Proceedings of Machine Learning Research 275:1–17, 2025 TerraBytes Workshop at the 42nd International Conference on Machine Learning

Smoothing Continual Segmentation Oscillations with Latent Domain PPCA Decoder

Marie-Ange Boum marie-ange.boum@onera.fr
and **Pierre Fournier** pierre.fournier@onera.fr
and **Dawa Derksen** dawa.derksen@cnes.fr
and **Stéphane Herbin** stephane.herbin@onera.fr

Editors: Nicolas Audebert, Hossein Azizpour, Valentin Barriere, Javiera Castillo Navarro, Mikolaj Czerkawski, Heng Fang, Alistair Francis, Valerio Marsocci, Andrea Nascetti, Ritu Yadav

Abstract

We study Domain Incremental Learning for the semantic segmentation of Earth Observation images. We demonstrate that controlling the oscillation of performance when a new domain arrives is more critical than controlling catastrophic forgetting. We propose an exemplar free architecture that combines a large pre-trained network well adapted to dense image processing (DINOv2) and a generative decoder head based on Probabilitic Principal Component Analysis (PPCA). We validate our approach on the FLAIR#1 high resolution dataset, which is structured as a sequence of domains.

Keywords: domain incremental learning, semantic segmentation, foundation models, dinov2, ppca

1. Introduction

Earth Observation (EO) datasets are massive and heterogeneous, continually capturing vast, evolving regions of the Earth's surface with different sensors and across multiple time points. However, local and global changes, driven by seasonal cycles, land use dynamics and climate change, render their statistical distribution inherently non stationary.

Our objective is to scale a semantic segmentation model to high resolution imagery, a model that delivers the same level of accuracy on data captured anywhere and at any time. Retraining globally on a growing dataset is virtually impossible, we therefore adopt incremental learning schemes that update the model's parameters whenever new data arrive, thereby progressively broadening its generalisation to unseen distributions. Incremental Learning or Continual learning (CL), a paradigm in which a model is updated incrementally on a non stationary data stream while preserving previously acquired knowledge (Wang et al., 2023), is therefore indispensable for processing such data.

With EO imagery, we can associate each change of location, season or other condition of acquisition to a new domain. The challenge we are seeking to meet is one of Domain Incremental Learning (DIL), a fundamental CL scenario, that allows for models to sequentially train on new distributions, while preserving earlier knowledge, thereby ensuring robust, long term performance. In this work, we focus on a growing set of highly similar domains produced by a single aerial imaging system that yields data streams with small, smooth shifts driven by geographic location. (van de Ven et al., 2022; Mirza et al., 2022; Kalb et al., 2021; Wang

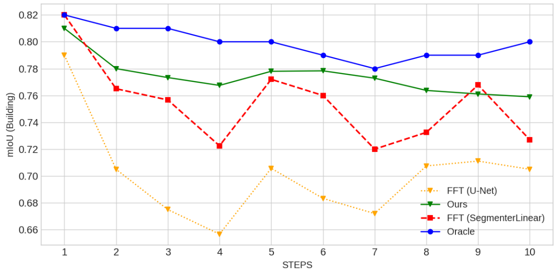


Figure 1: Although catastrophic forgetting is the usual concern in continual learning, our DIL study reveals a different obstacle: pronounced performance oscillations when a new domain is introduced. This phenomenon is clearly noticeable after fine-tuning the model when new domains arrive sequentially, whether using a vanilla U-Net architecture (yellow curve) or a large pre-trained feature extractor with a linear predictor (red curve). Our method (green curve) controls the oscillations via an exemplar-free approach and a PPCA-based generative classifier. The blue curve shows the oracle performance obtained by full training on all domains seen up to the current step on the FLAIR#1 dataset.

et al., 2022; Garg et al., 2022; Shi and Wang, 2023). Naive approaches to address continual learning, such as sequentially finetuning the model with incoming data distributions, are prone to catastrophic forgetting, mainly caused by a shift in the feature space (Castro et al., 2018) when learning new knowledge (Caccia et al., 2021; Driscoll et al., 2022), and occasion a steady performance decline. To tackle this issue, most methods use encoder-decoder architectures and focus on encoder focused remedies, such as parameter isolation, distillation or replay, in order to curb feature drift (Liu et al., 2025; Rui et al., 2023; Alfarra et al., 2024; Huang et al., 2024b; Saporta et al., 2022). However, in our specific case of DIL, when sequentially finetuning an encoder-decoder architecture like U-Net, the main issue is not catastrophic forgetting (Fig.1, yellow curve): rather, as we observe partiel recoveries.

Empirical evidence (e.g. (Prabhu et al., 2020) and related replay baselines) shows that forgetting is mitigated most effectively when the initial feature space is already adapted to the domain, as a consequence, the feature initialisation phase becomes a decisive factor in overall performance. The advent of self-supervised vision foundation models (FMs) promises a new baseline: pretrained on billions of natural images, these models supply high-quality, reusable features that lead to state-of-the-art segmentation with minimal fine-tuning (Zhou et al., 2024). Yet the standard workflow — sequential finetuning of the whole FM — produces an oscillatory performance pattern that is more pronounced than with a vanilla U-Net (Fig.1, red curve). Because existing CL methods rely on complex, tightly coupled decoder architectures that obscures how the encoder's representation can help mediate the plasticity–stability trade-off, it is unclear how a foundation model can be leveraged to mitigate performance degradation in a continual setting. Our study addresses this issue in a domain incremental setting.

Contributions In this paper, we make the following contributions:

- We analyze the oscillations that occur when implementing DIL, a question that has not been addressed in the literature, and hypothesize their origin from experiments on remote sensing dataset,

- We establish an architecture for leveraging an FM in a DIL setting, specifically tailored for semantic segmentation,
- We develop a simple algorithm that limit oscillations in DIL
- And finally, we develop a protocol for evaluating DIL on a HR remote sensing dataset built from FLAIR#1

2. Related Works

2.1. Domain Incremental Semantic Segmentation

Most CL methods for semantic segmentation methods have focused on class-incremental benchmarks and very few address domain incremental semantic segmentation. Existing studies either define domains via drastically different visual styles—such as those in DomainNet (Peng et al., 2019)—or are limited to only a handful of domains (Huang et al., 2024c; Rui et al., 2023).

