An Approach to Semantic Segmentation of Radar Sounder Data Based on Unsupervised Random Walks and User-Guided Label Propagation

Jordy Dal Corso, Lorenzo Bruzzone, *Fellow, IEEE*

***Abstract*—Radar sounders (RS) are utilized for the analysis of subsurface of Earth and other planets. Data acquired from RS can be processed to obtain radargrams, which are 2-D arrays containing the backscattered echo power received by the radar after sending pulses toward the surface. The study of radargrams offers crucial insights for the geological interpretation of the history of planets and for the monitoring of ice layers in glacial regions. Deep learning has emerged as a powerful tool for the automatic feature extraction and analysis of radargrams, yet they are still treated as conventional images. We propose a novel methodology for the semantic segmentation of radar sounder data based on a two step approach. The rationale of this methodology is exploiting the spatial horizontal correlation that exists among radargram features, which is an important property that distinguishes these data from standard images. In the first step, an encoder is trained in an unsupervised way, exploiting random walks to learn meaningful representations of sequential features within radargrams. In the second step, few reference labelled samples allows the model to propagate the labels to the full radargram. We also introduce a metric to quantify the degree of horizontal correlation among features and we use it to find the grounding zone in coastal radargrams of polar areas. We test our methodology on two datasets obtained by the MCoRDS radar sounder and a dataset from the orbital radar sounder SHARAD and we discuss the very promising results.**

***Index Terms*—Radar sounder, random walks, unsupervised learning, label propagation, MCoRDS, SHARAD.**

NOMENCLATURE

n_T	Number of traces
n_S	Number of samples per trace
(w, h)	Dimension of patches
(o_w, o_h)	Pixel overlapping between adjacent patches
θ	Parameters of the encoder
M_θ	Parametrized encoder
R	Radargram
R_t	Radargram t-th column of dimension $T \times N$
$R_{t,n}$	Radargram patch of dimension $H \times W$
T	Number of columns
N	Number of patches per column
n_C	Number of classes
S	Segmentation map
S_t	Segmentation map of column t
$\mathcal{P}$	Set of all patches of dimension $H \times W$
n_F	Length of the encoder output vector
Z_t	Feature matrix of column t
V	Set of vertices of the graph
E	Edge matrix of the graph
P_t^{t+1}	Transition matrix between column t and $t+1$
X_t	State of the random walk at time t
p_0	Initial prob. dist. of the random walk
p_t	Prob. dist. of the random walk at the t-th step
$m(i)$	Mean of patch i of R_0
$w(i)$	Cross-entropy weight of walker i
I_n	Identity matrix of dimension n
A_t	Similarity matrix between column t and $t+1$
$A_{\text{knn},t}$	Similarity matrix after row-wise kNN masking
$A_{\text{bank},t}$	Similarity matrix after context bank concatenation
$A_{\text{mask},t}$	Similarity matrix after radius masking
$\bar{c}$	Target class for backward label propagation
k	Number of nearest neighbors in label propagation
c	Number of samples in the context bank
r	Radius for masked attention
τ	Softmax temperature
$\mathcal{D}_1$	MCoRDS1 dataset
$\mathcal{D}_2$	MCoRDS3 dataset
$\mathcal{D}_3$	SHARAD dataset

Jordy Dal Corso and Lorenzo Bruzzone are with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy.

I. INTRODUCTION

RADAR sounders (RS) have emerged as indispensable tools for studying the subsurface of Earth and other planets. The key mechanism of RS consists in sending electromagnetic pulses toward the surface of planets and processing the received echoes. Pulses have relatively low frequency (3-200 MHz) and this allows them to penetrate up to kilometers in the subsurface, offering crucial insights on hidden structures that are completely inaccessible to other remote sensing instruments. Received echoes contains information about the materials and dielectric discontinuities crossed by the pulses and allow experts to identify surface and subsurface features of interest such as buried craters, lava flows, ice layers and crevasses. The analysis of radar sounder data led to groundbreaking discoveries, such as the presence of liquid water on Mars [1], and is consistently helping in understanding the geological history of planetary bodies and the evolution of glacial areas on polar caps.

On Earth, airborne RS like the Multichannel Coherent Radar Depth Sounder (MCoRDS) [2], [3] have been instrumental in mapping the thickness and internal layering of ice sheets in remote and inaccessible regions such as Antarctica and Greenland. These airborne systems offer high-resolution subsurface data, providing researchers with a comprehensive understanding

of ice sheet dynamics and contributing to more accurate predictions of sea-level rise.

Planetary RS, such as the Mars Advanced Radar for Subsurface and Ionosphere Sounding (MARSIS) [4] onboard the European Space Agency (ESA) Mars Express mission, have provided valuable insights into the geological structure of Mars and its polar ice caps. Similarly, the Shallow Radar (SHARAD) instrument [5] aboard NASA's Mars Reconnaissance Orbiter (MRO) has mapped the Martian subsurface with a shallower pulse penetration and an higher range resolution than MARSIS. Another example of orbital RS is the Radar for Icy Moons Exploration (RIME) [6] on board of the Jupiter Icy Moon Explorer (JUICE), the ESA spacecraft that is planned to enter the orbit of Jupiter in 2031 and study its icy moons. With all these instruments operating and more to come, such as in the ESA EnVision mission to Venus, with the Subsurface Radar Sounder (SRS) on board [7], there is a growing need of automatic analysis tools for their data.

Consecutive echoes collected by RS can be aggregated to obtain products called *radargrams*, which arrange the backscattered echo power received by the radar instrument in a 2-dimensional matrix (3D in the case of a multi-polarization RS). The analysis of radargrams is essential for achieving the aforementioned scientific objectives and techniques such as layer tracking and semantic segmentation are instrumental for extracting meaningful information from these data products.

Radargram data analysis has undergone significant evolution over the years, driven by advancements in computational techniques and the increasing availability of radar datasets. In the early stages, researchers primarily relied on statistical methods to extract meaningful information from radar echoes [8], [9], [10], [11], [12]. The advent of deep learning revolutionized RS data analysis by offering powerful models for automatic feature learning [13], [14], lately leveraging on pre-trained architectures [15] and transfer learning [16].

Despite the latest effort in creating models that could provide reliable segmentation maps and tracked layers, practitioners still tend to carry out the analysis of radargrams in a manual way. This is mainly due to the fact that deep learning models are usually tied to the data they have been trained on and, in particular, to the ground truth labels used. In the case of radar sounders, datasets vary significantly due to distinct acquisition parameters and operating scenarios across instruments. Statistical methods, on the other hand, are more versatile but require the manual setting of multiple hyperparameters to adapt to diverse input data. Furthermore, latest deep learning models carry out their analysis by considering radargrams as conventional images. This is reflected in the preprocessing steps (e.g. data augmentation and shuffling) radargrams are subject to and the architectures which are utilized, with the latter usually being part of the latest trends in the computer vision community. With this work, we frame the aforementioned limitations as opportunities. On the one hand, we aim at bridging the existing gap between full manual analysis and end-to-end deep learning methodologies, leveraging on as few human effort as possible. On the other hand, we investigate further and exploit better the characteristics of radargrams as data structures. We aim at going beyond the current trend of "radargrams as images" by exploiting the natural structure of radargrams as *sequences* of range-lines indexed in ascending order by their azimuth coordinates (also known as *slow time* [17]). Viewing radargrams as sequences opens up new classes of models to be applied (e.g. recurrent neural networks [18], vision transformers [19]) and exploits the correlation that naturally exists within neighboring range-lines.

With these rationales in mind, we propose a novel methodology for the semantic segmentation of radar sounder data. We leverage on an initial self-supervised learning step casting the training of a feature encoder as multiple random walks on a graph properly built on target radargrams. We exploit concepts from the video segmentation field together with domain-specific insights to obtain an efficient training step that encodes horizontal patterns within radargrams and reflects the sequential nature of the data. In a second step, we propagate the labels of few ground truth samples to the whole dataset using a label propagation algorithm that takes into account similarity in the neighborhood between radargram patches, suitably derived from the representations learned in the first step. Decoupling training and inference we give experts the possibility to provide labels that flexibly match their needs. Moreover, by training our encoder in a fully unsupervised manner, we distinguish our methodology from existing radar sounder data processing architectures and, with a very limited user interaction, we obtain results that are competitive with the ones achieved by supervised methods and networks with a considerably higher number of parameters. Along with this two-step methodology, we propose a metric to quantify the horizontal correlation of features within radargrams, providing the experts with a tool to visualize and automatically discern break points within horizontal features. We validate the metric by employing it to separate between in-land and floating ice within coastal MCoRDS radargrams and by distinguishing between the Martian polar ice cap and the rocky surroundings, successfully enhancing the label propagation step. The main contributions of this paper can be summarized as follows:

- Definition of a random walk-based unsupervised training strategy for the learning of latent representation in radargrams which focuses on their distinct physical characteristics.
- Proposal of a label propagation strategy with multiple enhancements to efficiently label entire radargrams starting from few reference labelled samples, which can also be provided in real time.
- Introduction of a metric for the quantification of the horizontal correlation in radargrams and its application to the automatic classification in benchmark radargrams.

The remainder of this paper is structured as follows. Section II provides a comprehensive review of works related to the proposed methodology, focusing on video-object segmentation and unsupervised learning of image features. Section III presents the problem of radargram semantic segmentation and the challenges associated with interpreting radargrams. Section IV illustrates the proposed methodology in detail, comprising the unsupervised training step, the label propagation step and the introduction of a metric to quantify the degree

of horizontality within radargrams. Section V presents and discusses the results on two datasets from MCoRDS, on a dataset from SHARAD and further ablation studies on the hyperparameters of the methodology. Finally Section VI provides concluding remarks.

II. Related works

A. Radargram segmentation

The automatic segmentation of radargrams remains relatively unexplored, with only a few recent works addressing this challenging problem, especially in the context of orbital radar sounder (RS) data. The early work in [9] focused on a statistical characterization of noise, followed by thresholding to characterise surface and bedrock returns in icy areas. An important step in the direction of full radargram semantic segmentation has been made by [11], which proposes a Support Vector Machine (SVM) classifier to classify pixels of a radargram in 4 classes: free space (air), ice layers, bedrock and noise. Semantic segmentation provides a richer information than layer tracking only: the distribution of power values within classes can be easily obtained using the segmentation mask and this may help to construct ad-hoc filters to attenuate the noise and magnify the important subsurface features. Moreover, this opened up the possibility to apply SVM to detect and quantify the extent of subsurface features of interest. In [12], the authors exploit a similar SVM-based method to detect polar subglacial lakes. In both cases a preliminary extraction of features from the radargrams is performed. Examples of handcrafted features are the depth in range of pixels, the parameters of a fitted distribution of neighboring pixels, the entropy of neighboring pixels, the similarity with pre-computed reference class distributions. The main disadvantage of SVM classifiers is the need to manually identify the features to be extracted. To address this limitation, with the advent of more powerful computers, many researchers have turned to deep learning (DL) to automatically learn features and apply them to tasks such as subsurface feature detection and radargram semantic segmentation.

A first and comprehensive example of DL applied to radargram semantic segmentation can be found in [15]. The authors develop a novel architecture tailored to data from MCoRDS3 in which features are automatically extracted at different depths in the network. In particular, the proposed architecture consists in a U-net [20] with dilated convolutions and a latent attention mechanism which is trained in a supervised way on radargram-segmentation pairs. This new architecture provides more accurate results than the SVM presented in [11] as well as the vanilla U-net. Also, authors point out the necessity of data augmentation to achieve competitive results.

With a similar fashion, [21] and [14] present a modified U-net with Vision Transformer (ViT) attention layers [19] applied to hidden features. These papers open up the idea of fusing convolutional neural networks (CNN) and Transformer-based architectures in order to harness the benefits of both. While CNNs are pointed out to excel at encoding local spatial information, Transformers achieve good results in encoding global information between different parts of the input [22].

Latest trends in the analysis of radargrams focus on overcoming the scarcity of labels. For instance, [13] presents an unsupervised method for the segmentation of radargrams. The method is based on a contrastive learning step in which features are learnt by bringing together the representations of similar patches of radargram and pushing apart different ones. In a later step, features are analysed and assigned to a class using the cosine similarity between them and a list of manually picked vectors representing the classes of interest. Although based on an initial unsupervised step, the final choice of class reference vectors makes the model semi-supervised. In order to overcome the label scarcity, [16] presents a weakly supervised transfer learning approach: a CNN autoencoder is pre-trained in an unsupervised way with the task of radargram reconstruction and then fine-tuned on a small labelled dataset. Results are promising and show how, with a relatively small labelled dataset ($\times 0.25 - \times 0.75$ with respect to the number of unlabelled samples), a weakly pre-trained model can match the performance of fully supervised methods.

B. Self-supervised learning

Self-supervised representation learning is a subfield of DL that has gained significant attention in recent years. It revolves around the idea that models can learn useful and meaningful representations from unlabeled data by solving pretext tasks. These tasks are constructed using the inherent structure of the data, and the learned representations can then be transferred to downstream tasks, such as image classification, object detection, or semantic segmentation.

