HYPERFLOW: CLASS-INCREMENTAL SEMANTIC CHANGE DETECTION VIA HYPERBOLIC FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic change detection in remote sensing aims to identify and characterize land-cover transitions observed in pairs of satellite or aerial images. Existing methods assume a fixed set of change categories, limiting their applicability in real-world scenarios where changes of interest may vary, for example across geographies or over time. In this work, we tackle the problem of class-incremental semantic change detection, where a model progressively learns to detect new types of semantic change. To do so, we exploit the hierarchical structure of land-cover semantics, where newly introduced categories correspond to fine-grained refinements of existing ones, and represent said semantic categories in hyperbolic space, which is particularly well suited to represent such hierarchical relationships. To capture semantic changes, we introduce HyperFlow, a method relying on Busemann prototypes, modeling change as an anomaly in a learned semantic flow within the hyperbolic embedding space induced by segmentation. HyperFlow couples change detection to class-incremental semantic segmentation, enabling class-incremental semantic change detection. We also present HiChange, a new large-scale, high resolution remote sensing dataset with detailed, fine-grained hierarchical labels. Experimental results demonstrate that HyperFlow achieves both strong change detection and semantic segmentation performance across incremental steps. In particular, we show that progressively refining semantic categories, rather than considering them all directly, yields improved change detection, highlighting the benefits of incremental learning for semantic change detection.

1 INTRODUCTION

Semantic change detection (SCD) aims at identifying and characterizing land cover transitions over time by comparing pairs of co-registered satellite or aerial images. Despite being a well established research field (Jiang et al., 2022) with applications in disaster response (Zheng et al., 2021; Ahn et al., 2025), urban planning (Wang et al., 2024; Zhao et al., 2025; Zheng et al., 2021) or land cover analysis (Li et al., 2016; Lv et al., 2023; Daudt et al., 2019), current approaches remain static and assume a fixed set of change categories. This assumption is misaligned with real-world scenarios, where changing environmental and societal priorities imply evolving monitoring objectives (Tasar et al., 2019; Weng et al., 2024; Zhang et al., 2023). Adapting models to new types of changes typically requires costly retraining and annotation, making current approaches unsuitable for long-term deployment. A key observation is that land-cover categories are naturally structured in hierarchies, from broad categories like *urban* or *vegetation* to fine-grained ones such as *residential building*, *crop field*, or *forest*. In the context of land use monitoring, new types of changes are rarely entirely new categories; rather, they usually correspond to refining coarse classes into more specific ones, as illustrated in fig. 1, panel (1). Leveraging this hierarchy is therefore essential to enable adaptable SCD methods and calls for change detection (CD) models that *(i)* explicitly represent hierarchical semantics and *(ii)* adapt to different levels of this hierarchy over time.

To tackle these challenges, we consider the task of class-incremental semantic change detection (CI-SCD), in which a model continuously expands the types of semantic changes it detects. First, we model land-cover semantics in hyperbolic space, which is particularly well suited to represent hierarchical structures (Nickel & Kiela, 2017; Ganea et al., 2018a;b). Second, we leverage advances in class-incremental semantic segmentation (CISS) to advance CI-SCD. We introduce a dynamic hyperbolic formulation of CI-SCD, where semantic categories are obtained from hierarchical em-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

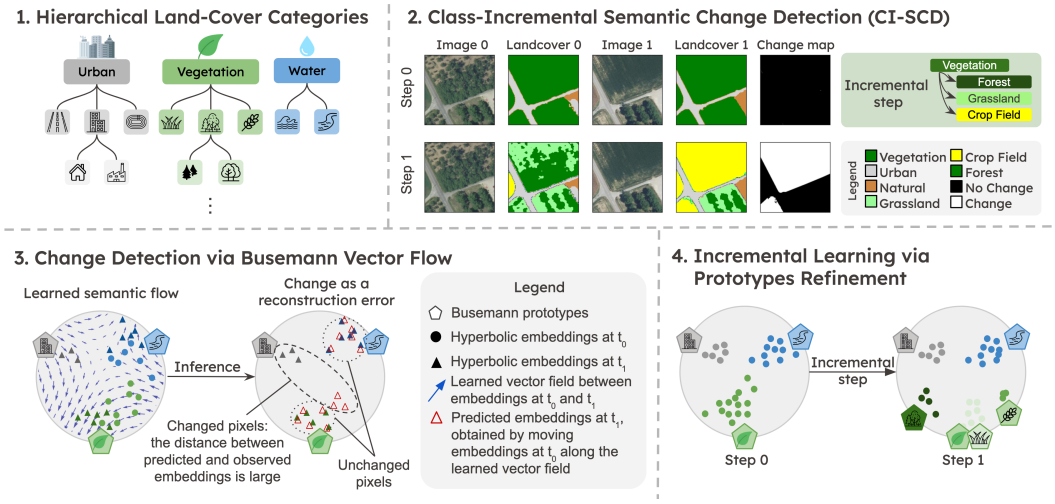


Figure 1: Overview of CI-SCD via Hyperbolic Flow. (1) Land-cover semantics are naturally organized in hierarchies. (2) During CI-SCD, the model leverages this taxonomy to progressively refine semantic categories from coarse to fine-grained classes. Change maps evolve to account for the progressively refined segmentation. (3) Hyperbolic semantic flow: a vector field transforming the embeddings of the first image into the embeddings of the second image is learned. The hyperbolic distance between the positions of the predicted embeddings and the observed positions indicates change. (4) Prototype-based incremental learning: we refine a coarse class by splitting it into leaves prototype nodes. The vector field in (3) is then retrained to adapt to the new semantic classes.

beddings in hyperbolic space, while change is modeled as an anomaly in a learned semantic flow within this structured space. This approach couples SCD to CISS, enabling CI-SCD, and ensures that the detected changes are adapted to the updated semantic classes (see fig. 1, panel (2)).

Specifically, we represent pixels at different time steps as points in a shared hyperbolic embedding space induced by the segmentation. We then model change as an anomaly in a learned semantic flow in this space, i.e., a displacement between embeddings at times t_0 and t_1: when a pixel changes (e.g., from *vegetation* to *building*), its embedding moves between regions of the manifold, producing a displacement in embedding space between t_0 and t_1, as illustrated in fig. 1, panel (3). By learning the expected flow accounting for small domain shifts naturally occurring between images (*e.g.* due to seasonal or illumination changes), the remaining deviations indicate the unexpected changes to be detected. To learn this semantic flow in hyperbolic space, we introduce HyperFlow, a method based on the Busemann function. Building on prior work in hyperbolic segmentation (Ghadimi Atigh et al., 2022; Berg et al., 2024), CISS (Hindel et al., 2024) and Riemannian vector fields (Katsman et al., 2023), we show that Busemann prototypes are the key to link this approach to CISS within a single hyperbolic method. In particular, incremental learning is handled directly through those prototypes, as depicted in fig. 1, panel (4): when a coarse category is refined, its prototype is split to initialize fine-grained subclasses, preserving geometric proximity to the parent while enabling specialization. The semantic flow is then retrained in this updated space, thus becoming specific to the newly introduced subclasses and allowing the detection of new semantic transitions.

In summary, this work makes three main contributions. First, we propose HyperFlow, a method that represents semantic categories using Busemann prototypes and models change via a semantic flow in a shared embedding space, coupling segmentation and CD in a geometrically grounded manner. Second, we show how HyperFlow can be leveraged to address the task of CI-SCD by extending CISS methods. In this formulation, incremental refinement is achieved by splitting parent prototypes to initialize newly introduced subclasses, preserving hierarchical structure while ensuring that the semantic flow, retrained to account for this new embedding space, adapts consistently as the taxonomy evolves. Third, as current benchmarks in SCD are limited to a small number of generic semantic categories (Zhang et al., 2025) lacking hierarchical structure, we present HiChange, a large-scale dataset with a rich hierarchical label structure designed to enable CI-SCD models to detect fine grained changes of increasing semantic complexity.

2 Related Work

2.1 Semantic Change Detection in Remote Sensing

CD enables the identification of differences between remotely sensed images of the same location acquired at different times. Initially based on hand-crafted features (Malila, 1980; Volpi et al., 2012), recent methods leverage deep learning, typically with Siamese designs (Daudt et al., 2018; Jiang et al., 2022). However, such approaches rely on the quality of ground-truth change maps, which are difficult to obtain and often noisy (Daudt et al., 2019). To overcome the need for accurate change labels, unsupervised CD has been widely studied (Malila, 1980; Nielsen & Conradsen, 1997; Volpi et al., 2012), and is currently framed as anomaly detection, linking changes to high reconstruction errors (Touati et al., 2020; Andresini et al., 2022; Béchaz et al., 2026; Růžička et al., 2022; Noh et al., 2022; Liu et al., 2021; Luppino et al., 2021). While most methods focus on binary change detection (BCD), real world scenarios require specifying the semantics of change. SCD therefore extends BCD by assigning semantic labels to changes (Daudt et al., 2019). Early SCD methods used post-segmentation comparisons (Huang et al., 2010; Hu & Zhang, 2013; Wu et al., 2017) and suffered from error accumulation and lack temporal modeling (Volpi et al., 2013). Deep learning SCD methods have since become dominant and adopt end-to-end architectures that couple semantic segmentation and change detection (Daudt et al., 2019; Zhu et al., 2022; Ding et al., 2024). Despite this progress, existing SCD methods remain rigid: they assume fixed categories, and treat them as independent labels, ignoring the hierarchical structure that naturally underlies land-cover semantics. These assumptions are unrealistic in real-world settings, where categories are progressively refined and semantically nested, and motivate our study of CI-SCD.

2.2 Class-Incremental Semantic Segmentation

CISS addresses the problem of extending a segmentation model to new categories without forgetting previously learned ones. Existing CISS methods are replay-based, expansion-based, and distillation-based. Replay-based methods store past examples (Cha et al., 2021), or encode them using prototypes (Chen et al., 2023; Zhu et al., 2025) and generative models (Maracani et al., 2021). Expansion-based approaches expand the model architecture to allocate components for new classes, e.g. via parallel branches or class-specific modules (Zhang et al., 2022; Xiao et al., 2023), or by preserving prior knowledge by freezing parameters associated with previously learned classes (Srivastava et al., 2019; Cha et al., 2021). Distillation-based methods constrain the updated model to preserve the previous responses on old classes, e.g. by distilling logits (Cermelli et al., 2020; Michieli & Zanuttigh, 2019; Baek et al., 2022), intermediate features (Douillard et al., 2021; Michieli & Zanuttigh, 2019) or attention relations (Qiu et al., 2023). Moreover, recent work showed that explicitly modeling semantic hierarchies, particularly using hyperbolic embedding spaces, significantly improves class-incremental segmentation via structured refinement (Hindel et al., 2024; Lin et al., 2022). In this work, we build upon these advances in CISS to address the specific problem of CI-SCD, where the taxonomy of the semantic classes is progressively refined on a corpus of images that remains fixed across incremental steps.

2.3 Hyperbolic Spaces

Hyperbolic geometry is particularly adapted to model hierarchical labels (Nickel & Kiela, 2017; Ganea et al., 2018a;b), with applications in classification (Khrulkov et al., 2020), object detection (Liu et al., 2020; Lang et al., 2022), and semantic segmentation (Weber et al., 2024). Ghadimi Atigh et al. (2022) showed that hyperbolic multinomial logistic regression improves performance and hierarchical consistency, particularly in low dimension, while Hindel et al. (2024) showed the importance of leveraging taxonomies. In this context, Busemann functions have shown promise (Chami et al., 2021; Fan et al., 2023; Ghadimi Atigh et al., 2021; Berg et al., 2024; 2025), extracting geometrically meaningful features that quantify dissimilarity to ideal prototypes located at infinity. More recently, hyperbolic spaces have been explored for class-incremental settings, both in image classification (Cui et al., 2023) and taxonomy-aware segmentation (Hindel et al., 2024), showing that hyperbolic embeddings can mitigate catastrophic forgetting by preserving hierarchical relations between classes. In our setting, where incremental refinement corresponds to splitting coarse land-cover categories into finer-grained ones, such hierarchical modeling is particularly well suited to maintain semantic consistency. We therefore adopt Busemann prototypes (Berg et al., 2024; 2025)

not only to structure hierarchical segmentation, but also to define the geometric basis of our semantic flow formulation for CI-SCD.

2.4 Riemannian Vector Fields

A natural way to model displacements in hyperbolic space is through vector fields and their flow. Parameterizable vector fields on manifolds for deep learning tasks were first considered for extending neural Ordinary Differential Equations to the Riemannian setting (Rezende et al., 2020; Falorsi & Forré, 2020; Mathieu & Nickel, 2020; Lou et al., 2020; Katsman et al., 2021). While most approaches rely on extrinsic, Euclidean, formulations, ignoring the geometry of the manifold, Katsman et al. (2023) designed intrinsic vector fields in hyperbolic space for Riemannian ResNets, relying on the Busemann functions. Other designs were specifically investigated for images (van Spengler et al., 2023). In this work, we adopt the approach of Katsman et al. (2023): we construct a vector field induced by the Busemann functions, in order to model natural shifts of the hyperbolic embeddings between the temporal images.

3 Methodology

3.1 Class-Incremental Semantic Change Detection

Let $I_0, I_1 \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ denote a co-registered pair of remote sensing (RS) images acquired at two times t_0 and t_1. SCD aims at characterizing, for each changed pixel, the type of change. This is achieved by predicting a binary change map $C \in \{0,1\}^{H\times W}$, indicating which pixels changed, together with semantic labels $Y_0, Y_1 \in \mathcal{Y}^{H\times W}$ detailing the change type (Daudt et al., 2019).

Traditional methods assume that the semantic label space $\mathcal{Y}$ is fixed and fully specified during training. Instead, we consider a class-incremental scenario where categories are progressively refined over a sequence of incremental steps $s \in \{0, ..., n\}$. Let $T^s_{K_s}$ be the task linked to the s-th incremental step associated with a task-specific label set $\mathcal{Y}_s$, of size $K_s = |\mathcal{Y}_s|$. This label space forms a taxonomic tree $\mathcal{H}$ whose internal nodes represent coarse concepts (*e.g.*, vegetation, built-up), and leaves represent fine-grained ones (*e.g.*, coniferous forest, lake), see fig. 1, panel (1). The same input images are considered for all incremental steps, but for each task the labels are drawn from the task-specific set $\mathcal{Y}_s \subseteq \mathcal{H}$. Note that new labels in $\mathcal{Y}_s$ arise from refinements of existing categories and are appended to the existing ones. For each task $T^{s-1}_{K_{s-1}}$, $s > 0$, if a coarse class $k \in \mathcal{Y}_{s-1}$ is refined into children $\text{child}(k) = \{k_1, \ldots, k_m\}$, then for the task $T^s_{K_s}$ the label set is

$$\mathcal{Y}_s = \mathcal{Y}_{s-1} \setminus k \cup \text{child}(k). \tag{1}$$

Incremental steps affect both the semantic segmentation and the CD: as the semantic label space is refined, the change maps are correspondingly updated to reflect transitions between categories in $\mathcal{Y}_s$ only, as depicted in fig. 1, panel (2).

3.2 Hyperbolic Semantic Segmentation

To tackle semantic segmentation in hyperbolic space, we adopt the horospherical classifier from Berg et al. (2024). It is based on Busemann prototypes, that play a central role in the change detection mechanism introduced later in section 3.5. We use a segmentation backbone where only the last network layer is mapped to hyperbolic space (Hindel et al., 2024; Ghadimi Atigh et al., 2022; Khrulkov et al., 2020). Following Hindel et al. (2024), we use the DeepLabV3 model with a ResNet-101 backbone pre-trained on ImageNet, and consider the d-dimensional Poincaré ball model of hyperbolic space defined as $\mathbb{B}_d = \{z \in \mathbb{R}^d \,|\, \|z\|^2 < 1\}$. Unless stated otherwise, $d = 2$, which already provides competitive performance compared to baselines. Experiments in higher dimensions and with other backbones are in the Supplementary Materials F.2 and F.5.

