TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation

Joachim Nyborg, Charlotte Pelletier, Sébastien Lefèvre, Ira Assent

TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation

Joachim Nyborg[a,c], Charlotte Pelletier[b], Sébastien Lefèvre[b], Ira Assent[a]

[a]*Department of Computer Science, Aarhus University, Aarhus, Denmark*
[b]*IRISA UMR 6074, Université Bretagne Sud, Vannes, France*
[c]*FieldSense A/S, Aarhus, Denmark*

Abstract

The recent developments of deep learning models that capture the complex temporal patterns of crop phenology have greatly advanced crop classification of Satellite Image Time Series (SITS). However, when applied to target regions spatially different from the training region, these models perform poorly without any target labels due to the temporal shift of crop phenology between regions. To address this unsupervised cross-region adaptation setting, existing methods learn domain-invariant features without any target supervision, but not the temporal shift itself. As a consequence, these techniques provide only limited benefits for SITS. In this paper, we propose TimeMatch, a new unsupervised domain adaptation method for SITS that directly accounts for the temporal shift. TimeMatch consists of two components: 1) temporal shift estimation, which estimates the temporal shift of the unlabeled target region with a source-trained model, and 2) TimeMatch learning, which combines temporal shift estimation with semi-supervised learning to adapt a classifier to an unlabeled target region. We also introduce an open-access dataset for cross-region adaptation with SITS from four different regions in Europe. On this dataset, we demonstrate that TimeMatch outperforms all competing methods by 11% in F1-score across five different adaptation scenarios, setting a new state-of-the-art for cross-region adaptation.

Keywords: Satellite Image Time Series, Temporal Shift, Crop Classification, Domain Adaptation, Deep Learning

1. Introduction

Today, the availability of satellite image time series (SITS) data is rapidly increasing. For instance, the twin Sentinel-2 satellites provide imagery of the entire Earth every two to five days [1]. A frequent acquisition of images is crucial for vegetation-related remote sensing applications such as crop type classification [2, 3]. Multi-temporal data enables capturing the phenological development of crops (*i.e.*, the progressions of crop growth), a key dimension to discriminate each crop type [4]. Recently, the increasing availability of SITS along with advances in deep learning has led to crop classifiers with temporal neural architectures using convolutions [5, 6], recurrent units [7–10], self-attention [11, 12], or combinations thereof [13, 14].

These crop classification models achieve impressive performance by capturing the temporal structure of the problem but rely on the existence of a large amount of labeled training data. While unlabeled SITS are plenty, access to labels in the region of interest (the *target* domain) is often either costly or otherwise unavailable. A possible solution is to train a model in a region with labels available (the *source* domain) and apply it to the unlabeled target region. However, when the two regions are geographically different, the dissimilarity between the source and target data distributions can cause a source-trained model to perform poorly when applied to the target region [15–17].

Solving the distributional shift problem to adapt a source-trained model to an unlabeled target domain is in machine learning known as unsupervised domain adaptation (UDA) [15, 18, 19]. Here, we consider the cross-region UDA problem for SITS [20], where we are provided with labeled data from a source region and unlabeled data from a target region. In this setting, the source and target data distributions differ due to changes in local conditions, such as the soil, climate, and farmer practices, which cause spectral and temporal shifts [15].

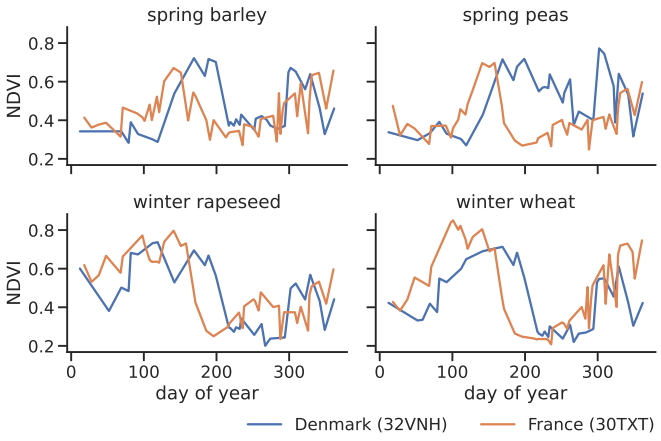


Figure 1: Normalized difference vegetation index (NDVI) time series for crops from two different Sentinel-2 tiles in Europe, indicating the growth of four crop types. Crops develop similarly in different regions, but the patterns are temporally shifted, *e.g.* if crops ripen at different times of the year.

Addressing the temporal shift is of particular importance when adapting crop classifiers to new regions, as we illustrate in Figure 1. While crops in different regions have similar growth patterns, the timing of key growth stages, such as the peak of greenness, is shifted along the temporal axis. As crops are classified primarily by their unique growth patterns, the temporal shift may cause inaccuracies when a source-trained model is applied to a target region. For example, the shift in time could cause the phenology of spring barley to appear similar to that of winter barley in the target. Hence, a key factor in reducing the domain discrepancy for cross-region adaptation is to account for the temporal shift.

Existing deep learning-based UDA methods typically tackle domain adaptation by constraining the classifier to operate on domain-invariant features [21]. This is achieved by training the classifier to perform well on the source domain while minimizing a divergence measure between features extracted from the source and the target domains [20, 22, 23]. While these methods have been successfully applied in various applications [19, 24], they do not directly account for the temporal shift in SITS and have thus been reported to provide limited benefits in cross-region UDA [25].

In this paper, we propose *TimeMatch*, where we directly account for the temporal shift of SITS to address the cross-region UDA problem. TimeMatch consists of two components: (i) the temporal shift estimation and (ii) the TimeMatch learning algorithm. As the target region is unlabeled, it is difficult to estimate the temporal shift directly from the data by comparing *e.g.* the vegetation indices for the individual crop types. Instead, we propose an unsupervised method for temporal shift estimation, where we determine the shift by the confidence and class distribution of predictions from a source-trained model applied to temporally shifted target data. Then, by estimating the temporal shift and applying it to the data, we reduce the domain discrepancy between the source and target regions. This changes the problem setting from UDA to semi-supervised learning (SSL) since the labeled and unlabeled data now come from similar distributions [26]. Thus, in TimeMatch learning, we use SSL to train with the unlabeled target domain. We generate accurate pseudo-labels [27, 28] using a source-trained model on target samples with a reduced temporal shift. Then, we adapt the crop classifier to the target domain using the pseudo-labeled target data along with the available labels for temporally shifted source data, resulting in an accurate crop classifier for the target region.

Lastly, we present the TimeMatch dataset, a challenging new open-access dataset for training and evaluating cross-region models on SITS with over 300.000 annotated parcels from four different regions in Europe. Evaluated on this dataset, our approach outperforms all competing methods by 11% in F1-score on average across five different cross-region UDA experiments.

In summary, our contributions are as follows:

- We propose a method for estimating the temporal shift between a labeled source region and an unlabeled target region to reduce their temporal discrepancy.
- We propose *TimeMatch*, a novel UDA method designed for the cross-region problem of SITS, where crop classification models are adapted to an unlabeled target region by semi-supervised learning on temporally shifted data for improved performance compared to existing UDA methods.
- We release the TimeMatch dataset [29], a new dataset for training and evaluating cross-region UDA models on SITS from four different European regions.

This paper is organized as follows. Section 2 describes the existing literature related to our work. Section 3 describes the proposed method for temporal shift estimation and the TimeMatch learning algorithm. Section 4 presents our dataset and the experimental setup, and Section 5 the experimental results. Lastly, Section 6 concludes this work.

2. Related Work

TimeMatch is related to existing work in unsupervised domain adaptation, time-series domain adaptation, cross-region crop classification, and semi-supervised learning.

2.1. Unsupervised Domain Adaptation

Unsupervised domain adaptation aims to transfer a model from a labeled source domain to an unlabeled target domain by reducing the domain discrepancy [21, 24]. In Ben-David *et al.* [21], it is shown that a classifier's target domain accuracy is bounded by the accuracy on the source and the domain discrepancy. Hence, most recent domain adaptation methods focus on reducing the domain discrepancy by learning domain-invariant deep features [22, 23, 30].

A popular family of approaches is based on adversarial methods [22, 30, 31]. In domain adversarial neural networks (DANN) [22, 31], the feature extractor is adversarially trained to produce domain-invariant features that are indistinguishable by a domain discriminator. Conditional domain adversarial networks (CDAN) [30] improves upon DANN by conditioning the domain discriminator on classifier predictions in addition to features to enable the alignment of multimodal data distributions.

Another approach is to align the feature distributions directly by minimizing a divergence measure. Choices for divergence measure include maximum mean discrepancy (MMD) [23], correlation alignment [32], or optimal transport [33, 34]. Recently, JUMBOT [34] achieves state-of-the-art UDA results by using mini-batch unbalanced optimal transport to minimize the domain discrepancy of joint deep feature and label distributions.

While the aforementioned methods achieve high performance on computer vision datasets, they do not handle the temporal dimension of SITS data.