A common pretext task in computer vision is the reconstruction of the input. This task has been brought to fame by the so-called *autoencoders*. These architectures consist in a deep encoder which encodes the input in a latent representation and a decoder which decodes the latent representation to the exact same dimensions of the input. At the end of the training step, the encoder is employed for downstream tasks and the decoder is usually discarded. The reconstruction task consists in asking the autoencoder to reconstruct the input. Since a trivial solution would be learning the identity function, an amount of noise is added to the input. An autoencoder which reconstructs the input from a noisy version of itself is called *denoising* autoencoder (DAE) [23].

Different sorts of noise can be added to the input and many autoencoder-based architectures can be reduced to special cases of DAEs. An important example of DAE is the *masked* autoencoder (MAE) [24], which masks different patches of the input and is asked to reconstruct them. This solution allows for faster pretraining since the encoder is applied to non-masked patches only. Another example of DAE is the architecture presented in [25], where the model is asked to solve a jigsaw puzzle of the image as a pretext task, i.e. predict the correct position of the shuffled patches.

A different class of pretext tasks involves time series of images. Given a time series of inputs, which can be images or tokenized representations, a model is asked to predict the next sample (or samples) in the time series or reconstruct an input sequence from a shuffled version of itself. Solving this task,

the model learns features which represent the dependencies that exist between samples at different time frames. Major examples of this pretext task can be found in the field of video-object-segmentation (VOS) and image time series forecasting. In [26] a Long Short-Term Memory (LSTM) [27] autoencoder is trained with the task of reconstructing the inverse of a given sequence of video frames. According to the authors, in order to tackle this task the model must retain information about motion as well as position of objects. To prevent the model from learning a naive "reverse" function, videos with arbitrary length are used as input. A similar task is used in [28], where an LSTM autoencoder is trained to reconstruct a randomly shuffled version of the input image time series.

Another main branch of self-supervised learning is *contrastive* learning. This paradigm trains models to learn representations of data by contrasting positive and negative pairs of samples. Positive pairs are samples that are similar to each other, while negative pairs are samples that are dissimilar to each other. In an unsupervised setting, it is not clear to the model which samples are reciprocally positives (or negatives) so a solution to this problem has been provided by [29] which sets the positives to be augmented versions of the input image and asks the model to discriminate these from negative samples within a memory bank.

III. Problem Formulation

We define a radargram R as a 2-dimensional matrix and we denote the number of columns (or *traces*) as n_T and the number of rows (or *samples*) as n_S. Hence we can define the set of all radargrams of dimension (n_T, n_S) as:

$$\mathcal{R} = \{R(x,y)|x \in [1,...,n_T], y \in [1,...,n_S]\} \quad (1)$$

where $R(x,y)$ represents the power of the backscattered echo in dB scale at coordinates (x,y) in trace and sample dimensions. We can see examples of radargrams acquired by different instruments in Figure 1.

Our methodology tackles the problem of the semantic segmentation of radargrams. Semantic segmentation is a task that assigns a specific class label to each pixel of the radargram. Semantic classes of interest in radargrams differ from case to case. There have been early examples in the literature of analysis with three classes [8] while latest models identify up to five classes [15]. The methodology presented in this paper can be used with an arbitrary number of classes $n_C \in \mathbb{N}^+$. We define the segmentation map $S \in [1,...,n_C]^{n_T \times n_S}$ of radargram R as a 2-dimensional matrix with dimension (n_T, n_S) where each entry (x,y) represents the correct class of the radargram pixel $R(x,y)$.

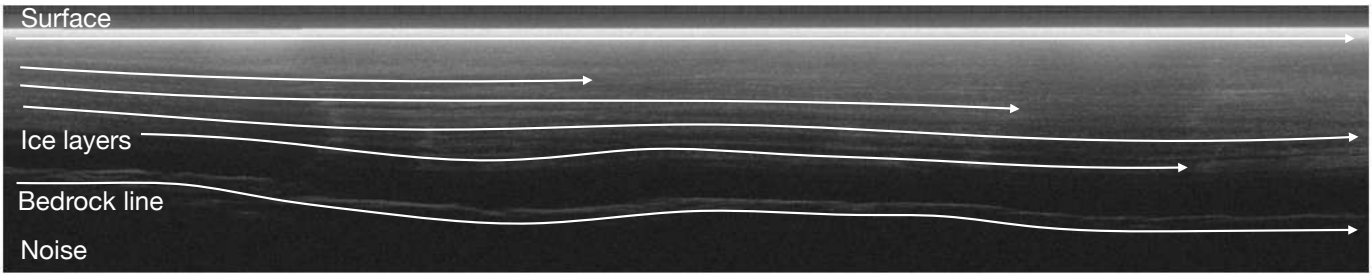
Specific classes and boundaries exist among all radargram products due to the particular nature of the acquisition process. In particular, the highest portion of the radargram is occupied by air (or vacuum, in the very highest portion of satellite-mounted radar sounders) and is often referred to as *free zone*. In this area, radar pulses travel at the speed of light and in the orbital case we may find a dielectric discontinuity caused by the interaction of the waves with the ionosphere. This interaction, which results in a phase distortion of the returned signal, is addressed during ground processing [30]. The first dielectric discontinuity encountered by pulses is the *surface*, which is the boundary separating the free space from the underlying subsurface layers. The surface often represents the brightest return within a radargram since an high portion of the incident pulse is backscattered due to the high dielectric difference between the interfacing layers (i.e. free space and subsurface). However, this is not always the case. Notable findings (see e.g. [1]) have shown that a subsurface water interface can reflect more power than the surface. Below the surface line, the subsurface strongly depends on the acquisition scenario. In a rocky scenario, such as a near-equator portion of Mars, as investigated for example by SHARAD, the subsurface can consist in layers of different geologic materials and may present features such as buried craters and lava tubes. On the other hand, in an icy scenario, such as Antarctica (used as test case in Section V) or the Mars polar layered deposits, the subsurface may consist in layered ice atop the bedrock line. This line can also be seen in radargrams as a bright reflector due to the high dielectric discontinuity between ice and rock (See Figure 1).

Aside from classes, the structure of radargrams differs from the one of conventional images in some key characteristics. In what follows we list the ones we believe to be more important.

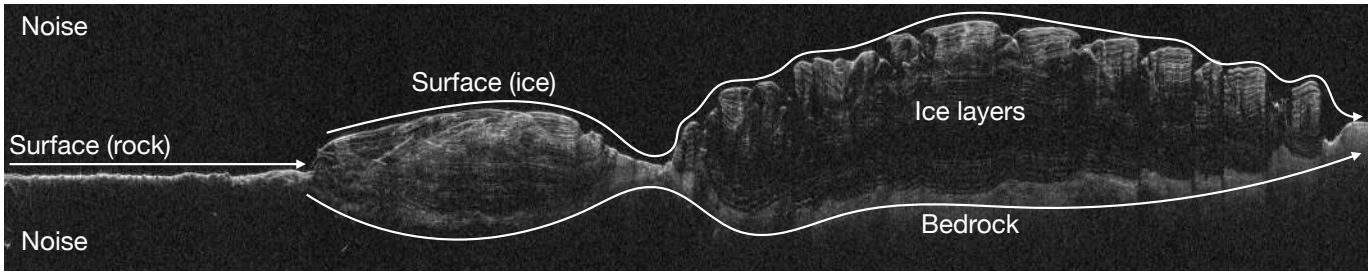
a) One-channel images: Radargrams are usually 1-channel images derived by a complex signal having an amplitude and a phase, while the majority of computer vision images are RGB (3 channels). Models which are pretrained on standard RGB datasets may need some adjustments before fine-tuning on a radargram dataset. The 1-channel nature also implies that there is no spectral information and the model has to rely on intensities of pixels, textures and shapes. In [16], this channel mismatch is overcome adding a convolutional layer at the beginning of a CNN pretrained on Imagenet that reduces the channels from 3 to 1 and discarding the last layers as they contain features that are more specific to Imagenet.

b) Horizontal correlation: There exists an high degree of similarity between adjacent pixels in the horizontal (azimuth) direction [15]. This is due to the geological structure of the subsurface which is exploited by layer tracking algorithms [8], [31]. This similarity is usually high for many consecutive rangelines, especially at ranges that lies above or just below the surface. In the subsurface the correlation between horizontally adjacent pixels is lower and this is mainly motivated by the volumetric scattering within solid materials and the existence of artifacts and non-linear subsurface features. However, we did not found any evidence of exploiting this horizontal correlation in recent deep learning papers regarding segmentation [14], [15], [16]. In [15], [16] the width in azimuth of the training patches is set to 64 pixels, denying the encoding of any strong horizontal dependencies.

c) Relational features: Radargrams are supposed to respect a global macro-structure containing the segmentation classes in a vertical order dictated by the physics of the scenario in question (e.g. in icy zones, starting from the top: free space, ice layers and bedrock). When employing deep learning architectures, we usually assume they can learn these relational features automatically during training. Some architectures (e.g. Transformer) are said to be more prone



(a)



(b)

Figure 1. Examples of radargrams. a) Radargram acquired by the MCoRDS1 radar sounder in Antarctica. We mark in white some examples of its main features, namely the surface line, the ice layers below the surface, the bedrock line and the noise area under the bedrock. b) Radargram acquired by SHARAD over the north polar cap of Mars. In white we can see how noise extends both in the upper and lower part of the radargram. Different surface interfaces, namely rock and ice, are present and the basal return is clearly visible below the ice layers. We highlight with horizontal lines how features develop following horizontal patterns.

than others to learn global shapes and information [22] and there exists way in computer vision to learn these relationships explicitly [25]. Relational features have not been enforced explicitly by latest deep learning models but were addressed in the past by statistical methods [8], [11] via the employment of positional features representing the height of pixels/patches during the processing. We argue that positional encoding could be beneficial for radargram segmentation models due to the presence of shared and physically driven relational features in every radargram instance.

d) Redundancy of information: Semantic information is usually redundant in large areas of radargrams. This is due to physical and instrumental constraints [32]. Firstly, the operating conditions and parameters of the radar sounder pose limits to the penetration of the echo within the subsurface: when the radar receiver is operating it may not receive any echo at all, especially in the latest portion of the receiving window. This converts in an only-noise region at the bottom of the radargram, which can be carefully removed from subsequent analysis. We acknowledge [33] for applying a preprocessing window to radargrams in order to select only regions where reflectors may appear. This windowing speeds up the methodology by masking out the free space region which is far higher than the surface return and noise which is far lower than the bedrock return. Moreover, the frequent elongated shape of surface and subsurface structures in the azimuth direction [16] [15] leads a consistent number of radargram layer tracking algorithms to rely on interpolation [8], [31], [34]. Interpolation assumes global shapes can be inferred by selected representatives. This may recall the latest trend of *masking* [24], which has yet to land in the radar sounders community.

e) Noise component: A radargram is affected by different sources of noise. A first source is the noise caused by the radar system and other electronic devices mounted near the radar [35]. Other noise factors can be clutter caused by cross-track reflections and interference from natural phenomena such as planetary radio emission [36], lightnings and plasma waves [37]. Radargram noise has been modelled, for instance, in [11] and we argue that a deep learning model could automatically learn filters to attenuate the noise and highlight important features.

IV. Methodology

The proposed methodology consists in two stages. In the first stage, an encoder is trained in an unsupervised way using random walks in order to learn meaningful features across radargrams. This stage aims at encoding both global and local characteristics of radargrams, with a particular focus on the horizontal correlation which is present within radargrams. In the second stage, user-provided reference labels are propagated to all radargram patches using a label propagation algorithm together with the features learned in the first step.

A. Random walk unsupervised training

The first stage of the methodology consists in training an encoder M_θ parametrized by parameters θ. The aim of the encoder is to compress a radargram patch of dimension (w, h) into a 1-D vector representation. Hence we define the encoder as

$$M_\theta : \mathcal{P} \rightarrow \mathbb{R}^{n_F} \tag{2}$$

Where $\mathcal{P} \subset \mathbb{R}^{w \times h}$ is the set of all radargram patches of dimension (w, h) and $n_F \in \mathbb{N}^+$ is the length of the feature

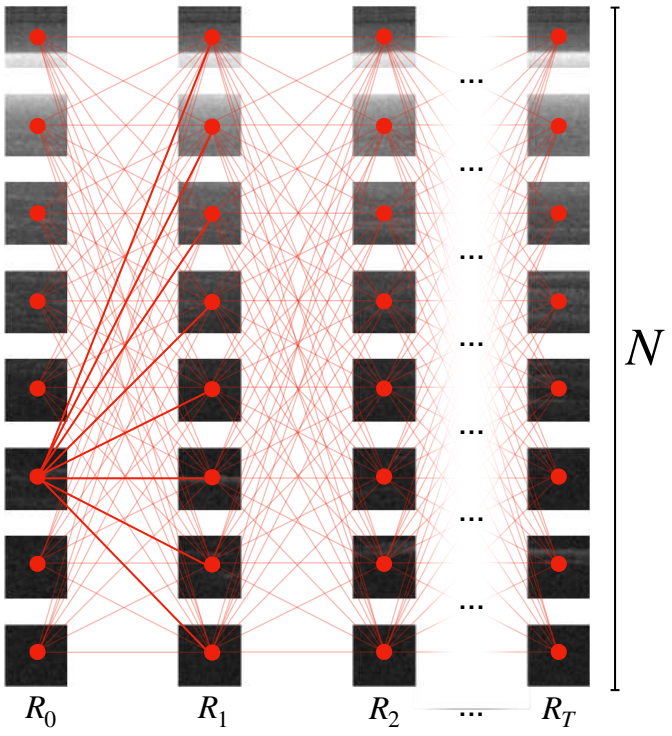


Figure 2. Depiction of a graph build on a dataset from MCoRDS1. Firstly, the dataset is divided in T columns with N patches. Then it is built using patches as nodes and neighborhood to define edges, as explained in Section IV. We highlight the structure of the graph in red with dots representing nodes and lines representing edges. There exist no edges in between the same column and every node of a column is connected to all the nodes of the previous and following column. Patches are non-overlapping for better visualization.

vector obtained as output of the encoder. The aim of the first stage is to train the encoder such that the learned 1-D representations are meaningful and carry information regarding each patch and its role within the radargram.

1) Graph on radargrams: For training the encoder, we take inspiration from [38] and we cast the unsupervised problem as learning the transitions of random walkers on the patches of each radargram. We divide each radargram into a grid of patches and we view this grid as a graph. In the graph, Nodes consists in patches and weighted edges connect patches from different neighboring columns. Formally, given a radargram $R \in \mathcal{R}$ we divide it into $T \in \mathbb{N}^+$ possibly overlapping columns $\{R_t\}_{t=1}^{T}$ where $R_t \in \mathbb{R}^{w \times n_s}$. We divide each column R_t into $N \in \mathbb{N}^+$ overlapping patches $\{R_{t,n}\}_{n=1}^{N}$ of dimension $R_{t,n} \in \mathbb{R}^{w \times h}$. We thus obtain a grid $\{R_{t,n}\}_{t,n}$ of $T \times N$ patches where T and V represent the number of patches in the horizontal and vertical dimension, respectively. We build a graph (V, E) on this grid in the following way. We define $V \in \mathcal{P}$ as the set of vertices of the graph and we instantiate it as:

$$V = \{x \in \mathcal{P} \mid \exists t, n : x = R_{t,n}\} \tag{3}$$

Hence each patch $R_{t,n}$ of the radargram grid is a node and $|V| = N \cdot T$. We define the adjacency matrix of edges $E \in \{0,1\}^{NT \times NT}$ where each element $e_{i,j} \in E$ represents the presence or absence of an edge between vertices $v_i \in V$ and $v_j \in V$. Specifically, if $e_{i,j} = 1$ there is an edge and if $e_{i,j} = 0$ there is no edge between the two vertices. Entries of the edge matrix are defined with the following rule:

$$E(R_{t,n}, R_{t',n'}) = \begin{cases} 1 & \text{if } |t - t'| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In simple terms, each patch is connected to all the patches of the previous and the following column. There are no edges within the same column and there are no edges connecting patches from columns with a distance of more than one column. The matrix is clearly sparse, with only $2N$ non-zero values per row representing the edges of a patch with the N patches of the preceding column and the N patches of the subsequent column. The structure of the graph is briefly illustrated in Figure 2. This structure is in contrast with the usual graphs built in the field of video segmentation. In particular, videos (in greyscale for simplicity) can be cast as 3-D array where the operating dimensions are height, width and frame index. A graph on a video can be built in a similar way than above, partitioning each frame into patches and allowing edges only between patches of neighboring frames. The case of radargrams is peculiar since we are operating on 2-D arrays. Radargrams as sequences of range-lines in the azimuth direction are juxtaposed to videos (i.e. sequences of frames) in this context. However, we argue that range-lines taken as singular entities do not carry enough information about global shapes and textures within radargrams. With this rationale in mind, we use columns R_i with w strictly greater than 1. On the one hand, we want columns (hence patches) which are large enough to contain enough spatial information to be compressed by our encoder. On the other hand we want w to be low enough to not lose the natural sequential behaviour of radargrams dictated by range-lines in the azimuth direction. An ablation study on the dimension of patches is performed in Section V-E.

2) Random walks: With the aim of instantiating random walks on the radargram graph (V, E), we weight each edge with a probability value. As mentioned before, each row of E contains N values corresponding to edges between a patch and the patches of the subsequent column. We use these values to build a transition matrix between the generic column R_t and the subsequent R_{t+1}. There exists an edge between each of the patches of R_t and R_{t+1}, hence we have $N \times N$ edges between the two columns. We assign weights to these edges in order to obtain a *transition matrix* between the two neighboring columns. In this way we are casting the random walk as a time non-homogeneous Markov chain where we have N states corresponding to the N patches of each column, and we have a different transition matrix at each timestep, dictated by the similarity between patches of neighboring columns. Formally, we call P_t^{t+1} the transition matrix between patches of columns R_t and R_{t+1}. We aim at building transition matrices that reward transitions between similar patches and penalise transitions between different patches. Crucially, we compute this similarity in the latent space, i.e. on representations instead of pixel space. Firstly, we input each of the $2N$ patches $R_{t,n}, R_{t+1,n}$ with $n = 1, ..., N$ in the encoder M_θ. We obtain N vector

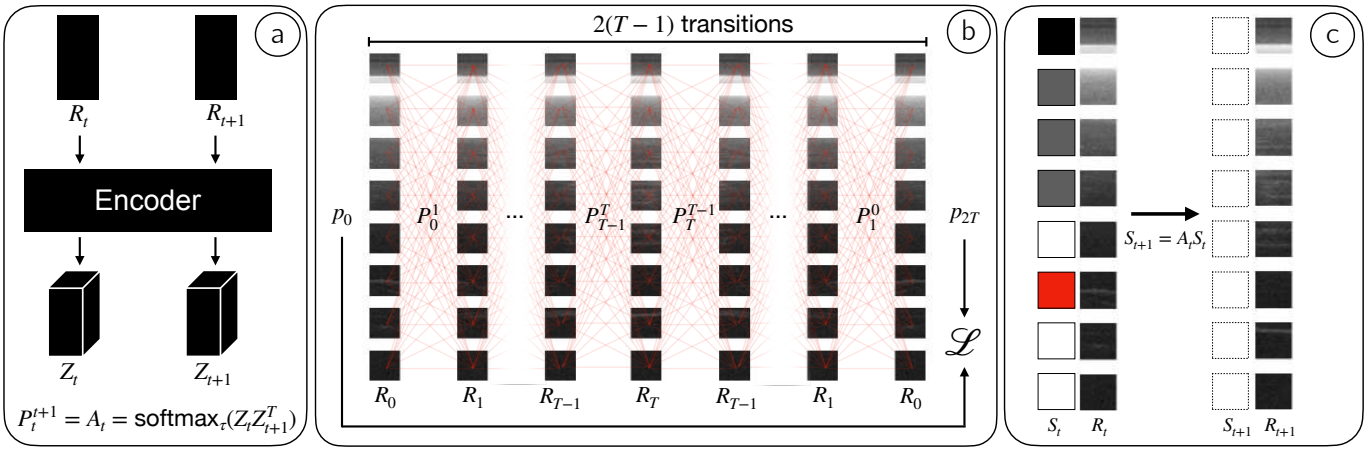


Figure 3. Overview of the methodology. a) Illustration of how representations are obtained from neighboring columns via the encoder and how similarity matrices are generated using the representations. Notice how the same matrices are used both during training and label propagation. b) Random walk training step. We append a flipped version of each radargram to itself and we obtain a graph with $2(T-1)$ transition matrices in-between columns. Given an initial probability p_0, we derive the final probabilities of a walker by multiplying the transition matrices and compute the loss as the cross entropy between the initial distribution and the arrival one. The lower part of the image shows how representations of each column are obtained and how they are converted to transition matrices. c) Label propagation step on two columns, in which, given the reference segmentation S_t of column R_t, the similarity A_t is again computed in the latent space and after the enhancement proposed in Section IV-B it is used to predict the segmentation S_{t+1} of column R_{t+1}. The process is repeated for T columns before a new reference segmentation is required.

representations of dimension n_F for patches of column R_t and N representations for column R_{t+1}. We pile up them, obtaining matrices $Z_t, Z_{t+1} \in \mathbb{R}^{N \times n_F}$, where each row of the matrix corresponds to the vector representation of a patch. We apply L2-normalization row-wise in order to obtain vector representations with norm equal to 1.

Given Z_t and Z_{t+1}, the transition matrix $P_t^{t+1} \in \mathbb{R}^{N \times N}$ between the nodes of the two columns is given by

$$P_t^{t+1}(i,j) = \text{softmax}_\tau(Z_t Z_{t+1}^T)_{ij} \tag{5}$$

where the softmax function is applied row-wise, $\tau > 0$ is the temperature parameters and $[\cdot]^T$ represents the transpose operator. The term within the softmax function is cosine similarity itself, applied on each pair of the L2 normalized rows of Z_t and Z_{t+1}. Softmax crucially converts rows into probability distributions (i.e. vectors with sum equal 1) and enhances high similarity values while lowering low values even more. A low temperature τ allows for an even stronger separation between high and low values, biasing transitions toward the highest probability value.

We compute similarities for each pair of columns on the graph, obtaining a list of transition matrices $\{P_t^{t+1}\}_{t=0}^{T-1}$. These matrices can be seen as transition matrices of a discrete time non-homogeneous Markov Chain where states consists in the nodes of the graph and transitions between timesteps t and $t+1$ occur according to the transition probabilities in P_t^{t+1}.

Consider a random walker with initial state $X_t = R_{t,n}$, the probability of the walker to be in state $n' \in [1, ..., N]$ at time $t+1$ is given by:

$$P_t^{t+1}(n,n') = \mathbb{P}(X_{t+1} = R_{t+1,n'} | X_t = R_{t,n}) \tag{6}$$

One can generalize this one-step formula with the Chapman-Kolmogorov equation, i.e. multiplying subsequent transition matrices, obtaining the transition probabilities for a walker at time t after k steps as:

$$P_t^{t+k} = \prod_{i=0}^{k-1} P_{t+i}^{t+i+1} \tag{7}$$

3) Loss function: We recall that the transition matrices $\{P_t^{t+1}\}_{t=0}^{T-1}$ ultimately depends on the 1-D representations of patches obtained via M_θ. With the aim of training the encoder at learning meaningful representations, we instantiate multiple walkers on the graph and optimise their transitions in order to follow the horizontal patterns that exists within the radargram. Precisely, we can obtain transitions for a walker on the graph by defining its initial state as a probability distribution $p_0 \in [0,1]^N$ and $\sum_i p_0(i) = 1$. Given the initial state as R_{0,n_0}, where $n_0 \in 1, ..., N$ is the index of the patch of the initial state, we define p_0 as:

$$p_0(i) = \begin{cases} 1, & \text{if } i = n_0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

We obtain the probability distribution of the walker after k steps as:

$$p_k = p_0 P_0^k \tag{9}$$

In order to provide a target for the walk, we use of the concept of *cycle consistency* [39] as previously introduced for videos by [38]. Cycle-consistency is a widespread concept in the video segmentation field and consists in the fact that a target within the frames of a video spans similar pixels positions in neighboring frames, going both forward and backward in time. We force cycle-consistency by appending to the radargram a flipped version of itself in the azimuth direction. We consequently append a flipped version of the graph to the original graph and we do the same with the transition matrices, obtaining a list of $2 \times (T-1)$ matrices. We train walkers to

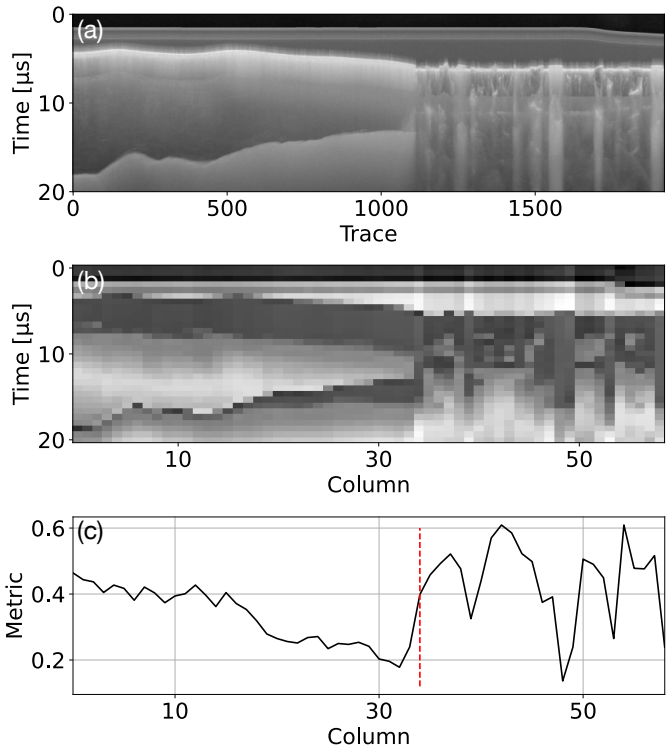


Figure 4. Example of application of the horizontality metric. (a) Segment of an MCoRDS3 radargram in which the grounding zone is clearly visible in the middle. (b) Horizontality metric applied to the segment: we can clearly distinguish dark areas corresponding to horizontal features and bright areas corresponding to patches in which walkers are more uncertain. Darker pixels corresponds to the surface and bedrock lines. The image also highlights the distinction between inland ice, which contains smoother pixel areas, and floating ice, which shows more irregular patterns due to the presence of liquid water. (c) Mean of columns of the metric computed as illustrated in Section IV-C1. The plot clearly shows two segments with different variances corresponding to the two distinct zones. We highlight in red the change point found via binary segmentation algorithm on plot (c), which corresponds to the grounding zone visible in (a).

start from position n_0 at time $t = 0$ and arrive at position n_0 at time $2 \times T$. Consequently, we obtain a loss function which is a cross-entropy on N classes (i.e. final positions) between the initial position (target) and the arrival position (prediction). We can represent this loss with the following formula:

$$\mathcal{L} = \text{CrossEnt}(p_0, p_{2T}) = -\sum_{i=0}^{N} p_0(i) \log(p_{2T}(i)) \tag{10}$$

Since p_{2T} is a function of $\{P_t^{t+1}\}_{t=0,...T-1}$, which ultimately depends on $\{Z_t\}_{t=0,...T}$, minimizing $\mathcal{L}$ optimizes the parameters θ of M_θ.

According to (10), the path of each of the N walkers has the same importance in terms of loss. We refine the loss equation to prioritize training on the brightest (i.e. high returning echo intensity) features present in radargrams. The following adjustment is based on the rationale that these features are the most scientifically relevant. As introduced in Section III, the brightest horizontal pattern is typically characterized by the interface between free space and the subsurface. Additionally, notable examples of bright subsurface features include the ice layers and the bedrock line (or basal return), in the case of acquisition on an icy scenario. With the aim of prioritizing walkers that follow these horizontal features, we compute the mean of each patch of column R_0 (i.e. the initial state of walkers), obtaining means $\{m(i)\}_{i=1}^{N}$. We then define the weights $\{w(i)\}_{i=1}^{N}$ as:

$$w(i) = \frac{m(i)}{\sum_i m(i)} \tag{11}$$

and rewrite the loss function as a weighted cross-entropy loss:

$$\mathcal{L} = -\sum_{i=0}^{N} w(i) p_0(i) \log(p_{2T}(i)) \tag{12}$$

With this loss, higher priority is given to walkers that start from brighter patches, which we rationally assume being part of a relevant horizontal pattern.

To speed up training and harness parallelization, we compute transitions for multiple agents on the same graph at the same time by substituting the vector-matrix product of (9) with the matrix-matrix product:

$$P_k = I_N P_0^k \tag{13}$$

Here each row of I_N (i.e. identity matrix of dimension N) represents a walker starting from a different position $n \in \{1, ..., N\}$ on the first column of the graph and each row of P_k represents the distribution of the arrival state at $t = k$ of each of the walkers. Loss becomes a row-wise cross-entropy between rows of I and rows of P_{2T}. We reduce the loss to a scalar value by computing the mean row-wise cross-entropy, obtaining a target for the optimization problem:

$$\underset{\theta}{\text{argmin}}\, \mathcal{L} \tag{14}$$

We iterate the random walks and computation of loss over multiple radargram column sequences of fixed length T and we minimise the loss via stochastic optimization, aiming at a global optima for the parameters θ of the model M_θ.

B. Label propagation

In the second phase, we leverage on the model M_θ trained in the previous step and on a reference segmentation, likely to be user-provided (also in real time), to propagate labels to all the patches in order to achieve a full semantic segmentation of the radargram. An overview of the label propagation step can be seen in Figure 3. Inspired by [40], we make use of a variation of *KNN label propagation* suitably refactored to operate on radargrams. We firstly introduce the propagation starting from the first two columns of a radargram and we later generalize to the full radargram, introducing also a list of proposed enhancements. Given a radargram R we suppose to have the patch-wise segmentation map $S_0 \in \{0, ..., n_C\}^{1 \times N}$ of the patches $\{R_{0,n}\}_{n=1}^{N}$ of the first column. We aim at predicting the segmentation of the N patches of column R_1. In order to do this, we again use the piled latent representations $Z_0, Z_1 \in \mathbb{R}^{N \times n_F}$ of R_0 and R_1, obtained via the trained encoder M_θ. As in the first phase, we compute the similarity in the latent space, i.e. between Z_0 and Z_1, using (5). We

obtain the similarity matrix $A_0 \equiv P_0^1 \in \mathbb{R}^{N\times N}$. We pick the $k > 0$ highest values for each row, thus masking and setting to 0 all the other values and obtaining the modified matrix $A_{\text{knn},0}$.

We predict the segmentation map S_1 as a linear combination of the labels of S_0, with weights given by $\tilde{A}$:

$$S_1 = A_{\text{knn},0} S_0 \tag{15}$$

We can then proceed in an iterative way, predicting the segmentation of S_{t+1} as:

$$S_{t+1} = A_{\text{knn},t} S_t \tag{16}$$

for $t \in \{0, ..., T-1\}$. We apply the process of propagation for T consecutive patches at which point we provide the algorithm with a new reference segmentation mask. The process is repeated until all columns of the radargram have been processed. We refine KNN label propagation with techniques from [38] and [41]. We also introduce a backward propagation step to refine the characterization of the extended linear features. We implement the following enhancements.

1) Memory bank: As we obtain new labelled samples, we harness them to compute the latest segmentation. Precisely, in order to compute S_{t+1}, we make use of the previously obtained $\{S_i\}_{i=t-c}^{t}$, where $c > 0$ represents the previous *context* taken into consideration by the label propagation. We compute $\tilde{A}$ between R_{t+1} and all the previous $\{R_i\}_{i=t-c}^{t}$, obtaining $t-c$ similarity matrices $\{\tilde{A}_t\}_{i=t-c}^{t}$. We concatenate the matrices to obtain $A_{\text{bank}} \in \mathbb{R}^{N\times N(t-c)}$ where each row n contains the similarities between $R_{t+1,n}$ and each patch of the columns in the context bank $\{R_i\}_{i=t-c}^{t}$. Again we find the top k values row-wise and we set to 0 the rest. Similarly to (16) we compute new labels as:

$$S_{t+1} = A_{\text{bank},t} S^* \tag{17}$$

where $S^* \in \{0, ..., n_C\}^{c\times N}$ is the concatenation of all segmentation maps in the memory bank at time t.

2) Masked attention: We further restrict the computation of similarities only within a radius $r \in \mathbb{N}^+$ from the query patch. This can be achieved mathematically by the element-wise multiplication of A_t and the r-diagonal matrix $I_r \in 0, 1^{N\times N}$ whose entries are 1 in the $1 + 2r$ central diagonals and 0 otherwise:

$$A_{\text{mask},t} = A_t I_r \tag{18}$$

3) Backward label propagation: We employ a backward label propagation step with the aim of focusing on a particular target class within radargrams. The choice of this class is arbitrary. For example, it could consist in a layer of interest in a rocky subsurface such as a buried lava flow and it could be in the bedrock line when dealing with products acquired on the ice sheet. As the propagation reaches column R_T the algorithm is provided with the reference segmentation S_{T+1} and start operating on the next sequence. We make use of S_{T+1} to provide a second segmentation map for the sequence up to T. Label propagation operates backward on the graph from T to 0. Notice that this does not harness the computational cost in an excessive way since the similarity matrices have already been computed in the forward cycle of label propagation. The only difference in the backward pass is the new reference segmentation. Given $\bar{c} \in \{0, ..., n_C\}$ as the index of the class of interest, after obtaining S_{back} we merge it with S_{fwd} with the following operation on S_{fwd}:

$$S_{\text{fwd}}(i,j) = \begin{cases} \bar{c}, & \text{if } S_{\text{back}}(i,j) = \bar{c} \\ S_{\text{fwd}}(i,j) & \text{otherwise} \end{cases} \tag{19}$$

which sets a pixel to the focused class if the pixel has been labelled as $\bar{c}$ in at least one segmentation map between S_{fwd} and S_{back}.

C. Horizontal correlation metric

Here we introduce a metric to quantify the degree of horizontal smoothness within portions of a radargram. The metric is important for various reasons. Firstly, the metric is tied to the unsupervised training step presented in Section IV-A and helps at quantifying and visualising the trajectories of random walkers and the linearity of horizontal features. This increases the explainability of the methodology and allows the transitions obtained during training to be used not only at inference time (i.e. label propagation) but also as an additional standalone analysis tool. Moreover, the evolution of the metric within the radargram can be used to detect abrupt changes in the physical scenario such as the *grounding zone* of ice within coastal radargrams [16] and missed range lines. We address this possible employment of the metric at the end of this section.

We suppose to have an encoder M_θ trained as per Section IV-A and a sequence of neighboring radargram columns $\{R_t\}_{t=1}^{T}$, where each column is composed of patches (possibly overlapping) of dimension $w \times h$. For each timestep $t \in \{1, ...T\}$ we input patches to the encoder and we obtain a sequence of transition matrices $\{P_t^{t+1}\}_{t=0,...T-1}$ as in (5). We define the *horizontal correlation metric* of column R_i as:

$$\mu_t = \text{CrossEnt}(P_t, I) \tag{20}$$

where I is the indentity matrix and CrossEnt is row-wise cross-entropy without reduction (i.e. without collapsing the resulting vector into a scalar using a sum or mean function). Notice that μ_I has dimension equal to the number of patches within column R_t. We can then visualise the metric on the neighboring column (i.e. a portion of radargram) by simply concatenating $\{\mu_t\}_{t=1}^{T}$ in the time dimension.

1) Change point detection: We present a way to employ the metric to enhance the segmentation maps obtained via label propagation. The following steps are applied in Section V to improve the results of the experiment on radargrams acquired on a coastal icy scenario.

We suppose to be operating on a radargram slice of T columns and we suppose an abrupt change exists in column $\bar{t}$. Since the label propagation step is based on very few reference samples, it is most likely that the column representing the change will not be the reference column of our selected radargram slice. We compute the horizontal correlation metric using (20) on each transition matrix in $\{P_t^{t+1}\}_{t=0,...T-1}$. We obtain T metric vectors $\{\mu_t\}_{t=0,...T-1}$. We concatenate these vectors in order to obtain a 2-dimensional array $\Gamma_{T_{\text{seq}}} \in \mathbb{R}^{T_{\text{seq}}\times N}$.

Table I
DETAILS ON THE DATASETS USED FOR EXPERIMENTS

	MCoRDS1 [11]	MCoRDS3 [16]	SHARAD
Number of traces	27350	105120	29011
Column size	410×32	1536×32	912×16
Patch size	32×32	32×32	16×16
Overlap	30×0	30×0	8×0

In Figure 4-b we can see an example of $\Gamma_{T_{\text{seq}}}$ for an MCoRDS radargram slice. We can clearly see that before the change point (i.e. the ice grounding zone) the metric tends to be lower while after the grounding zone, where the floating ice is present, pixels present higher values. This translates in random walk transition matrices being consistently different from the identity (i.e. horizontal walks) and are more spread toward multiple choices for the next step. We expect this behaviour to be always marked by a sharp change in the intensities of $\Gamma_{T_{\text{seq}}}$, especially in the case of changes which are spread across columns such as grounding zones. As we can see in Figure 4-c the change point is less evident using a simple rolling mean. With this behaviour in mind, we compute a rolling mean of $\Gamma_{T_{\text{seq}}}$ in the azimuth direction and we obtain a vector $\gamma \in \mathbb{R}^{T_{\text{seq}}}$ (Figure 4-d) which should present a change point at the level of the abrupt change. We detect this change point via binary segmentation [42], retrieving the index $\bar{t}$. We choose this algorithm for its fast computation time and the ability to identify change points by discerning changes in variance across segmented data. After obtaining the index $\bar{t}$ of the change point, we provide the reference sample $S_{\bar{t}}$ of the column $R_{\bar{t}}$ corresponding to that index to the model and we ask for a new iteration of label propagation on the remaining $T_{\text{seq}} - \bar{t}$ columns, potentially correcting the previously erroneous segmentation.