Let e denote the Euclidean embedding vector produced by the segmentation backbone for pixel x. These embeddings are mapped onto the Poincaré ball via the exponential map at the origin:

$$z = \exp_0(e) = \tanh\big(\|e\|_2\big) \cdot e/\|e\|_2 \in \mathbb{B}_d. \tag{2}$$

Each semantic class $k \in \mathcal{Y}_s$ is parameterized by a Busemann prototype $p_k \in \mathbb{S}^{d-1}$ located on the boundary of the Poincaré ball together with a scalar bias $a_k \in \mathbb{R}$, both learned during training. They

define a horosphere, which can be interpreted as the hyperbolic analog of a Euclidean hyperplane:

$$H_k = \{z \in \mathbb{B}_d \mid -B_{p_k}(z) + a_k = 0\}, \tag{3}$$

where B_{p_k} is the Busemann function (Chami et al., 2021; Bridson & Haefliger, 1999):

$$B_{p_k}(z) = \log\left(\frac{\|p_k - z\|_2^2}{1 - \|z\|_2^2}\right). \tag{4}$$

The logit associated with embedding z and class k is then defined as $\xi_k(z) = -B_{p_k}(z) + a_k$, and class probabilities are obtained via softmax. See Supplementary Materials C for more details on hyperbolic spaces and Busemann functions.

3.3 Class-Incremental Semantic Segmentation

When the label set evolves from $\mathcal{Y}_{s-1}$ to $\mathcal{Y}_s$, prototypes associated with newly introduced fine-grained categories are initialized from their parent prototypes. For a coarse class k being refined into subclasses $\{k_1, ..., k_m\}$, each subclass prototype p_{k_j} and bias a_{k_j} are initialized as

$$p_{k_j} = \frac{p_k + \epsilon_j}{\|p_k + \epsilon_j\|_2} \in \mathbb{S}^{d-1}, \qquad a_{k_j} = a_k \in \mathbb{R}, \tag{5}$$

where $p_k \in \mathbb{S}^{d-1}$ and $a_k \in \mathbb{R}$ are the parent prototype and bias. $\epsilon_j \in \mathbb{R}^d$ is a small random perturbation drawn from a d-dimensional Gaussian distribution. This preserves semantic continuity: new subclasses are positioned close to their parent yet far enough to support discrimination through learning. Crucially, expansion occurs only in the prototype space, avoiding modifications of the segmentation backbone. During incremental steps, parent prototypes are frozen, and only the child prototypes and biases, as well as the backbone are learned.

3.4 Segmentation Loss

To train the segmentation module, we follow previous work on hierarchical semantic segmentation (Li et al., 2023; Hindel et al., 2024) and leverage the class taxonomy. We frame the problem as a pixel-wise multi-label classification problem: for each pixel, the true leaf class and all its ancestors are labeled positive, while all other categories are labeled negative. We adopt a tree-min loss (Li et al., 2023) that penalizes hierarchical inconsistencies. For each class $k \in \mathcal{Y}_s$, let A_k and D_k denote its ancestors and descendants in the label hierarchy (including k), and let $l_k \in \{0, 1\}$ indicate whether k is labeled as positive. The tree-min loss is defined as:

$$\mathcal{L}_{\text{hier}} = \sum_{k \in \mathcal{Y}_s} -l_k \log\left(\min_{u \in A_k} s_u\right) - (1 - l_k) \log\left(1 - \max_{u \in D_k} s_u\right), \tag{6}$$

where $s_k = \text{sigmoid}(\xi_k)$. In addition to this hierarchical constraint, previous approaches (Hindel et al., 2024) use a categorical cross-entropy loss at each hierarchical level to enforce competition between sibling classes. However, since our logits are derived from Busemann distances to hyperbolic prototypes, this approach was found to lead to bias toward parent prototypes and hinder fine-grained discrimination. We instead apply cross-entropy $\mathcal{L}_{\text{CE}}$ on active leaf classes only. The same applies during inference, where only leaf logits are considered. The final segmentation loss is a weighted combination of the tree-min and cross-entropy terms:

$$\mathcal{L}_{\text{seg}} = \beta \cdot \mathcal{L}_{\text{hier}} + \mathcal{L}_{\text{CE}}. \tag{7}$$

3.5 Unsupervised Change Detection via Busemann Vector Fields

To detect changes, we freeze the trained segmentation module and monitor differences between the hyperbolic embeddings obtained from the segmentation. Due to natural perturbations such as seasonal or lighting differences, we cannot identify changes by relying on the embeddings distance directly. Instead, we learn the natural shift within semantic classes as the hyperbolic semantic flow of a learned vector field V, and reconstruct the expected second embeddings (z_2) from the first ones (z_1). Remaining differences between the predicted and true embeddings will indicate change, as depicted in fig. 1, panel (3), framing CD as an anomaly detection problem. This approach is based

on the assumption that structural changes constitute a minority that the model does not learn and fails to reconstruct. More precisely, the predicted position of the second embeddings is obtained as $\tilde{z}_2 = \exp_{z_1}\big(V(z_1)\big)$, which corresponds to following the vector field and is the hyperbolic analog of the Euclidean update $\tilde{z}_2 = z_1 + V(z_1)$. To define and parameterize such a vector field V in hyperbolic space, we follow Katsman et al. (2023) and construct it based on Busemann functions and their gradients. Given any neural network $n_\omega : \mathbb{R}^K \to \mathbb{R}^K$, the vector field is defined as:

$$V_{P,\omega}(z) = \sum_{k \in \mathcal{Y}_s} \big[n_\omega(f_P(z))\big]_k \nabla B_{p_k}(z), \tag{8}$$

where $f_P(z) = \big(B_{p_1}(z), \ldots, B_{p_{K_t}}(z)\big)$ and ∇ is the Riemannian gradient. More details are in the Supplementary Materials C.3. While $\{p_1, \ldots, p_{K_s}\}$ could in principle be any set of (learned or fixed) prototypes (See Appendix F.3), we reuse the segmentation prototypes and keep them fixed. This vector field is trained separately from the segmentation task, in an unsupervised way, with a loss corresponding to the mean reconstruction error:

$$\mathcal{L}_{CD} = d\big(z_2,\ \exp_{z_1}(V_{P,\omega}(z_1))\big), \tag{9}$$

where d is the hyperbolic distance, see Supplementary Materials C.1. We predict change when

$$d\big(z_2,\ \exp_{z_1}(V_{P,\omega}(z_1))\big) > d_{\text{thresh}}, \tag{10}$$

where d_{thresh} is computed using the triangle method (Zack et al., 1977), see Supplementary Materials F.1 for details and values. During incremental steps, when new prototypes are introduced, the vector field is relearned from scratch to capture transitions involving newly introduced classes.

4 Datasets

4.1 HiChange

The analysis of hierarchical and incremental change detection requires the creation of a specialized dataset that accurately represents hierarchical relations within remote sensing images, as existing SCD datasets are limited to a small number of generic and static classes. To address this gap, we introduce HiChange, a novel dataset specifically designed for hierarchical and incremental semantic change detection in aerial imagery. HiChange is constructed from BDOrtho aerial imagery. The dataset comprises 1000 RGB-NIR tiles, each of size 5000×5000 pixels at 0.2m spatial resolution, sampled over France, and partitioned geographically into training, validation, and test sets ($70\% - 10\% - 20\%$), ensuring no spatial overlap across splits. HiChange is fully annotated following a hierarchical taxonomy. Coarse semantic information derived from OCSGE is combined with detailed labels from FLAIR (Garioud et al., 2025) and BDTopo to label each image at different semantic levels, for which the user can specify between 4 and 32 classes of interest. Additional information on the dataset, including the complete hierarchical taxonomy, details on the label generation pipeline and the sampling strategy as well as statistics and comparison with existing benchmarks, is provided in the Supplementary Material A.

4.2 Additional Datasets

While HiChange is the only available dataset with sufficient semantic richness to support hierarchical CI-SCD, we complement our evaluation with experiments on two widely used SCD benchmarks: SECOND (Yang et al., 2020) and Hi-UCD-mini (Tian et al., 2020). Although these datasets do not support incremental evaluation, they allow us to validate the segmentation backbone and Busemann vector field–based change detection mechanism, demonstrating that our approach generalizes beyond HiChange. More details on these datasets can be found in Supplementary Materials B.

5 Experiments

5.1 Does Hyperbolic Semantic Flow Enable Effective CD?

We first compare HyperFlow against unsupervised CD baselines across the three datasets. We also report the performance of the fully supervised SCD method HRSCD4 (Daudt et al., 2019) as an

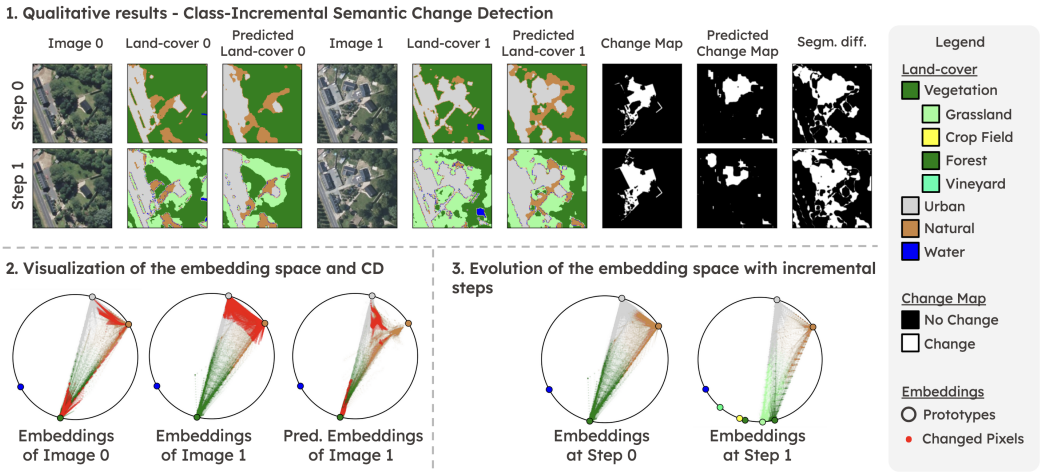


Figure 2: Qualitative illustration of HyperFlow. (1) SCD at two incremental steps. For each step, we show segmentation results at both timestamps, predicted change map, and a naive baseline obtained by differencing the two segmentation maps (Segm. diff.). (2) Hyperbolic 2-d embeddings of the first and second image, and predicted second embeddings obtained via the learned vector field. Pixels corresponding to true changes (in red) exhibit significant displacement between the timestamps, while unchanged regions show minor displacements. (3) Evolution of the embedding space across incremental steps. We visualize embeddings for a single image before and after the introduction of new classes. More qualitative examples are provided in Supplementary Materials *S5*.

upper-bound reference. Implementation details are given in the Supplementary Materials D, and qualitative examples are provided in fig. 2, panel (1). Numerically, table 1 shows that HyperFlow consistently outperforms competing approaches across all datasets. To isolate the contribution of the learned vector field, we additionally compare against a naive baseline that defines change as the difference between the segmentation of the two input images (Segm. diff.). Although this baseline already surpasses several unsupervised methods, and thus suggests that strong semantic segmentation alone can outperform many unsupervised CD approaches, an important performance gap remains with HyperFlow. This difference highlights that in the naive baseline small variations in appearance between the images can slightly modify the semantic embeddings, potentially leading to label changes in the output even when no real semantic change occurred. In contrast, HyperFlow operates directly in the embedding space and measures deviations of a semantic flow accounting for these variations. These irrelevant changes are therefore not accounted for, and only deviations that cannot be explained by the learned semantic flow, *i.e.*, semantically significant movement in the embedding space, are identified as changes. This mechanism is illustrated in fig. 2, panel (2): only pixels corresponding to true changes (shown in red) undergo significant displacement between timestamps. The hyperbolic flow therefore controls segmentation-induced errors, explaining the improvement over simple segmentation differencing, also visible in fig. 2, panel (1).

When compared with existing unsupervised methods, the strong CD results in table 1 may raise concerns about the unsupervised nature of HyperFlow, given that not only the prototypes but also the embeddings are obtained from supervised segmentation, therefore influencing the performance of Hyperflow. We emphasize that our unsupervised claim concerns the CD stage only. For CD, the supervised segmentation backbone acts purely as a feature extractor, and in this sense, Hyperflow follows common practices in unsupervised CD of using supervised pretrained feature extractors (Saha et al., 2019; Lei et al., 2026; Bergamasco et al., 2022; Du et al., 2018; Zheng et al., 2024). HyperFlow differs in that the features are specifically adapted to the target domain and semantic task, thus yielding stronger features for CD, but HyperFlow could also operate on embeddings obtained through either supervised or unsupervised pretraining, with their quality and task/domain specificity naturally affecting downstream CD performance. To support this statement, table 1 also reports results using embeddings from two foundation models: SAM3 (Carion et al., 2026) and AnySat (Astruc et al., 2025), the latter being RS-specific. Even with these fully unsupervised embeddings, HyperFlow achieves performance comparable to SOTA unsupervised CD methods. Moreover, for

Table 1: Comparison of HyperFlow with state-of-the-art unsupervised CD methods across the three datasets, for the task $T^0_{K_0}$, *i.e.* before any incremental step is performed. For HyperFlow, we additionally report CD performance using different embedding sources: HyperFlow uses the in-domain embeddings produced by the segmentation pipeline, while HyperFlow (SAM3, AnySat) use embeddings from the corresponding foundation models. Bold and underline indicate the first and second-best unsupervised scores, respectively. † The HRSCD4 model is supervised.

	HiChange				**SECOND**				**Hi-UCD-mini**			
	Acc.	**F1**	**Pre.**	**Rec.**	**Acc.**	**F1**	**Pre.**	**Rec.**	**Acc.**	**F1**	**Pre.**	**Rec.**
CVA (Malila, 1980)	74.2	8.0	4.5	39.3	67.6	30.5	25.9	37.1	75.5	14.4	9.3	32.3
SiROC (Kondmann et al., 2021)	78.0	9.3	5.3	39.1	69.4	29.6	26.5	33.4	74.4	15.6	9.9	36.9
CDRL (Noh et al., 2022)	73.0	11.6	6.4	61.9	62.3	33.2	25.2	48.8	81.6	17.9	12.5	31.3
X-Net (Luppino et al., 2021)	69.0	11.5	6.3	<u>70.1</u>	60.8	34.7	25.5	54.2	84.9	20.4	15.4	30.1
ACE-Net (Luppino et al., 2021)	69.9	11.0	6.0	65.0	65.7	33.2	26.5	44.4	78.1	20.0	13.1	42.6
KPCAMNet (Wu et al., 2021)	92.5	20.2	14.6	33.2	<u>77.9</u>	24.4	35.6	18.6	<u>89.2</u>	16.8	16.7	17.0
AnyChange (Zheng et al., 2024)	**94.0**	19.4	<u>15.8</u>	25.1	**81.2**	33.0	**52.3**	24.1	88.2	26.5	<u>22.1</u>	33.1
2Player (Béchaz et al., 2026)	92.0	17.1	12.2	28.7	69.0	34.1	28.6	42.1	76.7	21.4	13.7	48.6
Segm. diff.	86.7	<u>25.1</u>	14.9	**78.8**	60.9	<u>41.9</u>	29.3	73.3	73.3	<u>28.1</u>	17.0	<u>80.8</u>
HyperFlow	92.9	**29.6**	**20.5**	52.8	73.7	**43.0**	<u>36.8</u>	51.8	**91.7**	**44.3**	**39.0**	51.4
HyperFlow (SAM3)	<u>93.9</u>	15.1	12.3	19.6	48.5	40.3	25.9	**90.8**	49.7	18.3	10.2	**88.0**
HyperFlow (AnySat)	93.5	20.1	14.2	34.5	45.0	36.5	23.4	<u>82.2</u>	84.3	23.7	17.4	37.1
HRSCD4† (Daudt et al., 2019)	94.2	42.9	29.8	76.4	66.5	52.0	35.9	94.4	94.0	60.9	52.3	72.9

HiChange and Hi-UCD-mini, performance improves from SAM3 to AnySat, consistent with increasing specialization to the RS domain. Thus, while supervised segmentation embeddings contribute to the strong results observed with HyperFlow, they are not required for competitive unsupervised CD performance. More generally, HyperFlow benefits from strong representations, regardless of how they are obtained.