2.2. Time Series Domain Adaptation

Only a few UDA methods tackle the challenge of time series domain adaptation. Current methods for time series domain adaptation typically follow the approach in non-temporal UDA and learn domain-invariant features but instead use temporal network architectures [35, 36].

Recurrent domain adversarial neural network (R-DANN) and variational recurrent adversarial deep domain adaptation (VRADA) explore long short-term memory and variational recurrent neural networks as feature extractors, respectively, and learn domain-invariant features with the DANN domain discriminator [35]. Likewise, the convolutional deep domain adaptation model for time series data (CoDATS) learns domain-invariant features with a temporal convolutional network in combination with the DANN domain discriminator [36]. However, while these methods are effective at learning domain-invariant features, they are not designed to learn the temporal shift of SITS.

2.3. Cross-Region Crop Classification

Lucas *et al*. [25] reports that existing UDA methods, including existing domain-invariant methods [37, 38], perform poorly when applied to cross-region UDA of SITS due to the temporal shift problem and the change in class distribution between the two regions.

Recently, Wang *et al*. [20] proposed the phenology alignment network (PAN) as the first method for cross-region UDA of SITS. PAN learns domain-invariant features with MMD [23] and a feature extractor consisting of gated recurrent units and self-attention. Still, by learning domain-invariant features, PAN does not directly address the temporal shift problem. Different from the aforementioned methods, TimeMatch directly accounts for the temporal shift of SITS.

2.4. Semi-Supervised Learning

UDA and SSL are closely related. When the source and target data distributions are aligned, the UDA problem becomes an SSL problem [26]. A popular class of SSL methods can be viewed as producing an artificial label for unlabeled data. For example, pseudo-labeling [27] uses a model's own prediction as a label to train against [39, 40]. Similarly, consistency regularization [41, 42] obtains an artificial label using the model's predicted distribution after randomly augmenting the input or model function. In Mean Teacher [40], the model assumes a dual role as *teacher* and *student*. The student is updated by gradient descent with pseudo-labels generated by the teacher, whereas the teacher is updated by an exponential moving average (EMA) of student parameters to improve the quality of the teacher-generated pseudo-labels. The FixMatch algorithm [28] combines pseudo-labeling and consistency regularization. FixMatch generates pseudo-labels using the model's prediction on weakly-augmented images. If the model prediction is confident, the pseudo-label is then used to update the model on a strongly-augmented version of the same image. FixMatch takes advantage of both pseudo-labeling and consistency regularization between differently augmented images to achieve strong SSL performance.

By estimating the temporal shift between the source and target regions, we reduce their domain discrepancy which enables SSL as a method to learn from the unlabeled target data. We thus use SSL in TimeMatch learning to adapt a model to the target region by combining temporal shift estimation with FixMatch [28] and EMA training [40].

3. TimeMatch

In this section, we describe our proposed method TimeMatch for cross-region UDA. We begin by formally defining the problem setting, followed by an overview of how TimeMatch addresses it. We then give the details of the two TimeMatch components: temporal shift estimation and TimeMatch learning.

3.1. Problem Setting

In crop classification, the input is a sequence of satellite images $\boldsymbol{x}_i = (\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(T_i)})$ of length T_i to be classified into one of the K crop classes. In object-based classification, which we focus on in this work, each $\boldsymbol{x}_i \in \mathbb{R}^{T_i \times N_i \times C}$ contains a sequence of N_i pixels of C spectral bands within a field parcel. Each $\boldsymbol{x}_i$ is accompanied by a sequence $\boldsymbol{\tau}_i = (\tau_i^{(1)}, \ldots, \tau_i^{(T_i)})$ indicating the time $\tau_i^{(j)}$ at which each observation $\boldsymbol{x}_i^{(j)}$ is sampled. In practice, $\tau_i^{(j)}$ is typically represented by the days passed since the first observation [12]. This extra input makes it possible for models to account for the irregular temporal sampling of most satellites. The goal of the crop classification task is to learn a model which predicts class probabilities $p(y|(\boldsymbol{x}_i, \boldsymbol{\tau}_i)) \in \mathbb{R}^K$, typically learned with supervision from labels $y \in \{1, \ldots, K\}$.

In this work, we consider the problem of cross-region UDA. We are given a source domain $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s, y_i^s)\}_{i=1}^{n^s}$ of n^s labeled SITS and a target domain $\mathcal{D}^t = \{\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t\}_{i=1}^{n^t}$ of n^t unlabeled SITS. When the source and target domains consist of SITS from different geographical areas, the domains can be associated with two different joint distributions. The distribution shift is a result of changes in local conditions, *e.g*. soil, weather, climate, or farmer practices, causing temporal discrepancies [15]. The resulting distribution shift causes models trained with the labeled source domain to fail when applied to the unlabeled target domain [16], which hinders the large-scale application of crop classification to regions without available labels.

We aim to address cross-region UDA by adapting a classifier trained on $\mathcal{D}^s$ to make predictions on $\mathcal{D}^t$. We note that the classes in the source may not be exactly the same as the classes in the target. This complicates UDA, which typically assumes a closed-set setting [43], where the set of classes in the source and target domains are equal. For simplicity, we focus on a closed-set setting by adapting a classifier trained for the main $K-1$ crop types

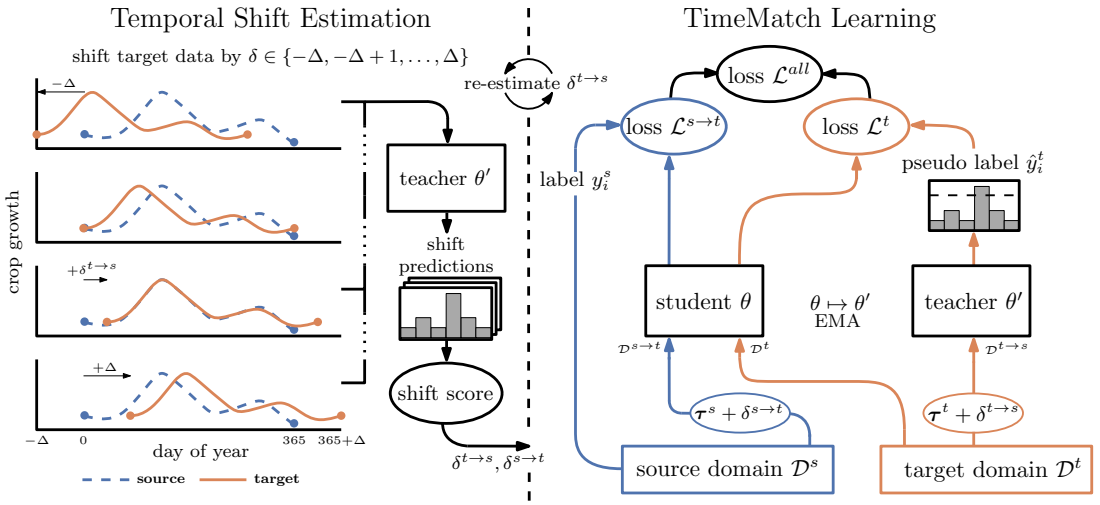


Figure 2: Overview of TimeMatch. Both the student and teacher are pre-trained on the source domain. *Temporal Shift Estimation*: We input shifted target data to the teacher model and obtain its predictions for each shift. We then score each shift by the confidence and diversity of the teacher predictions, and the shift with the best score is output as the temporal shift estimate $\delta^{t\to s}$ and $\delta^{s\to t} = -\delta^{t\to s}$. *TimeMatch Learning*: The teacher generates pseudo-labels for unlabeled target data shifted by $\delta^{t\to s}$. Then, the student is updated for (non-shifted) target data using the pseudo-labels, and for source data shifted by $\delta^{s\to t}$ using the available source labels. As a result, the student is adapted to the target domain with both generated target labels and actual source labels. After the student parameters have been updated with gradient descent, the teacher parameters are updated as an exponential moving average (EMA) of the student parameters. As both models adapt to the temporal shift of the target domain, the best shift for pseudo-labeling with the teacher changes and must be re-estimated. The EMA ensures the teacher adapts slowly which enables $\delta^{t\to s}$ to be re-estimated each epoch only for improved training efficiency and pseudo-label accuracy.

in the source region, plus an "unknown" class containing all remaining source data. This ensures that all target examples can be classified to either one of the $K-1$ crop classes or "unknown".

3.2. Approach Overview

Here we give an overview of how TimeMatch addresses the cross-region UDA problem before describing the full details. A visual presentation of TimeMatch is given in Figure 2. TimeMatch consists of two components (i) temporal shift estimation and (ii) TimeMatch learning.

We aim to estimate the temporal shift between the source and target regions to reduce their domain discrepancy (see Section 1). We represent the temporal shift by a scalar $\delta^{t\to s} \in \mathbb{Z}$ (as the number of days), here in the direction from target to source. Note that the shift in the opposite direction is obtained by $\delta^{s\to t} = -\delta^{t\to s}$, and we thus only have to estimate one. To shift the target domain by $\delta^{t\to s}$, we write $\boldsymbol{\tau}^t + \delta^{t\to s}$, meaning $\delta^{t\to s}$ is added element-wise to each target day-of-year. With our proposed method for temporal shift estimation (Section 3.3), we obtain estimates for $\delta^{t\to s}$ and $\delta^{s\to t}$.

