Make Some Noise: Unsupervised Remote Sensing Change Detection Using Latent Space Perturbations

Blaž Rolih Matic Fučka Filip Wolf Luka Čehovin Zajc
University of Ljubljana, Faculty of Computer and Information Science, Slovenia
blaz.rolih@fri.uni-lj.si

Abstract

*Unsupervised change detection (UCD) in remote sensing aims to localise semantic changes between two images of the same region without relying on labelled data during training. Most recent approaches rely either on frozen foundation models in a training-free manner or on training with synthetic changes generated in pixel space. Both strategies inherently rely on predefined assumptions about change types, typically introduced through handcrafted rules, external datasets, or auxiliary generative models. Due to these assumptions, such methods fail to generalise beyond a few change types, limiting their real-world usage, especially in rare or complex scenarios. To address this, we propose MaSoN (**Ma**ke **So**me **N**oise), an end-to-end UCD framework that synthesises diverse changes directly in the latent feature space during training. It generates changes that are dynamically estimated using feature statistics of target data, enabling diverse yet data-driven variation aligned with the target domain. It also easily extends to new modalities, such as SAR. MaSoN generalises strongly across diverse change types and achieves state-of-the-art performance on five benchmarks, improving the average F1 score by 14.1 percentage points. Project page: https://blaz-r.github.io/mason_ucd/*

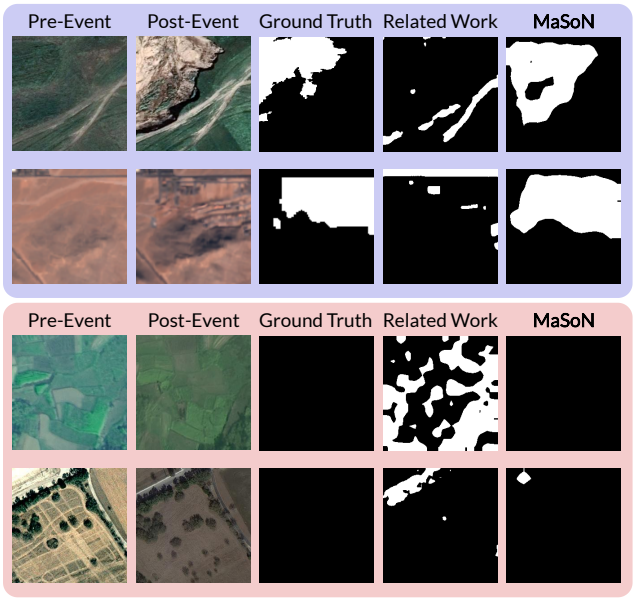

Figure 1. Related methods that rely on foundation models in a training-free manner or on changes generated in pixel-space often produce noisy or inaccurate masks, either missing *relevant changes* or overreacting to *irrelevant changes*. By generating changes in latent space, MaSoN learns from diverse and more data-aligned variations, producing better predictions with fewer false positives. This highlights MaSoN's improved ability to generalise across diverse and challenging change scenarios.

1. Introduction

Change detection (CD) is a fundamental task in remote sensing that involves localising semantic changes in a sequence of images (typically two) of the same geographic region [9, 14, 17, 75]. This task is essential to various applications, including disaster response, urban development monitoring, and land use mapping [14, 17, 23, 78]. Recent supervised methods have demonstrated impressive results [9, 11, 14, 33], but they rely on pixel-level annotations and fail to generalise to unseen change types or new geographic contexts [2, 16, 25, 54].

Dependence on labels presents a critical limitation for time-sensitive applications and situations where annotation is unfeasible. In these situations, unsupervised change detection (UCD) offers an effective alternative [17, 54, 75], as it removes the need for manual labelling and enables the use of large archives of unlabelled bi-temporal imagery. By enabling fast analysis without human supervision, UCD methods support rapid response [16, 25, 41, 62] and efficient large-area monitoring with reduced reliance on labour-intensive annotation pipelines [17].

Recent UCD methods largely follow two strategies, each with limitations. *Training free approaches* [31, 55, 56, 75] leverage frozen priors in a training-free manner (e.g., Seg-

ment Anything Model - SAM [28]), but these are typically derived from natural/urban imagery and degrade under domain shift [25] (e.g., landslides or croplands). State-of-the-art *learning-based methods* aim to overcome this by adopting a two-stage pipeline: they first *generate* synthetic changes in pixel space and then *train* a change detector on the generated samples [4, 10, 44, 74]. However, the generated changes are shaped by assumptions introduced via auxiliary models [44], handcrafted rules [10, 18], or external datasets [4, 73, 76], which limits diversity and generalisation to novel change types. Moreover, methods from both families often struggle with seasonal or radiometric variations (which we refer to as *irrelevant changes*[1]) and are, in most cases, restricted to RGB inputs.

We address these limitations with MaSoN (**Ma**ke **So**me **N**oise), a novel end-to-end UCD framework that trains on dynamically generated changes in the latent space. Instead of pixel-space augmentations, MaSoN injects Gaussian noise into the feature maps of a pretrained encoder. The noise scale is dynamically estimated using feature statistics from the target data, enabling on-the-fly generation of decoupled relevant and irrelevant changes. This design introduces diverse domain-data-aligned training signals without any labelled external data or a multi-stage setup. Because change synthesis occurs in feature space, MaSoN is modality-agnostic and can be extended to new modalities such as multispectral and SAR with a simple encoder swap. As a result, MaSoN achieves superior generalisation and consistently outperforms prior methods across multiple datasets and change types, as illustrated in Figure 1.

Our contributions are twofold: **(i)** *We propose the first end-to-end latent space change generation and detection framework* that can be trained in an unsupervised manner with our on-the-fly change synthesis strategy. Our framework overcomes the generalisation problems of related approaches, leading to improved detection performance across diverse change types. **(ii)** *We introduce a procedure for creating synthetic changes inside the latent space of an encoder.* The synthetic changes are modelled as Gaussian noise, which is dynamically estimated using simple statistics derived from the latent features of the target data. Through experiments and theoretical analysis, we show that real-world changes can be decoupled into irrelevant and relevant categories and approximated accordingly. This strategy enables variability that is hard to capture in pixel space, and it naturally supports non-RGB modalities.

We evaluate MaSoN on five benchmark datasets spanning diverse change types, including natural disasters, building changes, urban development, and cropland changes. MaSoN achieves superior performance on four out of five datasets, outperforming prior methods by an average of 14.1 percentage points, a relative improvement of 38.6%. We also demonstrate that MaSoN extends to multispectral and SAR modalities, achieving strong results.

[1] Seasonal changes can be relevant in some applications, but we follow the majority of related work and benchmarks that consider them irrelevant.

2. Related work

Remote sensing change detection (CD) is a long-established task, with early approaches based on pixel-level comparisons and statistical techniques [29, 42, 48, 59]. Deep learning advancements have since enabled end-to-end CD using Siamese convolutional architectures [8, 14, 33] which were later extended to transformers [1, 9, 69, 70], state-space models like Mamba [11, 22, 35] and diffusion-based approaches [3, 4, 76]. However, most recent models remain supervised and heavily depend on scarce annotated datasets, which are difficult to collect, especially for rare events like natural disasters [17, 25]. Furthermore, such models often fail to generalise to unseen scenarios [25, 54], requiring costly supervised retraining.

Unsupervised change detection addresses these limitations by operating solely on unannotated image pairs, making it especially valuable for rapid disaster response and other time-critical applications where annotated data is unavailable [25, 54, 75]. Methods like image-differencing [59], Markov Random Fields [6], and change vector analysis (CVA) [5, 6] rely on probability theory and statistics. While historically important, they are limited in expressiveness and are outperformed by modern deep-learning counterparts [21, 55, 56, 75].

Several recent approaches leverage large vision foundation models (VFMs). AnyChange [75], SCM [61], and DynamicEarth [31] use Segment Anything Model (SAM) [28] for segmentation, and CDRL-SA [45] uses it for refinement. Although effective in some scenarios, these models rely on SAM with frozen prior knowledge from the natural image domain, making them less suitable for remote sensing tasks. In contrast, our method is learning-based, allowing representations to adapt to the domain, thus improving generalisation and flexibility.

Other learning-based methods create synthetic changes in pixel-space [10, 44, 60, 76]. ChangeStar [73] constructs synthetic data from unrelated segmentation datasets. Other methods, such as CDRL [44], FCD-GAN [67], SyntheWorld [60], Changen [74, 76], and HySCDG [4] use generative models for style transfer or full image synthesis. These models require auxiliary external datasets to generate synthetic changes, which limits diversity and domain generalisation. I3PE [10] and S2C [18] remove such dependencies but still operate in pixel space using handcrafted transformations, leading to constrained variability. In contrast, MaSoN generates synthetic changes directly in the latent feature space during training. These changes are dynamically estimated using feature statistics of the target data, enabling diverse yet domain-consistent changes and stronger gener-

alisation.

Learning with Noise is an established strategy in deep learning for improving robustness and generalisation. DN-DETR [30] uses query denoising to accelerate training and stabilise bipartite matching for *object detection*. In *anomaly detection*, noise was applied to simulate anomalies [37, 50, 51]. GeoCLIP [63] leverages noise to model GPS inaccuracies and for contrastive learning in *geolocalisation*. It has also been applied to mitigate label uncertainty in hyperspectral change detection [32]. Building on these foundations, MaSoN introduces a novel strategy to generate changes with Gaussian noise directly in the feature space. Unlike previous methods that rely on fixed [37, 47] and non-decoupled [7] Gaussian noise, MaSoN decouples it into irrelevant and relevant noise and dynamically adjusts it during training to address the inherent variability in remote sensing change detection.

3. Let's Make Some Noise

The proposed method (MaSoN) consists of a *Shared Weight Encoder*, a *Latent Space Change Generation Strategy*, and a *Mask Decoder*. The features are first extracted using the *Encoder* (Φ). Then the *Latent Space Change Generation Strategy* is used (only during training) to generate synthetic changes on the feature level. Features from different time steps with generated synthetic changes are then fused (in our case via element-wise subtraction) and fed into the *Mask Decoder* ($\mathbb{D}$), which outputs the predicted change mask. The architecture of our method is depicted in Figure 3 and explained in the following subsections.

3.1. Encoder

Given a bi-temporal input image pair (I_1, I_2), we extract features (F_1, F_2) using a shared-weight pretrained encoder. For each image I_i, the encoder produces a hierarchical feature set:

$$\Phi(I_i) = F_i = \{f_i^{(l)} \mid l \in L\}, \tag{1}$$

where l denotes the layer level and $f_i^{(l)}$ is the feature map at layer l from image I_i.

3.2. Why Gaussian noise?

To better understand and motivate our synthetic change generation procedure, we analyse the behaviour of features extracted by the pretrained encoder. Our goal is to determine whether semantic changes can be meaningfully characterised and thus approximated directly in the latent space.

We perform the analysis on five different benchmark datasets (described later in Section 4). For each image pair, we extract features and compute element-wise difference at each layer: $f_{diff}^{(l)} = f_1^{(l)} - f_2^{(l)}$. Using the ground truth annotations, we group these differences into changed and unchanged regions. Figure 2 shows histograms of these difference values averaged across all datasets for each layer in the feature hierarchy.

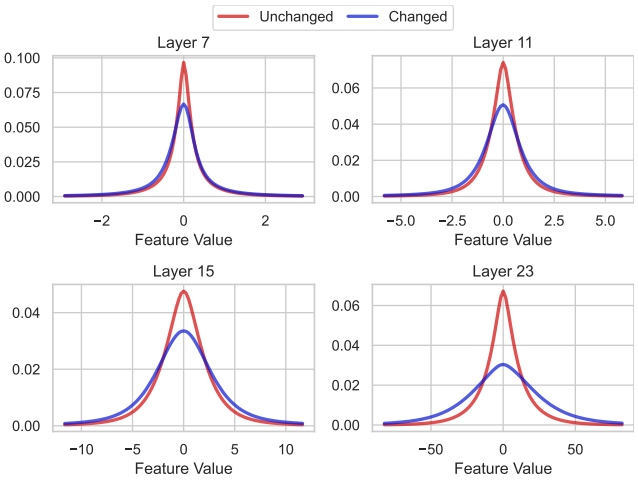

Figure 2. Histogram plot of feature differences $f_1^{(l)} - f_2^{(l)}$, averaged across all five datasets and all channels per layer. Unchanged regions are narrowly concentrated near zero, while changed regions exhibit broader variation, especially in deeper layers. Both distributions can be approximated by a zero-centred Gaussian, but each with a different variance parameter. This directly motivates our latent-space change generation strategy.

Main Takeaways According to our analysis, changed and unchanged regions exhibit distinct patterns in feature space. Unchanged areas produce small differences, resulting in sharp peaks near zero with a narrow distribution. Changed regions show a broader, heavier-tailed distribution, with fewer near-zero differences, especially in deeper layers that capture high-level semantic concepts. Although both distributions overlap and are *centred at zero*, their *variances differ*. From a theoretical perspective, the *maximum-entropy* principle states that, among all distributions consistent with the known constraints, the one with the largest entropy is the best representation of our current knowledge [27]. Given the observed zero mean and per-channel variances, the *Gaussian distribution satisfies these constraints* and *maximises differential entropy* [13].

The theoretical grounding and empirical similarity to the Gaussian distribution suggest that both unchanged and changed regions can be synthetically generated (or rather approximated) during training using properly calibrated Gaussian noise. We later demonstrate that calibration can be achieved using batch statistics. An extended theoretical justification is provided in Appendix D, while full empirical implementation details, per-dataset results, and discussion are given in Appendix C.

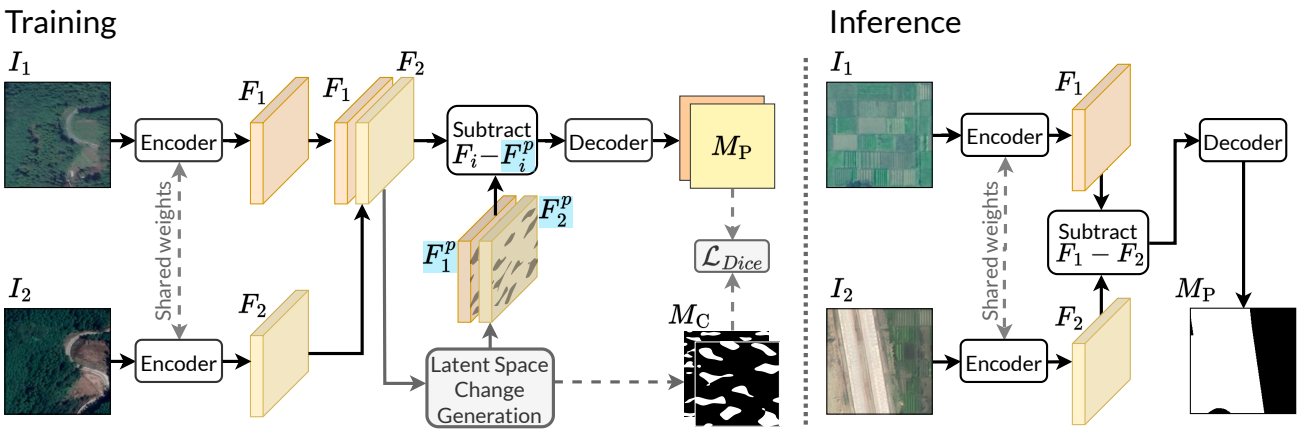

Figure 3. The architecture of the proposed unsupervised change detection framework, MaSoN.

3.3. Latent Space Change Generation Strategy

As illustrated in Figure 4, given feature map sets of an image pair, F_1 and F_2, our goal is to synthesise training pairs with controllable changes. Since image pairs may already contain (unknown) changes, directly using each pair in a bi-temporal scenario and generating additional synthetic changes would lead the model to ignore the unknown changes. Instead, we treat each image independently and generate synthetic changes by perturbing its own features.