V. EXPERIMENTS

A. Overview of the datasets

We validate our methodology on two terrestrial and one planetary datasets from the literature. The first two datasets have been obtained by MCoRDS1 [2] and MCoRDS3 [3], two radar sounder instruments operated by CReSIS. Both datasets refer to ice sheet scenarios. The third dataset consists in 5 radargrams obtained by SHARAD on the Mars North Pole. The reason for this choice is that icy scenarios are good benchmarks for assessing the effectiveness of the proposed technique as they allow the detection of multiple classes which are tied to each other via strict physical relationships. We can find the *free space* at the top of radargrams, which consists in the space between the instrument and the surface. Right below the surface there is the semantic class consisting in *ice layers*. Coastal radargrams usually separate *in-land* ice, i.e. ice deposits on the bedrock, and *floating* ice, which floats over the ocean. We can see this distinction in the MCoRDS3 dataset. Furthermore, a key class within radargrams from ice-covered regions is the *bedrock line*. It consists in a bright line within MCoRDS radargrams and identifying this interface is crucial, as it enables quantifying the ice deposits and detecting features of interest such as subglacial lakes [12]. As fewer knowledge exists on the nature of subsurface layers in SHARAD radargrams, we distinguish between *strong returns* and *weak returns* within the Martian polar cap, consistently with previous literature [9]. We define as *basal layer* the bright return characterized by diffuse scattering right under the polar cap. Another important class is *noise*, consisting in areas where there is no returning echo or it is weaker than the noise. This often happens when no relevant dielectric discontinuities are present in the subsurface and the weak returning echo is dominated by the instrumental noise. There have been several attempts in recent literature to characterize noise in statistical terms [11], [9] by exploiting theoretical models built up on radar theory, where it is usually assumed that noise is Rayleigh distributed and has a multiplicative behaviour over the signal of interest [9]. Noise areas are especially present right above the bedrock line (*echo-free zone*, EFZ) and below the bedrock line, when the pulse is attenuated the most and few-to-no echo returns back to the radar receiver.

1) MCoRDS1 Dataset: The first dataset, namely $\mathcal{D}_1$, has been obtained by the radar sounder MCoRDS1 in Antarctica. The chirp bandwidth of the radar sounder is $189 - 198$ MHz and this leads to a range resolution of 13.6 m in ice and an azimuth resolution of 25 m after synthetic aperture radar (SAR) processing. The dataset consists in a radargram from the campaign of 04/11/2010, for a total of $n_T = 27350$ rangelines ($\sim$ 520km of continuous acquisition) and $n_S = 410$ samples per rangeline. The dataset has been pre-processed by the team of CReSIS using SAR processing, motion compensation and the minimum variance distortionless reponse (MVDR) beam former. A segmentation map for this dataset has been provided by [11]. We use this map for extracting labelled reference samples during the label propagation step of the proposed approach. The map consists in 5 classes: free space, ice layers, bedrock line, noise and uncertain region. We use the first 4 classes to test our methodology and we remove pixels from the uncertain region when computing the classification metrics as previously done in literature on this dataset [11], [14] for a fair comparison with fully supervised methods.

2) MCoRDS3 Dataset: The second dataset, namely $\mathcal{D}_2$, has been obtained by the radar sounder MCoRDS3 in Antarctica. The chirp bandwidth of the sounder is again $165 - 215$ MHz and leads to a resolutions of 2.6 m in range (in ice) and 27.5 m in azimuth post SAR processing. The main difference from MCoRDS1 is the doubled number of RX channels (8 for MCoRDS1 vs 16 for MCoRDS3). The dataset consists in multiple radargrams from a campaign in 2018, for a total of $n_T = 105120$ traces and $n_S = 1536$ samples per trace. The dataset has been presented in [15] and has already been used in radar sounder literature [15], [16] to validate semantic segmentation methodologies. The segmentation map consists in 6 classes: free space, inland ice, floating ice, bedrock, noise and uncertain region. Consistently with literature, we discard the uncertain region when computing metrics, obtaining a semantic segmentation setting with 5 classes.

3) SHARAD Dataset: The third dataset, namely $\mathcal{D}_3$, has been obtained by the radar sounder SHARAD at the North Pole of Mars. The chirp bandwidth of the sounder is $15 - 25$ MHz and leads to a range resolution of around 10 m

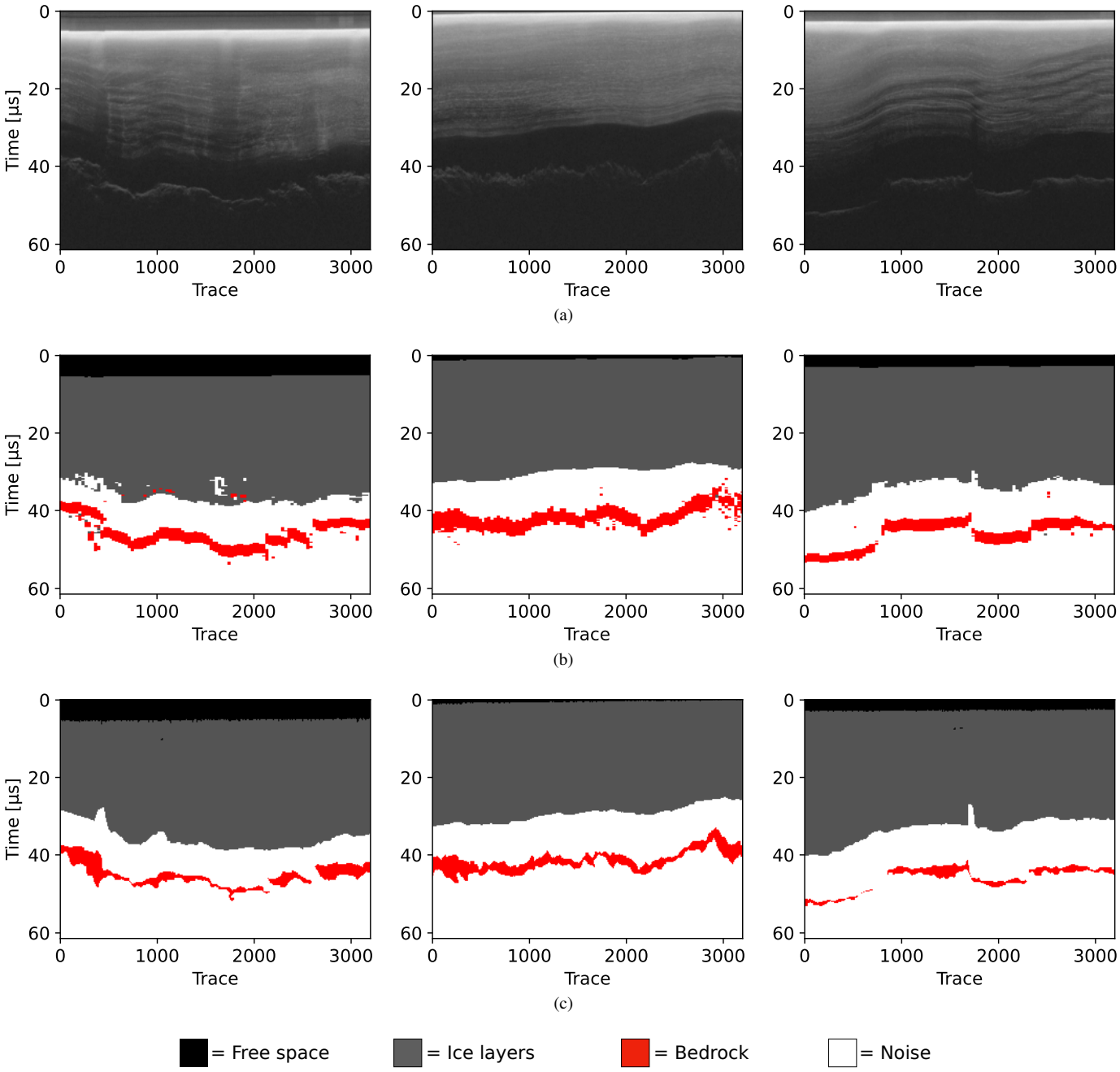


Figure 5. Results of the methodology applied to radargrams from dataset $\mathcal{D}_1$. a) Radargrams from MCoRDS1 which are part of $\mathcal{D}_1$. b) Result of the segmentation using the proposed methodology. c) Segmentation ground truth. The first and last traces of each radargram in c) are used as references for forward and backward label propagation, respectively. As we can see the resulting segmentation maps correctly follow the main horizontal features within the radargrams. In the leftmost radargram, the late break point within the bedrock is well represented. In the rightmost radargram, the methodology is able to follow abrupt changes in the bedrock horizontal pattern. The boundary between noise and ice layer is generally segmented correctly and the free-space line is depicted as expected.

in ice and an azimuth resolution of approximately 460 m after synthetic aperture radar (SAR) processing [43]. The main difference between this and the MCoRDS datasets is the orbital nature of the SHARAD instrument and the related acquisitions. SHARAD employs a much lower frequency compared to its airborne counterparts, allowing for a deeper penetration of the subsurface at the cost of a much smaller bandwidth. The dataset consists in 5 radargrams processed by the US SHARAD team [43], which are acquisitions number 00285902, 00287702, 00293902, 00331102, 01988401, for a total of $n_T = 29011$ rangelines. We manually cut the uppermost and lowermost parts of radargrams in order to obtain $n_S = 912$ samples per trace. The segmentation map consists in 5 classes: free space, strong returns, weak returns, basal layer and noise.

B. Experimental setup

We run experiments with the proposed technique on the aforementioned datasets. In particular, we first run the unsuper-

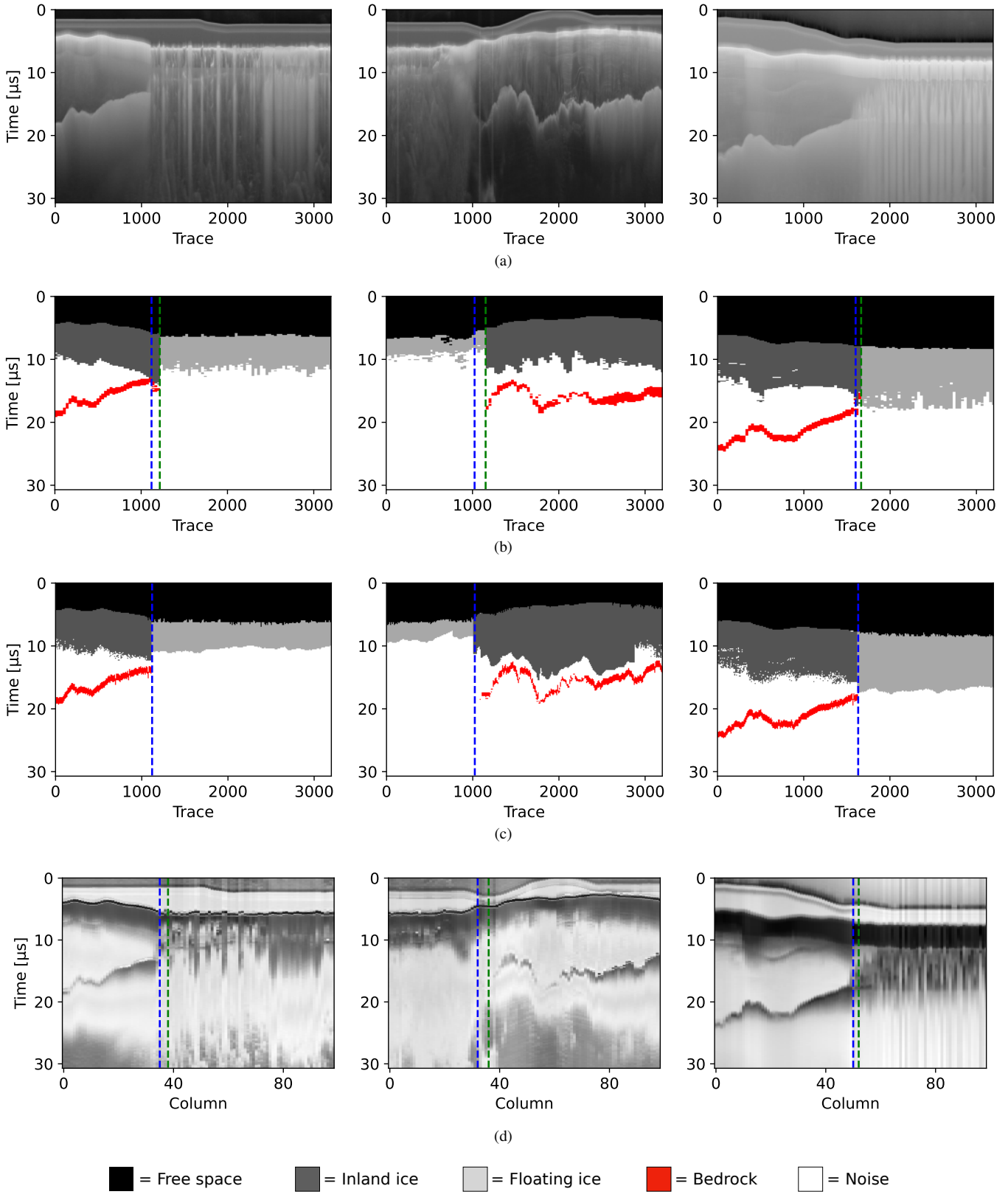


Figure 6. Results of the methodology applied to radargrams from dataset $\mathcal{D}_2$. a) Radargrams from MCoRDS3 in which we can clearly see the change point between floating and inland ice. b) Result of the segmentation using the proposed methodology. c) Segmentation ground truth from [16]. d) Horizontal correlation metric computed as in Section IV-C where bright pixels correspond to high cross-entropy according to (20). We highlight with green and blue dashed lines the change points found using the technique presented in Section IV-C and according to the ground truth maps, respectively. As we can see the resulting segmentation maps correctly follow the main horizontal features within the radargram. The methodology is able to detect the change points within the radargrams, which generally corresponds to the ones depicted in the ground truth segmentation maps. The methodology correctly recalls bedrock and ice thickness for long portions of the radargrams.

vised training step and then we apply label propagation. We perform the following experiments:

- Training and label propagation on $\mathcal{D}_1$
- Training and label propagation on $\mathcal{D}_2$
- Training and label propagation on $\mathcal{D}_3$

For the first two experiments we use a patch size of $(w, h) = (32, 32)$ and an overlapping of $(o_w, o_h) = (30, 0)$. For the experiment on $\mathcal{D}_3$ we set $(w, h) = (16, 16)$ and $(o_w, o_h) = (8, 0)$. The patch size is chosen with the rationale of obtaining patches large enough to contain the full vertical extent of the main interfaces (surface and bedrock lines). We set a relatively high overlapping in the range direction ($> 50\%$ of the patch size) to obtain a finer segmentation map. The lower patch size employed for $\mathcal{D}_3$ is motivated by the larger range resolution of the dataset, which allows fewer pixels to capture the vertical extent of horizontal features. We perform ablation studies on patch size and overlapping value in Section V-E.