In TimeMatch learning (Section 3.4), we use $\delta^{s\to t}$ to construct a target-shifted source domain $\mathcal{D}^{s\to t} = \{(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s + \delta^{s\to t}), y_i^s\}_{i=1}^{n^s}$, which is distributed similarly as the unlabeled target domain $\mathcal{D}^t$. Learning from the labeled $\mathcal{D}^{s\to t}$ and unlabeled $\mathcal{D}^t$ can thus be achieved with SSL. To do so, TimeMatch learning unifies temporal shift estimation with the loss function of FixMatch [28] and the exponential moving average (EMA) training of Mean Teacher [40], as we explain next.

We first obtain source-trained parameters by training a crop classifier with $\mathcal{D}^s$. We then duplicate the trained classifier into two models: the *teacher* and the *student*. Our TimeMatch learning algorithm aims to adapt both the teacher and the student to the new target region. The teacher generates pseudo-labels for the target domain to train the student, and the knowledge learned by the student is then updated back to the teacher, thus the pseudo-labels used to train the student itself are improved. We generate pseudo-labels by using $\delta^{t\to s}$ to create an adapted target domain $\mathcal{D}^{t\to s} = \{\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t + \delta^{t\to s}\}_{i=1}^{n^t}$. As $\mathcal{D}^{t\to s}$ is temporally aligned with $\mathcal{D}^s$, the source-initialized teacher generates more accurate pseudo-labels for $\mathcal{D}^{t\to s}$ than $\mathcal{D}^t$. The student is then trained with labeled $\mathcal{D}^{s\to t}$ and pseudo-labeled $\mathcal{D}^t$ via the FixMatch loss [28], thereby leveraging both the available source labels and the target pseudo-labels to adapt the student to the target domain.

After updating the student, the teacher is updated via an EMA of the student parameters. As the two models adjust to the temporal shift of the target domain, the best shift $\delta^{t\to s}$ for pseudo-labeling with the teacher gradually moves to zero during TimeMatch learning. To adjust to the changing shift and ensure the pseudo-labels are consistently accurate, it is necessary to re-estimate the temporal shift of the teacher as it learns. However, repeating temporal shift estimation is computationally expensive, and drastically increases training time if done each training iteration. Therefore, in Section 3.4.3, we discuss how EMA training alleviates this issue by enabling the re-estimation to be done only once per epoch.

Next, we first describe our method for estimating the

temporal shift before describing the loss function and learning algorithm of TimeMatch learning.

3.3. Temporal Shift Estimation

Estimating the temporal shift directly from the data is difficult, as labels are not available in the target domain. Without labels, we cannot separate the target data into each crop type, which prevents the computation of *e.g.* vegetation indices to compare the source and target phenology of each crop type directly.

Instead, we propose to estimate the temporal shift by calculating statistics on the predictions of a source-trained model when input temporally shifted target data. By doing so, we estimate the shift that aligns the target data with the source crop phenology learned by a model, leveraging the classification ability of the trained model to estimate the shift from unlabeled data. Another benefit of this approach is that it enables re-estimation of the best temporal shift for pseudo-labeling as the learned phenology of the model changes from source to target in TimeMatch learning.

One possible value to measure is the confidence of the model predictions. Intuitively, when a source-trained model is applied to correctly shifted target data, it should output more confident predictions than for incorrectly shifted target data. As correctly classified examples tend to have more confident predictions than wrongly classified or out-of-distribution examples [44], we argue that a confident temporal shift indicates a better alignment of the target domain with the source which results in accurate pseudo-labels and reduced domain discrepancy.

We can measure the confidence of a model for a shift $\delta^{t\to s}$ by the expected entropy:

$$\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)\sim\mathcal{D}^t}\left[H\big(p_\theta\big(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\big)\big)\right], \tag{1}$$

where H denotes the entropy, here computed over the predictions of the model θ when input temporally shifted target data sampled from $\mathcal{D}^t$. To estimate the temporal shift, we compute Equation 1 iteratively for each shift $\delta^{t\to s}\in\{-\Delta,-\Delta+1,\dots,\Delta\}$, and choose the shift with minimum entropy. Here, Δ defines the maximum possible shift (in days) to estimate between the source and target regions.

However, due to the class imbalance of SITS, relying on expected entropy alone could result in choosing a shift where the model outputs confident predictions for only the most frequent classes while ignoring the less frequent classes. This would hinder the adaptation of the model for the less frequent target classes. To address this problem, we propose to also choose the shift based on the diversity of the predicted marginal distribution. The marginal is given by:

$$p_\theta(y)=\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)\sim\mathcal{D}^t}\left[p_\theta\big(y|\left(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s}\right)\big)\right], \tag{2}$$

where we compute the expected predictions of the model parameterized by θ when input shifted target data.

Optimally, the marginal distribution should match the class distribution of the target domain, as this indicates a shift where the model predicts a diverse set of classes according to their actual frequency. However, as we assume target domain labels are unavailable, so is the target class distribution. Instead, inspired by metrics for evaluating image generative models, we consider two options to address this: the Inception score [45] (IS), and the activation maximization score [46] (AM). Both metrics consider the entropy and marginal of a pre-trained model, but IS scores the marginal distribution by its similarity to a uniform distribution, whereas AM uses the actual class distribution.

As these metrics were originally proposed to evaluate the quality of generated images, we describe next how they are repurposed for temporal shift estimation. Finally, we describe an algorithm where IS is used to bootstrap the temporal shift for estimating the target class distribution with pseudo-labels and enable a better temporal shift estimate with AM.

3.3.1. Inception Score

We compute IS for a temporal shift δ by:

$$\begin{aligned}&\mathrm{IS}(\delta^{t\to s},\theta)\\&=\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)}\left[D_{\mathrm{KL}}\big(p_\theta\big(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\big)\;\big\|\;p_\theta(y)\big)\right] &(3)\\&=H(p_\theta(y))-\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)}\left[H\big(p_\theta\big(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\big)\big)\right] &(4)\end{aligned}$$

where $D_{\mathrm{KL}}(\cdot\,\|\,\cdot)$ is the KL-divergence between two distributions, here the conditional distribution $p_\theta\big(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta)\big)$ and marginal distribution $p_\theta(y)$ predicted with model parameters θ. Higher values of IS indicate a better δ, as when the conditional and marginal distributions are different, this corresponds to a temporal shift where the former has low entropy (*i.e.*, the model is confident), and the latter has high entropy (*i.e.*, the model predicts a diverse set of classes). Hence, we estimate the temporal shift $\delta^{t\to s}$ with IS by:

$$\delta^{t\to s}_{IS}(\theta^s)=\underset{\delta^{t\to s}\in\{-\Delta,\dots,\Delta\}}{\operatorname{argmax}}\ \mathrm{IS}(\delta^{t\to s},\theta^s), \tag{5}$$

where we choose the shift which maximizes IS for a source-trained model parameterized by θ^s applied to target data.

3.3.2. AM Score

A shortcoming of IS is that the highest score is achieved when $p_\theta(y)$ is uniform [47], which corresponds to an even distribution of classes in the target domain. For SITS, where the class distribution is often highly imbalanced, this may cause IS to estimate a suboptimal shift. AM [46] addresses this issue by taking the actual class distribution C^t into account:

$$\begin{aligned}\mathrm{AM}(\delta^{t\to s},\theta,C^t)=\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)}&\left[H\big(p_\theta\big(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\big)\big)\right]\\&+D_{\mathrm{KL}}(C^t\,\|\,p_\theta(y)).\end{aligned} \tag{6}$$

Algorithm 1: EstimateTemporalShift

1 **Input:** Source-trained parameters θ^s, target domain $\mathcal{D}^t$, target class distribution estimate $\hat{C}^t$
2 **if** $\hat{C}^t = \mathbf{0}$ **then**
3 Estimate temporal shift $\delta^{t\to s} \leftarrow \delta_{IS}^{t\to s}(\theta^s)$ (Eq. 5)
4 Compute pseudo labels for each $(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t) \in \mathcal{D}^t$: $\hat{y}_i^t \leftarrow \text{argmax}_y \left(p_{\theta^s}(y|\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t + \delta^{t\to s})\right)$
5 Estimate class distribution $\hat{C}_y^t \leftarrow \frac{1}{n^t}\sum_{i=1}^{n^t} \mathbf{1}_{\hat{y}_i^t = y}$ for $y \in \{1, \ldots, K\}$
6 Estimate temporal shift $\delta^{t\to s} \leftarrow \delta_{AM}^{t\to s}(\theta^s, \hat{C}^t)$ (Eq. 7)
7 **Output:** Temporal shift $\delta^{t\to s}$

AM consists of two terms: the first term is an entropy term on the conditional distribution, and the second is the KL-divergence between the underlying class distribution C^t and the marginal distribution. Lower values of AM indicate a better δ, as the model is confident in its predictions, and the actual class distribution of the data matches the predicted distribution of classes. To estimate the temporal shift $\delta^{t\to s}$ with AM, we thus compute:

$$\delta_{AM}^{t\to s}(\theta^s, C^t) = \underset{\delta^{t\to s}\in\{-\Delta,\ldots,\Delta\}}{\text{argmin}} \text{AM}(\delta^{t\to s}, \theta^s, C^t). \quad (7)$$

where we choose the shift which minimizes AM.

