

On the Road to Online Adaptation for Semantic Image Segmentation

Riccardo Volpi
NAVER LABS Europe*

Pau De Jorge
NAVER LABS Europe
Oxford University

Diane Larlus
NAVER LABS Europe

Gabriela Csurka
NAVER LABS Europe

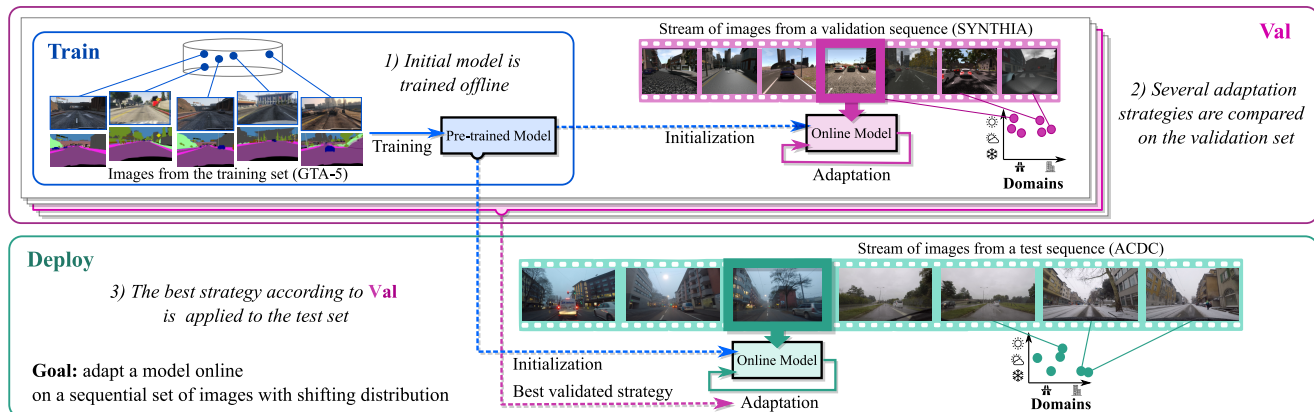


Figure 1. **The proposed OASIS benchmark.** We formalize a domain adaptation task which requires online, unsupervised adaptation of semantic segmentation models and propose a novel benchmark to tackle it. It is composed of three steps. **Train:** A model is trained offline on simulated data (top-left); **Val:** Several adaptation strategies are validated on simulated data organized in sequentially shifting domains (e.g., sunny-to-rainy, highway-to-city), to mimic deploy (top-right). **Deploy:** The best validated strategy is applied to the test set (bottom).

Abstract

We propose a new problem formulation and a corresponding evaluation framework to advance research on unsupervised domain adaptation for semantic image segmentation. The overall goal is fostering the development of adaptive learning systems that will continuously learn, without supervision, in ever-changing environments. Typical protocols that study adaptation algorithms for segmentation models are limited to few domains, adaptation happens offline, and human intervention is generally required, at least to annotate data for hyper-parameter tuning. We argue that such constraints are incompatible with algorithms that can continuously adapt to different real-world situations. To address this, we propose a protocol where models need to learn online, from sequences of temporally correlated images, requiring continuous, frame-by-frame adaptation. We accompany this new protocol with a variety of baselines to tackle the proposed formulation, as well as an extensive analysis of their behaviors, which can serve as a starting point for future research.

*<https://europe.naverlabs.com>

1. Introduction

Machine learning systems will often face unfamiliar conditions when deployed in the real world. A self-driving car will encounter a new urban environment or unexpected weather; a domestic robot will be deployed in a new house.

In this regard, semantic image segmentation, a key task for both examples above, is an important case study. Due to the extremely high cost of manually annotating segmentation masks, it is very common to train segmentation models on synthetic data [62, 65]; these models can then be adapted to real environments using domain adaptation techniques [16, 33, 34, 81, 86, 97]. Adaptation methods typically run offline, require multiple epochs, and generally rely on an annotated validation set from the target domain. Such assumptions are reasonable for certain scenarios; for instance, consider the problem of adapting a segmentation model for MRI from simulated data to real-scanner samples: adaptation can happen offline, and one can ask some experts to provide at least a few annotated target samples for hyper-parameter selection.

Unfortunately, such assumptions do not hold for all scenarios. In our initial example of a self-driving car equipped with a semantic segmentation module, once the car is on the

road, samples arrive *sequentially, unlabeled*, and the surrounding environment *continually changes*. The segmentation model needs to jointly produce a prediction and continuously adapt as individual images arrive. This is one among many such scenarios that require unsupervised adaptation to happen online, continuously and without the possibility of extensive hyper-parameter tuning on new domains. We argue that there is a strong need for an appropriate formulation, ad hoc evaluation protocols and metrics specifically designed for this challenging adaptation problem.

This paper aims at filling this gap. We introduce a constrained problem formulation, which requires *online and unsupervised adaptation of semantic segmentation models*. The model is expected to adapt to new environments by processing sequences of temporally correlated frames, from ever changing domains. Therefore, this formulation does not allow processing large amounts of target samples offline nor performing extensive hyper-parameter search.

Concretely, our **first contribution** is proposing a three-step **train-val-deploy** evaluation protocol that emulates as closely as possible the problem of adapting semantic segmentation models as the input distribution shifts. By design, cross-validating on “deploy” samples is not possible. The three steps are (i) Pre-training a segmentation model offline on simulated data; (ii) Validating the algorithms at hand, *i.e.* carefully choosing all the hyper-parameters – again on simulated data, yet mimicking the sequential nature of the real world; (iii) Testing the best adaptation algorithm – according to the previous validation step – on realistic, sequential conditions. Those steps are illustrated in Fig. 1. We formalize this new problem and devise a proper benchmark for **Online Adaptation for Semantic Image Segmentation, the OASIS benchmark**.

Endowed with this framework, our **second contribution** is tailoring and benchmarking a set of adaptation techniques from related fields for our problem (see Fig. 2). We also propose different learning strategies to accommodate the challenges that our protocol brings, reporting thorough comparisons among the different methods, and share our main findings, which can serve as a basis for future research on this task. As a **third contribution**, we assess the impact of *catastrophic forgetting* when continuously adapting without supervision and draft a family of methods based on a *reset mechanism* to mitigate this issue.¹

2. Related work

This paper studies domain adaptation for semantic segmentation and, hence, relates to both fields. Our scenario also has links with the domain generalization and the continual learning literature. Fig. 2 illustrates the connections with different research topics, which we detail below.

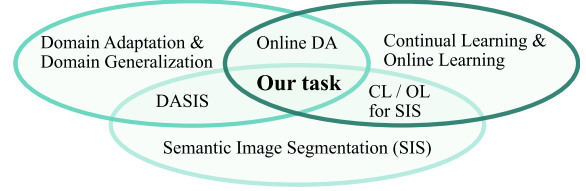


Figure 2. **Our task in the literature.** We focus on the task of online unsupervised domain adaptation for semantic image segmentation, which lies at the intersection of several fields.

Semantic image segmentation (SIS) consists in predicting the class label of every image pixel. Modern methods typically use fully convolutional networks [50] or recurrent neural networks [7, 47]. These models can be further combined with Conditional Random Fields (CRFs) [9, 48, 71, 104], multi-resolution architectures [29, 32, 49, 103], or attention mechanisms [14, 25, 43]. Encoder-decoder architectures can compress the image into a latent space that captures the underlying semantic information and then decode the latent representation into final predictions [5, 56, 64]. A successful family of SIS methods is DeepLab [12, 13, 15, 48], which combines dilated convolutions for resolution, Atrous Spatial Pyramid Pooling to capture the context at multiple scales, and CRFs to refine predictions. Lately, it has been shown that transformer architectures are very effective for SIS [74, 95, 105].

Domain adaptation for SIS (DASIS) aims at adapting a segmentation model trained on one or several source domains to one or more target domains [23, 78]. Deep DA [21, 89] is particularly relevant for SIS, as the recent approaches are data-hungry and pixel-level annotations are extremely tedious and time consuming to acquire [20]. Since game rendering engines can generate photo-realistic virtual worlds with arbitrary variations in weather, environment and lighting conditions, synthetic fully-annotated datasets [62, 65] have become a standard source to train SIS models, which are then adapted to real datasets. Such deep DASIS methods, according to [23, 78], can be grouped into adversarial feature alignment [34, 36, 100], output-level adaptation [57, 81, 86] or image-level style transfer between domains jointly learned with the segmentation model [10, 16, 33, 54, 69, 92, 97]. Self-training and target entropy minimization are also very popular techniques used to improve the adaptation [17, 24, 46, 86, 107, 107].

Differently from these standard DASIS methods, we are interested in algorithms that are able to adapt *online*. These methods cannot be straightforwardly applied in our settings since we never assume the availability of both source and target empirical distributions.

Domain generalization (DG) models learn on one or more source domains with the goal of generalizing to new, unseen domains. There is a large diversity of DG methods [88, 106], some of them effectively applied to SIS [30, 39, 84, 85, 99].

¹Code available at <https://github.com/naver/oasis>

While DG does not take into account model adaptability per-se, it is relevant in our context because, as we will show, starting with stronger representations leads to more effective adaptation results. Hence, DG can play a crucial role for model pre-training to foster adaptability.

Continual learning (CL) consists in gradually enriching the model as new domains/tasks are encountered while trying not to forget about previous ones [18, 58]. In this paper, we assume that the SIS task – including the set of semantic labels – remains unchanged, but the data distribution varies over time (e.g. from *urban environment* to *highway*, or from *sunny* to *rainy*). Only few methods have addressed this scenario, e.g. by gradually shifting target domains while replaying old memories to avoid forgetting [6], incremental adversarial training and periodic offline retraining with previously seen samples [94], incrementally adapting the model to changing environments via meta-learning [83] and style transfer [93], or fine-tuning the domain adapters on the target in a self-supervised fashion [61]. Yet, while these methods address adaptation to multiple domains sequentially, none of them considers an online scenario where the adaptation needs to be performed frame by frame.

Online domain adaptation (ODA), related to online learning (OL) [8], aims at adapting a model when confronted to a stream of data, potentially dealing with one sample at a time (preferably in real time). This is in contrast with incremental DA [6, 83, 93], where the model is trained to handle new domains *offline*. A few ODA methods have been proposed in the past, in particular to adjust pre-trained detectors for videos [90, 96] or object tracking [26, 27, 72]. Online updates of batch normalization (BN) [38] statistics have been shown to help when applied to robotic kitting [53]. More recently, online adaptation has been more extensively studied with respect to depth estimation, both in the stereo [79, 80, 101] and in the monocular [41, 102] cases.

Our formulation requires SIS models to be adapted both in a sequential and online manner. To the best of our knowledge, there is only one such previous ODA approach for SIS, OnAVOS [82]; yet, it only considers two classes (moving foreground/background) and assumes the first frame to be fully labeled. In this regard, one of our contributions is to provide a general formulation and companion benchmark to study ODA for SIS in a realistic framework.

Recently, **test-time adaptation/training** techniques have been proposed that can be applied to ODA. Sun et al. [75] optimize the model parameters at test time using self-supervised learning. Wang et al. [87] show that BN parameters can be optimized by minimizing the entropy of the model’s predictions for the target samples. Schneider et al. [70] show that mixing BN statistics of pre-trained models and those gathered from the target samples can significantly boost out-of-domain performance. In this work, we study different flavors of the latter two approaches, mixing

them with different strategies to propose variants tailored for our problem formulation.

3. The OASIS benchmark

This paper formalizes the problem of semantic image segmentation in a constantly changing environment, such as the one encountered by self-driving systems – where adaptation needs to happen image by image in a sequential fashion. With these problems, concrete research questions arise: (1) *How well can a model perform without adaptation when exposed to different conditions?* Tackling this question will help designing pre-training strategies that adaptation methods should start from. (2) Assuming that online adaptation at image-by-image level is effective, *should we adapt continuously*, or should the adaptation strategy for each image always start from a pre-trained model? (3) In the case of continual adaptation, *will the model suffer from catastrophic forgetting?* (4) If it does, *how to overcome it?*

After formalizing the task in Sec. 3.1, in Sec. 3.2 we describe **the OASIS benchmark** with its evaluation protocol, designed to answer the above questions.

3.1. Task definition

Let $\mathcal{X}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ be a set of annotated source images, with $x_i^s \in \mathbb{R}^{H \times W \times 3}$ and $y_i^s \in \mathbb{R}^{H \times W \times C}$. Images are of size (H, W) and their pixels are assigned to one of the C semantic classes. Let $\mathcal{X}^t = (x_i^t)_{i=1}^{N^t}$ be a sequence of non-annotated target samples, with $x_i^t \in \mathbb{R}^{H \times W \times 3}$.

We define the problem of **continual unsupervised domain adaptation for semantic image segmentation** starting from the following configuration. Let \mathcal{M}_{θ_0} be a parametric model for SIS that is learned on \mathcal{X}^s . This model has never seen any target images, not even unlabelled ones. Once the adaptation process starts, the model receives unlabelled target samples $x_i^t \sim \mathcal{X}^t$, one at a time. A given adaptation routine, provided with the current² sample x_i^t and the current model $\mathcal{M}_{\theta_{i-1}}$, outputs the adapted model \mathcal{M}_{θ_i} . The latter is used to compute the final predictions \hat{y}_i^t .

Given a proper metric – e.g., mean-Intersection-over-Union (mIoU) – one can compute the performance score on the whole sequence \mathcal{X}^t by averaging single samples’ scores, computed by comparing \hat{y}_i^t with the ground truth.

3.2. Protocol design

In order to answer the crucial questions that arise when developing adaptive methods in the real world, we need a carefully designed evaluation protocol, inherently different from the ones typically considered for domain adaptation.

Our core contribution is **the OASIS benchmark**, whose aim is filling this gap. Its protocol, also depicted in Fig. 1,

²One may also consider using a buffer of recent samples or an episodic memory where some previously encountered samples are stored – but we do not consider this scenario in this work.

comprises three steps: **train**, **val**, and **deploy**, which respectively correspond to the *pre-training*, *validation* and *real-world test* components of our benchmark. For those, we build on existing datasets. In the following, we provide the rationale behind each of them and a summary of their implementation (more details in the Supplementary Sec. A).

3.2.1 Pre-training (**train**)

Rationale. Regardless of whether the model should be adapted to new environments or not, it is crucial for a visual model deployed into the real world to be as robust as possible. The key ingredients to achieve robustness are generally a strong architecture and the availability of a large amount of labelled training data. Domain randomization [77, 99] and domain generalization [88, 106] can be used to boost robustness as well. Our protocol allows to explore SIS models trained *offline*, on large amounts of annotated data, and with arbitrary training strategies; the goal here is providing a strong starting point for further deployment.

Experimental details. Due to the high cost of human-annotations for SIS, it is common to rely on images rendered via game engines – as their segmentation annotation comes for free. Therefore, following previous work on DA-SIS [34, 81], we pre-train our SIS models on GTA-5 [62], a dataset of 25*k* images that serves as our source domain.

3.2.2 Validating (**val**)

Rationale. In order to predict how the model will behave in real-world environments, it is important to evaluate different adaptation strategies in conditions that mimic the real world as closely as possible. Concretely, provided with a satisfactory initial model resulting from the previous phase, we need to validate its out-of-domain performance, as well as the behavior of selected adaptation strategies – in order to choose the most promising methods and their hyper-parameters. To this end, we need to access or generate a large amount of annotated *sequences* of temporally correlated samples³, with occasional sub-domain shifts (weather condition and/or urban environment change). This validation step outputs the best model and adaptation strategy, together with their tuned hyper-parameters.

Experimental details. We use SYNTHIA [65] to validate pre-training and adaptation strategies. It comprises simulated sequences from different urban environments (we use *Highway*, *New York-like City*, *European Town*) and different weather/daylight/time-of-the-year conditions (we use *Summer*, *Spring*, *Fall*, *Winter*, *Dawn*, *Sunset*, *Night*, *Rain*, *Fog*). We design episodes where the model needs to process samples from ever changing sequences. We propose a total of 8

sequences, each built with 5 random sub-sequences of 300 frames each, where both environments and conditions are randomly picked, resulting in several domain shifts.

3.2.3 Real-world testing (**deploy**)

Rationale. To finally assess performance on realistic situations, the models are tested on real-world sequences *only once and without further hyper-parameter tuning* – that is, the hyper-parameter setting selected on the validation set must be used for each evaluated method.

Experimental details. We use two datasets built with real images for testing and comparing the validated models: Cityscapes [20] and ACDC [68]. Cityscapes comprises 5*k* real, annotated images from 50 cities. We further include sequences with artificial *Fog* [67] and *Rain* [35]; we will refer to the resulting dataset as Cityscapes A.W. (Artificial Weather), and to the original Cityscapes dataset as Cityscapes O. ACDC is a recent dataset introduced to study segmentation models on real domain shifts: it contains 4*k* samples from driving sequences uniformly distributed in *Fog*, *Rain*, *Night* and *Snow* conditions. For both datasets we design learning sequences where the model receives samples from a plurality of sequential environments and weather conditions (each contains 4 different sub-sequences). We have a total of 25 test sequences, with a number of frames per sequence in the order of hundreds.

4. Methods

To tackle the problem we discussed in Sec. 3.1, we can draw solutions from the literature related to domain adaptation [21, 78, 89], the recently introduced test-time adaptation [75, 87], and – to improve pre-training – from domain generalization [88, 106]. However, finding which of these families of techniques would perform best for our problem of interest is a nontrivial question.

This section details how we tailor a variety of different approaches from the research fields above for the proposed scenario. When selecting which approaches to start from, we took into account simplicity, computational cost, and memory requirements. While we selected a few representative methods to design our baselines, some of the components we consider can be replaced by or combined with more complex methods (see Sec. 2 for some examples). The same holds for the underlying segmentation model’s architecture: while we use DeepLab-V2 [12] for all methods (as it is widely used by the DA community [46, 81, 86]), any alternative segmentation network could replace it. Below we briefly describe the chosen baselines, as well as the variants we designed to adapt them to our scenario.

³Sequentiality does not play a role in the pre-training step, because *offline* training is performed on multiple epochs over the shuffled dataset.

4.1. No adaptation (NA)

One option to address the proposed scenario is to simply rely on a model pre-trained on the available source data (\mathcal{M}_{θ_0}), which can then be used in all circumstances. In this case, such model is directly used to predict the segmentation map for each sample $x_i^t \sim \mathcal{X}^t$ – without any adaptation strategy. In our experiments, we consider the two approaches detailed below, which will also be used as the starting point for the adaptation strategies defined later on.

Empirical risk minimization (ERM). The first model is simply trained via standard ERM, *i.e.* optimizing a loss (typically the pixel-level cross-entropy) using the annotated training set. This is the simplest strategy one can adopt.

Domain randomization (DR). We apply DR [77], namely we heavily randomize the appearance of the training samples – in order to reduce the chances that real-world ones will be perceived as out-of-distribution. While we use simulated data, we do not assume access to the simulator during training; hence, to randomize the training domain, we rely on standard image transformations for data augmentation [84] (find more details in Supplementary Sec. B).

4.2. Naive adaptation (N)

In the family of methods presented in this section, when the model processes a new frame x_i^t , it always starts the adaptation procedure from the pre-trained source model \mathcal{M}_{θ_0} (ERM or DR). We focus on test-time adaptation [87], self-training via pseudo-labels [42] and BN’s statistics adaptation [70]. We use the suffix **N-** to indicate approaches from this *Naive* adaptation family.

N-TENT. This method [87] adapts the BN’s parameters (the re-scaling parameters β and γ) by minimizing the average entropy of the target predictions. Provided with an image x_i^t and a segmentation model \mathcal{M}_θ , the method optimizes the following objective:

$$\operatorname{argmin}_{\beta, \gamma} \mathcal{L}_H := - \sum_{p \in x_i^t} \sum_c \hat{y}_{i,c}^p \log \hat{y}_{i,c}^p$$

where $\hat{y}_{i,c}^p$ is the prediction for class c of pixel p in image x_i^t . Given a fully differentiable model, we can optimize this objective via backpropagation [66].

N-PL. Self-training via pseudo-labels (PL) – drawn from the semi-supervised learning literature [42] – is often used in DASIS [17, 46, 107]. The idea is to make predictions on the target set (here the current image x_i^t), and then use these predictions as ground truth to fine-tune the model.⁴

N-BN. This method [70] adapts the model’s BN statistics by mixing mean and variance stored for each layer with the

statistics of the current target sample x_i^t :

$$\begin{aligned} \mu_l &:= (1 - \alpha) \cdot \mu_l + \alpha \cdot \mathbb{E}\{F^l(x_i^t)\} \\ \sigma_l^2 &:= (1 - \alpha) \cdot \sigma_l^2 + \alpha \cdot \mathbb{E}\{(F^l(x_i^t) - \mathbb{E}\{F^l(x_i^t)\})^2\} \end{aligned}$$

where $F^l(x_i^t)$ is the feature representation of the image x_i^t at layer l in the network. Note that typically BN averages across different channels *and* samples (batch). In our online case, as we process a single image at the time, the new mean μ_l and variance σ_l^2 are obtained by averaging only across channels. The normalization is then carried out as:

$$\widehat{F^l(x_i^t)} = \gamma \cdot \frac{F^l(x_i^t) - \mu_l}{\sigma_l^2} + \beta.$$

N-ST. Image-to-image (I2I) translation is a popular approach for DASIS to render target images in a style that resembles the source’s or vice-versa [19, 52, 63, 73, 92]. Yet, these methods assume that source and target distributions are known at train time, and an I2I module can be learned offline – generally, jointly with the segmentation network. On the other hand, single-image photorealistic Style Transfer (ST) can be carried out via methods that modify a *content* image to mimic the high-level appearance of a *style* image [4, 44, 51, 59, 98]. In our context, only this latter formulation can be adopted – assuming the source data is available at deployment. We test the WCT-2 [98] single-image ST algorithm⁵, by stylizing the current target image with the style of (i) a random source image **N-ST (random)** or (ii) the closest source image in the feature space **N-ST (NN)**.

4.3. Continual adaptation (C)

In contrast to the methods that we refer to as *Naive*, here models are adapted continuously – image by image – thus they accumulate knowledge from the target domains over iterations. To formalize the difference with their *Naive* counterparts, instead of adapting from \mathcal{M}_{θ_0} at each stage, *Continual* methods adapt to the frame x_i^t starting from the model $\mathcal{M}_{\theta_{i-1}}$ – which was adapted to the previous frame x_{i-1}^t .

Vanilla continual learning. Some of the methods proposed in the previous section (TENT, PL, BN) can be extended to this scenario, without particular modifications apart from the specific model the adaptation procedure starts from at each step. We use the suffix **C-**, to indicate continual learning approaches (**C-PL**, **C-TENT** and **C-BN**). Note that **C-BN**, which continuously adapts BN statistics, is an established ODA method [53, 101].

As we will show in Sec. 5, vanilla continual learning approaches are prone to a severe vulnerability: if some classes are not encountered for several frames (*e.g.*, a self-driving car may not encounter pedestrians for a while), continuous

⁴While solutions to select only confident pseudo-labels have been proposed, here we use all pseudo-labels, leaving such studies for the future.

⁵We choose this method because it is relatively fast and performs well, but any alternative style transfer method can potentially be adopted.

adaptation might lead to deteriorated performance on such classes if they are faced again. In extreme cases, the model will stop predicting some classes altogether. This is a particular form of *catastrophic forgetting* [58], at the intersection between DASIS and continual/online learning. In the following, we propose possible solutions to overcome it.

Regularizing with source data. We can regularize the chosen adaptation method, by relying on source data \mathcal{X}^s to maintain intact the patterns learned during pre-training. The idea is to ensure that the adapted model maintains the performance on the source domain – to avoid forgetting some of its categories. We achieve this by randomly sampling a small set of labeled source samples at each step and adding the cross-entropy loss w.r.t. these source samples to the objective of the adaptation method at hand (e.g., C-PL or C-TENT). This is similar to episodic memory approaches in continual learning [11]. We term the corresponding models **C-PL-SR** and **C-TENT-SR**, respectively.

Adaptive reset strategies. We store in memory a copy of the pre-trained model \mathcal{M}_{θ_0} . If at a generic step i the model \mathcal{M}_{θ_i} becomes brittle (for example, due to catastrophic forgetting), we overwrite the parameters \mathcal{M}_{θ_i} with \mathcal{M}_{θ_0} before adaptation. We formalize here a family of methods that *reset* the model parameters whenever the model meets some predefined conditions, defined as:

$$\mathcal{M}_{\theta_i} = \begin{cases} \mathcal{M}_{\theta_0}, & \text{if } \psi_i > \hat{\psi} \\ \mathcal{M}_{\theta_{i-1}}, & \text{otherwise} \end{cases} \quad (1)$$

where ψ_i is some metric we track and $\hat{\psi}$ is the corresponding threshold for triggering the reset.

Designing ψ properly is crucial, since we would like to only reset if the adapted model is performing worse than the pre-trained one. We propose here two instances of this method – (i) an oracle and (ii) a workable solution.

(i) *Oracle.* Assuming access to the ground truth, one can reset whenever the adaptive model is performing worse than the pre-trained one. Formally, given an operator Perf that evaluates a model’s performance (here, the mIoU), we have $\psi_i = \text{Perf}(\mathcal{M}_{\theta_0}) - \text{Perf}(\mathcal{M}_{\theta_{i-1}})$ and $\hat{\psi} = 0$. This oracle implementation is there to validate the potential of the reset. We will refer to this strategy as **Oracle-R-PL** or **Oracle-R-TENT**, according to the adaptation algorithm used.

(ii) *Discrepancy between predicted classes.* We propose to track the number of classes that \mathcal{M}_{θ_0} and $\mathcal{M}_{\theta_{i-1}}$ predict over different frames: if the latter consistently predicts a reduced number of classes with respect to the former, we assume that catastrophic forgetting has happened, and we reset the continual model’s parameters. Formally, given an operator NCl that outputs the number of categories predicted in a segmentation map, we have:

$$\psi_i = \sum_{j=i-K}^i \text{NCl}(\mathcal{M}_{\theta_0}(x_j^t)) - \sum_{j=i-K}^i \text{NCl}(\mathcal{M}_{\theta_{i-1}}(x_j^t)),$$

where K is the number of previous frames for which we count the number of predicted classes. We refer to this strategy as **Class-R-PL** or **Class-R-TENT**, according to the adaptation algorithm used. This is a very simple strategy, and more sophisticated reset strategies can be designed. Our goal here is simply showcasing the effectiveness of this idea in its simplest form; developing novel reset strategies is part of our future research.

5. Experiments

This section compares the methods described in Sec. 4 on our OASIS benchmark. To recap, (i) we pre-train models on GTA-5; (ii) we validate models and the hyper-parameters of the adaptation strategies on learning episodes from SYNTHIA⁶; (iii) we test the best methods on sequences from ACDC, Cityscapes A.W. and Cityscapes O.

Experimental details. We use DeepLab-V2 [12] as SIS network, implemented in PyTorch [60]. When running TENT- or PL-based adaptation algorithms, we compute a single training iteration⁷ per image. For what concerns the **evaluation metrics** we rely on, we compute the mIoU for each image encountered in each sequence, averaging over the classes that are present in each frame – according to the ground-truth. At the end of a sequence, we compute the average mIoU across all the images of that sequence.

Main results and takeaways. We report in Table 1 the average results for each dataset, together with details about computational and memory overheads with respect to the non-adapted model (NA baseline). More concretely, we report the average gain (in %) over the NA baseline and the standard deviation computed over the different sequences. All results were obtained with the hyper-parameters selected on SYNTHIA, without any hyper-parameter tuning on ACDC, Cityscapes A.W. and Cityscapes O.

As a first general observation, we see that several methods carry statistically significant gains over the NA baseline for SYNTHIA, ACDC and Cityscapes A.W. Importantly, we can appreciate that methods that rank best on SYNTHIA, in general also rank best on the test sets, showing that the validation sequences of our benchmark allow validating models and algorithms that transfer to real-world sequences. This is a crucial point, because in practice we cannot assume to be able to cross-validate the methods for every new environment the models are deployed in.

A peculiar case is Cityscapes O.: here, excluding the Oracle-R approach, none of the methods brings significant improvements. This can be due to the fact that i) the baseline model (DR $\uparrow\uparrow$) is already strong (c.f. performance

⁶A comprehensive description of the different methods’ hyper-parameters is provided in Supplementary Sec. B.

⁷While performing many iterations per frame may be computationally costly for real systems, we evaluate this scenario in Supplementary Sec. C.

		Validation	Test (Deploy)			
No adapt. baseline (NA)		SYNTHIA	ACDC	Cityscapes A.W.	Cityscapes O.	
		39.8 \pm 3.0	33.6 \pm 2.5	38.3 \pm 2.6	45.2 \pm 1.0	
Method		Improvements				Add. computation Add. memory
Style trans.	N-ST (random)	+0.7% \pm 1.7	-7.4% \pm 2.6	+4.1% \pm 1.7	+0.4% \pm 0.8	ST optim. (++) Source set (++)
	N-ST (NN)	+0.7% \pm 1.7	-5.1% \pm 0.8	+2.9% \pm 1.1	+1.0% \pm 0.3	ST optim. & NN (+++) Source set (++)
Naive adapt.	N-BN	+2.7% \pm 0.8	+2.4% \pm 0.6	+1.9% \pm 0.8	+1.2% \pm 0.1	BN stat. update (*) -
	N-PL	+3.5% \pm 1.0	+2.9% \pm 0.6	+2.4% \pm 1.0	+1.4% \pm 0.2	$\mathcal{O}(\text{trainsteps})$ (+) -
	N-TENT	+8.5% \pm 3.1	+4.9% \pm 2.0	+3.1% \pm 3.6	-1.2% \pm 0.7	$\mathcal{O}(\text{trainsteps})$ (+) -
CL Vanilla	C-BN	+6.1% \pm 3.7	+6.8% \pm 3.6	+7.7% \pm 4.3	-0.1% \pm 1.4	BN stat. update (*) -
	C-PL	-19.9% \pm 12.0	-11.7% \pm 8.1	-9.4% \pm 8.9	-17.4% \pm 3.2	$\mathcal{O}(\text{trainsteps})$ (+) -
	C-TENT	+2.5% \pm 6.8	+2.7% \pm 6.7	+6.4% \pm 5.6	-0.9% \pm 1.2	$\mathcal{O}(\text{trainsteps})$ (+) -
CL SrcReg	C-PL-SR	+4.9% \pm 3.9	+2.8% \pm 2.9	+3.9% \pm 3.2	+0.5% \pm 0.5	$\mathcal{O}(\text{trainsteps})$ (+) Source set (++)
	C-TENT-SR	+7.2% \pm 4.0	+5.8% \pm 3.7	+4.7% \pm 3.7	+0.2% \pm 0.5	$\mathcal{O}(\text{trainsteps})$ (+) Source set (++)
CL Reset	Class-R-PL	+7.2% \pm 3.9	+8.2% \pm 3.4	+9.0% \pm 5.1	+0.0% \pm 1.4	$\mathcal{O}(\text{trainsteps})$ (+) Backup net (+)
	Class-R-TENT	+8.3% \pm 4.2	+7.3% \pm 3.9	+9.1% \pm 4.9	+0.9% \pm 1.3	$\mathcal{O}(\text{trainsteps})$ (+) Backup net (+)
CL Oracle	Oracle-R-PL	+10.8% \pm 4.5	+11.6% \pm 3.8	+12.7% \pm 5.6	+2.9% \pm 1.4	$\mathcal{O}(\text{trainsteps})$ (+) Backup net (+)
	Oracle-R-TENT	+11.4% \pm 4.4	+10.9% \pm 4.1	+12.2% \pm 5.9	+1.9% \pm 1.4	$\mathcal{O}(\text{trainsteps})$ (+) Backup net (+)

Table 1. Results associated with SYNTHIA [65] ACDC [68], Cityscapes Original (O.) [20] and Cityscapes Artificial Weather (A.W.) [35, 67] sequences. Hyper-parameters used for ACDC, Cityscapes O. and Cityscapes A.W. are the ones selected via SYNTHIA experiments. Results are reported as average percentage improvements over the non-adapted baselines obtained with the pre-trained model \mathcal{M}_{θ_0} (see NA results, on top). For all experiments, we use DR $\uparrow\uparrow$ as pre-trained model. The final two columns provide insights about additional computations and additional memory requirements over the NA baseline; the symbols (+)/(++)/(++) provide qualitative orderings for the different methods (increasing overhead). *Updating BN statistics requires neglectable computations. Bold numbers indicate best results for each dataset; different numbers per column are highlighted in case of comparable results.

Effect of pre-training w/ domain randomization (DR) on NA				
Training	SYNTHIA	ACDC	Cityscapes A.W.	Cityscapes O.
ERM	35.9 \pm 2.5	29.5 \pm 2.5	35.6 \pm 1.9	40.3 \pm 0.9
DR \uparrow	34.3 \pm 3.3	29.5 \pm 2.4	36.2 \pm 2.3	41.2 \pm 1.0
DR $\uparrow\uparrow$	39.8 \pm 3.0	33.6 \pm 2.5	38.3 \pm 2.6	45.2 \pm 1.0
DR $\uparrow\uparrow\uparrow$	31.9 \pm 3.0	26.7 \pm 2.3	33.2 \pm 2.5	37.7 \pm 1.1

Table 2. No adaptation (NA) results obtained by the pre-trained model \mathcal{M}_{θ_0} , trained via standard ERM or DR, and applied to SYNTHIA [65], ACDC [68], and Cityscapes [20] (Artificial Weather and Original) – in mIoU. The \uparrow indicate the severity of randomization (image transformations) applied in each training batch (see Supplementary Sec. B.). Note that we compute mIoU results per image, and compute the average at the end (our own protocol); these results should not be compared with those reported in standard DASIS [78].

across the different benchmarks in Table 2) and ii) there is not much weather, lighting or even environment changes in this dataset despite the fact that images have been acquired in different cities (see examples in Supplementary Sec. A).

The rest of the section analyzes the methods we have evaluated. We provide our *main observations* below:

Domain randomization improves the robustness of the initial model. Table 2 compares the performance of SIS models trained without data augmentation (ERM) vs. the model trained with increasing level of domain randomiza-

tion (DR). We see that, if the randomization level is properly selected on the validation set (SYNTHIA), the corresponding best DR model (third row) not only significantly outperforms ERM (first row) on all test datasets, but also the models trained with the other randomization levels, showing good generalization. Therefore, for all adaptation experiments (Table 1) we use \mathcal{M}_{θ_0} corresponding to DR $\uparrow\uparrow$.

It is of utmost importance to start from a strong initial source model, *i.e.* we should use the best-performing NA baseline. For instance, applying N-TENT to an ERM model on ACDC leads to an average mIoU performance of 31.5, lower than the one achieved with the NA(DR $\uparrow\uparrow$) model. