PixelDINO: Semi-Supervised Semantic Segmentation for Detecting Permafrost Disturbances in the Arctic

Konrad Heidler, *Student Member, IEEE*, Ingmar Nitze, Guido Grosse, and Xiao Xiang Zhu, *Fellow, IEEE*

***Abstract*— Arctic permafrost is facing significant changes due to global climate change. As these regions are largely inaccessible, remote sensing plays a crucial rule in better understanding the underlying processes across the Arctic. In this study, we focus on the remote detection of retrogressive thaw slumps (RTSs), a permafrost disturbance comparable to slow landslides. For such remote sensing tasks, deep learning has become an indispensable tool, but limited labeled training data remains a challenge for training accurate models. We present PixelDINO, a semi-supervised learning approach, to improve model generalization across the Arctic with a limited number of labels. PixelDINO leverages unlabeled data by training the model to define its own segmentation categories (pseudoclasses), promoting consistent structural learning across strong data augmentations. This allows the model to extract structural information from unlabeled data, supplementing the learning from labeled data. PixelDINO surpasses both supervised baselines and existing semi-supervised methods, achieving average intersection-over-union (IoU) of 30.2 and 39.5 on the two evaluation sets, representing significant improvements of 13% and 21%, respectively, over the strongest existing models. This highlights the potential for training robust models that generalize well to regions that were not included in the training data.**

***Index Terms*— Permafrost, retrogressive thaw slumps (RTSs), self-distillation without labels, semantic segmentation, semi-supervised learning.**

Manuscript received 25 October 2023; revised 18 March 2024 and 20 June 2024; accepted 16 July 2024. Date of publication 22 August 2024; date of current version 3 September 2024. This work was supported by German Federal Ministry for Economic Affairs and Climate Action within the framework of the "National Center of Excellence ML4Earth" under Grant 50EE2201C. The work of Ingmar Nitze and Guido Grosse was supported in part by the Helmholtz-Gemeinschaft deutscher Forschungszentren (HGF) "Artificial Intelligence for Cold Regions" (AI-CORE) Projects; and in part by the Permafrost Discovery Gateway under Grant NSF2052107, Grant NSF1927872, and Google.org Impact Challenge PDG Grant. The work of Xiao Xiang Zhu was supported in part by the Bundesministerium für Bildung und Forschung (BMBF) Future Lab Artificial Intelligence for Earth Observation (AI4EO) under Grant 01DD20001 and in part by Munich Center for Machine Learning (MCML). *(Corresponding author: Konrad Heidler.)*

Konrad Heidler is with the Chair of Data Science in Earth Observation (SiPEO), Department of Aerospace and Geodesy, School of Engineering and Design, Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: k.heidler@tum.de).

Ingmar Nitze is with the Permafrost Research Section, Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research, 14473 Potsdam, Germany (e-mail: ingmar.nitze@awi.de).

Guido Grosse is with the Permafrost Research Section, Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research, 14473 Potsdam, Germany, and also with the Institute of Geosciences, University of Potsdam, 14469 Potsdam, Germany (e-mail: guido.grosse@awi.de).

Xiao Xiang Zhu is with the Chair of Data Science in Earth Observation (SiPEO), Department of Aerospace and Geodesy, School of Engineering and Design and Munich Center for Machine Learning, Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: xiaoxiang.zhu@tum.de).

Digital Object Identifier 10.1109/TGRS.2024.3448294

I. Introduction

IN STEP with global climate change, permafrost is changing rapidly. Rising temperatures in the Arctic have large implications for perennially frozen soil which can destabilize upon the thawing of ice-rich ground. Owing to their remoteness and sparse population, permafrost areas are often difficult to access physically. Therefore, in situ measurements are only available for specific study sites at specific dates when expeditions visited that site or when data are collected through local sensors [1]. Therefore, remote sensing techniques are a valuable method that can monitor permafrost on a pan-Arctic scale, and a useful approach for upscaling and understanding of broad spatiotemporal dynamics of permafrost thaw processes [2], [3]. To further improve the efficiency of remote sensing monitoring for these applications, machine learning techniques offer great potential in automating laborious annotation tasks.

Permafrost is generally a subsurface phenomenon, making it difficult to observe from satellite observations. Other than permafrost itself, permafrost degradation landforms like retrogressive thaw slumps (RTSs) are visible in optical satellite imagery due to their distinct shape and spectral signature compared with the surrounding regions. This makes them a viable target of study via remote sensing methods. RTSs are mass movements akin to slow-flowing landslides caused by the melting of massive ground ice in permafrost regions [4]. RTSs are rather small features generally measuring less than 10 ha in area [5], [6], with some notable exceptions, so-called megaslumps, exceeding 40 ha [7]. RTSs form due to specific local environmental conditions such as slope, landscape history, ground temperature, and disturbances [4]. They typically occur in glacial moraines with preserved remnant glacial ice, syngenetic ice-rich yedoma permafrost, or marine deposits, which were raised due to isostatic uplift [8]. Understanding and quantifying RTS dynamics is important as they pose potential hazards to infrastructure [9], directly affect water quality in downstream aquatic environments [10], and locally mobilize large amounts of formerly frozen sediment and organic matter [8].

Machine learning, specifically deep learning, can automate the identification of RTSs from satellite imagery. Existing studies often achieve mixed results, which in many cases can be attributed to the algorithms' requirements for an extensive collection of labeled training data that is hard to acquire in large volumes [11], [12], [13], [14], [15]. While decent

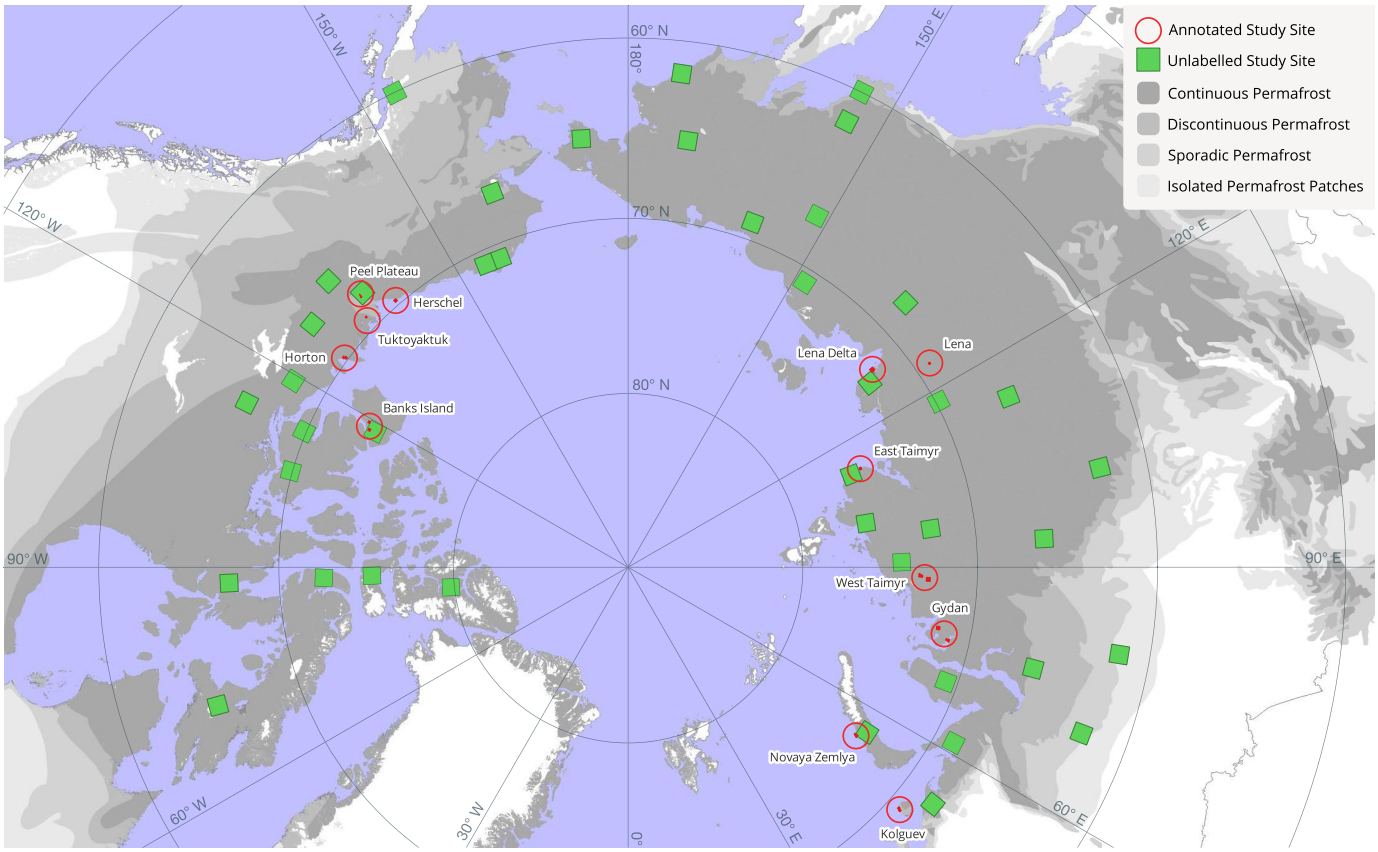

Fig. 1. Spatial distribution of the annotated training sites (red). It can be seen that the labeled data have quite limited spatial coverage. By using semi-supervised learning, it is possible to include large areas of unlabeled Sentinel-2 imagery (green) in the training process. Basemap source: [16].

prediction results are obtained for selected study sites, accurate pan-Arctic generalization remains an elusive goal [12], [15].