The sequence length is set to $T_{\text{seq}} = 10$ in the training. During label propagation we set $T_{\text{seq}} = 100$. We choose these values after a preliminary empirical evaluation: an higher T_{seq} during training leads to divergence and the choice of T_{seq} during label propagation is made to minimize the number of labelled samples required. Indeed, $T_{\text{seq}} = 100$ leads to the usage of one labelled sample every 100, which represents 1% of the datasets. Moreover, with this combination of patch size and sequence length we span $w \cdot T_{\text{seq}} = 3200$ range lines during label propagation on $\mathcal{D}_1$ and $\mathcal{D}_2$, which is roughly the length of radargrams provided by CReSIS [44] (i.e. ~ 50km of acquisition). In the case of $\mathcal{D}_3$, we span a total of approximately 640km per sequence. We recall and underline that these labelled samples are only employed during inference and not for training. The temperature of softmax is set to $\tau = 10^{-2}$. Parameters for the label propagation are set to $(k, c, r) = (10, 80, 30)$ for $\mathcal{D}_1$, $(k, c, r) = (10, 80, 60)$ for $\mathcal{D}_2$ and $(k, c, r) = (10, 80, 10)$ for $\mathcal{D}_3$ after a preliminary grid search. However, we see no particular fluctuations in results when varying the parameters, except for r. We set r with the rationale of spanning at least one non-overlapping patch above and below the current one (see ablations on this value in Section V-E). We choose $\bar{c}$ as the bedrock class (or basal layer, in the case of $\mathcal{D}_3$) for the backward label propagation (Section IV-B3), with the rationale of improving the recall of the class due to its low support and its importance as an example of relatively thin, coarse and isolated horizontal feature. We employ change point detection as described in Section IV-C1 during the experiment on $\mathcal{D}_2$ due to the coastal scenario of acquisition and the presence of multiple examples of grounding zones within radargrams. We also apply the technique on $\mathcal{D}_3$ in order to distinguish between the Mars polar deposit and the surrounding areas. We do not apply the technique on $\mathcal{D}_1$ due to the relative stability of horizontal features within the dataset and the fully inland scenario of acquisition. We train for 50 epochs with a batch size of 64 on two Nvidia GeForce 3060 RTX. Learning rate is set to 10^{-3}.

For the proposed technique, we use a customised Resnet18 [45] with randomly initialized weights as encoder M_θ. We add a convolutional layer, a batch normalization layer and a ReLU activation function at the beginning of the network in order to transform our 1 channel inputs to 3 channels. The central part of the network consists of residual blocks as per [45]. We remove four of the total eight residual blocks to lighten the network. We also add two final linear layers in order to transform the 2-dimensional output of the Resnet in the 1-dimensional embedding used for computing cosine similarities. The architecture has a total of 4971468 trainable parameters.

C. Evaluation metrics

We evaluate the performance of our methodology using precision, recall and F1 score. For each class, we define the *true positives* (TP) as the correctly labelled pixels of the class and the *true negatives* (TN) as the pixels that do not belong to the class and our model correctly attributes to other classes. We define the *false positives* (FP) as pixels which belong to other classes and our model attributes to the considered class and the *false negatives* (FN) as pixels which do belong to the considered class but are incorrectly labelled by the model as belonging to other classes. We can now define the precision of a predicted segmentation map with respect to a reference map as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{21}$$

We can also define the recall as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{22}$$

The F1-score is defined as a combination of the two:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

We also define the *overall accuracy* (OA) as:

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\% \tag{24}$$

We assess the performance of our methodology with these four metrics. For the two datasets, we also define the *support* of each class $c_i \in \{1, ..., n_C\}$ as:

$$\text{Support}_{c_i} = \frac{\#\text{Pixels in class } c_i}{\#\text{Total pixels}} \tag{25}$$

and we show it along with the metrics in order to quantify the unbalance between the classes. Moreover, we compare the OA of our methodology with the results obtained by other methods presented in the literature. In particular, we compare results on $\mathcal{D}_1$ with the SVM classifier from [11], the vanilla U-net presented in [14] as a baseline and the TransSounder architecture [14], which fuses Vision Transformer blocks and convolutional layers. Results on $\mathcal{D}_2$ are compared with the CNN proposed in [16], which is a double U-net pretrained using a reconstruction task and finetuned on $\mathcal{D}_2$, the U-net proposed as baseline in [15] and the U-net with dilated convolutions and latent attention layers presented [15] and further enhanced with post-processing using morphological filters. Finally, we compare $\mathcal{D}_3$ with the SVM classifier from [11] and a vanilla U-net architecture applied on a 90-10 train-test split (to be consistent with [15]). The proposed methodology requires a computational time which is, in general, similar to the

Table II
SEGMENTATION PERFORMANCE OF THE PROPOSED METHOD ON THE MCORDS1 DATASET

	Precision	Recall	F1-score	%Support
Free space	0.96	0.99	0.98	0.06
Ice layers	1.00	0.98	0.99	0.56
Bedrock	0.92	0.87	0.89	0.04
Noise	0.96	0.99	0.98	0.34
OA				0.98

Table III
SEGMENTATION PERFORMANCE OF THE PROPOSED METHOD ON THE MCORDS3 DATASET

	Precision	Recall	F1-score	%Support
Free space	0.98	1.00	0.99	0.21
Noise	0.96	0.96	0.96	0.50
Bedrock	0.68	0.65	0.66	0.02
Inland ice	0.96	0.95	0.96	0.20
Floating ice	0.90	0.90	0.90	0.07
OA				0.96

Table IV
SEGMENTATION PERFORMANCE OF THE PROPOSED METHOD ON THE SHARAD DATASET

	Precision	Recall	F1-score	%Support
Free space	0.95	0.97	0.96	0.40
Strong returns	0.64	0.64	0.64	0.08
Weak returns	0.80	0.66	0.73	0.09
Basal layer	0.65	0.54	0.60	0.03
Noise	0.94	0.97	0.95	0.40
OA				0.90

Table V
COMPARISON WITH SUPERVISED ARCHITECTURES

Dataset	Method	Training	OA	# Params
MCoRDS1	SVM [11]	Supervised	0.99	-
	U-Net [14]	Supervised	0.93	20M
	TransSounder [14]	Supervised	0.99	125M
	Proposed	Unsupervised	0.98	5M
MCoRDS3	CNN [16]	Supervised	0.99	62M
	U-Net+ASPP [15]	Supervised	0.98	12M
	U-Net [15]	Supervised	0.91	8M
	Proposed	Unsupervised	0.96	5M
SHARAD	SVM [11]	Supervised	0.92	-
	U-Net [15]	Supervised	0.88	8M
	Proposed	Unsupervised	0.90	5M

other mentioned literature techniques. All the methodologies employed for the comparison are fully supervised and use different train/test splits and data augmentation techniques, hence the comparison is affected by some uncertainty margin. Nonetheless, this allows us to show the competitiveness of our results against methods that both employ labels during training and are based on architectures with an higher number of trainable parameters.

D. Results

1) Dataset $\mathcal{D}_1$: The segmentation performance of the proposed method on dataset $\mathcal{D}_1$ is summarized in Table II. Across all classes, the method demonstrates promising precision and recall values, indicating its ability to accurately classify the various subsurface features.

For free space, ice layers, and noise, the method achieves high precision scores of 0.96, 1.00, and 0.96, respectively. This reflects a low false positive rate in identifying these classes. Additionally, the recall values for these classes are equally high, ranging from 0.98 to 0.99, indicating an accurate classification of most instances of these targets within the dataset. Our model correctly characterise free space as it is the easiest class to detect due to the surface line acting as a bright delimiter between free space and subsurface.

In contrast, the precision score for bedrock classification is slightly lower at 0.92. This could be attributed to the complexity and variability of bedrock signatures within radar sounder data. Moreover, bedrock is the class with the lowest support hence the model is trained on significantly less bedrock patches than other classes. Despite this, the method demonstrates a relatively high F1 score on the bedrock class, also considering it is almost unsupervised. Indeed, results are promising in lieu of the fact that the model infers the segmentation starting from only one reference column every 100.

Qualitative results are shown in Figure 5b. The segmentation maps obtained via the proposed methodology on three radargrams from $\mathcal{D}_1$ highlight the ability of the model to follow the main horizontal targets. In general, the surface line, the interface between ice layers and EFZ, and the bedrock are correctly tracked, as well as the major sudden changes in their horizontal pattern.

We compare the performance of our methodology on $\mathcal{D}_1$ with the results of other architectures from the literature in Table V. Our results are competitive and overcome the ones of the baseline supervised U-net presented in [14]. Moreover, the table shows how our encoder achieves competitive performances on $\mathcal{D}_1$ with only 5M parameters against the 20M of the baseline U-Net and the 125M of TransSounder [14]. This is crucial for two reasons: on the one hand we believe lighter architectures could be enough to encode radargram information due to the fact that radargram patches contain a low number of classes with respect to common segmentation benchmarking datasets. On the other hand, a lighter encoder allows for faster training time and potentially avoids overfitting on small datasets (which may be the case for $\mathcal{D}_1$).

2) Dataset $\mathcal{D}_2$: The segmentation performance of the proposed method on the MCoRDS3 dataset reported in Table III showcases varying degrees of accuracy across different classes.

The method exhibits high precision and recall values for free space segmentation, indicating a minimal false positive rate and an effective modelling of most instances of free space within the radar data, as well as correctly identifying the boundary between free space and the underlying ice (see Figure 6c).

For inland ice classification, the method demonstrates high precision and recall scores of 0.96 and 0.95. Despite a slightly lower precision compared to free space, the method still

maintains a balanced F1-score of 0.96, indicating overall effectiveness in identifying inland ice regions. As depicted in Figure 6c, also the contour separating noise and ice region is correctly identified, with the sole exception of the central radargram, where the EFZ and the ice layering are difficult to distinguish.

The F1 score for bedrock is 0.66. Nonetheless, recalling the unsupervised nature of the learned patch representations and the considerably low support of the bedrock class, results are very promising. As we can see in Figure 6c, despite some misinterpretation of the bedrock line thickness and shape significantly impacts the recall score, the upper bound of the bedrock is correctly identified in the three radargrams, confirming the capability of model to act as a layer tracker, especially when the interface between bedrock and EFZ presents an high contrast. Overall, the bedrock is properly identified by the model and the main flaws occur near the change points and where the line presents an high slope (i.e. the feature is less horizontal). The detection of change points via the metric presented in IV-C helped at characterising the bedrock pixels up to change points, as we can see in Figure 6c near blue dashed lines. In Figure 6d we can see the horizontal correlation metric computed on the radargrams as described in Section IV-C. Dark areas represent patches where the horizontal correlation is high, i.e. the transition probabilities of walkers are similar to identities. Conversely, brighter areas represent uncertainty in transitions: they mainly characterise noisy patches and highlight the difficulty of tracking bedrock under floating ice. The dichotomy of dark and bright areas helps at visualising where walkers tends to follow horizontal patterns (i.e. targets for our segmentation methodology) and where walkers gets lost. Moreover, the metric enhance the contrast between floating and inland ice, allowing the technique presented in Section IV-C1 to correctly identify the change columns in the radargrams. Relatively large value of $r = 60$ helped at mitigating the effect of high slopes.

The method performs well in floating ice classification, demonstrating high precision and recall scores of 0.90. This indicates its ability to accurately identify floating ice regions within the radargram and correctly separate them from the free space in the upper part of the radargram and the noise in the lower part, as we can see in Figure 6c. The overall thickness of the ice can be seen in the qualitative results and the main flaws occur in segments where bright noisy columns hide the the line of floating ice. Moreover, the detection of change points and the consequent reiteration of label propagation helps at separating the in-land and floating ice classes, successfully providing the model with a new reference label right after the change point. Finally, noise is correctly characterised and the segmentation of noise class leads to an F1-score of 0.96, with balanced values between precision and recall. The main flaws occur in regions where bedrock has high slopes.