Domain incremental semantic segmentation methods navigate the plasticity–stability trade-off through a mix of parameter isolation, distillation, and synthetic replay. For example, (Liu et al., 2025; Rui et al., 2023) inject domain-specific adapters into the encoder and instantiate fresh decoder heads for each domain, using feature- and output-level distillation to anchor past knowledge. SimCS (Alfarra et al., 2024) foregoes module freezing altogether by generating on-the-fly simulated batches that regularize both encoder and decoder. (Huang et al., 2024b) strikes a balance between approaches that leave the encoder untouched and those that adapt it, by freezing an initial feature extractor while fine-tuning later encoder layers and a single-branch decoder under strict distillation constraints. Finally, (Saporta et al., 2022) achieves encoder plasticity via full fine-tuning with distribution- and feature-level distillation, paired with dual decoder heads, a specialist head for rapid adaptation and a KL-distilled generalist head to preserve earlier mappings.

However, no existing DIL segmentation method, especially for high-resolution EO imagery, has paired a large, self-supervised foundation model encoder with a light decoder that lets us disentangle and quantify the respective contributions of an encoder's representations to the plasticity–stability trade-off.

2.2. Large Pre-trained Models for Continual Learning

Most continual-learning methods built on large, pretrained Vision Foundation Models (VFMs) address the class-incremental setting, where the backbone remains frozen and adaptation is confined to lightweight heads or regularizers. For example, RanPAC, Randumb and TSVD (McDonnell et al., 2023; Prabhu et al., 2020; Peng et al., 2025) append shallow decoders to fixed feature extractors; Lee and Wang (Lee et al., 2023) introduce small regularization terms to maintain CLIP's feature stability; and SimpleCIL (Zhou et al., 2025) employs fixed class prototypes from the embedding space. Similarly, (Wang and Barbu, 2022) train one PPCA head per class on frozen features, classifying via Mahalanobis distances, (Ostapenko et al., 2022) compare an (MLP), a (NMC) and a SLDA heads under the same paradigm. These "head-only" schemes excel at adding new classes but all assume a static input distribution and a growing label set.

By contrast, DIL methods aim to handle shifts in input distributions—such as weather or lighting changes—without duplicating model parameters for each domain. (Panos et al., 2023) propose a strategy very much like ours but for CIL and image classification: they perform a single encoder-only update during the first session and then freeze the entire feature extractor to preserve consistent representations across all subsequent domains. (Mirza et al., 2022)'s DISC introduces a two-stage pipeline: during a streaming phase, only the last transformer block is fine-tuned on incoming weather-shifted batches using a low learning rate and gradient masking, and during an offline phase a small memory buffer is used for replay with an annealed schedule. It stores only first- and second-order statistics for each condition, enabling immediate plug-and-play adaptation to rain, fog, or snow without retraining. Similarly, our approach retains only class-specific first- and second-order feature statistics per domain, using them to update a lightweight generative decoder while keeping the encoder frozen.

Building on FLAIR#1 dataset (Garioud et al., 2022), adapted for scene classification, DIPPCA (Boum et al., 2024) introduces a prototype-based Gaussian head for domain-incremental classification in remote sensing, where class parameters are updated incrementally in feature space. However, this approach is demonstrated only on simplified image-level tasks and does not address dense per-pixel segmentation. Our framework extends this idea by leveraging probabilistic PCA's closed-form moment-matching to update both the mean and covariance of each class distribution at the patch level. This mechanism enables adaptation to domain shifts in the inherently more complex setting of dense semantic segmentation in remote sensing.

Unlike DIPPCA, which maintains a single head for all domains, our framework allocates a distinct PPCA model for each domain. By explicitly encoding domain identity in each PPCA head, we can smoothly introduce new domains in high-resolution EO segmentation, without conflating current-domain adaptation with prior-domain knowledge.

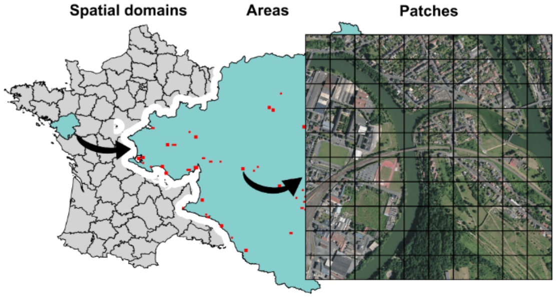


Figure 2: FLAIR#1 Dataset : Spatial Domains defined by French departments.

3. Problem Formulation

3.1. DIL benchmark for remote sensing

We derive a benchmark from the FLAIR#1 Dataset (Garioud et al., 2022) to evaluate Remote Sensing Domain Incremental Learning.

The latter consists of 50 spatial domains – each corresponding to a French department (see Figure 2) – that capture the diversity of landscapes and climates across metropolitan France. It contains 77,412 patches (each 512 × 512 pixels at 0.2 m GSD) covering about 810 km^2, annotated with nineteen semantic classes.

In order to focus on DIL dynamics, we restrict our study to the segmentation of the building class using RGB channels only, a common task in RS. First and foremost, as part of this work, we select a sequence of ten departments in northern France, ordered as follows: D70, D21, D60, D23, D35, D52, D14, D41, D49, and D78. Building on (Boum et al., 2024), we investigate domain shifts induced by geographical variations across the departments of the FLAIR#1 dataset. An extension of this work would be to evaluate other sequences of domains.

3.2. Domain Incremental Learning (DIL)

The goal of Domain Incremental Learning is to design incrementally a predictor that can be applied on a large distribution of data, where only part of the whole distribution – a domain – is available from sample data at each incremental session. At the end of the learning sequence, the final predictor is expected to become a universal expert over the union of all domains.

A sequence $[D_t]_{t=1}^{T}$ of T domains arrives sequentially. For semantic segmentation of images, each domain is a joint probability distribution over a space of dense images and labels $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^{H \times W \times 3}$ represents the set of images, and $\mathcal{Y} = \{1, \ldots, C\}^{H \times W}$ represents the set of labels that encode the correspondence between each pixel and a class label selected from C possible categories. In DIL, the set of possible categories is kept fixed during sessions, only the input distribution varies.