This study explores how to make models better generalize to previously unseen regions. While increasing the available training data through additional labeling efforts are one option, it comes at a large labor cost for the involved domain experts. In an attempt to tackle this issue from a methodological angle instead, we explore semi-supervised learning for improving model performance without the need for additional annotated training data. In classical *supervised learning*, a model is trained on labeled data only. In contrast to this, *self-supervised learning* aims to train models without any labels. Combining these two paradigms, *semi-supervised learning* trains models on both labeled and unlabeled data at the same time [17], [18]. This strategy allows for the inclusion of unlabeled satellite imagery into the training process. While labeling is a laborious task, the underlying satellite imagery is openly available. Therefore, semi-supervised learning methods are exceptionally well-suited for remote sensing tasks.

In this study, we propose a new framework for semi-supervised semantic segmentation called *PixelDINO*. Our framework builds on the successful self-supervised learning framework DINO [19], which was originally developed to learn features for image classification. The main idea behind DINO is *self-distillation with no labels*, which is a special case of knowledge distillation. In *knowledge distillation*, a model is trained to closely match another model's outputs in order to transfer learned knowledge from one model to another. Self-distillation with no labels describes distillating a model's knowledge into itself while applying certain transformations to the data [19]. We adopt this idea to pixelwise prediction tasks like semantic segmentation and then combine it with a regular supervised learning procedure into a semi-supervised learning framework.

As shown in Fig. 1, the spatial coverage of the Arctic can be greatly improved for RTS detection by including unlabeled data in a semi-supervised fashion. Using this dataset, we present experimental results for the task of RTS detection, where we demonstrate that PixelDINO outperforms both supervised baseline methods and other semi-supervised semantic segmentation approaches.

II. Related Works

In order to place our contributions into a larger scientific context, this section summarizes existing research on monitoring RTSs with remote sensing, and gives an overview of representation learning and semi-supervised segmentation methods in remote sensing.

A. Monitoring Retrogressive Thaw Slumps

As permafrost cannot be directly seen from space, many permafrost remote sensing studies focus instead on monitoring specific targets that are known or assumed to be correlated with the state of permafrost or its vulnerability [3]. Spatially consistent monitoring of specific permafrost degradation landforms with high temporal resolution is a desirable goal, since it

would allow assessments regarding the vulnerability of local infrastructure and the biogeochemical implications of rapid permafrost thaw for both the local environment and the global climate system [12].

The detection of such features in satellite imagery is not without challenges. RTSs in permafrost regions are often hard to detect due to their widespread distribution, small size, and their varying stages of activity [12], [15]. Furthermore, optical remote sensing is inhibited by snow cover, cloud cover, and polar night for large parts of the year, so that features can only be reliably detected during the summer months [3].

Regarding data sources, permafrost disturbances can be mapped using different remote sensing approaches, such as optical image analysis [12], optical time-series analysis [20], surface elevation data [21], or interferometric synthetic aperture radar (InSAR) measurements [22].

Many studies rely on the manual digitization of permafrost disturbance landforms in satellite imagery [23], [24]. While this approach ensures good accuracy, it quickly becomes infeasible when the study areas grow beyond small- to medium-sized regions. In order to automate the laborious manual digitization process, some studies explored computer vision methods like trend analyses combined with random forests [8], or graph-based analysis [25].

With deep learning becoming an indispensable tool in remote sensing, it was also used for the detection of RTS features. Huang et al. [11] adapted the DeepLab architecture for semantic segmentation [26] to the task of mapping permafrost features like RTSs using imagery from unmanned aerial vehicles (UAVs) over the northeastern Tibetan Plateau. Similarly, Nitze et al. [12] trained several convolutional neural network (CNN) architectures on PlanetScope satellite imagery for six study sites in northwest Canada and the Russian Arctic. Yang et al. [15] combine Maxar imagery with other information like NDVI derived from Sentinel-2 and elevation information to train a CNN model to detect RTS. Huang et al. [21] opted to detect RTS directly in elevation maps instead, training an object detector on the ArcticDEM data product.

Existing studies usually focus on a single region of interest, like the Canadian Arctic [13], the Tibetan Plateau [11], [27], or a few selected regions [8], [12], [15]. More recently, efforts toward a pan-Arctic RTS data product have gained traction [21].

Other permafrost features can also be mapped using remote sensing techniques, including thermokarst lakes [8], [28], wildfires [8], [29], and ice wedges [25], [30], [31]. These research areas face similar challenges as RTS mapping, so that approaches for these tasks can also inspire new approaches for RTS mapping.

B. Self-Supervised Representation Learning

Learning features from unlabeled images has been a highly active area of research in recent years. As acquiring images is relatively simple compared with labeling them, self-supervised methods seek to train models without any labels. Still, the features derived by such models often compare competitively to fully supervised models in evaluations [32], [33], [34], [35].

Most approaches train an image encoder to embed images to feature vectors in such a way that the embedding is invariant under certain data augmentations, meaning that perturbed versions of the same image should be represented by the same point in the embedding space [32], [33], [34], [35]. A trivial solution to this goal is reached when the encoder predicts the same constant feature vector for all inputs. Therefore, the main ideas that differentiate these models lie in the way that they address this representation collapse. SimCLR [33] employs the contrastive loss function to not only match embeddings of the same image closely in the representation space but also push apart embeddings from different images. Building on this idea, momentum contrast [35] introduces a momentum encoder that updates its weights as an exponential moving average (EMA) of the trained model's weights. Furthermore, a queue of embeddings is used in order to leverage a larger number of negative samples. Bootstrap your own latent [34] uses the momentum encoder to eliminate the need for negative samples. By carefully tuning the momentum and using a projection head, this method avoids representation collapse without using a contrastive loss.

Finally, self-distillation without labels (DINO) [32] uses a different approach to eliminate negative samples. Here, the model is tasked with defining its own classification scheme for images. Two versions of the model, called student and teacher, are trained following the self-distillation process.

For a given input image, two augmentations are generated. Out of these two augmentations, the first one is run through the teacher model. The features derived from the teacher model are then centered and rescaled. Finally, the teacher's classification is derived by applying a softmax activation to the rescaled outputs. Meanwhile, the second version of the image is run through the student model. Finally, the student is then trained to match the teacher's classifications with its own outputs [19]. Fig. 2 outlines the DINO training process. In the following, we will be referring to the classes automatically derived by the models as "pseudoclasses."

Naturally, one crucial step in this setup is the assignment of parameters to the teacher model. As there are no ground-truth labels in this setup, the teacher weights are taken to be an EMA of the student weights, hence the term "self-distillation."

Other than these methods, our use case does not require image-level features, but rather pixelwise features. With PixelDINO, we adopt the concept of self-distillation with no labels on the pixel level.

C. Semi-Supervised Semantic Segmentation in Remote Sensing

In remote sensing, many relevant tasks are semantic segmentation tasks. For each pixel, a class label needs to be predicted in order to partition the entire scene into separate regions of interest. Such tasks are encountered across a large number of research areas like crop-type mapping [36], urban mapping [37], or monitoring animal populations [38]. Generally, it is quite hard even for experts to perfectly annotate a given scene pixel by pixel, and the process of generating these annotations is often tedious and time-consuming [39].