The comparative results shown in Table V are in accordance with the ones obtained for $\mathcal{D}_1$. Our methodology demonstrate competitiveness compared to the other supervised methods in the literature and outperforms the fully supervised U-net baseline from [15]. It is worth noting again the lightweight design of our encoder (5M parameters) and our utilization of labels solely during inference, distinguishing our methodology from fully supervised architectures. The model benefits from weak supervision (i.e. reference labels) by the user during label propagation and from the automatic detection of change points via the horizontal correlation metric.

3) Dataset $\mathcal{D}_3$: The segmentation performance of the proposed method on the SHARAD dataset is reported in Table IV.

The method exhibits high accuracy metrics for the segmentation of free space, ice layers and noise, indicating an effective characterization of the three most prominent classes and a correct detection of the air/ice boundary, as we can see in the qualitative results shown in Figure 7b. A relatively high value of $r = 10$ helped at characterising the fast varying shape of the ice surface. This is particularly evident at the edges of the polar deposit in the first two radargrams of Figure 7b. Moreover, we can see how the model effectively captures the position and shape of the basal layer, which in the case of SHARAD is coarse and spread on multiple rows. We see relatively low accuracy metrics for the basal layer in Table IV and this is mainly due to the inherent uncertainty in its extent and the related uncertain characterization by the human labelling process. The performance on strong reflectors is promising: label propagation is correctly able to follow the uppermost bright line of the surface and distinguish it from the surrounding free space and noise. The strong ice reflectors over the polar cap are generally correctly identified and the model detects some areas where weak reflectors are present in-between stronger ones. The main flaws in the accuracy values regarding strong and weak returns may again arise from the uncertainty in their extent. We mark as a promising achievement the fact that the model is able to distinguish between weak and strong reflectors in a finer way than the reference maps. The depictions of the correlation metric in Figure 7d correctly highlights the horizontal features present in the radargrams, with particular emphasis on the surface, the basal layer and small strong reflectors characterizing the ice layering.

The comparative results shown in Table V are again promising: the methodology manages to overcome the results of the vanilla U-Net and the accuracy of the proposed technique is close to the one of the fully supervised SVM.

E. Ablation studies

We perform ablation on the main hyperparameters characterising the methodology. In particular we investigate on (1) the dimension of patches, (2) the overlapping of patches and (3) the radius of label propagation. We evaluate the performance varying the above mentioned parameters by measuring the F1 score on the bedrock class. We use it because the bedrock is the most challenging target to classify in our datasets and its score varies significantly when changing the parameters, while the overall accuracy remains relatively stable due to the low prior probability of this target class.

We empirically found out that a value of temperature of $\tau = 0.01$ all across the methodology provides the best results. Lower values (e.g. 0.001) are possible too. Higher order values lead to missed convergence in the training step and irregular

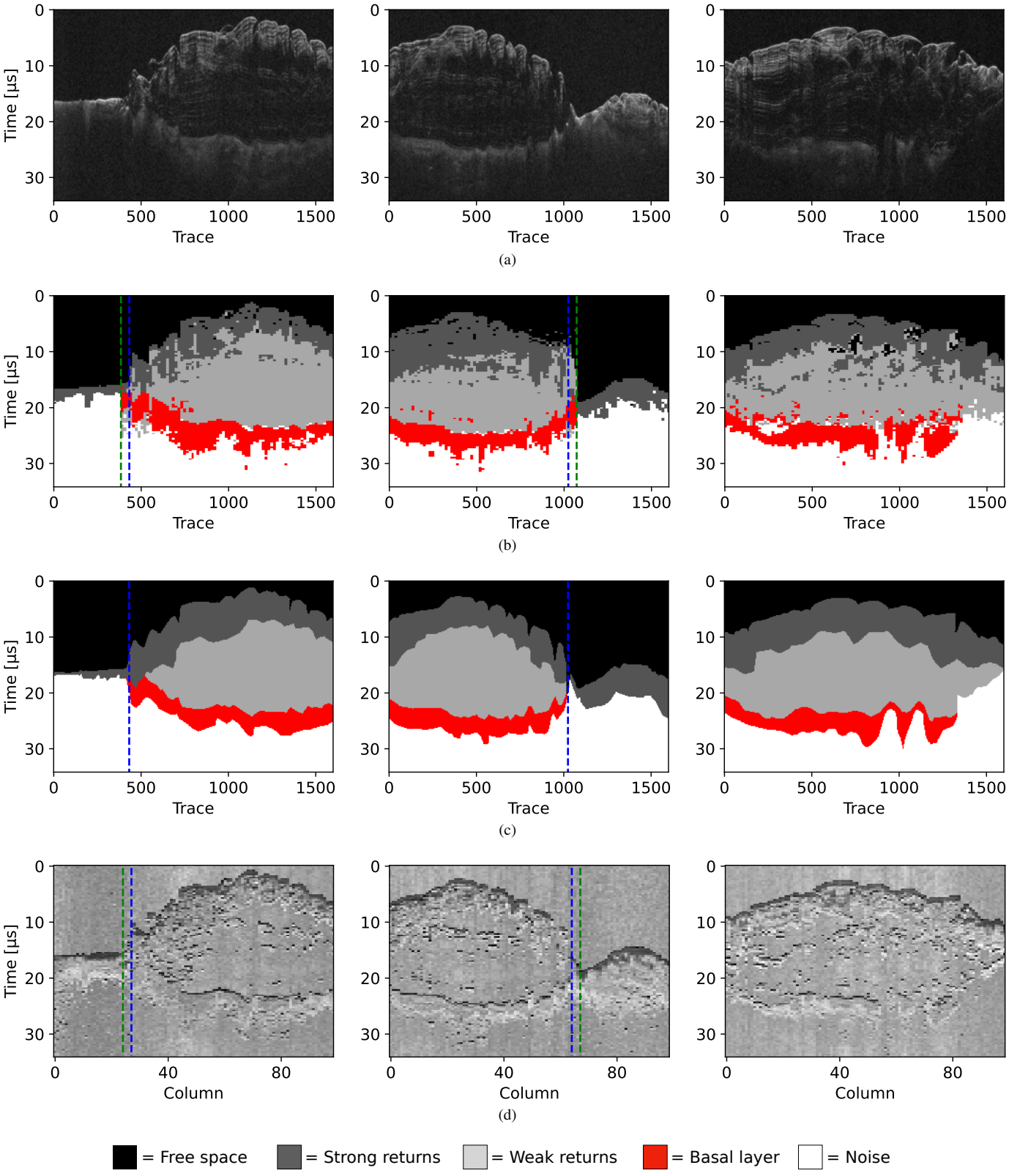
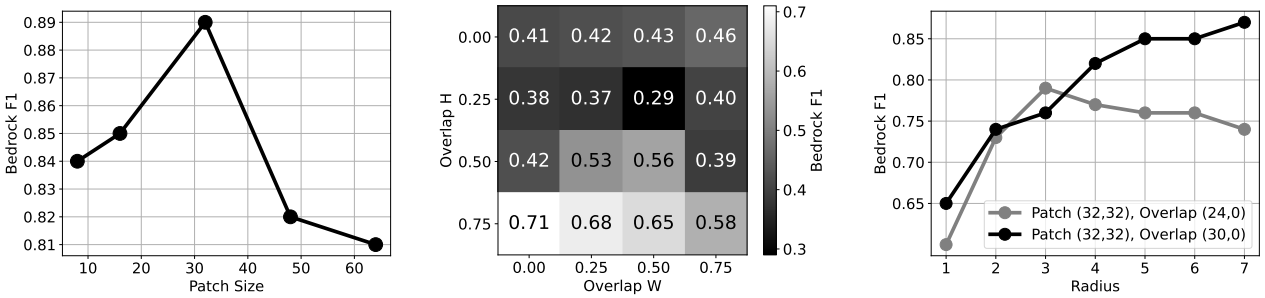


Figure 7. Results of the methodology applied to radargrams from dataset $\mathcal{D}_3$. a) Radargrams from SHARAD North polar region. b) Result of the segmentation using the proposed methodology. c) Segmentation reference maps. d) Horizontal correlation metric computed as in Section IV-C where bright pixels correspond to high cross-entropy according to (20). We can clearly see dark pixels accentuating the main horizontal features. We highlight with green and blue dashed lines the change points found using the technique presented in Section IV-C and according to the ground truth maps, respectively. The resulting segmentation maps correctly follow the main horizontal features within the radargrams. In particular, the label propagation is able to discern the basal layer from surrounding noise and captures strong ice reflections which have not been discerned by the reference maps.



(a) (b) (c)

Figure 8. Results of the ablation studies on $\mathcal{D}_1$. (a) Bedrock F1 score vs. patch size. (b) Bedrock F1 score varying the overlapping between patches. (c) Bedrock F1 score vs. label propagation radius

predictions in the label propagation step. This result is in agreement with the literature [46].

Regarding the context size in the label propagation step, our results are consistent with prior research [41] where an higher context size leads to better results and the limiting factors are the memory size and the total length of the sequence. We suggest to stick to a context length equal to T_{seq} when possible and always including the initial user-provided column in the context bank.

Finally, we find no particular variation in performance when modifying the value of k within kNN propagation. This is consistent with the fact that, under the assumption that meaningful representations are provided by the encoder, even few neighbors are enough to characterise a patch. Extreme values are consistent with literature [47]: $k = 1$ leads to irregular predictions and large k would lead too boundaries which are too smooth and lose high frequency information. The latter case is limited by the radius parameter, in our implementation.

1) Patch dimension: We test our model with different square patch dimensions, namely $(w, h) \in \{(8, 8), (16, 16), (32, 32), (48, 48), (64, 64)\}$. We change the patch dimension during the training step and fix a value of half the patch size for overlapping. We set the other parameters as in Section V-B and perform training and testing on dataset $\mathcal{D}_1$. Results of the ablation (Figure 8a) are in agreement to our interpretation, in particular small patch sizes achieve worse results since they can not encapsulate enough pixel information. We found that the bedrock line has a thickness of around 20 to 30 pixels. Hence a patch of dimension $(8, 8)$ or $(16, 16)$ does not manage to contain its full extent and potentially a glance of its slope. Moreover, a patch size which is too small fails at encoding inter-dependencies between medium-spaced subsurface features, e.g. multiple ice layers and fragmented bedrock lines. On the other hand, a patch size of $(48, 48)$ and $(64, 64)$ tends to encapsulate too many pixels and we argue it makes the graph lose its sequential pattern.

2) Overlapping: We vary overlapping during both training and label propagation on dataset $\mathcal{D}_1$. To maintain consistency we use the same patch dimension and overlapping in both training and label propagation. We fix the radius parameter of label propagation to $r = 16$ and the patch dimension to $(w, h) = (32, 32)$. The overlapping is tested in both range and azimuth direction using percentage values of $\{0.00, 0.25, 0.50, 0.75\}$ with respect to the patch size. We report a grid with the overall accuracy of each combination in Figure 8b. We argue that an high overlap in the range direction is beneficial for the model so it allows for more precise trajectories for the random walkers and a finer segmentation in the label propagation step. On the other hand, a low-to-zero overlap leads to a fast but coarse segmentation which may be useful to get a glance on the radargram aspect but does not match the performance of higher overlapping.

3) Radius: We fix patch dimension and overlapping in both training and label propagation and we vary the radius r of label propagation in the range of discrete values $\{1, 2, ..., 7\}$. We perform the ablation on two different configurations: (1) Patch size $(w, h) = (16, 16)$ and overlap $(o_w, o_h) = (12, 0)$, (2) Patch size $(w, h) = (32, 32)$ and overlap $(o_w, o_h) = (24, 0)$. We argue radius is strictly related to overlap in the range direction: in particular higher overlap requires higher radius to allow walkers to follow slope-varying features correctly. Furthermore, a radius which is too high may allow the model to choose between too many patches; this can lead to an highly irregular segmentation. Results of the ablation are shown in 8c. We clearly see how an high radius is needed to get accurate results on models with high vertical overlap. This is due to the fact that high overlap translates in more patches per column, which again translates in more patches to cover an high vertical steepness in subsurface features. Low radius values generally lead to segmentation maps which are more smooth in the horizontal direction, often losing track of slope-varying features.

VI. Conclusion

In this paper, we have presented a novel methodology for the segmentation of radar sounder data which utilizes a graph built on radargrams according to similarity between patches. Random walks are used to train in an unsupervised way an encoder at learning meaningful representation of patches which encapsulate information on horizontal structures within the subsurface and label propagation is later exploited to propagate the labels of a reference segmentation map to the full radargram. For the first time, we introduced an unsupervised training phase which is tailored to the sequential structure of radargrams.