5.2 Class-incremental settings

When considering the incremental setting (table 2), HyperFlow maintains strong SCD performance even as the number of semantic classes grows over incremental steps. While these scenarios are inherently more challenging, not only does HyperFlow still outperform the strongest unsupervised CD baselines presented in table 1, but it is also capable of characterizing the nature of a change at a much finer lever of the semantic taxonomy.

Table 2: SCD performance of HyperFlow across incremental steps for the HiChange dataset. We report segmentation performance with mIoU, CD with F1, and combined CD and segmentation using the Semantic Change Segmentation (SCS) score from Toker et al. (2022).

	T^0_4	T^1_7	T^2_{13}	T^3_{16}	T^4_{20}	T^5_{22}	T^6_{32}
Segm. [mIoU]	62.0	56.2	33.7	28.4	23.5	20.7	16.1
CD [F1]	29.6	29.6	29.8	29.2	27.7	31.6	26.0
SCD [SCS]	39.0	34.4	22.4	19.9	17.3	17.6	13.8

Table 3 compares HyperFlow segmentation results against state-of-the-art CISS methods across incremental steps for the three datasets considered. HyperFlow achieves competitive performance and, while improvements over prior approaches remain moderate, the results demonstrate that considering Busemann prototypes and incorporating CD do not degrade CISS performance.

We also investigate whether progressively refining the taxonomy provides advantages over directly training on the full set of fine-grained categories. Results in table 4 show that structuring semantic representations progressively, rather than directly considering fine-grained distinctions, leads to improved SCD. Figure 2, panel (3) visualizes this effect, showing how incremental refinement preserves global organization while progressively increasing semantic resolution. More embeddings

Table 3: Comparison of Hyperflow with state-of-the-art CISS methods for the three datasets. The mIoU is reported for each method and all incremental steps. Note that since SECOND and Hi-UCD do not enable incremental scenarios, only the base task is considered (Hyp: hyperbolic).

Method	Hyp.	HiChange T_4^0	T_7^1	T_{13}^2	T_{16}^3	T_{20}^4	T_{22}^5	T_{32}^6	SECOND T_6^0	Hi-UCD T_9^0
PLOP (Douillard et al., 2021)	✗	66.3	11.4	2.0	0.4	0.9	0.8	0.6	**53.7**	41.8
MiB (Cermelli et al., 2020)	✗	66.3	52.9	36.0	27.8	22.2	20.7	16.1	**53.7**	41.8
MiB+AWT (Goswami et al., 2023)	✗	66.3	54.7	**37.6**	28.2	22.9	20.2	15.9	**53.7**	41.8
DKD (Baek et al., 2022)	✗	64.7	**57.4**	35.0	27.8	22.1	**21.5**	**17.2**	38.8	**45.8**
TOPICS (Hindel et al., 2024)	✓	**67.6**	54.5	24.6	15.9	8.5	6.2	3.8	46.8	37.5
SAM3 (Carion et al., 2026)	✗	29.8	24.3	20.5	13.8	11.2	9.3	6.4	28.0	20.9
HyperFlow	✓	62.0	56.2	33.7	**28.4**	**23.5**	20.7	16.1	47.3	42.4

comparisons across incremental and non-incremental scenarios are presented in the Supplementary Materials E.

To isolate the effect of hierarchical initialization, we also investigate in table 4 a non-incremental scenario where prototypes are initialized according to the class hierarchy. Results improve compared to the non-incremental baseline, indicating that respecting class hierarchies at initialization helps segmentation. Performance remains however lower than in the full incremental setting with progressive taxonomy refinement, showing that benefits of incremental learning go beyond hierarchical initialization alone.

Table 4: Comparison between (i) class-incremental training with progressive taxonomy refinement, (ii) non-incremental training on the full fine-grained taxonomy, and (iii) non-incremental training with hierarchical prototype initialization for the HiChange dataset. The SCS score is reported for all incremental steps. At each step, the incremental and non incremental models are trained for the same total number of epochs.

Incr.	Hier.	T_4^0	T_7^1	T_{13}^2	T_{16}^3	T_{20}^4	T_{22}^5	T_{32}^6
✓	✓	**39.0**	**34.4**	**22.4**	**19.9**	**17.3**	**17.6**	**13.8**
✗	✗	**39.0**	30.9	16.6	13.7	11.7	11.7	9.1
✗	✓	**39.0**	31.7	17.8	16.2	14.4	14.1	10.0

6 Conclusion

In this work, we introduce HyperFlow, a method that bridges CISS and SCD via hyperbolic semantic flow. By representing semantic categories with Busemann prototypes and modeling temporal evolution of semantic classes as a learned vector field in hyperbolic space, we formulate CD as the identification of deviations from expected temporal dynamics. We train and evaluate HyperFlow on HiChange, a new large-scale, very high resolution and hierarchical SCD dataset that we propose. Our results demonstrate that formulating semantic change as a hyperbolic flow enables effective unsupervised CD while preserving competitive segmentation performance, both in standard and in incremental settings. Furthermore, we show that progressively refining the taxonomy, *i.e*. the semantic changes considered, rather than considering all changes from the beggining, helps considerably the task of fine-grained semantic change detection. Although the CD mechanism does not require supervision, HyperFlow still depends on the quality of the semantic segmentation model. Future works will focus on learning better representations for the semantic segmentation backbone, for instance leveraging foundation models based on large scale self-supervised pretraining (Sumbul et al., 2025; Cong et al., 2022). Moreover, we will explore leveraging longer time series, which could enable the modeling of more complex or gradual temporal transitions, such as seasonal vegetation cycles or multi-stage urban development, thereby providing a richer representation of temporal dynamics.

AI USE STATEMENT

In this work, we used generative AI tools to assist with debugging and improving existing code, and to reformulate some paragraphs of the manuscript to improve readability. The original code and scientific text were however written by the authors and were not generated by generative AI. Generative AI tools were not used to generate research ideas or design the methodology and experiments, nor to interpret the results. All AI-generated modifications were carefully reviewed by the authors. Code modifications were verified for correctness, and changes to the manuscript carefully reviewed by multiple authors to ensure that they accurately reflected the intended meaning. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, all code used in our experiments will be made publicly available upon acceptance of the paper. Detailed information on the generation of the HiChange dataset is provided in the supplementary material A. The images and label sources used to construct the HiChange dataset are publicly available, with their respective sources and links provided in Appendix A. The code used to fuse the different label sources will also be published. The resulting HiChange dataset will be publicly released upon acceptance of the paper. All other datasets used in our experiments are publicly available, and their sources are provided in Appendix B. Implementation details are provided in Appendix D. Pretrained model checkpoints used to obtain the reported results will also be released alongside the source code.

REFERENCES

Kyeongjin Ahn, Sungwon Han, Sungwon Park, Jihee Kim, Sangyoon Park, and Meeyoung Cha. Generalizable disaster damage assessment via change detection with vision foundation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27784–27792, 2025.

Giuseppina Andresini, Annalisa Appice, Daniele Iaia, Donato Malerba, Nicolo Taggio, and Antonello Aiello. Leveraging autoencoders in change vector analysis of optical satellite images. *Journal of Intelligent Information Systems*, 58(3):433–452, 2022.

Guillaume Astruc et al. Anysat: One earth observation model for many resolutions, scales, and modalities. In *CVPR*, 2025.

Donghyeon Baek, Youngmin Oh, Sanghoon Lee, Junghyup Lee, and Bumsub Ham. Decomposed knowledge distillation for class-incremental semantic segmentation. *Advances in neural information processing systems*, 35:10380–10392, 2022.

Manon Béchaz, Emanuele Dalsasso, Ciprian Tomoiagă, Marcin Detyniecki, and Devis Tuia. 2player: A general framework for self-supervised change detection via cooperative learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 232:34–47, 2026.

Paul Berg, Bjoern Michele, Minh-Tan Pham, Laetitia Chapel, and Nicolas Courty. Horospherical learning with smart prototypes. In *British Machine Vision Conference (BMVC)*, 2024.

Paul Berg, Léo Buecher, Björn Michele, Minh-Tan Pham, Laetitia Chapel, and Nicolas Courty. Multi-prototype hyperbolic learning guided by class hierarchy: P. berg et al. *International Journal of Computer Vision*, 133(11):7969–7984, 2025.

Luca Bergamasco, Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone. Unsupervised change detection using convolutional-autoencoder multiresolution features. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022.

Silvère Bonnabel. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, September 2013. ISSN 1558-2523. doi: 10.1109/TAC.2013.2254619.

Martin R. Bridson and André Haefliger. *Metric Spaces of Non-Positive Curvature*, volume 319 of *Grundlehren Der Mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 1999. ISBN 978-3-642-08399-0 978-3-662-12494-9.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2026. URL https://arxiv.org/abs/2511.16719.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9233–9242, 2020.

Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in neural information processing systems*, 34:10919–10930, 2021.

Ines Chami, Albert Gu, Dat P. Nguyen, and Christopher Re. HoroPCA: Hyperbolic Dimensionality Reduction via Horospherical Projections. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 1419–1429. PMLR, July 2021.

Jinpeng Chen, Runmin Cong, Yuxuan Luo, Horace Ip, and Sam Kwong. Saving 100x storage: Prototype replay for reconstructing training sample distribution in class-incremental semantic segmentation. *Advances in Neural Information Processing Systems*, 36:35988–35999, 2023.

Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-Margin Classification in Hyperbolic Space. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 1832–1840. PMLR, April 2019.

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.

Yawen Cui, Zitong Yu, Wei Peng, Qi Tian, and Li Liu. Rethinking few-shot class-incremental learning with open-set hypothesis in hyperbolic geometry. *IEEE Transactions on Multimedia*, 26: 5897–5910, 2023.

Rodrigo Caye Daudt, Bertr Le Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. In *2018 25th IEEE international conference on image processing (ICIP)*, pp. 4063–4067. IEEE, 2018.

Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau. Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding*, 187: 102783, 2019.

Lei Ding, Jing Zhang, Haitao Guo, Kai Zhang, Bing Liu, and Lorenzo Bruzzone. Joint spatio-temporal modeling for semantic change detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024.

Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4040–4050, 2021.

Bo Du, Lixiang Ru, Chen Wu, and Liangpei Zhang. Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images. *arXiv preprint arXiv:1812.00645*, 2018.

Luca Falorsi and Patrick Forré. Neural ordinary differential equations on manifolds. *arXiv preprint arXiv:2006.06663*, 2020.

Xiran Fan, Chun-Hao Yang, and Baba Vemuri. Horospherical Decision Boundaries for Large Margin Classification in Hyperbolic Space. *Advances in Neural Information Processing Systems*, 36: 11194–11204, December 2023.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1646–1655. PMLR, July 2018a.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b.

Anatol Garioud, Sébastien Giordano, Nicolas David, and Nicolas Gonthier. Flair-hub: Large-scale multimodal dataset for land cover and crop mapping. *arXiv preprint arXiv:2506.07080*, 2025.

Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *Advances in neural information processing systems*, 34:103–115, 2021.

Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne Van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4453–4462, 2022.

Dipam Goswami, René Schuster, Joost Van de Weijer, and Didier Stricker. Attribution-aware weight transfer: A warm-start initialization for class-incremental semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3195–3204, 2023.

Julia Hindel, Daniele Cattaneo, and Abhinav Valada. Taxonomy-aware continual semantic segmentation in hyperbolic spaces for open-world perception. *IEEE Robotics and Automation Letters*, 10(2):1904–1911, 2024.

Jinrong Hu and Yuanzhi Zhang. Seasonal change of land-use/land-cover (lulc) detection using modis data in rapid urbanization regions: A case study of the pearl river delta region (china). *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(4):1913–1920, 2013.

Zhi Huang, Xiuping Jia, and Linlin Ge. Sampling approaches for one-pass land-use/land-cover change mapping. *International Journal of Remote Sensing*, 31(6):1543–1554, 2010.

Deyi Ji, Siqi Gao, Mingyuan Tao, Hongtao Lu, and Feng Zhao. Changenet: Multi-temporal asymmetric change detection dataset. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2725–2729. IEEE, 2024.

Huiwei Jiang, Min Peng, Yuanjun Zhong, Haofeng Xie, Zemin Hao, Jingming Lin, Xiaoli Ma, and Xiangyun Hu. A survey on deep learning-based change detection from high-resolution remote sensing images. *Remote Sensing*, 14(7):1552, 2022.

Isay Katsman, Aaron Lou, Derek Lim, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Equivariant manifold flows. *Advances in Neural Information Processing Systems*, 34:10600–10612, 2021.

Isay Katsman, Eric Chen, Sidhanth Holalkere, Anna Asch, Aaron Lou, Ser Nam Lim, and Christopher M De Sa. Riemannian residual neural networks. *Advances in Neural Information Processing Systems*, 36:63502–63514, 2023.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6418–6428, 2020.

Lukas Kondmann, Aysim Toker, Sudipan Saha, Bernhard Schölkopf, Laura Leal-Taixé, and Xiao Xiang Zhu. Spatial context awareness for unsupervised change detection in optical satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2021.

Christopher Lang, Alexander Braun, Lars Schillingmann, and Abhinav Valada. On hyperbolic embeddings in object detection. In *DAGM German Conference on Pattern Recognition*, pp. 462–476. Springer, 2022.

Tao Lei, Shuxin Zhang, Shaoxiong Lin, Tongfei Liu, Zhiyong Lv, Tao Gao, Maoguo Gong, and Asoke K Nandi. Remote sensing image change detection using deep learning techniques: a comprehensive survey. *Artificial Intelligence Review*, 2026.

Liulei Li, Wenguan Wang, Tianfei Zhou, Ruijie Quan, and Yi Yang. Semantic hierarchy-aware segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2123–2138, 2023.

Xiaodong Li, Feng Ling, Giles M Foody, and Yun Du. A superresolution land-cover change detection method using remotely sensed images with different spatial resolutions. *IEEE Transactions on Geoscience and Remote Sensing*, 54(7):3822–3841, 2016.

Zhiqiu Lin, Deepak Pathak, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Continual learning with evolving class ontologies. *Advances in Neural Information Processing Systems*, 35:7671–7684, 2022.

Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9273–9281, 2020.

Zhun-Ga Liu, Zuo-Wei Zhang, Quan Pan, and Liang-Bo Ning. Unsupervised change detection from heterogeneous data based on image translation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.

Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Neural manifold ordinary differential equations. *Advances in Neural Information Processing Systems*, 33:17548–17558, 2020.

Luigi Tommaso Luppino, Michael Kampffmeyer, Filippo Maria Bianchi, Gabriele Moser, Sebastiano Bruno Serpico, Robert Jenssen, and Stian Normann Anfinsen. Deep image translation with an affinity-based change prior for unsupervised multimodal change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–22, 2021.

Zhiyong Lv, Ming Zhang, Weiwei Sun, Jón Atli Benediktsson, Tao Lei, and Nicola Falco. Spatial-contextual information utilization framework for land cover change detection with hyperspectral remote sensed images. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11, 2023.

William A Malila. Change vector analysis: An approach for detecting forest changes with landsat. In *LARS symposia*, pp. 385, 1980.

Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7026–7035, 2021.

Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. *Advances in neural information processing systems*, 33:2503–2515, 2020.

Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Allan Aasbjerg Nielsen and Knut Conradsen. Multivariate alteration detection (mad) in multispectral, bi-temporal image data: a new approach to change detection studies. *Dept. Mathematical Modelling, Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep*, 11:1997, 1997.

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*, pp. 1352–1361, 2022.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

Yiqiao Qiu, Yixing Shen, Zhuohao Sun, Yanchong Zheng, Xiaobin Chang, Weishi Zheng, and Ruixuan Wang. Sats: Self-attention transfer for continual semantic segmentation. *Pattern Recognition*, 138:109383, 2023.

Danilo Jimenez Rezende, George Papamakarios, Sébastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, pp. 8083–8092. PMLR, 2020.

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.

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.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic Neural Networks++. In *International Conference on Learning Representations*, October 2020.