The goal of DIL is to update at each session t a predictor $y = f(x; \theta_t)$ defined by parameters θ_t, using the N_t data $\{x_t^i, y_t^i\}_{i=1}^{N_t}$ sampled from domain D_t and the previous predictor parameters θ_{t-1}. At the end of each session, it is expected that the predictor minimizes the average prediction error $\bar{\epsilon}_T$ on both current domain D_T and historical domains $\{D_k\}_{k=1}^{T-1}$:

$$\epsilon_t(\theta) = E_{D_t}[\ell(f(X; \theta), Y)] \tag{1}$$

$$\bar{\epsilon}_T = \frac{1}{T} \sum_{t=1}^{T} \epsilon_t(\theta_T) \tag{2}$$

assuming that each domain is sampled uniformly, where E_{D_t} is the expectation on domain D_t and $\ell(x, y)$ is the sample-based evaluation loss or error (e.g. pixel-wise accuracy or Intersection over Union for segmentation).

We focus on exemplar-free methods to solve the DIL problem, which refers to a setting where, during training and inference, the model does not rely on previously seen data from earlier sessions, a solution usually considered the most efficient strategy for continual learning but requires managing a memory buffer.

4. Problem Analysis

Implementation Details We use two distinct encoder–decoder architectures in our study: a modern Vision Transformer (ViT-Base) and a conventional U-Net.

The ViT-Base model serves as the primary backbone for our proposed method. It is trained on 512 × 512 RGB images with a patch size of 14 and a batch size of 8, using stochastic gradient descent (SGD). Decoder heads are trained for 30 epochs at a fixed learning rate of 10^{-3}. In the FFT configuration, the encoder is further fine-tuned for 50 epochs with a learning rate annealed from 10^{-5} to 10^{-6}. Early stopping is applied based on the validation set. The complete setup for sequential learning protocols is summarized in Table 1.

In contrast, the U-Net architecture is used exclusively in Section 4.1 to illustrate domain-incremental behavior under full fine-tuning (FFT). This classical model, with its residual and skip connections, allows us to analyze performance dynamics in a well-understood setting.

4.1. Oscillatory Performances in DIL

To investigate domain-incremental learning for semantic segmentation, we sequentially fully fine-tune (FFT) a conventional encoder–decoder U-Net along the domain sequence described above.

Table 1: Sequential-learning protocols and model sizes. "Memory" denotes any auxiliary buffer or statistical moments stored and maintained during training or inference. Parameter counts correspond to the models fitted across all ten domains. Here $S_z = \frac{1}{|N|}\sum_i z_i$ and $S_{zz} = \frac{1}{|N|}\sum_i z_i z_i^\top$ are, respectively, the first- and second-moment estimates of latent vectors over a set N of training samples.

Method	**Encoder**	**Decoder**	**Decoder type**	**Memory**	**#Params**
FFT	Tuned	Tuned	Discriminative	–	8.6e8
LP	Fixed	Tuned	Discriminative	–	1.5e5
FFT_1 + LP	Tuned on D70	Tuned	Discriminative	–	1.5e5
FFT_1 + LP (Mem.)	Tuned on D70	Tuned	Discriminative	20 % replay	1.5e5
DIPPCA	Fixed	–	Generative	S_z, S_{zz}	1.6e5
MoPPCA	Fixed	–	Generative	Domain + class S_z, S_{zz}	2.3e6

The overall performance curve shown in Fig. 3 is oscillatory rather than monotonically decreasing as new domains are added. Starting from an IoU of 0.79 on the first domain, performance declines steadily up to stage 4, rises abruptly at stage 5, falls again from stages 5 to 7, then rebounds between stages 7 and 8 before stabilizing from stages 8 to 10. The network's architecture integrates a sophisticated encoder–decoder backbone with multiple residual connections that facilitate information transmission across spatial scales. This intricate design complicates the isolation of the individual roles of these modules in the observed losses and recoveries during sequential fine-tuning.

In our baseline protocol, a `SegmenterLinear` network is sequentially fine-tuned on the predefined domains. In Fig. 3, we analyze the model's average performance across all domains seen up to each step, thereby describing its global behavior over time. In contrast, Fig. 4 provides a domain-level breakdown, revealing that the oscillations observed globally are in fact rooted in domain-specific performance variations.

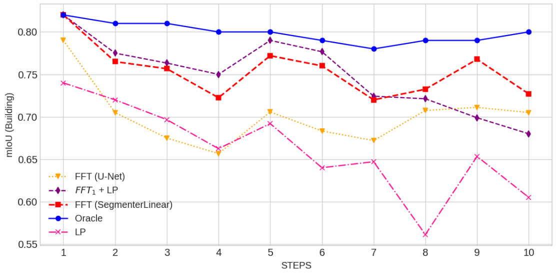


Figure 3: Oscillatory behaviour in domain-incremental learning. Each point on every curve reports the mean IoU for the *building* class, averaged over the i domains encountered up to step i. The curves therefore trace how performance fluctuates as new domains are sequentially added.

Fig. 3 shows an oscillatory FFT curve, with sharp performance drops at steps 2, 4, 7, and 10. The diagrams in Fig. 4 help attribute each of these drops to specific forgetting events. At step 4, for example, while adapting to `D23`, the model forgets the ini-

Table 2: Evaluation metrics. For each method we report the final mean-IoU after sequential training on the ten domains (mIoU_{fin}), the Oscillation Index (OI) and the Mean-Absolute Slope (MAS). Lower OI and MAS indicate smoother learning curves. Precise definitions of OI and MAS are given in the appendix.

Method	mIoU_{fin}	**OI**	**MAS**
FFT	0.73	0.60	0.022
LP	0.61	0.75	0.042
FFT_1 + LP	0.68	0.32	0.017
FFT_1 + LP (Memory)	0.72	0.60	0.015
DIPPCA (Q=100)	0.73	0.16	0.013
DIPPCA (Q=768)	0.76	0.20	0.013
MoPPCA	**0.76**	**0.16**	**0.008**

tial domain D70 (IoU 0.81 → 0.67, forgetting score $\Delta\text{IoU} = 0.15$). At step 7, training on D14 leads to performance drops on both D23 and D21 (IoU on D23: 0.73 → 0.58, $\Delta\text{IoU} = 0.15$; on D21: $\Delta\text{IoU} = 0.07$). At step 10, the largest degradation again occurs.