However, as the target domain is unlabeled, we cannot assume knowledge of the target class distribution C^t. To address this, we propose to approximate the target class distribution with pseudo-labels as shown in Algorithm 1. First, we use IS (Eq. 5) to estimate an initial shift $\delta^{t\to s}$ (line 3). This initial estimate allows us to shift the target domain so that more accurate pseudo-labels can be generated with a source-trained model. We then use the pseudo-labels to estimate the target class distribution $\hat{C}^t$ (lines 4-5). Finally, we re-estimate the temporal shift more accurately with AM and $\hat{C}^t$ (line 6).

3.4. TimeMatch Learning

With our method for estimating the temporal shift, we can reduce the domain discrepancy between the source and target domains. The TimeMatch learning algorithm uses the temporal shift to train the student model for the target domain from teacher-generated pseudo-labels via the FixMatch [28] loss and EMA training [40]. We present the complete TimeMatch algorithm in Algorithm 2, and describe the details of each step in the following.

3.4.1. Pre-training on the Source Domain

As we rely on the teacher to generate pseudo-labels to train the student, it is important to obtain a good initialization for both models. Additionally, temporal shift estimation requires a source-trained model. Thus, we first use the labeled source domain to obtain source-trained model parameters θ^s. Given a batch of labeled source data from $\mathcal{D}^s$, we optimize the following loss function:

$$\mathcal{L}^s = \frac{1}{B}\sum_{i=1}^{B} L\big(p_{\theta^s}\big(y|(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s)\big), y_i^s\big), \quad (8)$$

where $L(\cdot,\cdot)$ is a classification loss (*e.g.* cross-entropy or focal loss [48]) and B the batch size. After pre-training, we initialize the parameters of the student θ and teacher θ' from θ^s (line 2).

3.4.2. TimeMatch Loss

The TimeMatch loss is based on FixMatch [28]. As part of the consistency regularization, FixMatch applies two types of augmentation functions: *weakly*-augmented $a(\cdot)$ and *strongly*-augmented $A(\cdot)$, corresponding to simple and extensive augmentations of the input. We describe the form of augmentations we use for $a(\cdot)$ and $A(\cdot)$ in Section 4.4. Let $\delta^{s\to t}$ and $\delta^{t\to s}$ be temporal shifts estimated given by Algorithm 1 using the teacher (line 5-7).

The TimeMatch loss consists of two terms: a supervised loss $\mathcal{L}^{s\to t}$ applied to the adapted source domain $\mathcal{D}^{s\to t}$ and an unsupervised loss $\mathcal{L}^t$ applied to the unlabeled target domain $\mathcal{D}^t$. Using $\delta^{s\to t}$, we align the source domain with the target domain and optimize:

$$\mathcal{L}^{s\to t} = \frac{1}{B}\sum_{i=1}^{B} L\big(p_\theta\big(y|A(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s + \delta^{s\to t})\big), y_i^s\big), \quad (9)$$

using source labels y_i^s to update the student θ on strongly augmented source data shifted by $\delta^{s\to t}$. This loss makes it possible for the student to learn the target phenology from shifted source data (line 10).

To generate pseudo-labels for the target domain, we obtain the predicted class distribution from the teacher when input source-shifted target data:

$$\boldsymbol{q}_i^t = p_{\theta'}\big(y|a\big(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t + \delta^{t\to s}\big)\big), \quad (10)$$

where the teacher θ' is input a weakly-augmented target sample, shifted by $\delta^{t\to s}$. Then, we use

$$\hat{y}_i^t = \text{argmax}(\boldsymbol{q}_i^t) \quad (11)$$

as pseudo-label (line 11). The student θ is then updated on strongly-augmented target data for confident pseudo-labels (line 10):

$$\mathcal{L}^t = \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}_{\max(\boldsymbol{q}_i^t)>\epsilon} L\big(p_\theta\big(y|A\big(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t\big)\big), \hat{y}_i^t\big), \quad (12)$$

where $\mathbf{1}$ is the indicator function, and ϵ is the confidence threshold for using a pseudo-label. With this loss, the student is adapted to the target domain data using the pseudo-labeled target data. The total loss minimized by the student in TimeMatch is:

$$\mathcal{L}^{all} = \mathcal{L}^{s\to t} + \lambda\mathcal{L}^t, \quad (13)$$

where λ is a scalar hyperparameter to control the trade-off between the supervised and the unsupervised loss (line 13).

Algorithm 2: TimeMatch

1 **Input:** Labeled source domain $\mathcal{D}^s$, unlabeled target domain $\mathcal{D}^t$, source-trained parameters θ^s, total epochs n and iterations m, pseudo label threshold ϵ, trade-off value λ, EMA decay rate α, learning rate η
2 Initialize student parameters $\theta \leftarrow \theta^s$ and teacher parameters $\theta' \leftarrow \theta^s$
3 Initialize estimated target class distribution $\hat{C}^t = \mathbf{0}$
4 **for** epoch = 1 **to** n **do**
5 Estimate temporal shift with teacher: $\delta^{t\to s} \leftarrow$ EstimateTemporalShift(θ', $\mathcal{D}^t$, $\hat{C}^t$)
6 **if** epoch = 1 **then**
7 Initialize $\delta^{s\to t} \leftarrow -\delta^{t\to s}$
8 **for** iteration = 1 **to** m **do**
9 Sample mini-batches of size B from source $\mathcal{S} = \{(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s, y_i^s)\}_{i=1}^B$ and target $\mathcal{T} = \{(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t)\}_{i=1}^B$
10 With $\mathcal{S}$ shifted by $\delta^{s\to t}$, compute source loss $\mathcal{L}^{s\to t}$ (Eq. 9)
11 For each example in $\mathcal{T}$ shifted by $\delta^{t\to s}$, generate teacher prediction $\boldsymbol{q}_i^t$ and pseudo labels $\hat{y}_i^t$ (Eq. 10 and 11)
12 With $\mathcal{T}$ and confident pseudo labels $\hat{y}_i^t$ with $\max(\boldsymbol{q}_i^t) > \epsilon$, compute target loss $\mathcal{L}^t$ (Eq. 12)
13 Update student by gradient: $\theta \leftarrow \theta - \gamma\nabla_\theta(\mathcal{L}^{s\to t} + \lambda\mathcal{L}^t)$
14 Update teacher by EMA: $\theta' \leftarrow (1-\alpha)\theta + \alpha\theta'$
15 Re-estimate class distribution: $\hat{C}_y^t \leftarrow \frac{1}{mB}\sum_i \mathbf{1}_{\hat{y}_i^t = y}$ for $y \in \{1, \ldots, K\}$ (using all pseudo labels from epoch)
16 **Output:** Student parameters θ

3.4.3. EMA training and re-estimating temporal shift

By optimizing $\mathcal{L}^{all}$, the student and teacher are trained only for the target phenology, as $\mathcal{L}^{s\to t}$ shifts the time of the source to the target, while $\mathcal{L}^t$ keeps the target in its original time. This loss enables a source-trained model to adapt to the crop phenology of the target domain.

However, by doing so, the source domain is gradually "forgotten", and as a result, it becomes unnecessary to apply the temporal shift $\delta^{t\to s}$ for pseudo-labeling the target domain with the teacher. This causes $\delta^{t\to s}$ to gradually move to zero during TimeMatch learning. Thus, if $\delta^{t\to s}$ is fixed to the same shift, the target samples will be wrongly shifted, which results in incorrect pseudo-labels. To address this, we re-estimate the temporal shift for the teacher during TimeMatch learning. As Algorithm 1 chooses the shift based on the confidence and diversity of model predictions, re-estimating the temporal shift with the teacher ensures the generated pseudo-labels remain accurate during training.

However, if the teacher is a direct copy of the student, the model will rapidly adapt to the target domain, which requires the temporal shift to be re-estimated every few iterations. But doing so drastically increases training time, as Equation 7 requires forwarding a large sample of target data for each possible temporal shift. We address this by introducing EMA training, where the teacher is slowly updated via an EMA of the student parameters (line 14):

$$\theta' \leftarrow (1-\alpha)\theta + \alpha\theta', \tag{14}$$

where α is a decay rate. By choosing α close to 1, we reduce the rate at which the teacher adapts to the target domain, enabling the re-estimation of $\delta^{t\to s}$ to be done only once each epoch (line 5). Moreover, by averaging model weights via the EMA, we also obtain less noisy pseudo-labels [40]. By re-estimating the temporal shift, the teacher and the shift can both evolve jointly during training, resulting in better pseudo-labels for improved cross-region adaptation. Note that $\delta^{s\to t}$ is not re-estimated (line 7). The first shift estimate represents the shift of the data, whereas the re-estimated shift represents the shift of the teacher. By fixing $\delta^{s\to t}$ to the initial estimate, the source domain is kept aligned with the target domains during training, which enables semi-supervised learning.