Shivangi Srivastava, Maxim Berman, Matthew B Blaschko, and Devis Tuia. Adaptive compression-based lifelong learning. *arXiv preprint arXiv:1907.09695*, 2019.

Gencer Sumbul, Chang Xu, Emanuele Dalsasso, and Devis Tuia. Smarties: Spectrum-aware multisensor auto-encoder for remote sensing images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5569–5578, 2025.

Onur Tasar, Yuliya Tarabalka, and Pierre Alliez. Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3524–3537, 2019.

Shiqi Tian, Ailong Ma, Zhuo Zheng, and Yanfei Zhong. Hi-ucd: A large-scale dataset for urban semantic change detection in remote sensing imagery. *arXiv preprint arXiv:2011.03247*, 2020.

Aysim Toker, Lukas Kondmann, Mark Weber, Marvin Eisenberger, Andrés Camero, Jingliang Hu, Ariadna Pregel Hoderlein, Çağlar Şenaras, Timothy Davis, Daniel Cremers, et al. Dynamicearthnet: Daily multi-spectral satellite dataset for semantic change segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21158–21167, 2022.

Redha Touati, Max Mignotte, and Mohamed Dahmane. Anomaly feature learning for unsupervised change detection in heterogeneous images: A deep sparse residual model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:588–600, 2020.

Max van Spengler, Erwin Berkhout, and Pascal Mettes. Poincare resnet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5419–5428, 2023.

Michele Volpi, Devis Tuia, Gustavo Camps-Valls, and Mikhail Kanevski. Unsupervised change detection with kernels. *IEEE Geoscience and Remote Sensing Letters*, 9(6):1026–1030, 2012.

Michele Volpi, Devis Tuia, Francesca Bovolo, Mikhail Kanevski, and Lorenzo Bruzzone. Supervised change detection in vhr images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation*, 20:77–85, 2013.

Mingze Wang, Lili Su, Cilin Yan, Sheng Xu, Pengcheng Yuan, Xiaolong Jiang, and Baochang Zhang. Rsbuilding: Towards general remote sensing image building extraction and change detection with foundation model. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Simon Weber, Bar Zöngür, Nikita Araslanov, and Daniel Cremers. Flattening the parent bias: Hierarchical semantic segmentation in the poincaré ball. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28223–28232, 2024.

Lean Weng, Wenqing Yang, Boni Hu, Pengcheng Han, Shaocheng Xue, Yu Zhang, Haowei Li, Jie Jin, and Shuhui Bu. Mdinet: Multidomain incremental network for change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024.

Chen Wu, Bo Du, Xiaohui Cui, and Liangpei Zhang. A post-classification change detection method based on iterative slow feature analysis and bayesian soft fusion. *Remote Sensing of Environment*, 199:241–255, 2017.

Chen Wu, Hongruixuan Chen, Bo Du, and Liangpei Zhang. Unsupervised change detection in multitemporal vhr images based on deep kernel pca convolutional mapping network. *IEEE Transactions on Cybernetics*, 52(11):12084–12098, 2021.

Jia-Wen Xiao, Chang-Bin Zhang, Jiekang Feng, Xialei Liu, Joost van de Weijer, and Ming-Ming Cheng. Endpoints weight fusion for class incremental semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7204–7213, 2023.

Kunping Yang, Gui-Song Xia, Zicheng Liu, Bo Du, Wen Yang, Marcello Pelillo, and Liangpei Zhang. Semantic change detection with asymmetric siamese networks. *arXiv preprint arXiv:2010.05687*, 2020.

Gregory W Zack, William E Rogers, and Samuel A Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry & Cytochemistry*, 25(7):741–753, 1977.

Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7053–7064, 2022.

Hanchao Zhang, Ruiqian Zhang, Xiaogang Ning, Xiao Huang, You He, Yixin Chen, Mingzhu Li, Wei Cui, and Jiaming Wang. Lim-cd: A large-scale remote sensing change detection dataset for incremental monitoring. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10:903–910, 2023.

Haotian Zhang, Han Guo, Keyan Chen, Hao Chen, Zhengxia Zou, and Zhenwei Shi. Foba: A foreground–background co-guided method and new benchmark for remote sensing semantic change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–19, 2025.

Like Zhao, Mingwang Yang, Yiran Zhao, Yuqing Liu, Huawei Jiang, and Zhen Yang. Df2net: A differential-feature dynamic-fusion network for building change detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 64:1–15, 2025.

Zhuo Zheng, Yanfei Zhong, Junjue Wang, Ailong Ma, and Liangpei Zhang. Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters. *Remote Sensing of Environment*, 265:112636, 2021.

Zhuo Zheng, Yanfei Zhong, Liangpei Zhang, and Stefano Ermon. Segment any change. *Advances in Neural Information Processing Systems*, 37:81204–81224, 2024.

Yanpeng Zhou, Jinjie Wang, Jianli Ding, Bohua Liu, Nan Weng, and Hongzhi Xiao. Signet: A siamese graph convolutional network for multi-class urban change detection. *Remote Sensing*, 15 (9):2464, 2023.

Guilin Zhu, Dongyue Wu, Changxin Gao, Runmin Wang, Weidong Yang, and Nong Sang. Adaptive prototype replay for class incremental semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10932–10940, 2025.

Qiqi Zhu, Xi Guo, Weihuan Deng, Sunan Shi, Qingfeng Guan, Yanfei Zhong, Liangpei Zhang, and Deren Li. Land-use/land-cover change detection based on a siamese global learning framework for high spatial resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:63–78, 2022.

APPENDIX

Organization of the Appendix. These supplementary materials provide additional details and experiments complementing the main paper:

- Section A presents further information about the proposed HiChange dataset, including comparisons with existing SCD datasets, the sampling strategy, the label generation pipeline, and dataset statistics.
- Section B summarizes the characteristics of the SECOND and Hi-UCD-mini datasets used in our experiments.
- Section C reviews the hyperbolic geometry concepts underlying our method.
- Section D provides implementation details and the definition of the incremental tasks.
- Section E presents additional qualitative results.
- Section F reports complementary experiments analyzing the influence of the threshold, backbone architecture and embedding dimension.

A ADDITIONAL INFORMATION ON THE HICHANGE DATASET

A.1 COMPARISON WITH EXISTING SCD DATASETS

Table 5 compares HiChange with commonly used SCD datasets. The latter are typically limited to a small number of semantic categories. This limitation motivates the creation of HiChange, as such restricted semantic diversity makes them less suitable for studying CI-SCD scenarios. HiChange addresses these limitations by introducing a larger and hierarchically structured taxonomy designed to support progressive semantic refinement.

Table 5: Existing SCD datasets, and comparison with the proposed HiChange dataset.

Dataset	Image pairs	Image size	Res [m]	Categories
HRSCD Daudt et al. (2019)	291	10000×10000	0.5	5
Hi-UCD-mini Tian et al. (2020)	1293	1024×1024	0.1	9
SECOND Yang et al. (2020)	4662	512×512	not indicated	6
CNAM-CD Zhou et al. (2023)	2503	512×512	0.5	5
DynamicEarthNet Toker et al. (2022)	75 time series of 730 images	1024×1024	3	7
ChangeNet Ji et al. (2024)	31000	$\sim 1900 \times 1200$	0.3	6
LevirSCD Zhang et al. (2025)	3225	256×256	1-2	16
HiChange	1000	5000×5000	0.2	4 - 32, organized hierarchically

A.2 SAMPLING OF THE IMAGES

In this section, we detail the sampling procedure used to construct the HiChange dataset. The objective was to select a subset of tiles from the French national BDOrtho aerial imagery database[1] that maximizes the diversity and relevance of observed semantic changes. We fixed the total number of tiles to $N = 1000$, a choice made to balance dataset size, semantic diversity, and feasible training time for deep-learning models. Each tile has a spatial resolution of 0.2 m and a size of 5000×5000 pixels. The sampling procedure consists of three main stages described below.

Initial Filtering Based on the Number of Changes Observed. To ensure that the final dataset focuses on regions with actual changes, we perform an initial selection of tiles based on the density of changes within each tile. For this purpose, we use the OCSGE database[2], which we also employ

[1] https://geoservices.ign.fr/bdortho

[2] https://geoservices.ign.fr/ocsge

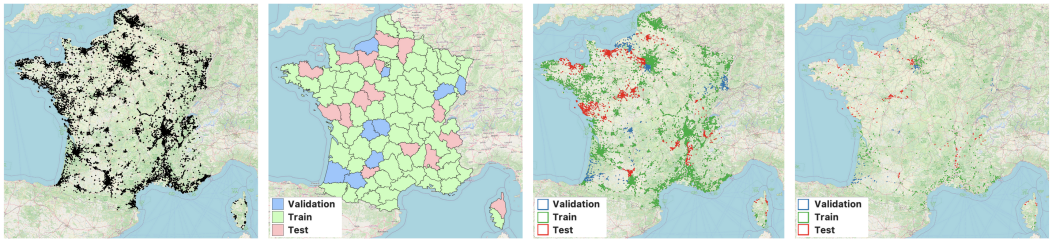


Figure 3: Sampling procedure for constructing the HiChange dataset. (a) Tiles retained after the initial filtering step based on the number of changes observed. (b) Geographical partitioning of French administrative departments used to define train, test and validation splits. (c) Assignment of tiles to the corresponding split according to their geographical location. (d) Final selection of the tiles after the greedy sampling algorithm, ensuring diverse and balanced semantic change types in all splits of the dataset.

for change label generation. As OCSGE is available only for specific years depending on the region, we select BDOrtho acquisition years to match the available OCSGE years to ensure temporal consistency. Acquisition years range from 2016 to 2023, and time between the 2 acquisitions for each pair is 2 to 3 years. OCSGE is provided as a vector dataset in which each changed object is represented as a polygon. For each tile in BDOrtho, we count the number of OCSGE polygons falling within its spatial extent. We retain only tiles containing more than 30 polygons of change. This threshold was chosen to filter out tiles without meaningful change activity. This procedure yields approximately 20000 tiles. As anticipated, the retained tiles concentrate around dynamic, predominantly urban, regions as depicted in fig. 3, panel (a).

Geographical train-validation-test Assignment. The second step consists in defining the dataset splits. We opt for a geographical partitioning based on administrative departments to prevent spatial overlap between training, validation, and testing. Each of the roughly 20000 candidate tiles is assigned to one of the three splits (train/validation/test), as depicted in fig. 3, panel (b) and (c).

Greedy Selection for Diversity and Class Balance. Even after the initial filtering, the number of candidate tiles remains far larger than the desired dataset size. The final selection aims to (i) maximize diversity in the types of observed semantic changes and (ii) balance the relative frequency of change categories. To this end, we exploit the land-cover attributes provided by OCSGE and adopt an iterative greedy selection procedure. For each change polygon, its change type is defined as the combination of the dominant semantic labels before and after change (*i.e.*, an ordered pair $(\text{category}_{\text{before}}, \text{category}_{\text{after}})$). For each train-test-validation split, we run the tile selection procedure as follows:

1. Initialize the selected tile set as empty.
2. For each unselected tile in the split:
   - Simulate adding the tile to the current selection.
   - Compute the deviation between the resulting distribution of change types and the target uniform distribution across all change types.
   - Measure the reduction in deviation that would result from adding the tile.
3. Select the tile that yields the greatest improvement.
4. Repeat until the desired number of tiles K for the split is reached.

This greedy algorithm progressively builds a tile set that is both diverse and balanced in terms of semantic change types. We sample $K = 700$ tiles for the train set, $K = 100$ for the validation set and $K = 200$ for the test set. The final selected tiles are depicted in fig. 3, panel (d).

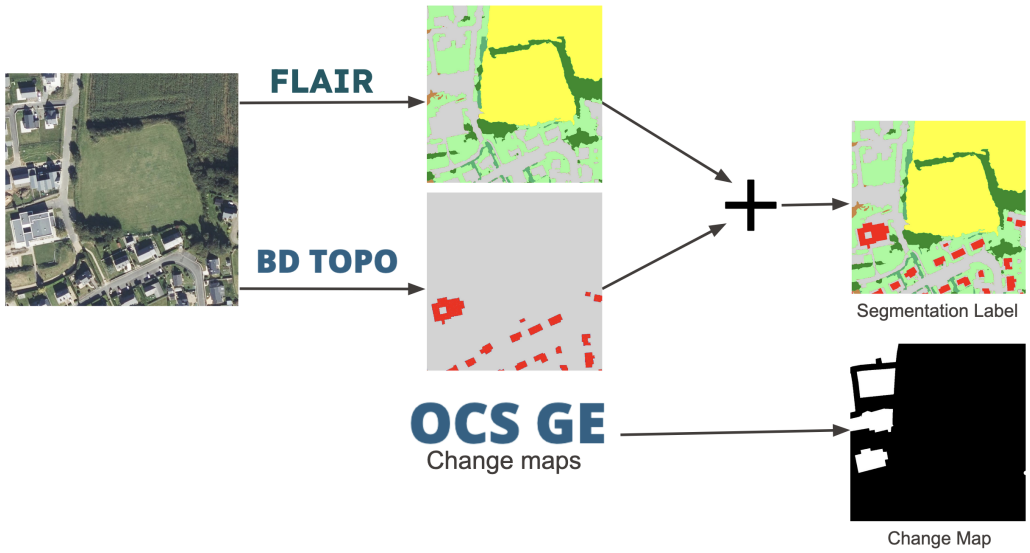


Figure 4: Pipeline for the generation of the labels.

A.3 Generation of the Labels

The pipeline for the generation of the labels is presented in Figure 4. The change detection labels are derived from the OCSGE database, which provides coarse annotations of both land-cover segmentation and change between two temporal acquisitions. We directly exploit the change information from OCSGE as a binary indication of change. This change map is later refined at usage time to align with the selected semantic granularity, as detailed later in this section. However, the semantic segmentation labels provided by OCSGE remain coarse and cover only a limited set of semantic categories[3]. This level of detail is insufficient for the objectives of HiChange, which require fine-grained semantic information. Therefore, additional sources of information are incorporated to construct richer semantic labels. In particular, we leverage the BDTopo database[4] from the French National Institute of Geographic and Forest Information (IGN) which provides high-precision vector layers describing buildings, roads, and other man-made infrastructure for multiple years. A key limitation, however, is that BDTopo is not spatially exhaustive: several regions of the territory remain unlabeled or only partially covered. One possible remedy would be to supplement BDTopo with the OCSGE semantic segmentation data, but the latter is too coarse for our target spatial resolution (0.2 m). To overcome this limitation, we employ the FLAIR land-cover model Garioud et al. (2025), which was also trained on BDOrtho imagery and has demonstrated high accuracy on segmentation tasks. We run FLAIR inference over all tiles and use its predicted land-cover labels in regions not covered by BDTopo. The resulting semantic segmentation combines BDTopo-derived vector information where available and FLAIR predictions elsewhere, thereby achieving both fine-grained semantic resolution and full spatial coverage.

Hierarchical Label Selection. The above procedure yields semantic labels at the finest level of granularity. These labels are organized into a hierarchical taxonomy structured in three levels, as shown in fig. 5. When using the dataset, the user selects the desired classes to consider. Fine-grained categories are then automatically mapped to their parent categories when coarser levels are requested.