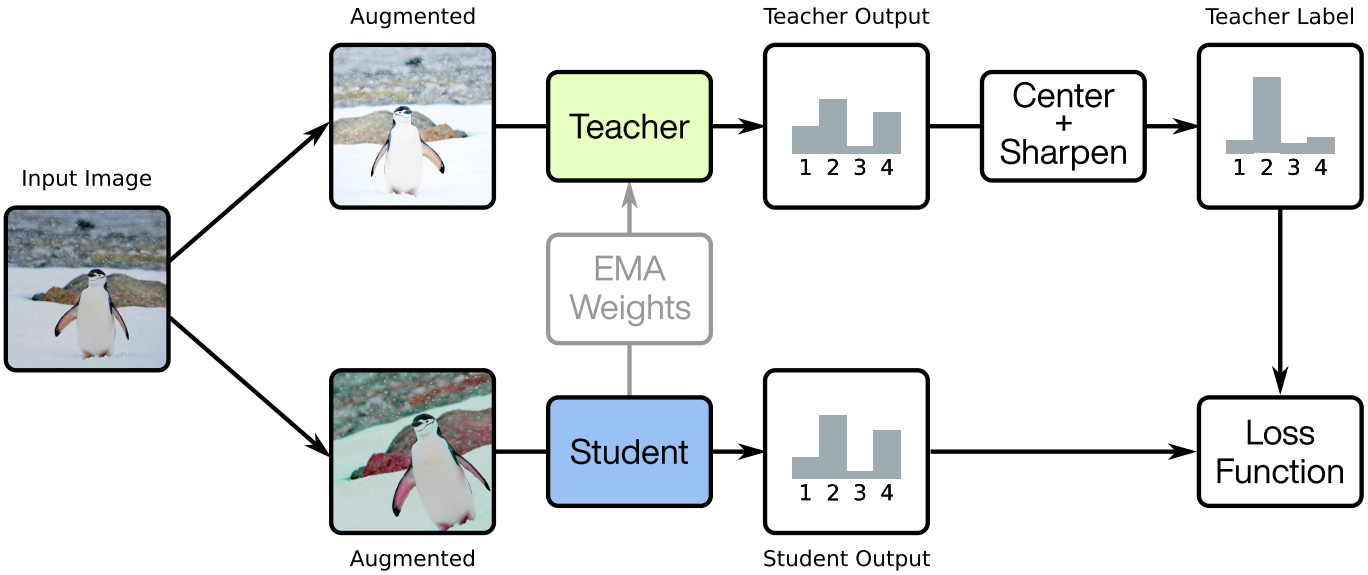

Fig. 2. Overview of the DINO framework [32] for feature learning. Two augmented versions of the input image are generated. The teacher model is then used to predict a class distribution for the first augmentation. This distribution is centered, sharpened, and the softmax function is applied. The student model is then given the second augmented image and trained to predict the label given by the teacher. Finally, the teacher model's weights are updated as an exponential moving average of the student's weights.

There are approaches to reducing the labeling burden through working with sparse labels like point labels or scribbled labels, but these come at a price in terms of classification accuracy [40]. On the other hand, unlabeled remote sensing data is generally easily available through programs like NASA's Landsat series or ESA's Copernicus missions. Therefore, the idea of combining small labeled datasets with large unlabeled data for semantic segmentation has been previously explored in remote sensing.

A large class of semi-supervised learning studies in remote sensing focuses on the idea of consistency regularization. The underlying assumption here is that even for unlabeled images, a model's representations or outputs should be consistent under a certain set of perturbations. For example, these perturbations can be data augmentation operations [41], feature dropout [42], additive noise in the feature space [43], [44], or interpolation between samples [45]. Under these perturbations, the model is then trained to stay consistent. This consistency can be enforced at different stages of the model calculation. Most common is the so-called pseudolabeling technique [42], where consistency is enforced in the final output classification of the network. Various extensions of this basic idea exist [46], [47].

In FixMatch, Sohn et al. [46] enforce consistency across two sets of data augmentatations called *weak augmentations*, denoted by $\alpha(\cdot)$, and *strong augmentations*, denoted by $\mathcal{A}(\cdot)$. Upretee and Khanal [41] formulated *FixMatchSeg*, an elegant way of generalizing this framework to the semantic segmentation case. As the labels themselves are also subject to geometric transformations such as rotations, converting them between augmentations is not trivial. FixMatchSeg solves this by chaining the weak and strong data augmentations as $\mathcal{A}(\alpha(\cdot))$, so that the pseudolabel can be augmented alongside with the image.

Another possibility is to enforce consistency in the intermediate feature space within a given layer of the neural network [43]. Such approaches have been successfully applied for mapping building footprints [43], mapping landslides [48], or aerial image segmentation [49]. Our presented approach is similar to these methods. The main difference in our approach is the change from pseudolabels to pseudoclasses. While pseudolabels are adhering to the original classification scheme of the task, we allow the network to come up with additional classes in order to oversegment the images. This should be particularly helpful for tasks with a large class imbalance, for example, when a background class with high intraclass variance dominates the scenery, which is the case in RTS detection.

The generator–discriminator approach from generative adversarial networks (GANs) has also been explored for semi-supervised semantic segmentation. Here, the basic idea is to conceptually understand the segmentation network as either the generator or the discriminator network. In the first setup, the discriminator learns to discern true segmentation maps from model outputs on a pixelwise level. At the same time, the segmentation network takes the role of the generator and is trained to convince the discriminator as a secondary loss objective [50]. In the other setting, a generator is used to generate synthetic data, and the discriminator is trained to differentiate these synthetic data points from the unlabeled data, while also generating class labels [51]. Adversarial semi-supervised learning approaches have been demonstrated on tasks like hyperspectral image classification [52] or change detection [53]. Other than these works, our method only requires training a single neural network. Also, it does not exhibit the well-known training instabilities or require any of the careful hyperparameter tuning that adversarial methods are known for.

Finally, some studies separate the training process into a self-supervised pretraining phase on a large unlabeled dataset, and a supervised fine-tuning phase on the labeled dataset. As self-supervised learning has been an area of great interest in computer vision recently, this approach is getting increasingly popular. For example, such approaches have been shown to improve model performance for tasks, such as hyperspectral image classification [54], land cover mapping [55], [56], or change detection [56]. Contrasting this, we present a semi-supervised training procedure, where the model is trained end-to-end in a single training phase.

III. PixelDINO for Semi-Supervised Semantic Segmentation

Inspired by the ideas behind DINO [32] and FixMatchSeg [41], we build PixelDINO, a semi-supervised semantic segmentation framework for remote sensing imagery.

A. Learning Pixel Features Without Labels

While natural imagery often has a clear object of focus, a remotely sensed satellite image can have dozens or hundreds of objects of interest in it. Therefore, working on the pixel level should lead to more discriminative features, which will be crucial for a successful segmentation of these objects in the end. The main idea for our PixelDINO framework is to adopt the explained above on a pixelwise level. Instead of classifying entire images, the student and teacher models will instead give a label to each pixel in the input image.

But in the original DINO framework, the teacher labels can be directly applied to train the student. In the pixelwise case, data augmentations like flips or rotations will change the location of objects in the image. Therefore, pixelwise segmentation labels also need to be augmented in the same fashion. When following the original DINO setup, doing this correctly is challenging, as it requires inverting the data augmentations applied to the first image. Furthermore, this procedure will introduce invalid pixel labels when inverting lossy augmentations like rotations by nonmultiples of 90° or cropping operators. To avoid these issues, we resort to an approach introduced by FixMatchSeg [41]. Instead of using two augmentations of the same base image, we will use a chain of augmented images.

Given an unlabeled input image $U \in \mathbb{R}^{H\times W\times C}$, we first apply a weak augmentation $\alpha(U)$ and calculate the teacher output $\mathcal{T}(\alpha(U))$. Then, the teacher's label is derived through centering, rescaling, and applying the softmax function

$$Y_U = \text{softmax}\left(\frac{\mathcal{T}(\alpha(U)) - \mu}{\tau}\right). \tag{1}$$

Here, μ is the center of past teacher outputs, which is updated using an EMA, and τ is the temperature parameter. A lower temperature leads to a stronger "sharpening" of the class distribution, which is desired in order to discourage the model from predicting a uniform distribution.

The student model $\mathcal{S}$ is applied to the strongly augmented input image to obtain the student's prediction $\mathcal{S}(\mathcal{A}(\alpha(U)))$. Finally, the PixelDINO loss is calculated as the cross entropy (CE) between the softmax of the student output and the strongly augmented teacher label

$$\mathcal{L}_{\text{PixelDINO}} = \text{CE}(\text{softmax}(\mathcal{S}(\mathcal{A}(\alpha(U))), \mathcal{A}(Y_U))) \tag{2}$$

where CE refers to the cross-entropy operator.

In this way, the student model $\mathcal{S}$ is trained to align its predictions in such a way that they are consistent with the teacher's outputs $\mathcal{T}$ under the set of strong augmentations $\mathcal{A}$. A graphical overview of this approach is given in Fig. 3.

B. Semi-Supervised Learning With PixelDINO

The goal of semi-supervised learning is to exploit the information present in a large, unlabeled dataset and combine that with the class information from a smaller, labeled dataset. For PixelDINO, embedding the information from a labeled dataset is rather straightforward. The DINO methodology already works with pseudoclasses, and PixelDINO extends that to pseudoclasses per pixel. If information about some specific classes is already known a priori in the form of a labeled dataset, this can be embedded into the training process in order to make the pseudoclasses align with the a priori classes. In our case, we would like to do exactly that for the RTS class from the labeled dataset.