We exploit distinctive characteristics that set radargrams apart from conventional images, especially the strong horizontal correlation between features and the relational features that exists among classes. We validated the methodology on three datasets from different instruments and results depict the high degree of fidelity between predictions and ground truth. In particular, the model is able to track the main horizontally-extended areas and boundaries between classes by propagating a single reference segmentation, possibly user-provided in real time. This sets apart our methodology from previous deep learning literature in the radar sounder field: we exploit labels of small fractions of the dataset and we make use of them only at test time. This has two main advantages. Firstly, it does not tailor the training step to labels, enabling a fully unsupervised training and the learning of features that are not aligned to any specific segmentation map. Secondly, we allow users to choose the initial segmentation in the test phase, allowing for multiple "correct" segmentation maps to be propagated on a single radargram. Comparative results shows how our unsupervised training step followed by label propagation guided by users leads to results that are comparable to state-of-the-art supervised techniques while employing lighter encoders and allowing flexibility in the segmentation map. We strongly believe deep learning could help experts in the field of radar sounders to process information automatically and we stand in the direction of a methodology that can bridge the gap between a fully manual analysis and a deep learning-driven technique. We argue that inference tailored to user-provided prompts can lead to results which are more explainable and useful under the eyes of experts. In future research, we aim at exploring the applicability of a sequence-based unsupervised training on larger datasets composed by radargrams from multiple instruments, in order to obtain a foundation model for the radar sounder field which extends its capabilities to multiple use cases beyond semantic segmentation.

ACKNOWLEDGMENT

This work was supported by the Italian Space Agency under Contract n. 2022-23-HH.0 (CUP: F63C22000650005) "Attività scientifiche per il radar sounder di EnVision fase B1".

We acknowledge the use of data and/or data products from CReSIS generated with support from the University of Kansas, NSF grant ANT-0424589, and NASA Operation IceBridge grant NNX16AH54G.

REFERENCES

[1] R. Orosei, S. E. Lauro, E. Pettinelli, A. Cicchetti, M. Coradini, B. Cosciotti, F. Di Paolo, E. Flamini, E. Mattei, M. Pajola *et al.*, "Radar evidence of subglacial liquid water on mars," *Science*, vol. 361, no. 6401, pp. 490–493, 2018.

[2] L. Shi, C. T. Allen, J. R. Ledford, F. Rodriguez-Morales, W. A. Blake, B. G. Panzer, S. C. Prokopiack, C. J. Leuschen, and S. Gogineni, "Multichannel coherent radar depth sounder for nasa operation ice bridge," in *2010 ieee international geoscience and remote sensing symposium*. IEEE, 2010, pp. 1729–1732.

[3] F. Rodriguez-Morales, S. Gogineni, C. J. Leuschen, J. D. Paden, J. Li, C. C. Lewis, B. Panzer, D. G.-G. Alvestegui, A. Patel, K. Byers *et al.*, "Advanced multifrequency radar instrumentation for polar research," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2824–2842, 2013.

[4] R. Jordan, G. Picardi, J. Plaut, K. Wheeler, D. Kirchner, A. Safaeinili, W. Johnson, R. Seu, D. Calabrese, E. Zampolini *et al.*, "The mars express marsis sounder instrument," *Planetary and Space Science*, vol. 57, no. 14-15, pp. 1975–1986, 2009.

[5] R. Seu, R. J. Phillips, D. Biccari, R. Orosei, A. Masdea, G. Picardi, A. Safaeinili, B. A. Campbell, J. J. Plaut, L. Marinangeli *et al.*, "Sharad sounding radar on the mars reconnaissance orbiter," *Journal of Geophysical Research: Planets*, vol. 112, no. E5, 2007.

[6] L. Bruzzone, J. J. Plaut, G. Alberti, D. D. Blankenship, F. Bovolo, B. A. Campbell, A. Ferro, Y. Gim, W. Kofman, G. Komatsu *et al.*, "Rime: Radar for icy moon exploration," in *2013 IEEE international geoscience and remote sensing symposium-IGARSS*. IEEE, 2013, pp. 3907–3910.

[7] L. Bruzzone, F. Bovolo, S. Thakur, L. Carrer, E. Donini, C. Gerekos, S. Paterna, M. Santoni, and E. Sbalchiero, "Envision mission to venus: Subsurface radar sounding," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 5960–5963.

[8] C. M. Gifford, G. Finyom, M. Jefferson, M. Reid, E. L. Akers, and A. Agah, "Automated polar ice thickness estimation from radar imagery," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2456–2469, 2010.

[9] A. Ferro and L. Bruzzone, "Analysis of radar sounder signals for the automatic detection and characterization of subsurface features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4333–4348, 2012.

[10] ——, "Automatic extraction and analysis of ice layering in radar sounder data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 3, pp. 1622–1634, 2013.

[11] A.-M. Ilisei and L. Bruzzone, "A system for the automatic classification of ice sheet subsurface targets in radar sounder data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3260–3277, 2015.

[12] A.-M. Ilisei, M. Khodadadzadeh, A. Ferro, and L. Bruzzone, "An automatic method for subglacial lake detection in ice sheet radar sounder data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3252–3270, 2018.

[13] E. Donini, M. Amico, L. Bruzzone, and F. Bovolo, "Unsupervised semantic segmentation of radar sounder data using contrastive learning," in *Image and Signal Processing for Remote Sensing XXVIII*, vol. 12267. SPIE, 2022, pp. 171–180.

[14] R. Ghosh and F. Bovolo, "Transsounder: A hybrid transunet-transfuse architectural framework for semantic segmentation of radar sounder data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[15] E. Donini, F. Bovolo, and L. Bruzzone, "A deep learning architecture for semantic segmentation of radar sounder data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[16] M. H. García, E. Donini, and F. Bovolo, "A weakly supervised transfer learning approach for radar sounder data segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

[17] G. Steinbrügge, M. S. Haynes, D. M. Schroeder, K. M. Scanlan, A. Stark, D. A. Young, C. Grima, S. Kempf, G. Ng, D. Buhl, J. R. C. Voigt, T. Roatsch, and D. D. Blankenship, "Altimetry measurements from planetary radar sounders and application to sharad on mars," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[18] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[21] R. Ghosh and F. Bovolo, "A hybrid cnn-transformer architecture for semantic segmentation of radar sounder data," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 1320–1323.

[22] A. Ghiasi, H. Kazemi, E. Borgnia, S. Reich, M. Shu, M. Goldblum, A. G. Wilson, and T. Goldstein, "What do vision transformers learn? a visual exploration," *arXiv preprint arXiv:2212.06727*, 2022.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings*

of the 25th international conference on Machine learning, 2008, pp. 1096–1103.
[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.
[25] S. Zhai, N. Jaitly, J. Ramapuram, D. Busbridge, T. Likhomanenko, J. Y. Cheng, W. Talbott, C. Huang, H. Goh, and J. Susskind, "Position prediction as an effective pretraining strategy," *arXiv preprint arXiv:2207.07611*, 2022.
[26] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*. PMLR, 2015, pp. 843–852.
[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[28] S. Saha, F. Bovolo, and L. Bruzzone, "Change detection in image time-series using unsupervised lstm," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.
[29] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.
[30] B. A. Campbell, N. E. Putzig, L. M. Carter, and R. J. Phillips, "Autofocus correction of phase distortion effects on sharad echoes," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 5, pp. 939–942, 2011.
[31] L. Carrer and L. Bruzzone, "Automatic enhancement and detection of layering in radar sounder data based on a local scale hidden markov model and the viterbi algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 962–977, 2016.
[32] P. Song, S. Gogineni, I. Galkin, J. Volakis, J. Soderblom, A. Hayes, B. Reinisch, R. Giles, R. Sood, H.-L. Zhang *et al.*, "Feasibility study of a high-resolution shallow surface penetration radar for space application," *Radio Science*, vol. 56, no. 2, pp. 1–20, 2021.
[33] X. Liu and W. Fa, "A fully automatic algorithm for reflector detection in radargrams based on continuous wavelet transform and minimum spanning tree," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–20, 2022.
[34] S. Xiong and J.-P. Muller, "Automated reconstruction of subsurface interfaces in promethei lingula near the martian south pole by using sharad data," *Planetary and Space Science*, vol. 166, pp. 59–69, 2019.
[35] M. Berens, M. Baroni, Z. Kiss, S. Verstaen, and J. Lange, "An ambitious journey to superior emc performance for the juice mission," in *2022 ESA Workshop on Aerospace EMC (Aerospace EMC)*. IEEE, 2022, pp. 1–6.
[36] B. Cecconi, S. Hess, A. Hérique, M. R. Santovito, D. Santos-Costa, P. Zarka, G. Alberti, D. Blankenship, J.-L. Bougeret, L. Bruzzone *et al.*, "Natural radio emission of jupiter as interferences for radar investigations of the icy satellites of jupiter," *Planetary and Space Science*, vol. 61, no. 1, pp. 32–45, 2012.
[37] D. Gurnett, W. Kurth, A. Roux, R. Gendrin, C. Kennel, and S. Bolton, "Lightning and plasma wave observations from the galileo flyby of venus," *Science*, vol. 253, no. 5027, pp. 1522–1525, 1991.
[38] A. Jabri, A. Owens, and A. Efros, "Space-time correspondence as a contrastive random walk," *Advances in neural information processing systems*, vol. 33, pp. 19 545–19 560, 2020.
[39] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2566–2576.
[40] ——, "Learning correspondence from the cycle-consistency of time," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2566–2576.
[41] Z. Lai, E. Lu, and W. Xie, "Mast: A memory-augmented self-supervised tracker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6479–6488.
[42] A. J. Scott and M. Knott, "A cluster analysis method for grouping means in the analysis of variance," *Biometrics*, pp. 507–512, 1974.
[43] B. Campbell, "Us shallow radar (sharad) data product description for the planetary data system," *Smithsonian Institute: Washington, DC, USA*, 2014.
[44] CReSIS, "Cresis radar depth sounder data," Lawrence, Kansas, USA. Digital Media., 2016. [Online]. Available: http://data.cresis.ku.edu/
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[46] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531
[47] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, vol. 2, pp. 5–43, 2006.

PLACE PHOTO HERE

Jordy Dal Corso Jordy Dal Corso received his Bachelor's degree in Applied Mathematics and his Master's degree in Data Science from the University of Verona, Italy, in 2020 and 2022 respectively. He is a former member of the Machine and Human Intelligence Laboratory at the University of Helsinki and he is currently pursuing a Ph.D. degree as a member of the Remote Sensing Laboratory at the Department of Information Engineering and Computer Science, University of Trento, Italy. His research interests are related to deep learning applied to efficient and automated analysis of data acquired from planetary radars.

PLACE PHOTO HERE

Lorenzo Bruzzone Lorenzo Bruzzone received the Laurea (M.S.) degree in electronic engineering (summa cum laude) and the Ph.D. degree in telecommunications from the University of Genoa, Italy, in 1993 and 1998, respectively. He is currently a Full Professor of telecommunications at the University of Trento, Italy, where he teaches remote sensing, radar, and digital communications. Dr. Bruzzone is the founder and the director of the Remote Sensing Laboratory in the Department of Information Engineering and Computer Science, University of Trento. His current research interests are in the areas of remote sensing, radar and SAR, signal processing, machine learning and pattern recognition. He promotes and supervises research on these topics within the frameworks of many national and international projects. He is the Principal Investigator of many research projects. Among the others, he is currently the Principal Investigator of the Radar for icy Moon exploration (RIME) instrument in the framework of the JUpiter ICy moons Explorer (JUICE) mission of the European Space Agency (ESA) and the Science Lead for the High Resolution Land Cover project in the framework of the Climate Change Initiative of ESA. He is the author (or coauthor) of 247 scientific publications in referred international journals (183 in IEEE journals), more than 310 papers in conference proceedings, and 21 book chapters. He is editor/co-editor of 18 books/conference proceedings and 1 scientific book. His papers are highly cited, as proven from the total number of citations (more than 31500) and the value of the h-index (83) (source: Google Scholar). He was invited as keynote speaker in more than 32 international conferences and workshops. Since 2009 he has been a member of the Administrative Committee of the IEEE Geoscience and Remote Sensing Society (GRSS), where since 2019 he has been Vice-President for Professional Activities. Dr. Bruzzone ranked first place in the Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (Seattle, July 1998). Since that he was recipient of many international and national honors and awards, including the recent IEEE GRSS 2015 Outstanding Service Award, the 2017 IEEE IGARSS Symposium Prize Paper Award and the 2018 IEEE IGARSS Symposium Prize Paper Award, and the 2019 WHISPERS Outstading Paper Award. In the past years joint papers presented by his students at international symposia and Master/PhD theses that he supervised have received awards. Dr. Bruzzone is the co-founder of the IEEE International Workshop on the Analysis of Multi-Temporal Remote-Sensing Images (MultiTemp) series and is currently a member of the Permanent Steering Committee of this series of workshops. Since 2003 he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. He was a Guest Co-Editor of many Special Issues of international journals. From 2010 to 2012 he was the Editor-in-Chief of the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER. He has been the founder of the IEEE GEOSCIENCE AND REMOTE SENSING MAGAZINE for which he has been the Editor-in-Chief between 2013 and 2017. From 2004 to 2006 he served as an Associated Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. Currently he is an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He has been Distinguished Speaker of the IEEE Geoscience and Remote Sensing Society between 2012 and 2016. He is a Fellow of IEEE and a member of the International Association for Pattern Recognition (IAPR) and of the Italian Association for Remote Sensing (AIT).