WATCH: Wide-Area Terrestrial Change Hypercube

Connor Greenwell[1]* Jon Crall[1] Matthew Purri[2] Kristin Dana[2]
Nathan Jacobs[3] Armin Hadzic[4] Scott Workman[4] Matt Leotta[1]

[1]Kitware, Inc. [2]Rutgers University [3]Washington University in St. Louis [4]DZYNE Technologies

Abstract

Monitoring Earth activity using data collected from multiple satellite imaging platforms in a unified way is a significant challenge, especially with large variability in image resolution, spectral bands, and revisit rates. Further, the availability of sensor data varies across time as new platforms are launched. In this work, we introduce an adaptable framework and network architecture capable of predicting on subsets of the available platforms, bands, or temporal ranges it was trained on. Our system, called WATCH, is highly general and can be applied to a variety of geospatial tasks. In this work, we analyze the performance of WATCH using the recent IARPA SMART public dataset and metrics. We focus primarily on the problem of broad area search for heavy construction sites. Experiments validate the robustness of WATCH during inference to limited sensor availability, as well the the ability to alter inference-time spatial or temporal sampling. WATCH is open source and available for use on this or other remote sensing problems. Code and model weights are available at: https://gitlab.kitware.com/computer-vision/geowatch

1. Introduction

Satellite imagery from a variety of commercial and government sources has become increasingly available, imaging the Earth's land surface at multiple resolutions, spectral bands, and various revisit rates with data collection spanning many years. The availability of such data opens up new opportunities in Earth monitoring, where algorithms can be trained to search through this enormous volume of data to detect specific man-made or natural activities and characterize the progression of those activities over time. Examples include monitoring deforestation, measuring destruction from natural disasters or military conflicts, and detecting construction or agricultural land use changes.

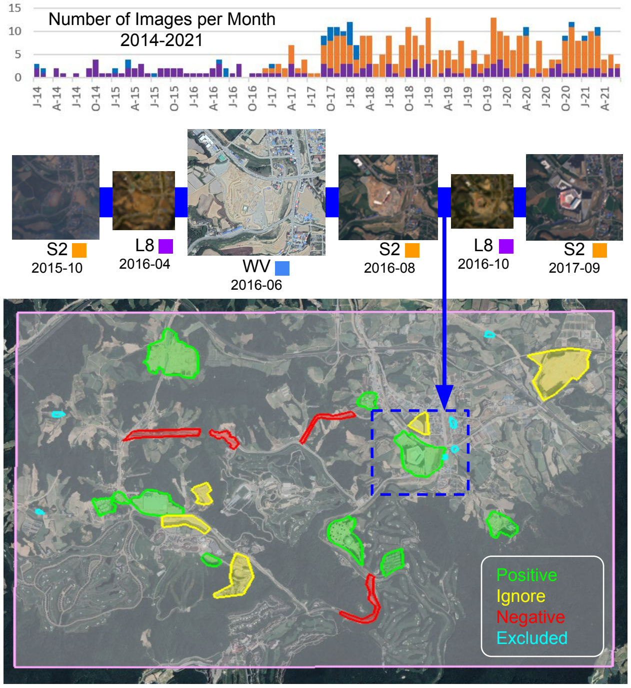

Figure 1. **Illustration of the IARPA SMART construction site BAS problem and associated data.** IARPA SMART [13] represents a challenging task as input images are sampled irregularly across time and from a variety of possible sensors (top), and the targeted construction events are sparse in both space and temporal duration (bottom). Example construction sites vary significantly in appearance both across sensors and as time progresses (middle).

Much of the current research in similar remote sensing tasks is focused on training networks to operate on data from a particular satellite platform, such as Landsat 8, Sentinel-2, WorldView, or Planet Dove. Individual satellite platforms have unique characteristics in terms of temporal collection, spatial coverage, and spectral sensitivity. For example, Landsat 8&9 provide regular 16-day revisits but the highest resolution color bands are 30m resolution. Sentinel-2 provides several 10m resolution bands and more frequent 10-day revisits, but only provides coverage from 2016 onward. Mean-

*Corresponding author: connor.greenwell@kitware.com while, WorldView 2&3 can provide sub-meter resolution, but with irregular and far less frequent revisits.

The advantage of working with a single platform (or family of related platforms) is that the data is fairly homogeneous. Each image in a time sequence has a similar ground sampling distance (GSD), spectral bands, and calibration, which is ideal for temporal analysis. However, using only imagery collected from a single platform limits temporal coverage (Fig. 1). As an alternative, the inclusion of multiple sources of data can provide complementary coverage for regions of interest.

In this work, we build a system to address the challenges posed in the IARPA SMART dataset and metrics [13]. This dataset accompanies the IARPA Space-Based Machine Automated Recognition Technique (SMART) program, which seeks to automatically detect, characterize, and monitor anthropogenic activity at a global scale using time-series imagery from multiple sensors. The SMART dataset includes a dozen city-sized regions distributed around the globe that contain heavy construction sites spanning from 2014–2021, represented as temporal sequences of geo-spatial polygons. In SMART terms, identifying the locations of these heavy construction sites is referred to as broad area search (BAS), while identifying the phase labels across time is called activity characterization (AC). Fig. 1 bottom shows an example of a small region with BAS annotations at one point in time.

Our system to address the SMART challenge is called *WATCH* (Wide-Area Terrestrial Change Hypercube). WATCH contains a general framework and network architecture to solve both the BAS and AC problems using a variety of imagery sources and derived features from those sources. We limit the scope of this paper to the BAS task, to freely available imagery sources (Landsat 8 and Sentinel-2), and to a couple of derived features (Sec. 3.3.1). Landsat 8 and Sentinel-2 have sufficient variability in spatial resolution (30m and 10m, respectively) and temporal coverage (See Fig. 1, top) to demonstrate the value of our approach. WATCH uses a vision transformer with tokenization and positional encoding separately across space, time, spectra, and platform which allows our network to be trained and evaluated on time series data from a mixture of platforms. We formulate BAS as a binary classification task where our method predicts per-image, per-pixel heatmaps representing the likelihood of a construction site being present.

The WATCH framework, described in Sec. 3, is designed to be robust at evaluation time to the availability of the sensors upon which it was trained. This was implemented with real-world situations in mind, including processing historical data where only a subset of the trained sensors had yet to be deployed, when sensor readings are sporadically unavailable due to inclement weather or technical malfunction, and limited data availability due to cost or licensing. Our experimental analysis in Sec. 4 demonstrates the value of training a model to combine image sources as well the impact of limiting the input imagery sources during inference.

2. Related Work

The increasing spatial and temporal availability of satellite imagery presents a significant opportunity for large-scale Earth monitoring. A fundamental task in this area is the identification of landscape changes, either anthropogenic or natural, directly from remotely sensed imagery captured at different times (often referred to as change detection). A large body of work has explored methods for solving this task using time-series analysis [33–35] and more recently, deep learning-based approaches have become standard practice [18, 23].

The canonical problem formulation for change detection is the comparison of two points in time, often on the order of years apart, using images sourced from the same sensor [5, 11, 12, 19]. This existing work has typically been limited to open data sources (e.g., Landsat) with well-defined calibration and pre-processing steps to ensure consistency. As Woodcock et al. [28] mention, supporting multi-sensor inputs "requires algorithms to account for differences between sensors that can complicate the analysis."

Discrepancies in visual characteristics across multiple sensors have traditionally been addressed by harmonizing the data sources before processing [3, 8, 14, 21]. The result is a merged product that is radiometrically, spectrally, and spatially consistent across sensors. The downside is that only common spectral bands can be aligned and the imagery is first downsampled to a common resolution. Other approaches ignore fusion and instead develop modality-specific sub-modules [15,22]. To support the recent shift towards continuous Earth monitoring, novel approaches for multi-sensor fusion are required [27].

Recently in traditional vision tasks, transformers networks [25] have arisen as a powerful way for combining information across modalities. In remote sensing, transformers have proven to be an effective way to learn feature representations from multi-spectral and temporal satellite data [9]. However, for the problem of monitoring and detecting change in remote sensing data, these approaches are generally concerned with pairs of images separated by some number of years, and propose using an intermediate network trained to blend information extracted from pairs of input frames [4, 6, 20, 32]. Taking inspiration from video transformer networks [24], we develop a framework capable of fusing information across space, time, spectra, and platform.

3. WATCH Overview

In this section, we present an overview of our proposed WATCH system, which is purpose-built to make multi-task predictions from geo-temporal input imagery sourced from

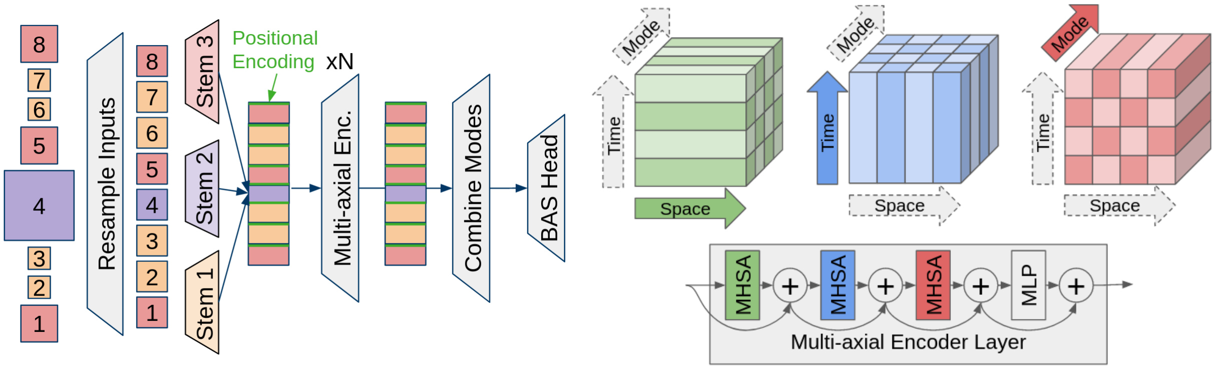

Figure 2. **Architectural overview.** To construct an input batch, we sample a spatial region from multiple input images, which may be from different sensors or contain different bands, resampled to a common resolution. Inputs are grouped by sensor / modality and passed through a shallow tokenizing network that normalizes the feature dimension and adds positional embeddings representing space, time, sensor, and mode. Tokens are processed by a sequence of multi-axial transformer encoders (right), then pooled across modalities to construct space/time features, after which a final BAS prediction is made with an MLP head. The multi-axial transformer encoder applies multiple steps of multi-head self-attention (MHSA) where tokens are first grouped such that attention is only applied to tokens that vary on a specific axis.

multiple sensor platforms. First, we detail our approach for ingesting and tokenizing heterogeneous input imagery (Sec. 3.1). Next, we provide an overview of the transformer encoder that we use as the backbone of our model, including the axis-separated attention scheme used and task-specific heads of our model (Sec. 3.2). Finally, we cover relevant implementation details, including multi-task losses, and more (Sec. 3.3). For a visual overview of our proposed architecture, please refer to Fig. 2.

3.1. Ingesting Heterogeneous Input Imagery

Our proposed system is designed to be flexible with respect to the types, arrangement, and distribution of inputs it makes predictions on. In particular, WATCH was built to handle geo-referenced multi-spectral imagery sourced from multiple satellite imaging platforms where each has a unique combination of spectra, resolutions, revisit rates, etc. In addition to multi-spectral imagery sources, our system is designed to ingest other geo-spatial features, including those estimated by other machine learning models, *e.g.* self-supervised learning features, land cover categories [29], material characteristics [1], *etc.*

We first convert input images into sequences of tokens, with each token representing a windowed view of the input image, along with encoded metadata about that token. We employ a standard approach to tokenizing where regularly spaced patches, $w \in \mathbb{R}^{H \times W \times C}$, are extracted from each image I, flattened, and projected to the target token dimension, $t \in \mathbb{R}^T$, via a shallow MLP, $f(w; \Theta) \mapsto \mathbb{R}^F$, to produce the final token sequence representing the input image $\{t_i = f(w_i, \Theta), \forall w_i \in I\}$.

Modality-Specific Tokenization Our model is designed to take inputs from a variety of sensors, specific treatments of which we refer to as an *image modality*. Each input modality differs in terms of which and how many spectra (bands) are contained in each image, in addition to other differences including sensor type, resolution, viewing angle, revisit rate, and more. Additionally, other geo-spatial features are treated as separate modalities. Due to the fundamental differences between these modes, we train a separate tokenizing model for each, with a distinct set of weights Θ_m, which translates the visual or semantic content of each input to the shared token space $t \in \mathbb{R}^T$. We also resample input images to have a shared GSD and produce the same number of tokens per modality; lower resolution inputs are upsampled.

Spatio-temporal Positional Encoding Each input token is enriched with positional encodings representing xy-spatial position, the temporal index, the temporal offset relative to the first timestep, the sensor type, and the input band combination. The spatial and temporal indexes use sinusoidal encodings [25], whereas the others are learned using 3 layer multi-layer perceptrons (MLP), each with a different input shape. The time delta from the first timestep in seconds is given directly as input to its MLP. For the sensor and channels, we hash representative strings and convert those bytes to tensors, which are the inputs to the encoding MLP. Each token is the concatenation of the reprojected image patch and these 6 embedding vectors: $t = \text{concat}(f(w_i, \Theta), e_{\text{xy}}, e_{\text{time}}, e_{\text{index}}, e_{\text{sensor}}, e_{\text{bands}})$.

3.2. Transformer Encoder Backbone

The backbone of our approach is a vision transformer which takes a sequence of multi-modal geo-spatial tokens as inputs and makes predictions for one or more tasks at each input spatio-temporal location. First, the input tokens are processed by our transformer model to predict a feature representation corresponding with each input token. To improve model efficiency, tokens are attended to spatially, temporally, and then across modalities. Next, we marginalize modalities away through a max pooling step to predict a token at each point in space-time. Finally, these tokens are passed to a number of task-specific heads to make our final predictions.

Multi-axial Attention Scheme Input sequences to this model can be particularly long, especially when processing large numbers of time steps. Following work on transformer video segmentation [2], we apply attention on a per-axis basis within each layer of our transformer backbone. Each input sequence has four axes to consider: height, width, time, and modality. As shown on the right of Fig. 2, we group height and width together, and apply attention spatially, temporally, then across modalities. This reduces the computational overhead of attention from $O(HWTM)$ to $O(HW + T + M)$, a significant savings.

Task-specific Output Heads We do not use a decoder in our transformer. Instead we build a spatio-temporal feature vector by max pooling the encoded tokens over modalities. Each task head is an MLP that uses this feature as its input.

3.3. Implementation Details

We pretrain our model in a self-supervised manner, described below. We finetune this pretrained model using the IARPA SMART dataset, during which we optimize the focal loss [17] for binary segmentation. WATCH is a 2M parameter model implemented in PyTorch and was trained on a single NVIDIA Quadro RTX 6000 GPU for 80k iterations. The model was fine-tuned with a batch size of 6 using the AdamW optimizer and a OneCycle learning rate schedule with a $4k$ step learning rate warmup from $3e^{-6}$ to $3e^{-4}$, annealing down to $5e^{-9}$.

***ad hoc* Curriculum Learning** As this project evolved and additional sources of imagery, labels, and imagery-derived features became available we found it useful to iteratively finetune from the previous best version of WATCH. At different times we varied the number of input channels for each sensor (S2, L8), typically RGB, RGB+NIR, or full spectrum, a choice we ensured was always consistent across sensors. Making this change necessitated resetting the weights of the tokenizing heads while keeping the main transformer weights from the previous run, and we hypothesize that doing so may have acted as a form of regularization.

3.3.1 Additional Feature Inputs

WATCH is designed to take input from multiple sensor platforms and modalities, including specialized features derived from input imagery. Here we present two such derived features: landcover/land-use categories, and a self-supervised (SSL) feature, both computed on Sentinel-2 imagery.

Table 1. **IARPA SMART - Observations per sensor over time.** Sentinel-2A and -2B came online in late 2016 and 2017 respectively, and thus there are no/few S2 observations before those dates.

	2014	2015	2016	2017	2018	2019	2020	2021
L8	262	236	261	385	462	444	414	298
S2	0	0	21	813	1549	1555	1545	626
TOT	262	236	282	1198	2011	1999	1959	924

Semantic Landcover Features We extract semantic features from a given image in the form of a per-pixel land cover labeling. For our segmentation network, we use a variant of UNet with a ResNet-18 backbone as in [29]. For the experiments, we train our network using Sentinel-2 composite images. Specifically, we created a global dataset comprising monthly cloud-free composite images over approximately 300 metropolitan areas. As the source of supervision, we align images with land cover annotations from ESA WorldCover [31].

Self-Supervised Learning Features We use a multi-task self-supervised learning (SSL) approach to learn to extract useful features for change detection. We use three tasks: augmentation invariance, temporal invariance, and a novel pixel-wise temporal ordering task. We use an attention-based U-Net architecture as the backbone to recognize features that indicate the progression of time in pairs of satellite images. Image pairs are given to the network in random order and predictions are made on a pixel-level basis. The use of pixel-level predictions allows the network to focus on fine-grained detail such as building construction instead of providing an overall estimate based on aggregated features. Along with the pixel level order prediction, we use also use two other tasks, 1) contrastive training with augmented images and 2) contrastive training with spatially non-aligned images [7, 16]. Overall, we introduce a multi-task learning framework with a novel pretext task that learns spatio-temporally rich features for change detection. For the experiments, the SSL networks are trained using Sentinel-2 imagery. The SSL networks are frozen during the downstream task training.

3.3.2 Time Sampling Scheme

Given a single frame and spatial location we sample additional frames to loosely match a requested distance between frames. Because we do not control the revisit rate or number of images per-year per-sensor (Tab. 1), we first construct an idealized sampling distribution which is a mixture of gaussians centered over the requested spacing. At train time this is chosen randomly according to this distribution. At test time we greedily choose the observation with maximum probability. This process is illustrated in Fig. 3. For the ex-

Table 2. **Per-region metric breakdown on SMART.** We present metrics for each of the labeled regions in SMART, for the three WATCH models we produced: one of which was trained following our *ad hoc* strategy (Sec. 3.3), one which was trained *from scratch* using all available L8/S2 imagery and derived features, and one trained *from scratch, on only L8/S2 imagery*. We also present the mean over all regions (OVR) for each model/metric. The best OVR results are printed in **bold**, second-best are underlined. All scores are expressed as percentages.

	Region	AE	BH	BR				CH	KR		LT	NZ	PE	US					OVR
				1	2	4	5		1	2				1	4	5	6	7	
ad hoc	F1	62.6	71.0	39.2	66.7	23.1	31.2	52.4	52.2	45.6	66.7	42.3	0.0	51.4	58.7	42.1	20.0	7.6	**43.1**
	FFPA	5.9	11.5	5.5	0.5	5.4	1.6	1.4	1.6	0.6	1.7	2.4	0.4	1.0	0.7	1.4	0.6	1.1	2.5
	AP	50.3	28.8	78.6	56.7	82.1	48.7	44.8	28.6	28.4	50.3	38.2	3.3	61.2	53.1	74.0	11.9	19.6	**44.6**
	AUC	98.3	96.4	97.9	99.7	99.1	99.2	97.9	96.6	91.8	98.4	97.9	97.8	98.8	98.7	99.4	99.0	99.7	**98.0**
scratch	F1	52.4	48.0	31.0	50.0	15.2	26.7	50.0	58.8	41.4	46.6	35.1	0.0	49.4	45.6	38.1	11.3	12.3	36.0
	FFPA	5.9	6.1	4.3	0.6	5.4	1.8	0.9	0.3	0.6	1.5	4.2	0.0	1.2	1.7	3.7	2.7	2.0	2.5
	AP	13.5	41.7	73.8	39.3	66.1	20.9	35.5	27.9	14.8	37.1	50.5	12.1	44.6	18.5	57.1	1.9	7.3	33.1
	AUC	89.4	96.0	97.5	99.6	98.0	98.4	97.1	96.3	90.4	97.3	97.9	98.4	96.8	96.8	99.0	98.1	98.2	96.8
scr. S2/L8	F1	5.2	49.3	39.3	75.0	14.3	18.7	44.2	50.0	50.0	43.5	23.1	0.0	29.0	19.3	38.7	6.6	12.3	30.5
	FFPA	0.6	8.7	3.4	0.2	4.6	4.1	1.2	2.0	2.4	1.5	2.0	0.2	1.0	1.9	2.1	4.2	2.3	2.5
	AP	4.7	20.4	47.0	22.2	19.4	7.2	31.4	11.6	24.3	26.3	14.0	1.1	10.6	6.2	40.6	0.6	1.6	17.0
	AUC	74.3	90.5	92.3	98.7	89.1	96.4	95.7	93.3	90.1	95.6	95.0	93.6	92.7	90.5	98.6	92.9	96.9	92.7

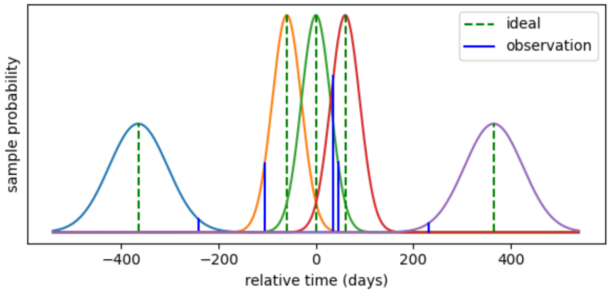

Figure 3. **Time sampling strategy.** Given a central frame we sample additional frames according to a chosen idealized spacing (shown in green). At train time we sample available observations according to a distribution centered at each ideal offset. At test time, we choose the observation that maximizes probability for each offset.

periments in this paper we choose a uniform time span 11 frames distributed ±3 years.

4. Experiments

We evaluate the predictive ability of our proposed WATCH model. Overall, we find that WATCH is flexible and performs well despite variations in the input it is provided. Our experimentation is focused on the recently released IARPA SMART heavy construction dataset (Sec. 4.1). When making predictions on a restricted set of sensors, performance degrades gracefully (Sec. 4.2). At prediction time, increasing the number of time steps provided to the model increases performance up to a point (Sec. 4.3). Next, we evaluate WATCH for its applicability and adaptability to the Onera Satellite Change Detection dataset (Sec. 4.4). Finally, we present some of our findings on the impact that various design decisions have on the models final performance (Sec. 4.5).

4.1. IARPA SMART Dataset

We train and evaluate our models using the IARPA SMART dataset [13], which seeks to automatically detect, characterize, and monitor anthropogenic activity at a global scale using time-series imagery from multiple sensors. The SMART dataset includes a dozen city-sized regions distributed around the globe that contain heavy construction sites, with each region containing up to 200 identified sites spanning from 2014–2022. Annotations identify both the spatial and temporal bounds of each site and are represented as temporal sequences of geo-spatial polygons. We formulate BAS as a binary classification task and our method outputs per-image, per-pixel heatmaps representing the likelihood of a construction site. Individual heatmaps are aggregated across time to generate site predictions.

As input to our method, we use publicly available Landsat 8 and Sentinel-2 imagery. While Landsat 8 covers the full temporal extent of the SMART dataset, it is comparatively low resolution, i.e. the smallest sites are around $8000m^2$ or 9x9 pixels. To increase both the temporal frequency and available detail for the 2016–2022 period we include Sentinel-2 imagery, which has a higher revisit rate than Landsat 8 and approx. 9x resolution. For a full breakdown, see Tab. 1.

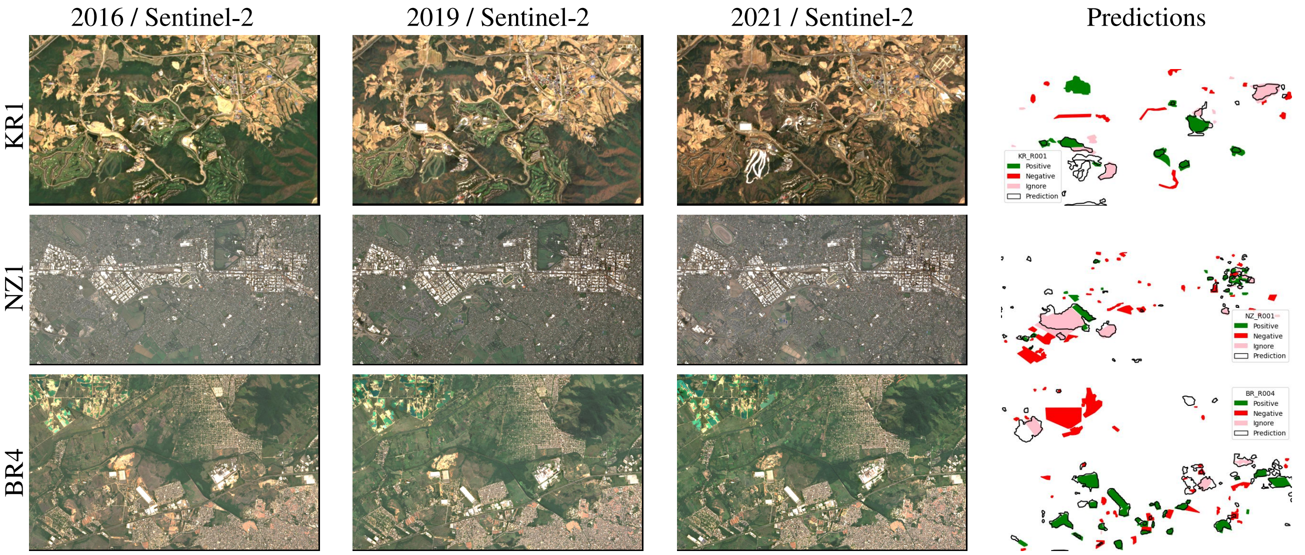

Figure 4. **Sample WATCH inputs and outputs.** Examples of our *ad hoc* WATCH models site predictions are shown as black contours for three IARPA SMART regions, as well as three time-averaged Sentinel-2 images used as input. Green filled regions are annotated as true construction, red are annotated explicit negatives, pink are ambiguous regions ignored in scoring. Predictions best viewed zoomed in.

Table 3. **Prediction with restricted sensors.** We report test-set polygon- and pixel-based metrics for our *ad hoc*-trained SMART site detection (BAS) model evaluated with varying sensor availability. We also compare two baseline models, one trained from *scratch* using all available data (S2, L8, and extra features), the other also from scratch but trained on *S2/L8 only*. Due to the SMART dataset including years that precede the Sentinel-2 launches, we compare modality subsets with and without Landsat 8 separately. Similarly, since the SSL and Landcover features are based on Sentinel-2 imagery, we only consider them when the source is available. Best results in each column (with and without Landsat 8) are printed in **bold**, second-best are underlined. All scores are expressed as percentages.

				ad hoc				*scratch*				*scratch, S2/L8 only*			
L8	S2	SSL	LC	F1	FFPA	AP	AUC	F1	FFPA	AP	AUC	F1	FFPA	AP	AUC
✓	✓	✓	✓	47.25	2.07	46.97	96.83	**44.50**	1.88	**41.07**	**95.99**	(invalid)			
✓	✓	✓		**48.15**	1.25	**49.07**	**97.03**	31.02	**1.57**	18.09	89.24				
✓	✓		✓	46.02	1.84	41.15	95.58	41.18	3.04	32.19	94.90				
✓	✓			43.22	**1.00**	35.94	94.57	37.16	1.76	21.26	92.07	**38.19**	**2.02**	**22.83**	**93.09**
✓				37.59	1.12	31.56	94.10	35.80	1.78	21.13	91.62	34.73	2.57	20.58	91.06
	✓	✓	✓	34.63	2.64	38.16	95.29	**40.01**	1.10	**38.10**	**94.86**	(invalid)			
	✓	✓		**41.28**	1.95	**40.30**	**95.33**	18.37	1.23	5.40	76.12				
	✓		✓	37.92	2.53	31.33	94.17	36.80	1.46	26.66	93.02				
	✓			33.96	**1.50**	25.32	91.44	34.78	**0.86**	12.73	85.78	33.38	0.79	14.63	88.26

A common method for removing nuisance information, i.e. cloud, shadows, and specularities, in optical remote sensing imagery is to merge imagery within a temporal window using mean or median operations [30]. Given a sequence of images for a region $X = x_0 \dots x_N$ and a temporal window duration t, images within the same temporal window are combined to form an composite image $x_{agg}^{t_k} = \varphi(x_j), j \subset t$. The sequence of composite images are then passed to the model during inference. For our experiments, we use a temporal window of $t = 1y$.

Polygon Estimation & Metrics The SMART evaluation framework [13] requires predictions in the form of a series of geo-spatial polygons in order to associate detected sites across images in a time series. To convert the output of our approach (per-image heatmaps) to polygons representing the spatial and temporal extents of a construction site we use a very simple process. We average predictions over the entire temporal range, threshold averaged heatmaps, trace contours to form polygons, and then connect spatially overlapping polygons across time.

For BAS evaluations we consider two classes of metrics: pixel-wise metrics and polygon metrics. For the pixel-wise metrics, we use two standard segmentation metrics applied to the aggregated heatmaps: average precision (AP), area under the receiver operating characteristic curve (AUC). We compute two additional metrics using the IARPA SMART evaluation framework: F1 Score, and Fractional False Positive Area (FFPA). FFPA corresponds to the portion of each search region covered by polygons which are false positives. For additional details on how these metrics are computed, please refer to [13].

We present our main findings in Tab. 2, where for each region in IARPA SMART we evaluate three different models for each of the metrics described above. We also present example outputs of the *ad hoc* model over three sample regions in Fig. 4. The three models we focus on are the *ad hoc* trained model, a *from scratch* model trained on all four input modalities, and a *from scratch* model trained only on Sentinel-2 and Landsat 8 imagery. What we found is that overall the *ad hoc* model performs best or tied for best on all four metrics, indicating the importance of additional training cycles and the curriculum learning aspect. Between the two *from scratch* models, the addition of features derived from Sentinel-2 imagery (LC and SSL) demonstrate a significant improvement in all metrics, especially F1.

4.2. Predicting with Sensor Restrictions

WATCH is designed to be flexible in terms of processing heterogeneous data during both training *and evaluation*. To analyze the ability of WATCH to make predictions given limited data availability, we simulate settings where entire sensors and/or upstream feature sources are unavailable during evaluation. This scenario could arise for many reasons, including imagery from a specific sensor being unavailable at a given location/time, or technical malfunctions.

In Tab. 3 we analyze site detection performance as various input sources are withheld, and we conduct this analysis across all three of the models from Tab. 2. The full-data setting, shown in the top row, is trained using Sentinel-2 (S2), Landsat 8 (L8), invariant features from self-supervised learning (SSL), and semantic landcover features (LC). The subsequent rows show how this model can be adapted at inference to support missing data modalities. Generally, performance degrades gracefully as modalities are withheld. For example, when SSL features are unavailable to the *scratch* model, F1 performances drops slightly from 44.50 to 41.18. Due to the nature of training the *ad hoc* model, we find it performs best without landcover features; we hypothesize that this is due the significantly larger number of training cycles that the model had with SSL features.

We also find that the inclusion of additional features during training makes the *scratch* model more robust to missing inputs than the *scratch with L8/S2 only* model. When considering the imagery only settings (from L8/S2 to L8-only and S2-only), *scratch* sees a -1.36 and -2.38 change in F1 respectively, while the *S2/L8 only* model sees -3.46 and -4.81. These results highlight a real-world benefit of our approach, training on a variety of input modalities, and robustness to those modalities being missing at inference time.

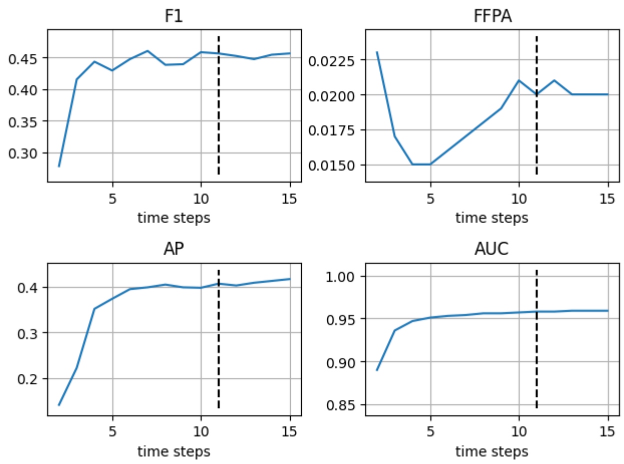

Figure 5. **SMART metrics with variable number of time steps.** We report polygon- and pixel-based metrics for our *ad hoc* WATCH model evaluated with a varying number of time steps. Times are sampled such that spacing between each step is approximately even given a fixed window of 6 years. Our model is trained with 11 time steps, indicated by the vertical dashed line in each plot.

4.3. Varying Time Steps

We report polygon- and pixel-based site detection metrics for the *ad hoc* model evaluated with varying numbers of provided time steps. For each setting, times are sampled such that spacing between each step is approximately even given a fixed window of ±3 years. In Fig. 5 we find that model performance is generally stable between 5 and 15 timesteps. Interestingly, despite the model being trained using 11 timesteps per example, FFPA gradually improves as the number of timesteps approaches 5 without negatively affecting F1 or pixel metrics.

4.4. Onera Satellite Change Detection Dataset

While not trained with adaptability in mind, WATCH is a flexible system trained for a version of the remote sensing change detection task. In this section, we evaluate WATCH's ability to detect change in other settings, both naively without and with finetuning. A commonly used benchmark in this space is the Onera Satellite Change Detection (OSCD) dataset [11], which features pairs of Sentinel-2 images sampled several years apart over 24 major cities from across the world. Accompanying these imagery pairs are change masks indicating change from timestep 1 to 2, including con-

Table 4. **Binary change detection results on OSCD.** We compare WATCH with and without finetuning against existing work on this dataset. All scores are computed per-pixel and expressed as %.

Method	3-chan. Acc./F1	13 Acc./F1
Siam. [10, 11]	76.76 / 33.85	85.37 / 37.69
EF [10, 11]	83.63 / 34.15	88.15 / 42.48
FC-EF [10]	94.23 / 48.89	96.05 / 56.91
FC-Siam-conc [10]	94.07 / 45.20	93.68 / 51.36
FC-Siam-diff [10]	94.86 / 48.86	95.68 / 57.92
DINO-MC RN-50 [26]	— / 52.46	—
WATCH (ours)	95.01 / 8.25	—
WATCH (ours), f.tuned	95.14 / 43.17	95.47 / 46.70

struction/demolition of buildings and roads, and major earth moving projects. The results of both of our OSCD experiments are presented in Tab. 4. A key difference in metrics is that OSCD is scored *pixel-wise*, *i.e.* each pixel contributes equally to the final scores. In contrast, SMART is scored such that each site contributes equally regardless of size.

Given that WATCH is trained to detect a subset of the changes covered by OSCD, we first apply WATCH naively with no finetuning. WATCH is a multi-task model, featuring a "saliency" head indicating the presence of a relevant construction phase in any given timestep. To predict change labels, we average the saliency prediction and select a threshold which maximizes F1 on the training split. We find that WATCH without finetuning performs reasonably well, with an accuracy of 95.01 and F1 of 8.25, which can be attributed to high precision but lower recall as WATCH was trained to identify only a subset of OSCD's change categories.

Next, we finetune WATCH for OSCD. We train WATCH for $10k$ steps, updating the *ad hoc* model following a $10k$ step OneCycle learning rate schedule with a $3k$ step learning rate warmup from $2e^{-6}$ to $5e^{-5}$, annealing down to $5e^{-9}$. Here we find that with a small amount of finetuning, WATCH's scores on OSCD increase to 95.14 Acc and 43.17 F1, showing that it can adapt and perform on par with all but the most recent OSCD approaches.

4.5. Ablation

Due to the quantity of tokens required to represent the full multi-modal spatio-temporal sequence of images, from the outset we sought more efficient transformer models and settled on a method inspired by video segmentation [2]. In Tab. 5, we show the results of a simple ablation study focused on the impact of using a baseline transformer model vs. our proposed "multi-axial" attention model, as well as the inclusion (or not) of additional input features. In order to make the largest of these models fit on a single GPU, we reduce the number of time steps to 7 and the batch size to 1, all other settings held the same and trained from scratch. We found that in general the baseline performed much worse and at a higher memory expense than the multi-axial attention model we use in the WATCH system. The reduced memory footprint of the multi-axial attention model enables processing a larger temporal sequence which provides a boost to the F1 polygon scores as well, from 39.44 F1 to 44.50.

Table 5. **Comparing baseline and multi-axial attention models on IARPA SMART.** Training with multi-axial attention significantly reduces the overall memory footprint of our model while also boosting performance on the F1 metric. "multi-axial*" indicates our proposed multi-axial attention approach presented in Tab. 3, specifically the top performing "scratch" and "scratch (L8/S2 only)" settings.

Model	Feats	Mem.	F1	FFPA	AP	AUC
baseline	✓	24GB	17.36	0.67	22.54	92.62
baseline		8GB	19.66	1.26	18.10	90.99
multi-axial	✓	8GB	39.44	1.25	48.56	96.63
multi-axial		4GB	38.09	1.42	39.59	96.10
multi-axial*	✓	24GB	44.50	1.88	41.07	95.99
multi-axial*		16GB	38.19	2.02	22.83	93.09

Resources When running our experiments we track the duration of each step. For heatmap prediction we use *codecarbon* (github.com/mlco2/codecarbon) to estimate the carbon footprint. Over the course of WATCH's development, there were more than 1200 unique heatmap prediction steps which took 7.25 days and 11.3 kg of CO_2. The polygon prediction and evaluation step had their duration but not their carbon footprint measured. There were 14,221 polygon prediction steps, which took 7.66 days. The corresponding evaluation step took 30.5 days.

5. Discussion and Conclusion

We introduced the WATCH system, which includes a flexible neural network architecture for broad-area change detection. Through extensive experiments on a large-scale evaluation dataset, we demonstrated that it is capable of exploiting images captured by different satellites, with different numbers of channels, spatial resolutions, and revisit times.

Acknowledgement We thank Benjamin Brodie for his contributions to this project. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2021-2011000005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] Peri Akiva, Matthew Purri, and Matthew Leotta. Self-supervised material and texture representation learning for remote sensing tasks. In *IEEE Computer Vision and Pattern Recognition*, 2022. 3

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021. 4, 8

[3] Douglas K Bolton, Josh M Gray, Eli K Melaas, Minkyu Moon, Lars Eklundh, and Mark A Friedl. Continental-scale land surface phenology from harmonized landsat 8 and sentinel-2 imagery. *Remote Sensing of Environment*, 240, 2020. 2

[4] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 2021. 2

[5] Hao Chen and Zhenwei Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sensing*, 12, 2020. 2

[6] Jie Chen, Ziyang Yuan, Jian Peng, Li Chen, Haozhe Huang, Jiawei Zhu, Yu Liu, and Haifeng Li. Dasnet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 2020. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020. 4

[8] Martin Claverie, Junchang Ju, Jeffrey G Masek, Jennifer L Dungan, Eric F Vermote, Jean-Claude Roger, Sergii V Skakun, and Christopher Justice. The harmonized landsat and sentinel-2 surface reflectance data set. *Remote Sensing of Environment*, 219, 2018. 2

[9] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In *Advances in Neural Information Processing Systems*, 2022. 2

[10] Rodrigo Caye Daudt, Bertr Le Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. In *IEEE International Conference on Image Processing*, 2018. 8

[11] Rodrigo Caye Daudt, Bertr Le Saux, Alexandre Boulch, and Yann Gousseau. Urban change detection for multispectral earth observation using convolutional neural networks. In *IEEE International Geoscience and Remote Sensing Symposium*, 2018. 2, 7, 8

[12] Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau. Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding*, 187, 2019. 2

[13] Hirsh Goldberg, Chris Ratto, Amit Banerjee, Michael Kelbaugh, E. D. Jansing, Mark Giglio, Christopher Barber, Kelcy Smith, and Eric Vermote. Automated global-scale detection and characterization of anthropogenic activity using multi-source satellite-based remote sensing imagery. In *Defense + Commercial Sensing*, 2023. 1, 2, 5, 6, 7

[14] Patrick Griffiths, Claas Nendel, Jürgen Pickert, and Patrick Hostert. Towards national-scale characterization of grassland use intensity from integrated sentinel-2 and landsat time series. *Remote Sensing of Environment*, 238, 2020. 2

[15] Dino Ienco, Roberto Interdonato, Raffaele Gaetano, and Dinh Ho Tong Minh. Combining sentinel-1 and sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158, 2019. 2

[16] Marrit Leenstra, Diego Marcos, Francesca Bovolo, and Devis Tuia. Self-supervised pre-training enhances change detection in sentinel-2 imagery. In *International Conference on Pattern Recognition*, 2021. 4

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, 2017. 4

[18] Mengxi Liu, Zhuoqun Chai, Haojun Deng, and Rong Liu. A cnn-transformer network with multiscale context aggregation for fine-grained cropland change detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 2022. 2

[19] Maria Papadomanolaki, Sagar Verma, Maria Vakalopoulou, Siddharth Gupta, and Konstantinos Karantzalos. Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data. In *IEEE International Geoscience and Remote Sensing Symposium*, 2019. 2

[20] Xueli Peng, Ruofei Zhong, Zhen Li, and Qingyang Li. Optical remote sensing image change detection based on attention mechanism and image difference. *IEEE Transactions on Geoscience and Remote Sensing*, 59, 2020. 2

[21] Kyle T Peterson, Vasit Sagan, and John J Sloan. Deep learning-based water quality estimation and anomaly detection using landsat-8/sentinel-2 virtual constellation and cloud computing. *GIScience & Remote Sensing*, 57(4), 2020. 2

[22] Zhenfeng Shao, Jiajun Cai, Peng Fu, Leiqiu Hu, and Tao Liu. Deep learning-based fusion of landsat-8 and sentinel-2 images for a harmonized surface reflectance product. *Remote Sensing of Environment*, 235, 2019. 2

[23] Li Shen, Yao Lu, Hao Chen, Hao Wei, Donghai Xie, Jiabao Yue, Rui Chen, Shouye Lv, and Bitao Jiang. S2looking: A satellite side-looking dataset for building change detection. *Remote Sensing*, 13, 2021. 2

[24] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S. Feris, David Harwath, James Glass, and Hilde Kuehne. Everything at once - multi-modal fusion transformer for video retrieval. In *IEEE Computer Vision and Pattern Recognition*, 2022. 2

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2, 3

[26] Xinye Wanyan, Sachith Seneviratne, Shuchang Shen, and Michael Kirley. Dino-mc: Self-supervised contrastive learning for remote sensing imagery with multi-sized local crops. *arXiv preprint arXiv:2303.06670*, 2023. 8

[27] Dawei Wen, Xin Huang, Francesca Bovolo, Jiayi Li, Xinli Ke, Anlu Zhang, and Jon Atli Benediktsson. Change detection from very-high-spatial-resolution optical remote sensing

images: Methods, applications, and future directions. *IEEE Geoscience and Remote Sensing Magazine*, 9(4), 2021. 2

[28] Curtis E Woodcock, Thomas R Loveland, Martin Herold, and Marvin E Bauer. Transitioning from change detection to monitoring with remote sensing: A paradigm shift. *Remote Sensing of Environment*, 238, 2020. 2

[29] Scott Workman, Armin Hadzic, and M. Usman Rafique. Handling image and label resolution mismatch in remote sensing. In *Winter Conference on Applications of Computer Vision*, 2023. 3, 4

[30] Scott Workman, M Usman Rafique, Hunter Blanton, Connor Greenwell, and Nathan Jacobs. Single image cloud detection via multi-image fusion. In *IEEE International Geoscience and Remote Sensing Symposium*, 2020. 6

[31] Daniele Zanaga, Ruben Van De Kerchove, Dirk Daems, W De Keersmaecker, Carsten Brockmann, Grit Kirches, Jan Wevers, Oliver Cartus, Maurizio Santoro, Steffen Fritz, et al. Esa worldcover 10 m 2021 v200. 2022. 4

[32] Cui Zhang, Liejun Wang, Shuli Cheng, and Yongming Li. Swinsunet: Pure transformer network for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 2022. 2

[33] Zhe Zhu. Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130, 2017. 2

[34] Zhe Zhu and Curtis E Woodcock. Continuous change detection and classification of land cover using all available landsat data. *Remote Sensing of Environment*, 144, 2014. 2

[35] Zhe Zhu, Curtis E Woodcock, and Pontus Olofsson. Continuous monitoring of forest disturbance using all available landsat imagery. *Remote Sensing of Environment*, 122, 2012. 2