To achieve that, we combine the PixelDINO training loop with a regular supervised training loop. In the combined training loop, the student model will be trained on both a mini-batch of labeled examples, as well as one of the unlabeled examples for each training step. For a labeled example given as a pair of an image $X \in \mathbb{R}^{H\times W\times C}$ and a mask $Y \in \{0, 1\}^{H\times W}$, the supervised loss term is the regular CE which is commonly used in semantic segmentation. In practice, we also apply weak and strong data augmentation to the labeled samples

$$\mathcal{L}_{\text{supervised}}(X, Y) = \text{CE}(\mathcal{S}(\mathcal{A}(\alpha(X))), \mathcal{A}(\alpha(Y))). \tag{3}$$

The final, semi-supervised training objective is simply the weighted the sum of the two loss terms, balanced by a hyperparameter β

$$\mathcal{L}(X, Y, U) = \mathcal{L}_{\text{supervised}}(X, Y) + \beta\mathcal{L}_{\text{PixelDINO}}(U). \tag{4}$$

In our experiments, we find $\beta = 0.1$ to be a good choice for this hyperparameter. We analyze the influence of this hyperparameter in Section V-C.

The pseudocode for this training procedure is outlined in Algorithm 1. By forcing the student model to adhere to the teacher outputs and the labeled ground-truth masks at the same time, it is very likely that the classification schemes will, indeed, align to include one class for our desired target.

C. Data Augmentations

Data augmentation is a commonly used technique to make models more robust to perturbations in the input, as well as encourage equivariance under certain geometric transformations like rotations or reflections [57]. Furthermore, it is a crucial component for semi-supervised learning, which is why we will briefly explain the employed data augmentation techniques.

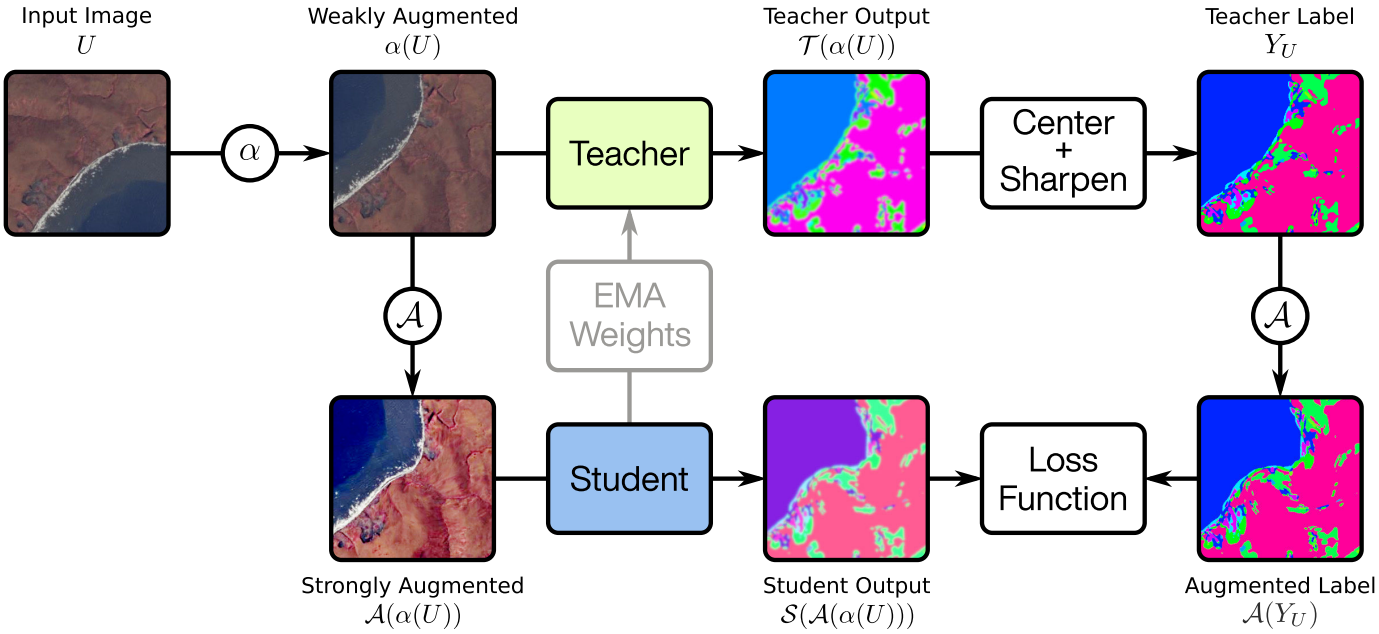

Fig. 3. Overview of the self-supervised part of the PixelDINO framework for pixelwise feature learning. First, the image is weakly augmented and a dense feature map is derived using the teacher model. These labels are turned into class labels by centering, sharpening, and applying the softmax function. Both the weakly augmented image and the teacher label are augmented using the set of strong augmentations. The student model is then trained on this pair of images and label. Finally, the teacher model's weights are updated as an exponential moving average of the student's weights.

Algorithm 1 Semi-Supervised PixelDINO (Pytorch-Style)

Hyper-Parameters:
`beta`: Weight of DINO loss
`temp`: Temperature used for softmax-scaling

```
def train_step(img, mask, unlabelled):
  # Supervised Training Step
  pred = student(img)
  loss_supervised = cross_entropy(pred, mask)

  # Get \text{pseudo-classes} from teacher
  view_1 = augment_weak(unlabelled)
  mask_1 = teacher(mask_1)
  mask_1 = (mask_1 - center) / temp
  batch_center = center.mean(dim=[0,2,3])
  mask_1 = softmax(mask_1)

  # Strongly augment image and label together
  view_2, mask_2 = augment(view_1, mask_1)

  pred_2 = student(view_2)
  loss_dino = cross_entropy(pred_2, mask_2)

  loss = loss_supervised + beta*loss_dino
  loss.backward() # Back-propagate losses
  update(student) # Adam weight update
  ema_update(teacher, student) # Teacher EMA
  ema_update(center, batch_center) # Center EMA
```

The semi-supervised learning methods introduced in this study require two different sets of data augmentation operations, in order to generate different views of the same data. Following the terminology of Sohn et al. [46], we separate the augmentations used in our study into *weak augmentations*, denoted by $\alpha(\cdot)$ and *strong augmentations*, denoted by $\mathcal{A}(\cdot)$. The conceptual difference is that weak augmentations should only add variation to the data without making the classification more difficult. Strong augmentations, on the other hand, distort the image in such a way that makes it harder for the model to perform the classification. During training, every sample is augmented randomly.

1) Weak Augmentations: In the class of weak augmentations, we only include the simple geometric transformations introduced before, namely, horizontal and vertical reflections of the input imagery, as well as rotations by multiples of 90°. These augmentations are very frequently used in remote sensing as models are expected to be equivariant under reflections and rotations for most tasks.

2) Strong Augmentations: Designing a class of strong augmentations for remote sensing imagery is considerably harder than weak augmentations. The commonly used colorspace transformations which are often used for RGB imagery do not generalize well to multispectral imagery. Therefore, we settle for two classes of adjustments. First, we make random adjustments to the image brightness, gamma curve, and contrast. In a second step, we apply rotations by arbitrary angles in the range $[-30°, 30°]$, Gaussian blurring with $\sigma = 2$ pixels, as well as the elastic transform that locally warps parts of the image.

IV. Datasets

As the main data source for this study, we use the fourth iteration of the openly available RTS inventory from Nitze et al. [12] and Nitze [58]. This inventory consists of polygons that were manually labeled using PlanetScope imagery, elevation data, and Landsat timeseries as the source data. Its extent amounts to 4335 polygon annotations of RTS footprints from the years of 2018–2021, with a combined area of ~84 km^2. The focus of the inventory lies on multiple regions in the terrestrial Arctic, mostly in coastal areas.

While Nitze et al. [12] base their analyses on PlanetScope imagery, we opt for Sentinel-2 imagery for this study due to its open availability, which is an important factor in building a large unlabeled dataset for semi-supervised learning. Practically speaking, these two satellite platforms mainly differ in their imaging resolution and their spectral channels. While PlanetScope imagery is provided at ground sampling distances of 3–4 m and contains the visible RGB channels as well as a near-infrared channel, Sentinel-2 imagery comes at a lower spatial resolution of 10 m/pixel per pixel, but in turn features 13 spectral channels.

Using the image footprints from the RTS inventory, we next download 83 matching Sentinel-2 Level 1C images sourced from Google Earth Engine. As the last step, the RTS annotation polygons are rasterized to match the satellite image pixel grids. The annotation masks then contain the binary values 0 and 1 for background and RTS pixels, respectively. Similar to Yang et al. [15], we observe good registration between the footprints and the Sentinel-2 imagery, so that no additional co-registration was performed.

Out of the annotated study regions in the original dataset, we set aside the Herschel Island and Lena sites for testing purposes. We chose the Herschel Island site for being spatially separated from the Canadian mainland. While all other study sites are in the Tundra zone, the Lena site is situated in the Boreal zone. Therefore, it includes land cover features not seen in the other study sites, such as forests. This makes the Lena site a good choice for evaluating spatial generalization, leading us to choose Lena as our second test region. All of the remaining annotated regions are used as the labeled training set.