Consistency Filtering of Change Labels. To ensure coherence between the semantic segmentation labels and the change annotations, we apply a filtering mechanism to the change map at usage time. Specifically, if the selected classes are coarse and a tile exhibits a change only between child

[3] https://geoservices.ign.fr/sites/default/files/2025-10/DC_OCSGE_NG.pdf
[4] https://geoservices.ign.fr/bdtopo

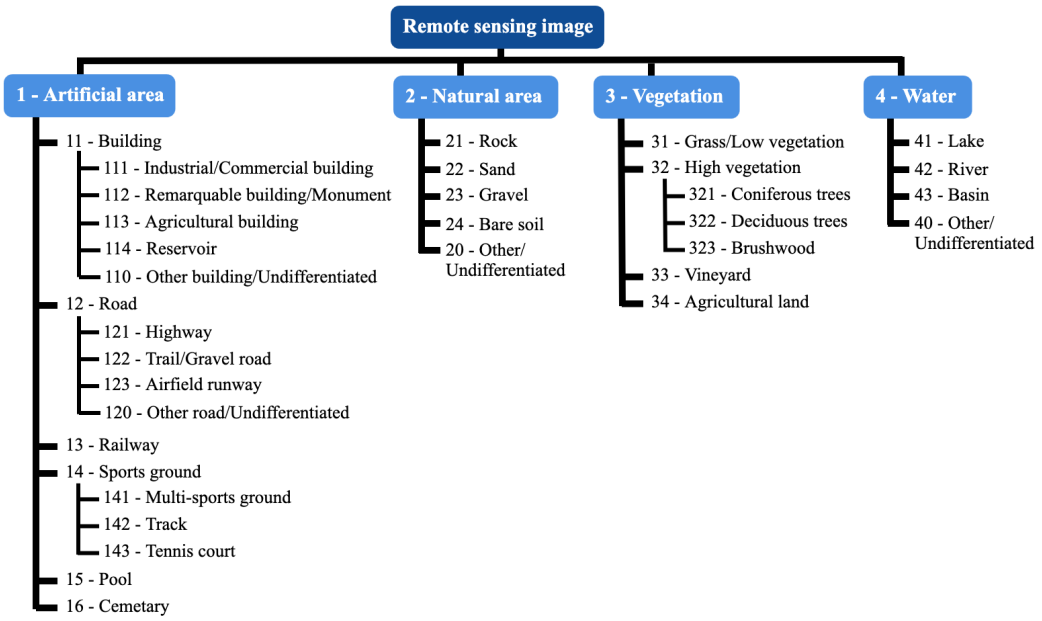


Figure 5: Taxonomy of the semantic labels of the HiChange dataset.

categories belonging to the same parent category (*e.g.*, a change from *forest* to *grass* when only *vegetation* is considered), the change should not be reported. We therefore mark as changed only pixels that are both (i) marked as changed by OCSGE and (ii) present different semantic labels between the two timestamps at the chosen semantic level. This guarantees that the binary change masks remain semantically meaningful and consistent across hierarchical configurations, and solves at the same time the problem of the coarse resolution of OCSGE.

Limitations The construction of such a large dataset involves a trade-off between annotation quality and coverage. Some regions in the dataset where BDTopo is unavailable are labeled using FLAIR. These automatically generated labels may inherit biases and errors from the model, and may therefore introduce some noise into the benchmark. In addition, the use of OCSGE may exclude changes that fall below its minimum mapping unit. However, given the scale of the dataset, manual annotation is not practically feasible. The adopted methodology provides a compromise that enables large-scale evaluation while acknowledging the limitations associated with automatically generated labels.

A.4 Examples from the dataset

Some example images from the HiChange dataset are provided in fig. 6.

A.5 Dataset Statistics

The distribution of the semantic labels at the three levels of depth of the hierarchy are presented in fig. 7. The confusion matrix for the land-cover for pixels that contain change is presented in fig. 8. Although not all types of changes are present in the dataset, especially at the finest level of the hierarchy, we observe a large variety.

B Additional Information on the SECOND and Hi-UCD-mini Datasets

SECOND The SECOND dataset consists of 4662 pairs of 512×512 aerial images collected from multiple platforms and sensors over several Chinese cities. Spatial resolution is not indicated.

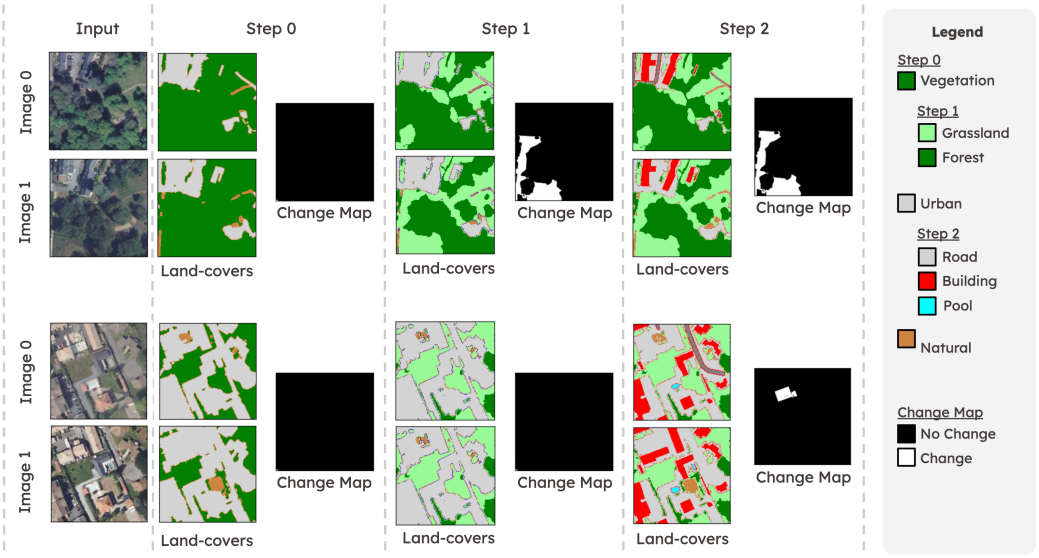


Figure 6: Examples from the HiChange dataset, illustrating the refinement of semantic labels. In the initial step, only coarse distinctions between vegetation and constructed areas are considered. At step 1, we have additional information on the type of vegetation and, at step 2, about the type of construction. Note that the change maps adapt to the set of semantic labels considered: in the first example, change happens within the vegetation, and is captured only at steps 1 and 2. Similarly, in the second example, a building is constructed and the change is only captured in the last incremental step.

Images are annotated by experts on six land-cover classes: *low vegetation, non-vegetated ground surface, trees, water, building* and *playground*. Semantic annotation is provided for changed regions only, and therefore only evaluated on those regions. The dataset is split in 2968 pairs for training and 1694 for testing.

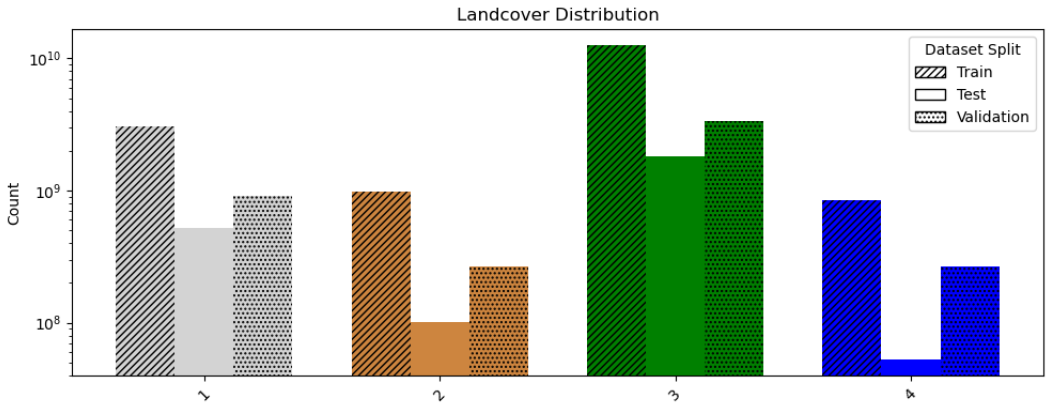
Hi-UCD-mini The Hi-UCD-mini dataset is an urban SCD dataset built from very-high-resolution aerial images (0.1 m spatial resolution). It focuses on the city of Tallinn and contains 1293 image pairs of size 1024×1024 acquired between 2017 and 2019. The images are annotated for both binary change and segmentation, covering 9 semantic categories: *water, grass, building, green house, road, bridge, bare land, woodland* and *undefined*.

C Hyperbolic Space and Vector Fields

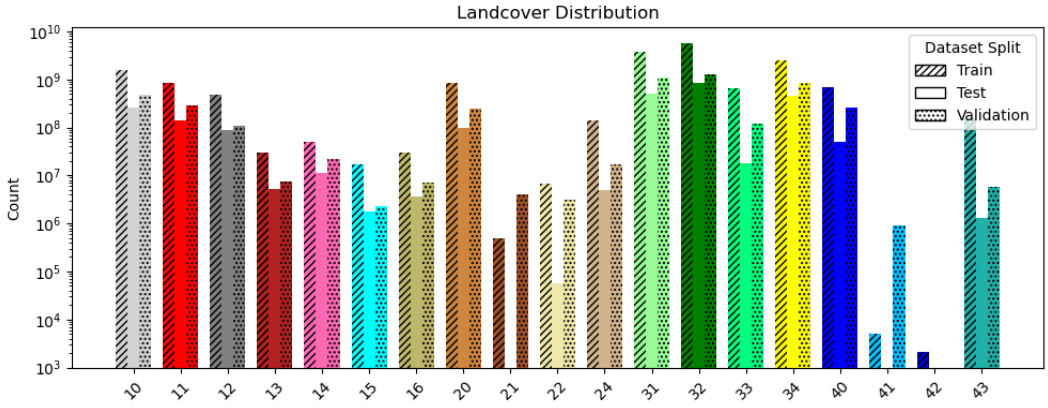
This section briefly reviews the hyperbolic geometry and vector field formulation that form the mathematical foundation of our method.

C.1 Hyperbolic Space

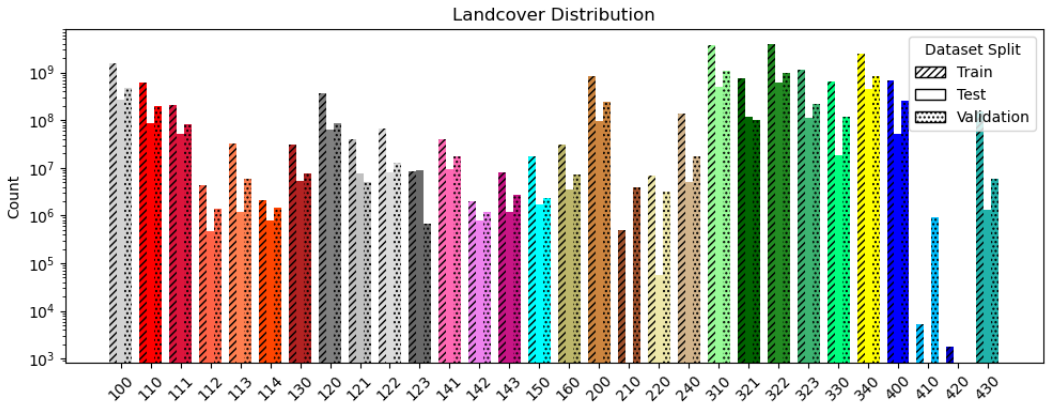
Hyperbolic spaces are smooth manifolds, meaning that they can be endowed with Cartesian coordinates locally around every point. This structure makes it possible to define tangent vectors in each point, corresponding to the derivatives of all possible smooth curves going through it. The set of tangent vectors in some point x of a manifold $\mathcal{M}$ is called the tangent space at x and is denoted $T_x\mathcal{M}$. It is a vector space that can be seen as the first order approximation of the manifold around this point. Riemannian manifolds are smooth manifolds where each tangent space $T_x\mathcal{M}$ is endowed with a scalar product $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$, called the Riemannian metric, and which completely characterizes the geometric structure of $\mathcal{M}$. Indeed, it makes it possible to define the length of a



(a)



(b)



(c)

Figure 7: Distribution of the land-cover categories in the HiChange dataset at different levels of the taxonomy.

smooth curve $\gamma : [a, b] \to \mathcal{M}$ as

$$L(\gamma) = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, \mathrm{d}t, \tag{11}$$

and thus geodesics as minimal-length paths between two points. We can then define the distance between two points as the length of such a geodesic. By extension, a curve $\gamma : \mathbb{R} \to \mathcal{M}$ which is a geodesic on every segment $[a, b]$ is also called a geodesic.

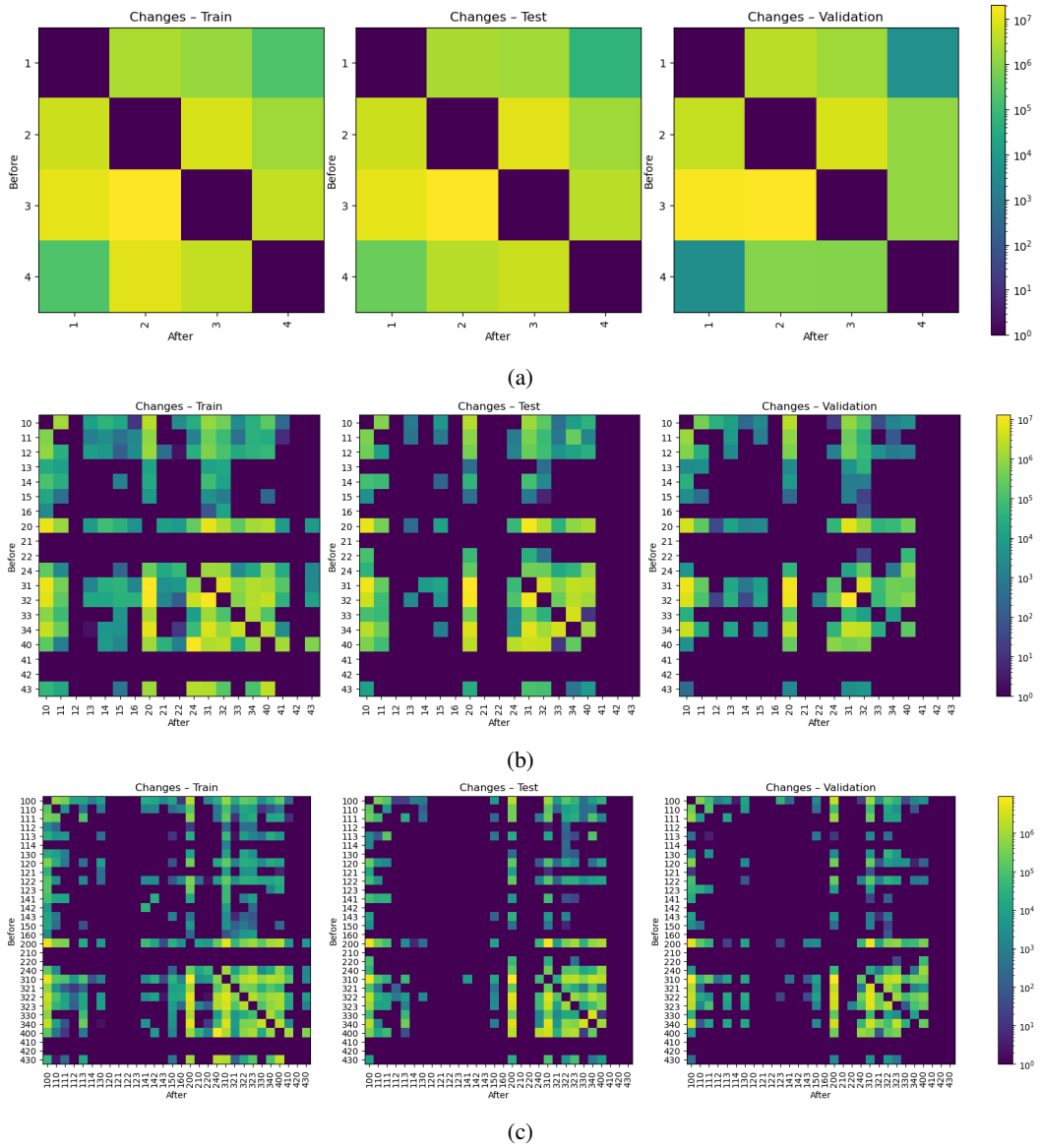


Figure 8: Semantic changes in HiChange. Confusion matrices of land-cover transitions for pixels labeled as change at the three hierarchical levels of the taxonomy.