4. Dataset and Materials

This section presents the TimeMatch dataset [29] and the materials for our experiments. We first introduce the crop classification model we use, followed by a description of the dataset and its pre-processing. Then, we describe the competitors and our implementation. Our source code is publicly available, and contains the implementation of TimeMatch and the competitors, a link to download our dataset, and the full experimental results: `https://github.com/jnyborg/timematch`.

4.1. Network Architecture

As our model, we use PSE+TAE, a state-of-the-art object-based crop classifier introduced by Sainte Fare Garnot *et al.* [12]. The network consists of two modules: the pixel-set encoder (PSE) and the temporal attention encoder (TAE).

The PSE module handles the spatial and spectral context of SITS. Rather than applying convolutions, which are time and memory-consuming when applied to irregularly sized parcels, PSE samples a random pixel-set of size S among the N_i available pixels within a parcel. As spatial information is lost by doing so, the PSE supports an optional extra input

with various geometrical properties of the given parcel, such as its area. We do not input this extra feature to avoid biasing the model towards the shapes of parcels in the source region, which typically change depending on the local farmer practices. Thus, we only input the sequence $\boldsymbol{x}_i \in \mathbb{R}^{T_i \times N_i \times C}$, which is then encoded by the PSE for each time step independently.

The TAE module handles the temporal context by applying self-attention [49]. Based on its performance and computational efficiency, we use the lightweight version of the TAE [50], which is a simplified self-attention network. The additional input $\boldsymbol{\tau}_i$ is input to TAE by encoding the days via a sinusoidal positional encoding function and adding the result to the output of PSE. As the positional encoding does not support negative inputs, we support negative temporal shifts by offsetting each $\boldsymbol{\tau}_i$ by the maximum temporal shift Δ. Given the sequence of PSE-embeddings and the encoded $\boldsymbol{\tau}_i$, TAE outputs a single embedding, which is then classified by a multi-layer perceptron to produce class probabilities $p(y|(\boldsymbol{x}_i, \boldsymbol{\tau}_i)) \in \mathbb{R}^K$.

4.2. The TimeMatch Dataset

The TimeMatch dataset [29] contains SITS from Sentinel-2 Level-1C products in top-of-atmosphere reflectance. Four Sentinel-2 tiles are chosen in various climates: 32VNH (Denmark), 30TXT (France), 31TCJ (France), and 33UVP (Austria). A map of the tiles is shown in Figure 3. We use all available observations with cloud coverage $\leq 80\%$ and coverage $\geq 50\%$ between January 2017 and December 2017. Figure 4 shows the resulting acquisition dates for the four tiles. We leave out the atmospheric bands (1, 9, and 10), keeping $C = 10$ spectral bands. The 20m bands are bilinearly interpolated to 10m.

For ground truth data, we retrieve geo-referenced parcel shapes and their crop type labels from the openly available Land Parcel Identification System (LPIS) records in Denmark[1], France[2], and Austria[3]. We select 15 major crop classes in Europe and label any remaining parcels as unknown. Figure 5 shows the selected classes and their frequency in each tile.

We pre-process the parcels by applying 20m erosion and removing all parcels with an area of less than 1 hectare. This reduces label noise by removing pixels near the border of parcels, which are often less representative of the given crop class compared to the pixels in the middle, and also by removing small or thin polygons, which are typically miscellaneous classes such as field borders. The SITS are pre-processed for object-based classification by cropping the pixels within each parcel to input sequences $\boldsymbol{x}_i \in \mathbb{R}^{T_i \times N_i \times 10}$. Each input is then randomly assigned to the train/validation/test sets of each Sentinel-2 tile by a 70%/10%/20% ratio. Note that this process assumes

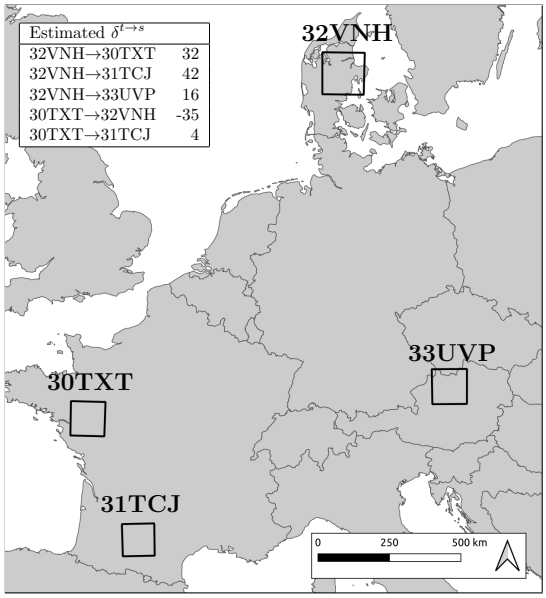


Figure 3: Locations of the four European Sentinel-2 tiles in the TimeMatch dataset. In the upper left corner, we show the temporal shifts $\delta^{t\to s}$ estimated by Algorithm 1 with a source-trained model.

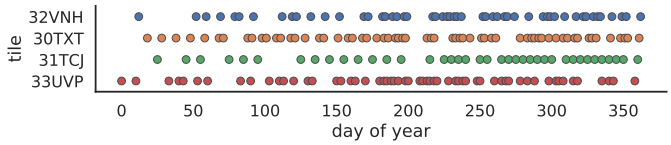


Figure 4: Acquisition dates for each Sentinel-2 tile in our dataset. The inputs are irregularly sampled with variable temporal length.

knowledge of parcel shapes in the target region. If this is not available, TimeMatch may instead be applied for pixel-based classification by inputting a single pixel ($S = 1$) to PSE+TAE.

We choose five different cross-region tasks (written as "source"→"target"): 32VNH→30TXT, 32VNH→31TCJ, 32VNH→33UVP, 30TXT→32VNH, and 30TXT→31TCJ. We focus on a subset of the 12 possible tasks to reduce the experimental running time. When a tile is the source region, all labels of the train and validation sets are available for training. When a tile is the target region, no labels are available, except for the final evaluation on the test set. Many UDA methods assume a labeled validation set for the target domain is available for training, and use it *e.g.* to select the best model [34]. However, this assumption is unrealistic, as if labels were available in real-world scenarios, they would be better used for training the model. Instead, we report all cross-region UDA test results with the model output at the end of training. Still, it is necessary to choose hyperparameters with a labeled validation set. Therefore, we tune hyperparameters with the validation set for only one task, 32VNH→30TXT, and apply the found hyperparameters to all remaining tasks.

The class distributions between regions differ significantly, and there may not be enough examples of a crop type in the source region for a model to learn their classi-

[1] https://kortdata.fvm.dk/download ("Marker")
[2] http://professionnels.ign.fr/rpg ("RPG")
[3] https://www.data.gv.at ("INVEKOS Schläge")

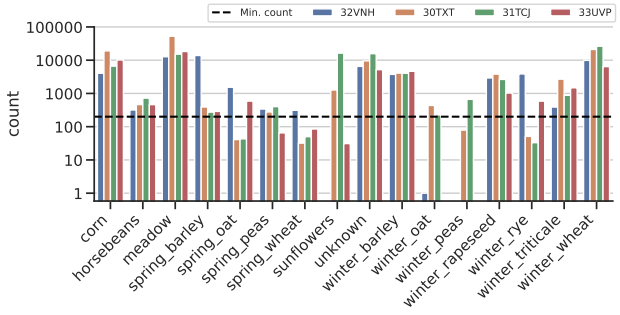


Figure 5: Class frequencies (log scale) for each Sentinel-2 tile in the TimeMatch dataset. The dashed line indicates the threshold for the source region when selecting a class as part of the K classes.

fication. Thus, when pre-training models on source data, we only use a subset of the available crop types with at least 200 examples in the source region (as indicated by the dashed line in Figure 5). The remaining classes are set as "unknown". When evaluating on the target data, we report results on the same selection of source classes no matter their frequency in the target.

4.3. Comparisons

Baselines. We consider the following baseline methods:

- *Source-Trained* is PSE+TAE trained on the source domain and applied to the target domain without domain adaptation. This represents the baseline cross-region performance of the model.
- *Target-Trained* is PSE+TAE trained with labeled target data using the same classes as the source-trained. This represents the upper bound cross-region performance possible if all target labels were available.
- *FixMatch* [28] is TimeMatch without temporal shift estimation and is thus an SSL method. This shows the benefit of the temporal shift estimation and reveals whether UDA or SSL is best for each adaptation task.

Competing UDA Methods. We compare TimeMatch to four of the top-performing UDA methods, which are based on learning domain-invariant features. We reproduce these methods for SITS by replacing the original feature extractor with PSE+TAE. We align the feature vector input to the classifier (the output of the TAE), similar to the original approach of these methods. We consider the following:

- *MMD* [23] learns domain invariant features by minimizing the maximum mean discrepancy metric.
- *DANN* [22] uses a domain classifier to align feature distributions through adversarial learning.
- *CDAN+E* [30] improves upon DANN by conditioning the domain classifier on the classification output and minimizing an entropy loss on target data.
- *JUMBOT* [34] aligns features between domains by a discrepancy measure based on optimal transport.

