Sensor Independent Cloud and Shadow Masking With Partial Labels and Multimodal Inputs

Alistair Francis

***Abstract*— A paradigm shift is underway in Earth observation, as deep learning (DL) replaces other methods for many predictive tasks. Nevertheless, most DL classification models for Earth observation are limited by their specificity with respect to both the sensors used (inputs) and classes predicted (outputs), leading to models that only perform well for specific satellites and on specific datasets. Cloud masking is typical of this, but is one of the most important tasks to generalize across sensors, given that it is required for all optical instruments. This work sets out a framework to relax DL's constraints on specific inputs and outputs, using cloud and shadow masking as a case-study. Centrally, a model which is sensor independent, and which can simultaneously learn from different labeling schemes is developed. The model, Spectral ENcoder for SEnsor Independence version 2 (SEnSeI-v2) extends the original version, by permitting multimodal data [in this case Sentinel-1 synthetic aperture radar (SAR) imagery and a digital elevation model (DEM)] to be ingested, along with several other architectural improvements. SEnSeI-v2, attached to SegFormer, is shown to have state-of-the-art performance, whilst being usable on a range of multispectral band combinations, alongside SAR and DEM inputs, without retraining. The labeling schemes of eight datasets are not made compatible through a reductive approach (e.g., converting to cloud versus noncloud), rather, an ambiguous cross-entropy loss is introduced that allows the model to learn from the different labeling schemes without sacrificing the class distinctions of each, leading to a model which predicts all of the constituent classes of the different datasets.**

***Index Terms*— Atmosphere, deep learning (DL), image analysis, multisource data fusion, optical data, synthetic aperture radar (SAR) data, thermal data.**

I. Introduction

DEEP learning (DL) approaches have shown great success when applied to satellite imagery for a number of predictive tasks, often with higher performance in comparison to traditional methods [1]. These traditional methods, though, are often applicable (or translatable) to multiple sensors by virtue of their use of physical rules. For example, Fmask [2]—originally developed for Landsat sensors—was straightforwardly translated to Sentinel-2, by removing the thresholding tests related to the thermal bands, which Sentinel-2 data does not include. Meanwhile, DL models are not readily adaptable to new sensors, because of their expectation of a fixed input structure (e.g., the same set of spectral bands, or for multimodal models, the same combination of instruments). The lack of immediate generalisability of DL models to multiple satellite sensors slows adoption and operational deployment. This is because the cycle of model development and validation, and the creation of datasets for training (in the case of supervised models) and validation (for all models) must be repeated for each sensor. Having previously introduced *sensor independence* with Spectral ENcoder for SEnsor Independence (SEnSeI) [3]—whereby a single model may be trained and used on multiple multispectral sensors—this work extends that effort, creating a cloud masking model which is further generalized than before, whilst achieving state-of-the-art performance.

Building on SEnSeI-v1, three main novel contributions are offered here. 1) Several architectural improvements are proposed for both SEnSeI and the DL model it is attached to, leading to a more accurate predictor. 2) Cloud masking is reformulated as a partial label learning problem, and a novel ambiguous cross-entropy loss is proposed, whereby the different labeling structures of the datasets are retained by the model, rather than simplified and reduced. 3) Support for multimodal inputs in SEnSeI is developed, with the model able to ingest data from other sources of raster data, e.g., Sentinel-1 synthetic aperture radar (SAR) and a digital elevation model (DEM) as optional, extra inputs complementing the primary multispectral image.

Regarding the partial label learning paradigm, more concretely, labels in the ground-truth represent possibilities (commonly referred to as *candidates* in the field of partial label learning), rather than precise, known solutions. This is a different scenario to both fuzzy and multilabel processes, which assert that `AND` statements can exist between different labels in the real world. Instead, one can consider partial labels as expressing `OR` statements in the labels. So, whilst it is assumed that a given input cannot truly correspond to multiple classes in the real world, the ground-truth does not necessarily pinpoint which specific class is true, only conveying possibilities.

The usefulness of multimodality is of great interest in contemporary remote sensing research, but so far little focus has been given to how it may affect the performance of cloud masking algorithms. Perhaps this is because neither of the two other obvious modalities one might use besides multispectral instruments (SAR and DEM) are sensitive to the cloud. Nevertheless, it may be that the information from SAR and DEM can help to disambiguate clouds from noncloud, despite

Manuscript received 27 October 2023; revised 25 March 2024; accepted 17 April 2024. Date of publication 19 April 2024; date of current version 1 May 2024. This work was supported by the European Space Agency.

The author is with the Φ-lab, ESRIN, 00044 Frascati, Italy (e-mail: alistair.francis@esa.int).

Digital Object Identifier 10.1109/TGRS.2024.3391625

not being sensitive to them. This hypothetical disambiguation could occur when spatial features between the multispectral image are masked and the SAR and DEM rasters correlate or not. When spatial features from the different instruments align and correlate, then it is likely that they are looking at the same surface (noncloud), and when they differ, then it is possible that it is because the multispectral sensor is viewing cloud features whilst the others still give information from the Earth's surface. This work seeks to explore such possibilities, using the sensor-independent model that is developed as a useful tool for rapidly testing different possible combinations of modalities, without any retraining being necessary.

Previous experiments with SEnSeI [3] tested a set of hypotheses about the generalisability of SEnSeI to different sensors, and how it affected the performance of the model it was attached to. These showed SEnSeI's generalization across different multispectral satellites (conveyed by its performance on multiple labeled datasets over different sensors). However, given the differences in labeling styles between these different datasets, the previous work simplified the classification task to a binary one: cloud versus noncloud.

In view of those previous findings, this work moves forward to explore the topic further with three points of focus as follows.

1) *Model Improvements:* The effects of the improvements to the model architecture are measured against the previous version of SEnSeI, and other published methods. Results found in Section V-A.
2) *Partial Label Learning:* The ability of a model to learn from partial labels, and how such a framework can unlock new capabilities whilst maintaining performance in simplified tasks when compared to a nonpartial labeling approach. Results found in Section V-B.
3) *Multimodality:* The effect of adding different modalities (SAR from Sentinel-1, and DEM data) to the inputs of a cloud masking algorithm, measuring whether this extra data is helpful or not, and in what circumstances. Results found in Section V-C.

The results for these three experiments are measured on Sentinel-2 cloud masking datasets. Therefore, a final experiment is included to verify that SEnSeI version 2 (SEnSeI-v2) is indeed sensor-independent, by showing its performance on Landsat 8 and 9 data. Then, Section VI moves on to discuss—in light of the results—how a data-centric approach could be more efficacious for the cloud masking community than a model-centric one. In particular, prioritizing the quality and quantity of available data, considering its sampling biases, and probing the utility of multimodal inputs.

II. Related Work

A. Cloud Masking

Locating where clouds and their associated shadows obstruct the view of the Earth's surface from space is a core problem that impinges on all optical satellite sensors' gathering of data. Cloud masking remains a challenging problem that both satellite operators and end-users of their data have a stake in. Understandably, many studies in cloud masking focus primarily on the quantitative performance of their methods against others. However, this work does not primarily seek to compete with other methods but rather offers concepts (sensor independence and partial label learning) that can complement and augment any approach. Given this focus, it does not make sense to solely focus this section on describing the various choices of model design, and their concomitant effects on performance, that exist in the literature. For this, several excellent review papers have compared and contrasted the different methods employed to mask clouds in satellite imagery [4], [5], [6], [7]. This section will focus instead on some of the remaining challenges that routinely degrade the practical utility of cloud masks.

Thin cloud and shadow detection is still unsatisfactory for many end-users of cloud masking algorithms. This is in part because the definition of cloud—and in particular thin cloud—is inherently difficult to pin down. Skakun et al. [8], for example, found large disagreements between expert annotators due to the subjective and diverse definitions used for the thin cloud. As Tarrio et al. [6] note, cirrus clouds are often situated at the boundary between what would be classed as cloudy and clear. Of course, as thin cirrus clouds demonstrate, cloudiness is not truly a categorical variable, it is a family of atmospheric phenomena which we typically associate with several parameters, including optical depth [9], cloud top height [10], and cloud type (e.g., cumulus and cirrus) [11], among others. All these parameters impact how clouds appear in the different spectral channels of a sensor situated in space above them.

Whilst the underlying physics and processes governing clouds are complex and diverse, the data with which we work is generally not capable of representing such rich information. Thermal sensors can provide important physical measurements of cloud and aerosol properties [12], but most multispectral sensors (e.g., Sentinel-2)—for which cloud masks are needed—do not give us much information regarding cloud physics or composition. Therefore, we are necessarily still in a situation where the majority of models output a coarse, categorical representation of the complex underlying system, and this coarse categorical representation is judged against an equally coarse categorical ground-truth dataset for validation. In the end, a boundary between what is and is not cloud must be drawn (both spatially at their edges, but also in terms of opacity and thickness), and this semantic boundary is always a site where interdataset [8], [13] and intermodel [4], [7] disagreement is high.

Motivated to optimize performance at this semantic boundary, the designs of several models are guided by the difficult issue of thin cloud retrieval. For example, Wu et al. [14] offered a rule-based model based on the observation that a thin cloud removes dark pixels from a given region. Many algorithms rightly focus on the 1.38 μm cirrus band in Sentinel-2 and Landsat 8 to detect thin clouds [2], [15], [16]. For DL approaches that consider larger spatial extents, the smoothness of thin cloud areas versus clear areas can be seen as a useful discriminative feature for the model to detect thin clouds (e.g., [17], [18]), whilst Zhang et al. [19] showed a

vision transformer-based architecture performs well for thin cloud detection.

B. Beyond Supervised Learning

Partial label learning, as will be shown in Section III-B, is a mode through which strictly supervised learning is relaxed. It has long been recognized as a more general learning paradigm to strictly supervised learning (e.g., [20], [21]). Two common families of methods can be defined for learning from partial labels: average-based strategies, and identification-based strategies. As Lv et al. [21] defined them, average-based strategies treat all classes labeled as possible as equally likely, whilst identification-based strategies seek to disambiguate the labels and treat the most likely label as the true class. Both of these approaches aim to *disambiguate* the problem, converting it into something resembling standard supervised learning. However, disambiguation-free learning has also been proposed by Zhang et al. [22].

Whilst partial label learning has not been previously proposed for cloud masking (or, to the best of our knowledge, any other problem in remote sensing), other approaches that reduce the need for precise labels have been. For example, Li et al. [23] showed how weak supervision can be leveraged to train a high-performance cloud masking model, where each patch during training is simply marked as cloudy or not cloudy, rather than using a pixel-wise mask. Several papers combine this patch-wise labeling approach with generative adversarial networks (e.g., [24], [25]). These patch-wise label methods show promise in reducing the amount of labeling time needed for cloud masking. However, it is not straightforward to extend this approach to cloud shadow detection, because patches containing only shadow pixels are difficult to find, given their tendency to be located close to clouds [23], [25]. The difference between these methods and partial label learning is the fact that they introduce ambiguity in the spatial dimension, whereas partial labels retain spatial exactness, but permit semantic ambiguity in the output classes of each pixel.

Categorizing the previous approaches as "weakly supervised," we can also consider training paradigms considered to be "unsupervised." For example, Xie et al. [26] recently proposed Auto-CM, a method that exploits the different spatio-temporal characteristics of atmospheric (clouds) and surface (clear) features, to perform unsupervised masking and cloud-free mosaicking on time series, through a self-supervised learning approach. By not using any specific labeled training datasets, this method is also somewhat sensor independent (though still requiring unlabeled multitemporal data from a new sensor), and is shown to perform well on Landsat 8, Sentinel-2, and PlanetScope data.

C. Multimodality

There has been a recent focus on multimodal models for Earth observation [27], given that many problems are poorly constrained when considering a single sensor's data. For example, Manakos et al. [28] fuse Sentinel-1 and Sentinel-2 data to accurately predict flood maps, noting that Sentinel-2 is commonly used for this task but is impractical when atmospheric conditions are unfavorable. Change detection using fused Sentinel-1 and Sentinel-2 data also shows promise [29]. Using the same combination of instruments—Sentinel-1 and Sentinel-2—Orynbaikyzy et al. [30] demonstrated a system for the mapping of crop types, remarking on the improvement in the performance of the multimodal approach versus single sensor models. These positive results are reinforced by Blickensdörfer et al. [31] who performed data fusion across Sentinel-1 and -2, Landsat 8, as well as topographical, meteorological, climatological, and environmental data to map crop types.

Closer to the field of cloud masking, cloud removal (where cloudy areas are inpainted with a predicted surface reflectance) approaches often use a multimodal approach, where a sensor typically unaffected by atmospheric conditions (most often SAR) is used to inpaint the cloudy regions [32], [33], [34]. Clearly, in a range of domains, fusion across different data sources permits more performant models, but necessitates the creation of multimodal datasets. Until the publication of the CloudSEN12 dataset [13], an openly available multimodal dataset for cloud masking did not exist. With this dataset, the community may now experiment with the utility of both SAR and DEM information when masking clouds.

III. Methods

This section provides an overview of the proposed approach, before moving on to detail the technical details of each of the primary contributions. The initial overview provides a context that is relevant to both this work and the previous implementation of SEnSeI [3].

Sensor independence can, presumably, be achieved through many different approaches, which do not necessarily correlate to the approach of SEnSeI. Where a nonsensor-independent model is specialized to the specific physical measurements made by its respective sensor, a sensor-independent one should be able to ingest and—with some level of success—use data that may come from some range or family of sensors. The level of sensor independence of a given method can be loosely defined by the breadth of different sensors that it can be used on. A restricted form of sensor independence, then, could be a model that is trained and used on a set of spectral bands that many different sensors measure (e.g., a model that uses only red green, and blue bands from many satellites). A more ambitious form of sensor independence, as examined in this work, can be achieved by designing a model that permits arbitrary combinations of spectral bands (and possibly other data, e.g., SAR or DEM) by a model.

In practice, SEnSeI achieves this by considering each band of data as a separate input. This contrasts with the standard approach of most multispectral vision models, which assume that the same physical measurement (e.g., "Red") appears at the same place in the inputted data and that such data consists of a fixed and unchanging set of bands. So, the task of a sensor-independent model is to use some arbitrary set of physical measurements (here assumed to be raster images) that come from sensors with different characteristics. In both SEnSeI-v1 and SEnSeI-v2, the problem is split into

two relatively independent steps. The first is to encode the information from the set of physical measurements into an embedding space that is of a fixed dimensionality. The second is to pass this embedding of fixed dimensionality to a DL model, such as DeepLabv3+ or SegFormer, or any other model that one might choose to apply.

In both versions of SEnSeI, descriptions of the physical characteristics of the sensor are given to the model, via a "descriptor vector." Whilst not exactly the same between versions 1 and 2 of SEnSeI, these descriptor vectors give the model a way to understand what data it is being given. Similar to the changes to the format of the descriptor vectors, the inner workings of each version of SEnSeI are different, however, both achieve the same goal. Namely, to build a representation of the data in a space of a fixed size, which is independent of the number of bands it is given. In both versions this is achieved via a pooling operation at the end of SEnSeI, which collapses the representation, making it independent of the number of input bands it is given.

Having covered the general approach taken by both versions of SEnSeI in pursuit of sensor independence, the following subsections focus on: the differences and novel aspects of SEnSeI-v2 (Section III-A), the ambiguous loss function (Section III-B), and the models that SEnSeI-v2 are used with (Section III-C).

A. Model Improvements

1) Spectral Encoding: SEnSeI uses descriptor vectors to provide information about the spectral characteristics of the satellite sensor's bands. Previously, in SEnSeI-v1 [3], the descriptor for a given band is a vector of length 3, with the minimum, peak, and maximum wavelengths of the spectral response curve of the band. This is a simple but somewhat crude representation, which has two obvious drawbacks.

The first drawback is that the exact shape of the spectral response curve is lost. The second failing is more complex. Using a single number to describe the change in wavelength between 400 nm and 12 μm does not adequately reflect the sharp nonlinearities in physical behavior that satellite observations have as a function of wavelength within that range. Whilst neural networks are able to represent nonlinear functions, it is difficult for a model to map the complex changes in the different sections of the spectrum (e.g., the "red edge" in the region around 700 nm, where a sharp change in reflectance occurs over vegetation).

One possible solution could be to fully describe the spectral response over the entire wavelength range, such that the descriptor is a long vector in which each value represents the detector's sensitivity in that wavelength region. This, however, creates several other issues. First, it adds a layer of complexity and prerequisite knowledge about the sensors that is not always available, in that the spectral response curve (or some approximation of it) is needed. Second, without careful regularization during training, it could lead quickly to overfitting. This is because of the large number of completely independent parameters each descriptor would have, and the still relatively small pool of possible spectral bands that one might encounter in the set of sensors used during training. Such a situation risks encouraging the model to focus on very small differences in the spectral response curves of different sensors (a single value in the descriptor representing the sensitivity at a specific wavelength on the edge of a spectral response curve, for example) in order to overfit to each sensors' datasets' biases, rather than on what we might expect are more useful features (larger discrepancies between spectral responses, which could cause measured top-of-atmosphere (TOA) reflectance values to differ significantly over the same surface).

Another solution is inspired by the work of Vaswani et al. [35] who proposed positional encodings to give transformer architectures explicit information about each input's position in a sequence. Transferring this idea directly to wavelength, a *spectral encoding* is the output of a set of sinusoidal functions, with a range of frequencies multiplied by the wavelength's value. Lower frequency components of the encoding change slowly with wavelength (giving coarse information about the wavelength), whilst high-frequency components oscillate more rapidly with wavelength (giving the model rich information about each local part of the wavelength range). For a wavelength λ, which is logarithmically scaled as

$$\lambda_{\text{norm}} = \log_{10}(\lambda - 300) - 2 \tag{1}$$

where λ is given in nanometres. so that the large differences between optical bands and thermal bands are not too extreme in the normalized value range. Then, a set of sinusoidal embeddings $f_i(\lambda)$ are computed as

$$f_i(\lambda) = \begin{cases} \sin(\omega_i \lambda_{\text{norm}}), & \text{if } i \text{ even} \\ \cos(\omega_i \lambda_{\text{norm}}), & \text{otherwise} \end{cases}$$

where

$$\omega_i = 10\,000^{-2i/N_\omega}, \quad \text{for } i = 1, \ldots, N_\omega.$$

These spectral encodings have some advantageous characteristics. Like the original descriptor vectors of SEnSeI-v1, they do not need the full spectral response function, requiring only the minimum and maximum wavelengths. They also provide the model with both coarse and fine details across the wavelength range, and change continuously and smoothly with wavelength, unlike a full spectral response function's features, which may be used by the model to overfit more easily to a specific band. These spectral encodings form the first part of the descriptor vectors shown in Fig. 1, with N_ω set as 32.

2) Multimodal Support: The spectral encoding described in the previous section is able to represent diverse multispectral bands, and these encodings form part of the descriptor vector used by SEnSeI-v2. However, descriptor vectors can also be constructed for nonoptical datatypes, with a straightforward scheme to extend the vector features. To do this, as shown in Fig. 1, some binary features are added to the end of the vector, denoting whether the data is from a certain instrument type. In the context of this work, data from multispectral (including thermal bands), SAR, and DEM are used, however, one can extend this concept to any other datatype—assuming it can be represented as a raster over the same area as the other bands used. When these binary variables are used, the space that

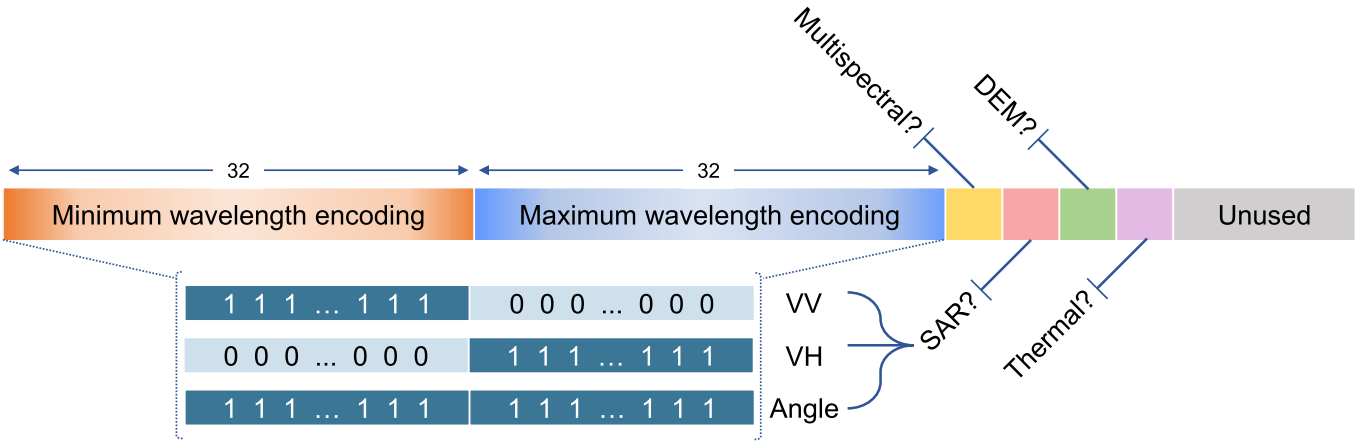

Fig. 1. Schematic of descriptor vectors used for SEnSeI-v2. The first 64 entries contain the spectral encodings of a band's minimum and maximum wavelength (if it is a spectral or thermal band). The later section of the vector contains binary flags used to add support for other kinds of nonoptical data. For SAR data, the first 64 positions—which for multispectral bands are used to encode the wavelengths—are instead used to indicate to the model whether a band is VV or VH backscatter, or the incidence angle. DEM data is also indicated with its own binary flag, in which case the first 64 entries are simply filled with 0. Some positions in the vector remain unused, allowing for future expansion to other data types.

the wavelength encodings occupy for multispectral bands is liberated, allowing for information (e.g., VV versus VH in the case of SAR) to be given to the model.

3) Multihead Attention: Computations within SEnSeI's permutation equivariant layers are done band-wise; each input band's descriptor corresponds to an output feature vector at each layer. Whilst a one-to-one correspondence between input descriptors and output feature vectors is needed, factoring in information from all other bands' corresponding feature vectors is also desirable, as it allows SEnSeI to consider the combinatorial effects of different bands on the outputted representation.

In SEnSeI-v1 this information sharing was performed with a "permutational block" which made N_b^2 pair-wise combinations for N_b bands, by concatenating their feature vectors in pairs, and then sent them through a set of fully connected layers. Then, it pooled those pair-wise combinations back to N_b feature vectors. In this way, whilst each output of the block corresponded to one of the inputs, each had access to information from the other bands. SEnSeI-v2 utilizes multihead attention to perform this same information-sharing with the transformer architecture [35], motivated by its remarkable recent performance in a number of applications, perhaps most notably natural language processing.

An attention layer has three learnable weight matrices, known as the query, key, and value matrices (W_Q, W_K, and W_V, respectively). The layer uses correlations between different members of the input sequence to generate useful information (in our case, the sequence is an unordered set of bands). A multiplication between the query and key weight matrices, and the inputs, x_i, are computed to generate the queries and keys ($Q = x_i W_Q$ and key $K = x_i W_K$, respectively). A dot product of each pair of query and key is then taken (leading to N_b^2 dot products for N_b bands). These dot products are softmax (conventionally scaled by the square root of the dimensionality of the vectors d) and multiplied by the value matrix, W_K, to produce the output y_i of the layer corresponding to each input x_i, such that

$$y_i = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) W_V. \tag{2}$$

Extending this concept to multihead attention, this operation is repeated with several weight matrices, and the outputs of each are concatenated, and then transformed by a linear layer into the desired final dimensionality of the output. The transformer block's specific hyperparameters used and its placement within SEnSeI-v2 can be seen in Fig. 2.

4) Band Embedding: The previous version of SEnSeI used "band multiplication" to mix the spectral information of each band with the spatial array of values. This consisted of multiplying each feature vector outputted by SEnSeI's neural network layers with the respective band's pixel values at every point in the image, leading to an X-by-Y-by-N output tensor, where C is the number of embedded channels in SEnSeI's outputted feature vector. At each spatial point, then, was a linearly scaled copy of the feature vector corresponding to that band. These tensors (each containing information corresponding to an input band) were then pooled, to produce a fixed-size output that could be straightforwardly used in a sensor-independent fashion for a downstream task such as cloud masking (by a model expecting a fixed number of input channels).

In SEnSeI-v2, this approach is improved by using learnable embeddings for the band values, rather than a fixed multiplication (see Fig. 2). For each band, b, with values in a spatial array $S^{(b)}$, the corresponding feature vector from SEnSeI's fully connected layers, $\boldsymbol{v}^{(b)}$, is sent to three further sets of fully connected layers, whereby embedding parameters—gains $\alpha^{(b)}$, frequencies $\omega^{(b)}$ and phase offsets $\phi^{(b)}$—are computed. These parameters are used to embed the band's information into the output tensor with a sinusoidal function. Many families of functions could have been used, but sinusoidal functions are a natural choice because of their simplicity and their bounded output range, which ensures the outputs do not diverge with extreme values. The embeddings are computed band-wise, meaning each band will be assigned a different group of

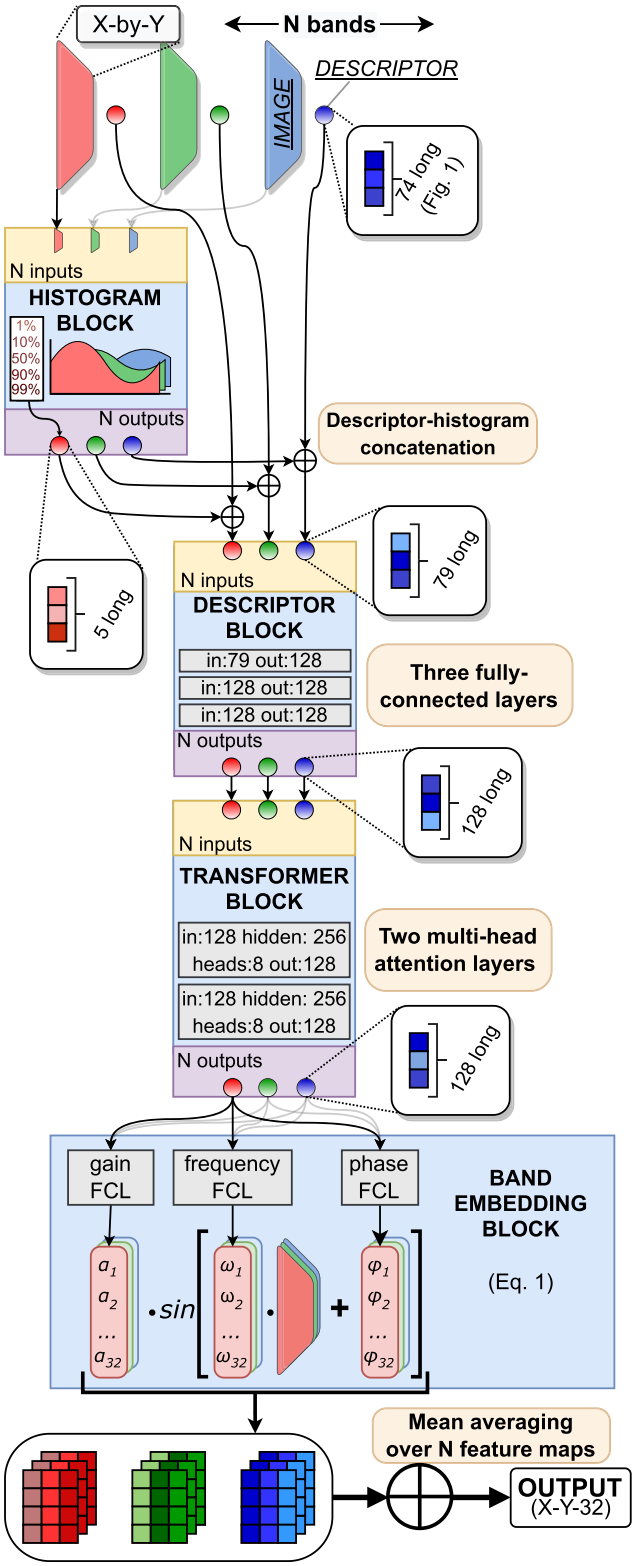

Fig. 2. Flowchart of SEnSeI-v2 model. The red, green, and blue inputs represent a set of bands, with corresponding descriptor vectors. In reality, an arbitrary number of such inputs can be used, but only three are visualized here for simplicity. Gray boxes represent neural networks with trainable parameters. The specific hyperparameters shown here are the ones used during training and testing of the model, however, performance did not seem strongly linked to these hyperparameters and were found through trial-and-error. This diagram can be compared and contrasted with [3, Fig. 2] to see how the design of SEnSeI has changed.

parameters, but those parameters remain constant across the spatial extent of a single image. The values of the embedded output for each band, $E^{(b)}$, at spatial position x and y are

$$E_i^{(b)}(x, y) = \alpha_i^{(b)} \sin\left(\omega_i^{(b)} S_i^{(b)}(x, y) + \phi_i^{(b)}\right) \tag{3}$$

where the ith index corresponds to one of the embedding space's C channels. This process is also demonstrated graphically in Fig. 2. Finally, the embedded representations are pooled across all N_b bands, by mean averaging their values, in order to produce the final fixed size output of SEnSeI-v2

$$E(x, y) = \frac{1}{N_b} \sum_b E^{(b)}(x, y). \tag{4}$$

As with the original version of SEnSeI, a variable number of input bands and their corresponding descriptor vectors have been translated into an output representation with a fixed number of channels. By necessity, pooling is used to create this shared, universal feature space, but this pooling risks a loss of information, especially for SEnSeI-v1, as the signal from each band is overlaid crudely on top of one another in a linear sum. The embedding parameters of SEnSeI-v2 give the model a far greater ability to coordinate how (and where in the space) it will represent each band's values in a manner that is potentially less prone to information loss, in comparison to the linear, parameter-free multiplication used in the original work.

5) Histogram Statistics: For computational simplicity, SEnSeI performs most of its neural calculations on the descriptor vectors, leaving the integration of the spatial image data held in the bands until the very end. This is because, once spatial data is introduced into the model, the subsequent layers must all compute their outputs across image space, multiplying the computational complexity enormously. However, failing to consider any information held within the images is something of a limitation, because SEnSeI-v1 could not use the TOA reflectance values within each band to inform how it encodes those same values. By analogy, this is akin to deciding precisely where every piece of furniture will go in a room that is to be decorated, without having any idea how big each piece is beforehand.

In SEnSeI-v2, histograms of the image are computed, and percentiles across this distribution for each band are fed into the model (see Fig. 2). This allows SEnSeI-v2 to gain some useful information about the kind of image it is dealing with, without having to compute different outputs across each and every pixel before the band embedding is performed. It can then learn to embed the different bands in a way that depends on the image statistics if it is helpful to do so. Through experimentation, it was found that taking five percentile values of the histogram—1%, 10%, 50%, 90%, and 99%—enhanced performance while not adding much to the computation time. These five percentiles are calculated for each band and then concatenated to the end of the descriptor vector before continuing to be ingested by SEnSeI-v2's neural layers. Returning to the interior decor analogy, it is now possible to see the overall size and shape of different pieces

of furniture before the delivery truck arrives and to plan the room's layout accordingly.

B. Ambiguous Loss Functions

Whilst SEnSeI translates the varied inputs from different sensors into a shared representation, there is also a need for a strategy to use the outputs of the DL model (in this case SegFormer [36] or DeepLabv3+ [37]) with labels from a variety of different datasets with different class structures.

One way to understand the difference between partial labeling and other binary labeling strategies (for fuzzy labels, the following distinction is not so informative) is to consider the labels' *precision* as judged against the real-world truth. Unambiguous labels are assumed to be maximally precise. This means (ignoring whatever errors are made in the annotations themselves) that a positive annotation of a given class corresponds to its existence in the real world 100% of the time. By contrast, for partial labels, the precision of the labels can be less than 100%, because a partial label will contain classes marked as possible, that are not true.

By combining sensor independence with partial label learning, a framework is established to build models that have the advantage of generalizing across sensors, whilst eliminating the need to simplify the output class structure (usually this is a simplification to cloudy versus noncloudy pixels). Instead, partial label learning leads in a sense to a model that can predict the union of classes included in the multiple datasets, rather than their intersection. In this work, sensor-independent models are trained which predict seven classes: *land*, *water*, *snow*, *thin cloud*, *thick cloud*, *cloud shadow*, and *no data* (which is included primarily to make inference easier on the edges of large scenes) despite no dataset containing exactly all these classes.

Training a model using partial labels requires us to find a way of backpropagating useful information in the labels without penalizing the model for making strong predictions within the set of possible answers. For example, if a pixel is known to be cloudy, but it is unknown whether it is a thin or thick cloud, then the model should be penalized for predicting other, noncloud classes, but it should not be penalized for strongly predicting either thin or thick cloud, or both of them with equal confidence. Here, two partial label losses which exhibit this behavior are offered, based on the standard cross-entropy and mean squared error losses.

For the set of all N_c classes $c_i \in \mathcal{C}$, let a partial label $\boldsymbol{y}$ be defined as a vector with its ith entry, y_i, corresponding to the possibility of class c_i being present. These are binary values, where 0 means the class is impossible, and 1 means it is possible. Similarly, let the predictor's softmax output, $\boldsymbol{p}$, have entries p_i, which are the confidences associated with each of the N_c classes in $\mathcal{C}$.

Our goal is to define differentiable loss functions to train the multiclass predictor with partial labels. First, we can use $\boldsymbol{p}$ to calculate the predictor's *possibility score*, ϕ, which we can define as the sum of the values of $\boldsymbol{p}$ for which the class is labeled as possible

$$\phi = \sum_i y_i \cdot p_i. \tag{5}$$

This value, ϕ, allows us to calculate an ambiguous analog to the standard cross-entropy loss, used commonly in classification tasks

$$\mathcal{L}_{\text{CE}} = -\log(\phi). \tag{6}$$

Alternatively, if we wish to compute a loss that is analogous to the mean squared error, we can create a *proxy prediction vector*, $\boldsymbol{\pi}$, which replaces the values of $\boldsymbol{p}$ where a class is possible, with an equal fraction of ϕ, such that at the index of each possible class there is ϕ divided by the number of possible classes

$$\pi_i = (1 - y_i) \cdot p_i + \frac{y_i \cdot \phi}{\sum_i y_i}. \tag{7}$$

Notably, ϕ (and by extension $\boldsymbol{\pi}$) have the useful feature that, unlike $\boldsymbol{p}$, they are not dependent on the individual confidence values that the model outputted in $\boldsymbol{p}$ amongst the possible classes, only their summed total, which allows the model to freely distribute confidences amongst the classes which are deemed possible for that sample.

The loss, $\mathcal{L}_{\text{MSE}}$, is then defined as the mean squared error between $\boldsymbol{y}$ (divided by the number of possible classes) and $\boldsymbol{\pi}$

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_c} \sum_i \left(\frac{y_i}{\sum_i y_i} - \pi_i \right)^2. \tag{8}$$

It is also useful to consider the case that the labels are exact and not partial and see what happens to the two loss functions that have been introduced. In the nonpartial, fully supervised case, where class k is the only possible label, then $\phi = p_k$. Therefore, $\boldsymbol{\pi} = \boldsymbol{p}$ in the nonpartial case, and so $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{MSE}}$ equal the standard cross-entropy and mean squared error between $\boldsymbol{y}$ and $\boldsymbol{p}$. In the models trained in Section V, the $\mathcal{L}_{\text{CE}}$ is used, as it was found to perform better, converging to a good solution faster than $\mathcal{L}_{\text{MSE}}$. Nevertheless, the derivation of this loss may be of use to other applications and domains for which cross-entropy is less well-suited.

C. Model Selection

Whilst one could use SEnSeI to directly predict values for a supervised task such as cloud masking, in practice, it is better to instead pass the embedded space outputted by SEnSeI to a DL model, because the computations within SEnSeI do not use the spatial information of the image. In this work, DeepLabv3+ [37] and SegFormer [36] are used. DeepLabv3+ is a convolutional segmentation model. It uses a backbone (in this work, the ResNet34 backbone is used) with atrous convolutions. Atrous convolutional kernels are constructed with gaps and are used to efficiently gather both local and global information from the image. Meanwhile, SegFormer is a transformer-based model, which uses an efficient self-attention mechanism that allows the model to consider complex relationships between different areas of an image. In picking these two quite different models (a convolutional one and a transformer-based one), it is hoped that sensor independence with SEnSeI is shown to not be contingent on pairing with a specific kind of DL model.

IV. Experimental Setup

A. Datasets and Class Structure

In total, eight labeled datasets are used during the training and testing of the models in this work. Four from Sentinel-2, two from Landsat 8, and one each from Landsat 7 and a PerùSat-1. Each one of these has a different labeling structure. In order to use these different datasets to train a single model, a class structure must be defined that can be mapped via partial labeling to each of the preexisting class structures used by the datasets. Table I outlines how these mappings are defined from the model's 7-D classification to the different datasets. This section first describes the format into which all datasets were preprocessed and then details some specifics about each of the datasets used.

All input images were made into 512-by-512 pixel tiles, where the resolution of all bands is made the same. In cases where datasets were smaller than this (e.g., the scenes from CloudSEN12 are 509 pixels across) they are bilinearly resampled to 512. For those larger than 512 pixels, cropped patches were taken with a sliding window. TOA reflectance values are kept as physical, unitless values (typically between 0 and 1, although certain geometrical and illumination conditions can create values higher than this).

The masks are preprocessed in different formats, depending on whether the partial or nonpartial loss is used. In the nonpartial case, masks are simply one-hot encoded arrays with the same spatial dimensions as the corresponding images (512-by-512 pixels). The classes used depend on the labeling strategy of the dataset. Meanwhile, for the experiments using partial label learning, masks are *not* one-hot encoded but can contain values of 1 in multiple classes per pixel. These classes are held constant across every dataset: *land*, *water*, *snow*, *thin cloud*, *thick cloud*, *cloud shadow*, and *no data*.

As well as an image and mask, each preprocessed sample also comes with a metadata file, containing information about the bands within the image, and the classes in the mask. This information is useful in creating the descriptor vectors that are given to SEnSeI and handling datasets with different class structures.

1) CloudSEN12: CloudSEN12 is the largest cloud masking dataset for Sentinel-2, with the most diverse set of annotated images by some margin. Whilst the full dataset contains many partially labeled and unlabelled images, in this work, only the fully labeled portion of the dataset is used, comprising 10 000 image patches of 509-by-509 pixels. Each image patch is labeled with four classes: *clear*, *thick cloud*, *thin cloud*, and *cloud shadow*. The annotations were made using the IRIS annotation tool [38], which allows for semiautomated labeling at a significantly higher speed than when using purely manual tools.

A compelling aspect of this dataset is its inclusion of multimodal input data. For each Sentinel-2 image patch, a coregistered Sentinel-1 GRD product from a similar date is also provided, as well as a MERIT DEM patch. These are used in Section V-C to test how auxiliary multimodal data (as an optional addition to the primary sensor data) can impact cloud masking performance.

Across the dataset, 2000 regions of interest are sampled, each five times, leading to a total of 10 000 images. In the experiments of Section V, training, validation, and testing splits follow exactly those used by Aybar et al. [13], in order to maintain comparability. CloudSEN12 is used as the primary dataset for testing and comparison of models in cloud and shadow masking performance because, unlike some other datasets used, shadows are consistently marked, and thin and thick clouds are distinguished, making it the most complete and suitable of the datasets for testing. Another advantage is that the dataset comes with several models' results precomputed, allowing for comparisons across different cloud and shadow masking models.

2) Sentinel-2 Cloud Mask Catalogue: The Sentinel-2 Cloud Mask Catalogue [39], contains 513 patches, each 1022-by-1022 pixels across, with three labels (*clear*, *cloud*, and *cloud shadow*) marked. However, in some images, where shadow was too difficult to mark, the annotations revert to simply *cloud*, *noncloud*. Similar to CloudSEN12, the Sentinel-2 Cloud Mask Catalogue was annotated semi-manually using IRIS [38].

Alongside these pixel-wise annotations, nonmutually-exclusive patch-wise tags are provided, offering details on properties such as surface type and cloud thickness. In this work, these are used to partially constrain the partial labels of the training set. By way of example, if a patch in the dataset is described as having "forest/jungle," "mountainous," and "open water" attributes, but not "snow/ice," then all clear pixels in the image would be partially labeled as possibly *land* and *water*, but not *snow*. Similarly, if "thin cloud" is marked as present but not "thick cloud," then all cloud pixels are treated as unambiguously *thin cloud*, whereas if both "thin cloud" and "thick cloud" tags were associated with the patch, then all cloudy pixels would be classed as possibly being *thin cloud* or *thick cloud*, in a partial label.

3) KappaSet: The KappaSet cloud masking dataset comprises 9251 patches, from 1038 Sentinel-2 products, of 512-by-512 pixels at 10 m/pixel. The dataset is labeled with classes including *clear*, *cloud shadow*, *thin cloud*, and *thick cloud*. Whilst globally distributed, there is a more dense sampling over Europe than elsewhere, with a roughly equal split between the different seasons.

4) CESBIO Reference Masks: Baetens and Hagolle [40] provide labels for 31 full Sentinel-2 products, taken between 2016 and 2018 (seven other masks are also provided, primarily to test the labeling scheme against the Hollstein [41] dataset, and are usually excluded in model validation [7], [40]). The scenes cover ten specific regions of interest, with between two and four scenes from each location. The classification scheme used separates clouds into *low cloud* and *high cloud* classes, which do not correspond directly to thin and thick clouds. Because of this, these classes are combined during the experiments presented here, producing a single cloud class.

Interestingly, this is the only Sentinel-2 dataset used in this work for which there are comprehensive pixel-wise classifications for *land*, *water*, and *snow* (where the Sentinel-2 Cloud Mask Catalogue only provides scene-wise classification tags). For this reason, this dataset is used in Section V-B to test the

TABLE I
OVERVIEW OF THE EIGHT DATASETS USED IN THIS STUDY. TARGET CLASSES ON THE LEFT REFER TO THE CLASSES OUTPUTTED BY THE MODELS TRAINED IN SECTIONS V-B AND V-C

Satellite	Sentinel-2				Landsat 7	Landsat 8/9			PeruSat-1
Dataset	CloudSEN12	CMC	KappaSet	CESBIO	CCA	SPARCS	CCA	CCA-Ext	CloudPeru2
Used for training?	✓	✓	✓	✗	✓	✓	✓	✗	✓
Used for testing?	✓	✗	✗	✓	✗	✗	✗	✓	✗
SAR & DEM data?	✓	✗	✗	✗	✗	✗	✗	✗	✗
No. of scenes	10,000	513	1038	38	206	80	96	126	153
No. of megapixels	2600	540	2400	130	7600	80	4000	4400	5700
Target Classes: Land	Clear	Land*	Clear	Land	Clear	Land‡	Clear	No Cloud	No Cloud
Target Classes: Water		Water*		Water		Water‡			
Target Classes: Snow		Snow*		Snow		Snow			
Target Classes: Thin Cloud	Thin Cloud	Cloud	Thin Cloud	Cloud†	Thin Cloud	Cloud	Thin Cloud	Thin Cloud	Cloud
Target Classes: Thick Cloud	Thick Cloud		Thick Cloud		Thick Cloud		Thick Cloud	Thick Cloud	
Target Classes: Shadow	Shadow	Shadow*	Shadow	Shadow	Shadow*	Shadow	Shadow*	No Cloud	No Cloud
Target Classes: No data			No data	No data	No data		No data	No data	

* Class labels are incomplete because in certain images of the dataset they were not considered, or not disambiguated from other classes. In these images, partial labels are created between the possible target classes. E.g. when shadows are not labelled, all non-cloud pixels are marked as possibly shadow. Or, when land, water and snow are not separable, all are marked as possible.
† Cloud class of CESBIO dataset is in fact labelled as two separate classes: 'low' and 'high' cloud. Given that these do not map directly onto 'thick' and 'thin' cloud, they are combined and labelled ambiguously as both.
‡ SPARCS defines a class for 'flooded' areas. These pixels have characteristics of both the 'land' and 'water' target classes, and so are given partial labels.

models' abilities in distinguishing between *land*, *water*, and *snow*.

5) SPARCS: SPARCS [42] is a relatively small but high-quality dataset of diverse cloud and cloud shadow masks in Landsat 8. Consisting of 80 1000-by-1000 pixel masks at 30 m/pixel, the dataset has labels for *land*, *water*, *snow*, *flooded*, *cloud*, *cloud shadow*, and *cloud shadow over water*. During training, the *flooded* pixels are treated as having partial labels where both *land* and *water* are possible, and the *cloud shadow over water* pixels are simplified to just *cloud shadow*. This dataset is very useful for training in Section V-B because it offers pixel-wise labels for *land*, *water*, and *snow*.

6) Landsat 8 CCA: The USGS released a dataset of labeled Landsat 8 images, totaling 96 full scenes, referred to here as Landsat 8 Cloud Cover Assessment (CCA) [43]. From each of the eight biomes, 12 scenes are sampled globally, with a range of different cloud cover conditions. Pixels are labeled manually as *clear*, *thin cloud*, *thick cloud*, and *cloud shadow*, however, there are only annotations for shadows in a subset of scenes, as some were too difficult to annotate. In training, the pixels of those scenes without specific cloud shadow labels are treated as partial labels, possibly being *cloud shadow*, *land*, *water*, or *snow*.

7) Landsat 8/9 CCA-Ext: A set of annotated scenes from Landsat 8 and 9 were taken from the various validation datasets [44], [45], [46] released by USGS (separate from the Landsat 8 CCA dataset) in what is referred to in Table I as "CCA-Ext." Whilst it was originally planned to use all scenes from the datasets, some were not possible to retrieve and so were omitted. This dataset is used in Section V-D to test models on Landsat data that has been unused during training by any published algorithm to date.

The annotations were created with a similar style and by the same annotator, as the Landsat 8 CCA dataset. Annotations of shadows cast by clouds, however, are omitted, leaving three annotated classes. In practice, when used in Section V-D, the *thin cloud* and *thick cloud* classes are combined to reduce the problem to binary classification of cloud versus noncloud.

8) Landsat 7 CCA: Similar in class structure and annotation style to the Landsat 8 CCA, this dataset includes 207 scenes (although, as in [3], only 197 could be processed properly). The 197 scenes are sampled from a range of different latitudinal bands, providing a diverse set of scenes with manually derived labels [47].

9) CloudPeru2: Launched in 2016, PerúSat-1 is an RGB-NIR instrument with a resolution of around 2 m/pixel. CloudPeru2 [48] is a labeled dataset of 153 scenes from this satellite, split into 22 000 patches. The dataset is labeled as cloud and noncloud.

B. Model Training

All models developed for this study were implemented and trained using the PyTorch framework. The AdamW optimizer was used with an initial learning rate of 1e−4 (after a warmup schedule beginning at 5e−6), and a weight decay term of 1e−4. After validation loss reached a plateau, the learning rate was then lowered to 2e−5 and then finally again to 5e−6.

All models were trained with a batch size of 8, on a 24 GB Nvidia RTX 3090 GPU. Training took between 10 h for the simplest, smallest model (SegFormer-B0, without SEnSeI, trained on a single dataset) to two and a half days for the most complex (Segformer-B2, with SEnSeI-v2, trained on all available datasets).

Rather than pretraining SEnSeI as was done previously, it can now trained simultaneously during the main supervised learning task, using the same autoencoder architecture for estimating band values from SEnSeI's outputs [3]. The autoencoder is constructed by using another neural network, which takes SEnSeI's output and predicts the original values of each band that was given to SEnSeI, optimized using a mean squared error loss. For each band that was inputted, its descriptor vector is concatenated onto SEnSeI's output at every pixel. Then, the neural network (a set of two fully connected layers with 128 channels each, and a final layer which had one output channel), predicts the value of each pixel for that band. This ensures that the output of SEnSeI carries information about the precise values that it was given, and was found to greatly improve training speed, whilst simplifying the training procedure by completely removing the pretraining step that was implemented previously.

V. Results

In all experiments, similar metrics are used to judge the models' performance. In all the tables of results, P refers to precision, R to recall, F_1 to the harmonic mean of precision and recall, BA to the balanced accuracy (the mean of recalls of the positive and negative classes), and IoU to the intersection-over-union (the ratio of successful positive detections to the combined set of positives in predictions and labels).

A. Model Improvement Experiment

This first experiment compares performance in the non-multimodal, nonambiguous case. The goal of the comparative exercise is to ascertain whether and by how much SEnSeI-v2 outperforms SEnSeI-v1, and which DL model performs best both in isolation and when working with SEnSeI when compared against other published methods. DeepLabv3+ and SegFormer are used as archetypal examples of convolutional networks (DeepLabv3+) and vision transformers (SegFormer), to show SEnSeI's ability to be paired with a diverse range of models.

The other published methods are those provided by Aybar et al. [13] and are not recomputed, with the exception of the UNetMobV2 model, for which the authors' github package [49] was used. Instead of recomputing masks for each method, the masks provided alongside the CloudSEN12 dataset are used. Unlike [13], the metrics reported here are pixel-wise metrics across the entire dataset (rather than medians of the metrics across each image). Both methods of presenting the results have advantages, as a user may indeed be more interested in median performance, whilst in a statistical sense it is difficult to compare precision and recall across images with very different relative distributions of classes. Given the interest here is on general model performance, the standard pixel-wise approach is used.

For each of the four classes labeled in the dataset (*clear*, *thick cloud*, *thin cloud*, and *cloud shadow*), precision P and recall R are calculated, alongside metrics for super-classes. These super-classes (*cloud* versus *noncloud* and *invalid* versus *valid*) are formed by combining the separate classes to find relationships that are of more interest and utility to a user of cloud masking algorithms. They also allow for comparison with methods (such as Fmask and s2cloudless) that do not distinguish between all the original classes.