Because the decoder is merely a single linear layer, these oscillations are best explained by a drift in the feature space as the encoder is repeatedly fine-tuned. To test this hypothesis, the sequential linear probing scheme freezes the DINOv2 encoder and adjusts only the linear head. Under this setting the mean IoU still slides from 0.74 to 0.66 between steps 1 and 4, driven by the same D23-induced loss on D70. Performance rebounds after training on D35, yet falls sharply again at step 8 when adapting to D41, which provokes heavy forgetting on D21 ($\Delta\text{IoU} = 0.13$) and on D23 and D35 ($\Delta\text{IoU} = 0.17$ each). The final update on D78 repeats the pattern, degrading D21 ($\Delta\text{IoU} = 0.13$), D23 ($\Delta\text{IoU} = 0.17$) and D41 ($\Delta\text{IoU} = 0.18$). Freezing the encoder therefore lowers overall accuracy and amplifies the per-domain swings, with D23, D21 and D35 suffering the most.

Another stabilization attempt fully fine-tunes the network on the first domain D70 before restricting subsequent updates to the linear head. Adapting the features in this way lessens the cumulative performance loss and yields a higher mean IoU, yet the oscillatory behaviour endures. Pronounced forgetting still appears on D21 and D23 at steps 7 and 10, while a familiar rebound is observed at step 5. Thus, even after feature realignment, the sequential updates of a single linear classifier cannot completely suppress the recurrent drops and recoveries characteristic of this task.

Across the three incremental learning protocols—plain sequential fine-tuning, sequential linear probing with a frozen DINOv2 encoder, and one-off full fine-tuning on the first domain followed by linear probing—the same qualitative phenomenon persists: the global performance curve oscillates, with abrupt drops that coincide with sharp bursts of domain-specific forgetting and partial rebounds driven by positive transfer from certain domains.

Freezing the encoder does stabilize the representation in the strict sense that feature drift is eliminated; however, forgetting remains—and its amplitude even grows for some domains when freezing DINOv2 features. Re-centring the feature space by fine-tuning the whole network on the first domain ($\text{FFT}_1 + LP$) improves the average IoU and dampens forgetting, yet the oscillatory pattern survives. And we can attribute those oscillations to the decoder.

The oscillations metrics 2 highlight those observations. LP exhibits the poorest temporal behaviour: its MAS (0.042) indicates an average four-point swing in mIoU at every update, and its OI (0.75) confirms frequent, high-amplitude reversals—hence a highly erratic learning curve. Jointly tuning the encoder and decoder in FFT halves the step-to-step variability (MAS = 0.022) and lowers the oscillation index to 0.60, yet pronounced peaks and troughs persist. Pre-training the encoder once on the first domain further stabilises the trajectory: FFT_1+LP reduces MAS to 0.017 and OI to 0.32, cutting both the magnitude and the frequency of performance reversals by more than 50 % relative to LP.

4.2. Explaining Oscillations

For all three training regimes, forgetting is not uniform but is more pronounced for particular domains and at specific steps, hinting at outlier domains whose feature distributions deviate from the rest of the sequence.

The most obvious cases are D21 and D23. Both exhibit abrupt mIoU drops—D21 at steps 7, 8, and 10, D23 at the same points—and, in addition, updating on D23 triggers forgetting on the initial domain D70.

We attribute these oscillations to inter-domain interference: when an incoming domain is far from certain historical domains in feature space, the parameter shift that benefits the newcomer harms those earlier domains. We quantify this interference with the 2-Wasserstein distance $d_2(D_i, D_j)$ (Mallasto et al., 2022) and we make the assumption that the larger $d_2(D_i, D_j)$, the larger the ensuing mIoU drop on D_j after training on D_i.

We quantified the discrepancy between domains by measuring pairwise 2-Wasserstein distances in the DINOv2 feature space learned after full training on the reference domain D070. The distance matrix (5) highlights several dissimilar pairs whose values exceed $d^2 > 800$: (D021,D014), (D021,D049), (D021,D078) as well as (D023,D014), (D023,D049) and (D023,D078). The forgetting matrix for the FFT_1+LP configuration (4) shows the largest negative jumps exactly for those pairs: ΔmIoU $= -0.13$ on D023 and -0.7 on D021 when adapting to D014; -0.24 (D023) and -0.08 (D021) when adapting to D078; and so forth for D049. We then correlate those distances with ΔIoU (Figure 6): the trend is clear and positive, larger 2-Wasserstein distances entails larger forgetting, a few outliers remain, however: e.g. a large distance $W_2(D070, D021) > 500$ produces almost no forgetting, and $W_2(D021, D023) < 500$ produces rather high forgetting implying that another factor is at play.

5. Problem solution

In the previous section, we identified the critical phenomenon to control in DIL as the randomness of the geometric distribution of domains, which generates performance oscillations rather than the monotonic performance decrease — i.e., catastrophic forgetting — observed in CIL. Given this finding, we will now examine ways to counteract this behavior. First, we discuss the nature of neural architectures used for semantic segmentation and propose two schemes that rely on a