We note that time-series domain adaptation methods VRADA [35] and CoDATS [36] also employ DANN to align the features extracted by temporal network architecture. Thus, the only difference between VRADA, CoDATS, and the DANN approach mentioned here is the architecture of the feature extractor, which in our case is based on the temporal feature extractor of PSE+TAE.

PAN [20] learns domain-invariant features for cross-region adaptation by minimizing the MMD loss for features extracted by an RNN and self-attention-based crop classifier. Unfortunately, we were unable to gain access to the source code of PAN for comparison. As an alternative, we include the MMD comparison, which is similar to PAN, except the crop classifier is changed to PSE+TAE.

4.4. Implementation Details

All experiments are implemented in PyTorch [51] and trains on a single NVIDIA 1080 Ti GPU. Our implementation is based on the source code of PSE+TAE [50].

Pre-training on the source domain. To train models with the labeled source domain, we follow the original approach of PSE+TAE [12]. We train for 100 epochs with the Adam [52] optimizer with an initial learning rate of 0.001 and we decay the learning rate using a cosine annealing schedule [53]. We use weight decay of 0.0001, batch size 128, focal loss $\gamma = 1$. Inputs are normalized to $[0, 1]$ by dividing by the max 16-bit pixel value $2^{16} - 1$. The best source-trained model is selected using the source validation set. We augment the inputs by randomly sub-sampling 30 time steps and 64 pixels during training. The same setup is used for the target-trained model, using the target domain instead. For the final evaluation, we do not sample time steps or pixels, and instead input all available time steps and pixels to the model. This ensures the test results are deterministic, and also slightly improves results by providing all available data to the model.

TimeMatch. We use the same training setup as the source-trained model but instead train for 20 epochs with a lower initial learning rate of 0.0001. We define an epoch as 500 iterations to fix the frequency in which the temporal shift is re-estimated. We use maximum temporal shift $\Delta = 60$ days, as we did not observe shifts greater than 2 months for our dataset in Europe.

We set the trade-off hyperparameter $\lambda = 2.0$ in Eq. 13, EMA keep-rate $\alpha = 0.9999$, and pseudo-label threshold $\tau = 0.9$. A sensitivity analysis of these hyperparameters is provided in Section 5.5. For the FixMatch [28] augmentation functions, we use the identity function for the weak $a(\cdot)$ and randomly sub-sample time steps for the strong $A(\cdot)$. These are used for simplicity; it may be possible to further improve performance by more advanced SITS-based augmentations. At each iteration, we sample two mini-batches of size 128, one from the source and one from the target, in order to calculate the TimeMatch objective in Eq. 13. We use a class-balanced mini-batch sampler

Method	32VNH→30TXT	32VNH→31TCJ	32VNH→33UVP	30TXT→32VNH	30TXT→31TCJ	Avg.
Source-trained	28.3±1.9	29.0±5.2	43.4±4.0	24.9±2.0	70.3±1.9	39.2±3.0
FixMatch [28]	24.2±4.0	28.2±6.9	37.4±5.6	26.2±1.8	70.4±0.9	37.3±3.8
MMD [23]	36.6±0.7	35.5±0.6	49.7±2.0	32.5±2.0	61.6±2.6	43.2±1.6
DANN [22]	38.7±0.7	37.3±0.6	52.0±1.4	34.0±1.8	71.0±0.2	46.6±0.9
CDAN+E [30]	39.3±0.6	37.9±0.3	51.5±2.9	36.5±1.3	71.7±0.6	47.4±1.1
JUMBOT [34]	36.8±0.2	33.6±1.3	50.5±0.6	35.6±3.0	63.7±3.0	44.0±1.6
TimeMatch	**57.4±1.5**	**47.0±0.9**	**61.7±4.9**	**52.1±1.4**	**73.0±0.5**	**58.2±1.8**
Target-trained	74.6±0.6	72.4±1.4	86.9±2.7	90.6±4.3	85.7±0.7	82.0±1.9

Table 1: Macro F1-score (%) results on our dataset for the unsupervised cross-region adaptation setting.

for the source domain to ensure each source mini-batch contains roughly the same number of samples for each class. This reduces the class imbalance problem for the source domain for improved performance. Additionally, we apply domain-specific batch normalization [54–56] by forwarding the source and target mini-batches separately instead of concatenated. This ensures the batch normalization [57] statistics are calculated separately for each domain, for improved adaptation.

Existing UDA Methods. We re-implement the competitors MMD, DANN and CDAN+E following the domain adaptation library in [58], and JUMBOT from the original source code [34]. All methods are initialized from a source-trained model, train for 20 epochs, similar to TimeMatch. We also tune their hyper-parameters on the task 32VNH→30TXT. The full details of the implementation and chosen hyper-parameters can be found in our source code.

5. Experimental Results

5.1. Main Results

Table 1 shows the performance obtained with our approach and the re-implemented baselines and competitors. We report the mean and standard deviation of macro F1 scores, calculated from the results of three runs with different dataset splits. We observe a source-trained model transfers very poorly to a different target region, with an average loss of 43% in F1-score on target data, which strongly motivates UDA. However, existing UDA methods only slightly increase the performance of the source-trained model when applied to SITS. In comparison, our approach significantly outperforms all existing methods (+11% on average). This shows that addressing the temporal shift is crucial for the cross-region adaptation problem of SITS.

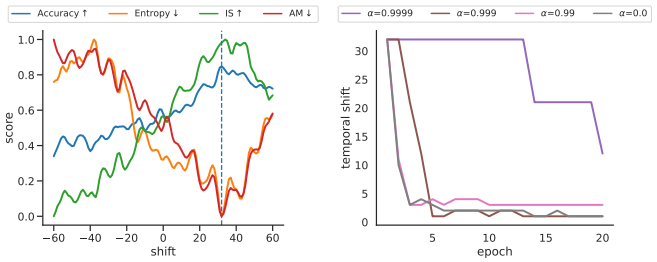
The results of the target-trained model are the highest achievable performance if target labels were available. Our approach recovers a significant part of this, but we also find that there still is room for improvement. We note that the results of the target-trained model have large variation between cross-region tasks since the F1-score is computed for the same classes as the source-trained model, no matter their frequency in the target domain. If some classes have too few examples in the target, the model may be unable to properly learn their classification. But as we aim to compare the performance of the target-trained model with the other methods, it has to consider the same set of classes.

Additionally, the results show that FixMatch often fails to improve the source-trained model, indicating that semi-supervised learning alone is not able to transfer models across regions. Only by incorporating temporal shift estimation via TimeMatch are we able to use SSL. Interestingly, the performance of FixMatch is bad for all tasks except 30TXT→31TCJ. As these two regions are geographically close, their temporal shift is also closer to zero (see the top-left table in Figure 3). This indicates that the problem of transferring models across regions changes from UDA to SSL, depending on the temporal shift between regions. Notably, TimeMatch still outperforms FixMatch for this task, as our approach can dynamically change between UDA and SSL depending on how close to zero the estimated shifts are. This makes TimeMatch highly practical to solve real-world cross-region problems, as, without labels for the target region, we do not know if it is better to apply UDA or SSL.

5.2. Analysis of Temporal Shift Estimation

In Figure 6a, we show the change in the overall accuracy of a source-trained model when applied to target data with different temporal shifts for 32VNH→30TXT. We also show the change in entropy, IS, and AM scores of the model. We observe a significant increase in accuracy by temporally shifting the target data. Calculating the statistics of entropy, IS, and AM from the predictions of the model works well as an unlabeled proxy to accuracy. We aim to estimate the shift with the highest accuracy (dashed blue line) for the highest quality pseudo-labels. For the shown example, the minimum of both entropy and AM correspond to the best shift. However, we find the AM to be the most consistent across different adaptation tasks.

In Figure 6b, we show the rate at which the estimated temporal shift for the teacher goes to zero in TimeMatch learning when training with different EMA decay rates.



(a) Score of temporal shifts. (b) Temporal shift during training.

Figure 6: (a) Overall accuracy, entropy, IS, and AM scores of a source-trained model when applied to the target domain with different shifts. The dashed line indicate the most accurate shift. (b) The re-estimated temporal shifts of the teacher model during TimeMatch learning with different EMA decay rates.

When the shift changes, the previous estimate becomes sub-optimal for generating accurate pseudo-labels. We address this by re-estimating the temporal shift during training. We observe that low decay rates (*e.g.* 0.99) require the shift to be re-estimated after a few iterations, which is inefficient. In comparison, a decay rate of 0.9999 allows us to only re-estimate the shift only once every epoch.

The table in the upper left corner of Figure 3 shows the initial temporal shifts estimated by our method. We find the estimated shifts are connected to the climatic differences between regions. For example, the temporal shift ($\delta^{t\to s}$) from the warmer 30TXT (western part of France) to the colder 32VNH (Denmark) is estimated as 32 days. Due to the warmer climate, crops in 30TXT mature earlier than in 32VNH, and a positive shift is required to align the former with the latter. In the other direction, the opposite is true, and indeed, we estimate a negative temporal shift of -35 days. Note that these are may not be exact inverses, as they are estimated with two different models trained on different source datasets.

5.3. Visual Analysis

In Figure 7, we visualize t-SNE [59] embedded TAE features from the source-trained, CDAN+E, TimeMatch, and target-trained models for 32VNH→30TXT. The colors of the points represent their class (black is the unknown class). With TimeMatch, the target features are better clustered into their respective classes compared to the best competing method CDAN+E, which does not result in much better feature separation than the source-trained model. The target-trained plot shows the best possible learned features when training with all available target labels. Even with labels, the classes are not perfectly separated, *e.g.* for unknown/meadow or winter triticale/winter wheat.

Figure 8 visualizes example predictions of the source-trained and TimeMatch models compared to the ground truth. The colors represent the same classes as before. We observe a large class confusion for the source-trained model, in particular between winter barley (blue) and winter wheat (dark pink), which are also not separated well in Figure 7. Without using any target labels, TimeMatch resolves this issue, resulting in predictions that closely resemble the ground truth.

Ablation	32VNH→30TXT
No EMA ($\alpha = 0.0$)	49.9±3.7
No source temporal shift ($\delta^{s\to t} = 0$)	51.9±1.9
No balanced batch sampler for source	53.3±3.6
IS instead of AM	56.3±2.6
Entropy instead of AM	56.9±1.8
No domain-specific batch norm.	56.9±4.1
TimeMatch	57.4±1.5

Table 2: Ablation study of TimeMatch components, sorted by increasing F1-score (%).

5.4. Ablation Study

To better understand how TimeMatch is able to obtain state-of-the-art results, we perform an ablation study on its components for the task 32VNH→30TXT. We report the results in Table 2.

We first study the impact of the EMA training. Instead of the EMA, we set the teacher as a direct copy of the student (No EMA). We observe that training without the EMA introduces a significant drop in F1-score—though the performance is still better than the competing methods. Setting $\delta^{s\to t} = 0$ disables the temporal shift of the source domain, and the student is trained with datasets with increased domain discrepancy. We observe a significant decrease in F1-score as a result. Disabling the balanced mini-batch sampler for the source domain also leads to a degradation of the performance. If the model is trained with class imbalanced source data, the teacher will make biased pseudo-labels for the samples from the target domain [60]. This hinders the TimeMatch learning process, as pseudo-labels for infrequent classes in the source domain are less likely to be generated for the target. By applying a balanced mini-batch sampler for the source, we address this problem by ensuring each source batch contains roughly the same number of samples for each category. Estimating the temporal shift with IS or entropy instead of AM results in a slight performance drop. Experimentally, however, we find AM to be most consistent across different tasks in estimating the temporal shift. Domain-specific batch normalization is simple to implement, as it just requires forwarding source and target batches separately instead of concatenated. Disabling this component results in a small average performance loss with notably higher variance.

5.5. Sensitivity Analysis

Next, we study the sensitivity of the TimeMatch hyper-parameters. The results are shown in Figure 9. Higher values of α lead to better results, with a decay rate of 0.9999 being the best. However, increasing it to 1.0, so the teacher

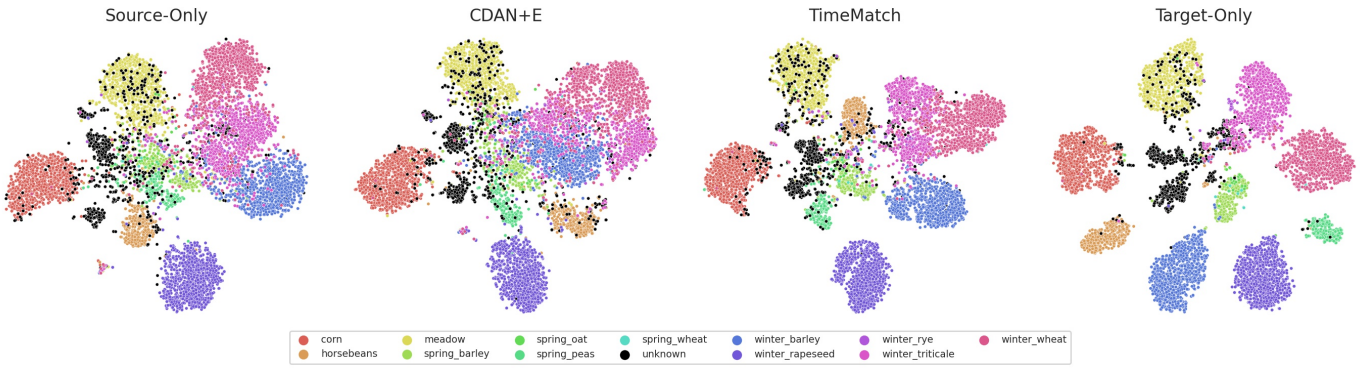
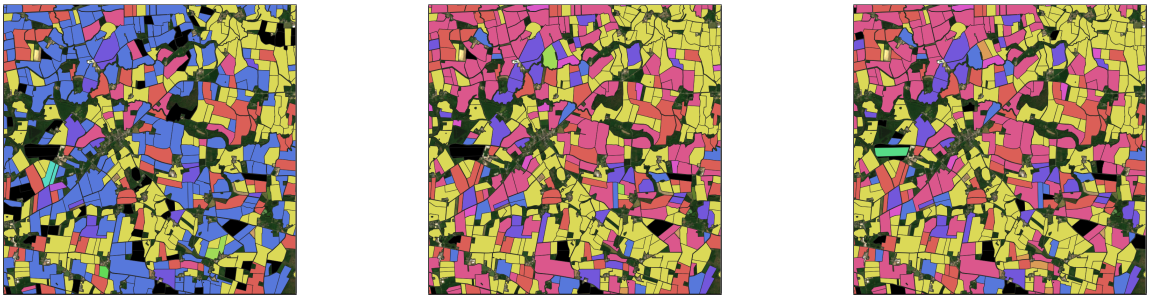


Figure 7: t-SNE [59] visualizations of PSE+TAE features for the 32VNH→30TXT cross-region task.



(a) Source-Only (b) TimeMatch (c) Ground Truth

Figure 8: Example predictions for the 32VNH→30TXT cross-region task comparing (a) Source-Only, (b) TimeMatch, and (c) the corresponding ground truth. The colors map to crop types following the legend in Figure 7.

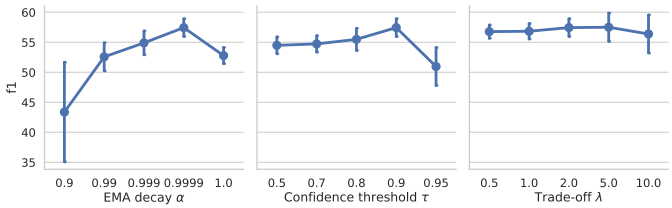


Figure 9: Sensitivity analysis of TimeMatch for the EMA decay rate, pseudo-label confidence threshold, and the trade-off in Eq. 13. The error bars show standard deviation.

is not updated, results in a drop in F1, as the teacher cannot benefit from the knowledge learned by the student. The confidence threshold ϵ controls the trade-off between the quality and quantity of pseudo-labels. A threshold of 0.9 gives the best F1 score and further increasing the threshold to 0.95 drops performance as a result of too few pseudo-labels, which particularly decreases performance for the less frequent classes. Finally, the trade-off parameter λ controls the importance of the source domain loss $\mathcal{L}^{s\to t}$ with respect to the target domain loss $\mathcal{L}^{t}$. We observe that this hyperparameter is less important than the other two. Setting $\lambda = 2.0$ gives the best F1-score. Increasing λ too much, however, results in large variance.

6. Conclusion

This paper presented TimeMatch, a novel cross-region adaptation method for SITS. Unlike previous methods that solely match the feature distributions across domains, TimeMatch explicitly captures the underlying temporal discrepancy of the data by estimating the temporal shift between two regions. Through TimeMatch learning, we adapt a crop classifier trained in a source region to an unlabeled target region. This is achieved by a learning algorithm that unifies temporal shift estimation with semi-supervised learning, where pseudo-labels are generated for unlabeled, temporally shifted target data to train the classifier for the target region. Lastly, we presented the TimeMatch dataset, a new large-scale cross-region UDA dataset with SITS from four different regions in Europe. Evaluated on this dataset, TimeMatch outperforms all existing approaches by 11% in F1-score on average across five different adaptation tasks, setting a new state-of-the-art in unsupervised cross-region adaptation.

We hope our proposed method and released dataset will encourage the remote sensing community to consider the challenging cross-region adaptation problem and its temporal aspect.

7. Acknowledgements

The work of Joachim Nyborg was funded by the *Innovation Fund Denmark* under reference *8053-00240*.

References

[1] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, et al., Sentinel-2: ESA's optical high-resolution mission for GMES operational services, Remote Sensing of Environment 120 (2012) 25–36.

[2] B. C. Reed, J. F. Brown, D. VanderZee, T. R. Loveland, J. W. Merchant, D. O. Ohlen, Measuring phenological variability from satellite imagery, Journal of Vegetation Science 5 (5) (1994) 703–714.

[3] F. Vuolo, M. Neuwirth, M. Immitzer, C. Atzberger, W.-T. Ng, How much does multi-temporal Sentinel-2 data improve crop type classification?, International Journal of Applied Earth Observation and Geoinformation 72 (2018) 122–130.

[4] J. B. Odenweller, K. I. Johnson, Crop identification using landsat temporal-spectral profiles, Remote Sensing of Environment 14 (1) (1984) 39–54. doi:10.1016/0034-4257(84)90006-3.

[5] C. Pelletier, G. I. Webb, F. Petitjean, Temporal convolutional neural network for the classification of satellite image time series, Remote Sensing 11 (5) (2019) 523. doi:10.3390/rs11050523.

[6] L. Zhong, L. Hu, H. Zhou, Deep learning based multi-temporal crop classification, Remote Sensing of Environment 221 (2019) 430–443.

[7] E. Ndikumana, D. Ho Tong Minh, N. Baghdadi, D. Courault, L. Hossard, Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France, Remote Sensing 10 (8) (2018) 1217.

[8] M. Rußwurm, M. Korner, Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 11–19.

[9] D. Ienco, R. Gaetano, C. Dupaquier, P. Maurel, Land cover classification via multitemporal spatial data by deep recurrent neural networks, IEEE Geoscience and Remote Sensing Letters 14 (10) (2017) 1685–1689.

[10] D. H. T. Minh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, P. Maurel, Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR Sentinel-1, IEEE Geoscience and Remote Sensing Letters 15 (3) (2018) 464–468.

[11] M. Rußwurm, M. Körner, Self-attention for raw optical satellite time series classification, ISPRS Journal of Photogrammetry and Remote Sensing 169 (2020) 421–435. doi:10.1016/j.isprsjprs.2020.06.006.

[12] V. Sainte Fare Garnot, L. Landrieu, S. Giordano, N. Chehata, Satellite image time series classification with pixel-set encoders and temporal self-attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12325–12334.

[13] M. Rußwurm, M. Körner, Multi-temporal land cover classification with sequential recurrent encoders, ISPRS International Journal of Geo-Information 7 (4) (2018) 129. doi:10.3390/ijgi7040129.

[14] R. Interdonato, D. Ienco, R. Gaetano, K. Ose, DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn, ISPRS Journal of Photogrammetry and Remote Sensing 149 (2019) 91–104. doi:https://doi.org/10.1016/j.isprsjprs.2019.01.011.

[15] D. Tuia, C. Persello, L. Bruzzone, Domain adaptation for the classification of remote sensing data: An overview of recent advances, IEEE Geoscience and Remote Sensing Magazine 4 (2) (2016) 41–57.

[16] B. Lucas, C. Pelletier, D. Schmidt, G. I. Webb, F. Petitjean, A bayesian-inspired, deep learning-based, semi-supervised domain adaptation technique for land cover mapping, Machine Learning (2021) 1–33.

[17] L. Kondmann, A. Toker, M. Rußwurm, A. Camero, D. Peressuti, G. Milcinski, P.-P. Mathieu, N. Longépé, T. Davis, G. Marchisio, L. Leal-Taixé, X. X. Zhu, DENETHOR: The DynamicEarthNET dataset for harmonized, inter-operable, analysis-ready, daily crop monitoring from space, in: Neural Information Processing Systems Datasets and Benchmarks Track, 2021.

[18] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2009) 1345–1359. doi:10.1109/TKDE.2009.191.

[19] B. Kellenberger, O. Tasar, B. Bhushan Damodaran, N. Courty, D. Tuia, Deep Domain Adaptation in Earth Observation, John Wiley & Sons, Ltd, 2021, Ch. 7, pp. 90–104.

[20] Z. Wang, H. Zhang, W. He, L. Zhang, Phenology alignment network: A novel framework for cross-regional time series crop classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2940–2949.

[21] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine Learning 79 (1) (2010) 151–175. doi:10.1007/s10994-009-5152-4.

[22] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International Conference on Machine Learning, PMLR, 2015, pp. 1180–1189.

[23] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance, CoRR abs/1412.3474 (2014).

[24] G. Wilson, D. J. Cook, A survey of unsupervised deep domain adaptation, ACM Transactions on Intelligent Systems and Technology 11 (5) (2020) 1–46. doi:10.1145/3400066.

[25] B. Lucas, C. Pelletier, D. Schmidt, G. I. Webb, F. Petitjean, Unsupervised domain adaptation techniques for classification of satellite image time series, in: International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2020, pp. 1074–1077. doi:10.1109/IGARSS39084.2020.9324339.

[26] O. Chapelle, B. Scholkopf, A. Zien, Semi-supervised learning, IEEE Transactions on Neural Networks 20 (3) (2009) 542–542.

[27] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on Challenges in Representation Learning, ICML, Vol. 3, 2013, p. 896.

[28] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, C.-L. Li, FixMatch: Simplifying semi-supervised learning with consistency and confidence, Advances in Neural Information Processing Systems 33 (2020).

[29] J. Nyborg, C. Pelletier, S. Lefèvre, I. Assent, The TimeMatch Dataset (2021). doi:10.5281/zenodo.5636422.

[30] M. Long, Z. Cao, J. Wang, M. I. Jordan, Conditional adversarial domain adaptation, in: Advances in Neural Information Processing Systems, 2018, pp. 1647–1657.

[31] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, The Journal of Machine Learning Research 17 (1) (2016) 2096–2030.

[32] B. Sun, K. Saenko, Deep coral: Correlation alignment for deep domain adaptation, in: European Conference on Computer Vision, Springer, 2016, pp. 443–450.

[33] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, N. Courty, DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation, in: European Conference on Computer Vision, 2018, pp. 447–463.

[34] K. Fatras, T. Séjourné, N. Courty, R. Flamary, Unbalanced minibatch optimal transport; applications to domain adaptation, in: International Conference on Machine Learning, 2021.

[35] S. Purushotham, W. Carvalho, T. Nilanon, Y. Liu, Variational recurrent adversarial deep domain adaptation, in: International Conference on Learning Representations, 2017.

[36] G. Wilson, J. R. Doppa, D. J. Cook, Multi-source deep domain adaptation with weak supervision for time-series sensor data, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1768–1778.

[37] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2066–2073.

[38] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, Unsuper-

vised visual domain adaptation using subspace alignment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2013, pp. 2960–2967.
[39] Q. Xie, M.-T. Luong, E. Hovy, Q. V. Le, Self-training with noisy student improves ImageNet classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10687–10698.
[40] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: Advances in Neural Information Processing Systems, 2017, pp. 1195–1204.
[41] M. Sajjadi, M. Javanmardi, T. Tasdizen, Regularization with stochastic transformations and perturbations for deep semi-supervised learning, Advances in Neural Information Processing Systems 29 (2016) 1163–1171.
[42] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: International Conference on Learning Representations, 2017.
[43] P. Panareda Busto, J. Gall, Open set domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 754–763.
[44] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in: International Conference on Learning Representations, 2017.
[45] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, Advances in Neural Information Processing Systems 29 (2016) 2234–2242.
[46] Z. Zhou, H. Cai, S. Rong, Y. Song, K. Ren, W. Zhang, J. Wang, Y. Yu, Activation maximization generative adversarial nets, in: International Conference on Learning Representations, 2018.
[47] S. Barratt, R. Sharma, A note on the inception score, in: Workshop on Theoretical Foundations and Applications of Deep Generative Models, ICML, 2018.
[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2017, pp. 2980–2988.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[50] V. Sainte Fare Garnot, L. Landrieu, Lightweight temporal self-attention for classifying satellite images time series, in: International Workshop on Advanced Analytics and Learning on Temporal Data, Springer, 2020, pp. 171–181.
[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019) 8026–8037.
[52] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, 2015.
[53] I. Loshchilov, F. Hutter, SGDR: Stochastic gradient descent with warm restarts, in: International Conference on Learning Representations, 2017.
[54] Y. Li, N. Wang, J. Shi, J. Liu, X. Hou, Revisiting batch normalization for practical domain adaptation, Pattern Recognition 80 (03 2016). doi:10.1016/j.patcog.2018.03.005.
[55] W.-G. Chang, T. You, S. Seo, S. Kwak, B. Han, Domain-specific batch normalization for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7354–7362.
[56] K. Saito, D. Kim, S. Sclaroff, T. Darrell, K. Saenko, Semi-supervised domain adaptation via minimax entropy, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8050–8058.
[57] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.
[58] J. Jiang, B. Chen, B. Fu, M. Long, Transfer learning library, https://github.com/thuml/Transfer-Learning-Library (2020).
[59] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE., Journal of Machine Learning Research 9 (11) (2008).
[60] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.