In bi-temporal imagery, we divide changes between the images into two categories:

(i) *irrelevant changes*, such as illumination shifts, minor vegetation growth, or seasonal changes, and

(ii) *relevant changes*, which correspond to large semantic modifications like building construction or landslide damage. Based on our analysis in Subsection 3.2, we simulate these two types of changes using separate noise scales.

Irrelevant Change Noise Empirically, the distribution of feature differences between unchanged regions follows a Gaussian-like shape (see Section 3.2). Motivated by this, we model irrelevant changes using Gaussian noise. Specifically, we sample a noise ε_{I}:

$$\varepsilon_{\mathrm{I}}^{(l)} \sim \mathcal{N}(0, (\sigma_I^{(l)})^2), \tag{2}$$

where the $\sigma_I^{(l)}$ is set to the q_I-th quantile of the absolute feature difference of specific feature layer l:

$$\sigma_I^{(l)} = \text{Quantile}(|f_1^{(l)} - f_2^{(l)}|, q_I). \tag{3}$$

q_I is a learnable parameter, which is initialised with a lower value (in our case, 0.85) to approximate the narrow distribution observed in our analysis. Quantile sampling is performed dynamically for each layer and feature channel within every training batch, allowing the noise magnitude to adapt to local feature statistics. The *key idea* is sampling from feature difference, which enables the extraction of irrelevant variation directly from the feature differences. This enables estimation of intra-image irrelevant variation without relying on fixed values.

Without injecting ε_{I}, the model would associate unchanged regions with exactly zero feature difference, making it overly sensitive to minor variations at test time.

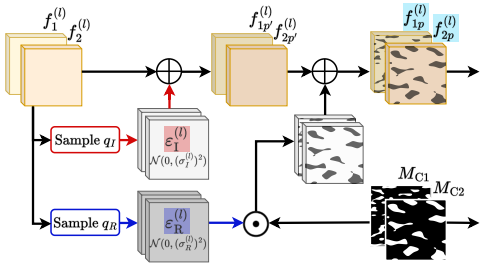

Figure 4. Illustration of the change generation process.

Relevant Change Noise Similarly, motivated by our feature analysis in Section 3.2, the distribution of feature differences in changed regions approximately follows a Gaussian pattern. To approximate this, noise ε_{R} is sampled as:

$$\varepsilon_{\mathrm{R}}^{(l)} \sim \mathcal{N}(0, (\sigma_R^{(l)})^2), \tag{4}$$

where $\sigma_R^{(l)}$ is the q_R-th quantile computed from the concatenation of both feature sets along the batch dimension:

$$\sigma_R^{(l)} = \text{Quantile}(\text{Concat}([f_1^{(l)}, f_2^{(l)}], \text{dim} = \text{batch}), q_R). \tag{5}$$

We sample quantiles from both feature sets (concatenation) so the sampled value stays within the realistic bounds. q_R is a learnable parameter initialised to a value close to one (in our case, 0.98), so it captures the broader variance observed for relevant (changed) regions. As with irrelevant noise, the quantiles are sampled independently per channel.

The noise is applied selectively using a binary mask M_{C}, generated from thresholded Perlin noise [49] (see Figure 4). The mask ensures that the changed region is random

but spatially consistent, while conveniently serving as the ground truth during training.

Final Synthetic Pairs We generate the final synthetic training pairs (also depicted in Figure 4) for each layer as:

$$\begin{aligned}(f_1^{(l)}, f_1^{(l)} + \varepsilon_{\text{I}}^{(l)} + M_{\text{C1}} \odot \varepsilon_{\text{R}}^{(l)}) \\ (f_2^{(l)}, f_2^{(l)} + \varepsilon_{\text{I}}^{(l)} + M_{\text{C2}} \odot \varepsilon_{\text{R}}^{(l)}) \\ = (f_1^{(l)}, f_{1p}^{(l)})(f_2^{(l)}, f_{2p}^{(l)}),\end{aligned} \tag{6}$$

where $\odot$ denotes element-wise multiplication. The synthetic training pairs are then assembled back into a set across all layers, resulting in two training pairs:

$$\begin{aligned}\left(\{f_1^{(l)} \mid l \in L\}, \{f_{1p}^{(l)} \mid l \in L\}\right) \\ \left(\{f_2^{(l)} \mid l \in L\}, \{f_{2p}^{(l)} \mid l \in L\}\right) \\ = (F_1, F_1^p)\ (F_2, F_2^p),\end{aligned} \tag{7}$$

where F_i^p denotes the synthetic perturbed feature set. This formulation enables the model to learn to detect relevant semantic changes while remaining robust to irrelevant variations, all without requiring any labelled change annotations.

Unlike prior works from other fields (e.g., anomaly detection) where noise is used as a static [37, 47, 50], non-decoupled [7] or robustness-oriented [30, 63] perturbation, MaSoN *dynamically* estimates two *decoupled* noise components (irrelevant and relevant) from feature statistics, a design motivated by the higher intra- and inter-image variability of remote sensing data.

3.4. Mask Decoder

To produce the final change mask prediction, feature pairs are first subtracted elementwise:

$$\begin{aligned}F_i^{diff} &= F_i - F_i^p \quad \text{during } \textit{training}, \quad \text{and} \\ F^{diff} &= F_1 - F_2 \quad \text{during } \textit{inference}.\end{aligned} \tag{8}$$

The resulting set of feature differences is passed to the UPerNet decoder [68], which processes a hierarchy of feature differences to produce the final change map $M_{\text{P}i}$, which is of the same spatial dimensions as the input image:

$$M_{\text{P}i} = \mathbb{D}(F_i^{diff}) \ . \tag{9}$$

Since we make two training pairs from a single input pair during training (Equation (6) and Equation (7)), the decoder also predicts two maps M_{P1} and M_{P2}. During inference, the decoder produces just a single map per feature pair (M_{P}, Figure 3 right).

3.5. Training and Inference

The model is trained using Dice loss [43] where the prediction is the change map $M_{\text{P}i}$ from Equation (9) and the target is set to the binary mask $M_{\text{C}i}$ from Equation (6):

$$\mathcal{L} = \mathcal{L}_{Dice}(M_{\text{P1}}, M_{\text{C1}}) + \mathcal{L}_{Dice}(M_{\text{P2}}, M_{\text{C2}}) \ . \tag{10}$$

During inference, the change generation mechanism is removed, and the model directly predicts the change map from a pair of images (right part of Figure 3). To obtain the binary mask, a sigmoid and a threshold of 0.5 are applied.

4. Experiments

Implementation Details As the encoder, we use the ViT-L [19] version of DINOv3 [58], which we do not finetune. Following related work [64], we take features from layers 7, 11, 15, and 23. Input images are cropped to 256×256 pixels. During training, we apply the flip and rotate augmentations. The model is trained using the AdamW [38] optimiser, with an initial learning rate of 10^{-7} for the learnable quantile parameters, and 10^{-5} for the decoder. A cosine scheduler without restarts is used. Each training run lasts 1000 iterations with a batch size of 16 and completes in approximately 7 minutes on a single NVIDIA A100 GPU. Additional details can be found in the Appendix J.

Evaluation Metrics and Datasets Following standard practice [1, 9, 14, 52, 74, 75], we evaluate change detection performance using **binary** precision, recall, and F1 score considering only **change class**. Experiments are repeated across five random seeds, and we report the mean and standard deviation. For each run, the model checkpoint from the final epoch is used. For training-free baselines, we use author-provided weights.

To ensure robustness, we evaluate on five change detection datasets covering *worldwide locations* and spanning *diverse change types*, from *building* and *urban* changes to *cropland* and *natural disaster* changes. The datasets cover different ground sample distances, sensors, and global geographic regions. *SYSU* [57] includes multiple change types, such as buildings, groundwork, vegetation, and sea changes. *LEVIR* [8] focuses on building changes, while *CLCD* [34] captures cropland changes. We also include datasets that are particularly challenging for unsupervised methods: *GVLM* [72], containing natural disaster changes resulting from landslides, and *OSCD* [15], a low-resolution Sentinel-2 dataset covering worldwide urban changes. The models are trained on the dedicated training sets (using no labels) and evaluated on the corresponding test sets. Additional dataset details are provided in the Appendix B.

4.1. Experimental Results

Unsupervised Change Detection We compare MaSoN against a range of unsupervised change detection methods. We evaluate image pixel differencing [26, 59], DCVA [55],

Table 1. Unsupervised change detection results across five diverse datasets and their average. We report Precision (Pr.), Recall (Re.), and F1 score. Results of additional metrics for individual datasets are provided in the Appendix E. FPS was benchmarked on an NVIDIA A100 (details and more computation metrics are in Appendix H). First and second place results are marked.

	FPS [img/s]	Param. [M]	SYSU F1	LEVIR F1	GVLM F1	CLCD F1	OSCD F1	*Average* Pr.	*Average* Re.	*Average* F1
Pixel Difference [59]	456.4±54.5	0.0	42.9	8.9	23.4	16.1	18.5	16.2	49.3	22.0
DCVA [55]$_{TGRS19}$	0.4±0.0	0.5	2.5	0.2	3.2	3.1	38.9	30.0	9.3	9.6
DINOv3-CVA [75]	24.4±0.3	303.1	56.5	19.1	19.3	27.1	21.3	19.0	**80.4**	28.7
AnyChange [75]$_{NeurIPS24}$	0.4±0.0	641.1	46.0	25.4	20.8	26.5	26.1	26.0	43.0	29.0
Changen2-S9 [76]$_{TPAMI25}$	33.3±0.1	99.9	44.8	19.9	19.1	15.6	1.5	17.3	39.4	20.2
I3PE [10]$_{ISPRS23}$	65.9±0.2	24.9	33.7	16.8	20.3	11.5	3.6	15.6	24.8	17.2
HySCDG [4]$_{CVPR25}$	41.0±0.1	65.1	52.7	15.4	17.3	24.9	17.2	29.2	65.2	25.5
SCM [61]$_{IGARSS24}$	1.2±0.0	223.5	25.7	27.3	28.0	18.9	16.6	21.6	30.1	23.3
DynEarth [31]$_{AAAI26}$	0.3±0.0	877.6	56.2	46.2	16.6	29.5	26.6	30.8	45.9	35.0
DynE. (DINOv3) [31]$_{AAAI26}$	0.3±0.0	877.6	60.3	**49.5**	19.3	33.7	18.2	34.9	42.0	36.2
S2C (DINOv3) [18]$_{AAAI26}$	3.8±0.0	303.6	37.6	32.0	38.1	52.2	5.1	41.4	42.1	36.5
MaSoN	39.7±0.1	337.5	**60.6** ±2.2	38.9 ±1.2	**55.4** ±2.9	**54.3** ±1.5	**43.5** ±1.0	**44.4** ±1.9	67.2 ±3.8	**50.6** ±0.4

and DINOv3 [58] based CVA [75]. We include state-of-the-art methods HySCDG [4], I3PE [10], Changen2 [76], AnyChange [75], SCM [61], DynamicEarth [31] and S2C [18]. Their implementation details are provided in the Appendix J.2. Results are summarised in Table 1.

MaSoN achieves the highest average F1 score of 50.6, surpassing the previous best S2C [18] by 14.1 percentage points (p. p.) on average (a 45.41% relative gain). While S2C performs well on CLCD, it degrades substantially in other settings. We attribute this to MaSoN's *data-grounded latent-space change synthesis*, which produces diverse, realistic semantic variations directly in a more robust latent representation space, yielding stronger generalisation.

Comparison against training-free methods. AnyChange [75], SCM [61], and DynamicEarth [31], all built on SAM [28], underperform MaSoN on CLCD, GVLM, and OSCD. DynamicEarth exceeds MaSoN on LEVIR (building change detection) as it produces more precise building borders (see qualitative results discussion), consistent with SAM's strength in object-centric scenes. Overall, this suggests that training-free SAM does not always transfer well to remote sensing regimes that are less object-centric and/or lower-resolution, where foreground–background cues are weaker than in natural images.

Comparison against pixel-space change generation. Pixel-space synthesis is consistently less effective than latent-space generation. S2C [18] and I3PE [10] results suggest that hand-crafted pixel transformations cannot capture the breadth of semantic change. Even deep generative pixel-space methods (Changen2 [76], HySCDG [4]) remain far behind MaSoN, supporting the view that pixel-level synthetic data generalises poorly across diverse change types and acquisition conditions.

Comparison against traditional methods. Compared to change vector analysis (CVA) [6] computed on features from the same DINOv3 [58] encoder, MaSoN doubles precision and improves average F1 by 21.9 p.p. This confirms that strong representations help, but MaSoN's gains are not due to the backbone alone. The learnable pipeline enabled by latent-space change synthesis is essential for turning features into accurate unsupervised change detection.

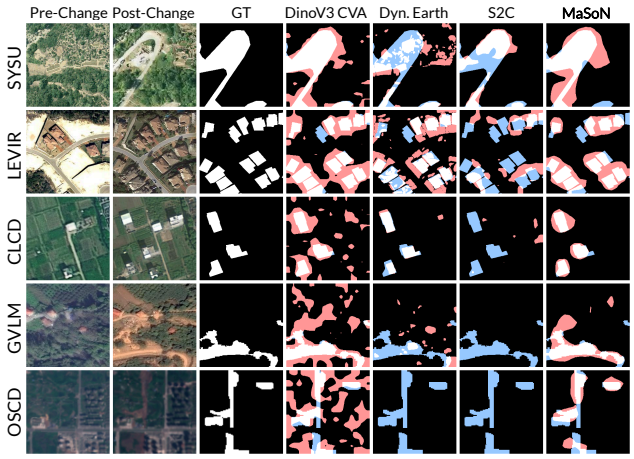

Figure 5. Qualitative comparison of the predictions. The pair of considered images is shown in the first and second columns, followed by the ground truth mask and predictions for each method. False positives are marked in red and false negatives in blue. Additional qualitative results, including discussion and failure cases, are presented in Appendix F.

Qualitative examples Figure 5 illustrates MaSoN's strong generalisation across diverse change types, particularly on GVLM (landslides) and OSCD (low-resolution), where previous SOTA methods DynamicEarth and S2C fail. On LEVIR, MaSoN occasionally overextends building regions, an issue shared by most methods except DynamicEarth, which uses SAM and is more biased toward object-centric

Table 2. Result on OMBRIA - S1 SAR flood change detection.

	Pixel Diff.	Cop.-FM CVA	MaSoN
OMBRIA (F1)	39.98	47.11	53.88 ±0.6

detection. While DINOv3-based CVA captures most changes, it does so at the cost of excessive false positives, an issue that MaSoN substantially reduces.

Extension to Specific Changes Filtering Certain applications may require isolating or filtering specific types of changes. To address this, MaSoN can be extended with a text-guided filtering procedure similar to one from DynamicEarth [31] (see Appendix J.4). To evaluate its performance in this scenario, we run the experiment on the LEVIR [8] dataset, strictly focusing on buildings. The performance *improves from* 38.9% F1 *to* 50.3% F1, surpassing the original DynamicEarth, which achieves 49.5%. As seen in Table 1, MaSoN already outperforms DynamicEarth on all other datasets without filtering. We experimented with filtering on other datasets; however, this failed to yield consistent improvements. We hypothesise that this comes from the complex nature of many change types (e.g., landslides or cropland), which are difficult to describe unambiguously by text. Consequently, we evaluated filtering on LEVIR, as building changes are textually well-defined. These findings highlight both MaSoN's flexibility and the current limitation of text-based filtering for domain-specific changes.

Extension to a new modality The RGB modality can be unreliable in certain UCD scenarios, for example, for cloud-covered flood imagery. MaSoN can be directly applied to SAR data by replacing the RGB encoder with a SAR-capable alternative, such as Copernicus-FM [65]. On the Sentinel-1 OMBRIA flood-change dataset [20] (Table 2), MaSoN outperforms both pixel differencing and CVA when using Copernicus-FM features. Other methods in Table 1 cannot be easily extended to SAR, as they rely on RGB-only SAM or restrict their change-generation mechanisms to the RGB domain.