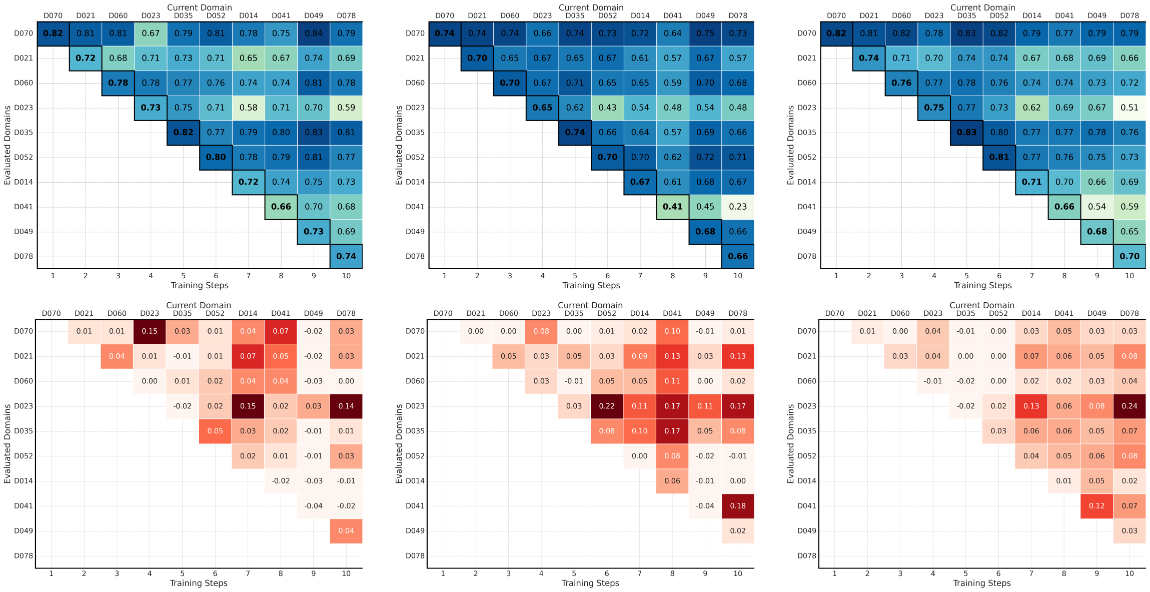


Figure 4: Continual-learning performance matrices for the three training protocols: (a) FFT (left), (b) LP (centre) and (c) FFT_1+LP (right). **Top row**—per-step, per-domain IoU for the *building* class (darker = higher). **Bottom row**—ΔIoU relative to the diagonal (initial score); negative values indicate an improvement rather than forgetting.

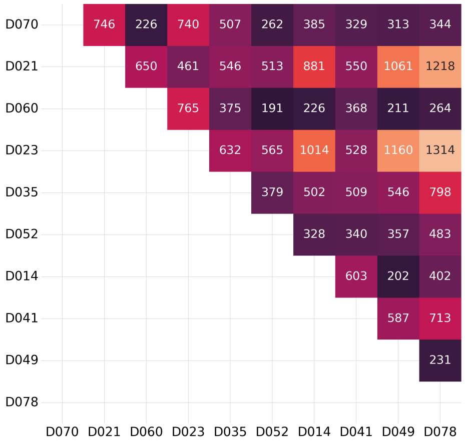
rich fixed encoder and a light decoder-centric adaptation.

5.1. Architecture for segmentation DIL

Encoder-decoder architectures for segmentation In deep learning, a standard strategy is to use a pretrained model to encode an image along with a decoder that solves a specific task. The encoder is typically fine-tuned to the target data, while the decoder is fully learned. The differences between neural architectures depend roughly on the relationship between the encoder and decoder, as well as their nature.

A large variety of architectures have been adapted to solve semantic segmentation, implementing several trade-offs in complexity between the encoder and the decoder (Huang et al., 2024a). In classical architectures, such as U-Net (Ronneberger et al., 2015), the encoder concentrates useful information in a tensor with low spatial resolution and many channels. This requires a complex upsampling decoder to produce the semantic map. The adaptation effort to a new domain is shared equally between the encoder and decoder during learning. Many recent transformer-based architectures, such as Mask2Former (Cheng et al., 2021), also require a complex decoder that combines a query-based transformer and a multi-scale, pixel-based upsampling pyramid.

When rich decoder architectures are used for continual learning, the plasticity required to adapt to a new domain or task is distributed between two complex functional structures. This makes controlling their sta-



	D070	D021	D060	D023	D035	D052	D014	D041	D049	D078
D070		746	226	740	507	262	385	329	313	344
D021			650	461	546	513	881	550	1061	1218
D060				765	375	191	226	368	211	264
D023					632	565	1014	528	1160	1314
D035						379	502	509	546	798
D052							328	340	357	483
D014								603	202	402
D041									587	713
D049										231
D078										

Figure 5: 2-Wasserstein distance matrix between the feature distributions of the ten domains. Entry (i, j) reports the distance between domains D_i and D_j, computed in the fixed feature space obtained after fully fine-tuning on the first domain (D_{70}).

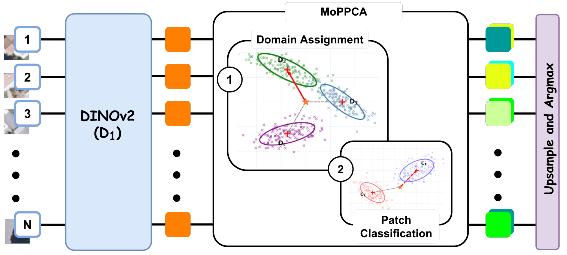


Figure 7: Pipeline of the latent–domain PPCA decoder (**MoPPCA**). **(1) Feature extraction** — input patches are embedded by a frozen DINOv2 backbone. **(2) Domain assignment** — in the latent space, a Probabilistic PCA mixture models each *domain* as a Gaussian, every patch is assigned to the nearest domain component via the Mahalanobis score $(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)$. **(3) Patch classification** — within the selected domain, another PPCA mixture models the class-specific clusters and assigns the patch to a class with the same Mahalanobis criterion. **(4) Upsampling** — the per-patch logits are upsampled to the original image resolution and an ARGMAX yields the final segmentation map. Colours illustrate domain and class components; red crosses denote PPCA means and ellipses the projected covariances.

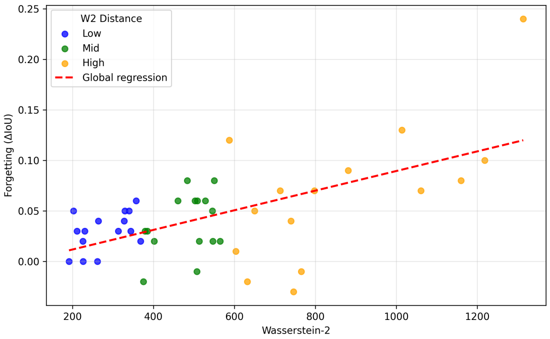


Figure 6: Correlation between the 2-Wasserstein distance and forgetting (ΔIoU). Pairwise distances are binned into three categories—Low, Mid and High. A global least-squares regression line (dashed red) highlights the overall trend.

bility potentially difficult because both structures contribute to the final output. The role of each structure in performance degradation due to the incremental setting is unclear.

