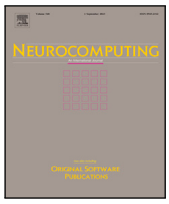
Neurocomputing 638 (2025) 130175

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Missing data as augmentation in the Earth Observation domain: A multi-view learning approach

Francisco Mena [a,b,*], Diego Arenas [b], Andreas Dengel [a,b]

[a] Computer Science, University of Kaiserslautern-Landau, Gottlieb-Daimler-Straße, Kaiserslautern, 67663, Germany
[b] SDS, German Research Center for Artificial Intelligence, Trippstadter Str. 122, Kaiserslautern, 67663, Germany

ARTICLE INFO

Communicated by C. Xu

Dataset link: https://github.com/fmenat/com-views/tree/main/data

Keywords:
Multi-view learning
Earth Observation
Missing data
Data augmentation
Robustness

ABSTRACT

Multi-view learning (MVL) leverages multiple sources or views of data to enhance machine learning model performance and robustness. This approach has been successfully used in the Earth Observation (EO) domain, where views have a heterogeneous nature and can be affected by missing data. Despite the negative effect that missing data has on model predictions, the ML literature has used it as an augmentation technique to improve model generalization, like masking the input data. Inspired by this, we introduce novel methods for EO applications tailored to MVL with missing views. Our methods integrate the combination of a set to simulate all combinations of missing views as different training samples. Instead of replacing missing data with a numerical value, we use dynamic merge functions, like average, and more complex ones like Transformer. This allows the MVL model to entirely ignore the missing views, enhancing its predictive robustness. We experiment on four EO datasets with temporal and static views, including state-of-the-art methods from the EO domain. The results indicate that our methods improve model robustness under conditions of moderate missingness, and improve the predictive performance when all views are present. The proposed methods offer a single adaptive solution to operate effectively with any combination of available views.

1. Introduction

Nowadays, the usage of multiple data sources, sensors, or views has become a standard practice in ML models for various domains [1]. The reason is that by using multiple sources of information, the individual data sources can be complemented to enhance model predictions [2,3]. Earth Observation (EO) is a domain where Multi-View Learning (MVL) has been used to provide comprehensive insights in various applications [4]. In this context, our work refers to a *view* as all features in a specific data source. For example, a view can be an optical or radar Satellite Image Time Series (SITS), weather conditions, topographic information, or metadata. Thus, there can be *temporal views*, with multi-temporal data, and *static views*, with single-date data, consisting of a heterogeneous scenario with different spatio-temporal resolutions. This diversity distinguishes research done in the EO domain from others such as vision and natural language [5]. For example, a multi-view model in EO may consider fusing a static optical image of high spatial resolution and a time series of optical data at low spatial resolution, as shown in [6]. Nevertheless, in scenarios characterized by operational constraints, EO views might not be a persistent source of information as researchers commonly assume, making it infeasible to access them in both training and inference stages.

The EO domain faces data problems due to the finite lifespan of remote sensors, limited spatial coverage, noise, and cloudy conditions [7]. Moreover, unexpected errors can affect the availability of the data, such as the Landsat 7 ETM+ SLC-off problem in 2003 [8], and the failure of the Sentinel-1B satellite in 2021 [9]. This is common in EO as data collection occurs under operational constraints in real-world environments, where different situations and human decisions may affect its consistent and global availability. Hence, this leads to prediction scenarios with missing views, as illustrated in Fig. 1, where the major challenge relies on the heterogeneity of the EO data and its different spectral-spatio-temporal resolutions. Unlike other domains, filling in or reconstructing missing views can be intractable or meaningless. For example, reconstructing a radar image (spectral–spatial data) from a weather time series (temporal data) may not make sense based on the information each one carries. Efremova et al. [10] have shown that reconstructing an optical image from a radar one is more difficult than a radar from an optical view. Therefore, missing views hinders accurate predictions and introduces biases in ML models [2,3,11]. For instance, Mena et al. [12] evidence the negative impact that missing views have on model predictions over different vegetation applications, highlighting the lack of robustness of MVL

* Corresponding author at: Computer Science, University of Kaiserslautern-Landau, Gottlieb-Daimler-Straße, Kaiserslautern, 67663, Germany.
E-mail address: f.menat@rptu.de (F. Mena).

https://doi.org/10.1016/j.neucom.2025.130175
Received 6 December 2024; Received in revised form 25 March 2025; Accepted 29 March 2025
Available online 8 April 2025

models. Even current ML models, like Transformers, are not naturally robust to missing views [13–15]. This leaves open questions such as how to increase the robustness of MVL models to missing views in the EO domain.

Despite the negative impact that missing data has in ML models, various studies actively make use of this during the learning phase. For instance, masking out the input data has been used for self-supervision (i.e. masked reconstruction) in the natural language [16], signal [17], and vision [18] domains. Besides, random layers can be used to increase the model robustness, such as sampling random features among views [11]. Recently, masked reconstruction has been used with EO data, in models like SatMAE [19], SITS-Former [20], Presto [14], and OmniSat [21]. Moreover, simulating missing data can be used as a generalization technique, named Missing data as Augmentation (MAug), such as the dropout layer [22] and augmentation operations in the vision domain (e.g. crop). In the EO domain, MAug techniques are employed in MVL models to learn inter-sensor representations [23], to avoid overfitting to dominant sensors [15], to assess sensors contribution [24], and to increase model robustness to missing data [25,26]. One option is to randomly simulate missing views during training by replacing all features associated with specific views with zero [25]. Another option simulates all combinations of missing views during training, as experimented by Gawlikowski et al. [27] in out-of-distribution detection. However, most of these works focus on masking out data at the input-level or using a fixed-size merge function (like concatenation). This translates into a fake value imputation on the missing features to obtain a fixed-size input [2,27]. Moreover, these works validate on EO datasets with only static views, limiting their applicability to temporal data, which is common in the EO domain.

To overcome these disadvantages, our work introduces feature-level fusion models based on two major components, all Combinations of Missing views (CoM) and a dynamic merge function. The CoM acts as a parameter-free MAug technique, simulating all combinations of missing views in the training set, which relates to literature [25,27]. However, we apply the MAug technique at the feature-level instead of at the input as in Mena et al. [25], and we use dynamic merge functions instead of fixed ones as in Gawlikowski et al. [27]. Unlike models that insert fake data on the missing views, we find that integrating the CoM with a merge function that ignores missing data enhances the predictions and robustness of the MVL model. This dynamic merge function can be a simple average or a more complex function. Inspired by literature that uses ML models as aggregators [28–31], we include alternative merge functions based on a gated modeling, cross-attention layers, and memory-based modeling.

We validate the proposed models on four EO datasets considering both temporal and static views. Besides, we compare them to five state-of-the-art models in the EO domain. To assess the model's robustness to missing data, we simulate missing views during inference and compare the predictions to a full-view data scenario. Unlike models in the literature validated on specific missing views cases [3,32,33], we consider diverse missing cases. Moderate *missingness*, when only *top* views (the ones with the best individual performance) are missing, and extreme *missingness*, when only one *top* view is available. The evidence suggests that our models have better robustness than competing ones in moderate missingness, and even improve the predictive performance in some cases without missing views.

Overall, our main contributions are as follows:

- We introduce a parameter-free MAug technique at the feature level, named Combinations of Missing views (CoM), tailored to diverse missing views scenarios in MVL with EO data.
- Unlike works in the EO domain that use fixed-size merge function and replace missing views by zero [2,23,25–27], we adapt the MAug technique to ignore the features of the missing views using a dynamic fusion.
- Contrary to standard evaluations of model robustness in classification tasks with static views [2,23,26,27], we use various EO datasets considering classification and regression tasks with both, temporal and static views.

Our work, inspired by sensor invariant design [25,34], is focused on allowing a model to yield adaptive and robust predictions from the available views in each case. Our code is available at github.com/fmenat/com-views.

2. Related work

MVL with EO data. Recently, there has been an increment in works using multiple EO data sources to enhance model predictions [35]. The main difference in the MVL models investigated is how the data is fused [4]. Input-level fusion has been the common choice for this, i.e. merging the data before feeding it to a ML model. For instance, Kussul et al. [36] feed a CNN model with just the concatenation of different sensors (multi-spectral and radar images) for land-use classification, while Ghamisi et al. [37] concatenate specialized hand-crafted features from hyper-spectral and LiDAR images. However, learning view-dedicated feature extractors (encoders) has shown to improve predictions, since there is no need for spatio-temporal alignment [2,3,38,39]. For example, Audebert et al. [40] propose a MVL model for multi-spectral and topographic images that fuses across multiple CNN layers (encoders). Later, Zhang et al. [41] show that including a fusion in the decision layer (as a hybrid fusion) improves the results for land-use segmentation. Additionally, Ofori-Ampofo et al. [42] evidence that using specialized encoder architectures for optical and radar Satellite Image Time Series (SITS) of different lengths benefits the feature-level fusion in crop-type classification. Furthermore, there have been efforts to explore geospatial foundational models using multi-view data. For instance, Presto [14] is a Transformer model employing an input-level fusion strategy, while SkySense [31] uses a feature-level fusion based on encoder models that are fine-tuned based on the available views in the downstream tasks.

Missing data in EO applications. Different forms of missing (such as errors and anomalies) are present in EO data [7]. As expected, when the missing information in the input data increases (at spectral, spatial, or temporal dimensions), the predictions of ML models get worse [42–44]. In addition, specific data sources are more relevant than others. For example, the lack of an optical view critically affects models' accuracy [2,12]. Nevertheless, incorporating additional views in the modeling can supplement and increase the model robustness when a view is missing [2,45]. For instance, Ferrari et al. [38] and Sainte Fare Garnot et al. [3] show that when optical images are missing (affected by clouds) in SITS, placing the fusion far away from the input layer increases the model robustness. Furthermore, in Mena et al. [12] three data processing techniques that mitigate the effect of missing views are compared. The first is to impute the missing view with a numerical value. The second replaces the missing view with a similar sample in the training set. The last one ignores the missing views in the aggregation through an adaptive fusion. The latter technique shows greater robustness when views are missing in various EO datasets [12], as well when images are missing in SITS [46]. Furthermore, modifying model components can also increase the robustness to missing views, as when parameters are shared in view-dedicated layers [25,32].

Leverage missing data. Simulate missing data has been progressively used in ML design, from standard augmentation techniques in the vision domain (like crops) to masking out image patches for reconstruction in a self-supervised framework [18]. In the natural language domain, it has been used to learn token embeddings from a reconstruction task [16]. In the signal domain, it has been studied how to best impute data in time series [17,47]. In the EO domain, masking out input data and learning to reconstruct it has been widely used for

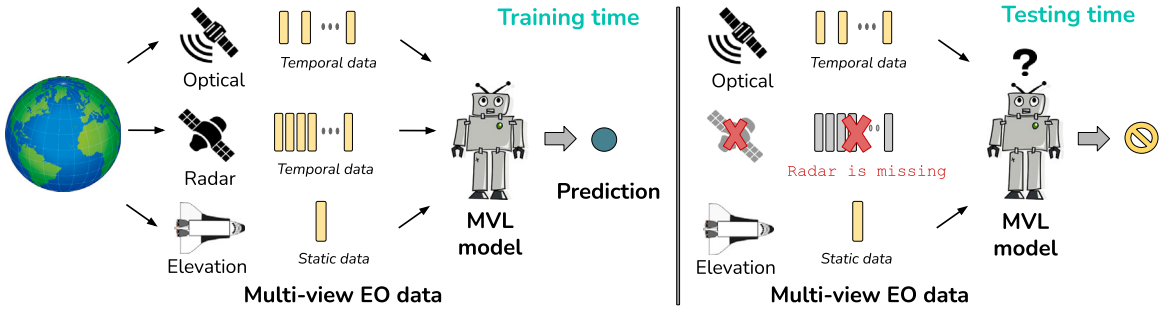


Fig. 1. Illustration of a MVL scenario with three views available during training, while at inference time, one view is missing.

self-supervised learning, such as in SatMAE [19], SITS-Former [20], Presto [14], and OmniSat [21]. Similarly, the dropout operator has not only been considered as a data augmentation [48] but has been used as such. For instance, Fasnacht et al. [43] introduce a spectral dropout to increase the model robustness to missing spectral bands for hyper-spectral image segmentation. In the same task, Haut et al. [49] use spatial dropout (random occlusion) as an effective MAug technique. Furthermore, randomly dropping images in SITS, i.e. Temporal Dropout (TempD), has been employed to improve MVL models prediction [3]. Recently, the MAug has been applied in a sensor-wise manner, i.e. Sensor Dropout (SensD), as a way to learn inter-sensor representations [23], to avoid overfitting to dominant sensors [15], to assess sensors contribution [24], and to increase model robustness to missing data [25,26]. Additionally, Gawlikowski et al. [27] show that the MAug applied to sensors can be used for out-of-distribution detection in two-view image classification tasks.

Most of the works that use MAug techniques with EO data focus on masking out data at the input-level or using a fixed-size merge function. In practice, this means the missing features are imputed with a *fake* value to obtain a fixed-size input. Moreover, the literature positions the input-level fusion as a strategy with limited predictive capacity and robustness [4,12]. To overcome these disadvantages, our work focuses on applying the MAug at the feature-level by ignoring the missing views. We achieve this by using dynamic merge functions, as we show in the following.

3. Multi-view learning with missing data

3.1. Problem setting

Notation. We use the following symbol notation in our work. For variables, we use $a, \boldsymbol{a}, \mathbf{A}$ for single, vector, and matrix cases respectively. In functions, we use $\mathcal{F}(\cdot)$ for single-letter and $\mathrm{function}(\cdot)$ for longer names. In sets, we use $\mathbb{S}$, while in lists, we use $\mathtt{L}$. Each sample is indexed in the superscript as $\boldsymbol{a}^{(i)}$, while the rest of the dimensions are indexed with subscript as $\boldsymbol{a}_v$.