For the semi-supervised learning methods, we build a secondary unlabeled training dataset by selecting 42 Sentinel-2 tiles over permafrost areas with a focus on regions of continuous permafrost with high estimated ice content. For each one of these tiles, we then randomly select a year from the Sentinel-2 acquisition range and download the least cloudy tile taken between May and August of that year. The time-span from May to August was chosen to match the temporal distribution of the annotated data.

The obtained Sentinel-2 scenes are much larger than even modern GPU cards can handle for neural network training. Furthermore, mini-batch training requires a uniform image size. To fulfill these requirements, all imagery is cut into patches of size 192 × 192 pixels as part of the training pipeline.

After all the preprocessing steps, we arrive at a labeled training dataset with 6464 patches, an unlabeled training dataset with 266 168 patches, and two test datasets, Herschel and Lena, with 1052 and 4420 patches, respectively. Fig. 1 shows the spatial distribution of the labeled and unlabeled training sites and Table I shows the statistical properties of the labeled training sites.

TABLE I
STATISTICS FOR THE STUDY REGIONS (ORDERED BY LONGITUDE)

	RTS		Satellite Images	
Region	Count	Area [km^2]	Count	Area [km^2]
Herschel	148	1.6	10	442.9
Peel Plateau	37	0.68	1	87.9
Tuktoyaktuk	391	1.3	19	899.4
Horton	534	13.2	18	866.0
Banks Island	552	28.2	20	814.6
Kolguev	319	12.6	34	1814.1
Novaya Zemlya	982	12.3	3	454.0
Gydan	50	0.2	2	966.9
West Taimyr	110	0.5	2	1057.1
East Taimyr	839	9.2	3	148.9
Lena	238	4.2	41	2020.6
Lena Delta	136	0.8	1	625.5

V. EXPERIMENTS AND RESULTS

A. Generalization Study

In order to quantify the improvements from the modified training procedure, we conduct experiments with different configurations. Starting with a baseline study without any training improvements, we keep the model architecture fixed and only modify the training process. For good comparability, we also use both the weak and strong data augmentations we defined in Section III-C for this experiment.

Specifically, we train and evaluate models in the following configurations.

1) Baseline: Models trained only using supervised learning, without any data augmentation.

2) Baseline + Aug: Same as baseline, but trained using the weak and strong data augmentation as described in Section III-C.

3) FixMatchSeg: Models trained in the semi-supervised setting using the methodology described by Upretee and Khanal [41].

4) Adversarial: Semi-supervised models trained using the adversarial approach proposed by Hung et al. [50].

5) PixelDINO: Models trained in the semi-supervised setting using our proposed methodology as outlined in Algorithm 1.

As the introduced methodology focuses on adapting the training process itself rather than making changes to the model architecture, it is invariant to the specific model architecture used. Therefore, any semantic segmentation model can be used in practice. For our experiments, we use the UNet model [59] as it is a widely used network architecture for image segmentation tasks in remote sensing.

For each configuration, we train four models with different random seeds to also quantify the effects of the randomness in model initialization, mini-batch sampling, and data augmentation. Models were trained on a GPU server equipped with NVIDIA A6000 GPUs. The implementation was carried out in JAX [60] and Haiku [61]. The code is available online at: https://github.com/khdlr/PixelDINO.

In the semi-supervised setting, the model is being trained on two datasets, the labeled data and the unlabeled data. These two datasets are vastly different in size, with the labeled dataset being much smaller than the unlabeled dataset. Therefore, the concept of "training epochs" is no longer appropriate for specifying the training duration of the model. In order to still keep comparable training schedules for different model configurations, we instead count the number of training steps

TABLE II
RESULTS OF THE GENERALIZATION STUDY: MEAN AND STANDARD DEVIATION OF FOUR RUNS EACH (VALUES IN %)

	Herschel					Lena				
	IoU	mIoU	F1	Precision	Recall	IoU	mIoU	F1	Precision	Recall
Baseline	19.8 ± 1.7	59.6 ± 0.9	33.0 ± 2.3	28.8 ± 3.0	39.4 ± 5.0	28.8 ± 4.0	64.3 ± 2.0	44.6 ± 5.0	52.8 ± 5.9	39.0 ± 6.0
Baseline+Aug	22.9 ± 3.0	61.3 ± 1.5	37.2 ± 3.9	44.2 ± 7.5	32.3 ± 2.0	25.8 ± 10.2	62.8 ± 5.1	40.2 ± 13.0	69.4 ± 3.2	29.4 ± 12.5
FixMatchSeg [41]	23.4 ± 0.8	61.5 ± 0.4	37.9 ± 1.1	34.1 ± 2.3	43.2 ± 4.5	32.4 ± 3.2	66.1 ± 1.6	48.8 ± 3.7	59.4 ± 2.7	41.6 ± 5.0
Adversarial [50]	26.6 ± 3.9	63.2 ± 1.9	41.9 ± 4.9	60.0 ± 9.2	32.3 ± 3.1	25.1 ± 15.1	62.4 ± 7.5	38.2 ± 20.5	87.3 ± 7.5	26.8 ± 16.7
PixelDINO	30.2 ± 2.7	65.0 ± 1.4	46.4 ± 3.2	52.7 ± 9.2	42.0 ± 3.0	39.5 ± 6.5	69.7 ± 3.3	56.4 ± 6.6	77.7 ± 6.3	44.5 ± 6.8

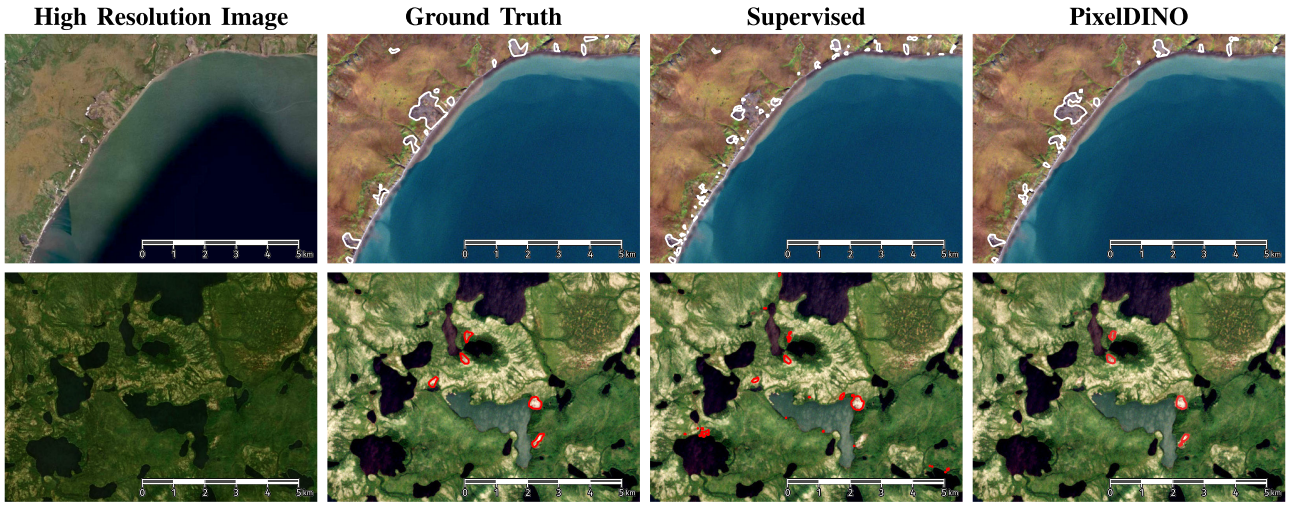

Fig. 4. High-resolution imagery (first column), ground truth (second column), and prediction results for parts of the Herschel Island (top) and Lena (bottom) study sites for the Baseline + Aug (third column) and PixelDINO (fourth column) training methods. Most prominent is the large reduction in false positives due to the semi-supervised training method. The visualizations in columns 2–4 are displayed on top of Sentinel-2 data from the test datasets, high-resolution imagery in column 1 courtesy of Esri, Maxar, Earthstar Geographics, and the GIS User Community.

applied to each model. This should keep the comparison between the models as fair as possible, as each model has gone through the same training schedule. In all reported experiments, the models were trained for 200 000 steps.

B. Evaluation Metrics

The foreground and background classes in this dataset are highly imbalanced. Even though the study areas were chosen to feature regions of high-RTS density, only around 0.7% of all pixels contain a target, while all other pixels belong to the background class. Therefore, pixelwise accuracy is an unfit metric for this task. Instead, we evaluate the models using other metrics which are widely used for such imbalanced segmentation tasks.

1) *Intersection Over Union (IoU):* Fraction of true positive pixels among all pixels that are true targets and/or classified positive.
2) *Mean IoU mIoU:* mIoU for the RTS class and the IoU for the background class.
3) *Precision:* Fraction of true positive pixels among positive classifications.
4) *Recall:* Fraction of true positive pixels among true target pixels.
5) *F1 Score:* The harmonic mean of precision and recall.