Table II summarizes the results of the experiment. When one considers the three models used with SEnSeI (DeepLabv3+ and the two SegFormer models) a general trend emerges. Performance of the nonsensor independent models is all very high, with SegFormer-B2 consistently best in global metrics (F_1, BA, and IoU), although DeepLabv3+ performs at almost the same level. Meanwhile, each model receives a hit to performance when SEnSeI-v1 is used to make its sensor independent (as has been previously shown [3]). However, SEnSeI-v2 reverses this loss; sensor independence is achieved with a negligible reduction in performance. Models are created that can mask clouds using any and all combinations of bands from Sentinel-2, whilst maintaining performance when using all the bands, meaning models with added utility—applicable to RGB, RGB-NIR, and other satellite sensors which consist of a subset of Sentinel-2 bands—have been created with essentially no loss in accuracy. A visual comparison between several of the models can be found in Fig. 3.

Looking at all models, including those taken from Aybar et al. [13], it seems that models trained on CloudSEN12 (DeepLabv3+, SegFormer-B0, SegFormer-B2, and UNetMobV2) all consistently outperform those which are not (KappaMask, s2cloudless, and Fmask). Whilst the differences are large enough to assume that there is some real gap in objective performance, it should still be noted that training on different datasets gives a model different biases, leading to masks that are not necessarily incorrect but that can disagree with the test dataset's labels. The implications of these trends are discussed further in Section VI.

B. Partial Labeling Experiment

This experiment introduces the partial labeling framework to the model training, using the ambiguous cross-entropy loss function detailed in Section III-B. The benefits of partial labeling are more qualitative than quantitative, in that it expands the functionality of a model (by creating a richer, more specific output), whilst not necessarily effecting the performance in the simpler, original task. That being said, it is nevertheless important to measure any difference in performance between models with or without the ambiguous loss applied, to verify that there is not a deleterious effect on performance.

Three models were trained using a standard (nonambiguous) cross-entropy loss, with two classes (*cloud* versus *noncloud*) or four classes (*clear*, *thin cloud*, *thick cloud*, and *cloud shadow*). When training with two classes, it is possible to combine training across different datasets and sensors, by reducing all their respective class structures to this more basic one.

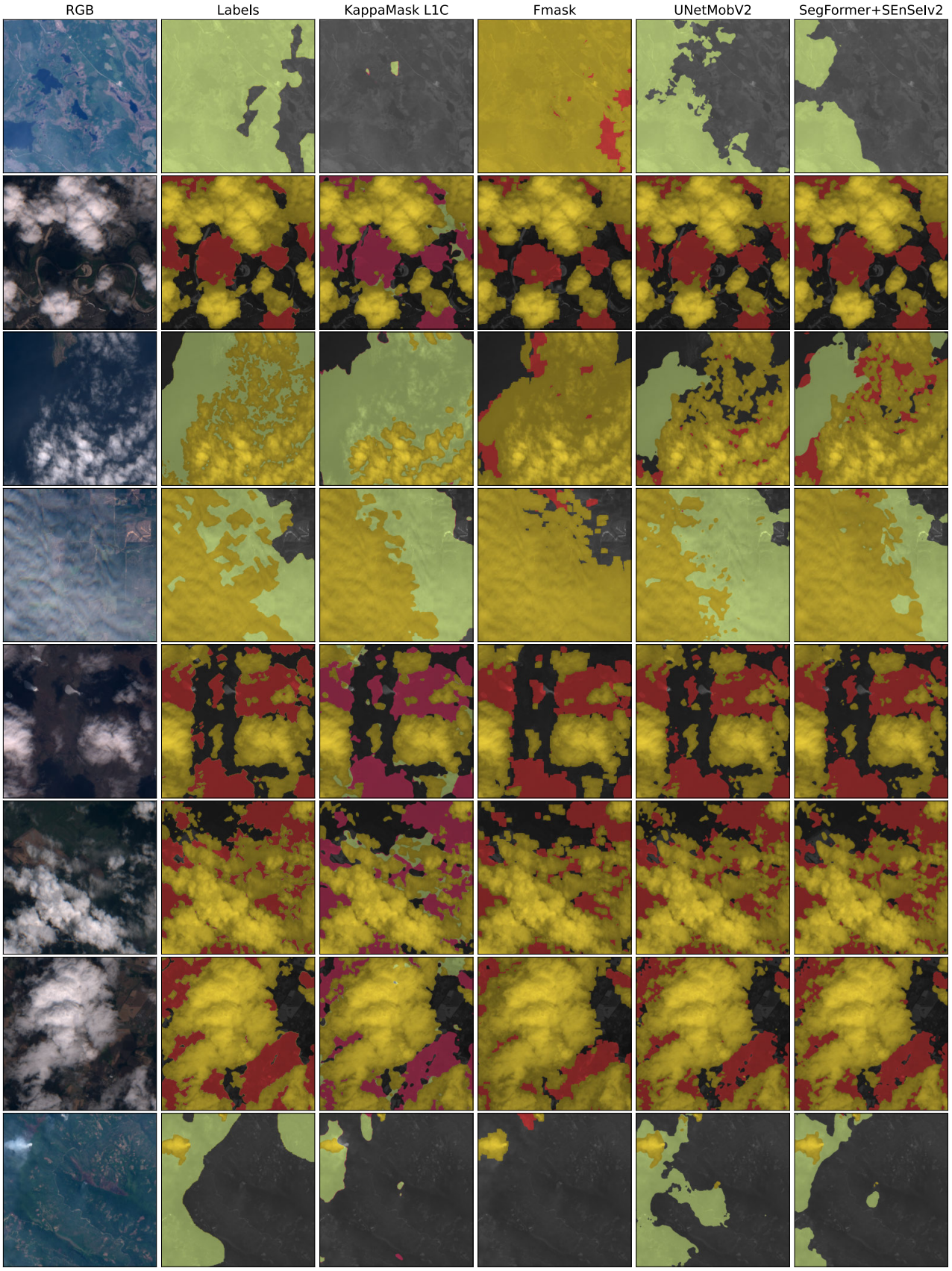

Fig. 3. Visual results across a random sample from the CloudSEN12 test split. Not all models tested are displayed here, for the sake of conciseness. *Thick cloud* is marked in ■, *thin cloud* in ■, and *cloud shadow* in ■, whilst *clear* areas are left transparent. Fmask does not separate thin and thick cloud classes and so is just marked with the color of thick cloud.

TABLE II

RESULTS OF EXPERIMENT ON SENSEI'S MODEL DESIGN COMBINED WITH DIFFERENT DL MODELS (FIRST NINE ROWS), AND OTHER PREVIOUSLY PUBLISHED METHODS (LAST FIVE ROWS). METRICS CALCULATED OVER THE TEST SPLIT OF CLOUDSEN12 DATASET. IN THE FINAL FOUR COLUMNS, "INVALID" PIXELS (CLOUDS AND CLOUD SHADOWS) ARE TREATED AS THE POSITIVE CLASS SO THAT THE DEFINITION OF PRECISION AND RECALL ARE CONSISTENT BETWEEN IT AND CLOUD/NONCLOUD

Model	SEnSeI	Clear		Thick Cloud		Thin Cloud		Shadow		Cloud/non-cloud					Invalid/valid				
		P	R	P	R	P	R	P	R	P	R	F_1	BA	IoU	P	R	F_1	BA	IoU
DeepLabV3+	-	93.06	95.26	**92.31**	90.83	70.14	65.06	80.51	79.90	94.07	95.64	94.85	93.09	90.20	93.75	94.65	94.20	93.93	89.03
DeepLabV3+	v1	92.87	94.11	90.86	91.42	67.82	62.02	78.95	78.42	93.82	94.66	94.24	92.44	90.10	93.75	93.18	93.46	93.24	87.73
DeepLabV3+	v2	93.55	94.77	92.02	91.12	69.02	**68.26**	**82.00**	79.50	94.57	95.02	94.79	93.22	90.10	94.32	94.12	94.22	94.01	89.07
SegFormer-B0	-	92.87	**95.52**	91.84	91.48	71.41	62.91	80.71	78.64	93.92	95.66	94.78	92.97	90.08	93.34	95.13	94.23	93.91	89.08
SegFormer-B0	v1	91.94	94.13	89.26	92.16	71.43	56.85	77.60	75.43	93.56	94.98	94.26	92.36	89.15	92.71	93.53	93.12	92.81	87.12
SegFormer-B0	v2	92.74	94.84	91.39	91.89	71.53	61.86	79.07	78.56	93.91	95.42	94.66	92.85	89.86	93.34	94.38	93.86	93.57	88.43
SegFormer-B2	-	93.26	**95.52**	91.93	91.63	**71.89**	63.84	79.97	79.10	94.22	95.81	**95.01**	**93.29**	**90.49**	93.62	**95.23**	**94.42**	**94.12**	**89.43**
SegFormer-B2	v1	85.53	95.37	91.75	88.57	67.44	45.58	67.79	54.08	90.39	**96.45**	93.32	90.17	87.48	86.26	94.90	90.37	89.32	82.44
SegFormer-B2	v2	93.42	94.94	91.78	91.76	70.53	65.29	81.16	**79.98**	94.31	95.26	94.78	93.12	90.08	94.05	94.46	94.25	94.01	89.13
UNetMobV2 [13]	-	**93.65**	94.08	88.98	**93.30**	68.05	61.67	81.12	74.24	**94.84**	94.12	94.47	93.03	89.54	**94.40**	93.33	93.86	93.69	88.43
KappaMask L1C [50]	-	85.79	85.38	78.76	72.04	34.35	51.32	64.27	48.08	89.89	86.45	88.14	85.39	78.79	84.02	84.46	84.24	84.92	72.77
KappaMask L2A [50]	-	86.14	74.32	70.04	81.54	29.16	48.54	65.18	36.72	91.49	74.95	82.40	81.86	70.07	75.48	86.86	80.77	80.59	67.74
Fmask [51]	-	-	-	-	-	-	-	-	-	89.53	89.77	89.65	86.42	81.24	86.64	84.01	85.30	86.11	74.38
s2cloudless [52]	-	-	-	-	-	-	-	-	-	81.94	80.84	81.36	84.89	68.61	-	-	-	-	

TABLE III

RESULTS OF PARTIAL LABELING EXPERIMENT ON THE TEST SPLIT OF CLOUDSEN12. ALL MODELS USE THE SAME ARCHITECTURE (SENSEI-V2 WITH SEGFORMER-B2), GIVEN IT'S HIGH PERFORMANCE IN SECTION V-A. THE FIRST THREE MODELS ARE TRAINED USING A "REGULAR" CATEGORICAL CROSS-ENTROPY LOSS, EITHER PREDICTING CLOUD VERSUS NONCLOUD (TWO CLASSES) OR ON THE FOUR CLASSES FROM CLOUDSEN12. MEANWHILE, THE OTHER TWO ARE TRAINED WITH AN AMBIGUOUS CROSS-ENTROPY, AS DESCRIBED IN (6), WITH THE SEVEN TARGET CLASSES FROM TABLE I

Loss	Classes	Datasets used in training	Clear		Thick Cloud		Thin Cloud		Shadow		Cloud/non-cloud					Invalid/valid				
			P	R	P	R	P	R	P	R	P	R	F_1	BA	IoU	P	R	F_1	BA	IoU
regular	2	CloudSEN12	-	-	-	-	-	-	-	-	94.23	**95.62**	**94.92**	93.22	**90.33**	-	-	-	-	-
regular	2	ALL	-	-	-	-	-	-	-	-	93.80	95.42	94.61	92.76	89.76	-	-	-	-	-
regular	4	CloudSEN12	93.42	94.94	91.78	**91.75**	**70.53**	65.29	81.16	79.98	94.31	95.26	94.78	93.12	90.08	94.05	94.46	94.25	94.01	89.13
ambiguous	7	CloudSEN12	93.60	**95.06**	91.76	91.70	70.52	65.22	80.97	**80.26**	94.43	95.38	94.91	**93.28**	90.30	94.05	**94.71**	**94.38**	**94.13**	**89.36**
ambiguous	7	ALL	**93.65**	94.62	**92.77**	88.91	61.57	**68.71**	**81.39**	77.78	**94.72**	94.69	94.70	93.20	89.94	**94.15**	94.18	94.17	93.94	88.97

However, for a model with four classes, only datasets containing those four classes can be used, which means only the CloudSEN12 training split can be considered. These models allow us to compare how performance is affected when adding the ambiguous loss, with the two models trained using ambiguous learning. Table III shows the performance of all these models on the CloudSEN12 dataset. Fortunately, all models perform similarly with respect to cloud/noncloud and invalid/valid classification. This result is strong evidence that using the ambiguous loss to introduce more specific classes does not affect the model's ability to classify pixels with the original simpler classes.

Interestingly, the only large difference in performance between the models is seen in the model trained on all datasets with an ambiguous loss, which shows a lower precision and somewhat higher recall than the models trained only on CloudSEN12. As discussed in Section II-A, this may be a sign that the definitions of thin and thick cloud differ somewhat in the other datasets, meaning that the model's understanding of thin cloud drifts further away from CloudSEN12's definition when other datasets are introduced.

Next, the CESBIO dataset is primarily used to explore the ability of the model to learn to separate land, water, and snow classes. The results from this experiment can be found in Table IV. Results for other models on the cloud/noncloud task are taken from [7], which does not report the performance on other classes for any of the models. SEnSeI-v2 with SegFormer is able to learn how to separate *land*, *water*, and *snow* classes when given training datasets that include such distinctions, however the recall of *water* and *snow* is somewhat lacking. Visually, the model seems able to pick up larger bodies of water and snowy regions but misses smaller areas (see Fig. 4).

It is worth considering how challenging the task of separating *land*, *water*, and *snow* is in this context, given the data available to the model during training. Only two datasets—the Sentinel-2 Cloud Mask Catalogue, and the SPARCS

TABLE IV

RESULTS ON THE SENTINEL-2 CESBIO DATASET. NO SCENES FROM THIS DATASET WERE USED IN TRAINING ("ALL" MEANS ALL AVAILABLE DATASETS EXCEPT CESBIO). THE FIRST TWO MODELS WERE TRAINED AND TESTED FOR THIS WORK, AND ARE THE SAME MODEL WEIGHTS AS THOSE OF THE FINAL TWO ROWS OF TABLE III. ALL OTHER MODELS' RESULTS ARE TRANSCRIBED DIRECTLY FROM [7]. THE "OVERALL" METRICS ARE CALCULATED ACROSS THE 5 CLASSES, WHILST "CLOUD/NONCLOUD" COLLAPSES THE *Land*, *Water*, *Snow*, AND *Cloud shadow* CLASSES INTO A SINGLE CLASS. UNSURPRISINGLY, THE MODEL TRAINED WITH ONLY CLOUDSEN12 HAS LARGE CONFUSION BETWEEN THE *Land*, *Water*, AND *Snow* CLASSES, BECAUSE IT IS NEVER GIVEN EXAMPLES THAT DISAMBIGUATE THEM. THE RELATIVELY LIMITED NUMBER OF LABELS THAT DO DISAMBIGUATE *Land*, *Water*, AND *Snow* CLASSES IN THE OTHER DATASETS GREATLY IMPROVE THE MODEL'S PERFORMANCE IN THOSE CLASSES WHEN TRAINED ON "ALL" DATASETS

Model	Datasets used in training	Land		Water		Snow		Cloud		Shadow		Overall			Cloud/non-cloud	
		P	R	P	R	P	R	P	R	P	R	Av. Prec.	BA	OA	BA	OA
SegFormer-B2	CloudSEN12	87.33	25.81	4.08	18.77	0.91	**61.93**	88.03	83.73	50.81	77.14	46.23	53.48	41.12	91.45	93.37
SegFormer-B2	ALL	**93.75**	**95.79**	**97.01**	**52.61**	**95.31**	34.17	89.51	84.39	**54.68**	**79.25**	**86.05**	**69.24**	**91.34**	**92.35**	**93.89**
ATCOR [15]	-	-	-	-	-	-	-	84.9	64.4	-	-	-	-	-	80.4	88.6
CD-FCNN [53]	-	-	-	-	-	-	-	**94.1**	60.3	-	-	-	-	-	79.5	89.5
Fmask [51]	-	-	-	-	-	-	-	90.8	80.4	-	-	-	-	-	88.9	93.3
FORCE [54]	-	-	-	-	-	-	-	79.9	84.7	-	-	-	-	-	88.9	91.1
Idepix [55]	-	-	-	-	-	-	-	86.9	77.5	-	-	-	-	-	86.9	91.7
InterSSIM [56]	-	-	-	-	-	-	-	93.1	77.8	-	-	-	-	-	88.0	93.2
LaSRC [8]	-	-	-	-	-	-	-	57.6	85.6	-	-	-	-	-	82.7	81.2
MAJA [57]	-	-	-	-	-	-	-	72.7	**92.9**	-	-	-	-	-	90.5	89.2
s2cloudless [52]	-	-	-	-	-	-	-	90.2	80.4	-	-	-	-	-	88.8	93.1
Sen2Cor [58]	-	-	-	-	-	-	-	88.7	72.3	-	-	-	-	-	84.7	91.0

dataset—contain any annotations which separate *land*, *water*, and *snow* from each other. Whilst partial labeling allows the model to output several classes where previously it outputted fewer, it remains necessary for there to be enough training samples that disambiguate those classes, for the model to successfully learn to separate them. To this end, the Sentinel-2 Cloud Mask Catalogue is of limited use, because there are only image-wise classifications that never provide the model with sharp boundaries between the three classes within a patch (see Section IV-A). This leaves only the 80 images from the SPARCS dataset—the smallest of the seven datasets used in training, and from a different sensor to the test set—to fully disambiguate these classes. Therefore, model accuracy between these classes is not expected to be high. Rather, it is a success that the model learns something useful from this relatively small number of labels, whilst also retaining the ability to disentangle the more well-labeled class boundaries (e.g., cloud versus noncloud).

C. Multimodality Experiment

The third experiment in this study concerns the effect of multimodal inputs on the performance of the cloud masking algorithm. The experiment was conducted by using a single model (SegFormer-B2 with SEnSeI-v2, trained on all training datasets) and using different band combinations (multispectral, SAR, and DEM) at inference. Whilst this model has the same architecture as some of those used in previous experiments, it has been trained separately, with the multimodal data included, hence the weights are not shared between this model and those in other experiments.

Four different multispectral combinations from the Sentinel-2 bands are used for testing. RGB (bands B02-4), RGB and cirrus (bands B02-4 and B10), NIR and SWIR (B05-12), and finally a combination of all 13 Sentinel-2 bands. The results of these, with and without SAR and DEM inputs, can be seen in Table V. Overall, some interesting patterns emerge when looking at the impact of the multimodal inputs on the performance metrics. For band combinations with fewer bands, a substantial increase in thin cloud recall is found when adding SAR data, which becomes negligible (even slightly negative) when using all Sentinel-2 bands. This amplified the positive effect of Sentinel-1 data when using fewer spectral bands suggesting that the model is able to use the SAR in a complementary way, which subsequently becomes redundant when all of Sentinel-2's bands are used. Meanwhile, the effect of adding the DEM seems to be negligible for most band combinations, with a minor exception being the NIR and SWIR, where a very small increase in performance across most metrics is seen. This band combination contains more low-resolution bands than the others, and perhaps the DEM, which has a resolution of 30 m/pixel, gives the model slightly more information about small-scale features in the image, that aid in its predictions.

Interestingly, whilst SAR data boosts certain performance metrics considerably for band combinations with fewer bands (e.g., RGB), it has a limited effect on the metrics regarding more simplistic classifications (*cloud* versus *noncloud* and *invalid* versus *valid*). This seems to suggest that the information from SAR data is particularly helpful for the classification of a relatively small population of pixels, whilst there is a large majority of pixels for which there is no real impact.

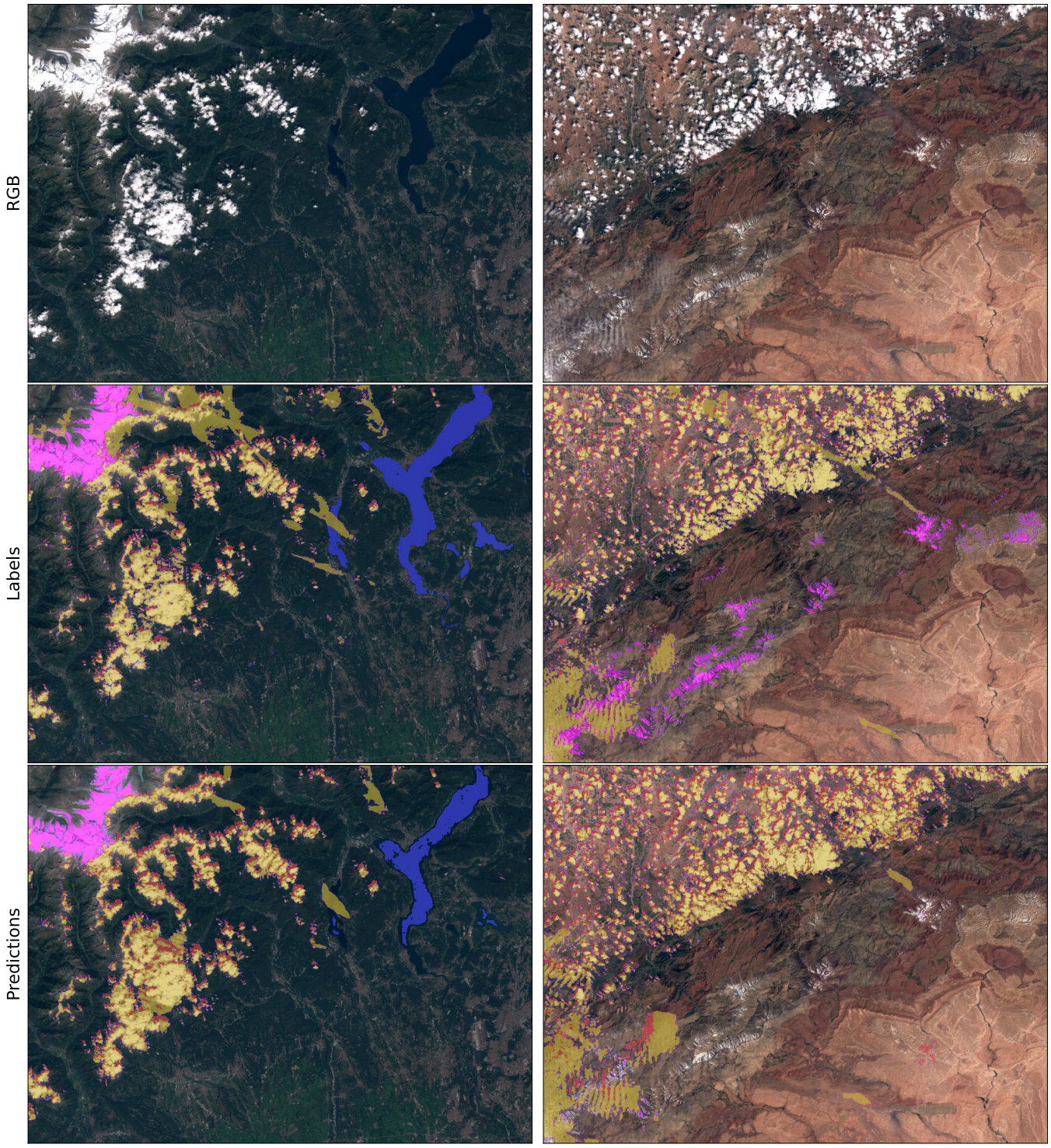

Fig. 4. Examples from the Sentinel-2 CESBIO dataset. Predictions were made using the model in the second row of Table IV, using all the available Sentinel-2 bands. In the masks, *land* is left transparent, whilst *water* is marked in ■, and *snow* in ■. Meanwhile, *cloud* (both thick and thin) is marked in ■, and *cloud shadow* in ■. The model is able to pick out larger areas of *snow* and *water*, but fails to segment smaller regions, misclassifying them as *land*. Meanwhile, *cloud* and *cloud shadow* segmentation is generally good.

D. Sensor Independence

This final experiment is used to verify that SEnSeI-v2 does indeed have sensor-independent characteristics as designed. Three SegFormer models with SEnSeI-v2 were tested, only differing in the training data used. The first is a model specifically for this experiment, trained on only Landsat 8 (SPARCS and CCA datasets). The second and third models are trained on Sentinel-2 data, and all available data, respectively, and are the exact same models used in the final two rows of Table III. For additional comparisons, several well-known openly-available Landsat cloud masking algorithms (Fmask [51], Cloud-Net [59], and ukis-csmask [60]) were also

TABLE V

EXPERIMENTAL RESULTS FOR MULTIMODAL MODEL. ALL ROWS IN THE TABLE COME FROM A SINGLE SENSEI-ENABLED SEGFORMER-B2 MODEL, TRAINED ON THE CLOUDSEN12 DATASET, INCLUDING THE SAR AND DEM BANDS. THE EFFECT OF ADDING SAR AND DEM DATA TO FOUR SPECTRAL BAND COMBINATIONS IS DISPLAYED AS THE RELATIVE DIFFERENCE IN PERFORMANCE VERSUS THE NONMULTIMODAL INPUT (GRAY LINES). LARGER CHANGES ARE MORE STRONGLY COLORED, WITH ANY CHANGE LARGER THAN ±1% HAVING THE STRONGEST COLOR

Spectral Bands	SAR?	DEM?	Clear		Thick Cloud		Thin Cloud		Shadow		Cloud/non-cloud				Invalid/valid			
			P	R	P	R	P	R	P	R	P	R	F_1	BA	P	R	F_1	BA
RGB [B02-4]	✗	✗	91.23	94.71	90.72	90.84	68.24	57.94	78.16	73.21	91.83	94.2	93.0	92.59	93.17	95.23	94.19	92.14
	✓	✗	+0.35	0.0	+0.03	+0.18	+0.32	+2.11	+0.64	-0.21	+0.36	+0.01	+0.19	+0.22	+0.33	-0.14	+0.09	+0.21
	✗	✓	+0.01	+0.01	+0.02	0.0	+0.02	+0.03	-0.01	+0.04	+0.01	0.0	0.0	+0.01	+0.01	+0.01	+0.01	0.0
	✓	✓	+0.37	0.0	+0.05	+0.18	+0.34	+2.16	+0.64	-0.14	+0.38	+0.01	+0.2	+0.23	+0.33	-0.14	+0.1	+0.22
RGB & Cirrus [B02-4, B10]	✗	✗	91.6	94.81	90.88	90.68	68.93	60.26	78.27	73.5	92.19	94.31	93.24	92.86	93.47	95.31	94.38	92.43
	✓	✗	+0.26	0.0	+0.01	+0.16	-0.02	+1.27	+0.66	+0.02	+0.27	+0.01	+0.14	+0.16	+0.22	-0.13	+0.05	+0.12
	✗	✓	0.0	+0.01	+0.01	-0.01	+0.03	0.0	-0.01	+0.05	+0.01	+0.01	+0.01	+0.01	0.0	+0.02	+0.01	+0.01
	✓	✓	+0.27	+0.01	+0.03	+0.16	+0.01	+1.3	+0.65	+0.08	+0.29	+0.02	+0.15	+0.18	+0.22	-0.12	+0.05	+0.13
NIR & SWIR [B05-12]	✗	✗	90.69	94.94	91.2	89.54	64.73	52.88	79.44	78.07	91.26	94.51	92.85	92.38	91.88	95.14	93.48	90.97
	✓	✗	+0.16	0.0	-0.07	+0.06	+0.4	+0.62	+0.01	+0.09	+0.16	-0.01	+0.08	+0.1	+0.12	-0.01	+0.06	+0.1
	✗	✓	+0.03	+0.02	-0.01	+0.03	+0.15	+0.08	-0.01	+0.03	+0.04	+0.02	+0.03	+0.04	+0.03	+0.02	+0.03	+0.04
	✓	✓	+0.19	-0.01	-0.08	+0.1	+0.44	+0.68	0.0	+0.11	+0.19	-0.02	+0.1	+0.11	+0.15	-0.01	+0.07	+0.12
ALL [B01-12]	✗	✗	93.12	95.12	91.75	91.67	71.2	62.98	80.56	80.62	93.51	94.79	94.15	93.86	94.09	95.64	94.86	93.1
	✓	✗	+0.02	+0.03	-0.03	+0.03	+0.13	-0.12	-0.03	-0.04	+0.02	+0.03	+0.02	+0.02	+0.01	+0.01	+0.01	+0.01
	✗	✓	0.0	0.0	+0.01	0.0	+0.01	0.0	-0.02	+0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	✓	✓	+0.02	+0.05	-0.02	+0.03	+0.16	-0.14	-0.05	-0.03	+0.02	+0.04	+0.03	+0.03	0.0	+0.02	+0.01	+0.02

used to process the scenes. A comparison with the model of Mohajerani and Saeedi [61] was not possible because no open-source implementation was available, and because the results in their work seem to include large no data regions at the sides of each scene in their final statistics, making any intercomparison without full recomputation invalid.

In Table VI, results from SEnSeI-v2 with SegFormer-B2 are shown with various band combinations, which align with the bands used by the other models that were tested. Overall, the models show mixed success on the dataset, with a large spread in performance. Focusing first on the model trained with only Sentinel-2 data, the performance is, as one might expect, worse than the models trained with Landsat 8 data, although, it is relatively similar in performance to Cloud-Net and ukis-csmask. The model trained with Landsat 8 data performs well, with a substantial increase across all metrics from the model trained with Sentinel-2, and outperforming the other models by some margin. Finally, and most promisingly, the model trained with all available data outperforms all others. This is evidence that the model has the capacity to learn from multiple sensors in a mutually beneficial, additive way, suggesting that sensor independence does not just lead to the wider usability of models, but also to higher performance.

TABLE VI

RESULTS ON THE LANDSAT 8/9 CCA-EXT DATASET. THE SEGFORMER-B2 MODEL WHICH WAS ONLY TRAINED WITH SENTINEL-2 SHOWED ONLY THE BANDS OF LANDSAT 8/9 WHICH ARE SIMILAR TO THE ONES FOUND IN SENTINEL-2 (SIX BANDS IN TOTAL). THE THREE MODELS WHICH ARE FROM OTHER WORKS WERE RUN USING THEIR DEFAULT PARAMETERS

Model	Datasets used in training	Cloud/non-cloud					
		P	R	F_1	OA	BA	IoU
SegFormer-B2	S2	80.33	79.45	79.89	87.37	85.24	66.51
	L8	**93.53**	86.07	89.65	93.72	91.66	81.24
	ALL	91.81	**90.48**	**91.14**	**94.44**	**93.38**	**83.72**
Fmask [51]	-	81.73	87.20	84.38	89.67	89.01	72.97
Cloud-Net [59]	-	78.76	86.49	82.45	88.24	87.77	70.14
ukis-csmask [60]	-	85.87	77.72	81.59	88.80	85.86	68.91

VI. DISCUSSION

The partial labeling strategy is extremely helpful when combining different cloud masking datasets. However, partial labeling could have general applicability in the domain of supervised learning. Currently, classification datasets are generally created with the assumption of precise labels, such that if an annotator is unsure of the label because multiple interpretations could be considered valid, then they still must choose a single class for that sample. In this work, ambiguous class structures were found in the mapping of differing categorizations, offering a coherent, joint representation of the unaligned labeling schemes. However, in the future, datasets could be created that more naturally reflect ambiguities, at the

point of annotation. For fields such as cloud masking in which considerable uncertainty exists at the semantic boundaries between classes (e.g., what is *thin cloud* versus *thick cloud*), annotating with partial labels may produce a less biased dataset. Clearly, the strict adherence to precise per-pixel labeling leads to datasets that must define a boundary in a fuzzy region of the input space, whilst permitting ambiguity in labels may allow for consistent, comparable definitions across different sensors and datasets. Whilst the models trained in this article did not seem to be negatively impacted by the more complex class systems permitted by the ambiguous loss (see Table III), it is possible that for significantly smaller models, where the capacity to learn the more specific class structures is limited, that partial label learning may hurt performance in the more basic tasks such as binary cloud/noncloud classification.

Cloud masking datasets do not only differ in their output structures and semantic class definitions but also in the sampling distribution of input data. As demonstrated clearly by the violin plots of Jeppesen et al. [62, Fig. 3], the Landsat 8 SPARCS and CCA datasets differ greatly with respect to their reflectance distributions, for example. Differences like these inevitably exist between many of the datasets used here, because of the varied aims and priorities of their creators. The diversity of dataset sample distributions impinges strongly on any claims made about a cloud masking method's generalizability from results on a single dataset. To this end, sensor independence and partial labeling offer a way of smoothing such sampling biases, whereby the combination of multiple datasets entails a more diverse spread of data, both in training and validation.

Moreover, the specific architecture and size of DL cloud masking models seem to have a minimal effect on performance when trained on the same data. For example, in Section V-A, there is less than half a percent difference in all F_1 and BA metrics (see Table II) for DeepLabv3+, SegFormer-B0, SegFormer-B2, with or without SEnSeI-v2, and also the UNetMobV2 of Aybar et al. [13]). This remarkably consistent performance—across convolutional networks and vision transformers—is a strong indication of the singular importance of data, over model design, in the current state-of-the-art cloud masking. Many near-optimal architectures exist, and measured performance is in fact modulated primarily by the size and quality of training data available. CloudSEN12 has increased by an order of magnitude the quantity of Sentinel-2 available for training and testing, however, it is likely that this is still a limiting factor to supervised methods' performance. Rather than creating or tweaking state-of-the-art models, it may be more fruitful for the cloud masking community to follow a data-centric approach, prioritizing a continual increase in the quality, quantity, and diversity of available data, through large, open, collaborative dataset creation.

The recent focus of the Earth observation community on multimodality is well-founded. Richer, more diverse input spaces permit more complex, nonlinear relationships to be found, with the results of Section V-C being but one example. One potential drawback to multimodality, however, is the extra dependencies it attaches to the deployment of a model. If all modes of data used in training are necessary for the running of the model, then gaps in data availability may become an issue (and an ever more serious one as the number of different input modes increases). Multimodality, in this work, is treated as an auxiliary, optional feature of the model, through SEnSeI's ability to ingest and fuse different combinations of data. Therefore, if those auxiliary inputs are not available for a given scene, then cloud masking may still be performed with reasonable performance. The same logic is also true during training, as the optional nature of the auxiliary inputs means that SEnSeI can be trained both on data with and without those auxiliary bands.

Whilst the results of Section V-C show limited value for multimodal inputs when used alongside the full set of Sentinel-2 bands, satellites with fewer spectral channels stand to gain from utilizing multimodal models. Of particular interest to the cloud masking community, the boost to thin cloud recall seen from SAR inputs should motivate future works to include SAR in model inputs, given how challenging thin cloud retrieval can be, as is argued in Section II-A. Tentatively, these results may aid operational satellite providers to weigh up the additional engineering requirements of multimodal models against the potential gain in performance, for a given spectral combination. SEnSeI's design certainly lessens the difficulty of including multimodal inputs in a cloud mask, however, for operational products it may nevertheless be challenging to access other modalities in real-time during cloud masking.

VII. Conclusion

SEnSeI-v2 is considerably more capable than SEnSeI-v1, adding sensor independence to models that maintain their performance in comparison to specialized single-sensor versions (see Section V-A), or even surpassing them (see Section V-D), and extends the descriptor vector scheme to permit SAR and DEM data (see Section V-C). The partial labeling and loss strategy leads to a model that can learn more specific classes (e.g., *land*, *water*, and *snow*) from datasets that have such labels, whilst also being able to continue to use datasets for which those classes are not disentangled. The ambiguous loss, however, still assumes that classes are mutually exclusive. In reality, such a constraint is not always desirable (e.g., the surface under *thin cloud* can still be seen, and therefore perhaps classified). In the future, the ambiguous loss could be extended or replaced to allow for these complex interclass relationships to be expressed by the model, whereby certain classes could remain mutually exclusive (e.g., a pixel cannot be both *land* and *water*) but others are permitted (e.g., *cloud shadow* falling over *snow*).

Assumptions and constraints remain in the descriptor vector's parameter space that future work could relax. For example, temporal information is not provided in the descriptor vectors' parameters, which would allow SEnSeI to ingest multitemporal time series data. By doing so, this could boost the performance of cloud masking directly, with information from other times giving the model a helpful prior with which to

judge the cloudiness of the target scene. In addition, such temporal information could also allow a model to perform cloud removal, as done by Ebel et al. [63] and Stucker et al. [64], but with the ability to do so on a flexible, cross-sensor basis. Temporal information in the descriptors could also permit masking of clouds at night, which is important for sensors that observe the Earth at night (e.g., [65]). Resolution (and other geometric factors) are not encoded by the current descriptor vectors, but could also be included in future work to allow the model to treat different resolutions of data in different ways, given that clouds' spatial nature changes with the resolution one uses.

This article necessarily focused on a single, well-known case study: cloud masking. However, neither the architecture of SEnSeI-v2 nor the definition of the ambiguous loss function, are in any way specific to this task. For many tasks (e.g., land cover and land use, crop type mapping, etc.) there are multiple existing datasets that could be combined using the partial labeling strategy in a similar fashion to here.

A repository with code relating to this project can be found on GitHub [66].

ACKNOWLEDGMENT

The author thanks the many creators and distributors of the datasets and models used in this study, which have made this work possible. He would also like to thank Peter Naylor for his review of the manuscript and James Wheeler for engineering support. Finally, he wishes to express his gratitude to Pat Scaramuzza for his help regarding Landsat 8/9 datasets.

REFERENCES

[1] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019.

[2] Z. Zhu, S. Wang, and C. E. Woodcock, "Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images," *Remote Sens. Environ.*, vol. 159, pp. 269–277, Mar. 2015.

[3] A. Francis, J. Mrziglod, P. Sidiropoulos, and J.-P. Müller, "SEnSeI: A deep learning module for creating sensor independent cloud masks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5406121.

[4] S. Foga et al., "Cloud detection algorithm comparison and validation for operational Landsat data products," *Remote Sens. Environ.*, vol. 194, pp. 379–390, Jun. 2017.

[5] S. Mahajan and B. Fataniya, "Cloud detection methodologies: Variants and development—A review," *Complex Intell. Syst.*, vol. 6, no. 2, pp. 251–261, Jul. 2020.

[6] K. Tarrio et al., "Comparison of cloud detection algorithms for Sentinel-2 imagery," *Sci. Remote Sens.*, vol. 2, Dec. 2020, Art. no. 100010.

[7] S. Skakun et al., "Cloud mask intercomparison eXercise (CMIX): An evaluation of cloud masking algorithms for Landsat 8 and Sentinel-2," *Remote Sens. Environ.*, vol. 274, Jun. 2022, Art. no. 112990.

[8] S. Skakun, E. F. Vermote, J.-C. Roger, C. O. Justice, and J. G. Masek, "Validation of the LaSRC cloud detection algorithm for Landsat 8 images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2439–2446, Jul. 2019.

[9] J. C. Chiu et al., "Cloud optical depth retrievals from the aerosol robotic network (AERONET) cloud mode observations," *J. Geophys. Res., Atmos.*, vol. 115, no. D14, Jul. 2010. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/action/showCitFormats?doi=10.1029%2F2009JD013121

[10] R. Marchand, T. Ackerman, M. Smyth, and W. B. Rossow, "A review of cloud top height and optical depth histograms from MISR, ISCCP, and MODIS," *J. Geophys. Res., Atmos.*, vol. 115, no. D16, Aug. 2010. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/action/showCitFormats?doi=10.1029%2F2009JD013422

[11] W. Zhang, J. Wang, D. Jin, L. Oreopoulos, and Z. Zhang, "A deterministic self-organizing map approach and its application on satellite data based cloud type classification," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2018, pp. 2027–2034.

[12] H. Letu et al., "High-resolution retrieval of cloud microphysical properties and surface solar radiation using Himawari-8/AHI next-generation geostationary satellite," *Remote Sens. Environ.*, vol. 239, Mar. 2020, Art. no. 111583.

[13] C. Aybar et al., "CloudSEN12, a global dataset for semantic understanding of cloud and cloud shadow in Sentinel-2," *Sci. Data*, vol. 9, no. 1, p. 782, Dec. 2022.

[14] W. Wu, J. Luo, X. Hu, H. Yang, and Y. Yang, "A thin-cloud mask method for remote sensing images based on sparse dark pixel region detection," *Remote Sens.*, vol. 10, no. 4, p. 617, Apr. 2018. [Online]. Available: https://www.mdpi.com/2072-4292/10/4/617

[15] R. Richter and D. Schläpfer, *Atmospheric and Topographic Correction (ATCOR Theoretical Background Document)*, document 03-0564, 2019.

[16] S. Qiu, Z. Zhu, and C. E. Woodcock, "Cirrus clouds that adversely affect Landsat 8 images: What are they and how to detect them?" *Remote Sens. Environ.*, vol. 246, Sep. 2020, Art. no. 111884.

[17] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, and J. Huang, "Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks," *Remote Sens. Environ.*, vol. 225, pp. 307–316, May 2019.

[18] Z. Shao, Y. Pan, C. Diao, and J. Cai, "Cloud detection in remote sensing images based on multiscale features-convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4062–4076, Jun. 2019.

[19] Z. Zhang, Z. Xu, C. Liu, Q. Tian, and Y. Wang, "CloudFormer: Supplementary aggregation feature and mask-classification network for cloud detection," *Appl. Sci.*, vol. 12, no. 7, p. 3221, Mar. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/7/3221

[20] N. Nguyen and R. Caruana, "Classification with partial labels," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 551–559.

[21] J. Lv et al., "On the robustness of average losses for partial-label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 2569–2583, May 2023.

[22] M.-L. Zhang, F. Yu, and C.-Z. Tang, "Disambiguation-free partial label learning," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2155–2167, Oct. 2017.

[23] Y. Li, W. Chen, Y. Zhang, C. Tao, R. Xiao, and Y. Tan, "Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning," *Remote Sens. Environ.*, vol. 250, Dec. 2020, Art. no. 112045.

[24] J. Nyborg and I. Assent, "Weakly-supervised cloud detection with fixed-point GANs," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 4191–4198.

[25] J. Li et al., "A hybrid generative adversarial network for weakly-supervised cloud detection in multispectral images," *Remote Sens. Environ.*, vol. 280, Oct. 2022, Art. no. 113197.

[26] Y. Xie et al., "Auto-CM: Unsupervised deep learning for satellite imagery composition and cloud masking using spatio-temporal dynamics," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 12, pp. 14575–14583. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26704

[27] J. Li et al., "Deep learning in multimodal remote sensing data fusion: A comprehensive review," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102926. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222001248

[28] I. Manakos, G. A. Kordelas, and K. Marini, "Fusion of Sentinel-1 data with Sentinel-2 products to overcome non-favourable atmospheric conditions for the delineation of inundation maps," *Eur. J. Remote Sens.*, vol. 53, no. 2, pp. 53–66, Jul. 2020, doi: 10.1080/22797254.2019.1596757.

[29] S. Hafner, A. Nascetti, H. Azizpour, and Y. Ban, "Sentinel-1 and Sentinel-2 data fusion for urban change detection using a dual stream U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[30] A. Orynbaikyzy, U. Gessner, B. Mack, and C. Conrad, "Crop type classification using fusion of Sentinel-1 and Sentinel-2 data: Assessing the impact of feature selection, optical data availability, and parcel sizes on the accuracies," *Remote Sens.*, vol. 12, no. 17, p. 2779, Aug. 2020.

[31] L. Blickensdörfer, M. Schwieder, D. Pflugmacher, C. Nendel, S. Erasmi, and P. Hostert, "Mapping of crop types and crop sequences with combined time series of Sentinel-1, Sentinel-2 and Landsat 8 data for Germany," *Remote Sens. Environ.*, vol. 269, Feb. 2022, Art. no. 112831.

[32] C. Grohnfeldt, M. Schmitt, and X. Zhu, "A conditional generative adversarial network to fuse SAR and multispectral optical data for cloud removal from Sentinel-2 images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 1726–1729.

[33] A. Meraner, P. Ebel, X. X. Zhu, and M. Schmitt, "Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion," *ISPRS J. Photogramm. Remote Sens.*, vol. 166, pp. 333–346, Aug. 2020.

[34] A. Sebastianelli et al., "Sentinel-1 and Sentinel-2 spatio-temporal data fusion for clouds removal," 2021, *arXiv:2106.12226*.

[35] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[36] E. Xie et al., "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 12077–12090.

[37] L.-C. Chen et al., "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 833–851.

[38] J. Mrziglod and A. Francis. (2022). *IRIS*. [Online]. Available: https://github.com/ESA-PhiLab/iris

[39] A. Francis, J. Mrziglod, P. Sidiropoulos, and J.-P. Müller. (2020). *Sentinel-2 Cloud Mask Catalogue*. [Online]. Available: https://zenodo.org/record/4172871

[40] L. Baetens and O. Hagolle, "Sentinel-2 reference cloud masks generated by an active learning method," Zenodo, Oct. 2018, doi: 10.5281/zenodo.1460961.

[41] A. Hollstein, K. Segl, L. Guanter, M. Brell, and M. Enesco, "Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in Sentinel-2 MSI images," *Remote Sens.*, vol. 8, no. 8, p. 666, Aug. 2016.

[42] J. M. Hughes. (2016). *L8 SPARCS Cloud Validation Masks*. [Online]. Available: https://www.usgs.gov/core-science-systems/nli/landsat/spatial-procedures-automated-removal-cloud-and-shadow-sparcs

[43] (2016). *L8 Biome Cloud Validation Masks*. [Online]. Available: https://landsat.usgs.gov/landsat-8-cloud-cover-assessment-validation-data

[44] P. Scaramuzza. (2021). *Landsat 8 Collection 1 Cloud Truth Mask Validation Set*. U.S. Geological Survey. [Online]. Available: https://www.sciencebase.gov/catalog/item/60536863d34e7eb1cb3ebfb1

[45] P. Scaramuzza. (2021). *Landsat 8 Collection 2 Cloud Truth Mask Validation Set*. U.S. Geological Survey. [Online]. Available: https://www.sciencebase.gov/catalog/item/61015b2fd34ef8d7055d6395

[46] P. Scaramuzza. (2021). *Landsat 9 Collection 2 Cloud Truth Mask Validation Set*. U.S. Geological Survey. [Online]. Available: https://www.sciencebase.gov/catalog/item/653fbac9d34ee4b6e05bc5c7

[47] U.S. Geological Survey. (2016). *L7 Irish Cloud Validation Masks*. [Online]. Available: https://landsat.usgs.gov/landsat-7-cloud-cover-assessment-validation-data

[48] G. Morales, A. Ramírez, and J. Telles, "End-to-end cloud segmentation in high-resolution multispectral satellite imagery using deep learning," in *Proc. IEEE 25th Int. Conf. Electron., Electr. Eng. Comput. (INTERCON)*, Aug. 2019, pp. 1–4.

[49] C. Aybar and G. M. Garcia. (2022). *CloudSEN12/models*. [Online]. Available: https://github.com/cloudsen12/models

[50] M. Domnich et al., "KappaMask: AI-based cloudmask processor for Sentinel-2," *Remote Sens.*, vol. 13, no. 20, p. 4100, Oct. 2021.

[51] S. Qiu, Z. Zhu, and B. He, "Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4–8 and Sentinel-2 imagery," *Remote Sens. Environ.*, vol. 231, Sep. 2019, Art. no. 111205. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425719302172

[52] A. Zupanc. (2017). *Improving Cloud Detection With Machine Learning*. Accessed: Aug. 15, 2023. [Online]. Available: https://medium.com/sentinel-hub/improving-cloud-detection-with-machine-learning-c09dc5d7cf13

[53] D. López-Puigdollers, G. Mateo-García, and L. Gómez-Chova, "Benchmarking deep learning models for cloud detection in Landsat-8 and Sentinel-2 images," *Remote Sens.*, vol. 13, no. 5, p. 992, Mar. 2021.

[54] D. Frantz, "FORCE—Landsat + Sentinel-2 analysis ready data and beyond," *Remote Sens.*, vol. 11, no. 9, p. 1124, May 2019.

[55] J. Wevers, D. Müller, J. Scholze, G. Kirches, R. Quast, and C. Brockmann, "IdePix for Sentinel-2 MSI algorithm theoretical basis document," Zenodo, Dec. 2021, doi: 10.5281/zenodo.5788067.

[56] J. Puc. (2017). *Improving Cloud Detection With Machine Learning*. Accessed: Aug. 15, 2023. [Online]. Available: https://medium.com/sentinel-hub/on-cloud-detection-with-multi-temporal-data-f64f9b8d59e5

[57] L. Baetens, C. Desjardins, and O. Hagolle, "Validation of Copernicus Sentinel-2 cloud masks obtained from MAJA, Sen2Cor, and FMask processors using reference cloud masks generated with a supervised active learning procedure," *Remote Sens.*, vol. 11, no. 4, p. 433, Feb. 2019.

[58] M. Main-Knorn, B. Pflug, J. Louis, V. Debaecker, U. Müller-Wilm, and F. Gascon, "Sen2Cor for Sentinel-2," *Proc. SPIE*, vol. 10427, pp. 37–48, Oct. 2017.

[59] S. Mohajerani and P. Saeedi, "Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 1029–1032.

[60] M. Wieland, Y. Li, and S. Martinis, "Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network," *Remote Sens. Environ.*, vol. 230, Sep. 2019, Art. no. 111203. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425719302159

[61] S. Mohajerani and P. Saeedi, "Cloud and cloud shadow segmentation for remote sensing imagery via filtered Jaccard loss function and parametric augmentation," 2020, *arXiv:2001.08768*.

[62] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard, "A cloud detection algorithm for satellite imagery based on deep learning," *Remote Sens. Environ.*, vol. 229, pp. 247–259, May 2019.

[63] P. Ebel, Y. Xu, M. Schmitt, and X. X. Zhu, "SEN12MS-CR-TS: A remote-sensing data set for multimodal multitemporal cloud removal," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5222414, doi: 10.1109/TGRS.2022.3146246.

[64] C. Stucker, V. S. F. Garnot, and K. Schindler, "U-TILISE: A sequence-to-sequence model for cloud removal in optical satellite time series," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5408716.

[65] Q. Wang, C. Zhou, X. Zhuge, C. Liu, F. Weng, and M. Wang, "Retrieval of cloud properties from thermal infrared radiometry using convolutional neural network," *Remote Sens. Environ.*, vol. 278, Sep. 2022, Art. no. 113079.

[66] A. Francis. *SEnSeIv2*. Accessed: Oct. 10, 2023. [Online]. Available: https://github.com/aliFrancis/SEnSeIv2