A patch-level decoder-centric architecture To avoid dealing with two complex structures for controlling performance degradation due to DIL, we propose an architec-

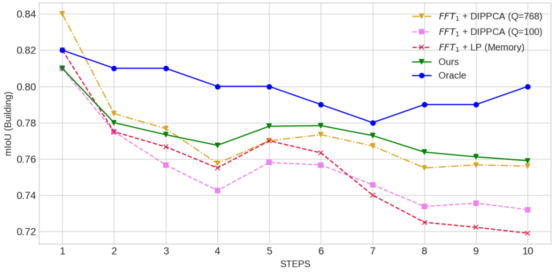


Figure 8: Mean IoU on the *building* class across the ten domain-incremental steps. MoPPCA stays closest to the joint-training oracle and shows fewer oscillations than DIPPCA and replay baselines.

ture that relies on a rich image encoder and a simple class prediction decoder. This is the architecture already tested in 4.1. The goal is to minimize modifications to the encoder when a new domain arrives — potentially even freezing it — while enabling a more significant adaptation of the decoder. This scheme implies that we have an encoder that can produce high-quality features with a good invariance/discrimination trade-off for a wide variety of images. The advent of self-supervised techniques has made this possible (Jing and Tian, 2021).

Of the possible encoder candidates, DINOv2 (Oquab et al., 2023) appears to be the most suitable choice. DINOv2 relies on a ViT architecture that provides patch-level embeddings for each input image and has been proven effective for dense recognition tasks.

We propose feeding the DINOv2 features to a simple decoder that computes patch-level class logits, followed by bilinear upsampling of these logits to match the original image size. This is a global architecture similar to the linear variant of Segmenter (Strudel et al., 2021). The key difference lies in how the logits are computed. Our approach is based on probabilistic principal component analysis (PPCA) (Tipping and Bishop, 1999) to compute the class conditional logits. We use such a generative classifier because it allows for a simple incremental updating strategy based on running averages of the first and second statistical moments of class-conditional distributions over each session. This property has motivated other strategies proposed for CIL settings, such as those in (Panos et al., 2023; Wang and Barbu, 2022), and DIL settings, such as those in (Boum et al., 2024), but for classification.

To adapt to a segmentation problem, computations are performed at the patch level of the DINOv2 ViT architecture. Given an input image, the encoder produces a 3D tensor. The first dimensions of this tensor are the numbers of the patch indices, and the last dimension is the feature space associated with each patch. The prediction is obtained by computing the argmax of the class-conditional likelihood after applying PPCA models and bilinear upsampling the logits.

Class conditional PPCA models are computed by considering the distribution of patch features from each image. The dataset used to estimate the first and second moments of the model is a collection of patch features gathered from all images in the current domain. Each patch (14 x 14 pixels in DINOv2) is annotated with the most prevalent class from the ground truth; patches with ambiguous classes are discarded. Working at the patch level enables the manipulation of populations of moderate size to estimate the model parameters and frames segmentation as a classification problem with independent, patch-based feature samples. The computation of the model parameters follows the scheme proposed in (DIPPCA) (Boum et al., 2024), which computes the Mahalanobis distance in a small projection space obtained by singular value decomposition.

Latent domain conditioning In the DIL approach described above, stability results from keeping features fixed, and plasticity is satisfied by updating the parameters of a single generative PPCA model per class. However, the unimodal Gaussian distribution assumption underlying this model may not adequately fit the global distribution of all the domains observed at this point. Indeed, as shown in 4.1, one possible explanation for the oscillatory behavior of performance is the presence of outlier domains in feature space.

A simple solution for handling erratic domain geometry is to condition the classification on the predicted domain from which the data was sampled. This step is analogous to the task identity inference component defined in (Wang et al., 2025). First, the most likely task or domain is assigned, and then the class is inferred using domain-dependent parameters. Domain assignment can be solved using maximum likelihood with a PPCA model in the patch feature space. This time, the model is class agnostic, and we have a single PPCA model per domain. The combination of domain assignment and domain-conditional class prediction constitutes our method : Mixture of PPCA (MoPPCA). We set the latent dimension to Q=100 as a compromise between representation power and computational cost.

5.2. Validation

The FFT_1+LP (Memory) curve still displays a saw-tooth pattern: mIoU falls from steps 2 to 4, rebounds at step 5, then declines smoothly to the end. Table 2 shows that the replay buffer dampens only the small variations—the mean-absolute slope drops from 0.017 to 0.015—but leaves the large peaks and troughs intact (OI stays high at 0.60). Although the extra exemplars lift the final mIoU from 0.68 to 0.72, the method remains vulnerable to strong inter-domain interference. The generative decoder in DIPPCA (Q = 768) further reduces oscillations: its OI falls to 0.20 and MAS to 0.013. The curve still shows marked dips at steps 2, 4 and 8, yet it stays above the replay baseline except at steps 4–5, where the two traces intersect before FFT_1+LP (Memory) resumes its slow slide while DIPPCA stabilises.

The improvement suggests that modelling feature variance preserves global structure better, but the gain is limited by parameter budget and by outlier domains, whose arrival still triggers noticeable drops. Our method, MoPPCA(Section 5), eliminates almost all residual oscillations. As Figure 8 illustrates, the curve tracks the oracle closer than the other methods, and its critical dips are visibly attenuated. With a latent dimension of only $Q = 100$, MoPPCA already outperforms DIPPCA, achieving the joint-highest final mIoU (0.76) while posting the best stability scores in Table 2 (OI = 0.16, MAS = 0.008).

Because its parameters are estimated once and reused, no additional fine-tuning is required, and the learning trajectory remains both accurate and smooth throughout the ten-domain sequence. Quantitatively, the three generative variants exhibit the following profile (see Table 2). DIPPCA(Q=100) yields a final mIoU of 0.73 with an OI of 0.16 and a MAS of 0.013. Increasing the latent rank to DIPPCA (Q=768) lifts the mIoU to 0.76 but also raises the oscillation index to 0.20, while the MAS remains at 0.013.