We assume a full-view training set, with expected missing views at inference time, as illustrated in Fig. 1. Consider the multi-view input data for a training sample i as $\mathbb{X}^{(i)} = \{\mathbf{X}_v^{(i)}\}_{v\in\mathbb{V}}$, with $\mathbb{V}$ the set of available views for training. During inference, instead, we observe $\tilde{\mathbb{X}}^{(i)} = \{\mathbf{X}_v^{(i)}\}_{v\in\mathbb{V}^{(i)}}$, with $\mathbb{V}^{(i)} \subseteq \mathbb{V}$, the subset of available views for an inference sample i. Thus, the number of views is $m = |\mathbb{V}|$ and $m^{(i)} = |\mathbb{V}^{(i)}|$, with $m^{(i)} > 0$. We consider a view in the MVL framework as all features from a specific EO data source. Hence, the views $\mathbf{X}_v$ can be temporal (time-series) or static (single-date) data. Here, the objective is to find a MVL model $\mathcal{G}(\cdot)$ that approximates the corresponding target $y^{(i)}$ regardless of the available views for a sample i, i.e. $\hat{y}^{(i)} = \mathcal{G}(\mathbb{X}^{(i)})$. To this end, the learning is through minimizing a loss function of the form $\mathcal{L}(y^{(i)}, \mathcal{G}(\mathbb{X}^{(i)}))$, over a training set of N samples as $\mathbb{D} = \{\mathbb{X}^{(i)}, y^{(i)}\}_{i=1}^{N}$.

3.2. Basis of multi-view learning

Input-level fusion. This strategy directly merges the views at the input-level, usually through the concatenation of the data. As EO views have different spatio-temporal resolutions, an alignment step is required to match all dimensions before the concatenation, expressed by $\mathbf{X}_{\text{fused}}^{(i)} = \mathrm{concat}(\mathrm{align}(\mathbb{X}^{(i)}))$. Then, these merged features are fed to a single ML model: $\hat{y}^{(i)} = \mathcal{G}(\mathbf{X}_{\text{fused}}^{(i)})$. However, in this fusion strategy, there is no clear way to deal with missing views, $\tilde{\mathbb{X}}^{(i)}$. For instance, Hong et al. [2] present a zero-imputation as a data processing of the missing data, i.e. $\mathbb{X}^{(i)} = \tilde{\mathbb{X}}^{(i)} \cup \{\mathbf{0}\}_{v\in\mathbb{V}\setminus\mathbb{V}^{(i)}}$. Subsequent research on MAug has used the zero-imputation in the missing features [25–27]. Nonetheless, zero is an arbitrary value that creates bias depending on data normalization and transformations applied.

Feature-level fusion. To avoid forcing a view-alignment, and have a single model that handles the multi-view information, this fusion strategy extracts high-level feature representation through view-dedicated encoders: $\boldsymbol{z}_v^{(i)} = \mathcal{G}_v^{\text{enc}}(\mathbf{X}_v^{(i)})\ \forall v \in \mathbb{V}$. In addition, a normalization layer (with learnable parameters) is used in each encoder to scale and harmonize the different representations. Then, a merge function combines this multi-view information, obtaining a joint representation, $\boldsymbol{z}_{\text{fused}}^{(i)} = \mathcal{M}(\{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}})$. The merge function $\mathcal{M}(\cdot)$ can take any form, such as concatenation or dynamic functions. Then, a prediction head is used to generate the MVL model prediction as $\hat{y}^{(i)} = \mathcal{G}^{\text{head}}(\boldsymbol{z}_{\text{fused}}^{(i)})$. In the following, we explain how we handle the missing views cases in this fusion strategy.

3.3. Dynamic feature-level fusion

Inspired by permutation [28] and sensor [34] invariant design, we rely on ignoring the features associated with the missing views. For this, our MVL model encodes and merges only the partial available views $\mathbb{V}^{(i)}$ for a sample i:

$$\boldsymbol{z}_v^{(i)} = \mathcal{G}_v^{\text{enc}}(\mathbf{X}_v^{(i)}) \quad \forall v \in \mathbb{V}^{(i)}, \tag{1}$$

$$\boldsymbol{z}_{\text{fused}}^{(i)} = \mathcal{M}\left(\{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}^{(i)}}\right), \tag{2}$$

with $\boldsymbol{z}_v^{(i)} \in \mathbb{R}^d$ the encoded features of the view v, and $\boldsymbol{z}_{\text{fused}}^{(i)} \in \mathbb{R}^{d_{\text{fused}}}$ the fused representation from the available views $\mathbb{V}^{(i)}$ in the sample i. Here, we define d as the dimensionality of the per-view features and d_{fused} the dimensionality of the fused features that depend on the merge function $\mathcal{M}(\cdot)$. Thus, in the case of fixed-size merge functions like concatenation, the fused dimension (d_{fused}) depends on the number of views ($m^{(i)}$), depriving it of an adaptive fusion. Instead, we use merge functions that yield the same fused dimension regardless of the fused views, i.e $d_{\text{fused}} = d$, named dynamic merge function. The reason is that it generates a d-dimensionality vector independently of the views available to fuse. A simple case is a linear combination with the same weight, i.e. the average function, given by

$$\mathcal{M}\left(\{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}^{(i)}}\right) = \frac{1}{m^{(i)}} \sum_{v\in\mathbb{V}^{(i)}} \boldsymbol{z}_v^{(i)}\ . \tag{3}$$

This function has an adaptive scalability of $\mathcal{O}(m^{(i)})$ depending on the number of available views $m^{(i)} = |\mathbb{V}^{(i)}|$ for each sample i, compared to a fixed-cost of $\mathcal{O}(m)$ in concatenation [27]. In the following, we present alternative functions.

Gated fusion. Instead of using the same weight for all views as in the average function, we use a data-driven weighted fusion inspired by Mena et al. [39]. Considering the encoded features of all views for a sample i as $\mathbf{Z}^{(i)} = \text{stack}(\{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}})$, with $\mathbf{Z}^{(i)} \in \mathbb{R}^{m\times d}$, the gated merge function is expressed by

$$\mathcal{M}\left(\mathbf{Z}^{(i)}\right) = \sum_{v\in\mathbb{V}} \text{softmax}\left(\mathbf{A}^{(i)}\right)_v^\top \odot \mathbf{Z}_v^{(i)} , \tag{4}$$

with $\mathbf{A}^{(i)}$ the fusion weights that multiply the features of each view v, $\mathbf{Z}_v^{(i)}$. Then, instead of modeling a single per-view fusion weight for all dimensions (d) as in [39], $\mathbf{A}^{(i)} \in \mathbb{R}^{1\times m}$, we use a per-dimension and per-view weight, i.e. $\mathbf{A}^{(i)} \in \mathbb{R}^{d\times m}$. These fusion weights are calculated with a linear layer over the encoded multi-view features, as

$$\mathbf{A}^{(i)} = \text{flatten}\left(\mathbf{Z}^{(i)}\right)\cdot \mathbf{W}_{\text{gate}} + \boldsymbol{b} , \tag{5}$$

with $\mathbf{W}_{\text{gate}} \in \mathbb{R}^{(m\cdot d)\times(m\cdot d)}$ and $\boldsymbol{b} \in \mathbb{R}^{(m\cdot d)}$ some learnable parameters. In case of missing views, the fusion weights of the unavailable views ($\mathbb{V}\setminus\mathbb{V}^{(i)}$) are set to zero, such as $\text{softmax}(\mathbf{A}^{(i)})_v = \mathbf{0}\ \forall v \in \mathbb{V}\setminus\mathbb{V}^{(i)}$. With this weight modification, the features associated with the missing views are ignored in the merging, Eq. (4). Since the fusion weights require all views to be calculated, Eq. (5), and we only forward over the available views, Eq. (1), we zero-impute the missing encoded features during the implementation, i.e. $\boldsymbol{z}_v^{(i)} = \mathbf{0}\ \forall v \in \mathbb{V}\setminus\mathbb{V}^{(i)}$. This function has a scalability of $\mathcal{O}(m)$, which is not adapted to the partial views available for each sample i.

Cross-attention fusion. Inspired by Transformer layers used to fuse EO data [13,15], we use a learnable parameter $\boldsymbol{f} \in \mathbb{R}^d$, called *fusion token*, to query the multi-view data. Consider the features from the partial available views with the fusion token as $\mathbf{Z}^{(i)} = \text{stack}(\boldsymbol{f}, \{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}^{(i)}})$, with $\mathbf{Z}^{(i)} \in \mathbb{R}^{(1+m^{(i)})\times d}$, the cross-attention merge function is given by

$$\mathcal{M}\left(\mathbf{Z}^{(i)}\right) = \text{softmax}\left(\mathbf{A}^{(i)}\right)_0 \cdot \mathbf{Z}^{(i)}\mathbf{W}_{\text{value}} , \tag{6}$$

with $\mathbf{A}^{(i)} \in \mathbb{R}^{(1+m^{(i)})\times(1+m^{(i)})}$ the cross-view (and token) attention weights that multiply the features projected with learnable parameters $\mathbf{W}_{\text{value}} \in \mathbb{R}^{d\times d'}$ as in [50]. The values $\mathbf{A}_0^{(i)} \in \mathbb{R}^{1\times(1+m^{(i)})}$ are the view-attention weights of the fusion token used to aggregate the multi-view data. These weights are computed by a self-attention mechanism via the dot product in a projected d'-dimensional space, expressed by

$$\mathbf{A}^{(i)} = \mathbf{Z}^{(i)}\mathbf{W}_{\text{query}} \cdot \mathbf{Z}^{(i)}\mathbf{W}_{\text{key}} + \mathbf{B} , \tag{7}$$

with $\mathbf{W}_{\text{query}} \in \mathbb{R}^{d\times d'}$, $\mathbf{W}_{\text{key}} \in \mathbb{R}^{d\times d'}$, and $\mathbf{B} \in \mathbb{R}^{d'}$ some learnable parameters. As the matrix computation in Eq. (7) depends exclusively on the available views $\mathbb{V}^{(i)}$, the model naturally avoids attending to the missing views when merging, Eq. (6), i.e. $\mathbf{A}_v^{(i)} = \mathbf{0}\ \ \forall v \in \mathbb{V}\setminus\mathbb{V}^{(i)}$. This function has an adaptive quadratic scalability of $\mathcal{O}(m^{(i)}\cdot m^{(i)})$ depending on the number of available views for each sample i. Furthermore, we use a multi-head mechanism and stacked layers to increase the learning of cross-view features [50]. In contrast to previous works [15,28], we include a view-specific positional encoding before the cross-attention is applied.

Memory fusion. Inspired by Recurrent Neural Networks (RNNs) used to fuse multi-view EO data [30], we employ a memory-based fusion. The memory is updated one view at a time from the encoded features per view v, expressed by

$$\boldsymbol{h}_v^{(i)} = \mathcal{R}\left(\boldsymbol{z}_v^{(i)}, \boldsymbol{h}_{v-1}^{(i)}\right) \quad \boldsymbol{h}_0 = \mathbf{0}, \tag{8}$$

with $\mathcal{R}(\cdot)$ a RNN model receiving the previous memory $\boldsymbol{h}_{v-1}^{(i)} \in \mathbb{R}^d$, and the current view representation $\boldsymbol{z}_v^{(i)}$, with $v \in \{1,\dots,m^{(i)}\}$. Then, the fused vector corresponds to the memory (or hidden state in RNNs) at the last step, i.e. $\mathcal{M}(\{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}^{(i)}}) = \boldsymbol{h}_{m^{(i)}}^{(i)}$. This means that the fused data is the memory after being recursively updated with all views. As the recursive operation, Eq. (8), is invariant to how many views are given as input, the fusion naturally ignores the missing view cases. Since RNNs are order-dependent, a random permutation can be used to avoid bias towards the order of the views. However, our MAug technique produces the same effect, as shown in the Appendix. This sequential function has an adaptive scalability of $\mathcal{O}(m^{(i)})$ depending on the number of available views for each sample i. Similar to the cross-attention fusion, we stack multiple layers in $\mathcal{R}(\cdot)$ to increase the learning of cross-view features.

Algorithm 1 CoM technique at feature-level

Input: $\mathbb{D} : \{\mathbb{X}^{(i)}, y^{(i)}\}_{i=1}^N$ - multi-view dataset
Input: $\mathcal{G}(\cdot)$ - initialized MVL model
Input: $\mathbb{T}$ - set of all missing views cases
Output: $\mathcal{G}(\cdot)$ - Trained MVL model

for $(\mathbb{X}^{(i)}, y^{(i)}) \in \mathbb{D}$ **do**
  Obtain $\boldsymbol{z}_v^{(i)}$ by forwarding over all view-encoders as Eq. (1)
  Initialize $\Upsilon^{(i)}$ as an empty list
  for $\mathbb{V}^{(j)} \in \mathbb{T}$ **do**
    Obtain $\boldsymbol{z}_{\text{fused}}^{(i)}$ from the available views $\mathbb{V}^{(t)}$ with the dynamic function as Eq. (2)
    Obtain $\hat{y}_t^{(i)}$ by applying $\mathcal{G}^{\text{head}}(\boldsymbol{z}_{\text{fused}}^{(i)})$
    Update $\Upsilon^{(i)}$ by attaching the prediction $\hat{y}_t^{(i)}$
  end for
  Calculate the loss function based on $y^{(i)}$ and $\hat{y}_t^{(i)} \in \Upsilon^{(i)}$ as Eq. (9)
  Update $\mathcal{G}(\cdot)$ by gradient descent
end for

3.4. All combinations of missing views

As previous works have shown, randomly dropping views during training increases the model robustness to missing views [15,26]. However, it can negatively affect model accuracy in the full-view data scenario [25]. Thus, we consider augmenting the training samples by modeling all combinations of missing views at feature-level. Then, assuming a full-view training set, the augmented features extracted from the ith sample are $\{\{\boldsymbol{z}_v^{(i)}\}_{v\in\mathbb{V}^{(t)}}\}_{\mathbb{V}^{(t)}\in\mathbb{T}}$, with $\mathbb{T} = \{\mathbb{V}^{(t)} : \mathbb{V}^{(t)} \subseteq \mathbb{V}, \mathbb{V}^{(t)} \neq \emptyset\}$ the set containing all sub-set combinations of the available views for training $\mathbb{V}$. Here, the number of possible view combinations is the same as the power set of $\mathbb{V}$ minus the no-view case, i.e. $\left|\mathbb{T}\right| = 2^m - 1$. For instance, when $\mathbb{V} = \{\text{optical}, \text{radar}, \text{elevation}\}$, the augmented list is $\mathbb{T}$ ={{optical, radar, elevation}, {radar, elevation}, {optical, elevation}, {optical, radar}, {elevation}, {optical}, {radar}}. This parameter-free MAug technique is named **Combinations of Missing views (CoM)**. The forward pass of our model during training, combining the CoM technique with the dynamic merge function, is illustrated in Fig. 2.

We consider the same contribution between the predictions with missing views and full-view data. This means that all augmented samples coming from $\mathbb{T}$ have the same weight in the loss function during training, expressed by

$$\mathcal{L}_{\text{total}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\left|\mathbb{T}\right|}\sum_{\mathbb{V}^{(t)}\in\mathbb{T}} \mathcal{L}(y^{(i)}, \hat{y}_t^{(i)}) , \tag{9}$$

$$\text{with}\ \ \hat{y}_t^{(i)} = \mathcal{G}^{\text{head}}\left(\mathcal{M}\left(\ \{\mathcal{G}_v^{\text{enc}}\left(\mathbf{X}_v^{(i)}\right)\}_{v\in\mathbb{V}^{(t)}}\right)\right) . \tag{10}$$

For the loss function $\mathcal{L}(\cdot,\cdot)$ in Eq. (9) we use a cross entropy in classification tasks, i.e. $\mathcal{L}(p,q) = -\sum_k p_k \log q_k$, and squared error in regression tasks, i.e. $\mathcal{L}(p,q) = (p-q)^2$. The training of our final model is illustrated in Algorithm 1.

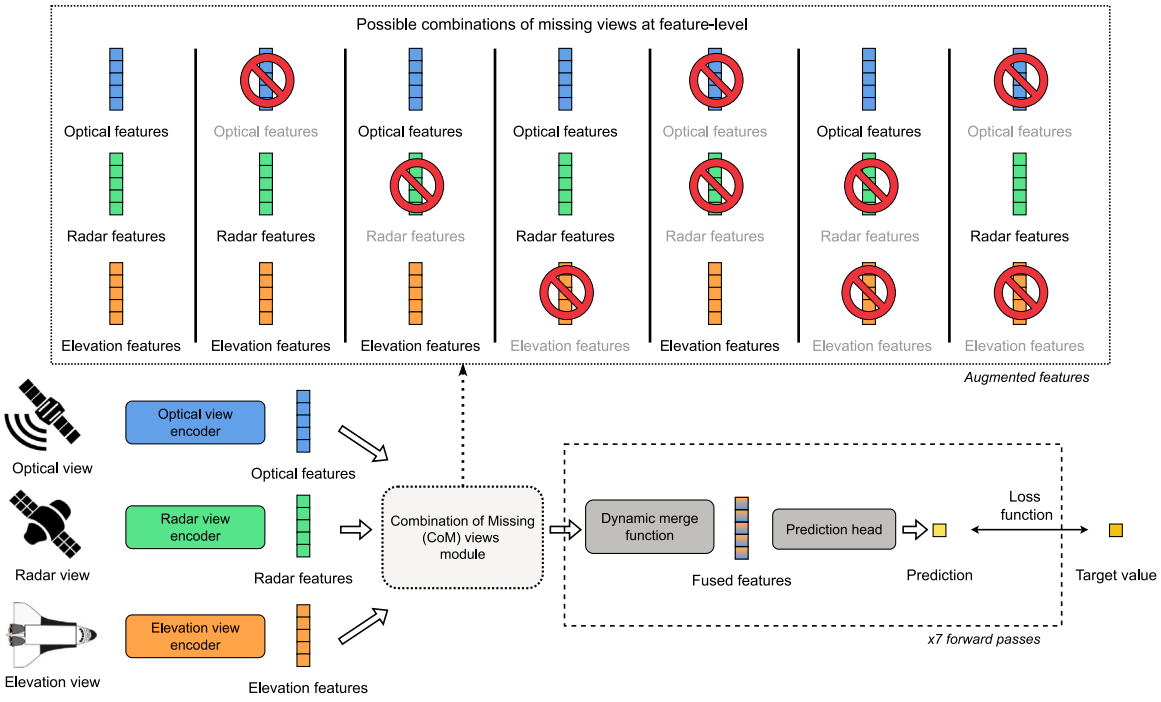


Fig. 2. Illustration of the forward pass during training of a feature-level fusion model using the CoM technique and dynamic merge function. The example considers $m = 3$ views (optical, radar, and elevation), where $|\mathbb{T}| = 7$ augmented samples are simulated with different missing view patterns. This means that the merge function and prediction head are applied $|\mathbb{T}| = 7$ times on each original sample.

Computational cost. To reduce the computational operations of the multiple predictions, we forward over the encoders (biggest computation bottleneck) only once, while the fusion and prediction are done $|\mathbb{T}|$ times, see Algorithm 1 for details. This allows the increase in training time to be less noteworthy, as shown in Section 4.6. Then, considering the forward pass for a sample i during training, the computational cost is $t = \sum_v \gamma(\mathcal{G}_v^{\text{enc}}) + (2^m - 1) \cdot (\gamma(\mathcal{M}) + \gamma(\mathcal{G}^{\text{head}}))$, with $\gamma(\cdot)$ the time cost for each function. If we simplify by considering that all encoders have the same cost, i.e. $\sum_v \gamma(\mathcal{G}_v^{\text{enc}}) = m \cdot \gamma(\mathcal{G}_v^{\text{enc}})$, the scalability regarding the number of views m is $\mathcal{O}(m \cdot 2^m)$ when using average, gated, and memory fusion, or $\mathcal{O}(m^2 \cdot 2^m)$ when using cross-attention fusion. When compared to models using a single augmentation or without MAug, where the complexity scales as $\mathcal{O}(m)$, we have an additional term based on all possible view combinations, 2^m. However, our model increases and diversifies the patterns of missing data, translating into better generalizability and robustness, as shown in Section 4.3.

Inference. The CoM technique is only used during training. Thus, at inference, the model employs a standard feature-level fusion with a dynamic merging, ignoring missing views. The computational scalability during inference is adapted to the available views for each inference sample i, $\mathcal{O}(m^{(i)})$, as discussed in Section 4.6.

4. Experiments

4.1. Datasets

We use the following pixel-wise EO datasets with static and temporal views. More details on the feature description can be found in Appendix A.1.

CropH-b. We use the CropHarvest dataset for crop recognition with four views [33]. This involves a binary task in which the presence of any crop growing at a given location is predicted. It has 69,800 samples around the globe between 2016 and 2021. Each sample has three temporal views: multi-spectral optical SITS (11 bands), radar SITS (2 polarization bands), and weather data (2 bands). These series have one value per month over 1 year. Besides, the samples have one static view, the topographic information (2 bands). All features have a pixel resolution of 10 m. Furthermore, we use a multi-class version of this dataset, named CropH-m. This is a subset of 29,642 samples with 10 crop types to distinguish, and the same input views in CropH-b.

LFMC. We use a dataset for moisture content estimation with six views [51]. This considers a regression task in which the vegetation water (moisture) per dry biomass (in percentage) in a given location is predicted. There are 2578 samples in the western US between 2015 and 2019. Each sample has two temporal views: multi-spectral optical (8 bands) and radar (3 bands) SITS. These time series have one value per month over 4 months. In addition, the samples have four static views: topographic information (2 bands), soil properties (3 bands), canopy height (ordinal feature), and land-cover (12 classes). All features were interpolated to a pixel resolution of 250 m.

PM25. We use a dataset for PM2.5 estimation with three views [52]. This involves a regression task in which the concentration of PM2.5 in the air (in ug/m^3) in a particular city is predicted. The dataset has 167,309 samples in five Chinese cities between 2010 and 2015. Each sample has three temporal views: atmospheric conditions (3 bands), atmospheric dynamics (4 bands), and precipitation (2 bands). The data are at hourly resolution, of which we keep a 3-day window for the estimation, i.e. signals of 72 time-steps are used as input.

4.2. Setup and competing models

We named our models using the CoM technique at feature-level as **FCoM-av** (using average), **FCoM-ga** (using gated fusion), **FCoM-cr** (using cross-attention fusion), and **FCoM-me** (using memory fusion). In the FCoM-cr model, the dynamic merge function consists of one attention layer with eight heads and 40% of dropout, while for the FCoM-me, it consists of two bidirectional layers with LSTM units and 40% of dropout. Variations in the selection of these hyper-parameters are shown in the Appendix A.2.

Table 1
Predictive performance in the classification tasks (F1 score) for different cases (moderate and extreme) of missing views at inference time. The **best** and second best values are highlighted. In parentheses is the number of available views.

Model	CropHarvest binary (CropH-b)					CropHarvest multi (CropH-m)				
	(4/4) No	(3/4) Only missing		(1/4) Only available		(4/4) No	(3/4) Only missing		(1/4) Only available	
	Missing	Radar	Optical	Optical	Radar	Missing	Radar	Optical	Optical	Radar
ITempD-co	$0.817_{\pm 0.008}$	$0.798_{\pm 0.009}$	$0.717_{\pm 0.013}$	$0.668_{\pm 0.015}$	$0.506_{\pm 0.075}$	$0.635_{\pm 0.017}$	$0.552_{\pm 0.019}$	$0.336_{\pm 0.021}$	$0.364_{\pm 0.029}$	$0.154_{\pm 0.017}$
ISensD-co	$0.787_{\pm 0.015}$	$0.780_{\pm 0.015}$	$0.747_{\pm 0.022}$	$0.765_{\pm 0.02}$	$0.572_{\pm 0.092}$	$0.608_{\pm 0.013}$	$0.587_{\pm 0.016}$	$0.437_{\pm 0.038}$	$0.587_{\pm 0.017}$	$0.207_{\pm 0.052}$
FSensD-cr	$0.776_{\pm 0.045}$	$0.748_{\pm 0.091}$	$0.707_{\pm 0.106}$	$0.569_{\pm 0.21}$	$0.527_{\pm 0.169}$	$0.559_{\pm 0.126}$	$0.451_{\pm 0.232}$	$0.302_{\pm 0.173}$	$0.367_{\pm 0.259}$	$0.196_{\pm 0.169}$
FCoMl-co	$0.832_{\pm 0.007}$	$0.827_{\pm 0.006}$	$0.804_{\pm 0.006}$	$0.787_{\pm 0.010}$	$0.686_{\pm 0.010}$	$0.650_{\pm 0.020}$	$0.614_{\pm 0.019}$	$0.543_{\pm 0.014}$	$0.601_{\pm 0.019}$	$0.388_{\pm 0.015}$
FEmbr-sa	$0.821_{\pm 0.007}$	$0.817_{\pm 0.007}$	$0.786_{\pm 0.007}$	$0.764_{\pm 0.008}$	$0.676_{\pm 0.012}$	$0.633_{\pm 0.015}$	$0.598_{\pm 0.022}$	$0.478_{\pm 0.015}$	$0.543_{\pm 0.024}$	$0.312_{\pm 0.012}$
ESensI-av	$0.802_{\pm 0.011}$	$0.806_{\pm 0.010}$	$0.767_{\pm 0.012}$	$0.791_{\pm 0.013}$	$\mathbf{0.701}_{\pm 0.012}$	$0.605_{\pm 0.019}$	$0.577_{\pm 0.021}$	$0.478_{\pm 0.015}$	$0.635_{\pm 0.023}$	$\mathbf{0.444}_{\pm 0.022}$
FCoM-av	$0.837_{\pm 0.005}$	$\mathbf{0.834}_{\pm 0.005}$	$0.804_{\pm 0.007}$	$\mathbf{0.809}_{\pm 0.006}$	$0.615_{\pm 0.040}$	$\mathbf{0.686}_{\pm 0.018}$	$\mathbf{0.648}_{\pm 0.020}$	$0.549_{\pm 0.011}$	$\mathbf{0.649}_{\pm 0.018}$	$0.428_{\pm 0.017}$
FCoM-ga	$\mathbf{0.839}_{\pm 0.005}$	$\mathbf{0.834}_{\pm 0.006}$	$\mathbf{0.810}_{\pm 0.008}$	$0.805_{\pm 0.006}$	$0.681_{\pm 0.018}$	$0.678_{\pm 0.016}$	$0.642_{\pm 0.015}$	$\mathbf{0.566}_{\pm 0.015}$	$0.595_{\pm 0.139}$	$0.399_{\pm 0.093}$
FCoM-cr	$0.826_{\pm 0.007}$	$0.823_{\pm 0.007}$	$0.799_{\pm 0.007}$	$0.801_{\pm 0.008}$	$0.693_{\pm 0.008}$	$0.660_{\pm 0.019}$	$0.637_{\pm 0.019}$	$0.549_{\pm 0.013}$	$0.632_{\pm 0.017}$	$0.439_{\pm 0.015}$
FCoM-me	$0.836_{\pm 0.006}$	$0.818_{\pm 0.007}$	$0.800_{\pm 0.007}$	$0.800_{\pm 0.008}$	$0.681_{\pm 0.014}$	$0.670_{\pm 0.015}$	$0.633_{\pm 0.016}$	$0.535_{\pm 0.011}$	$0.629_{\pm 0.015}$	$0.422_{\pm 0.013}$

For comparison, we consider the following supervised models in the EO domain. Three models employing MAug techniques with zero-imputation: **ITempD-co** [3], using the TempD technique, and **ISensD-co** [25], using the SensD technique, both at the input-level. Besides, **FCoMl-co** [27], a feature-level fusion model using the CoM technique at the feature-level with concatenation and a weighted loss that we extend to multiple views. In addition, we include three models that ignore the missing views: **FSensD-cr**, adapted from images [15] to pixel-wise time series, using cross-attention fusion at feature-level with the SensD technique, **FEmbr-sa** [11], a feature-level fusion model that randomly samples features from different views in the fused representation, and **ESensI-av** [25], a view-invariant model using ensemble aggregation (averaging) without MAug. Since self-supervised models use a bunch of data outside the training set and more input views, we do not consider them as the comparison would not be fair.

Implementation. We apply a z-score normalization to the input data. The categorical and ordinal views (like land-cover and canopy height) are one-hot-vector encoded. We use the best encoder architectures for the selected datasets [12,53]. This corresponds to a 1D convolutional network encoder for temporal views, and a Multi Layer Perceptron (MLP) encoder for static views. We use two layers with 128 units on all encoder architectures, with 20% of dropout and a final layer normalization. After fusion, we use an MLP with one layer as the prediction head. For optimization, we use the Adam optimizer [54] with a batch-size of 128 and early stopping with a patience of five. The stopping criterion is applied over the full-view prediction. The loss function is cross-entropy in classification and mean squared error in regression tasks. We use a weight in the loss function (inverse to the number of samples in each class) to balance the class distribution in classification. For competing models, we use 30% of dropout in ITempD-co and the no ratio version in the ISensD-co model [25].

Evaluation. We use 10-fold cross-validation repeated three times to reduce results variability. We simulate missing views during inference, as illustrated in Fig. 1. We experiment with different degrees of *missingness*: (i) moderate missingness, when only one view is missing, (ii) extreme missingness, when all views are missing except one. We include the results with no missing views (full-view scenario) for reference. For assessing the predictive performance, we use the macro F1 (F1) in classification and the Coefficient of Determination (R^2) in regression tasks.

Missing views analysis. To standardize the analysis of which views are missing in each dataset, we decide to select a few views for this. The selected views to be missing or available are the most effective for the task individually. For this, we train a model on each view to predict the task and then select two views by which the best individual results are obtained. We refer to these as the *top* views. For CropH-b, CropH-m, and LFMC these are optical and radar views, as observed in the literature [2,12,32], while for PM25 these are dynamic and condition views. The results of all the auxiliary views are in Table A.11 in Appendix A.3.

4.3. Results with missing views

In Table 1 we display the F1 score of the models in different missing view cases during inference for the classification tasks. We note that the best results are obtained by our FCoM-av and FCoM-ga models when there are no missing views or moderate missingness. However, in extreme missingness, our models become comparable to competing models. In these extreme cases, the ESensI-av model handles missing views effectively, competing with our models and outperforming them only when the radar view is available. On the other hand, we notice that the models based on the SensD and TempD are highly affected by missing views, i.e. they are less robust to missing data.

For the regression tasks, we display the R^2 for the compared models in Table 2. Here, the impact of missing views is more severe than in classification, with most models reaching negative R^2 values in extreme missingness. This difficulty in regression is expected as models predict a continuous value that disperses in different magnitudes, while in classification the change is only binary, see Section 4.5 for a visual display. Nonetheless, some of our models are robust enough to obtain the best results in extreme cases, such as the FCoM-me model. Moreover, the FCoM-av and FCoM-ga models effectively handle the moderate missingness in both datasets. Although the ITempD-co has the best results in the full-view scenario of PM25 data, it has poor robustness, strongly decreasing performance when views are missing. This outcome is inverse for FCoM-ga, FCoM-me, and FCoMl-co, suggesting that these models learned better the missing views scenarios than the full-view ones. Furthermore, leaving the fusion to a random view-selection as the FEmbr-sa model employs, does not work in the PM25 data, reaching negative R^2 values in all extreme cases.

Overall, we observe that the FCoM-av model obtains the best predictive performance without missing data and in moderate missingness, despite using the simplest aggregation function in our models. For the extreme missingness, the best combination for the merge function in the CoM depends on the task type. In classification, the best predictions are obtained with FCoM-av and FCoM-cr models, while in regression is with FCoM-ga and FCoM-me models. Furthermore, some models are greatly affected in these extreme cases. For instance, the FCoM-av model gets a significant performance drop in regression. In addition, the FCoMl-co model has poor results in most regression cases, reflecting the poor transferability and effectiveness of the zero-imputation as noted in the literature [25,46].

4.4. Results with a fraction of missing views

In Fig. 3 we display the predictive performance when one top view for prediction is missing in some samples. We include the results when another top view is missing in the Appendix B.1. We present our two best models in each dataset. In classification, our FCoM-ga model has the best results along the percentages of missing views, competing with FCoM-av and FCoMl-co. However, the competing model FCoMl-co has a

Table 2
Predictive performance in the regression tasks (R^2 score) for different cases (moderate and extreme) of missing views at inference time. The **best** and second best values are highlighted. In parentheses is the number of available views.

Model	Live Fuel Moisture Content (LFMC)						Particulate Matter 2.5 (PM25)			
	(6/6) No	(5/6) Only missing		(1/6) Only available		(3/3) No	(2/3) Only missing		(1/3) Only available	
	Missing	Radar	Optical	Optical	Radar	Missing	Condition	Dynamic	Dynamic	Condition
ITempD-co	0.691$_{\pm 0.052}$	0.036$_{\pm 0.100}$	0.036$_{\pm 0.100}$	−0.036$_{\pm 0.06}$	−0.036$_{\pm 0.06}$	**0.866**$_{\pm 0.093}$	0.074$_{\pm 0.035}$	−0.124$_{\pm 0.14}$	0.073$_{\pm 0.035}$	−0.124$_{\pm 0.14}$
ISensD-co	0.546$_{\pm 0.053}$	0.537$_{\pm 0.047}$	0.349$_{\pm 0.053}$	0.315$_{\pm 0.057}$	0.121$_{\pm 0.049}$	0.511$_{\pm 0.058}$	0.319$_{\pm 0.065}$	0.083$_{\pm 0.056}$	0.318$_{\pm 0.066}$	0.083$_{\pm 0.056}$
FSensD-cr	0.433$_{\pm 0.178}$	0.383$_{\pm 0.243}$	0.123$_{\pm 0.147}$	−0.031$_{\pm 0.557}$	−0.264$_{\pm 0.47}$	0.187$_{\pm 0.289}$	−1.046$_{\pm 2.77}$	−0.971$_{\pm 5.10}$	−1.091$_{\pm 2.78}$	−1.202$_{\pm 5.07}$
FCoMl-co	−0.643$_{\pm 2.14}$	−0.505$_{\pm 2.17}$	−0.434$_{\pm 2.00}$	−0.661$_{\pm 2.13}$	−1.172$_{\pm 1.93}$	−0.316$_{\pm 0.30}$	0.157$_{\pm 0.186}$	−0.010$_{\pm 0.12}$	0.358$_{\pm 0.391}$	0.125$_{\pm 0.104}$
FEmbr-sa	0.559$_{\pm 0.065}$	0.532$_{\pm 0.060}$	0.270$_{\pm 0.058}$	−5.300$_{\pm 1.80}$	−0.796$_{\pm 0.52}$	−1.366$_{\pm 4.39}$	−0.389$_{\pm 2.51}$	−0.135$_{\pm 0.42}$	[a]	−1.077$_{\pm 1.52}$
ESensI-av	0.321$_{\pm 0.038}$	0.294$_{\pm 0.036}$	0.239$_{\pm 0.035}$	0.194$_{\pm 0.224}$	−0.130$_{\pm 0.36}$	0.231$_{\pm 0.077}$	0.255$_{\pm 0.059}$	0.069$_{\pm 0.106}$	0.334$_{\pm 0.078}$	0.034$_{\pm 0.135}$
FCoM-av	0.628$_{\pm 0.051}$	0.606$_{\pm 0.066}$	0.435$_{\pm 0.058}$	−9.633$_{\pm 12.90}$	−7.425$_{\pm 7.24}$	0.660$_{\pm 0.103}$	**0.556**$_{\pm 0.162}$	−0.441$_{\pm 0.65}$	−6.570$_{\pm 3.81}$	**0.409**$_{\pm 0.147}$
FCoM-ga	**0.700**$_{\pm 0.043}$	**0.683**$_{\pm 0.044}$	**0.491**$_{\pm 0.05}$	0.326$_{\pm 0.09}$	0.165$_{\pm 0.094}$	0.046$_{\pm 0.151}$	0.096$_{\pm 0.133}$	**0.135**$_{\pm 0.122}$	0.239$_{\pm 0.345}$	0.237$_{\pm 0.155}$
FCoM-cr	0.425$_{\pm 0.154}$	0.139$_{\pm 1.187}$	0.227$_{\pm 0.153}$	−2.403$_{\pm 4.533}$	−2.292$_{\pm 3.112}$	0.414$_{\pm 0.108}$	0.374$_{\pm 0.126}$	−0.130$_{\pm 0.222}$	−0.279$_{\pm 2.271}$	−0.203$_{\pm 0.717}$
FCoM-me	0.644$_{\pm 0.066}$	0.641$_{\pm 0.06}$	0.479$_{\pm 0.061}$	**0.330**$_{\pm 0.072}$	**0.169**$_{\pm 0.077}$	−0.033$_{\pm 0.172}$	0.079$_{\pm 0.133}$	−0.004$_{\pm 0.137}$	**0.430**$_{\pm 0.087}$	0.230$_{\pm 0.141}$

[a] Represent values below −10.

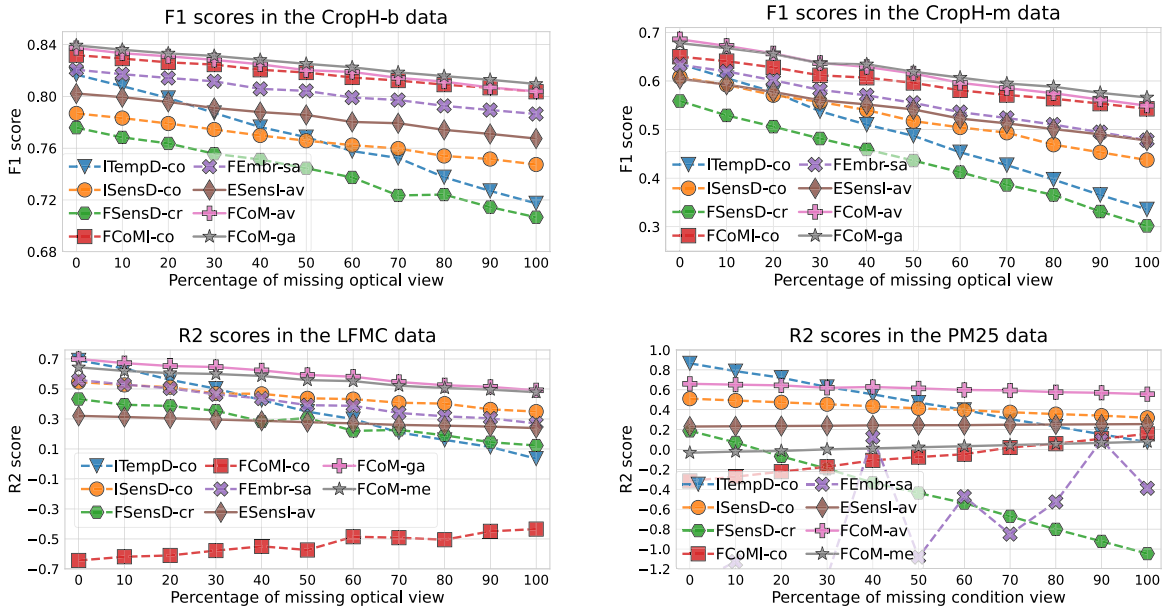


Fig. 3. Predictive performance when varying percentages of samples during inference time have a *top* view missing.

curious behavior in regression, focusing only on the missing views cases and obtaining poor results overall. Besides, the FEmbr-sa model has a strange behavior in the PM25 data, being ineffective for this scenario. In the PM25 data, the ITempD-co model has the best results until 30% of the samples have the condition view missing, from there, our FCoM-av model becomes the best. This is due to the greater robustness of our model to missing views. Overall, we notice that our models have the best robustness behavior when the number of samples with missing views increases. This behavior corresponds to a good balance between a small slope and a high value in the predictive performance curve.

4.5. Prediction shift due to missing views

We analyze how the model predictions are shifted because of missing views, regardless of the target values. To this end, we plot the class change ratio and the deformation score at different percentages of missing views in the CropH-m and LFMC data in Fig. 4. The deformation is calculated as the error difference in the prediction with and without missing views divided by the deviation of the full-view prediction, i.e. $\mathrm{RMSE}(\hat{y}_{\mathrm{full}}, \hat{y}_{\mathrm{miss}})/\mathrm{std}(\hat{y}_{\mathrm{full}})$. These graphs show the shift of the model predictions when there are more missing views during inference. We observe that the prediction shift curves of our models are among the three lower values in the models compared, together with FCoMl-co and ESensI-av models.

As qualitative support for previous results, we plot the class change in the CropH-m data in Fig. 5, and the shift in the real-value predicted in the LFMC data in Fig. 6. We notice that in moderate missingness, the prediction change in our models is insignificant, while in extreme cases is shifted to a greater extent. As the predicted value in classification is categorical, the change in prediction is binary (the class changes to another one or not), while in the regression task, the predicting value is continuous. Therefore, we can see how the prediction disperses from the original value to different degrees in each sample, which could be associated with the increased difficulty of robustness in regression tasks.