Hyperbolic spaces are Riemannian manifolds with the particular property of having constant negative curvature. This curvature controls how much the space looks like a tree. The more negative, the more the geometry becomes comparable to that of an (infinite) tree. The closer to 0, the more the hyperbolic space gets similar to a Euclidean space. There exist different models of hyperbolic space that are, for a given dimension d and curvature $\kappa < 0$, all isometric to each other. The Poincaré ball model is one of them. Using $c = -\kappa$, it is defined as

$$\mathbb{B}_d = \left\{ z \in \mathbb{R}^d \mid \|z\|_2^2 < \frac{1}{c} \right\}, \tag{12}$$

endowed with the Riemannian metric $g_x(u, v) = \lambda_x^2 \langle u, v \rangle$, where $\lambda_x = \frac{2}{1 - c\|x\|_2^2}$ is called the conformal factor.

In such a space, geodesics are given by arcs of circle (or straight lines) orthogonal to the "boundary" $\mathbb{S}_c^{d-1} = \{z \in \mathbb{R}^d \mid \|z\|_2^2 = 1/c\}$ (which is not reachable in finite time), and the Riemannian distance

between two points $x, y \in \mathbb{B}_d$ is given by

$$d(x,y) = \frac{1}{\sqrt{c}} \cosh^{-1}\left(1 + 2c\frac{\|x-y\|_2^2}{(1-c\|x\|_2^2)(1-c\|y\|_2^2)}\right). \tag{13}$$

Given a tangent vector $v \in T_x\mathbb{B}_d$ in a point $x \in \mathbb{B}_d$, the exponential map $\exp_x : T_x\mathbb{B}_d \to \mathbb{B}_d$ returns $\gamma_v(1) \in \mathbb{B}_d$ where $\gamma_v : \mathbb{R} \to \mathbb{B}_d$ is the constant-speed geodesic passing through x in $t = 0$ with velocity $\gamma'(0) = v$, as such geodesic is unique. The exponential map therefore gives a way to move in the manifold $\mathbb{B}_d$ from the starting point x following a displacement vector v. Its expression is given by

$$\exp_x(v) = x \oplus \frac{1}{\sqrt{c}} \tanh\left(\sqrt{c}\|v\|_2\right)\frac{v}{\|v\|_2}, \tag{14}$$

where $\oplus$ denotes the Möbius addition, expressed, for $x, y \in \mathbb{B}_d$, as

$$x \oplus y = \frac{(1 + 2c\langle x, y\rangle + c\|y\|_2^2)x + (1 - c\|x\|_2^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|_2^2\|y\|_2^2}. \tag{15}$$

C.2 Busemann Function

In a complete Riemannian manifold $\mathcal{M}$, for a unit-speed geodesic $\gamma : \mathbb{R} \to \mathcal{M}$, the Busemann function along γ is defined as

$$B_\gamma : x \in \mathcal{M} \mapsto \lim_{t\to\infty} d(\gamma(t), x) - t, \tag{16}$$

when the limit exists. It can be interpreted as a sort of coordinate of the point x along γ.

This Busemann function is well-defined in Euclidean space, and for a unit-speed geodesic $\gamma_w : t \mapsto tw$, with $w \in \mathbb{S}^{d-1}$, going through the origin, we have $B_w(x) = -\langle x, w\rangle$. The Busemann function can therefore be seen as a generalization of the (negative) scalar product to other manifolds, and thus can be used as a tool for Riemannian machine learning.

In hyperbolic space, the Busemann function exists as well. For a prototype p at infinity ($p \in \mathbb{S}_c^{d-1}$), which can be seen as a direction, the unit-speed geodesic *going to* p is written:

$$\gamma_p(t) = \exp_0\left(\frac{\sqrt{c}t}{2}p\right) = \tanh\left(\frac{\sqrt{c}t}{2}\right)p. \tag{17}$$

The Busemann function can then be shown to be

$$B_p(x) = \frac{1}{\sqrt{c}} \log\left(\frac{c\|x-p\|_2^2}{1-c\|x\|_2^2}\right). \tag{18}$$

Its level-sets

$$H_{p,a} = \{z \in \mathbb{B}_d \mid -B_p(z) + a = 0\} \tag{19}$$

are called horospheres. In the Poincaré model, they are represented by spheres tangent to the prototype. They can be seen as hyperbolic equivalents for Euclidean hyperplanes. Other generalizations of hyperplanes in hyperbolic space exist, such as geodesic hyperplanes that have been used in different machine learning works on hyperbolic geometry Ganea et al. (2018b); Cho et al. (2019); Shimizu et al. (2020). However, one advantage of horospherical learning is that the distance from a point to a horosphere admits the simple expression

$$d(x, H_{p,a}) = |B_p(x) - a|, \tag{20}$$

which is analogous to the relation between Euclidean hyperplanes and scalar product. This expression is straightforward to compute and shows that the distance between two horospheres sharing the same prototype p is constant. In particular, this distance is independent of the points at which it is evaluated, which provides a natural interpretation of horospherical classification.

C.3 BUSEMANN VECTOR FIELDS

The change detection part of HyperFlow rely on the flow of a hyperbolic vector field. In a general manifold $\mathcal{M}$, several possible approaches exist to define and parameterize such a vector field. A straightforward approach is to see the manifold $\mathcal{M}$ of dimension d as a subset of a Euclidean embedding space $\mathbb{R}^n$ where $n \geq d$, and to naively project the vectors to their corresponding tangent space. However, this construction is not geometrically grounded and may prove inefficient.

We instead adopt the intrinsic approach proposed in Katsman et al. (2023). Let us assume there exists a parameterizable feature function $g_\theta : \mathcal{M} \to \mathbb{R}$ on the manifold. By considering K different parameter values $\theta_1, ..., \theta_K$, we can construct a map $f_\Theta : \mathcal{M} \to \mathbb{R}^K$ by concatenating the corresponding feature functions:

$$f_\Theta = [g_{\theta_1}, ..., g_{\theta_K}], \tag{21}$$

with $\Theta = (\theta_1, \dots, \theta_K)$. We interpret this function as a coordinate system on $\mathcal{M}$, although it is not required to be either injective or surjective. At a point $x \in \mathcal{M}$, its differential $D_x f_\Theta$ maps $T_x\mathcal{M}$ to $T_{f_\Theta(x)}\mathbb{R}^K \cong \mathbb{R}^K$. The corresponding pullback operator $(D_x f_\Theta)^*$ maps the dual space $(\mathbb{R}^K)^*$ of $\mathbb{R}^K$ to the dual space $(T_x\mathcal{M})^*$ of $T_x\mathcal{M}$. By identifying Euclidean spaces with their dual spaces, we can see the pullback operator as a map

$$(D_x f_\Theta)^* : \mathbb{R}^K \to T_x\mathcal{M}. \tag{22}$$

It is expressed by the gradients of the different coordinate features, which together form what we can see as a set of elementary vector fields.

Following Katsman et al. (2023), a vector field can then be defined on $\mathcal{M}$ as

$$V_{\Theta,\omega} : x \mapsto (D_x f_\Theta)^*\Big(n_\omega(f_\Theta(x))\Big) = \sum_{k=1}^{K} n_\omega(f_\Theta(x))_k \nabla g_{\theta_k}(x), \tag{23}$$

where $n_\omega : \mathbb{R}^K \to \mathbb{R}^K$ is any neural network and ∇ is the Riemannian gradient. While f_Θ feeds the neural network with some kind of coordinates, the outputs of this network serve as weights in a linear combination of the elementary vector fields. In the hyperbolic case, Busemann functions are natural candidates for feature functions, as proposed in Katsman et al. (2023).

Adaptation to Computer Vision Tasks. Under the above formulation, the vector field, when applied to the embeddings of the pixels of an image, does not take the spatial context into account. Consequently, the displacement of a pixel embedding is determined solely by its position in space, independently of the embeddings of neighboring pixels. In this work, since we argue that the spatial context provides valuable information, we instead replace the neural network n_ω, acting pixel-wise, by a neural network $N_\omega : (\mathbb{R}^K)^{H\times W} \to (\mathbb{R}^K)^{H\times W}$ that takes the full image of Busemann features as input (in our case, a UNet). The Busemann functions f_P and their gradients ∇B_{p_k} are however still computed pixel-wise. This creates a vector field on $(\mathbb{B}_d)^{H\times W}$, and is equivalent to applying the design defined by eq. (23) to $(\mathbb{B}_d)^{H\times W}$ as a product manifold, with the particular choice of weight-sharing feature functions $(g_{\theta_k,i,j})_{i,j,k}$ defined by

$$g_{\theta_k,i,j} : E \in (\mathbb{B}_d)^{H\times W} \mapsto g_{\theta_k}(E_{i,j}) \in \mathbb{R}, \tag{24}$$

for every $1 \leq i \leq H$, $1 \leq j \leq W$ and $1 \leq k \leq K$.

D IMPLEMENTATION DETAILS

Segmentation. Our segmentation model is composed of a DeepLabV3 model, based on a ResNet101 backbone and pretrained on ImageNet, together with the horospherical classification head from Berg et al. (2024). We train it on 60 epochs with a batch size of 4 and a learning rate of 0.01. It is fed with RGB images of size 512×512, obtained by randomly cropping the original images. We choose a hyperbolic embedding dimension of 2 in all our experiments, and we study the impact of the dimension in Supplementary Materials F.5. The prototypes from the horospherical head are optimized on the sphere, together with all the other weights, using a Riemannian stochastic gradient descent algorithm (Bonnabel, 2013).The weighting term β in eq. (7) is set to $1e-5$. During incremental steps, the new prototypes are initialized according to eq. (5), with $\epsilon_j \sim \mathcal{N}(0, \sigma^2 I_d)$ and dispersion $\sigma^2 = 0.1$.

Change detection. The change detection part of HyperFlow is trained on the embeddings produced by the segmentation model (which is, for this CD part, frozen) and learns the vector field introduced in section 3.5. As we choose to reuse the segmentation prototypes to define this vector field and to fix them, the only thing that needs to be learned is the network n_ω that we implement with a UNet architecture. The model is trained with an ADAM optimizer, with learning rate 10^{-4}, batch size 4 on 20 epochs.

Incremental settings. As more incremental steps are considered, the segmentation model's training is resumed for 60 more epochs with the same hyperparameters. Note that its architecture is slightly different as its (hyperbolic) head now contains new prototypes. The change detection model's architecture is then redefined to take into account these new prototypes, which leads to a higher number of channels in the UNet, and is retrained from scratch.

For HiChange, classes are introduced progressively according to the incremental scheme summarized in table 6. The table reports only the classes introduced at each step; the full class set at step t includes all classes introduced in previous steps together with those added at step t. Newly introduced classes are added simultaneously rather than one after another. Class notation is consistent with fig. 5.

For SECOND and Hi-UCD-mini, no incremental setting is used. The tasks T_6^0 and T_9^0 correspond to the standard setting where all dataset classes are considered from the beginning.

Table 6: Incremental tasks for the HiChange dataset. The table summarizes the classes considered at each incremental step.

Step	Task	Classes introduced
0	T_4^0	$[1, 2, 3, 4]$
1	T_7^1	$[31, 32, 33, 34]$
2	T_{13}^2	$[10, 11, 12, 13, 14, 15, 16]$
3	T_{16}^3	$[40, 41, 42, 43]$
4	T_{20}^4	$[20, 21, 22, 23, 24]$
5	T_{22}^5	$[321, 322, 323]$
6	T_{32}^6	$[110, 111, 112, 113, 114, 121, 122, 123, 141, 142, 143]$

Semantic change detection maps and consistency between change and segmentation maps. Following established work in SCD, our formulation deliberately keeps (and evaluates) the change map and the semantic segmentation maps as separate outputs rather than combining them into a single *from-to* transition map. The latter would couple the errors of individual predictions without providing additional information. The notion of change consistency in our work should therefore be understood at the level of the incremental task rather than as a hard pixel-wise constraint. HyperFlow does not enforce change and semantic prediction to be consistent at every pixel. However, as the semantic taxonomy evolves, both semantic segmentation and change detector evolve to enable the change maps to describe changes within the new semantic classes considered.

E Additional Qualitative Results

This section presents additional qualitative results illustrating the behavior of HyperFlow across datasets.

E.1 HiChange

For the HiChange dataset, we provide additional visualizations illustrating different aspects of the proposed approach. First, fig. 9 presents additional examples of segmentation prediction across incremental steps, together with the associated hyperbolic embeddings. These examples illustrate the capacity of HyperFlow to capture semantic information of increasing complexity as well as the evolution of the organization of the embedding space as the taxonomy is progressively refined.

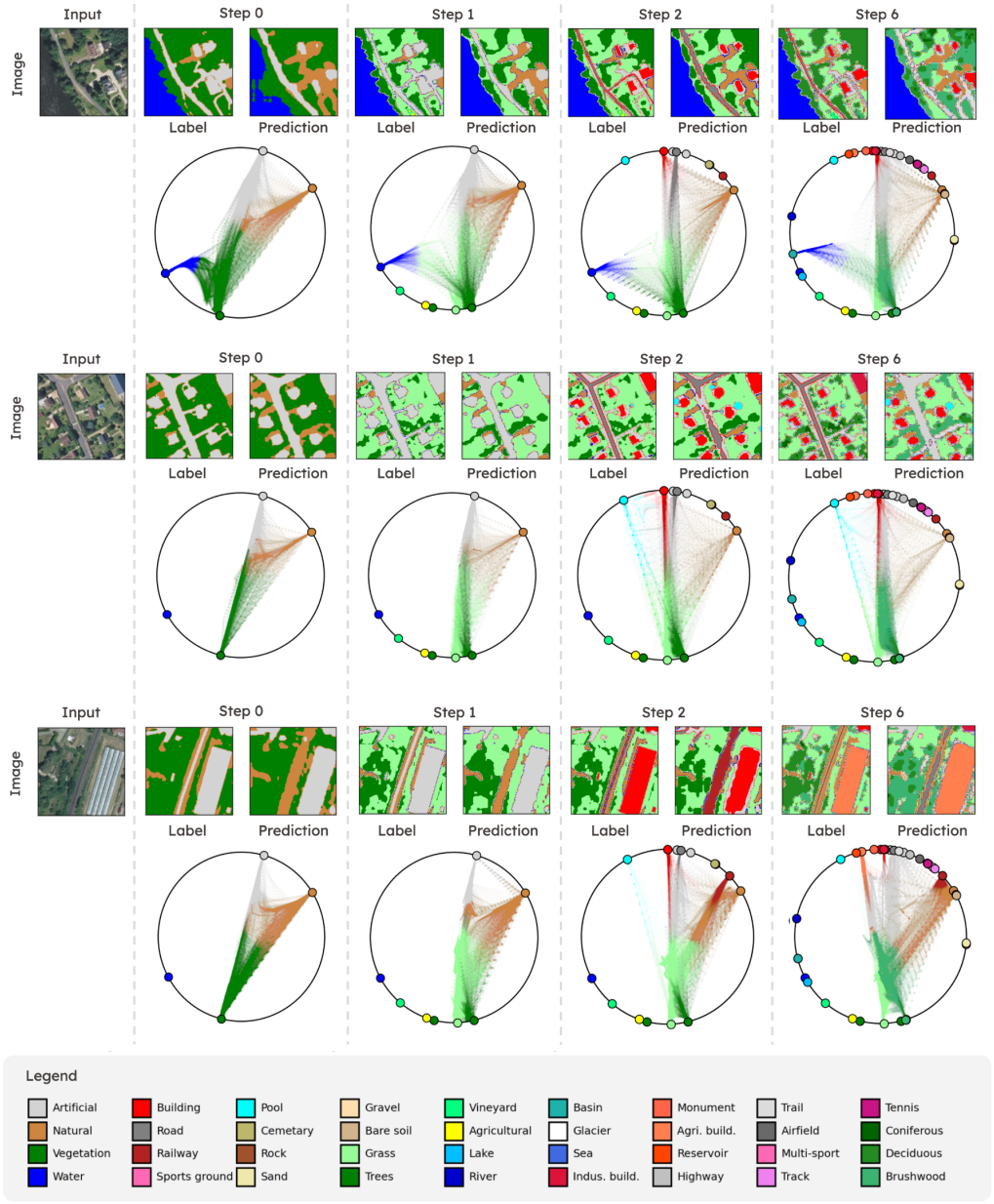


Figure 9: Qualitative examples on HiChange of segmentation across incremental steps and associated embedding space visualizations. For each step, we show the predicted segmentation together with the corresponding hyperbolic embeddings, illustrating how the embedding space evolves as the taxonomy is refined and new classes are introduced. The color of each embedding corresponds to the ground-truth semantic label.