The evaluation results of the generalization study are displayed in Table II. Overall, the trend shows better performance of semi-supervised learning methods compared with the supervised baselines. Among the semi-supervised methods, our proposed PixelDINO approach demonstrates the strongest performance, achieving IoU scores of 30.2% for Herschel and 39.5% for Lena. The second best models score 26.6% for Herschel (Adversarial) and 32.4% for Lena (FixMatchSeg).

Although the main focus of this evaluation lies with the relative improvements from semi-supervised learning over supervised learning, we try to give an overview of how our results compare to those obtained by existing studies. Due to differences in data modalities, study regions, spatial sampling, and evaluation metrics, directly comparing this study's results with existing studies is challenging. For the Herschel site, Nitze et al. [12] observe average IoU scores in the range of 20%–25% for the trained models, which is similar to the Baseline + Aug model in this study achieving an IoU of 22.9 ± 3.0. This comparison suggests that the Sentinel-2 and Planet imagery products are comparable for RTS detection. The most comparable training setup by Yang et al. [15] is the model trained on "Extensive Sites" and evaluated on Yamal and Gydan. For this model, the study reports an mIoU of 57%, which is comparable with our baselines, which achieve mIoUs in the range of 60%–65%.

C. Influence of Hyperparameter β

The PixelDINO framework introduces a tunable hyperparameter in 4, namely, the parameter β that determines the

TABLE III
MODEL PERFORMANCE FOR DIFFERENT CHOICES OF β

	Herschel		Lena	
β	IoU	F1	IoU	F1
0.01	28.0 ± 7.3	43.4 ± 9.0	41.7 ± 2.1	58.8 ± 2.1
0.05	24.9 ± 3.6	39.7 ± 4.7	33.3 ± 2.7	49.9 ± 3.0
0.1	30.2 ± 2.7	46.4 ± 3.2	39.5 ± 6.5	56.4 ± 6.6
0.2	30.4 ± 7.7	46.2 ± 9.4	35.1 ± 15.3	50.3 ± 19.2
0.5	36.1 ± 3.8	53.0 ± 4.1	28.7 ± 15.5	42.6 ± 21.2
1.0	31.9 ± 5.3	48.2 ± 6.0	12.9 ± 3.7	22.8 ± 6.0

TABLE IV
RUNTIME OF THE EVALUATED TRAINING METHODS

Method	Training Duration	Change
Baseline	88.9 min	–
Baseline+Aug	91.3 min	+ 2.7%
FixMatchSeg	178.1 min	+ 100.3%
Adversarial	182.4 min	+ 105.2%
PixelDINO	174.9 min	+ 96.8%

weighting of the PixelDINO loss term compared with the supervised loss term. This raises the question of how to choose the hyperparameter β. When β approaches 0, the setup becomes plain supervised learning. For very large values of β, on the other hand, the self-supervised loss term will dominate the supervised learning signal, preventing the model from learning the target classes. Intuitively, there should, therefore, be an optimal choice of β that balances supervised and self-supervised learning in such a way that the model performance is maximized.

We repeat our experiments for different choices of β in the range $[0.01, 1]$, the results of which are shown in Table III. Indeed, we observe that the performance generally decreases toward both edges of this interval. A choice of $\beta = 0.1$ yields good performance on both evaluation datasets. Therefore, we recommend $\beta = 0.1$ as a starting point for tuning this hyperparameter.

D. Effects on Training Duration

One common concern with increasingly complex training schemes is the increase in training time that they incur. In order to assess this, we report the average runtime of our experiments in Table IV. While the impact of data augmentations on the training duration is negligible, all semi-supervised training methods roughly double the duration of training. This is easily explained by the fact that the semi-supervised methods process both a batch of labeled imagery and a batch of unlabeled imagery during each iteration. However, we stress that these duration increases only occur during training and not during inference. During inference, all the presented models will run at the same speed since they share the same model architecture.

VI. DISCUSSION

The results show that for the task of RTS detection, semi-supervised learning can, indeed, yield a strong performance boost. In this section, we will discuss our observations during the experiments, what sets apart PixelDINO from the other semi-supervised learning methods, and implications for follow-up research.

A. Isolating the Effect of Data Augmentations

As consistency across data augmentations makes up a large part of the semi-supervised training methods, the improvements in segmentation accuracy might in fact be explained by the use of data augmentations instead of the semi-supervised training itself. In order to isolate the direct effects of data augmentation on the training process, we trained the baseline supervised model with and without data augmentations.

While the data augmentations improve the model performance on the Herschel evaluation site from an IoU of 19.8%–22.9%, they actually decrease performance for the Lena evaluation site from an IoU of 28.8%–25.5%. This is surprising, as it is generally believed that data augmentation improves the generalizability of machine learning models [57]. We attribute that this to the higher land cover complexity of the Lena site, which features lakes, forests, bright bare ground, and RTSs. Meanwhile, the Herschel site only features tundra, RTSs, and coastal water, matching the training data distribution more closely. Therefore, data augmentation allows the model to better detect coastal thaw slumps, while the generalization performance to inland regions suffers slightly.

At the same time, semi-supervised learning improves the performance of the baseline model much more than just applying data augmentations. From this, we conclude that the improved training performance is not explained by the data augmentations alone, but can instead be attributed to the semi-supervised learning methods.

B. Benefits of Semi-Supervised Learning

The evaluated semi-supervised methods were generally able to improve over the baselines in terms of the IoU and $F1$ metrics, as shown in Table II. Overall, semi-supervised learning has a large positive influence on the performance of the models, with the potential to increase IoU scores by around eight basis points and $F1$ scores by around 12 basis points across both datasets.

The only exception here is the performance of the adversarially trained models on the Lena evaluation site. Here, this class of models actually underperforms the baselines on average. At the same time, the standard deviation is quite high, implying a large spread in model performances for this particular group. This behavior is likely tied to the most common point of criticism for adversarial training, namely, that the training objective dictates a saddle point optimization problem. These are known to be hard to solve and lead to unstable training [62]. In our experiments, this leads to unstable generalization. As the Lena test site differs much more from the training data than the Herschel site, the unstable generalization manifests itself in the Lena dataset but not in Herschel. Meanwhile, FixMatchSeg and PixelDINO do not exhibit this issue.

Generally, our proposed PixelDINO methodology achieves the strongest improvement in the segmentation metrics. This confirms that it is not only competitive with other approaches for semi-supervised semantic segmentation but also, at least for this task, is in fact the preferrable option.

C. Effects of PixelDINO Training

Our hypothesis for the strong performance of PixelDINO models lies in the fact that RTS detection is a task that has only two classes and a strong class imbalance. Therefore, the consistency regularization in approaches based on pseudolabels like FixMatchSeg does not regularize the model sufficiently when it comes to correctly segmenting background features. This hypothesis is supported by visual inspection (see Fig. 4) and the recall and precision metrics in Table II. While FixMatchSeg and PixelDINO have comparable recall values, PixelDINO is far ahead in precision, which suggests that our method is able to greatly reduce the number of false positives while maintaining a constant number of false negatives. Our findings align with Yang et al. [15], who observe that false positives are a large issue in RTS detection and address this by including negative data.

Visual inspection of the results in Fig. 4 supports our hypothesis that PixelDINO training reduces false positives. Furthermore, while the supervised baseline sometimes fragments a single RTS target into multiple polygons, the PixelDINO predictions appear less fragmented, suggesting that our method leads to more robust predictions.

Interestingly, an inverted phenomenon can be observed for the adversarial training method. Here, the precision values are greatly increased, beating even the models trained with PixelDINO. But this comes at the cost of poor Recall values, which means that the adversarially trained model will miss many more RTS targets than the other methods. We believe that this to be related to the adversarial training method. As the discriminator is tasked with discerning true masks from predicted masks, it teaches the segmentation network mainly about the shapes of the features. While it is hard for the model to generate realistic RTS shapes, it is really easy to generate a realistic background tile by not predicting any targets. For ambiguous scenes, the adversarial model might therefore tend to predict only background, as this will always be accepted by the discriminator.

While PixelDINO appears to improve the models' robustness against false positives, we do observe slightly more false negatives in some regions, such as the Lena test set in Fig. 4. Furthermore, as outlined in Section V-D, the semi-supervised models, including PixelDINO, need roughly twice as long to train fully, as they need to ingest both unlabelled and labeled data. While the potential benefits are large, researchers therefore need to carefully consider whether the tradeoffs are justified for a specific task at hand.

Overall, our PixelDINO approach greatly benefits from its ability to further subdivide the background class into regions of different semantic content, which makes the semi-supervised training feedback much more valuable, which in turn leads to more accurate predictions on the test set.

D. Avenues for Follow-Up Research

PixelDINO is easy to implement and can train more accurate RTS detectors without additional labels. We expect that these properties generalize well to other use cases in remote sensing, where data are scarce, large regional variations exist, or classes are highly imbalanced. Examples for such tasks are detecting landslides [63], flood mapping [64], or deforestation mapping [65].

It is hypothesized that satellite imagery of higher resolution will be beneficial for detecting RTSs, as oftentimes the targets can be quite small [12]. While we do not make use of such imagery due to reasons of data availability, the introduced methodology is applicable to any imagery source. It is up to future research to explore the possibilities of such methods for high-resolution satellite or even aerial imagery sources.

While not the focus of this study, a fully self-supervised version of PixelDINO might be able to learn feature maps of high spatial detail. Recent developments in foundation models [66] suggest that this is the way forward for many remote sensing tasks.

VII. Conclusion

Large volumes of remote sensing data are readily available to the public through platforms like the NASA Landsat or ESA Copernicus archives. These open up many possible use cases for monitoring applications. Many use cases for deep learning in remote sensing are, however, hindered by a lack of sufficient labeled training data. This is particularly true for semantic segmentation tasks, because these require all pixels to be labeled. Semi-supervised learning can help relieve the labeling workload on domain experts by a large amount, simply by using readily available unlabeled data.

Our proposed PixelDINO framework achieves this by encouraging the trained model to come up with its own scheme of segmentation classes, for which it is then trained to be consistent across data augmentations as well as to align its classes to the label classes from the annotated training set.

In our experiments, we demonstrated that PixelDINO can train models that generalize well to previously unseen regions in the Arctic and do so better than both supervised baselines and other semi-supervised approaches.

As described in Section VI-C, handling highly imbalanced classes is a strong property of PixelDINO. While our introduced framework is flexible in terms of the number of output channels, further research is needed to understand how well PixelDINO will generalize to semantic segmentation problems with many classes.

We expect the methods developed in this study to be transferrable to many different use cases in remote sensing even outside of permafrost monitoring. Therefore, we hope to inspire follow-up research in improving the automated mapping of ground features using semi-supervised semantic segmentation methods.

Data and Code Availability

The ground-truth data used in this study was published in [12]. It is available online at [58] and https://github.com/initze/ML_training_labels/. The project page containing code and other materials for this study can be found at: https://khdlr.github.io/PixelDINO/.

REFERENCES

[1] E. Buch et al., "Arctic in situ data avilability," Eur. Environ. Agency, Kobenhavn, Denmark, Tech. Rep. 2.1, 2019.

[2] C. Gabarró et al., "Improving satellite-based monitoring of the polar regions: Identification of research and capacity gaps," *Frontiers Remote Sens.*, vol. 4, Feb. 2023, Art. no. 952091.

[3] A. Bartsch, T. Strozzi, and I. Nitze, "Permafrost monitoring from space," *Surv. Geophys.*, vol. 44, no. 5, pp. 1579–1613, Mar. 2023.

[4] N. Nesterova et al., "Review article: Retrogressive thaw slump theory and terminology," *EGUsphere*, vol. 2024, pp. 1–36, Jan. 2024.

[5] C. R. Burn, "The thermal regime of a retrogressive thaw slump near Mayo, Yukon Territory," *Can. J. Earth Sci.*, vol. 37, no. 7, pp. 967–981, Jul. 2000.

[6] H. Lantuit and W. H. Pollard, "Temporal stereophotogrammetric analysis of retrogressive thaw slumps on Herschel Island, Yukon Territory," *Natural Hazards Earth Syst. Sci.*, vol. 5, no. 3, pp. 413–423, May 2005.

[7] A. I. Kizyakov et al., "Landforms and degradation pattern of the Batagay thaw slump, northeastern Siberia," *Geomorphology*, vol. 420, Jan. 2023, Art. no. 108501.

[8] I. Nitze, G. Grosse, B. M. Jones, V. E. Romanovsky, and J. Boike, "Remote sensing quantifies widespread abundance of permafrost region disturbances across the Arctic and subarctic," *Nature Commun.*, vol. 9, no. 1, pp. 1–11, Dec. 2018.

[9] J. Hjort, D. Streletskiy, G. Doré, Q. Wu, K. Bjella, and M. Luoto, "Impacts of permafrost degradation on infrastructure," *Nature Rev. Earth Environ.*, vol. 3, no. 1, pp. 24–38, Jan. 2022.

[10] S. V. Kokelj, R. E. Jenkins, D. Milburn, C. R. Burn, and N. Snow, "The influence of thermokarst disturbance on the water quality of small upland lakes, Mackenzie delta region, northwest territories, Canada," *Permafrost Periglacial Processes*, vol. 16, no. 4, pp. 343–353, 2005.

[11] L. Huang, L. Liu, L. Jiang, and T. Zhang, "Automatic mapping of thermokarst landforms from remote sensing images using deep learning: A case study in the northeastern Tibetan Plateau," *Remote Sens.*, vol. 10, no. 12, p. 2067, Dec. 2018.

[12] I. Nitze, K. Heidler, S. Barth, and G. Grosse, "Developing and testing a deep learning approach for mapping retrogressive thaw slumps," *Remote Sens.*, vol. 13, no. 21, p. 4294, Oct. 2021.

[13] L. Huang, T. C. Lantz, R. H. Fraser, K. F. Tiampo, M. J. Willis, and K. Schaefer, "Accuracy, efficiency, and transferability of a deep learning model for mapping retrogressive thaw slumps across the Canadian Arctic," *Remote Sens.*, vol. 14, no. 12, p. 2747, Jun. 2022.

[14] C. Witharana et al., "Automated detection of retrogressive thaw slumps in the high Arctic using high-resolution satellite imagery," *Remote Sens.*, vol. 14, no. 17, p. 4132, Aug. 2022.

[15] Y. Yang et al., "Mapping retrogressive thaw slumps using deep neural networks," *Remote Sens. Environ.*, vol. 288, Apr. 2023, Art. no. 113495.

[16] J. Brown, O. Ferrians, J. A. Heginbottom, and E. Melnikov, "Circum-arctic map of permafrost and ground-ice conditions, version 2," Nat. Snow Ice Data Center, Boulder, CO, USA, Tech. Rep. GGD318, 2002.

[17] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, Feb. 2020.

[18] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-supervised representation learning: Introduction, advances, and challenges," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 42–62, May 2022.

[19] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9630–9640.

[20] A. Brooker, R. H. Fraser, I. Olthof, S. V. Kokelj, and D. Lacelle, "Mapping the activity and evolution of retrogressive thaw slumps by tasselled cap trend analysis of a landsat satellite image stack," *Permafrost Periglacial Processes*, vol. 25, no. 4, pp. 243–256, Oct. 2014.

[21] L. Huang et al., "Identifying active retrogressive thaw slumps from ArcticDEM," *ISPRS J. Photogramm. Remote Sens.*, vol. 205, pp. 301–316, Nov. 2023.

[22] P. Bernhard, S. Zwieback, S. Leinss, and I. Hajnsek, "Mapping retrogressive thaw slumps using single-pass TanDEM-X observations," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3263–3280, 2020.

[23] R. A. Segal, T. C. Lantz, and S. V. Kokelj, "Acceleration of thaw slump activity in glaciated landscapes of the western Canadian Arctic," *Environ. Res. Lett.*, vol. 11, no. 3, Mar. 2016, Art. no. 034025.

[24] M. Leibman, N. Nesterova, and M. Altukhov, "Distribution and morphometry of thermocirques in the north of West Siberia, Russia," *Geosciences*, vol. 13, no. 6, p. 167, Jun. 2023.

[25] T. Rettelbach et al., "A quantitative graph-based approach to monitoring ice-wedge trough dynamics in polygonal permafrost landscapes," *Remote Sens.*, vol. 13, no. 16, p. 3098, Aug. 2021.

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[27] L. Huang, J. Luo, Z. Lin, F. Niu, and L. Liu, "Using deep learning to map retrogressive thaw slumps in the Beiluhe region (Tibetan Plateau) from CubeSat images," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111534.

[28] L. Hughes-Allen, F. Bouchard, A. Séjourné, G. Fougeron, and E. Léger, "Automated identification of thermokarst lakes using machine learning in the ice-rich permafrost landscape of Central Yakutia (Eastern Siberia)," *Remote Sens.*, vol. 15, no. 5, p. 1226, Feb. 2023.

[29] C. M. Gibson, L. E. Chasmer, D. K. Thompson, W. L. Quinton, M. D. Flannigan, and D. Olefeldt, "Wildfire as a major driver of recent permafrost thaw in boreal peatlands," *Nature Commun.*, vol. 9, no. 1, p. 3041, Aug. 2018.

[30] C. J. Abolt, M. H. Young, A. L. Atchley, and C. J. Wilson, "Brief communication: Rapid machine-learning-based extraction and measurement of ice wedge polygons in high-resolution digital elevation models," *Cryosphere*, vol. 13, no. 1, pp. 237–245, Jan. 2019.

[31] C. Witharana et al., "An object-based approach for mapping Tundra ice-wedge polygon troughs from very high spatial resolution optical satellite imagery," *Remote Sens.*, vol. 13, no. 4, p. 558, Feb. 2021.

[32] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33. Red Hook, NY, USA: Curran Associates, 2020, pp. 9912–9924.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.

[34] J.-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[35] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 9726–9735.

[36] L. Kondmann et al., "DENETHOR: The dynamicearthNET dataset for harmonized, inter-operable, analysis-ready, daily crop monitoring from space," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2021, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=uUa4jNMLjrL

[37] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2017.

[38] E. Bowler, P. T. Fretwell, G. French, and M. Mackiewicz, "Using deep learning to count albatrosses from space: Assessing results in light of ground truth uncertainty," *Remote Sens.*, vol. 12, no. 12, p. 2026, Jun. 2020.

[39] I. Nitze et al., "A labeling intercomparison of retrogressive thaw slumps by a diverse group of domain experts," *EarthArXiV Preprint*, pp. 1–24, Apr. 2024.

[40] Y. Hua, D. Marcos, L. Mou, X. X. Zhu, and D. Tuia, "Semantic segmentation of remote sensing images with sparse annotations," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[41] P. Upretee and B. Khanal, "FixMatchSeg: Fixing FixMatch for semi-supervised semantic segmentation," 2022, *arXiv:2208.00400*.

[42] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, vol. 3, no. 2, p. 896.

[43] Q. Li, Y. Shi, and X. X. Zhu, "Semi-supervised building footprint generation with feature and output consistency training," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5623217.

[44] L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi, "Revisiting weak-to-strong consistency in semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7236–7246.

[45] V. Verma et al., "Interpolation consistency training for semi-supervised learning," *Neural Netw.*, vol. 145, pp. 90–106, Jan. 2022.

[46] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 596–608.

[47] B. Zhang et al., "Semi-supervised deep learning via transformation consistency regularization for remote sensing image semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1–15, 2022.

[48] F. Zhang, Y. Shi, Q. Xu, Z. Xiong, W. Yao, and X. X. Zhu, "On the generalization of the semantic segmentation model for landslide detection," in *Proc. CDCEO@ IJCAI*, 2022, pp. 96–100.
[49] J. Wang, C. H. Q. Ding, S. Chen, C. He, and B. Luo, "Semi-supervised remote sensing image semantic segmentation via consistency regularization and average update of pseudo-label," *Remote Sens.*, vol. 12, no. 21, p. 3603, Nov. 2020.
[50] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, "Adversarial learning for semi-supervised semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–12.
[51] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5688–5696.
[52] Z. He, H. Liu, Y. Wang, and J. Hu, "Generative adversarial networks-based semi-supervised learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 10, p. 1042, Oct. 2017.
[53] J. Liu et al., "Semi-supervised change detection based on graphs with generative adversarial networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 74–77.
[54] N. A. A. Braham, L. Mou, J. Chanussot, J. Mairal, and X. X. Zhu, "Self supervised learning for few shot hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2022, pp. 267–270.
[55] K. Heidler et al., "Self-supervised audiovisual representation learning for remote sensing data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 116, Feb. 2023, Art. no. 103130.
[56] O. Manas, A. Lacoste, X. Giro-i-Nieto, D. Vazquez, and P. Rodriguez, "Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9414–9423.
[57] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017.
[58] I. Nitze, Jul. 2024, "Initze/ml_training_labels: v1.0," doi: 10.5281/zenodo.12706221.
[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[60] J. Bradbury et al. (2018). *JAX: Composable Transformations of Python+NumPy Programs, Version 0.4.25*. [Online]. Available: http://github.com/google/jax
[61] T. Hennigan, T. Cai, T. Norman, L. Martens, and I. Babuschkin. (2020). *Haiku: Sonnet for JAX, Version 0.0.12*. [Online]. Available: http://github.com/deepmind/dm-haiku
[62] D. Saxena and J. Cao, "Generative adversarial networks (GANs): Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 63:1–63:42, May 2021.
[63] P. Li, Y. Wang, G. Xu, and L. Wang, "LandslideCL: Towards robust landslide analysis guided by contrastive learning," *Landslides*, vol. 20, no. 2, pp. 461–474, Feb. 2023.
[64] A. Shastry, E. Carter, B. Coltin, R. Sleeter, S. Mcmichael, and J. Eggleston, "Mapping floods from remote sensing data and quantifying the effects of surface obstruction by clouds and vegetation," *Remote Sens. Environ.*, vol. 291, Jun. 2023, Art. no. 113556.
[65] A. Jamali, S. K. Roy, J. Li, and P. Ghamisi, "TransU-Net++: Rethinking attention gated TransU-Net for deforestation mapping," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 120, Jun. 2023, Art. no. 103332.
[66] X. Xiang Zhu et al., "On the foundations of earth and climate foundation models," 2024, *arXiv:2405.04285*.

Konrad Heidler (Student Member, IEEE) received the bachelor's degree (B.Sc.) in mathematics, the master's degree (M.Sc.) in mathematics in data science, and the Doctorate in Engineering degree (Dr.-Ing.) from the Technical University of Munich (TUM), Munich, Germany, in 2017, 2020, and 2024, respectively.

He is currently a Post-Doctoral Researcher with TUM, where he is leading the working group for visual learning and reasoning at the Chair for Data Science in Earth Observation. His research work focuses on the application of deep learning for remote sensing in polar regions, solving reasoning tasks with deep learning, and applications of self- and semi-supervised learning in Earth observation.

Ingmar Nitze received the bachelor's degree (B.Sc.) in geography from the Free University (FU) of Berlin, Berlin, Germany, in 2009, and the master's degree (M.Sc.) in geoinformation and visualization and the Doctorate degree in remote sensing (Ph.D./Dr.) from the University of Potsdam, Potsdam, Germany, in 2012 and 2018, respectively.

From 2012 to 2014, he worked as a Research Assistant at the University College Cork, Cork, Ireland. He is currently a Researcher with the Permafrost Research Section, Alfred Wegener Institute (AWI) Helmholtz Centre for Polar and Marine Research, Potsdam. His research focuses on the detection and quantification of landscape dynamics in the circum-Arctic permafrost region using remote sensing and machine learning.

Guido Grosse received the Diploma (M.Sc.) degree in geology from the Technical University and Mining Academy Freiberg (TUBA), Freiberg, Germany, in 2021, and the Dr. rer. nat. (Ph.D.) degree in geosciences from the University of Potsdam, Potsdam, Germany, in 2005.

He became a Post-Doctoral Researcher and then a Research Assistant Professor at the Geophysical Institute, University of Alaska Fairbanks, Fairbanks, AK, USA, in 2006 and 2009, respectively. He returned to Germany at the Alfred Wegener Institute (AWI) Helmholtz Centre for Polar and Marine Research, Potsdam and became a Full Professor on Permafrost in the Earth System jointly appointed by AWI and the University of Potsdam. Since 2016, he has been the Head of the Permafrost Research Section, AWI. His team increasingly develops and applies computer vision, machine learning, and deep learning methods in remote sensing of Arctic permafrost. He has authored more than 195 peer-reviewed publications, participated in more than 35 arctic expeditions and is involved in multiple international permafrost-related networks and research projects. His research focuses on remote sensing of landscape dynamics across broad spatial and temporal scales, hydrology, carbon cycling, and the impacts of climate change in Arctic permafrost regions.

Dr. Grosse won an ERC Starting Grant in 2013.

Xiao Xiang Zhu (Fellow, IEEE) received the master's (M.Sc.), the Doctor of Engineering (Dr.-Ing.), and "Habilitation" degrees in the field of signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She was the Founding Head of the Department "EO Data Science," Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany. Since May 2020, she been the PI and Director of the International Future AI Lab "AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond." Since October 2020, she has been the Director of the Munich Data Science Institute (MDSI), TUM. From 2019 to 2022, she has been a Co-Coordinator of the Munich Data Science Research School and the Head of the Helmholtz Artificial Intelligence—Research Field "Aeronautics, Space and Transport." She was a Guest Scientist or a Visiting Professor at the Italian National Research Council (CNR-IREA), Naples, Italy; Fudan University, Shanghai, China; The University of Tokyo, Tokyo, Japan; and University of California at Los Angeles, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. She is currently the Chair Professor for Data Science in Earth Observation at TUM. She is also a Visiting AI Professor at ESA's Phi-Lab, Frascati, Italy. Her main research interests are remote sensing and Earth observation, signal processing, machine learning and data science, with their applications in tackling societal grand challenges, e.g., global urbanization, united nations (UNs) societal development goals (SDGs), and climate change.

Dr. Zhu is a fellow of the Academia Europaea (Academy of Europe), Asia-Pacific Artificial Intelligence Association (AAIA), and European Laboratory for Learning and Intelligent Systems (ELLIS). She has been a member of the Young Academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities and the German National Academy of Sciences Leopoldina and the Bavarian Academy of Sciences and Humanities. She serves on the Scientific Advisory Board in several research organizations, among others the German Research Center for Geosciences (GFZ) from 2020 to 2023 and the Potsdam Institute for Climate Impact Research (PIK). She is an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and *Pattern Recognition* and served as the Area Editor responsible for special issues of *IEEE Signal Processing Magazine* from 2021 to 2023.