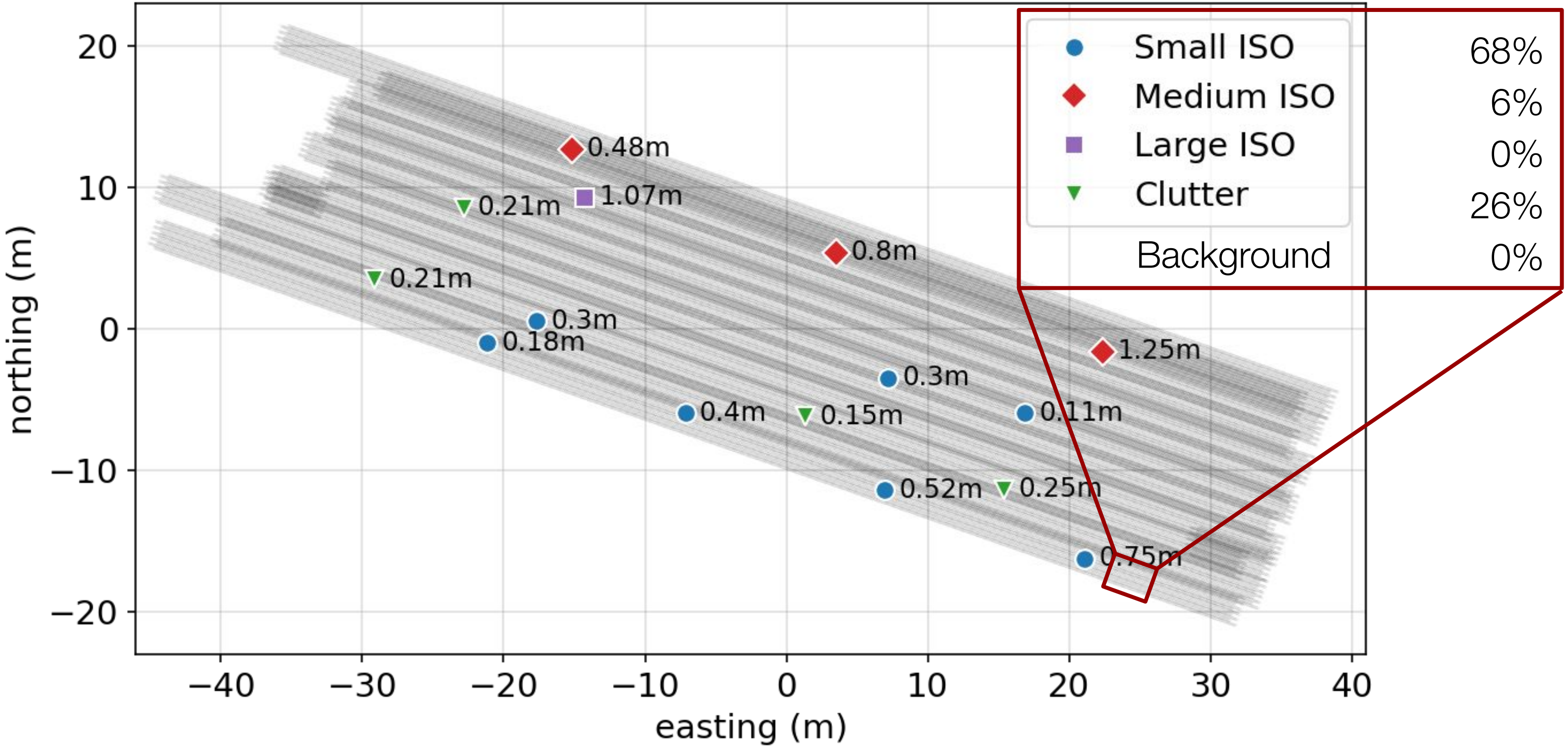
Synthetic survey



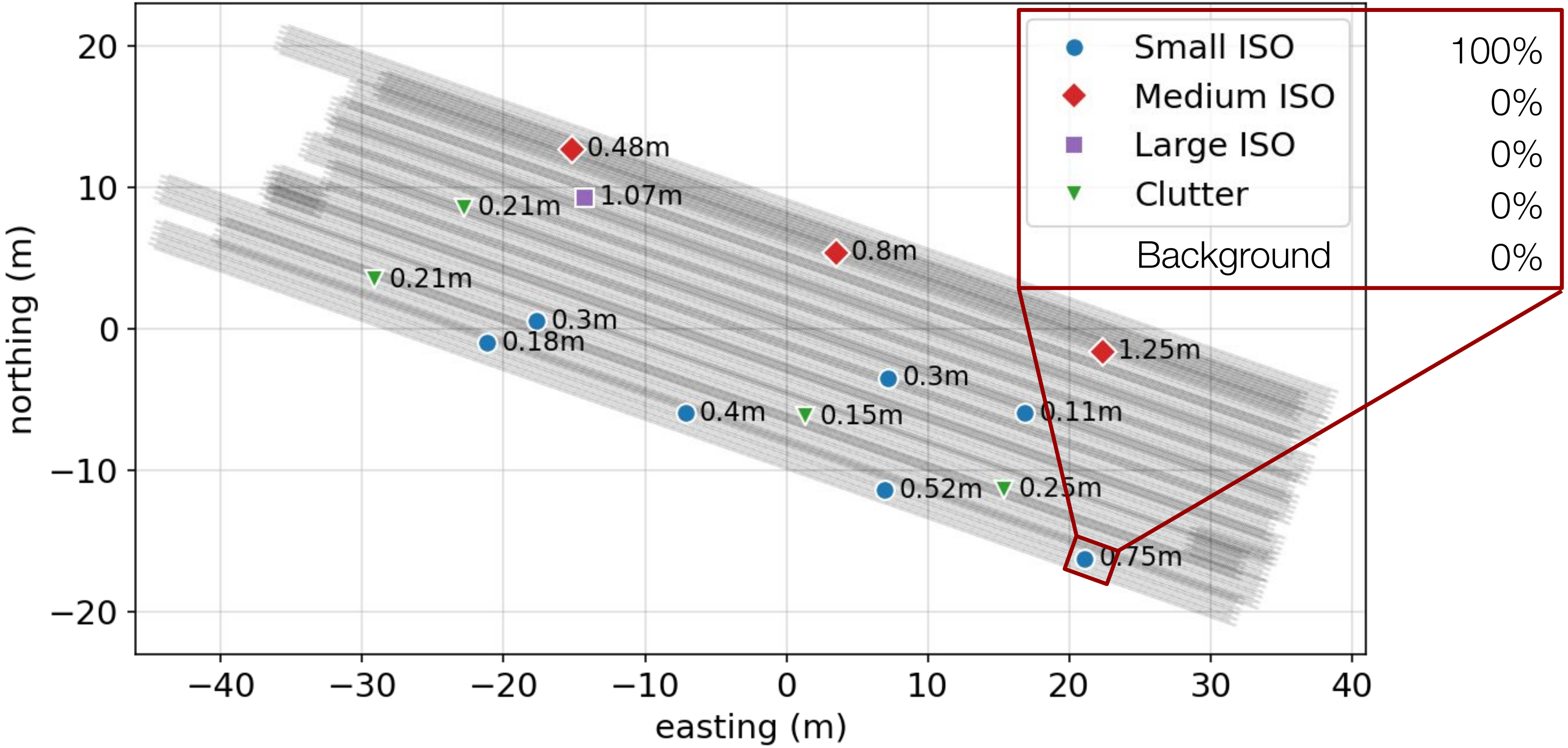
Synthetic survey



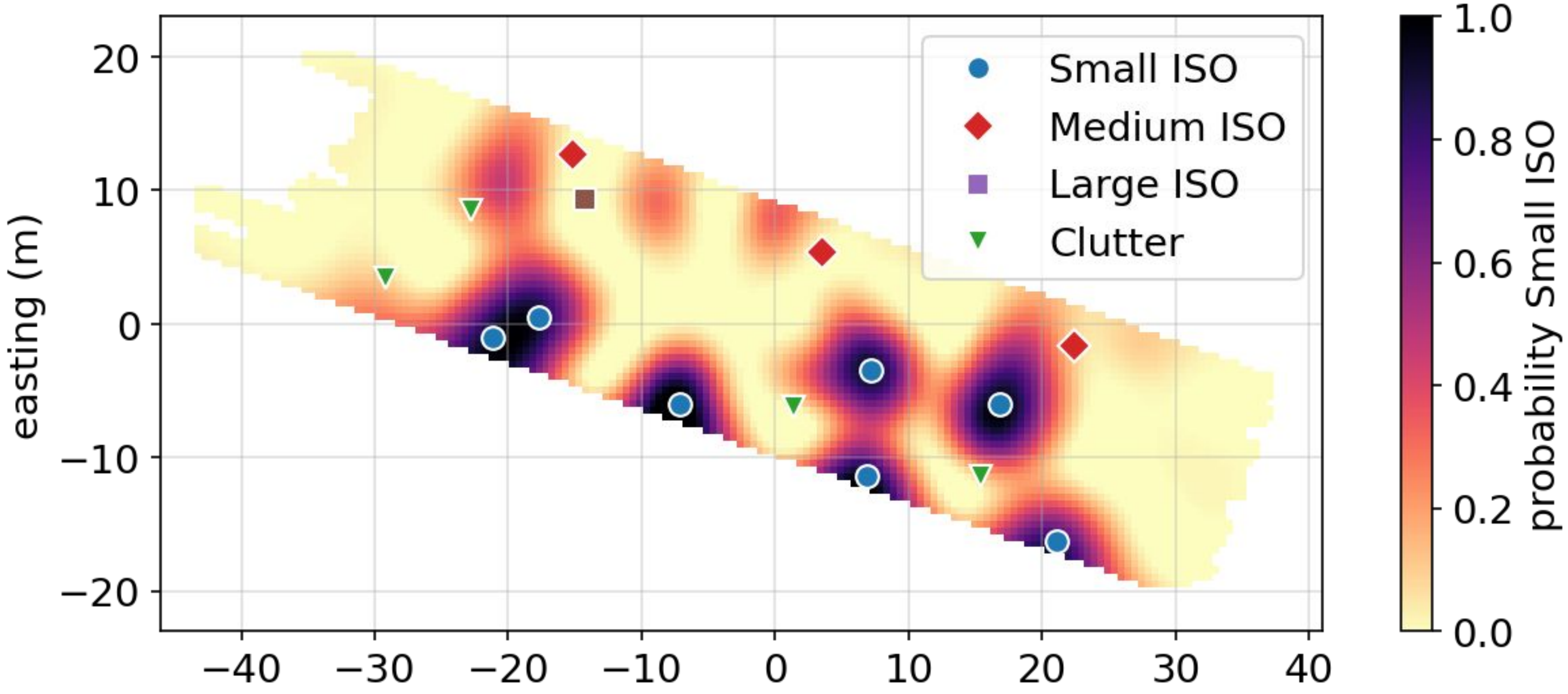
Synthetic survey



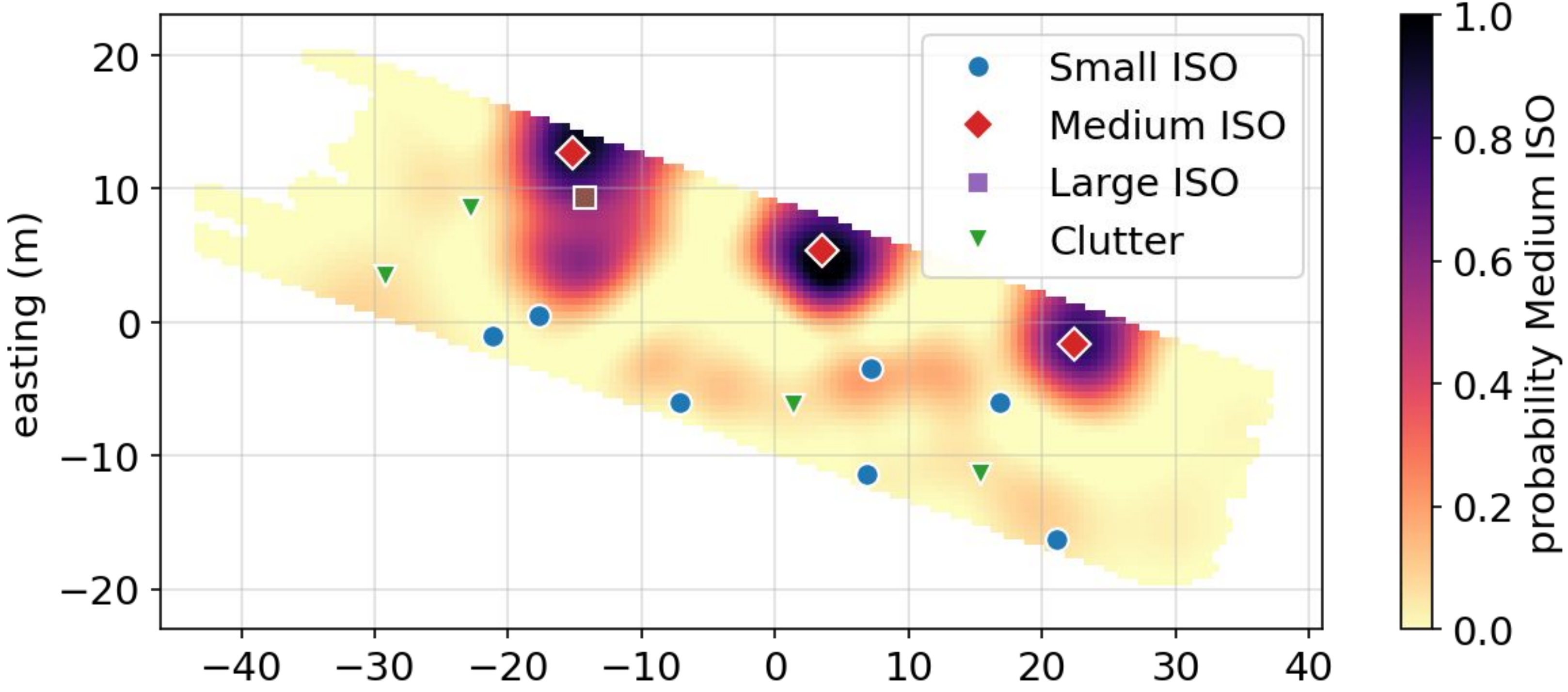
Synthetic survey



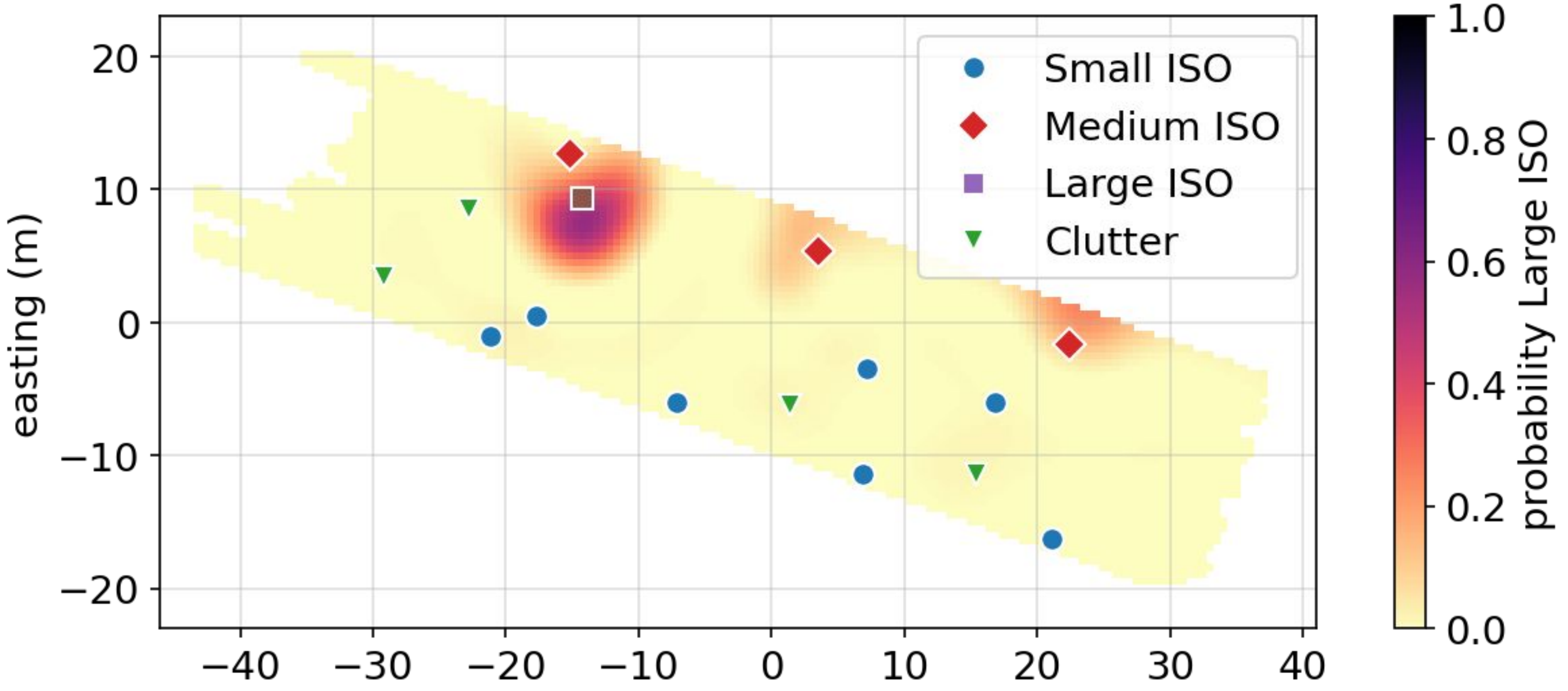
Synthetic example: Small ISO



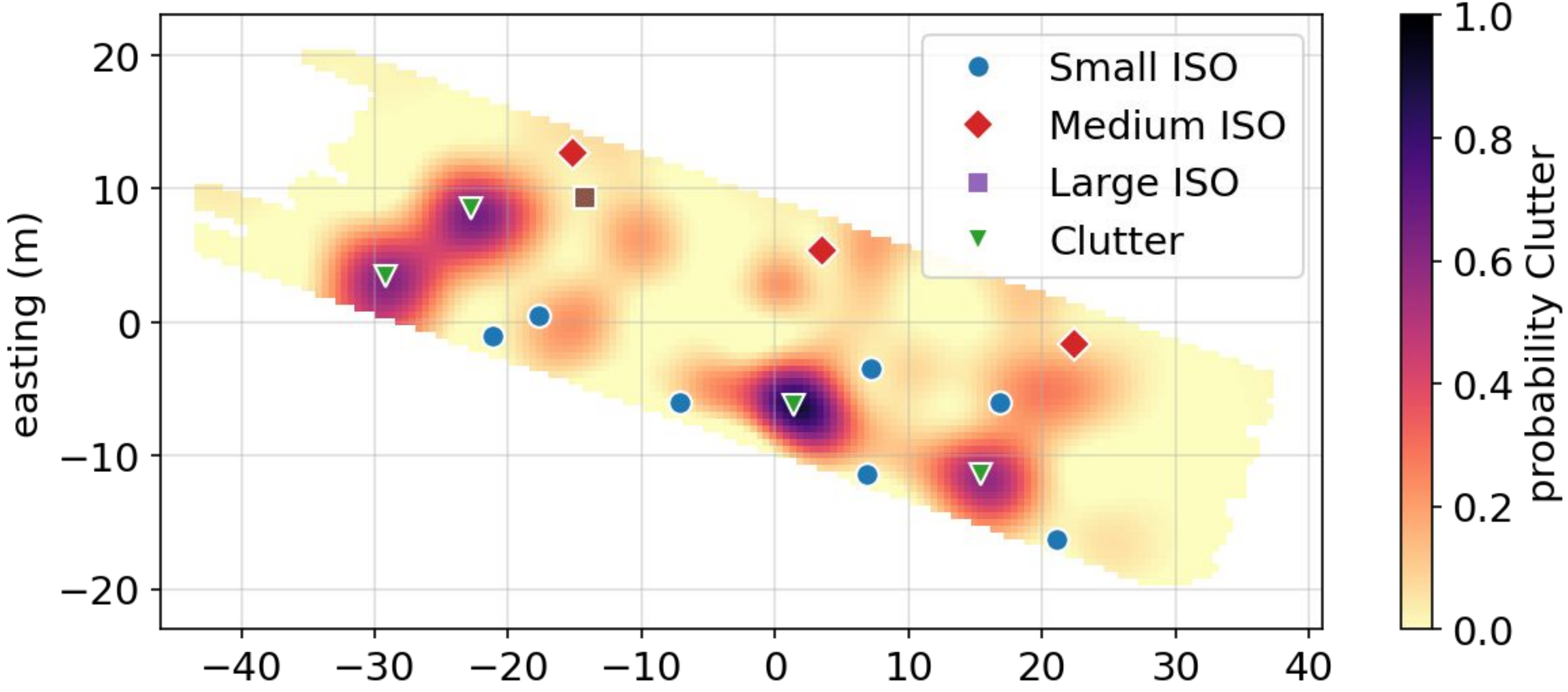
Synthetic example: Medium ISO



Synthetic example: Large ISO



Synthetic example: clutter



UXO: Open avenues & next steps

Results demonstrate:

- Proof-of-concept for classification of UXO directly from data

Questions and next steps

- Constructing clutter model: what else should we include?
- Multi-object scenarios
- Exploring behaviour in challenging geologic settings (e.g. magnetic soils)
- Neural Network architecture:
  - Input data: other features to input?
  - Regularization or parameterization of network parameters?
- **How to integrate information from ML with traditional analysis to make clearance more effective and less costly?**

Thank you

@lheagy

lheagy@berkeley.edu

@lindsey_jh

https://github.com/simpeg-research/heagy-et-al-2020-uxo-seg