By contrast, MoPPCA (Q=100) attains the same top-line accuracy as DIPPCA (Q=768) (0.76) yet restores the lower OI of 0.16 and cuts the step-to-step variability to MAS = 0.008 roughly a 40% reduction relative to both DIPPCA settings. Thus, MoPPCA matches the best accuracy obtained

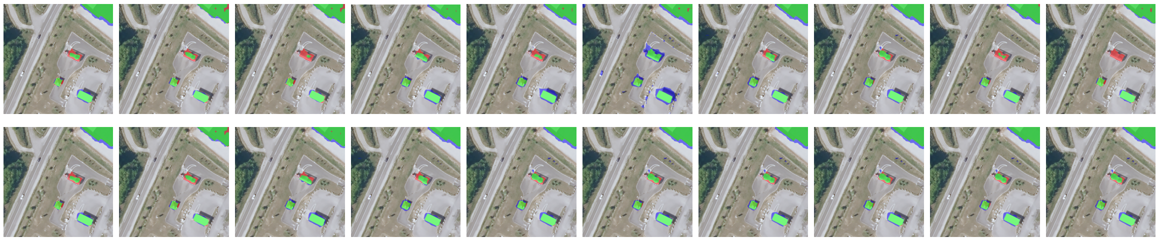


Figure 9: Qualitative evaluation over ten incremental steps (left-to-right) on the FLAIR#1 sequence.**Top**: baseline FFT. **Bottom**: proposed method MoPPCA. The baseline exhibits pronounced oscillations, evidenced by a whole building that goes completely undetected at one step then re-appears in the following step. MoPPCA maintains every building across all steps, only minor boundary variations remain, confirming its superior stability over the sequential steps.

with a high-rank DIPPCA while delivering the smoothest learning trajectory of all three methods.

9 corroborates the quantitative results. In the baseline FFT (top row) oscillations translates to buildings vanish entirely at successive incremental steps before re-appearing at later steps. By contrast, the proposed MoPPCA decoder (bottom row) preserves all buildings throughout the sequence: although contour delineation fluctuates slightly from step to step, no building is completely lost, indicating markedly greater stability.

6. Discussion

In this work, we have established a protocol for leveraging foundation model's features for domain incremental semantic segmentation of VHR remote sensing imagery (FLAIR#1). Using this protocol we show that the overall performance does not decay smoothly but oscillates and we traced those oscillations to outlier domains whose feature distributions diverge notably from the rest of the data stream. To remedy those oscillations we developed a latent domain–conditioned architecture that first infers the active domain with a class-agnostic PPCA in the feature space and then carries out domain-specific classification with a second PPCA.

By coupling automatic task identification with a lightweight, domain-aware decoder, the approach removes negative backward transfer and consistently exceeds the DIPPCA baseline, even when the latter is trained to capture the full variance of the data. The present study leaves two questions open. First, we have not yet characterised how the latent dimension Q and the number of class-agnostic PPCAs influence overall accuracy. Second, the assignment statistics of the latent domains themselves—particularly the case of patches belonging to outlier domains—remain unexplored. Investigating these aspects will refine our understanding of domain structure in incremental remote-sensing scenarios and may suggest further improvements to the model.

In addition, although the current experiments are restricted to binary building-background segmentation, the same methodology could be applied in a multiclass setting on the original FLAIR#1 annotation scheme and subsequently extended to other high-resolution Earth Observation data sets.

Acknowledgments

Acknowledgements The research is supported by a PhD scolarship from ONERA and CNES for M-AB, and by the STORE project, funded by the European Defence Fund (EDF) under grant agreement EDF-2022-101121405-STORE for SH and PF.

References

Motasem Alfarra, Zhipeng Cai, Adel Bibi, Bernard Ghanem, and Matthias Müller. Simcs: Simulation for domain incremental online continual segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10795–10803, 2024. [2, 3]

Marie-Ange Boum, Stéphane Herbin, Pierre Fournier, and Pierre Lassalle. Continual learning in remote sensing: Leveraging foundation models and generative classifiers to mitigate forgetting. In *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 8535–8540. IEEE, 2024. [4, 11]

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021. [2]

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. [2]

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. *arXiv:2112.01527 [cs]*, December 2021. [9]

Laura N Driscoll, Lea Duncker, and Christopher D Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609, 2022. [2]

Prachi Garg, Rohit Saluja, Vineeth N Balasubramanian, Chetan Arora, Anbumani Subramanian, and CV Jawahar. Multi-domain incremental learning for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 761–771, 2022. [1]

Anatol Garioud, Stéphane Peillet, Eva Bookjans, Sébastien Giordano, and Boris Wattrelos. Flair# 1: semantic segmentation and domain adaptation dataset. *arXiv preprint arXiv:2211.12979*, 2022. [4]

Liwei Huang, Bitao Jiang, Shouye Lv, Yanbo Liu, and Ying Fu. Deep-Learning-Based Semantic Segmentation of Remote Sensing Images: A Survey. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:8370–8396, 2024a. ISSN 2151-1535. doi: 10.1109/JSTARS.2023.3335891. [9]

Wubiao Huang, Mingtao Ding, and Fei Deng. Domain-incremental learning for remote sensing semantic segmentation with multifeature constraints in graph space. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024b. doi: 10.1109/TGRS.2024.3481875. [2, 3]

Wubiao Huang, Mingtao Ding, and Fei Deng. Domain incremental learning for remote sensing semantic segmentation with multifeature constraints in graph space. *IEEE Transactions on Geoscience and Remote Sensing*, 2024c. [3]

Longlong Jing and Yingli Tian. Self-Supervised Visual Feature Learning With

Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, 2021. [11]

Tobias Kalb, Masoud Roschani, Miriam Ruf, and Jürgen Beyerer. Continual learning for class-and domain-incremental semantic segmentation. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 1345–1351. IEEE, 2021. [1]

Kuan-Ying Lee, Yuanyi Zhong, and Yu-Xiong Wang. Do Pre-Trained Models Benefit Equally in Continual Learning? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6485–6493, 2023. [3]

Yazhou Liu, Haoqi Chen, Pongsak Lasang, and Zheng Wu. Domain-incremental semantic segmentation for traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 2025. [2, 3]

Anton Mallasto, Augusto Gerolin, and Hà Quang Minh. Entropy-regularized 2-wasserstein distance between gaussian measures. *Information Geometry*, 5(1): 289–323, 2022. [8]

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random Projections and Pre-trained Models for Continual Learning. *Advances in Neural Information Processing Systems*, 36:12022–12053, December 2023. [3]

M Jehanzeb Mirza, Marc Masana, Horst Possegger, and Horst Bischof. An efficient domain-incremental learning approach to drive in all weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3001–3011, 2022. [1, 4]

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. [11]

Oleksiy Ostapenko, Timothee Lesort, Pau Rodriguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual Learning with Foundation Models: An Empirical Study of Latent Replay. In *Proceedings of The 1st Conference on Lifelong Learning Agents*, pages 60–91. PMLR, November 2022. [3]

Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E. Turner. First Session Adaptation: A Strong Replay-Free Baseline for Class-Incremental Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18820–18830, 2023. [3, 11]

Liangzu Peng, Juan Elenter, Joshua Agterberg, Alejandro Ribeiro, and René Vidal. TSVD: Bridging Theory and Practice in Continual Learning with Pre-trained Models, March 2025. [3]

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment Matching for Multi-Source Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019. [3]

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Pro-*

ceedings, Part II 16, pages 524–540. Springer, 2020. [2, 3]

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

Xue Rui, Ziqiang Li, Yang Cao, Ziyang Li, and Weiguo Song. Dilrs: Domain-incremental learning for semantic segmentation in multi-source remote sensing data. *Remote Sensing*, 15(10):2541, 2023. [2, 3]

Antoine Saporta, Arthur Douillard, Tuan-Hung Vu, Patrick Pérez, and Matthieu Cord. Multi-Head Distillation for Continual Unsupervised Domain Adaptation in Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3751–3760, 2022. [2, 3]

Haizhou Shi and Hao Wang. A unified approach to domain incremental learning with memory: Theory and algorithm. *arXiv preprint arXiv:2310.12244*, 2023. [1]

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. [11]

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999. [11]

Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, pages 1–13, 2022. [1]

Boshi Wang and Adrian Barbu. Scalable learning with incremental probabilistic pca. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5615–5622. IEEE, 2022. [3, 11]

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A Comprehensive Survey of Continual Learning: Theory, Method and Application, January 2023. [1]

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Hang Su, and Jun Zhu. Hide-pet: continual learning via hierarchical decomposition of parameter-efficient tuning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. [12]

Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022. [1]

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting Class-Incremental Learning with Pre-Trained Models: Generalizability and Adaptivity are All You Need. *International Journal of Computer Vision*, 133 (3):1012–1032, March 2025. ISSN 1573-1405. doi: 10.1007/s11263-024-02218-0. [3]

Tianfei Zhou, Fei Zhang, Boyu Chang, Wenguan Wang, Ye Yuan, Ender Konukoglu, and Daniel Cremers. Image Segmentation in Foundation Model Era: A Survey, August 2024. [2]

Appendix A. First Appendix

A.1. Stability metrics.

Let m_i denote the mean IoU observed after step i ($i = 1, \dots, T$). We report two complementary, metrics for temporal stability.

(i) Oscillation Index (OI). We first remove the global trend by subtraction and quantify the average amplitude of each reversal of slope:

$$\mathrm{OI} = \frac{1}{T-2} \frac{\sum_{i=3}^{T} \big|(m_i - m_{i-1}) - (m_{i-1} - m_{i-2})\big|}{m_{\max} - m_{\min}}.$$

OI equals 0 for a perfectly monotone curve and increases with both the frequency and the height of peaks/troughs; a lower value thus indicates smoother behaviour.

(ii) Mean Absolute Slope (MAS). While OI focuses on the high-frequency component, MAS captures the average step-to-step variability:

$$\mathrm{MAS} = \frac{1}{T-1} \sum_{i=2}^{T} |m_i - m_{i-1}|.$$

MAS is insensitive to the direction of the drift and isolates the magnitude of local changes. In both metrics, smaller numbers translate into greater stability.

A.2. MoPPCA : Domain Scale Analysis

The per-domain IoU matrix reveals a stable performance across training steps, with values remaining high and consistent across previously seen domains. No severe forgetting is observed: once a domain is learned, its IoU remains largely unchanged in subsequent steps.

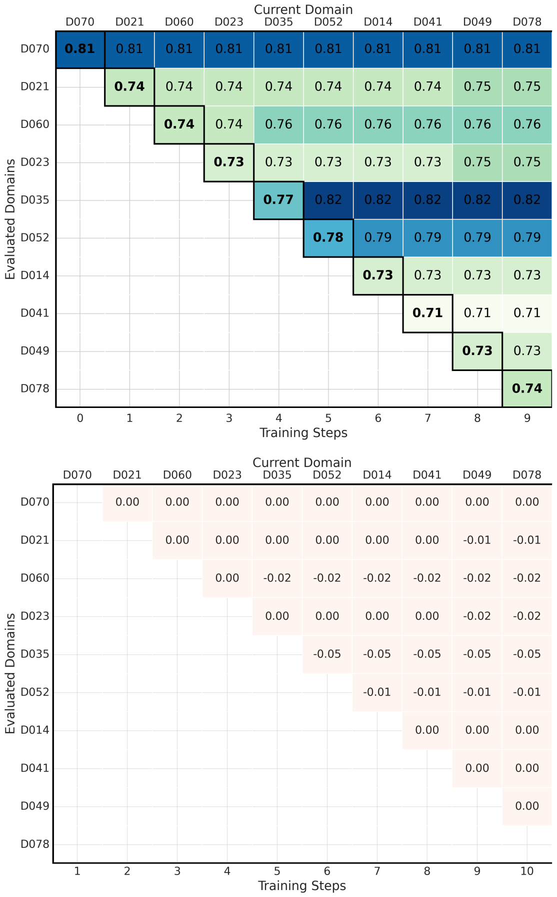


Figure 10: Continual-learning performance matrices for the MoPPCA training protocol. **Top row**—per-step, per-domain IoU for the *building* class (darker = higher). **Bottom row**—ΔIoU relative to the diagonal (initial score); negative values indicate an improvement rather than forgetting.

Some domains (e.g., `D035`, `D052`) even exhibit slight performance gains over time, suggesting mild positive forward transfer. For example, `D035` maintains an IoU of 0.82 across steps, and `D052` improves slightly (0.78 → 0.79). Similarly, early domains such as `D070` and `D021` retain their scores (0.81 and 0.75, respectively), indicating good knowledge retention.