Second, fig. 10 presents additional examples of semantic change detection for the second incremental step (T_{15}^2). For each pair of images, we show the predicted segmentation at both timestamps, the resulting change map, and the embeddings of the first and second image, as well as the predicted embeddings of the second image. These examples illustrate how pixels corresponding to changes exhibit larger displacements in the hyperbolic space compared to unchanged regions, enabling HyperFlow to detect changes in an unsupervised way.

Finally, fig. 11 compares both prediction and embedding spaces obtained under incremental and non-incremental settings. These visualizations provide an intuitive illustration of the effect discussed in the main paper: incremental refinement preserves the global hierarchical organization of the embedding space whereas directly training on the full fine-grained taxonomy tends to produce less structured representations. For the non incremental but hierarchical scenario, the hierarchical organization is also preserved. However, because the model must learn all fine-grained categories from scratch, the task is harder and the resulting performance is lower.

E.2 SECOND

For the SECOND dataset, incremental scenarios are not possible. We therefore present qualitative examples of SCD only. In fig. 12, we visualize the predicted segmentation at both timesteps, the predicted change map and the embeddings of the first and second image, compared with the predicted embeddings of the second image.

E.3 HI-UCD-MINI

Similarly, for the Hi-UCD-mini dataset, we provide qualitative examples of SCD in fig. 13. We show for each pair of images the segmentation and change predictions, as well as the associated embedding space visualizations.

F ADDITIONAL EXPERIMENTS

F.1 INFLUENCE AND CHOICE OF THE THRESHOLD

The CD part of HyperFlow, based on hyperbolic semantic flows, needs a threshold at inference to transform a continuous reconstruction error into a binary prediction. To obtain binary CD from the distance between the embeddings without relying on ground-truth labels, we require an unsupervised thresholding method. Otsu's method (Otsu, 1979), although commonly used, is unsuitable here, as it assumes a bimodal histogram, whereas our distance distribution is dominated by the unchanged mode with a much smaller tail of change scores, as observed in fig. 14. Under this imbalance, Otsu does not provide a suitable threshold. We instead use the Triangle algorithm (Zack et al., 1977), which is designed for exactly this type of skewed distribution: it selects the threshold as the point of maximum perpendicular distance between the histogram and the line connecting its peak to its tail.

The triangle method, computed on the validation set, for HiChange and the base task T_4^0 gives the value $d_{\text{thresh}} = 4.2$. We study the impact of this threshold in fig. 15. 4.2 appears to be a relevant choice, although not being the best one. While the performance depends on the choice of the threshold, we note that the range in which our model outperforms other unsupervised methods remains large. We therefore keep this value for all experiments involving HiChange. The threshold value is however adapted for the other two datasets, and when using SAM3 and AnySat as embeddings encoder, as the distribution of the change scores may vary significantly in those cases. Similarly, when considering the Euclidean vector fields, the thresholds are adapted to the range of change scores values encountered. The values retained are summarized in table 7.

F.2 INFLUENCE OF THE BACKBONE ARCHITECTURE

All experiments presented in the main paper rely on a single backbone (Deep-LabV3 with a ResNet-101 backbone pre-trained on ImageNet) to ensure fair comparison across methods. We now investigate how the choice of backbone affects both segmentation and change detection performance. In particular, we consider a simpler UNet backbone. Results are provided in table 8. As expected, the performance obtained with the UNet backbone is lower than with DeepLabV3. This difference is

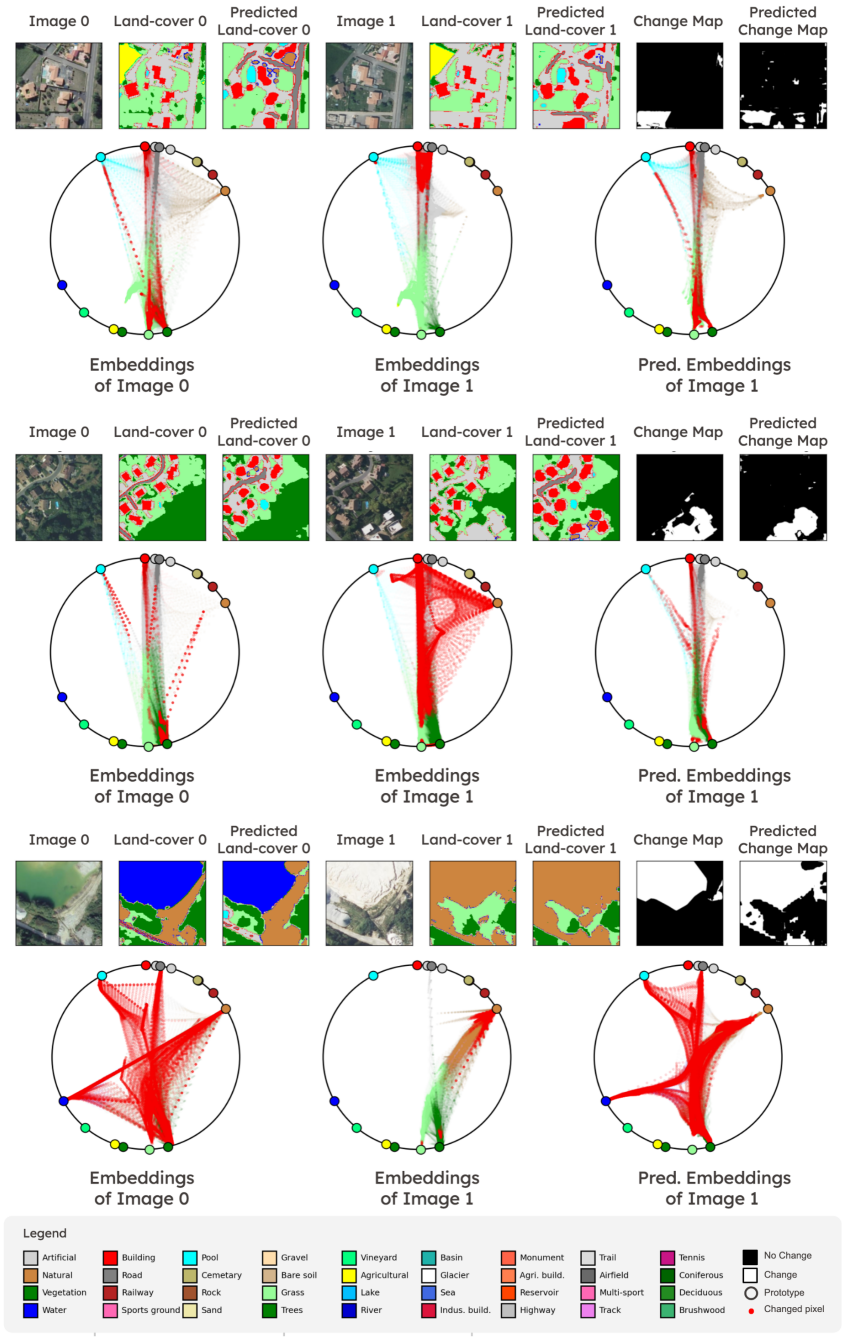


Figure 10: Qualitative examples of SCD at task T_{15}^{2}. For each image pair, we show the predicted segmentation at both timesteps, the predicted change map, the embeddings of the two images and the predicted embeddings of the second image. Pixels corresponding to true semantic changes (in red) exhibit larger displacement in the embedding space, that are not captured by the learned flow and therefore not accounted for in the predicted embeddings.

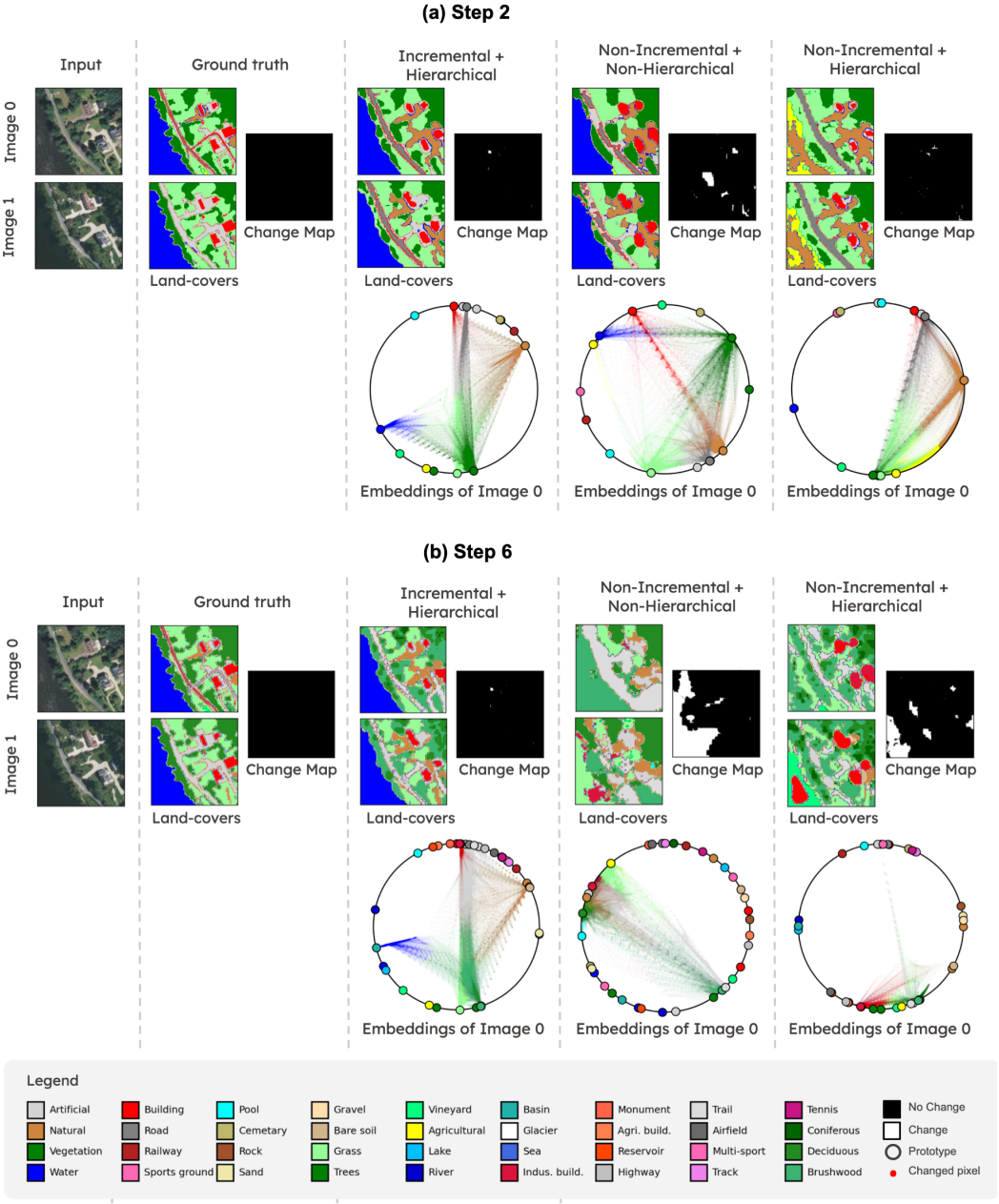


Figure 11: Comparison between incremental and non-incremental scenarios. We visualize the segmentation and change predictions at (a) T_{13}^{2} and (b) T_{32}^{6}, along with corresponding embedding spaces. Incremental refinement preserves the hierarchical organization, enabling better prediction, whereas directly learning fine-grained categories leads to less structured embeddings and lower performances.

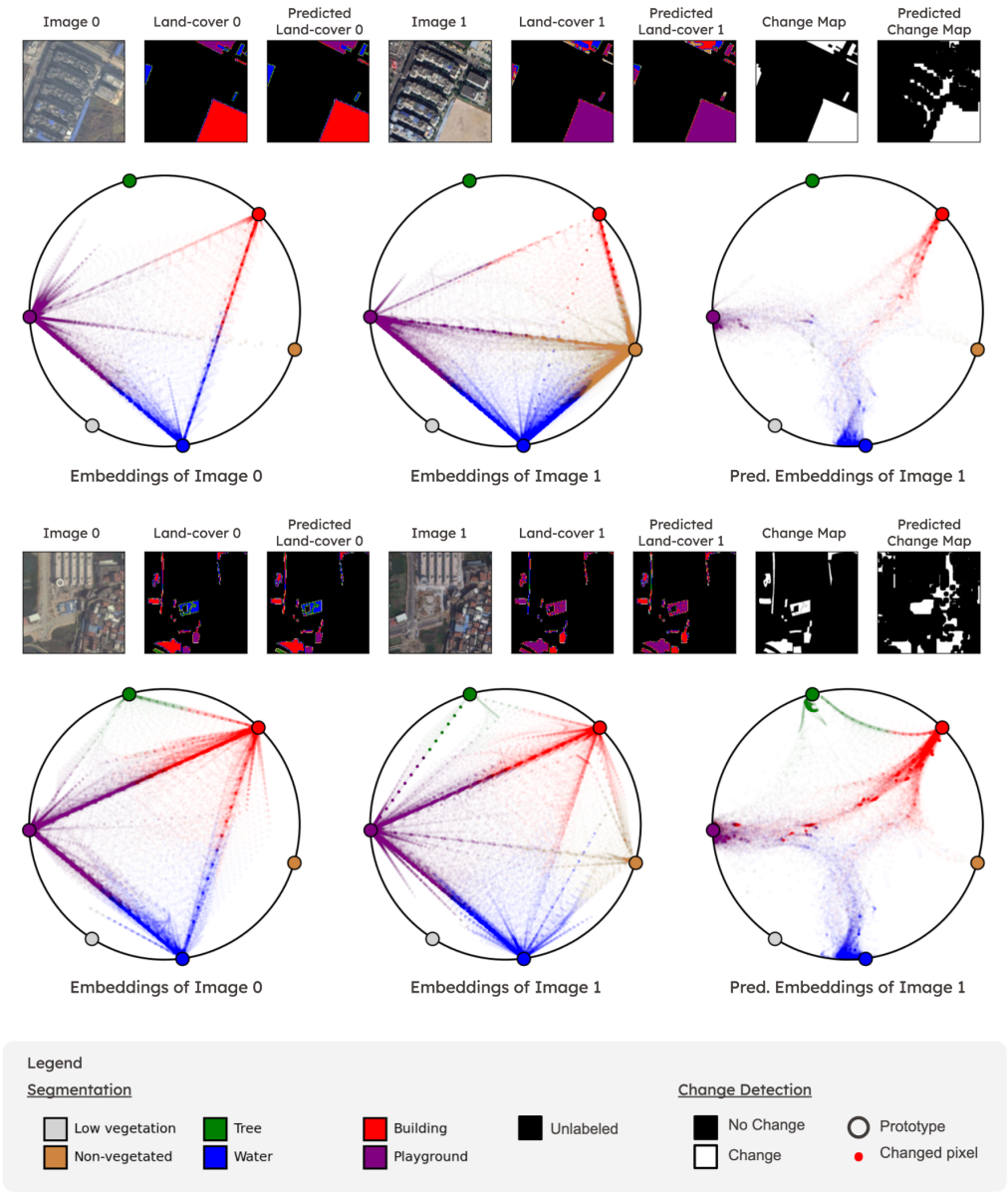


Figure 12: Qualitative examples on the SECOND dataset. For each image pair, we show segmentation and change prediction, and the associated embeddings.

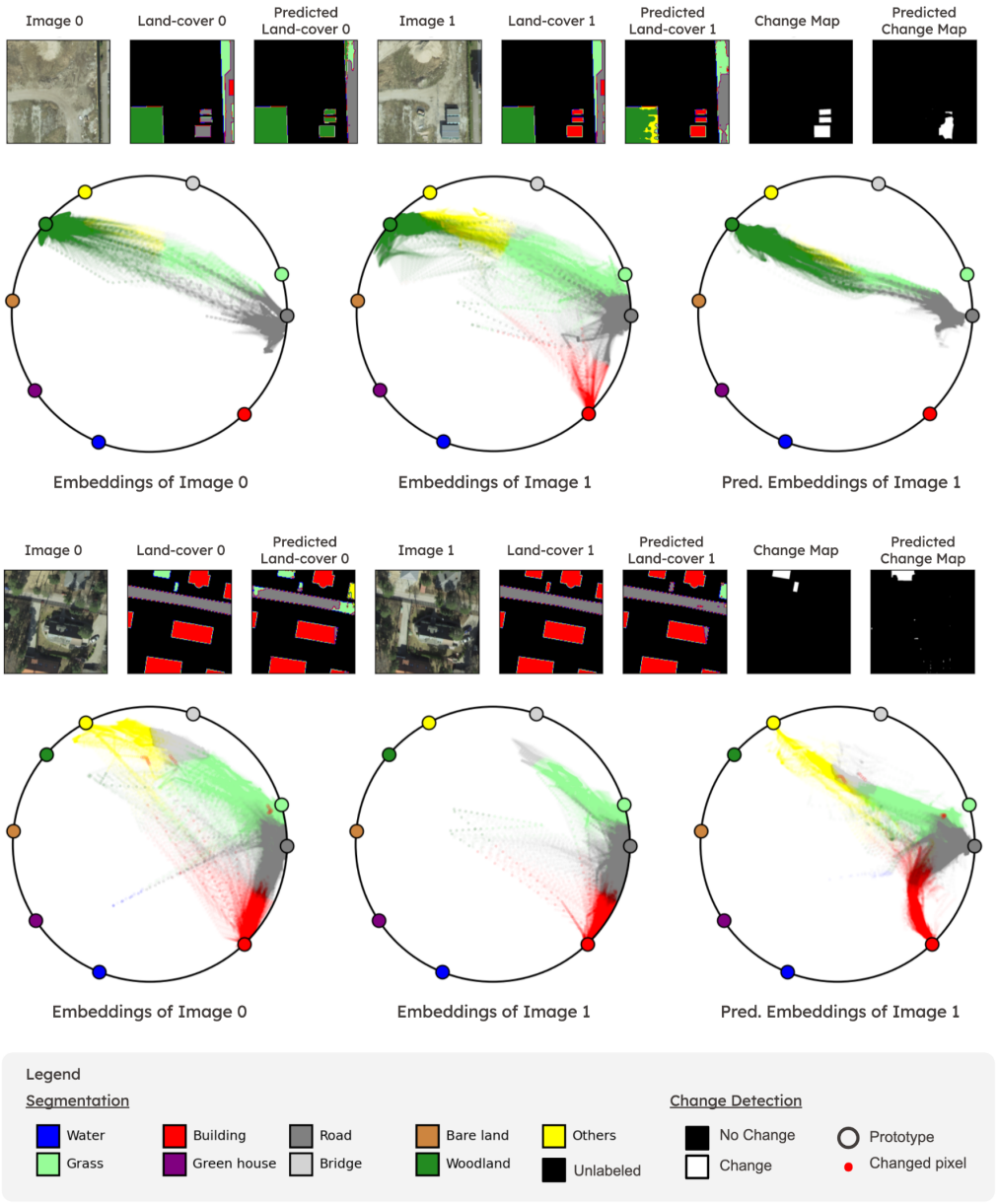


Figure 13: Qualitative examples on the Hi-UCD-mini dataset. For each image pair, we show segmentation and change prediction, and the associated embeddings.

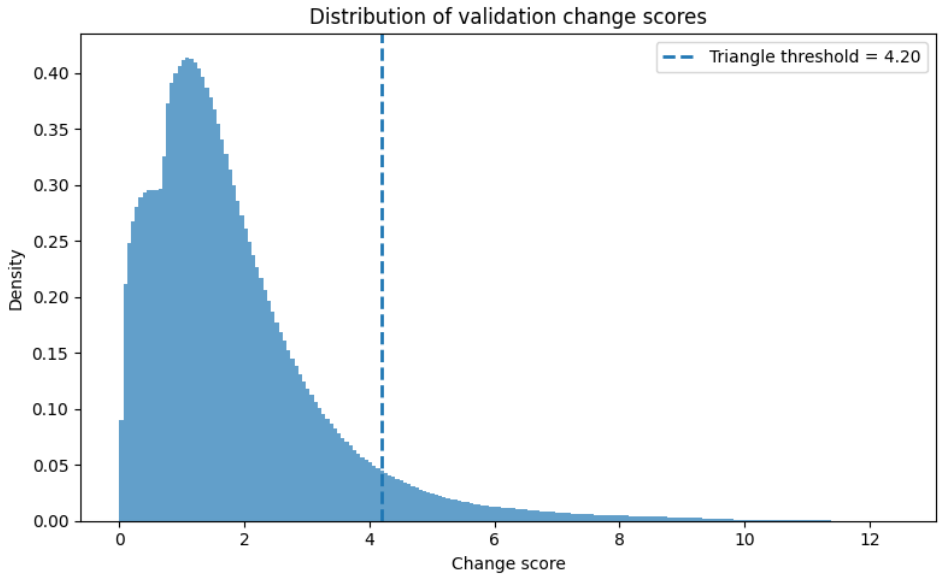


Figure 14: Distribution of change scores and triangle threshold for the HiChange dataset and the base task T_4^0.

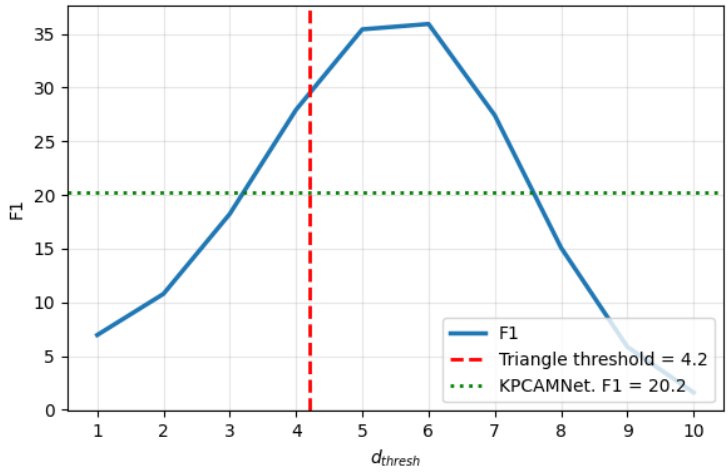


Figure 15: Influence of d_{thresh} on CD performance. We present the F1 score of HyperFlow with respect to the threshold used, for HiChange and the base task T_4^0. Results are compared with the best unsupervised CD baseline, KPCAMNet.

mainly explained by the smaller model size (1.3M parameters compared to 58M), and the absence of pretraining. Consequently, segmentation performance decreases, which impacts CD and SCD, as good change detection relies on the quality of the underlying segmentation. Nevertheless, it can be observed that for T_4^0, the CD performance for the UNet backbone (F1$=$ 14.5) remains similar to those of the unsupervised methods reported in table 1. This result indicates that HyperFlow still provides effective CD even when relying on a weaker segmentation model.

F.3 Do we really need to use the segmentation prototypes?

The current formulation reuses the Busemann prototypes learned for segmentation as anchors for the vector field. An alternative is to use uniformly distributed prototypes, either learned or fixed. While this provides greater flexibility and allows HyperFlow to operate without segmentation-derived em-

Table 7: Thresholds values obtained from the triangle method. Unless stated otherwise, these values are used for all experiments.

Dataset	Embeddings	Hyperbolic/Euclidean	Threshold
HiChange	Segmentation	Hyp.	4.2
HiChange	Segmentation	Euc.	0.2
HiChange	Segmentation	Euc-Bus.	0.2
HiChange	SAM3	Hyp.	6.3
HiChange	AnySat	Hyp.	3.6
SECOND	Segmentation	Hyp.	7.0
SECOND	SAM3	Hyp.	6.0
SECOND	AnySat	Hyp.	5.4
Hi-UCD-mini	Segmentation	Hyp.	5.1
Hi-UCD-mini	SAM3	Hyp.	6.0
Hi-UCD-mini	AnySat	Hyp.	5.6

Table 8: Influence of the backbone architecture on the performances of HyperFlow.

Metric	Backbone	T_4^0	T_7^1	T_{13}^2	T_{16}^3	T_{20}^4	T_{22}^5	T_{32}^6
Segm. [mIoU]	**DeepLabV3**	62.0	56.2	33.7	28.4	23.5	20.7	16.1
	UNet	50.5	24.3	12.3	9.1	8.0	6.9	4.7
CD [F1]	**DeepLabV3**	29.6	29.6	29.8	29.2	27.7	31.6	26.0
	UNet	14.5	16.3	17.1	17.0	17.2	18.8	17.3
SCD [SCS]	**DeepLabV3**	39.0	34.4	22.4	19.9	17.3	17.6	13.8
	UNet	31.1	16.7	10.3	8.8	8.6	8.5	7.1

beddings, which is particularly relevant for non-semantic CD tasks or when using general embeddings not obtained from segmentation pretext tasks, as in table 1 with SAM3 and AnySat, it may discard part of the semantic information encoded in the learned segmentation prototypes. Table 9 compares CD performance using prototypes derived from the segmentation task and prototypes independent of segmentation, both with and without prototype learning. While using non-segmentation prototypes leads to a performance drop, the resulting CD performance remains competitive, demonstrating that HyperFlow can also operate on prototypes not derived from a segmentation pretext task. Learning the prototypes partially closes the gap with segmentation-based prototypes, although not entirely. This can be explained by the fact that CD ultimately concerns changes in semantic content. Segmentation embeddings and prototypes therefore provide a natural basis for modeling this semantic flow.

Table 9: Comparison of segmentation-based and independent prototypes for CD on the HiChange dataset, showing the impact of prototype initialization on HyperFlow performance.

Segmentation prototypes	Trained	Acc.	F1	Pre.	Rec.
✓	✗	92.9	29.2	20.2	52.6
✗	✗	93.1	26.9	19.1	45.5
✗	✓	93.9	27.9	20.8	42.5

F.4 Does working in Hyperbolic space help?

One of the main hypotheses of this work is that exploiting hyperbolic geometry can improve the CI-SCD task. While Hindel et al. (2024) demonstrated the benefits of hyperbolic representations for CISS on natural images, it remains unclear whether these advantages also extend to change detection (CD) in remote sensing imagery. To investigate this question, we consider two Euclidean counterparts to the proposed Busemann vector field framework. The first preserves the overall methodology while replacing the hyperbolic Busemann function with its Euclidean analogue, namely a scalar product. This variant therefore retains the same formulation and training procedure, differing only

in the underlying geometry. The second removes the Busemann formulation entirely, as it is no longer required in Euclidean space, and instead trains a neural network to directly predict the next embeddings. We refer to these two variants as *Euc-Bus.* and *Euc*, respectively, and compare their performance on the CD task with the proposed hyperbolic approach in table 10a.

We observe a clear superiority of the hyperbolic formulation. However, in our experiments, the threshold obtained using the triangle-based method for the Euclidean variants was not optimal. This can be attributed to the quite different distribution of embeddings in Euclidean space: compared to hyperbolic embeddings, the embeddings exhibit smaller pairwise distances, as those distances do not grow exponentially. To provide a fairer comparison and remove artifacts of thresholding behavior, we therefore manually selected the optimal threshold for both the *Euc-Bus.* and *Euc* formulations. Table 10b reports their performance using these thresholds, alongside that of the standard hyperbolic HyperFlow with its optimal threshold ($d_{\text{thresh}} = 6$, see fig. 15). Although the performance gap is reduced, the overall trend remains unchanged: the hyperbolic version outperforms its Euclidean counterparts.

Table 10: Comparison of hyperbolic and Euclidean formulations for HyperFlow on the HiChange dataset and the base task T_4^0, using (a) triangle thresholds and (b) optimal thresholds.

(a)

Space	Acc.	F1	Pre.	Rec.
Hyp.	92.9	29.6	20.5	52.8
Euc-Bus.	82.1	18.9	10.8	74.7
Euc.	83.1	19.7	11.4	73.4

(b)

Space	Acc.	F1	Pre.	Rec.
Hyp.	97.1	35.9	47.1	29.0
Euc-Bus.	94.4	34.9	25.8	53.8
Euc.	94.4	35.5	26.5	53.7

F.5 Embedding Dimension

We evaluate the impact of the embedding dimension d on performance for the base task T_4^0. Results are reported in table 11. Increasing the embedding dimension does not provide a clear improvement on both semantic segmentation and CD performance. The results fluctuate across dimensions, without showing a clear superiority. This behavior is likely influenced by the choice of the threshold that might not be optimal in other dimensions. In our experiments, we used the same threshold value ($d_{\text{thresh}} = 4.2$) for all dimensions, while the optimal threshold can be expected to vary slightly with the embedding dimension.

Despite these results and the fact that higher dimension could potentially provide better performances, especially if we consider the SCD performance, we keep $d = 2$ in the main experiments for three reasons. First, this configuration already surpasses competing methods. Second, 2-dimensional embeddings enable direct visualization of the learned representations, which provides useful insights into the semantic structure and the learned flow. Third, it reduces the number of parameters in the model. While higher dimensions slightly improve the SCD (SCS) score, the gains remain limited and do not justify the additional parameters.

F.6 Failure Cases

Figure 16 presents typical examples of failure cases. Two main sources of errors can be identified. The first is errors in the predicted segmentation. As illustrated by the first two examples, inaccuracies in the predicted land-cover propagate to the change map. This is consistent with the results discussed in section 5.1: the quality of the embeddings directly affects the quality of the resulting CD, and, for the segmentation embeddings used here, this quality is closely related to the accuracy of the land-cover segmentation. The second reason comes from errors in the ground-truth annotations, as observed in the last two examples.

F.7 Computational Complexity

Our model has 59.2M parameters (225.6 MB) and requires 276.7 GFLOPs per forward pass.

Table 11: Impact of the embedding dimension d on segmentation, CD, and SCD performance for HiChange and the base task T_4^0.

Dimension	Segm. [mIoU]	CD [F1]	SCD [SCS]
2	62.0	29.6	39.0
16	60.8	21.7	37.1
64	62.2	28.4	39.8
128	58.6	32.6	41.8

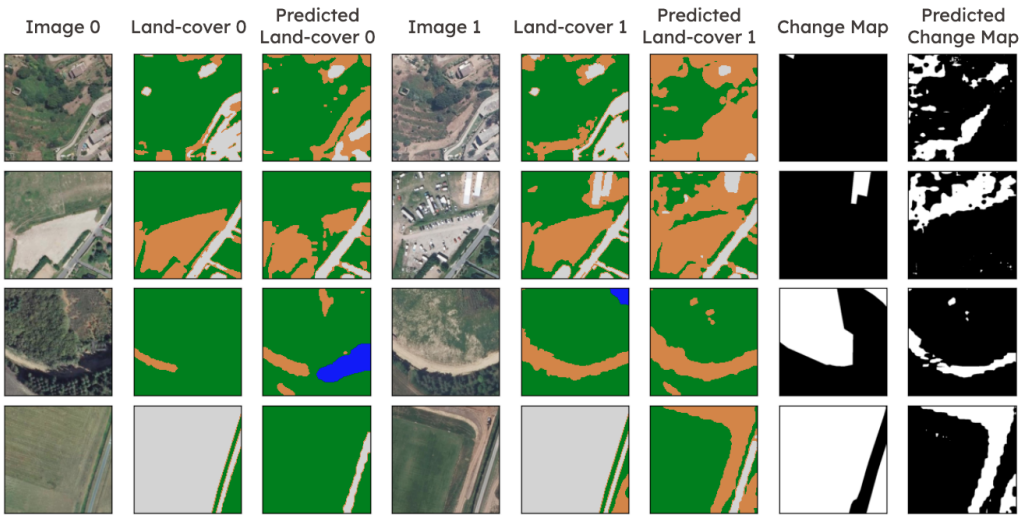


Figure 16: Failure cases of HyperFlow. The first two examples illustrate errors caused by inaccuracies in the predicted land-cover, which propagate to the change map. The last two illustrate cases where the ground-truth annotations present errors.