We also extend MaSoN to the multispectral Sentinel-2 dataset using the DEO [66] backbone and evaluate it on OSCD-multispectral. It achieves an F1 score of 45.1, which is better than RGB-MaSoN and all related methods that process only the RGB part of OSCD (see Table 1).

5. Ablation Study

In this section, we experimentally validate the contributions proposed in this paper. We report the average F1 of five datasets repeated with 5 random seeds for every experiment. The full individual dataset results with additional metrics are in Appendix E.2, additional ablations in Appendix I, and ablation implementation details in Appendix J.4.

Change Generation Strategy We validate our latent-space change generation strategy through a series of ablations presented in Table 3.

Table 3. Ablations of the change generation strategy.

ID		*Avg.* F1
Ours	Fully latent space generation	50.6±0.4
Rect. M_C	Random rectangles M_C mask	46.9±0.8
No M_C	No binary mask for changed	36.5±1.7
CycGan	Irrelevant with CycGAN	36.9±0.7
No Dyn.	No dynamic estimation	25.3±1.9
No Irrel.	No irrelevant changes	15.6±0.8
Pix. Gen.	Pixel space noise generation	12.8±1.8

Replacing the binarised Perlin mask with a random rectangles mask (**Rect.** M_C) slightly lowers performance, because most changes are non-rectangular and since our approach models them better due to its higher randomness. Removing the binary mask for relevant noise (**No** M_C), where noise is instead applied to half of the features and the ground truth for those is set to "all changed", reduces performance by 14.1 p.p., confirming the benefit of spatially structured perturbations.

Next, we replace the irrelevant latent noise ε_I with CycleGAN [77]-transformed images that simulate irrelevant changes in pixel space via a learned transformation (**CycGan**). This reduces the F1 by 13.7 p.p. to 36.9%, highlighting the advantage of modelling irrelevant variation directly in latent space using our approach.

Removing dynamic noise estimation (described in Section 3.3) and instead using a fixed noise magnitude reduces performance by 25.3 p.p. (**No Dyn.**). This demonstrates the importance of data-driven estimation of noise from features that change during training and vary across geographical domains. Eliminating irrelevant change noise (**No Irrel.**, refer to Equation (2)) decreases the performance by 34.9 p.p. as the model incorrectly learns that unchanged regions correspond to zero feature difference. Together, these results show that both mechanisms are essential. Unlike machine learning works from other fields (e.g., anomaly detection - see Section 2), MaSoN *dynamically* estimates two *decoupled* noise components based on feature statistics, which is crucial for remote sensing imagery.

Applying the noise perturbation directly to the pixels of input images (**Pix. Gen.**) leads to the poorest result (12.8% F1), showing that raw pixel-level noise lacks the semantic structure and is not as nicely behaved as latent feature space.

Backbone Ablation MaSoN's performance does not stem solely from DINOv3 [58], as shown in Table 4. If we replace it with DINOv2 [46], and some weaker backbones like Swin [36] and ResNet50 [24], it still remains effective and performs better or comparable to existing approaches.

Layer Noise Importance To verify the importance of injecting noise into each layer, we conducted an experiment where we excluded noise injection for individual layers.

Table 4. Backbone ablation.

Architecture	*Avg.* F1
ViT-L (DINOv3) (Ours)	50.6±0.4
ViT-L (DINOv2)	47.4±1.5
ViT-B (DINOv3)	45.4±2.3
Swin-T (ADE20k)	41.7±1.2
ResNet50 (ADE20k)	34.1±0.5

Table 5. Target layers for noise.

L. 7	L. 11	L. 15	L. 23	*Avg.* F1
✓	✓	✓	✓	50.6±0.4
✓	✓	✓		48.6±0.3
✓	✓		✓	44.6±1.6
✓		✓	✓	46.2±1.6
	✓	✓	✓	48.7±1.2

Table 6. Noise sampling dimension.

	Avg. F1
Per-channel in batch (Ours)	50.6±0.4
Per-channel in sample	29.6±10.7
Per-sample	44.3±0.8
Per-batch	32.9±1.7

The results in Table 5 show that not applying noise to the middle 15th layer brings the largest drop in performance (5 p. p.). Nevertheless, all four layers play an important role in the overall performance.

Quantile Sampling Dimension MaSoN calculates the quantiles per channel in batch (Section 3.3). It could, however, use the same value across all channels in a batch, or separate values for each sample, or even separate values for each channel within a sample. The results of such strategies are presented in Table 6. We hypothesise that our strategy of calculating statistics per channel in batch yields the best performance due to generalisation, since the scope of sampling is neither too general (per-batch) nor too specific (per-channel).

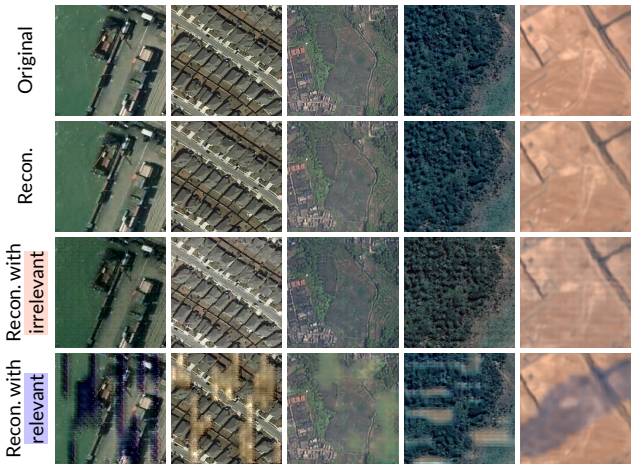

Figure 6. First row contains original input image, second row reconstructions from features, third row reconstructions with irrelevant noise added to features, and the final row reconstructions with relevant noise added to selected regions (as explained in Section 3.3). The addition of smaller irrelevant noise in the latent space is reflected in a minor colour and texture variation. The addition of relevant noise is expressed as a significant but coherent change, even in reconstructed pixel space.

Feature Reconstruction Analysis We also analyse how the latent space feature changes behave when reconstructed back to pixel space. We train a simple UNet [53] decoder to reconstruct the input image from the features (see Appendix C.2 for more details). The results shown in Figure 6 display both the reconstructions without noise and with the addition of either irrelevant or relevant noise, generated as explained in Section 3.3. The changes synthesized using our methodology in the latent space also represent a meaningful shift in the pixel space. This further demonstrates how a well-behaved structure of latent space enables a mechanism as simple as Gaussian noise to produce changes that approximate real changes. Based on this and the insights from the analysis in Section 3.2, we believe that latent-space generation has significant potential for future work.

Limitations and Future Work MaSoN can be regarded as a self-supervised approach within the broader UCD field [17]. Consequently, its limitation is the need for light finetuning rather than full training-free operation. Given its stronger performance compared to zero-shot methods and short training time, we see this as a reasonable trade-off, though further acceleration remains an important direction. Additional discussion and societal impact are provided in Appendix A.

6. Conclusion

We propose MaSoN, a framework for unsupervised change detection (UCD) that addresses the limited generalisation of existing approaches, which rely on priors from foundation models in a training-free manner or train on changes generated in pixel space. MaSoN trains on synthetic changes generated via Gaussian noise injected directly into the latent feature space, where the noise is dynamically estimated per-dataset using the data's own statistics. This enables highly diverse, data-driven change generation without external data or auxiliary generative models. MaSoN yields substantial improvements, outperforming prior unsupervised state-of-the-art by 14.1 percentage points on average across five diverse benchmarks. MaSoN also extends to multispectral and SAR modalities via a simple encoder swap, achieving great results. Beyond UCD, this work establishes latent-space synthesis as a viable and effective direction for unsupervised learning in general. The idea holds promise for sample generation and representation learning in low-data regimes, offering a path toward more general and data-efficient vision models in remote sensing as well as other domains.

Acknowledgements

This work was in part supported by the ARIS research projects J2-60045 (RoDEO) and GC-0006 (GeoAI), research programme P2-0214, and the supercomputing network SLING (ARNES, EuroHPC Vega).

References

[1] Wele Gedara Chaminda Bandara and Vishal M Patel. A Transformer-Based Siamese Network for Change Detection. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210. IEEE, 2022. 2, 5, 8, 11

[2] Wele Gedara Chaminda Bandara and Vishal M Patel. Deep Metric Learning for Unsupervised Remote Sensing Change Detection. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5125–5135. IEEE, 2025. 1

[3] Wele Gedara Chaminda Bandara, Nithin Gopalakrishnan Nair, and Vishal Patel. DDPM-CD: Denoising Diffusion Probabilistic Models as Feature Extractors for Remote Sensing Change Detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5250–5262, 2025. 2

[4] Yanis Benidir, Nicolas Gonthier, and Clément Mallet. The Change You Want To Detect: Semantic Change Detection In Earth Observation With Hybrid Data Generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2204–2214, 2025. 2, 6, 7, 9, 10

[5] Francesca Bovolo and Lorenzo Bruzzone. A Theoretical Framework for Unsupervised Change Detection Based on Change Vector Analysis in the Polar Domain. *IEEE Transactions on Geoscience and Remote Sensing*, 45(1):218–236, 2006. 2

[6] Lorenzo Bruzzone and Diego F Prieto. Automatic Analysis of the Difference Image for Unsupervised Change Detection. *IEEE Transactions on Geoscience and Remote sensing*, 38 (3):1171–1182, 2000. 2, 6

[7] Jinyu Cai and Jicong Fan. Perturbation learning based anomaly detection. *Advances in Neural Information Processing Systems*, 35:14317–14330, 2022. 3, 5

[8] Hao Chen and Zhenwei Shi. A Spatial-Temporal Attention-Based Method and A New Dataset for Remote Sensing Image Change Detection. *Remote Sensing*, 12:1662, 2020. 2, 5, 7

[9] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote Sensing Image Change Detection With Transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 1, 2, 5, 8, 11

[10] Hongruixuan Chen, Jian Song, Chen Wu, Bo Du, and Naoto Yokoya. Exchange Means Change: An Unsupervised Single-Temporal Change Detection Framework Based on Intra-and Inter-Image Patch Exchange. *ISPRS Journal of Photogrammetry and Remote Sensing*, 206:87–105, 2023. 2, 6, 7, 9, 10

[11] Hongruixuan Chen, Jian Song, Chengxi Han, Junshi Xia, and Naoto Yokoya. ChangeMamba: Remote Sensing Change Detection With Spatiotemporal State Space Model. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024. 1, 2, 8, 11

[12] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. SatMAE: Pre-Training Transformers for Temporal and Multi-Spectral Satellite Imagery. *Advances in Neural Information Processing Systems*, 35:197–211, 2022. 8, 11

[13] Thomas M. Cover and Joy A. Thomas. *Maximum Entropy*, chapter 12, pages 409–425. John Wiley & Sons, Ltd, 2005. 3, 6

[14] Rodrigo Caye Daudt, Bertr Le Saux, and Alexandre Boulch. Fully Convolutional Siamese Networks for Change Detection. In *IEEE International Conference on Image Processing*, pages 4063–4067. IEEE, 2018. 1, 2, 5, 8, 11

[15] 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*, pages 2115–2118. IEEE, 2018. 5, 2, 11

[16] Olivier Dietrich, Merlin Alfredsson, Emilia Arens, Nando Metzger, Torben Peters, Linus Scheibenreif, Jan Dirk Wegner, and Konrad Schindler. The Potential of Copernicus Satellites for Disaster Response: Retrieving Building Damage from Sentinel-1 and Sentinel-2. *arXiv*, 2025. 1

[17] Lei Ding, Danfeng Hong, Maofan Zhao, Hongruixuan Chen, Chenyu Li, Jie Deng, Naoto Yokoya, Lorenzo Bruzzone, and Jocelyn Chanussot. A Survey of Sample-Efficient Deep Learning for Change Detection in Remote Sensing: Tasks, Strategies, and Challenges. *IEEE Geoscience and Remote Sensing Magazine*, 2025. 1, 2, 8

[18] Lei Ding, Xibing Zuo, Danfeng Hong, Haitao Guo, Jun Lu, Zhihui Gong, and Lorenzo Bruzzone. S2c: Learning noise-resistant differences for unsupervised change detection in multimodal remote sensing images. *AAAI*, 2026. 2, 6, 7, 9, 10

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 5

[20] Georgios I Drakonakis, Grigorios Tsagkatakis, Konstantina Fotiadou, and Panagiotis Tsakalides. OmbriaNet—Supervised flood mapping via convolutional neural networks using multitemporal sentinel-1 and sentinel-2 data fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:2341–2356, 2022. 7

[21] Bo Du, Lixiang Ru, Chen Wu, and Liangpei Zhang. Unsupervised Deep Slow Feature Analysis for Change Detection in Multi-Temporal Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9976–9992, 2019. 2

[22] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling With Selective State Spaces. In *First Conference on Language Modeling*, 2024. 2

[23] Ronny Hänsch and Mousmi Ajay Chaurasia. Earth Observation and Machine Learning for Climate Change. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 1676–1682. IEEE, 2024. 1

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7

[25] Victor Hertel, Christian Geiß, Marc Wieland, and Hannes Taubenböck. Rapid Domain Adaptation for Disaster Impact Assessment: Remote Sensing of Building Damage After the 2021 Germany Floods. *Science of Remote Sensing*, page 100287, 2025. 1, 2

[26] Masroor Hussain, Dongmei Chen, Angela Cheng, Hui Wei, and David Stanley. Change Detection From Remotely Sensed Images: From Pixel-Based to Object-Based Approaches. *ISPRS Journal of photogrammetry and remote sensing*, 80:91–106, 2013. 5

[27] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957. 3, 6

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 6, 10

[29] Bertrand Le Saux and Hicham Randrianarivo. Urban Change Detection in SAR Images by Interactive Learning. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 3990–3993. IEEE, 2013. 2

[30] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate Detr Training by Introducing Query Denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 3, 5

[31] Kaiyu Li, Xiangyong Cao, Yupeng Deng, Chao Pang, Zepeng Xin, Deyu Meng, and Zhi Wang. Dynamicearth: How Far Are We From Open-Vocabulary Change Detection? *AAAI*, 2026. 1, 2, 6, 7, 9, 10, 11

[32] Xuelong Li, Zhenghang Yuan, and Qi Wang. Unsupervised Deep Noise Modeling for Hyperspectral Image Change Detection. *Remote Sensing*, 11(3):258, 2019. 3

[33] Zhenglai Li, Chang Tang, Xinwang Liu, Wei Zhang, Jie Dou, Lizhe Wang, and Albert Y. Zomaya. Lightweight Remote Sensing Change Detection With Progressive Feature Aggregation and Supervised Attention. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2023. 1, 2

[34] 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*, 15:4297–4306, 2022. 5, 2

[35] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual State Space Model. In *Advances in Neural Information Processing Systems*, pages 103031–103063, 2024. 2

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 7

[37] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A Simple Network for Image Anomaly Detection and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023. 3, 5

[38] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2017. 5

[39] Utkarsh Mall, Bharath Hariharan, and Kavita Bala. Change-Aware Sampling and Contrastive Learning for Satellite Images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5261–5270, 2023. 8, 11

[40] Matías Mendieta, Boran Han, Xingjian Shi, Yi Zhu, and Chen Chen. Towards Geospatial Foundation Models via Continual Pretraining. In *IEEE/CVF International Conference on Computer Vision*, pages 16806–16816, 2023. 8, 11

[41] SF Meneses III and AC Blanco. Rapid Mapping and Assessment of Damages Due to Typhoon Rai Using Sentinel-1 Synthetic Aperture Radar Data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:1139–1146, 2022. 1

[42] Nando Metzger, Mehmet Özgür Türkoglu, Rodrigo Caye Daudt, Jan Dirk Wegner, and Konrad Schindler. Urban Change Forecasting From Satellite Images. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 91(6):443–452, 2023. 2

[43] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *International Conference on 3D Vision*, pages 565–571, 2016. 5

[44] Hyeoncheol Noh, Jingi Ju, Minseok Seo, Jongchan Park, and Dong-Geol Choi. Unsupervised Change Detection Based on Image Reconstruction Loss. In *proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1352–1361, 2022. 2, 7, 9, 10, 11

[45] HyeonCheol Noh, JinGi Ju, YuHyun Kim, MinWoo Kim, and Dong-Geol Choi and. Unsupervised Change Detection Based on Image Reconstruction Loss With Segment Anything. *Remote Sensing Letters*, 15(9):919–929, 2024. 2, 7, 9, 10

[46] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning Robust Visual Features Without Supervision. *Transactions on Machine Learning Research*, 2024. 7

[47] Guansong Pang, Chunhua Shen, and Anton Van Den Hengel. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 353–362, 2019. 3, 5

[48] Daifeng Peng, Xuelian Liu, Yongjun Zhang, Haiyan Guan, Yansheng Li, and Lorenzo Bruzzone. Deep Learning Change

Detection Techniques for Optical Remote Sensing Imagery: Status, Perspectives and Challenges. *International Journal of Applied Earth Observation and Geoinformation*, 136: 104282, 2025. 2

[49] Ken Perlin. An Image Synthesizer. *ACM Siggraph Computer Graphics*, 19:287–296, 1985. 4

[50] Blaž Rolih, Matic Fučka, and Danijel Skočaj. SuperSimpleNet: Unifying Unsupervised and Supervised Learning for Fast and Reliable Surface Defect Detection. In *International Conference on Pattern Recognition*, 2024. 3, 5

[51] Blaž Rolih, Matic Fučka, and Danijel Skočaj. No Label Left Behind: a Unified Surface Defect Detection Model for All Supervision Regimes. *Journal of Intelligent Manufacturing*, pages 1–21, 2025. 3

[52] Blaž Rolih, Matic Fučka, Filip Wolf, and Luka Čehovin Zajc. Be the Change You Want to See: Revisiting Remote Sensing Change Detection Practices. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–11, 2025. 5

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 8, 11

[54] Vít Růžička, Anna Vaughan, Daniele De Martini, James Fulton, Valentina Salvatelli, Chris Bridges, Gonzalo Mateo-Garcia, and Valentina Zantedeschi. RaVæn: Unsupervised Change Detection of Extreme Events Using Ml on-Board Satellites. *Scientific reports*, 12(1):16939, 2022. 1, 2

[55] Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone. Unsupervised Deep Change Vector Analysis for Multiple-Change Detection in Vhr Images. *IEEE transactions on geoscience and remote sensing*, 57(6):3677–3693, 2019. 1, 2, 5, 6, 7, 9, 10

[56] Sudipan Saha, Yady Tatiana Solano-Correa, Francesca Bovolo, and Lorenzo Bruzzone. Unsupervised Deep Transfer Learning-Based Change Detection for Hr Multispectral Images. *IEEE Geoscience and Remote Sensing Letters*, 18(5): 856–860, 2020. 1, 2

[57] Qian Shi, Mengxi Liu, Shengchen Li, Xiaoping Liu, Fei Wang, and Liangpei Zhang. A Deeply Supervised Attention Metric-Based Network and an Open Aerial Image Dataset for Remote Sensing Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022. 5, 2

[58] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 5, 6, 7, 10

[59] Ashbindu Singh. Review Article Digital Change Detection Techniques Using Remotely-Sensed Data. *International journal of remote sensing*, 10(6):989–1003, 1989. 2, 5, 6

[60] Jian Song, Hongruixuan Chen, and Naoto Yokoya. Syntheworld: A Large-Scale Synthetic Dataset for Land Cover Mapping and Building Change Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8287–8296, 2024. 2

[61] Xiaoliang Tan, Guanzhou Chen, Tong Wang, Jiaqi Wang, and Xiaodong Zhang. Segment Change Model (SCM) for Unsupervised Change Detection in Vhr Remote Sensing Images: a Case Study of Buildings. In *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 8577–8580. IEEE, 2024. 2, 6, 7, 9, 10

[62] Dimitris Valsamis, Alexandros Oikonomidis, Chrysoula Chatzichristaki, Anastasia Moumtzidou, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. Towards Advanced Wildfire Analysis: A Siamese Network-Based Change Detection Approach Through Self-Supervised Learning. In *2024 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–7. IEEE, 2024. 1

[63] Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. Geoclip: Clip-Inspired Alignment Between Locations and Images for Effective Worldwide Geo-Localization. *Advances in Neural Information Processing Systems*, 36:8690–8701, 2023. 3, 5

[64] Di Wang, Jing Zhang, Minqiang Xu, Lin Liu, Dongsheng Wang, Erzhong Gao, Chengxi Han, Haonan Guo, Bo Du, Dacheng Tao, et al. MTP: Advancing Remote Sensing Foundation Model via Multi-Task Pretraining. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. 5, 8, 11

[65] Yi Wang, Zhitong Xiong, Chenying Liu, Adam J Stewart, Thomas Dujardin, Nikolaos Ioannis Bountos, Angelos Zavras, Franziska Gerken, Ioannis Papoutsis, Laura Leal-Taixé, et al. Towards a Unified Copernicus Foundation Model for Earth Vision. *arXiv preprint arXiv:2503.11849*, 2025. 7, 10, 11

[66] Filip Wolf, Blaž Rolih, and Luka Čehovin Zajc. Brewing stronger features: Dual-teacher distillation for multispectral earth observation. *arXiv preprint*, 2026. 7, 10, 11

[67] Chen Wu, Bo Du, and Liangpei Zhang. Fully Convolutional Change Detection Framework With Generative Adversarial Network for Unsupervised, Weakly Supervised and Regional Supervised Change Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9774–9788, 2023. 2

[68] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified Perceptual Parsing for Scene Understanding. In *European Conference on Computer Vision*, pages 418–434, 2018. 5, 11

[69] Weikang Yu, Xiaokang Zhang, Samiran Das, Xiao Xiang Zhu, and Pedram Ghamisi. Maskcd: A Remote Sensing Change Detection Network Based on Mask Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 2

[70] 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*, 60:1–13, 2022. 2, 8, 11

[71] Haotian Zhang, Hao Chen, Chenyao Zhou, Keyan Chen, Chenyang Liu, Zhengxia Zou, and Zhenwei Shi. Bifa: Remote Sensing Image Change Detection With Bitemporal Feature Alignment. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 8, 11

[72] Xiaokang Zhang, Weikang Yu, Man-On Pun, and Wenzhong Shi. Cross-Domain Landslide Mapping From Large-Scale Remote Sensing Images Using Prototype-Guided Domain-Aware Progressive Representation Learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197:1–17, 2023. 5, 2

[73] Zhuo Zheng, Ailong Ma, Liangpei Zhang, and Yanfei Zhong. Change is Everywhere: Single-Temporal Supervised Object Change Detection in Remote Sensing Imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15193–15202, 2021. 2

[74] Zhuo Zheng, Shiqi Tian, Ailong Ma, Liangpei Zhang, and Yanfei Zhong. Scalable Multi-Temporal Remote Sensing Change Data Generation via Simulating Stochastic Change Process. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21818–21827, 2023. 2, 5

[75] Zhuo Zheng, Yanfei Zhong, Liangpei Zhang, and Stefano Ermon. Segment Any Change. In *Advances in Neural Information Processing Systems*, 2024. 1, 2, 5, 6, 7, 9, 10

[76] Zhuo Zheng, Stefano Ermon, Dongjun Kim, Liangpei Zhang, and Yanfei Zhong. Changen2: Multi-Temporal Remote Sensing Generative Change Foundation Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):725–741, 2025. 2, 6, 7, 9, 10

[77] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 7, 11

[78] Zhe Zhu, Shi Qiu, and Su Ye. Remote Sensing of Land Change: A Multifaceted Perspective. *Remote Sensing of Environment*, 282:113266, 2022. 1

Make Some Noise: Unsupervised Remote Sensing Change Detection Using Latent Space Perturbations

Supplementary Material

In this Appendix, we provide extensive additional details and supporting information that extend beyond the scope of the main manuscript. The Appendix is organised as follows:

- **Limitations and Societal Impact** in Section A.
- **Extended dataset details** in Section B.
- **Extended feature analysis** with per-dataset results and noised feature pixel-space reconstruction in Section C.
- **Theoretical justification of latent Gaussian noise modelling** in Section D
- **Main results with additional metrics and standard deviation** in Section E.
- **Additional qualitative results**, including failure cases in Section F.
- **Supervised results** in Section G
- **Computational efficiency** protocol, extended results, and discussion on noise generation overhead in Section H.
- **Additional ablation studies**, including multispectral and SAR data, batch size effect, statistics sampling, and noise distributions in Section I.
- **Extended implementation details** for our model, related works, and ablation studies in Section J.

A. Limitations and Societal Impact

A.1. Limitations

MaSoN can be regarded as a self-supervised approach within the broader UCD paradigm [17], and this choice introduces some practical limitations. The most central constraint is its reliance on light finetuning rather than operating in a fully zero-shot setting. Although the required adaptation is minimal, the need for downstream training introduces overhead in scenarios where compute resources are limited and rapid deployment is essential. This stands in contrast to fully zero-shot methods, which offer maximum flexibility and immediacy but typically at the cost of lower performance. However, compared to current zero-shot methods that are prohibitively slow (for example, DynamicEarth processes only 0.4 images per second), *MaSoN becomes more time-efficient overall* (including finetuning) for any *dataset larger than roughly 170 images.*

Despite these constraints, the observed performance gains relative to zero-shot approaches, as well as MaSoN's short and data-efficient training process, make this training trade-off reasonable in many practical applications. Nonetheless, reducing or eliminating the finetuning step remains an important direction for future work. Potential improvements include exploring hybrid zero-/few-shot mechanisms, more efficient learning objectives, or techniques that enable stronger generalisation from the base representation without task-specific updates.

Finally, while our empirical results show consistent improvements across evaluated benchmarks, MaSoN's generality may be constrained when applied to tasks that diverge significantly from the structure or distribution of the data used during its training. Understanding such failure modes and systematically characterising transferability remain open challenges.

A.2. Societal Impact

An unsupervised change detection model enables efficient monitoring of environmental and human-driven changes. By eliminating the need for labelled data, our method enables timely and cost-efficient monitoring of a wide range of environmental and human-driven changes, including deforestation, urban expansion, agricultural shifts, and the aftermath of natural disasters. This can be particularly valuable in regions where annotated data is scarce or in applications requiring rapid deployment, such as disaster response or early warning systems.

By avoiding reliance on external auxiliary datasets and operating in an end-to-end manner, MaSoN is suitable for real-world deployment in resource-constrained settings. However, as with all automated systems, care must be taken to interpret results responsibly and to ensure ethical use, particularly in applications that involve sensitive geographic or political regions. Ensuring transparency and human oversight in downstream use is critical to promoting ethical use. We, however, deeply believe that the advantages of our method greatly outweigh the risks and hope that this work brings us closer to enabling safer living conditions for everyone on the planet.

B. Dataset Details

Extended dataset specifications are presented in Table 1. The datasets encompass a broad range of change types, including building and urban development, cropland shifts, and natural disaster impacts. They also differ in spatial resolution (GSD), acquisition sensors, and dataset size: ranging from a few hundred to several thousand image pairs. This diversity contributes to the robustness and broader applicability of our findings.

A common challenge in remote sensing change detection is the highly imbalanced nature of the data, as changed pixels typically account for less than 10% of the total. An ex-

	Acquisition	Resolution	Change Type	Interval	Region	Image count train\val\test	Patch	Changed Pixels	Unchanged Pixels
SYSU [57]	Aerial	0.5m	Buidling, urban, groundwork, road, vegetation, sea	2007-2014	Hong Kong	12000 4000 4000	256×256	21.8 %	78.2 %
LEVIR [8]	Google Earth satellite	0.5m	Buidling	2002-2018	20 regions in US	7120 1024 2048	256×256	4.7 %	95.3 %
GVLM [72]	Google Earth satellite (SPOT-6)	0.59m	Landslide	2010-2021	17 regions worldwide	4558 1519 1519	256×256	6.6 %	93.4 %
CLCD [34]	Satellite (Gaofen-2)	0.5m-2m	Multiple types limited to croplands	2017-2019	Guangdong, China	1440 480 480	256×256	7.6 %	92.4 %
OSCD [15]	Satellite (Sentinel-2)	10m	Urban	2015-2018	24 regions worldwide	827 - 385	96×96	3.2 %	96.8 %

Table 1. Additional details for the datasets used in the paper.

ception is the SYSU dataset, which shows a higher change ratio due to its coarse annotations over large change regions.

B.1. Data Implementation Details

Dataset splits Official train-test splits are used for OSCD, SYSU, CLCD, and LEVIR, while GVLM relies on already available public random splits from Huggingface (courtesy of Weikang Yu) to ensure full reproducibility and fair comparison.

The source for the data used are as follows:

- SYSU: ericyu: SYSU_CD
- LEVIR: ericyu: LEVIRCD_Cropped256
- GVLM: ericyu: GVLM_Cropped_256
- CLCD: ericyu: CLCD_Cropped_256
- OSCD: Official webpage. Since this dataset is not prepared, we manually crop the images from a predefined train-test split into correctly shaped patches. This dataset also provides multispectral data, which we will evaluate later in further ablation studies.

C. Extended Feature Analysis

C.1. Per-Dataset Feature Differences

We use a trained model setup as described in Subsection J for supervised learning. The features are then extracted from each image in the test set for every dataset used, and feature differences are computed. This is done for each hierarchical layer in the multi-scale features elementwise. A per-channel histogram of these differences is calculated, and the mean histogram of all channels is visualised.

The average histogram over all datasets is reported in the main paper. Here, we also include an individual mean histogram for all the datasets used in our study. They can be seen in Figures 1- 5. The features of the changed and unchanged regions exhibit similar Gaussian-like behaviour across all datasets, but the distributions slightly differ for each one.

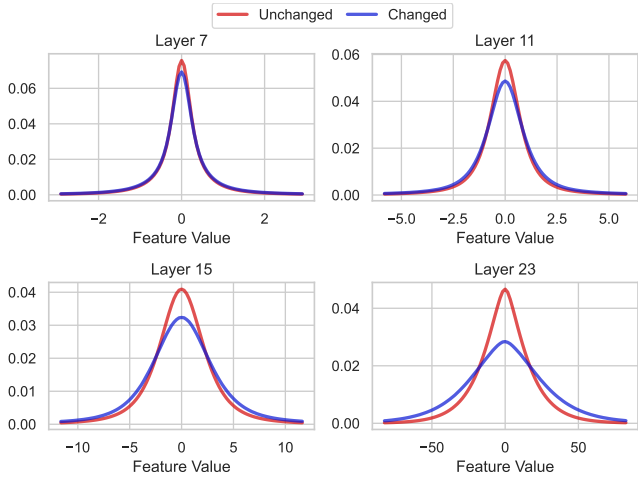

Figure 1. Distributions of feature differences $f_1^{(l)} - f_2^{(l)}$, on the SYSU Dataset.

C.2. Noised Feature reconstruction

To assess the semantic impact of our latent space perturbation strategy, we conducted a reconstruction-based visualisation experiment. Starting with the supervised model detailed in Subsection J for supervised learning, we froze the encoder and trained a separate U-Net-like decoder (differ-

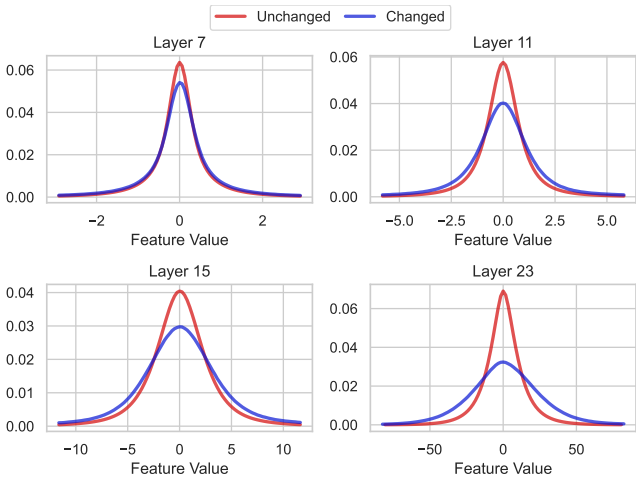

Figure 2. Distributions of feature differences $f_1^{(l)} - f_2^{(l)}$, on the LEVIR Dataset.

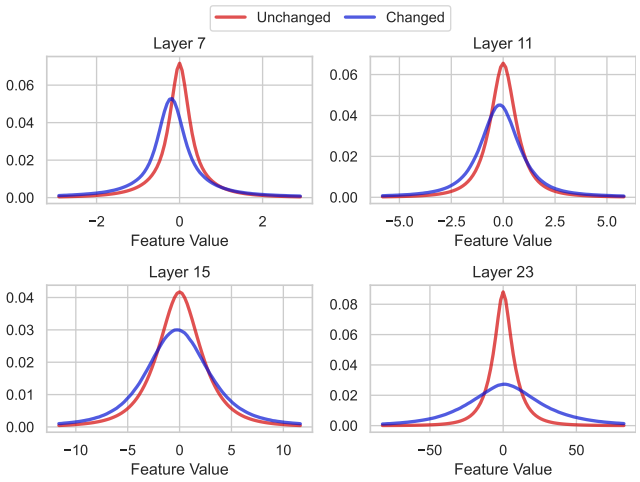

Figure 3. Distributions of feature differences $f_1^{(l)} - f_2^{(l)}$, on the GVLM Dataset.

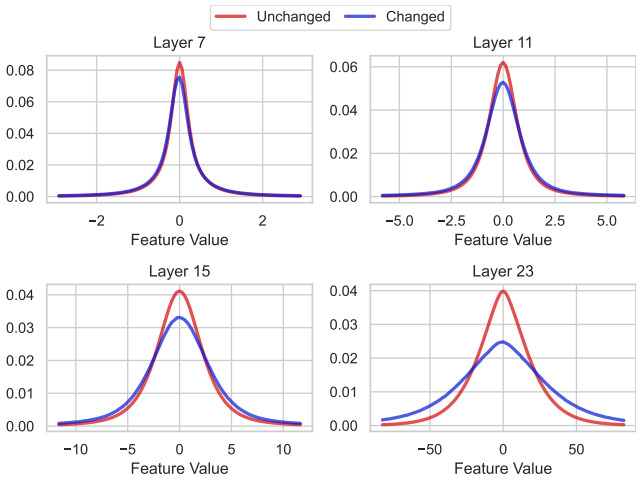

Figure 4. Distributions of feature differences $f_1^{(l)} - f_2^{(l)}$, on the CLCD Dataset.

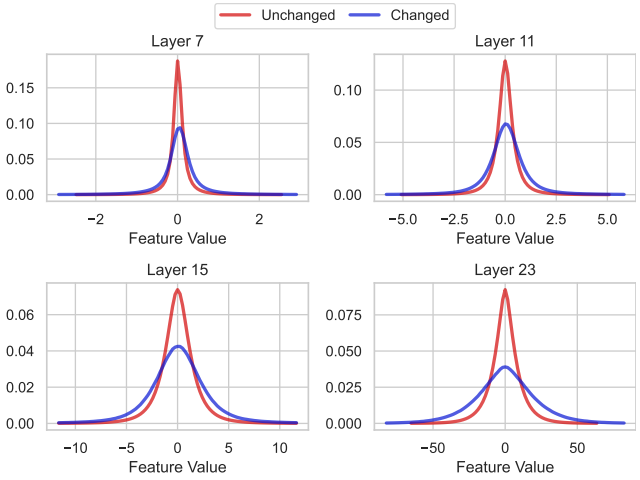

Figure 5. Distributions of feature differences $f_1^{(l)} - f_2^{(l)}$, on the OSCD Dataset.

ent from the change mask decoder) that simply reconstructs the original input image from the encoder's feature representations. This reconstruction decoder was trained for 100 epochs using an L2 loss function.

After training, we extracted features from the original encoder and applied noise sampled identically to our unsupervised change generation strategy. These perturbed features were then passed through the reconstruction decoder to generate image representations. The resulting reconstructions (Figure 6- 10) reveal that the injected noise induces plausible and semantically meaningful changes, mimicking real-world variations.

Added irrelevant noise manifests as a small variation in texture and colour, but the larger relevant noise is reflected as a noticeable change. While not totally realistic, it interestingly resembles changes like the addition of dirt or concrete or a change to grassland in some regions.

We also notice that a similar type of change occurs across the entire batch, which is a result of our per-channel in-batch sampling strategy. This strengthens our claim that the synthetic changes introduced by our strategy produce meaningful perturbations within the latent space.

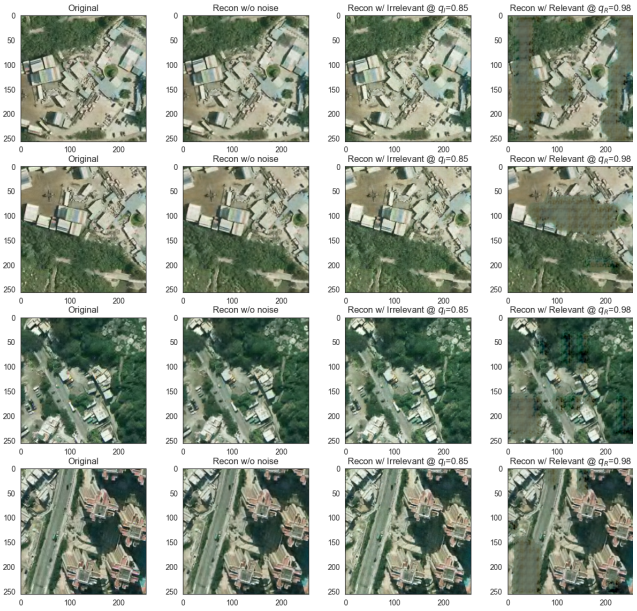

Figure 6. Qualitative examples of feature reconstructions on SYSU Dataset. In the first column, the original image is shown, in the second, the reconstructed image without any noise, in the third, the features with the addition of irrelevant noise and in the fourth, the features with the addition of relevant noise (added only to some sections of the image).

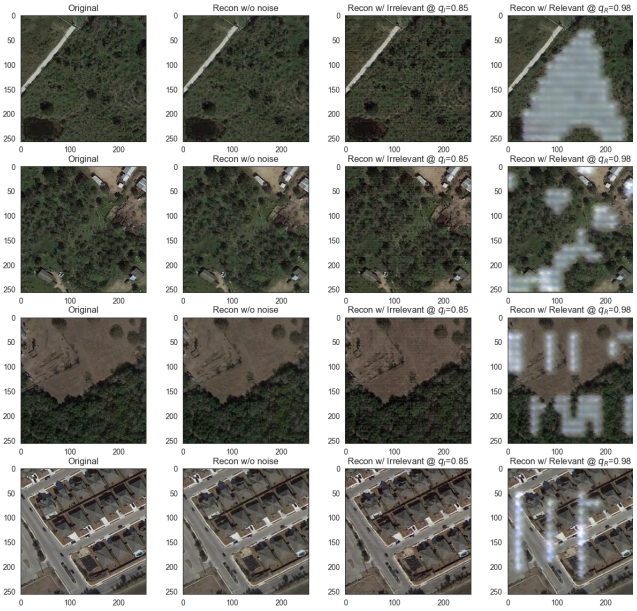

Figure 7. Qualitative examples of feature reconstructions on LEVIR Dataset.

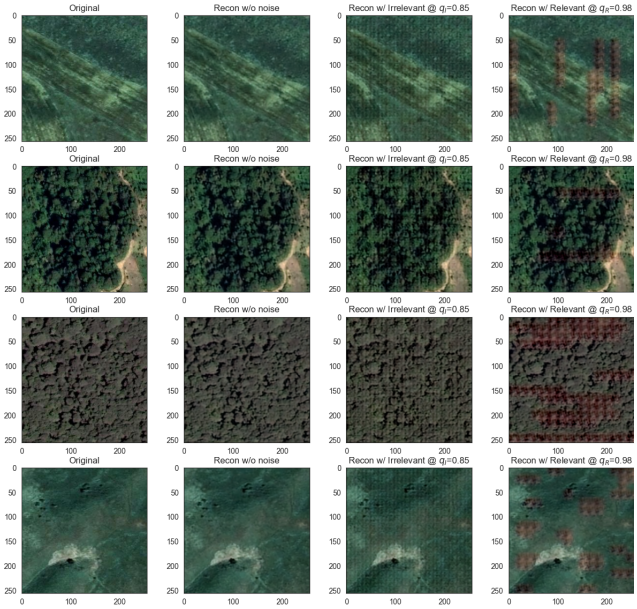

Figure 8. Qualitative examples of feature reconstructions on GVLM Dataset.

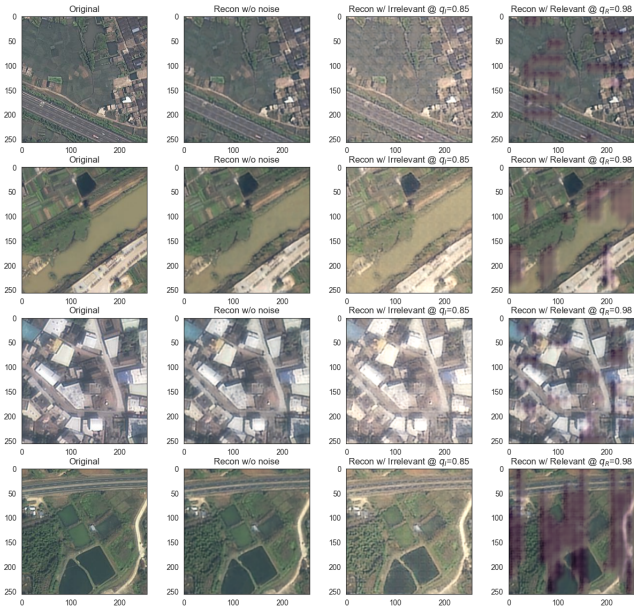

Figure 9. Qualitative examples of feature reconstructions on CLCD Dataset.

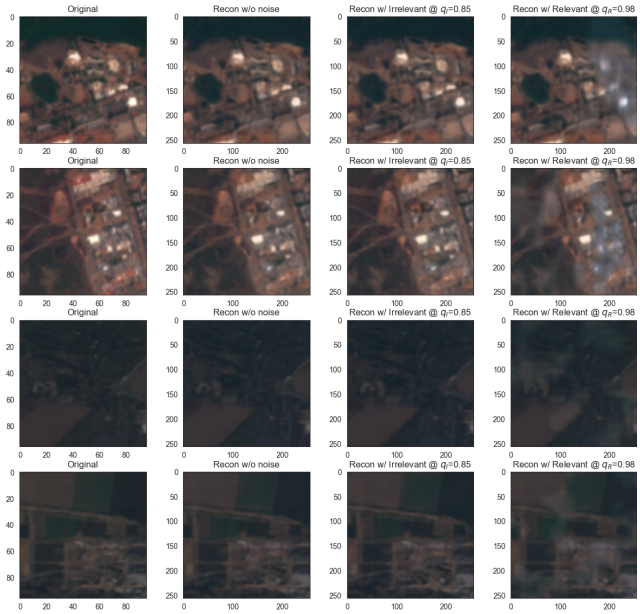

Figure 10. Qualitative examples of feature reconstructions on OSCD Dataset.

D. Theoretical justification of latent Gaussian noise modelling

From our empirical analysis (see Section 3.2), we infer the constraints $\mathbb{E}[\varepsilon] = 0$ and that the per-channel variance of relevant changes is larger than that of irrelevant changes, i.e., $\sigma_R^2 > \sigma_I^2$. We further simplify the modelling by injecting noise independently per channel (diagonal covariance). Concretely, we dynamically estimate per-channel scale parameters (e.g., via quantile-based spread estimates) from the data during training and use them to fix the corresponding second-order statistics (Eq. (2)–(5) of the main paper). The remaining question is then: given only these constraints, what distribution should we assign to ε?

The principle of maximum entropy provides an answer: among all distributions that satisfy the known constraints, the one with the largest entropy is the least biased and most faithful to the current state of knowledge [27].

Among all continuous distributions on $\mathbb{R}$ with fixed mean and variance, the Gaussian satisfies the constraints and achieves the maximum-entropy [13]:

$$\varepsilon \sim \mathcal{N}(0, \sigma^2), \tag{1}$$

Thus, modelling latent irrelevant and relevant changes as channel-wise Gaussian noise does not require assuming that individual patch values or patch differences are Gaussian. Rather, it is the maximum-entropy choice implied by the first- and second-order per-channel statistics we estimate from data. Empirically, we also observe that aggregate feature-difference distributions are approximately Gaussian (see Section 3.2), and MaSoN achieves strong results with this modelling choice, suggesting that per-channel Gaussian noise is sufficient in practice.

This theoretically motivates our design while leaving room for future work exploring more structured, constrained, or data-adaptive modelling beyond the maximum-entropy Gaussian baseline.

E. Results With Complete Metrics and Standard Deviation

E.1. Complete Unsupervised Results

In Table 2, the complete results for the unsupervised setting are presented, including precision, recall, and F1 with the accompanying standard deviation. The results show that MaSoN balances recall and precision well and achieves the best F1.

E.2. Complete Ablation Results

We provide detailed per-dataset results and additional metrics for the ablation studies discussed in the main paper. Table 3 presents the complete results for different change generation strategies. Table 4 compares different encoder architectures. Table 5 evaluates the impact of injecting noise at different feature layers, and Table 6 analyses the effect of varying the noise sampling dimension.

F. Additional Qualitative Results

In this section, we provide additional qualitative examples of MaSoN and the related methods: pixel-differencing, DINOv3 CVA [58, 75], I3PE [10], HySCDG [4], Changen2 [76], AnyChange [75], DynamicEarth [31], and S2C [18]. Compared to related work, MaSoN more accurately predicts actual changes and is more robust towards irrelevant changes.

F.1. Failure Cases

In this section, we highlight a few typical failure cases. MaSoN struggles in ambiguous scenarios where the definition of "change" is context-dependent. For instance, in the third row (CLCD), it predicts a change due to visible vegetation growth in a field. While this could reasonably be considered a change, it is not considered in this particular dataset. A similar observation can be made in the case of LEVIR (second row), where only buildings are labelled, but many unsupervised methods detect all changes, therefore marking the construction of the road as a change. Such cases reflect the inherent limitations of unsupervised learning. However, these ambiguities could be mitigated by filtering (results for this were presented in the main paper Section 5) or finetuning the model with a small number of labelled examples.

G. Supervised Results

MaSoN can also be trained in a supervised fashion. The results for this setting are presented in Table 7. Interestingly, MaSoN achieves only 10 p.p. higher recall to the one achieved in the unsupervised setting. Contrary to the unsupervised setting, the model achieves a much higher Precision. This likely happens due to the nature of most datasets, where only certain changes, like buildings, need to be detected, a task that requires at least some sort of supervision (or filtering) to convey what a building is. The other reason is that since we generate changes in the unsupervised case at a smaller resolution (i.e., that of feature maps, which are 16 times smaller than image for ViT), supervision targets don't offer the same precision, leading to more blobby predicted masks.

H. Computational Efficiency

Implementation of our protocol for measuring computational efficiency is described here (Appendix H.1). Some additional metrics, such as GFLOPs and inference time, are also provided for our method and other state-of-the-art methods (Appendix H.2). Finally, we also discuss the computational overhead of noise generation in Appendix H.3.

Table 2. Unsupervised change detection results across five diverse datasets and their average. We report Precision (Pr.), Recall (Re.), and F1 score. The symbol * indicates methods for which reproducibility with public code was not possible. Methods without standard deviations are deterministic. First and second place results are marked.

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
Pixel Difference	40.5	45.7	42.9	5.1	35.1	8.9	14.0	70.7	23.4	9.9	43.4	16.1	11.3	51.5	18.5	16.2	49.3	22.0
DCVA [55]$_{TGRS19}$	63.0	1.3	2.5	3.6	0.1	0.2	33.4	1.7	3.2	13.4	1.7	3.1	36.6	41.6	38.9	30.0	9.3	9.6
DinoV2-CVA	36.4	**81.2**	50.3	8.6	**93.9**	15.7	10.2	81.1	18.2	13.0	**87.6**	22.7	10.1	81.8	18.1	15.7	**85.1**	25.0
DINOv3-CVA	44.7	76.7	56.5	10.7	91.8	19.1	11.3	65.4	19.3	16.1	85.8	27.1	12.3	**82.2**	21.3	19.0	80.4	28.7
AnyChange [75]$_{NeurIPS24}$	48.1	44.1	46.0	15.2	77.6	25.4	17.1	26.7	20.8	19.1	43.5	26.5	30.3	22.9	26.1	26.0	43.0	29.0
Changen2-S9 [76]$_{TPAMI25}$	40.8	49.6	44.8	11.5	73.0	19.9	11.9	48.3	19.1	11.3	25.2	15.6	10.9	0.8	1.5	17.3	39.4	20.2
CDRL* [44]$_{CVPRW22}$	24.8 $\pm$0.6	1.5 $\pm$0.1	2.9 $\pm$0.1	3.8 $\pm$0.1	2.1 $\pm$0.1	2.7 $\pm$0.1	13.5 $\pm$0.0	11.6 $\pm$0.0	12.5 $\pm$0.0	0.4 $\pm$0.2	0.0 $\pm$0.0	0.1 $\pm$0.0	13.6 $\pm$1.5	0.2 $\pm$0.0	0.4 $\pm$0.0	11.2 $\pm$0.2	3.1 $\pm$0.0	3.7 $\pm$0.0
CDRL-SA* [45]$_{RSL24}$	58.4 $\pm$6.9	1.3 $\pm$0.3	2.6 $\pm$0.6	19.2 $\pm$2.3	3.0 $\pm$0.8	5.1 $\pm$1.1	15.2 $\pm$9.7	0.2 $\pm$0.1	0.4 $\pm$0.3	18.8 $\pm$11.5	0.4 $\pm$0.3	0.7 $\pm$0.6	30.2 $\pm$10.5	0.3 $\pm$0.2	0.5 $\pm$0.4	28.4 $\pm$4.2	1.0 $\pm$0.2	1.9 $\pm$0.3
I3PE [10]$_{ISPRS23}$	31.0 $\pm$1.9	38.0 $\pm$9.7	33.7 $\pm$4.2	10.7 $\pm$0.3	38.7 $\pm$3.7	16.8 $\pm$0.6	19.2 $\pm$2.6	23.2 $\pm$8.0	20.3 $\pm$3.1	7.9 $\pm$1.0	21.3 $\pm$5.4	11.5 $\pm$1.8	9.2 $\pm$1.3	2.8 $\pm$3.8	3.6 $\pm$3.8	15.6 $\pm$0.3	24.8 $\pm$3.1	17.2 $\pm$1.6
HySCDG [4]$_{CVPR25}$	44.3	65.0	52.7	8.4	90.3	15.4	9.6	**91.6**	17.3	15.1	69.5	24.9	**68.5**	9.8	17.2	29.2	65.2	25.5
SCM [61]$_{IGARSS24}$	37.2	19.7	25.7	19.5	45.3	27.3	23.5	34.7	28.0	15.8	23.5	18.9	11.9	27.4	16.6	21.6	30.1	23.3
DynEarth [31]$_{AAAI26}$	52.9	59.9	56.2	33.7	73.8	46.2	11.6	29.0	16.6	22.0	45.0	29.5	33.8	21.9	26.6	30.8	45.9	35.0
DynE. (DINOv3) [31]$_{AAAI26}$	70.2	52.8	60.3	**37.3**	73.2	**49.5**	14.8	27.7	19.3	28.3	41.6	33.7	23.9	14.7	18.2	34.9	42.0	36.2
S2C (DINOv3) [18]$_{AAAI26}$	**73.3** $\pm$2.8	25.3 $\pm$2.0	37.6 $\pm$2.5	21.7 $\pm$2.2	60.8 $\pm$5.9	32.0 $\pm$3.1	33.1 $\pm$4.6	45.5 $\pm$4.1	38.1 $\pm$3.2	**50.6** $\pm$2.0	54.0 $\pm$3.9	52.2 $\pm$2.8	12.7 $\pm$8.1	3.2 $\pm$2.3	5.1 $\pm$3.6	41.4 $\pm$4.2	42.1 $\pm$5.9	36.5 $\pm$4.5
Ours	72.3 $\pm$1.2	52.3 $\pm$3.7	**60.6** $\pm$2.2	25.9 $\pm$1.2	78.5 $\pm$4.2	38.9 $\pm$1.2	**43.2** $\pm$4.9	78.5 $\pm$6.1	**55.4** $\pm$2.9	46.3 $\pm$1.7	66.4 $\pm$6.8	**54.3** $\pm$1.5	34.2 $\pm$2.1	60.3 $\pm$2.8	**43.5** $\pm$1.0	**44.4** $\pm$1.9	67.2 $\pm$3.8	**50.6** $\pm$0.4

Table 3. Different noise generation techniques

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg.*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
Fully latent space generation (Ours)	72.3 $\pm$1.2	52.3 $\pm$3.7	60.6 $\pm$2.2	25.9 $\pm$1.2	78.5 $\pm$4.2	38.9 $\pm$1.2	43.2 $\pm$4.9	78.5 $\pm$6.1	55.4 $\pm$2.9	46.3 $\pm$1.7	66.4 $\pm$6.8	54.3 $\pm$1.5	34.2 $\pm$2.1	60.3 $\pm$2.8	43.5 $\pm$1.0	44.4 $\pm$1.9	67.2 $\pm$3.8	50.6 $\pm$0.4
Random rectangles mask	77.0 $\pm$1.2	39.9 $\pm$3.5	52.5 $\pm$2.7	25.9 $\pm$0.5	65.1 $\pm$6.8	36.9 $\pm$1.3	47.9 $\pm$3.1	63.3 $\pm$4.8	54.3 $\pm$0.8	50.1 $\pm$2.2	54.9 $\pm$5.5	52.2 $\pm$1.7	38.5 $\pm$2.4	39.3 $\pm$2.8	38.8 $\pm$0.4	47.9 $\pm$1.8	52.5 $\pm$3.9	46.9 $\pm$0.8
No Perlin mask for changed	88.5 $\pm$0.7	20.1 $\pm$2.6	32.7 $\pm$3.4	26.7 $\pm$0.9	55.9 $\pm$5.6	36.1 $\pm$1.2	58.0 $\pm$3.5	51.2 $\pm$4.8	54.2 $\pm$2.6	60.0 $\pm$1.3	37.5 $\pm$3.2	46.1 $\pm$2.1	44.0 $\pm$2.2	8.0 $\pm$1.4	13.5 $\pm$1.9	55.5 $\pm$1.4	34.6 $\pm$2.8	36.5 $\pm$1.7
Irrelevant changes with CycGAN	62.6 $\pm$1.7	46.5 $\pm$3.3	53.3 $\pm$1.6	13.9 $\pm$1.0	95.0 $\pm$1.7	24.3 $\pm$1.5	20.4 $\pm$2.4	56.7 $\pm$4.1	29.9 $\pm$2.2	26.2 $\pm$1.2	63.0 $\pm$2.1	37.0 $\pm$1.0	44.8 $\pm$2.6	36.7 $\pm$4.4	40.1 $\pm$1.9	33.6 $\pm$1.0	59.6 $\pm$1.8	36.9 $\pm$0.7
No dynamic calibration	28.3 $\pm$1.8	95.9 $\pm$2.2	43.7 $\pm$1.9	6.0 $\pm$0.4	99.5 $\pm$0.4	11.3 $\pm$0.7	8.8 $\pm$1.0	96.4 $\pm$2.5	16.1 $\pm$1.6	9.4 $\pm$0.8	97.5 $\pm$1.7	17.1 $\pm$1.2	26.1 $\pm$4.3	74.3 $\pm$7.4	38.3 $\pm$3.9	15.7 $\pm$1.6	92.7 $\pm$2.7	25.3 $\pm$1.9
Pixel space generation	71.1 $\pm$2.9	1.7 $\pm$1.1	3.3 $\pm$2.0	21.6 $\pm$0.6	3.3 $\pm$2.0	5.5 $\pm$2.9	11.4 $\pm$0.7	58.2 $\pm$13.9	18.8 $\pm$0.8	41.1 $\pm$1.9	5.0 $\pm$1.8	8.8 $\pm$2.9	40.6 $\pm$5.0	21.6 $\pm$5.2	27.5 $\pm$3.2	37.2 $\pm$1.4	17.9 $\pm$4.2	12.8 $\pm$1.8
No irrelevant changes	22.8 $\pm$0.8	67.1 $\pm$19.6	33.7 $\pm$3.2	4.2 $\pm$0.5	54.2 $\pm$23.1	7.8 $\pm$1.1	6.2 $\pm$0.4	75.3 $\pm$19.8	11.5 $\pm$0.9	6.5 $\pm$0.6	63.8 $\pm$20.9	11.7 $\pm$1.3	7.3 $\pm$1.3	92.5 $\pm$5.8	13.5 $\pm$2.1	9.4 $\pm$0.3	70.6 $\pm$17.1	15.6 $\pm$0.8

Table 4. Different backbone ablation.

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg.*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
ViT-L (DinoV3)(Ours)	72.3 $\pm$1.2	52.3 $\pm$3.7	60.6 $\pm$2.2	25.9 $\pm$1.2	78.5 $\pm$4.2	38.9 $\pm$1.2	43.2 $\pm$4.9	78.5 $\pm$6.1	55.4 $\pm$2.9	46.3 $\pm$1.7	66.4 $\pm$6.8	54.3 $\pm$1.5	34.2 $\pm$2.1	60.3 $\pm$2.8	43.5 $\pm$1.0	44.4 $\pm$1.9	67.2 $\pm$3.8	50.6 $\pm$0.4
ViT-L (DinoV2)	65.6 $\pm$3.5	67.4 $\pm$9.3	66.0 $\pm$3.0	20.8 $\pm$1.5	76.3 $\pm$11.0	32.5 $\pm$0.8	42.8 $\pm$6.1	79.7 $\pm$9.2	55.0 $\pm$3.3	29.1 $\pm$3.9	86.2 $\pm$4.8	43.3 $\pm$3.7	28.3 $\pm$4.0	73.3 $\pm$7.4	40.4 $\pm$2.9	37.3 $\pm$3.6	76.6 $\pm$8.0	47.4 $\pm$1.5
ViT-B (DinoV3)	73.7 $\pm$2.0	43.2 $\pm$7.8	54.0 $\pm$6.0	25.4 $\pm$3.1	84.3 $\pm$8.5	38.7 $\pm$2.6	37.9 $\pm$4.0	45.2 $\pm$19.5	38.7 $\pm$7.4	43.9 $\pm$5.5	64.1 $\pm$13.0	51.1 $\pm$2.0	40.6 $\pm$4.3	50.5 $\pm$6.3	44.6 $\pm$0.8	44.3 $\pm$3.5	57.5 $\pm$10.6	45.4 $\pm$2.3
Swin Tiny	51.9 $\pm$4.3	65.1 $\pm$7.4	57.4 $\pm$1.9	16.2 $\pm$0.6	94.9 $\pm$1.8	27.7 $\pm$0.8	26.3 $\pm$4.3	73.3 $\pm$2.9	38.5 $\pm$4.4	24.2 $\pm$1.8	67.2 $\pm$5.4	35.5 $\pm$1.8	38.7 $\pm$1.3	67.8 $\pm$2.3	49.3 $\pm$0.5	31.5 $\pm$1.3	73.7 $\pm$1.7	41.7 $\pm$1.2
ResNet50	43.0 $\pm$1.0	54.3 $\pm$3.3	47.9 $\pm$0.8	14.3 $\pm$0.8	87.2 $\pm$5.0	24.6 $\pm$1.1	22.6 $\pm$0.8	66.2 $\pm$5.6	33.7 $\pm$1.0	17.4 $\pm$0.5	54.9 $\pm$6.8	26.3 $\pm$1.0	24.7 $\pm$2.0	84.8 $\pm$2.8	38.1 $\pm$2.0	24.4 $\pm$0.6	69.5 $\pm$3.2	34.1 $\pm$0.5

Table 5. Different layer ablation.

Noise on layer				SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg.*		
7	11	15	23	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
✓	✓	✓	✓	72.3 $\pm$1.2	52.3 $\pm$3.7	60.6 $\pm$2.2	25.9 $\pm$1.2	78.5 $\pm$4.2	38.9 $\pm$1.2	43.2 $\pm$4.9	78.5 $\pm$6.1	55.4 $\pm$2.9	46.3 $\pm$1.7	66.4 $\pm$6.8	54.3 $\pm$1.5	34.2 $\pm$2.1	60.3 $\pm$2.8	43.5 $\pm$1.0	44.4 $\pm$1.9	67.2 $\pm$3.8	50.6 $\pm$0.4
✓	✓	✓		70.5 $\pm$1.6	53.0 $\pm$3.8	60.4 $\pm$1.9	26.4 $\pm$1.2	72.0 $\pm$8.6	38.5 $\pm$1.3	36.5 $\pm$2.6	84.5 $\pm$4.4	50.9 $\pm$2.0	43.2 $\pm$1.7	70.6 $\pm$1.6	53.6 $\pm$1.1	27.8 $\pm$2.1	71.7 $\pm$3.8	40.0 $\pm$1.6	40.9 $\pm$1.1	70.3 $\pm$3.2	48.6 $\pm$0.3
✓	✓		✓	63.9 $\pm$2.3	71.9 $\pm$3.9	67.5 $\pm$0.5	20.3 $\pm$1.9	94.3 $\pm$2.1	33.3 $\pm$2.5	23.4 $\pm$2.3	94.5 $\pm$1.7	37.4 $\pm$2.8	33.1 $\pm$1.5	85.9 $\pm$1.4	47.8 $\pm$1.4	24.5 $\pm$2.3	77.2 $\pm$3.8	37.1 $\pm$2.4	33.0 $\pm$1.9	84.8 $\pm$2.4	44.6 $\pm$1.6
✓		✓	✓	66.5 $\pm$1.6	67.4 $\pm$3.4	66.9 $\pm$1.0	21.4 $\pm$2.0	92.1 $\pm$3.2	34.7 $\pm$2.4	27.8 $\pm$2.9	92.2 $\pm$2.5	42.6 $\pm$3.3	34.9 $\pm$1.7	84.1 $\pm$1.9	49.3 $\pm$1.4	25.2 $\pm$2.4	75.0 $\pm$4.1	37.6 $\pm$2.2	35.2 $\pm$1.9	82.1 $\pm$2.8	46.2 $\pm$1.6
	✓	✓	✓	69.8 $\pm$1.5	60.1 $\pm$4.3	64.5 $\pm$1.9	23.4 $\pm$1.9	87.1 $\pm$5.3	36.8 $\pm$1.9	34.7 $\pm$4.0	85.8 $\pm$5.6	49.2 $\pm$3.2	40.0 $\pm$1.5	78.0 $\pm$2.9	52.9 $\pm$0.8	28.9 $\pm$2.9	67.8 $\pm$5.1	40.3 $\pm$2.1	39.4 $\pm$2.2	75.8 $\pm$4.2	48.7 $\pm$1.2

Table 6. Noise sampling dimension ablation.

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg.*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
Per channel in batch	72.3 $\pm$1.2	52.3 $\pm$3.7	60.6 $\pm$2.2	25.9 $\pm$1.2	78.5 $\pm$4.2	38.9 $\pm$1.2	43.2 $\pm$4.9	78.5 $\pm$6.1	55.4 $\pm$2.9	46.3 $\pm$1.7	66.4 $\pm$6.8	54.3 $\pm$1.5	34.2 $\pm$2.1	60.3 $\pm$2.8	43.5 $\pm$1.0	44.4 $\pm$1.9	67.2 $\pm$3.8	50.6 $\pm$0.4
Per channel in sample	74.8 $\pm$2.9	14.9 $\pm$8.1	24.2 $\pm$10.7	28.3 $\pm$4.6	14.0 $\pm$17.3	15.3 $\pm$13.1	59.0 $\pm$3.9	27.3 $\pm$18.0	34.8 $\pm$16.0	53.4 $\pm$2.3	25.7 $\pm$12.3	33.4 $\pm$10.9	50.2 $\pm$6.3	35.4 $\pm$10.5	40.2 $\pm$4.2	53.1 $\pm$1.6	23.5 $\pm$13.0	29.6 $\pm$10.7
Per sample	64.0 $\pm$2.6	49.8 $\pm$10.6	55.4 $\pm$5.5	23.0 $\pm$2.3	67.2 $\pm$16.8	33.9 $\pm$2.8	29.5 $\pm$6.1	75.9 $\pm$13.6	41.4 $\pm$4.8	36.9 $\pm$2.6	68.2 $\pm$10.2	47.5 $\pm$1.6	37.1 $\pm$6.4	55.6 $\pm$10.5	43.4 $\pm$2.0	38.1 $\pm$3.8	63.4 $\pm$11.7	44.3 $\pm$0.8
Per batch	45.7 $\pm$2.4	84.3 $\pm$3.1	59.2 $\pm$1.4	13.8 $\pm$1.3	97.2 $\pm$1.1	24.1 $\pm$2.0	9.7 $\pm$0.8	98.2 $\pm$1.0	17.7 $\pm$1.4	17.7 $\pm$0.9	93.3 $\pm$1.2	29.7 $\pm$1.3	21.6 $\pm$2.7	82.4 $\pm$4.4	34.0 $\pm$3.1	21.7 $\pm$1.6	91.1 $\pm$2.1	32.9 $\pm$1.7

H.1. Protocol Implementation Details

3 different computational efficiency metrics are reported: parameter count, inference time (also expressed as frames per second - FPS) and GFLOPs. GFLOPs are measured using the official PyTorch profiler.

For inference time, we use a pair of 256×256 RGB float16 images as input. Models are cast to float16 where supported. We perform 1000 warm-up forward passes, followed by 1000 timed forward passes. This procedure is repeated five times, and the average runtime per forward pass is reported. All measurements are conducted on an NVIDIA A100-SXM4 40GB GPU.

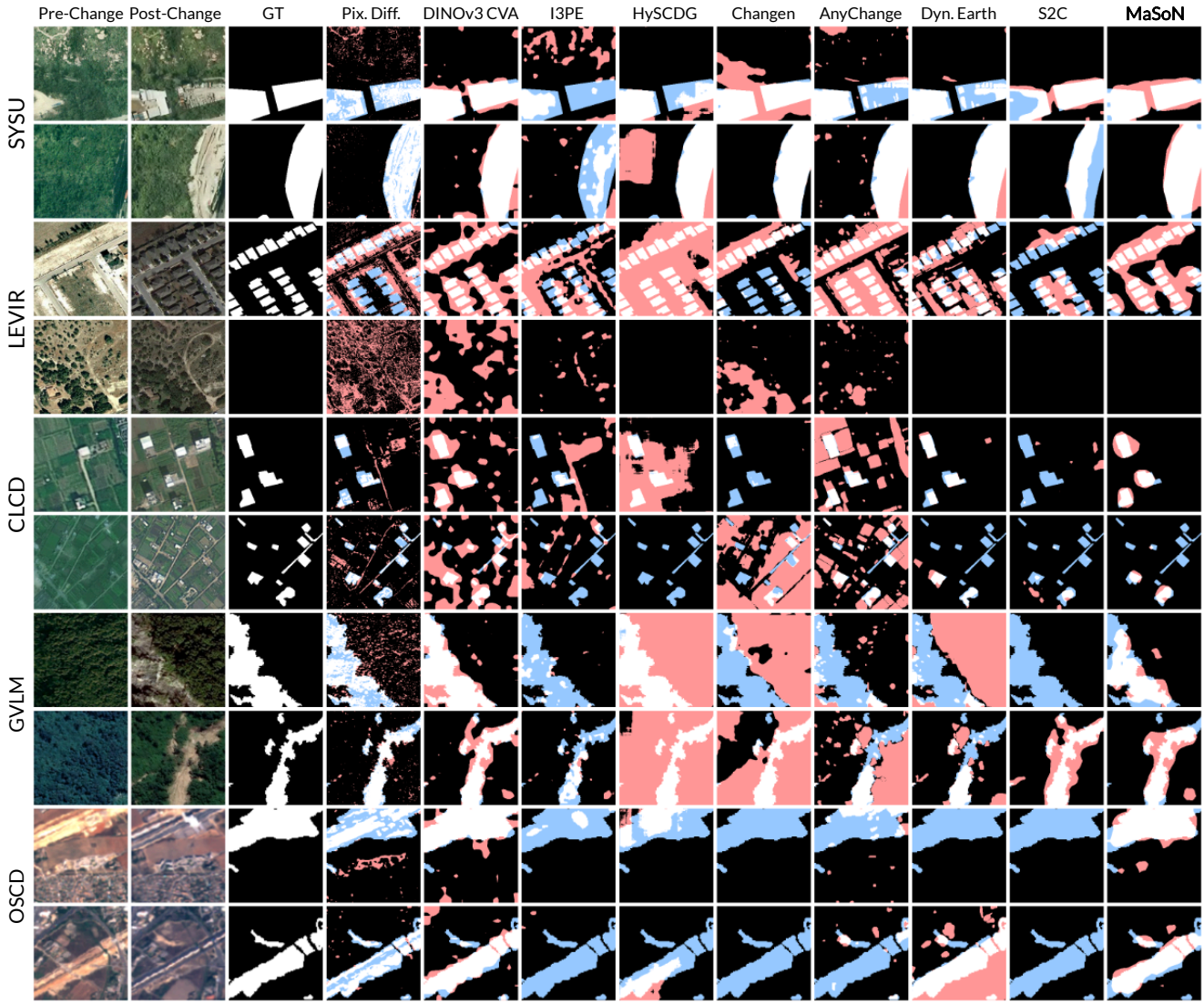

Figure 11. Qualitative comparison of the predictions made by our method and related methods. The pair of considered images is shown in the first and second columns, followed by the ground truth mask and predictions for each method. False positives are marked in red and false negatives in blue.

Table 7. Supervised change detection results across five diverse datasets and their average. We report Precision (Pr.), Recall (Re.), and F1 score.

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
FCS-Diff [14]	83.5 ±0.4	61.5 ±0.3	70.8 ±0.1	83.0 ±0.9	80.6 ±2.0	81.8 ±0.7	82.7 ±3.1	67.6 ±2.5	74.3 ±0.8	54.0 ±1.0	54.3 ±2.6	54.1 ±1.2	27.8 ±1.1	**68.5** ±6.8	39.4 ±1.4	66.2 ±1.0	66.5 ±2.2	64.1 ±0.4
BIT [9]	79.1 ±2.3	75.1 ±2.1	77.0 ±0.1	92.0 ±0.4	88.1 ±0.2	90.0 ±0.1	89.8 ±1.2	87.9 ±1.1	88.8 ±0.1	66.5 ±5.8	60.8 ±1.3	63.4 ±2.0	48.7 ±1.9	37.9 ±2.2	42.6 ±2.0	75.2 ±2.2	70.0 ±0.8	72.4 ±0.6
ChFormer [1]	82.8 ±0.4	73.5 ±1.3	77.9 ±0.5	91.7 ±0.2	87.3 ±0.5	89.5 ±0.2	90.4 ±0.6	86.6 ±0.4	88.5 ±0.2	61.4 ±1.3	60.4 ±5.1	60.8 ±2.6	60.2 ±0.5	40.1 ±1.1	48.1 ±1.0	77.3 ±0.2	69.6 ±1.1	73.0 ±0.5
BiFA [71]	87.4 ±0.2	**80.4** ±0.6	**83.8** ±0.3	90.9 ±1.1	88.1 ±0.9	89.5 ±0.9	89.9 ±0.7	88.2 ±1.0	89.0 ±0.2	79.4 ±2.3	70.1 ±0.7	74.5 ±0.6	61.5 ±2.3	27.1 ±4.4	37.4 ±3.7	81.8 ±0.7	70.8 ±1.1	74.8 ±0.9
SwinSUNet [70]	89.2 ±0.6	67.2 ±2.3	76.6 ±1.7	86.9 ±0.1	**91.7** ±0.3	89.3 ±0.1	88.2 ±1.8	**92.0** ±1.0	90.0 ±0.5	79.5 ±1.6	72.5 ±2.6	75.8 ±0.8	61.7 ±1.3	46.3 ±5.3	52.8 ±3.0	81.1 ±0.9	73.9 ±1.4	76.9 ±0.4
ChangeMamba [11]	89.6 ±0.3	74.7 ±0.6	81.5 ±0.3	92.4 ±0.2	91.2 ±0.1	91.8 ±0.1	91.2 ±0.9	90.4 ±0.8	**90.8** ±0.1	**87.3** ±0.6	74.4 ±1.5	80.3 ±1.1	**63.4** ±3.7	36.1 ±3.2	45.8 ±1.8	**84.8** ±0.5	73.4 ±0.6	78.0 ±0.5
CaCo [39]	88.4 ±0.7	73.4 ±0.3	80.2 ±0.4	92.0 ±0.1	89.9 ±0.2	90.9 ±0.0	**91.4** ±0.1	89.2 ±0.2	90.3 ±0.0	83.9 ±0.8	72.6 ±1.2	77.8 ±0.5	54.5 ±4.1	49.8 ±2.8	51.9 ±0.9	82.0 ±0.6	75.0 ±0.4	78.2 ±0.0
SatMAE [12]	89.0 ±0.0	75.0 ±0.1	81.4 ±0.1	92.6 ±0.1	90.2 ±0.1	91.4 ±0.1	90.2 ±0.2	89.3 ±0.4	89.8 ±0.2	84.0 ±1.2	74.9 ±0.5	79.2 ±0.8	53.8 ±1.5	45.9 ±2.7	49.5 ±1.2	82.0 ±0.3	75.1 ±0.6	78.3 ±0.2
GFM [40]	89.7 ±0.3	74.3 ±0.6	81.2 ±0.2	90.8 ±0.2	88.8 ±0.1	89.8 ±0.1	90.5 ±0.1	89.2 ±0.1	89.8 ±0.1	82.2 ±1.0	73.2 ±0.9	77.5 ±0.9	55.9 ±0.5	52.5 ±2.1	54.1 ±1.4	81.8 ±0.3	75.6 ±0.5	78.5 ±0.4
MTP [64]	88.5 ±0.5	75.2 ±0.9	81.3 ±0.3	92.8 ±0.1	90.7 ±0.1	91.7 ±0.0	90.8 ±0.5	89.0 ±0.4	89.9 ±0.2	85.4 ±0.9	75.8 ±0.3	80.3 ±0.3	43.9 ±2.4	66.5 ±4.4	52.8 ±0.4	80.3 ±0.7	**79.4** ±1.0	79.2 ±0.1
Ours	**90.7** ±0.1	77.6 ±0.6	83.7 ±0.4	**92.9** ±0.1	91.2 ±0.1	**92.1** ±0.1	91.1 ±0.2	89.4 ±0.2	90.3 ±0.2	86.8 ±0.9	**77.3** ±1.0	**81.8** ±0.7	57.0 ±0.8	57.5 ±3.9	**57.2** ±1.9	83.7 ±0.3	78.6 ±0.6	**81.0** ±0.2

H.2. Additional Results

The results for unsupervised methods are reported in Table 8. Our method outperforms the second-best Dynam-

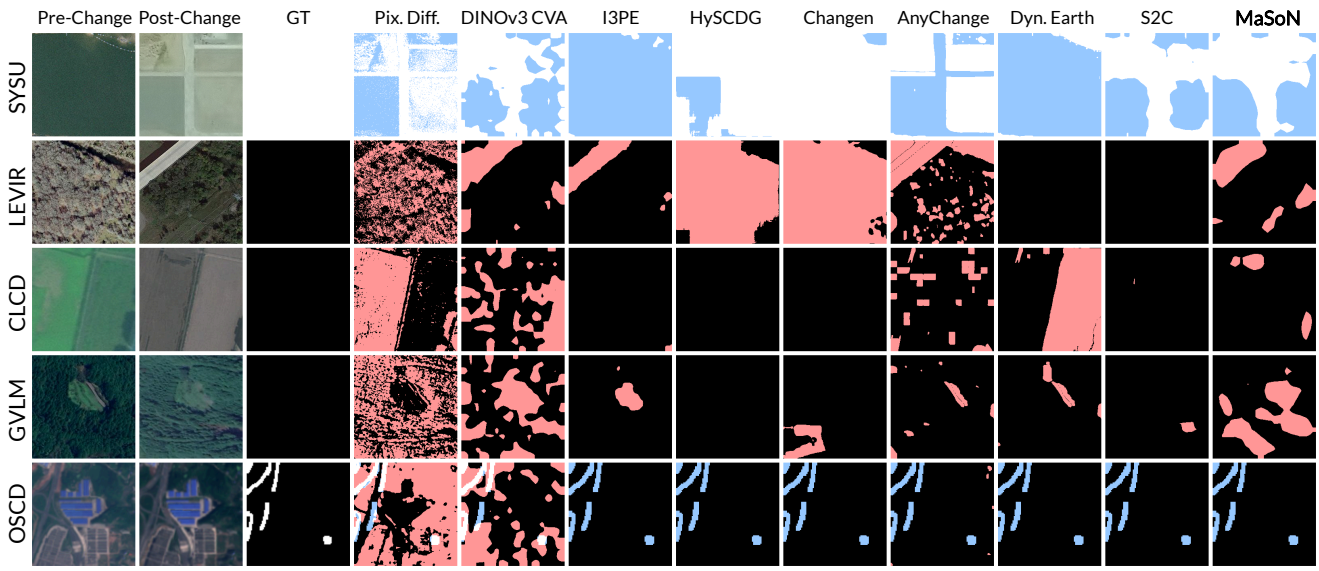

Figure 12. Failure cases made by our method and related methods.

icEarth [31] by 14.4 percentage points of F1 score, while having half the parameters and offering significantly faster inference speed. Pixel differencing still runs significantly faster than any other method and has the benefit of having no parameters. However, the performance it achieves is substantially below the best deep learning methods.

We can also see that DINO-based CVA is slower than our model. This comes from the fact that CVA is a post-processing step on DINO features, essentially replacing our UPerNet decoder. It relies on norm calculation and Otsu threshold, which are not as fast as a simple forward decoder pass. This shows the benefit of fully end-to-end solutions.

Table 8. Computational efficiency results for each model. We report FPS (derived from inference time), inference time, parameter count, GFLOPs, and average Precision, Recall, and F1 across 5 datasets. All results were obtained using an Nvidia A100-SXM4 40GB GPU.

	FPS [img/s]	Inference Time [ms]	Param. [M]	FLOPS [10^9]	*Avg. Change Detection* Pr.	Re.	F1
Pixel Difference	456.4±54.5	2.2±0.2	0.0	0.0	16.2	49.3	22.0
DCVA [55]$_{TGRS19}$	0.4±0.0	2824.9±6.1	0.5	21.9	30.0	9.3	9.6
DINOv2-CVA	24.4±0.3	41.0±0.5	303.1	316.2	15.7	**85.1**	25.0
DINOv3-CVA	24.4±0.3	41.0±0.5	303.1	316.2	19.0	80.4	28.7
AnyChange [75]$_{NeurIPS24}$	0.4±0.0	2234.7±0.2	641.1	5789.3	26.0	43.0	29.0
Changen2-S9 [76]$_{TPAMI25}$	33.3±0.1	30.0±0.1	99.9	197.6	17.3	39.4	20.2
CDRL* [44]$_{CVPRW22}$	130.8±0.1	7.6±0.0	54.5	8.9	11.2±0.2	3.1±0.0	3.7±0.0
CDRL-SA* [45]$_{RSL24}$	0.3±0.0	3055.1±0.8	682.5	17742.1	28.4±4.2	1.0±0.2	1.9±0.3
I3PE [10]$_{ISPRS23}$	65.9±0.2	15.2±0.0	24.9	35.2	15.6±0.3	24.8±3.1	17.2±1.6
HySCDG [4]$_{CVPR25}$	41.0±0.1	24.4±0.0	65.1	64.8	29.2	65.2	25.5
SCM [61]$_{IGARSS24}$	1.2±0.0	817.0±16.8	223.5	423.2	21.6	30.1	23.3
DynEarth [31]$_{AAAI26}$	0.3±0.0	3676.7±0.5	877.6	19541.3	30.8	45.9	35.0
DynE. (DINOv3) [31]$_{AAAI26}$	0.3±0.0	3702.8±0.8	877.6	19542.6	34.9	42.0	36.2
S2C (DINOv3) [18]$_{AAAI26}$	3.8±0.0	260.7±0.6	303.6	1266.8	41.4 ±4.2	42.1±5.9	36.5 ±4.5
Ours	39.7±0.1	25.2±0.1	337.5	332.8	**44.4** ±1.9	67.2 ±3.8	**50.6** ±0.4

H.3. Discussion on Noise Generation Overhead

The noise generation strategy introduces a small training overhead of approximately 20 milliseconds per single pass. As mentioned in the main paper, the entire training run for each dataset lasts only 7 minutes. This is relatively little compared to the hundreds of hours needed to train diffusion models like HySCDG [4] (as per Appendix A of HySCDG paper).

I. Additional Ablation Studies

Effect of Batch Size The calculation of irrelevant and relevant noise relies on the images within a batch. This outcome is dependent on the batch size; therefore, we examine how batch size influences downstream performance. The findings are presented in Table 9. It is evident that MaSoN demonstrates a reasonable robustness to this parameter.

I.1. Relevant noise sampling source

Our model samples relevant noise from the concatenation of features, but it can alternatively sample from the feature difference, as is the case for irrelevant noise. We evaluate this variant in Table 10. While sampling from the feature difference is necessary to capture intra-image variability, we find that for relevant noise, it is more effective to sample directly from the features. We hypothesise that this is because relevant changes should manifest as higher-magnitude variations, yet remain constrained within the realistic bounds of the underlying feature distribution. However, the difference in results is only minor, indicating that both options are viable as long as the sampled distribution is dynamically scaled and broader than the distribution of irrelevant

Table 9. Effect of batch size ablation.

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg.*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
BS 16 (Ours)	72.3 ±1.2	52.3 ±3.7	60.6 ±2.2	25.9 ±1.2	78.5 ±4.2	38.9 ±1.2	43.2 ±4.9	78.5 ±6.1	55.4 ±2.9	46.3 ±1.7	66.4 ±6.8	54.3 ±1.5	34.2 ±2.1	60.3 ±2.8	43.5 ±1.0	44.4 ±1.9	67.2 ±3.8	50.6 ±0.4
BS 8	72.2 ±3.4	54.7 ±7.7	61.8 ±3.3	25.3 ±1.5	80.4 ±5.9	38.4 ±1.3	39.1 ±6.1	82.2 ±7.4	52.4 ±4.5	37.5 ±5.2	78.9 ±7.0	50.4 ±3.5	31.8 ±4.0	63.1 ±6.8	41.9 ±2.3	41.2 ±3.8	71.9 ±6.2	49.0 ±1.6
BS 32	74.5 ±1.7	45.3 ±7.0	56.0 ±4.7	27.6 ±1.3	69.6 ±9.9	39.3 ±1.1	46.2 ±6.2	72.1 ±10.9	55.5 ±2.1	46.2 ±2.4	64.2 ±8.2	53.4 ±1.6	36.0 ±3.1	58.3 ±4.7	44.3 ±0.9	46.1 ±2.8	61.9 ±7.9	49.7 ±0.9

changes (refer to Section 3.2).

I.2. Multispectral and SAR Data

To verify that our contributions go beyond the RGB domain, we have also evaluated MaSoN on multispectral (MS) and SAR data. We use DEO [66] backbone for multispectral and CopernicusFM [65] for SAR. Most hyperparameters remain the same, except for backbone-dependent settings, for example, layer-ids (for swin we use all 4, and for ViT-based Cop-FM we use [3, 5, 7, 11]) and input normalisation parameters, which we take from the authors. The results for MS are shown in Table 11 and for SAR in the original paper. The model achieves superior performance compared to RGB models on OSCD-multispectral. The same holds for SAR, where only pixel-differencing and CVA directly support this modality. Other related methods cannot be easily extended to new modalities. This is especially restrictive in the SAM [28] based methods, since SAM is a purely RGB foundation model.

I.3. Different Noise Distribution

To verify how important the random distribution is from which we sample both the irrelevant and relevant noise, we have exchanged each with the Laplace random distribution. The results can be seen in Table 12. The probability distribution plays an important role. Additionally, it is important to point out that all of these results are still above the previous state-of-the-art.

J. Extended Implementation Details

In this section, we present additional implementation details. Additional details for our model are in Appendix J.1, for related unsupervised methods in Appendix J.2, for supervised in Appendix J.3, and finally, ablation study experiments with additional details in Appendix J.4. For implementation details of computational efficiency experiments, refer to Appendix H (including the discussion on computational overhead of noise generation strategy).

J.1. Our model

We use pretrained weights from Huggingface in our experiments. The DINOv3 [58] pretrained ViT-L is the following: facebook: dinov3-vitl16-pretrain-lvd1689m. The main hyperparameters are listed in Section 4 of the main paper. Here we list some additional details. In the case of supervised learning, we train the model for 100 epochs instead of iterations, with a learning rate of 10^{-4} and a batch size of 32, while all other parameters stay the same. Rotation and flip augmentations are always applied with a chance of 30 %. Each synthetic change is applied with a chance of 50 %. The Perlin mask is sampled at full image dimension (256 x 256), thresholded at 0.5 and is then bilinearly downscaled to feature dimension (16x16 with DINOv3). All code is implemented with PyTorch and PyTorch Lightning. The metrics implementation comes from torchmetrics. All used code will be made available on GitHub upon acceptance.

J.2. Unsupervised Related Methods

We use official code, provided by the authors, for all methods and just integrate our datasets. The following are versions of their code on GitHub:

- DCVA [55]: dcvaVHROptical, commit: e1d6af8365ca27062635cfa6fadee91984302e1e
- AnyChange [75]: pytorch-change-models, commit: 9c564feecfb577069b9a9bc5859264f768581046
- Changen2 [76]: pytorch-change-models commit: 9c564feecfb577069b9a9bc5859264f768581046
- CDRL [44]: CDRL commit: 9305057f00acfc1472673005c56949a90a2eb6b7
- CDRL-SA [45]: CDRL-SA commit: a2e1ea203c455e28631bbca3bcbfc78f53420869
- I3PE [10]: I3PE commit: 3182c2918bd32a5b7dd44dc7ee71fe09ab92ed7b
- HySCDG [4]: HySCDG commit: d3e93feb7d4a59b723655186e47f9edb2de9acad
- SCM [61]: UCD-SCM commit: 72a6c3cf5c336604be4f6a8dc5e2401288e1779a
- DynamicEarth [31]: DynamicEarth commit: c9ffd90cafbd791cd75a48a5717a902966c2436c
- S2C [18]: S2C commit: 7bcec8658af25bfed87bc0be2109c74e5106a677

We keep all the hyperparameters the same as set by the authors. Since the weights for CycleGAN in CDRL are not available, we train our own with the official code using the default options.

J.3. Supervised Related Methods

Remote Sensing Foundation Models We use the official code and pre-trained weights provided by the authors for all remote sensing foundation models. The specific versions of the code used in our experiments can be accessed here:

Table 10. Relevant noise statistics sampling source results.

	SYSU			LEVIR			GVLM			CLCD			OSCD			*Avg.*		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
Feat. concat(Ours)	72.3 ±1.2	52.3 ±3.7	60.6 ±2.2	25.9 ±1.2	78.5 ±4.2	38.9 ±1.2	43.2 ±4.9	78.5 ±6.1	55.4 ±2.9	46.3 ±1.7	66.4 ±6.8	54.3 ±1.5	34.2 ±2.1	60.3 ±2.8	43.5 ±1.0	44.4 ±1.9	67.2 ±3.8	50.6 ±0.4
Feat. diff.	75.2 ±1.5	45.9 ±6.6	56.7 ±4.8	27.5 ±2.3	66.9 ±12.5	38.6 ±1.7	48.3 ±6.1	69.7 ±12.0	56.1 ±0.8	45.6 ±2.9	66.8 ±8.2	53.9 ±1.4	26.6 ±3.1	72.9 ±5.2	38.8 ±2.8	44.6 ±2.8	64.5 ±8.4	48.8 ±1.1

Table 11. Performance with RGB/MS OSCD data

	OSCD		
	Pr.	Re.	F1
Ours (RGB)	34.2	60.3	43.5
Ours (MS)	43.4	47.0	45.1

- CaCo [39]: CACo commit: 7c4fbc8d9e65a4ef715596c5c275ab3ddb6a8193
- SatMAE [12]: SatMAE commit: e31c11fa1bef6f9a9aa3eb49e8637c8b8952ba5e
- GFM [40]: GFM commit: 4dd248e8544b3b6a49f5173b0931d97a17a7f424
- MTP [64]: MTP commit: 962f7fd8781c095eb26db65ead3016e666b6d417

Since foundation models don't have a predefined change detection architecture, we adopt the authors' architecture code and load the weights as the *encoder* into our framework. The configuration is the same as for our model, with minor adjustments according to the authors' settings.

All methods use an initial learning rate of 10^{-4}, except for the ViT-based models MTP and SatMAE, which follow the MTP [64] setup with a learning rate of $6 \cdot 10^{-5}$. Weight decay is set to 0.05 for all methods, except for GFM, which uses $5 \cdot 10^{-4}$ as in [40]. Feature extraction for CaCo and GFM is performed across all four hierarchical levels. For MTP and SatMAE, we follow [64] and extract features from layers with IDs 3, 5, 7, and 11 (MTP) and 7, 11, 15, and 23 (SatMAE). All methods use the UPerNet [68] decoder, except ViT-based models, where UNet [53] yields better performance and aligns with MTP's original configuration. Other training settings, including Dice loss, cosine learning rate scheduling, and flip augmentations, remain consistent with our method.

Change Detection Specific Methods We use official code, provided by the authors, for all methods and just integrate our datasets. The following are versions of their code:

- FCS-Diff [14]: fully_convolutional_change_detection commit: 4dd83231f25319a7ebb16cbfa9912541ceabac9a
- BIT [9]: BIT_CD commit: adcd7aea6f234586ffffdd4e9959404f96271711
- ChangeFormer [1]: ChangeFormer commit: afd1b7ed640aa265a2c730de958416ae7356a2f9
- BiFA [71]: BiFA commit: 56cd0da461e5e4b0d6a9b4f3321f0a81a91d21b8
- SwinSUNet [70]: SwinSUNet commit: 721daf84238eda40fb49d626c21df4ed2246aa9e
- ChangeMamba [11]: ChangeMamba commit: a91b82ee45059ce159f5f6f5d8e5818c33b84e68

We keep all the hyperparameters the same as set by the authors, except the epoch number for BiFA and FC-Siam-Diff. For these two models, we halve the default number of epochs in the case of OSCD [15] dataset, since they start to overfit due to the dataset's small size.

J.4. Ablation Study Implementation Details

Sample Generation Strategy For the pixel-space irrelevant change generation experiment, we use the same images as the related method CDRL [44] produced by learned transformation using CycleGan [77]. It is trained with official options and following code.

Backbone ablation The following architectures and weights from Huggingface are used for architecture ablation:

- ViT-L DINOv3: facebook: dinov3-vitl16-pretrain-lvd1689m
- ViT-L DINOv2: facebook: dinov2-large
- ViT-B DINOv3: facebook: dinov3-vitb16-pretrain-lvd1689m
- Swin: facebook: mask2former-swin-tiny-ade-semantic
- ResNet50: facebook: maskformer-resnet50-ade20k-full

Multispectral and SAR ablation For multispectral encoder we use DEO [66] Swin B. For SAR, we use CopernicusFM [65] CopernicusFM_ViT_base_varlang_e100.pth from wangyi111: Copernicus-FM. We use the authors' normalisation in both cases. The rest of the architecture remains the same.

Extension to change filtering We integrate MaSoN as the change comparer in the pipeline of DynamicEarth [31] as a simple drop in replacement of raw DINOv2/v3. This means that the first (mask proposal) and last (filtering) phase remain unchanged, but we keep only the proposed masks that have at least 30% of overlap with masks predicted by MaSoN. We also use the same filtering prompts: "building" for buildings and "background" for everything else.

Table 12. Different noise distributions ablation

Irrel. noise	Rel. noise	SYSU Pr.	SYSU Re.	SYSU F1	LEVIR Pr.	LEVIR Re.	LEVIR F1	GVLM Pr.	GVLM Re.	GVLM F1	CLCD Pr.	CLCD Re.	CLCD F1	OSCD Pr.	OSCD Re.	OSCD F1	*Avg.* Pr.	*Avg.* Re.	*Avg.* F1
Gauss	Gauss	72.3 ±1.2	52.3 ±3.7	60.6 ±2.2	25.9 ±1.2	78.5 ±4.2	38.9 ±1.2	43.2 ±4.9	78.5 ±6.1	55.4 ±2.9	46.3 ±1.7	66.4 ±6.8	54.3 ±1.5	34.2 ±2.1	60.3 ±2.8	43.5 ±1.0	44.4 ±1.9	67.2 ±3.8	50.6 ±0.4
Gauss	Laplace	52.2 ±3.6	67.2 ±9.4	58.3 ±1.6	16.0 ±2.3	87.9 ±8.0	26.9 ±3.1	15.5 ±3.6	88.7 ±6.7	26.1 ±5.0	21.7 ±3.6	82.2 ±9.1	34.0 ±3.5	28.9 ±4.8	69.0 ±8.3	40.2 ±3.7	26.9 ±3.4	79.0 ±7.9	37.1 ±2.7
Laplace	Gauss	64.2 ±2.6	66.8 ±5.5	65.3 ±1.4	20.1 ±2.0	93.0 ±2.9	32.9 ±2.6	17.4 ±3.2	94.9 ±2.5	29.2 ±4.5	30.1 ±3.1	85.5 ±4.6	44.3 ±2.7	24.8 ±3.0	76.9 ±5.2	37.3 ±2.9	31.3 ±2.6	83.4 ±3.9	41.8 ±2.1
Laplace	Laplace	76.5 ±1.1	28.8 ±8.7	41.3 ±9.4	29.3 ±2.3	48.9 ±17.0	35.6 ±4.8	47.6 ±6.9	58.3 ±16.4	50.6 ±2.2	49.7 ±1.6	45.3 ±13.6	46.3 ±8.1	47.0 ±5.0	41.8 ±7.5	43.6 ±1.5	50.0 ±2.9	44.6 ±12.2	43.5 ±5.0