4.6. Computational scalability

We depict the computational usage of all compared models in Table 3. As expected, the models employing the input-level fusion have the lowest number of parameters, based on the single encoder architecture these models use. Conversely, the model with more number of parameters is the FCoM-me, as this model uses two layers in the dynamic merge function, followed by the FCoM-ga model and the ones based on the cross-attention fusion, FCoM-me, and FSensD-cr. Among the models using multiple encoders, the FCoM-av model has the lowest number of parameters, as well as FEmbr-sa and ESensI-av models. The same pattern is followed in the memory usage of the models. This is because

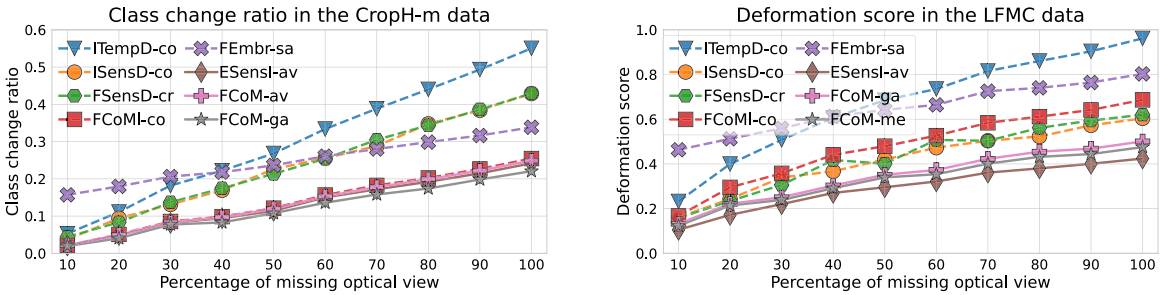


Fig. 4. Prediction shift score in classification (class change ratio) and regression (deformation score) tasks.

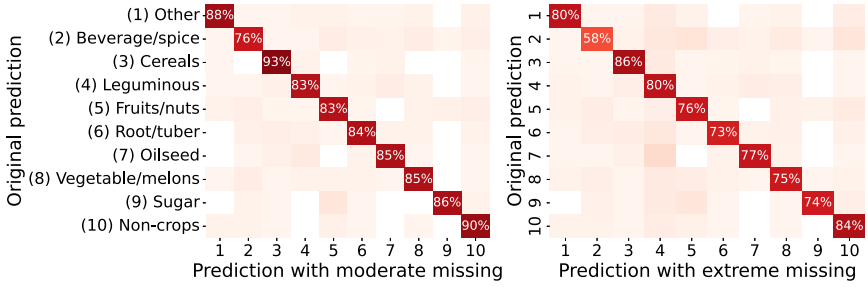


Fig. 5. Class prediction shift with moderate (radar view missing), and extreme (optical view available) missingness. The FCoM-ga model is shown in the CropH-m data. The overall class change ratio is 12.5% in the moderate and 20.4% in the extreme cases.

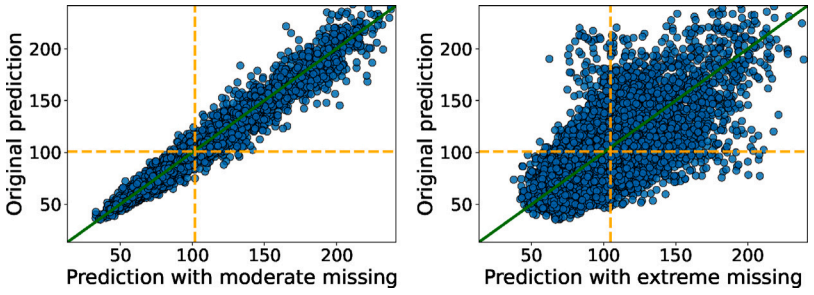


Fig. 6. Real-value prediction shift with moderate (radar view missing), and extreme (only optical view available). The FCoM-ga model is shown in the LFMC data. The overall deformation score is 0.208 in moderate and 0.722 in extreme cases.

Table 3
Computational use of MVL models, where MB stands for the megabytes unit. The forward usage considers the model usage and adds the memory use of the batch (128-size) in the forward pass of each model.

Model	Parameters (millions)	Model usage (MB)	Forward usage (MB)
ITempD-co	1.05	4.01	42.67
ISensD-co	1.05	4.01	42.67
FSensD-cr	3.20	12.24	96.05
FCoMl-co	3.15	12.04	96.38
FEmbr-sa	3.10	11.86	96.77
ESensI-av	3.10	11.85	90.10
FCoM-av	3.10	11.86	96.77
FCoM-ga	3.36	12.86	108.34
FCoM-cr	3.20	12.24	135.38
FCoM-me	3.76	14.37	204.38

the memory usage is directly related to the number of parameters the models have. In the forward usage of the models, we notice that all competing models have a memory use of less than 100 MB, with FCoM-av the only one of our models that accomplishes this. All other variants using the CoM technique with sophisticated merge functions (-ga, -cr, and -me suffix), have a forward usage of more than 100 MB, even 200 MB in the case of the FCoM-me model. This shows the capacity of our FCoM-av model to match the efficiency of methods from the literature.

We depict the execution time of all compared models in Fig. 7. During training, we note that models ignoring missing views led to the most efficient training time, such as FSensD-cr, ESensI-av, and FCoM-av models. Comparing FSensD-cr and FCoM-av, both using dynamic merge functions in feature-level fusion models, our FCoM-av model has a small training time increase of 33% per epoch. This is a special case in FCoM-av and its simple aggregation function, as our models using the CoM technique scale with all combinations of missing views, 2^m, as discussed in Section 3.4. Thus, we notice that the combination of the CoM technique with sophisticated merge functions (-ga, -cr, and -me suffix) increases the training time (per epoch) by almost the double, from around 4 s per epoch in FCoM-av to more than 7 s. However, the highest computation time is associated with the forward over the encoders, which occurs only once in our methods. On the inference time, the most efficient prediction time over the entire dataset is associated with the models employing input-level fusion, i.e. ITempD-co and ISensD-co. This outcome is related to the low number of parameters in these models, as displayed in Table 3. Nevertheless, our CoM-based models have a more efficient prediction time when more data is missing. This is caused by the dynamic merge function, as it allows ignoring missing views and performing calculations on just the available data (no fake data insertion). This relates to the adaptable scalability of our dynamic merge functions during inference based on the available views $m^{(i)}$ for each sample inference i, discussed in Section 3.3.

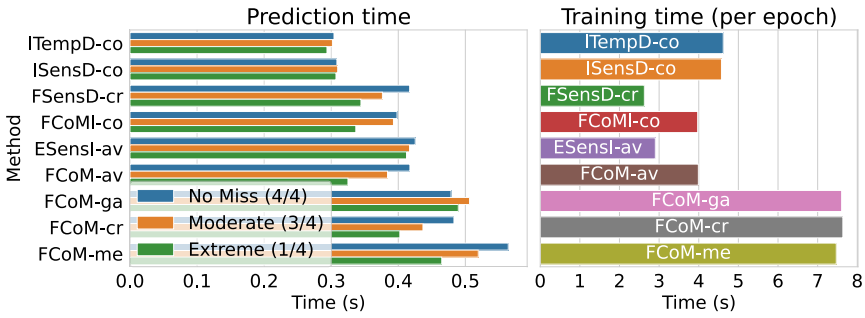


Fig. 7. Execution time of different MVL models. The times are calculated in the CropH-m data with 4 views. Prediction times are separated into no missing (4/4), only one view missing (3/4), and only one view available (1/4).

Table 4
Different MAug techniques applied at the input (concatenation) and feature (average) level.

MAug	Level	CropH-m (F1 scores)					LFMC (R^2 scores)				
		(4/4) No	(3/4) Only missing		(1/4) Only available		(6/6) No	(5/6) Only missing		(1/6) Only available	
		Missing	Radar	Optical	Optical	Radar	Missing	Radar	Optical	Optical	Radar
–	Input	0.655	0.569	0.359	0.230	0.081	0.638	0.593	0.262	0.293	0.057
SensD	Input	0.566	0.522	0.422	0.401	0.206	0.506	0.468	0.342	0.305	0.134
CoM	Input	0.652	0.637	0.385	0.602	0.123	0.515	0.484	0.355	**0.317**	**0.118**
–	Feature	0.647	0.561	0.492	0.269	0.224	**0.648**	0.531	0.151	[a]	[a]
SensD	Feature	0.634	0.615	0.526	0.590	0.386	0.511	0.496	0.302	[a]	−1.709
CoM	Feature	**0.679**	**0.646**	**0.549**	**0.645**	**0.430**	0.625	**0.604**	**0.437**	−9.632	−8.362

[a] A value below −10.

5. Discussion

MAug variations. First, we compare different ways of applying the MAug techniques. We consider the random-wise version (SensD, [15, 26]), and the all-combinations one (CoM), applied at the input-level like literature [23,25] or at the feature-level. The results are shown in Table 4. We zero-impute views if they are missing at input-level, and ignore them if missing at feature-level via average as fusion. Similar results are observed when other merge functions are used (see Appendix B.2). We notice an increase in the robustness of different scenarios of missing views when the CoM is used, compared to SensD and without MAugs. Randomly dropping sensors, used in Chen et al. [15], Mena et al. [25] and Xu et al. [26], have good relative robustness but quite low overall performance, unlike CoM, which improves overall. The CoM technique even allows improving the full-view performance in the CropH-m data. Moreover, we notice that the CoM technique works better at the feature-level by ignoring missing views than at the input-level with zero imputation. In addition, the SensD technique has the additional step of finding the optimal dropout parameter, while the CoM is a parameter-free MAug alternative. The evidence suggests that, in most cases, our usage of CoM at feature-level and dynamic fusion is optimal for model robustness.

Outlook on transformer-based modeling. We remark that Transformer models are not naturally robust to missing data [13]. They can handle missing data without intervention, but that does not imply they will obtain the same performance as when there is no missing data [15,47]. For instance, we show that using the cross-attention fusion (based on Transformer models) is not optimal in our evaluation of different tasks and missing data cases.

Results variation. Along the experimentation, we note a slight variation in which model obtains the best results in each dataset and missing view scenario. However, this variability is expected in the EO domain, as the data is quite heterogeneous and region-dependent [2,5,35,42, 55]. In addition, this variation depends on the metrics used to assess the models. In our study, standard performance metrics are used from the literature. However, additional values are included in the Appendix C. Nevertheless, our combination of CoM with the simple average function (FCoM-av) shows good overall results, without significantly increasing the training time, as well as having an adaptive prediction time based on the available views.

Limitation on the validation scope. In our work, we assess the effect of missing views only at inference time, assuming a full-view training dataset. However, based on the dynamic fusion, it could be easily extended to other settings. Nevertheless, recently, there has been more focus on research and infrastructure for model inference and its practical energy consumption [56]. This aligns with our research to make models adaptive to the available data for their inference. Furthermore, we validate our models using only pixel-wise EO datasets. Although this validation is conducted across four datasets, its effectiveness in other domains needs to be verified.

Extension beyond EO data. The individual components presented in our work can be applied to other domains where the views complement each other, but at the same time, they can be replaceable. Thus, the dynamic merge functions can be included in any MVL model implementing feature-level fusion where the encoded representations have the same dimensionality. For instance, in text–image–audio data, the text, image, and audio can be independently encoded into a vector of the same dimensionality and then the dynamic merge function can be applied, as mentioned in Section 3.3. Moreover, the CoM technique can be included in any multi-view dataset to augment the training samples by simulating missing views, as described in Section 3.4. For the same example mentioned, all combinations of missing audio, image, or text can be simulated and replaced by zero (or the mean features). In addition, this can be combined with the dynamic merge function and ignore the features coming from text, image, or audio if they are dropped.

General outlook. Throughout the experimentation it can be noted that there is no single model that is best for all cases, one does not fit all. However, our models (FCoM-av, FCoM-ga, FCoM-cr, FCoM-me) have a consistent advantage in various scenarios of missing views

(moderate and extreme), different top views missing, as well as when different percentages of samples have missing views. In addition, our validation is not only on classification using static views as in the literature [2,26,27], but considers broader settings including regression tasks and temporal views data. However, there is an increase in computational complexity during training when the CoM technique is used. The complexity scales with an additional factor associated with the number of missing views that can be simulated, $\mathcal{O}(m \cdot 2^m)$ for FCoM-av compared to $\mathcal{O}(m)$ for other feature-level fusion models like FSensD-cr, and FEmbr-sa. However, this increase occurs only at training time and can be reduced by forwarding over the encoders only once. During inference, FCoM-av, FCoM-cr, and FCoM-me scale as any feature-level fusion model but, unlike models based on fixed-size merge function, $\mathcal{O}(m)$ [23,25–27], are adaptable to the available views as $\mathcal{O}(m^{(i)})$.

6. Conclusion

The lack or unavailability of views during inference is intrinsic in the EO domain. This is because data collection occurs under real-world operational constraints, such as limited spatial coverage, maintenance, or errors. In the literature, limited exploration has been done into the robustness of Multi-View Learning (MVL) models to these potential missing views. For example, most models rely on randomly simulating missing views based on a dropout ratio and replacing data with zeros. To address these limitations, we introduce a parameter-free Missing data as Augmentation (MAug) technique tailored to missing views in MVL, named Combinations of Missing views (CoM). Contrary to the literature, we apply this technique at the feature-level in combination with four dynamic merge functions. For evaluation, we simulate missing views during inference to assess model robustness in two *missingness* scenarios, moderate (single-sensor missing) and extreme (single-sensor available). The findings show that our models outperform competing models in moderate missingness, particularly with an average-based fusion. Moreover, due to the MAug effect, we observe a classification improvement in the full-view scenario. In addition, we identify challenging scenarios, particularly in regression tasks and extreme missingness. For these challenges, future work should consider designing models that operate with any available data at decision-level fusion, such as weighing predictions with missing data.

CRediT authorship contribution statement

Francisco Mena: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Diego Arenas:** Writing – review & editing, Supervision, Conceptualization. **Andreas Dengel:** Supervision, Resources, Funding acquisition.

Data and code availability

The data used in this manuscript corresponds to public benchmark datasets released in Tseng et al. [33], Rao et al. [51] and Chen [52]. We provide functions to facilitate the processing of these to a machine learning ready structure. This is available on our GitHub that will be released at https://github.com/fmenat/com-views.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

F. Mena acknowledges support through a scholarship of the University of Kaiserslautern-Landau.

Table A.5
Name of features in each view in the **PM25** data [52]. The source of the views are ground-based stations.

View	Features
Conditions	Dew point, temperature, and humidity
Dynamics	Pressure, combined wind direction, cumulated wind speed, season
Precipitation	Precipitation, and cumulated precipitation

Table A.6
Name of features in each view in the **CropH-b** and **CropH-m** data [33]. The views are at 10 m spatial resolution.

View	Source	Features
Optical	Sentinel-2 (level 1C)	B2 (blue), B3 (green), B4 (red), B5, B6, B7, B8, B8A, B9, B11, B12, NDVI
Radar	Sentinel-1 (C-band)	VV and VH polarization bands
Weather	ERA5	Temperature and precipitation
Topographic	NASA's SRTM	Elevation and slope

Table A.7
Name of features in each view in the **LFMC** data [51]. The views are at 250 m spatial resolution.

View	Source	Features
Optical	Landsat 8	Red, green, blue, near infrared, short-wave infrared, NDVI, NDWI, NIRv
Radar	Sentinel-1 (C-Band)	VV, VH, and VH/VV polarization bands
Topographic	National Elevation Database	Elevation and slope
Soil	Unified North American Soil Map	Silt, sand, and clay
LiDAR	Global Laser Altimetry System	Canopy height (ordinal value)
Land-cover	GLOBCOVER	Class label between 12 options

Appendix A. Initial setup

A.1. Dataset description

We show the features from each view in the different datasets in Tables A.5, A.6, and A.7 for PM25, CropHarvest, and LFMC data respectively. The abbreviations used in the tables correspond to: normalized difference vegetation index (NDVI), normalized difference water index (NDWI), and near infrared vegetation index (NIRv).

A.2. Architecture selection

In Table A.8 we compare the MAug techniques to a view-permutation in the memory-based fusion with a LSTM architecture. Since views are processed sequentially, the view-permutation is included to prevent the model from overfitting the views order [28]. We observe that the usage of the MAug technique has a greater positive effect on the predictive performance than the view-permutation in the full-view scenario and with missing views. This suggests that the memory fusion with the MAug does not require an explicit permutation to become order invariant and increase generalization. Nevertheless, the good behavior without permutation in LFMC data might be just overfitting, caused by the small dataset (less than 2000 samples to train).

In Table A.9 we compare different architectures in the memory fusion. We notice a tendency to get better results (in full-view and with missing views) with a more complex network. Overall, the best results are obtained with two bidirectional LSTM layers, while the second-best results are associated with an architecture based on GRU layers.

In Table A.10 we compare different architectural options of the cross-attention fusion, based on Transformer layers. We observe an

Table A.8
Memory fusion-based MVL model with different configurations of view-permutation and MAug techniques.

MAug	Permutation	CropH-m (F1)		LFMC (R^2)	
		No missing	Missing optical	No missing	Missing optical
–	–	0.656	0.503	**0.735**	0.095
–	All	0.649	0.487	0.706	0.192
SensD	All	0.643	0.536	0.551	0.353
–	Random	0.643	0.496	0.666	0.224
SensD	Random	0.641	0.531	0.507	0.327
CoM	Random	0.666	0.555	0.595	0.428
CoM	–	**0.672**	**0.559**	0.613	**0.412**

Table A.9
Memory fusion-based MVL model with different network architectures. "Bi" stands for bidirectional layer.

Gate	Layers	CropH-m (F1)		LFMC (R^2)	
		No missing	Missing optical	No missing	Missing optical
GRU	1	0.666	0.554	0.627	0.422
GRU	1 (Bi)	0.671	0.554	0.636	0.451
GRU	2 (Bi)	0.663	0.552	0.634	0.468
GRU	3 (Bi)	0.661	0.546	0.626	0.450
LSTM	1 (Bi)	0.670	0.552	0.634	0.461
LSTM	2 (Bi)	**0.677**	**0.557**	**0.657**	**0.477**

Table A.10
Cross-attention fusion-based MVL model with different network architectures.

Heads	Layers	CropH-m (F1)		LFMC (R^2)	
		No missing	Missing optical	No missing	Missing optical
1	1	0.637	0.468	0.545	0.375
2	1	0.622	0.462	0.551	0.380
4	1	0.650	0.511	0.553	0.380
4	2	0.654	0.511	0.536	0.351
8	1	**0.659**	**0.520**	**0.569**	**0.400**
8	2	0.650	0.504	0.541	0.371
8	3	0.652	0.519	0.541	0.362

Table A.11
Predictive performance of individually trained models (per view). The F1 scores are shown in classification tasks and R^2 scores in the regression tasks. The **best** and second best values are highlighted.

View	CropH-b	CropH-m	LFMC	PM25
Optical	$\mathbf{0.791}_{\pm 0.013}$	$\mathbf{0.635}_{\pm 0.023}$	$\mathbf{0.194}_{\pm 0.224}$	
Radar	$0.752_{\pm 0.012}$	$0.444_{\pm 0.022}$	$0.050_{\pm 0.360}$	
Topographic	$0.631_{\pm 0.044}$	$0.095_{\pm 0.028}$	$-0.124_{\pm 0.590}$	
Weather	$0.701_{\pm 0.012}$	$0.346_{\pm 0.013}$		
Soil			$-0.245_{\pm 0.557}$	
LiDAR			$-0.033_{\pm 0.147}$	
Land-cover			$-0.021_{\pm 0.102}$	
Conditions				$0.034_{\pm 0.135}$
Dynamics				$\mathbf{0.334}_{\pm 0.078}$
Precipitation				$-0.072_{\pm 0.068}$

optimal value across datasets and missing scenarios with eight heads and just one layer. Besides, we notice a slight tendency to get better results when using a more complex network architecture.

A.3. Individual view performance

In order to detect the *top* views for prediction in each dataset, we train an individual model on each view. The results for each dataset are shown in Table A.11. For CropH-b, CropH-m, and LFMC these are optical and radar views, while for the PM25 dataset the top views are dynamic and condition. We note that the static views usually have a low predictive performance, only serving as a complement to the *top* views for prediction.

Appendix B. Additional results

B.1. Another top view missing

In Fig. B.8 we display the predictive performance in all datasets when additional views are missing in some samples, as a complement to Fig. 3. We present our two best models in each dataset. In the classification tasks, the proposed FCoM-ga model has the best predictive performance along the percentage of missing data, as observed when the top view is missing (Fig. 3). Overall, our models show the best behavior (of a good balance between a small slope and a high value) when increasing the level of samples with missing views. Furthermore, most of the models have a high robustness to missing the radar view in the classification tasks. Surprisingly, FCoM-av has a strange behavior in the PM25 data, being the only one greatly affected by missing the precipitation view. Perhaps our model learned a prediction quite dependent on this view in that dataset.

B.2. Dynamic merge function comparison

We analyze the effect of applying two MAug techniques at different levels of the MVL models, similar to Table 4 that shows this analysis for average fusion. The Tables B.12, B.13, and B.14 include the results when using gated fusion, cross-attention fusion, and memory fusion respectively. When views are missing at input-level, they are imputed, and when views are missing at feature-level, they are ignored. We notice the same behavior observed for the average, i.e. (i) tendency to increase the model robustness to missing data when the CoM technique is used, compared to SensD and without MAug techniques, (ii) generalization behavior (increase in performance) of MAug techniques when there is no missing views. Nevertheless, we notice that in some cases the MAug technique impairs the model performance due to the difficulty in estimating the target with missing views.

Appendix C. Results with additional metrics

We assess the predictive performance with alternative metrics: area under the curve (AUC) of the precision–recall plot in classification, and mean average percentage error (MAPE) in regression tasks. In addition, for assessing the robustness, we use the Performance Robustness Score (PRS) presented in [57]. The PRS is based on the predictive error with missing views relative to the predictive error in the full-view scenario:

$$\mathrm{PRS}(y, \hat{y}_{miss}, \hat{y}_{full}) = \exp\left(1 - \frac{\mathrm{RMSE}(y, \hat{y}_{miss})}{\mathrm{RMSE}(y, \hat{y}_{full})}\right), \tag{C.1}$$

then it is normalized as $PRS = \min(1, PRS)$. The results for the CropH-b, CropH-m, LFMC, and PM25 data are in Tables C.15, C.16, C.17, and C.18 respectively. We notice that the model robustness cannot be assessed only with relative robustness metrics, such as PRS. This is because the relative metrics hide the overall predictive performance. For instance, a horizontal line behavior in Fig. 3, such as from ISensD-co, will get a PRS of one, independently of the position of this line on the y-axis (performance). Even, in some cases, the prediction shift due to missing views can go towards correcting the original prediction, as shown in FCoMl-co in Fig. 3. In our work, we include these metrics for further analysis, but metrics that can mix these concepts could allow a more succinct analysis.

We plot the PRS value when different numbers of samples have the top views missing in Fig. C.9. We notice a different behavior in this relative score compared to the results in Fig. 3. In the PRS analysis, the best results are obtained by the ESensI-av model followed by FCoMl-co. The curve of our models is between the third and fourth best in this relative score. This reflects that, despite the good behavior of our models in the predictive performance, there is still a gap in reaching the predictive robustness (based on PRS) of the competing models, such as the one of ESensI-av.

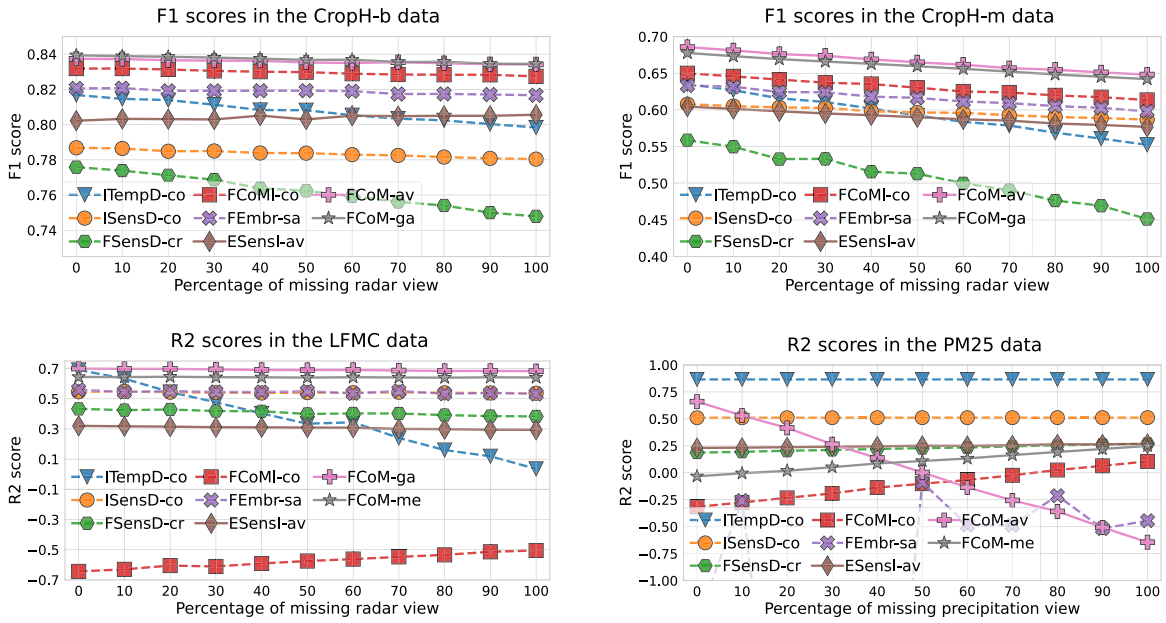


Fig. B.8. Predictive performance and robustness when varying percentages of validation samples have a *top* view missing.

Table B.12
Different configurations of the MAug technique applied at input or feature level with **gated fusion**. The F1 scores are shown in the CropH-m data and R^2 scores in the LFMC data.

MAug	Level	CropH-m (F1)					LFMC (R^2)				
		(4/4) No	Only missing (3/4)		Only available (1/4)		(6/6) No	Only missing (5/6)		Only available (1/6)	
		missing	Radar	Optical	Optical	Radar	missing	Radar	Optical	Optical	Radar
–	Input	0.643	0.506	0.354	0.189	0.073	0.735	0.631	0.238	0.215	−0.011
SensD	Input	0.592	0.562	0.479	0.505	0.323	0.545	0.522	0.324	−3.148	−5.625
CoM	Input	0.662	0.619	0.417	0.565	0.089	0.615	0.620	0.292	0.234	0.012
–	Feature	0.653	0.574	0.494	0.355	0.250	**0.738**	0.660	0.225	[a]	[a]
SensD	Feature	0.668	0.632	0.538	0.610	0.387	0.532	0.499	0.336	−3.151	−0.550
CoM	Feature	**0.678**	**0.647**	**0.568**	**0.627**	**0.418**	0.684	**0.678**	**0.471**	**0.326**	**0.158**

[a] A value below −10.

Table B.13
Different configurations of the MAug technique applied at input or feature level with **cross-attention fusion**. The F1 scores are shown in the CropH-m data and R^2 scores in the LFMC data. The † is a value below −10.

MAug	Level	CropH-m (F1)					LFMC (R^2)				
		(4/4) No	Only missing (3/4)		Only available (1/4)		(6/6) No	Only missing (5/6)		Only available (1/6)	
		missing	Radar	Optical	Optical	Radar	missing	Radar	Optical	Optical	Radar
–	Input	0.646	0.585	0.239	0.432	0.072	0.575	0.521	0.289	0.242	0.053
SensD	Input	0.569	0.54	0.438	0.523	0.302	0.522	0.503	0.303	−0.005	−0.205
CoM	Input	0.655	0.636	0.216	0.620	0.071	0.554	0.481	**0.361**	**0.295**	**0.089**
–	Feature	0.637	0.589	0.495	0.506	0.287	**0.579**	**0.539**	−0.388	−0.058	−2.702
SensD	Feature	0.639	0.616	0.510	0.617	0.403	0.536	0.525	0.315	0.013	0.019
CoM	Feature	**0.665**	**0.636**	**0.547**	**0.638**	**0.441**	0.520	0.471	0.278	−0.363	−1.005

Table B.14
Different configurations of the MAug technique applied at input or feature level with **memory fusion**. The F1 scores are shown in the CropH-m data and R^2 scores are shown in the LFMC data.

MAug	Level	CropH-m (F1)					LFMC (R^2)				
		(4/4) No	Only missing (3/4)		Only available (1/4)		(6/6) No	Only missing (5/6)		Only available (1/6)	
		missing	Radar	Optical	Optical	Radar	missing	Radar	Optical	Optical	Radar
–	Input	0.651	0.572	0.492	0.287	0.237	0.735	0.634	0.072	−4.165	†
SensD	Input	0.586	0.547	0.471	0.490	0.319	0.568	0.549	0.359	−0.015	−0.383
CoM	Input	**0.662**	**0.630**	0.556	**0.618**	0.413	0.648	0.630	**0.420**	**0.346**	0.146
–	Feature	0.641	0.568	0.499	0.316	0.235	**0.741**	**0.671**	0.022	−0.753	−3.337
SensD	Feature	0.638	0.610	0.533	0.594	0.378	0.557	0.547	0.372	−0.410	−0.045
CoM	Feature	0.661	0.624	**0.558**	0.617	**0.421**	0.548	0.563	0.410	0.342	**0.155**

Table C.15
Additional results for different cases of missing views (moderate and extreme) in the **CropH-b** data. We highlight the **best** and second best value in each scenario. The value in parentheses is the number of available views.

Model	AUC value (↑)							PRS value (↑)					
	(4/4) No	(3/4) Only missing			(1/4) Only available			(3/4) Only missing			(1/4) Only available		
	Missing	Radar	Optical	Weather	Optical	Radar	Weather	Radar	Optical	Weather	Optical	Radar	Weather
ITempD-co	0.920	0.908	0.813	0.755	0.772	0.666	0.604	0.958	0.810	0.751	0.761	0.606	0.600
ISensD-co	0.903	0.899	0.864	0.801	0.869	0.736	0.676	0.984	0.930	0.815	0.982	0.739	0.790
FSensD-cr	0.876	0.847	0.798	0.734	0.687	0.637	0.542	0.950	0.861	0.790	0.783	0.665	0.645
FCoMl-co	0.930	0.925	0.903	0.858	0.889	0.791	0.697	**0.996**	**0.948**	0.863	0.906	0.733	0.707
FEmbr-sa	0.923	0.917	0.889	0.843	0.866	0.763	0.676	0.987	0.915	0.814	0.800	0.651	0.648
ESensI-av	0.907	0.907	0.879	0.858	0.900	**0.805**	0.691	**0.996**	0.943	**0.911**	**0.995**	**0.863**	**0.794**
FCoM-av	0.933	**0.931**	0.912	0.875	**0.908**	0.800	**0.714**	0.991	0.890	0.780	0.859	0.478	0.701
FCoM-ga	**0.934**	**0.931**	**0.915**	**0.876**	0.904	0.801	0.711	0.986	0.908	0.781	0.897	0.645	0.723
FCoM-cr	0.925	0.922	0.903	0.866	0.901	0.800	0.706	0.991	0.922	0.849	0.919	0.672	0.733
FCoM-me	0.932	0.921	0.908	0.865	0.903	0.800	0.707	0.944	0.899	0.769	0.904	0.682	0.711

Table C.16
Additional results for different cases of missing views (moderate and extreme) in the **CropH-m** data. We highlight the **best** and second best value in each scenario. The value in parentheses is the number of available views.

	AUC value (↑)							PRS value (↑)					
	(4/4) No	(3/4) Only missing			(1/4) Only available			(3/4) Only missing			(1/4) Only available		
Model	Missing	Radar	Optical	Weather	Optical	Radar	Weather	Radar	Optical	Weather	Optical	Radar	Weather
ITempD-co	0.962	0.946	0.837	0.888	0.850	0.661	0.740	0.893	0.674	0.788	0.716	0.649	0.558
ISensD-co	0.904	0.869	0.732	0.949	0.760	0.595	0.762	0.937	0.759	0.979	0.821	0.718	0.706
FSensD-cr	0.928	0.864	0.796	0.874	0.779	0.673	0.692	0.905	0.769	0.895	0.834	0.695	0.676
FCoMl-co	0.964	0.958	0.937	0.954	0.945	0.856	0.872	0.953	0.894	0.978	0.950	0.776	0.748
FEmbr-sa	0.956	0.950	0.911	0.938	0.929	0.797	0.847	0.937	0.820	0.860	0.761	0.562	0.704
ESensI-av	0.948	0.945	0.912	0.950	**0.957**	**0.885**	0.867	**0.971**	**0.895**	**0.997**	**0.991**	**0.884**	**0.813**
FCoM-av	**0.967**	**0.962**	0.937	**0.960**	0.954	0.864	0.873	0.929	0.810	0.945	0.851	0.622	0.601
FCoM-ga	**0.967**	**0.962**	**0.944**	0.938	0.932	0.854	0.861	0.938	0.838	0.951	0.885	0.661	0.620
FCoM-cr	0.962	0.958	0.935	0.959	0.953	0.875	0.879	0.949	0.820	0.987	0.941	0.703	0.632
FCoM-me	0.965	0.959	0.934	0.959	0.954	0.873	**0.880**	0.942	0.824	0.975	0.943	0.738	0.669

Table C.17
Additional results for different cases of missing views (moderate and extreme) in the **LFMC** data. We highlight the **best** and second best value in each scenario. The value in parentheses is the number of available views.

Model	MAPE value (↓)					PRS value (↑)			
	(6/6) No	(5/6) Only missing		(1/6) Only available		(5/6) Only missing		(1/6) Only available	
	Missing	Radar	Optical	Optical	Radar	Radar	Optical	Optical	Radar
ITempD-co	0.157	0.293	0.293	0.335	0.335	0.465	0.465	0.433	0.433
ISensD-co	0.191	0.189	0.230	**0.249**	0.292	0.980	0.819	0.794	0.674
FSensD-cr	0.211	0.221	0.287	0.327	0.377	0.966	0.772	0.727	0.627
FCoMl-co	0.406	0.368	0.333	0.378	0.510	**0.999**	**0.977**	**0.926**	**0.793**
FEmbr-sa	0.191	0.196	0.265	0.721	0.381	0.964	0.748	0.073	0.381
ESensI-av	0.254	0.276	0.278	0.252	**0.283**	0.980	0.943	0.911	0.769
FCoM-av	0.173	0.178	0.235	0.907	0.623	0.968	0.791	0.073	0.093
FCoM-ga	**0.150**	**0.156**	**0.198**	0.252	0.297	0.971	0.737	0.608	0.513
FCoM-cr	0.217	0.228	0.250	0.405	0.429	0.920	0.847	0.432	0.383
FCoM-me	0.167	0.165	**0.198**	0.268	0.299	0.981	0.806	0.685	0.586

Table C.18
Additional results for different cases of missing views (moderate and extreme) in the **PM25** data. We highlight the **best** and second best value in each scenario. The value in parentheses is the number of available views.

Model	MAPE value (↓)					PRS value (↑)			
	(3/3) No	(2/3) Only missing		(1/3) Only available		(2/3) Only missing		(1/3) Only available	
	Missing	Condition	Dynamic	Dynamic	Condition.	Condition	Dynamic	Dynamic	Condition
ITempD-co	**0.439**	**0.439**	1.242	1.033	1.594	0.186	0.146	0.185	0.146
ISensD-co	0.879	0.878	1.301	1.151	1.644	0.835	0.690	0.834	0.690
FSensD-cr	1.003	1.149	1.301	1.147	1.300	0.783	0.862	0.757	0.832
FCoMl-co	1.281	0.713	1.088	0.747	1.422	**0.994**	**1.000**	0.984	**1.000**
FEmbr-sa	0.717	0.712	0.976	1.038	1.330	0.933	0.903	0.591	0.693
ESensI-av	0.953	0.754	1.223	0.699	1.561	0.989	0.906	0.991	0.889
FCoM-av	0.581	0.561	2.891	4.957	1.162	0.869	0.408	0.080	0.720
FCoM-ga	0.651	0.658	1.030	0.672	1.051	0.980	0.995	0.996	0.992
FCoM-cr	0.597	0.614	**0.728**	**0.651**	**0.805**	0.949	0.680	0.847	0.701
FCoM-me	0.672	0.653	0.884	0.803	1.318	0.979	0.980	**1.000**	**1.000**

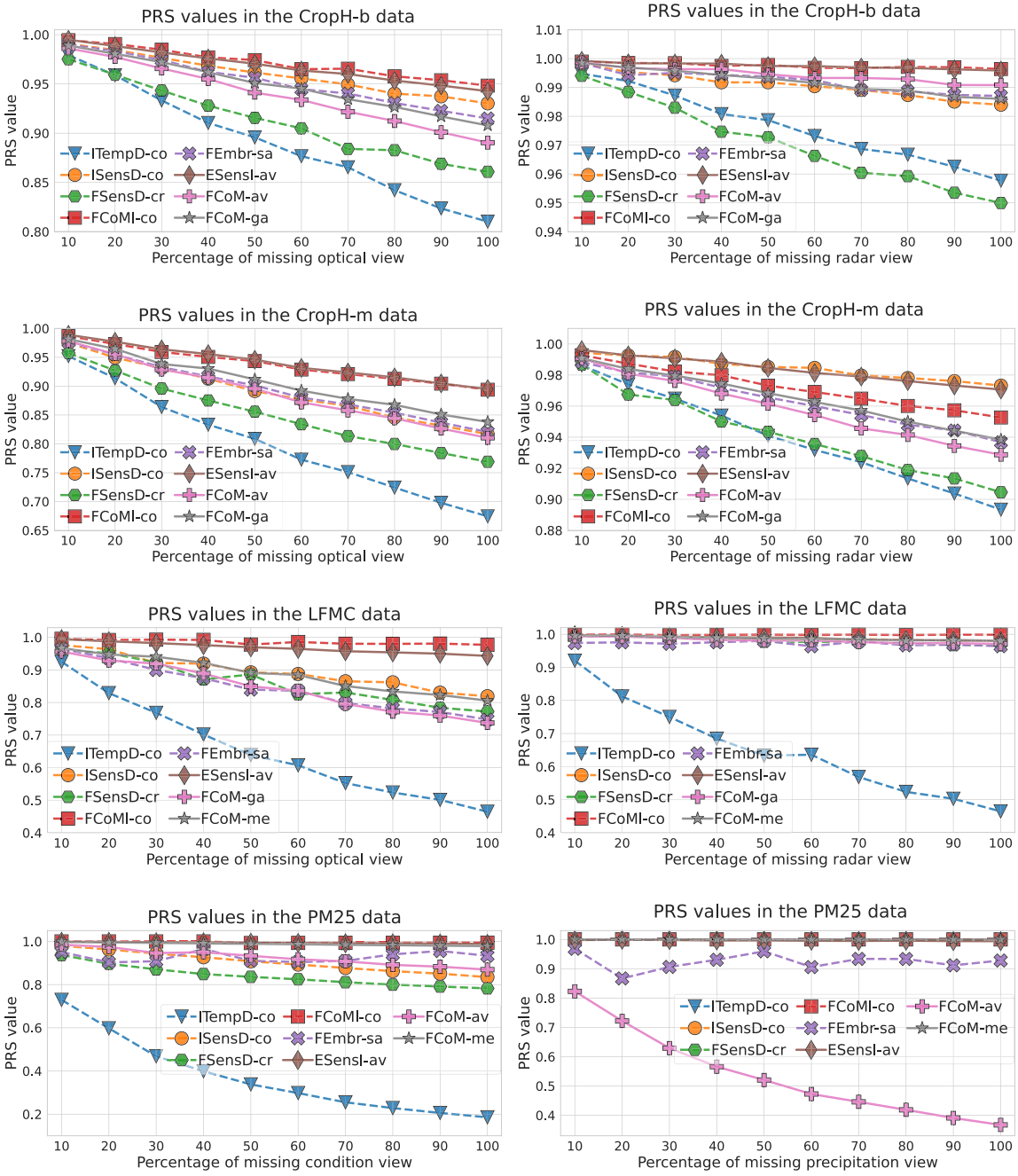


Fig. C.9. Predictive robustness results when the *top* views are missing for a given percentage of samples at inference.

Data availability

The data used in this manuscript corresponds to public benchmark datasets released in Tseng et al. [33],Rao et al. [51],Chen [52]. We provide functions to facilitate the processing of these to a machine learning ready structure. This is available on our GitHub at https://github.com/fmenat/com-views/tree/main/data.

References

[1] X. Yan, S. Hu, Y. Mao, Y. Ye, H. Yu, Deep multi-view learning methods: A review, Neurocomputing 448 (2021) 106–129, http://dx.doi.org/10.1016/j.neucom.2021.03.090.

[2] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, B. Zhang, More diverse means better: Multimodal deep learning meets remote-sensing imagery classification, IEEE Trans. Geosci. Remote Sens. 59 (5) (2021-05) 4340–4354, http://dx.doi.org/10.1109/TGRS.2020.3016820.

[3] V. Sainte Fare Garnot, L. Landrieu, N. Chehata, Multi-modal temporal attention models for crop mapping from satellite time series, ISPRS J. Photogramm. Remote Sens. 187 (2022-05-01) 294–305, http://dx.doi.org/10.1016/j.isprsjprs.2022.03.012.

[4] F. Mena, D. Arenas, M. Nuske, A. Dengel, Common practices and taxonomy in deep multi-view fusion for remote sensing applications, IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. (2024) 4797–4818, http://dx.doi.org/10.1109/JSTARS.2024.3361556.

[5] E. Rolf, K. Klemmer, C. Robinson, H. Kerner, Mission critical–Satellite data is a distinct modality in machine learning, 2024, arXiv preprint arXiv:2402.01444.

[6] K.K. Gadiraju, B. Ramachandra, Z. Chen, R.R. Vatsavai, Multimodal deep learning based crop classification using multispectral and multitemporal satellite imagery, in: Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining, SIGKDD, 2020, pp. 3234–3242, http://dx.doi.org/10.1145/3394486.3403375.

[7] H. Shen, X. Li, Q. Cheng, C. Zeng, G. Yang, H. Li, L. Zhang, Missing information reconstruction of remote sensing data: A technical review, IEEE Geosci. Remote. Sens. Mag. 3 (3) (2015) 61–85, http://dx.doi.org/10.1109/MGRS.2015.2441912.

[8] B.L. Markham, J.C. Storey, D.L. Williams, J.R. Irons, Landsat sensor performance: History and current status, IEEE Trans. Geosci. Remote Sens. 42 (12) (2004) 2691–2694, http://dx.doi.org/10.1109/TGRS.2004.840720.

[9] P. Potin, O. Colin, M. Pinheiro, B. Rosich, A. O'Connell, T. Ormston, J.-B. Gratadour, R. Torres, Status and evolution of the Sentinel-1 mission, in: IEEE International Geoscience and Remote Sensing Symposium, IGARSS, IEEE, 2022, pp. 4707–4710, http://dx.doi.org/10.1109/IGARSS46834.2022.9884753.

[10] N. Efremova, M.E.A. Seddik, E. Erten, Soil moisture estimation using Sentinel-1/-2 imagery coupled with cycleGAN for time-series gap filing, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–11, http://dx.doi.org/10.1109/TGRS.2021.3134127.

[11] J.-H. Choi, J.-S. Lee, EmbraceNet: A robust deep learning architecture for multimodal classification, Inf. Fusion 51 (2019) 259–270, http://dx.doi.org/10.1016/j.inffus.2019.02.010.

[12] F. Mena, D. Arenas, M. Charfuelan, M. Nuske, A. Dengel, Impact assessment of missing data in model predictions for Earth observation applications, in: IEEE International Geoscience and Remote Sensing Symposium, IGARSS, 2024, pp. 967–971, http://dx.doi.org/10.1109/IGARSS53475.2024.10640375.

[13] M. Ma, J. Ren, L. Zhao, D. Testuggine, X. Peng, Are multimodal transformers robust to missing modality? in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 18177–18186, http://dx.doi.org/10.1109/CVPR52688.2022.01764.

[14] G. Tseng, I. Zvonkov, M. Purohit, D. Rolnick, H. Kerner, Lightweight, pre-trained transformers for remote sensing timeseries, 2023, arXiv preprint arXiv:2304.14065.

[15] Y. Chen, M. Zhao, L. Bruzzone, A novel approach to incomplete multimodal learning for remote sensing data fusion, IEEE Trans. Geosci. Remote Sens. (2024) http://dx.doi.org/10.1109/TGRS.2024.3387837.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2018, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423.

[17] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, Y. Li, BRITS: Bidirectional recurrent imputation for time series, Adv. Neural Inf. Process. Syst. (NIPS) 31 (2018).

[18] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 16000–16009, http://dx.doi.org/10.1109/CVPR52688.2022.01553.

[19] Y. Cong, S. Khanna, C. Meng, P. Liu, E. Rozi, Y. He, M. Burke, D. Lobell, S. Ermon, SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery, Adv. Neural Inf. Process. Syst. 35 (2022) 197–211.

[20] Y. Yuan, L. Lin, Q. Liu, R. Hang, Z.-G. Zhou, SITS-Former: A pre-trained spatio-spectral-temporal representation model for Sentinel-2 time series classification, Int. J. Appl. Earth Obs. Geoinf. 106 (2022) 102651, http://dx.doi.org/10.1016/j.jag.2021.102651.

[21] G. Astruc, N. Gonthier, C. Mallet, L. Landrieu, OmniSAT: Self-supervised modality fusion for Earth observation, in: European Conference on Computer Vision, Springer, 2025, pp. 409–427, http://dx.doi.org/10.1007/978-3-031-73390-1_24.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[23] Y. Wang, C.M. Albrecht, X.X. Zhu, Self-supervised vision transformers for joint SAR-optical representation learning, in: IEEE International Geoscience and Remote Sensing Symposium, IGARSS, 2022, pp. 139–142, http://dx.doi.org/10.1109/IGARSS46834.2022.9883983.

[24] B. Ekim, M. Schmitt, Deep occlusion framework for multimodal Earth observation data, IEEE Geosci. Remote. Sens. Lett. (2024) http://dx.doi.org/10.1109/LGRS.2024.3460812.

[25] F. Mena, D. Arenas, A. Dengel, Increasing the robustness of model predictions to missing sensors in earth observation, 2024, arXiv preprint arXiv:2407.15512.

[26] G. Xu, X. Jiang, Y. Zhou, J. Fu, Z. Huang, X. Liu, Transformer-based incomplete multi-modal learning for land cover classification, in: IEEE International Geoscience and Remote Sensing Symposium, IGARSS, IEEE, 2024, pp. 7276–7281, http://dx.doi.org/10.1109/IGARSS53475.2024.10641815.

[27] J. Gawlikowski, S. Saha, J. Niebling, X.X. Zhu, Handling unexpected inputs: Incorporating source-wise out-of-distribution detection into SAR-optical data fusion for scene classification, EURASIP J. Adv. Signal Process. 2023 (1) (2023) 47, http://dx.doi.org/10.1186/s13634-023-01008-z.

[28] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, Y.W. Teh, Set Transformer: A framework for attention-based permutation-invariant neural networks, in: International Conference on Machine Learning, ICML, PMLR, 2019, pp. 3744–3753.

[29] M. Bugueño, M. Mendoza, Learning to combine classifiers outputs with the Transformer for text classification, Intell. Data Anal. 24 (S1) (2020) 15–41, http://dx.doi.org/10.3233/IDA-200007.

[30] C. Wang, X. Liu, J. Pei, Y. Huang, Y. Zhang, J. Yang, Multiview attention CNN-LSTM network for SAR automatic target recognition, IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 14 (2021) 12504–12513, http://dx.doi.org/10.1109/JSTARS.2021.3130582.

[31] X. Guo, J. Lao, B. Dang, Y. Zhang, L. Yu, L. Ru, L. Zhong, Z. Huang, K. Wu, D. Hu, et al., SkySense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 27672–27683.

[32] Z. Zheng, A. Ma, L. Zhang, Y. Zhong, Deep multisensor learning for missing-modality all-weather mapping, ISPRS J. Photogramm. Remote Sens. 174 (2021) 254–264, http://dx.doi.org/10.1016/j.isprsjprs.2020.12.009.

[33] G. Tseng, I. Zvonkov, C.L. Nakalembe, H. Kerner, CropHarvest: A global dataset for crop-type classification, in: Proceedings of NIPS Datasets Benchmarks Track, 2021-08-19.

[34] A. Francis, Sensor independent cloud and shadow masking with partial labels and multimodal inputs, IEEE Trans. Geosci. Remote Sens. (2024) http://dx.doi.org/10.1109/TGRS.2024.3391625.

[35] G. Camps-Valls, D. Tuia, X.X. Zhu, M. Reichstein, Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science and Geosciences, John Wiley & Sons, New York, 2021.

[36] N. Kussul, M. Lavreniuk, S. Skakun, A. Shelestov, Deep learning classification of land cover and crop types using remote sensing data, IEEE Geosci. Remote. Sens. Lett. 14 (5) (2017) 778–782, http://dx.doi.org/10.1109/LGRS.2017.2681128.

[37] P. Ghamisi, B. Höfle, X.X. Zhu, Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network, IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 10 (6) (2016) 3011–3024, http://dx.doi.org/10.1109/JSTARS.2016.2634863.

[38] F. Ferrari, M.P. Ferreira, C.A. Almeida, R.Q. Feitosa, Fusing Sentinel-1 and Sentinel-2 images for deforestation detection in the Brazilian Amazon under diverse cloud conditions, IEEE Geosci. Remote. Sens. Lett. 20 (2023) 1–5, http://dx.doi.org/10.1109/LGRS.2023.3242430.

[39] F. Mena, D. Pathak, H. Najjar, C. Sanchez, P. Helber, B. Bischke, P. Habelitz, M. Miranda, J. Siddamsetty, M. Nuske, et al., Adaptive fusion of multi-modal remote sensing data for optimal sub-field crop yield prediction, Remote Sens. Environ. 318 (2025) http://dx.doi.org/10.1016/j.rse.2024.114547.

[40] N. Audebert, B. Le Saux, S. Lefèvre, Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks, ISPRS J. Photogramm. Remote Sens. 140 (2018-06-01) 20–32, http://dx.doi.org/10.1016/j.isprsjprs.2017.11.011.

[41] P. Zhang, P. Du, C. Lin, X. Wang, E. Li, Z. Xue, X. Bai, A hybrid attention-aware fusion network (HAFNet) for building extraction from high-resolution imagery and LiDAR data, Remote. Sens. 12 (22) (2020-01) http://dx.doi.org/10.3390/rs12223764.

[42] S. Ofori-Ampofo, C. Pelletier, S. Lang, Crop type mapping from optical and radar time series using attention-based deep learning, Remote. Sens. 13 (22) (2021-01) http://dx.doi.org/10.3390/rs13224668.

[43] L. Fasnacht, P. Renard, P. Brunner, Robust input layer for neural networks for hyperspectral classification of data with missing bands, Appl. Comput. Geosci. 8 (2020) 100034, http://dx.doi.org/10.1016/j.acags.2020.100034.

[44] P. Ebel, V.S.F. Garnot, M. Schmitt, J.D. Wegner, X.X. Zhu, UnCRtainTS: Uncertainty quantification for cloud removal in optical satellite time series, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 2085–2095, http://dx.doi.org/10.1109/CVPRW59228.2023.00202.

[45] J. Inglada, A. Vincent, M. Arias, C. Marais-Sicre, Improved early crop type identification by joint use of high temporal resolution SAR and optical image time series, Remote. Sens. 8 (5) (2016) 362, http://dx.doi.org/10.3390/rs8050362.

[46] X. Che, H.K. Zhang, Z.B. Li, Y. Wang, Q. Sun, D. Luo, H. Wang, Linearly interpolating missing values in time series helps little for land cover classification using recurrent or attention networks, ISPRS J. Photogramm. Remote Sens. 212 (2024) 73–95, http://dx.doi.org/10.1016/j.isprsjprs.2024.04.021.

[47] W. Du, D. Côté, Y. Liu, SAITS: Self-attention-based imputation for time series, Expert Syst. Appl. 219 (2023) 119619, http://dx.doi.org/10.1016/j.eswa.2023.119619.

[48] X. Bouthillier, K. Konda, P. Vincent, R. Memisevic, Dropout as data augmentation, 2015, arXiv preprint arXiv:1506.08700.

[49] J.M. Haut, M.E. Paoletti, J. Plaza, A. Plaza, J. Li, Hyperspectral image classification using random occlusion data augmentation, IEEE Geosci. Remote. Sens. Lett. 16 (11) (2019) 1751–1755, http://dx.doi.org/10.1109/LGRS.2019.2909495.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. (NIPS) 30 (2017).

[51] K. Rao, A.P. Williams, J.F. Flefil, A.G. Konings, SAR-enhanced mapping of live fuel moisture content, Remote Sens. Environ. 245 (2020) 111797, http://dx.doi.org/10.1016/j.rse.2020.111797.

[52] S. Chen, PM2.5 data of five Chinese cities, 2017, http://dx.doi.org/10.24432/C52K58, UCI Machine Learning Repository.

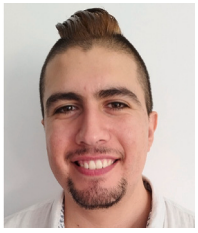
[53] H. Najjar, M. Nuske, A. Dengel, Data-centric machine learning for Earth observation: Necessary and sufficient features, 2024, arXiv preprint arXiv:2408.11384.

[54] D.P. Kingma, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, ICLR, 2015.

[55] F. Mena, D. Arenas, M. Nuske, A. Dengel, A comparative assessment of multi-view fusion learning for crop classification, in: IEEE International Geoscience and Remote Sensing Symposium, IGARSS, IEEE, 2023, pp. 5631–5634, http://dx.doi.org/10.1109/IGARSS52108.2023.10282138.

[56] R. Desislavov, F. Martínez-Plumed, J. Hernández-Orallo, Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning, Sustain. Comput.: Inform. Syst. 38 (2023) 100857, http://dx.doi.org/10.1016/j.suscom.2023.100857.

[57] R. Heinrich, C. Scholz, S. Vogt, M. Lehna, Targeted adversarial attacks on wind power forecasts, Mach. Learn. (2023) http://dx.doi.org/10.1007/s10994-023-06396-9.



Francisco Mena received the master's and bachelor's degree in computer engineering from the Federico Santa María Technical University, Chile, in 2020. He is currently pursuing a Ph.D. in computer science at the University of Kaiserslautern-Landau and researching at the German Research Center for Artificial Intelligence (DFKI), under the supervision of Prof. Andreas Dengel. The Ph.D. topic involves multi-view learning with missing views in the Earth observation domain. His research interests include deep neural networks, multi-view or multi-modal learning, data fusion, representation learning, robustness, and Earth observation applications.

Diego Arenas holds a bachelor's degree in computer science from the University of Talca, Chile (2007). He worked in information systems as a consultant and project manager for 10 years in finance, banking, retail, education, among others. He holds an M.Sc. in data science from the University of Edinburgh, Scotland (2016) and an EngD in computer science from the University of St. Andrews, Scotland (2021). His interests are in anti-corruption, biodiversity, AI4Good, food security, and remote sensing. Since 2022, he works at the German Research Center for Artificial Intelligence (DFKI) at the intersection of artificial intelligence and Earth observation.

Andreas Dengel received a diploma in computer science from the University of Kaiserslautern and a Ph.D. from the University of Stuttgart. He is a Scientific Director at the German Research Center for Artificial Intelligence (DFKI), a Professor at the University of Kaiserslautern (1993), and Kyakuin at Osaka Prefecture University (2009). Formerly with IBM, Siemens, and Xerox Parc. He has authored and supervised more than 300 publications and 170 theses. Moreover, he has co-edited international computer science journals and has edited 12 books. His research focuses on pattern recognition, document understanding, information retrieval, multimedia mining, semantic technologies, and social media.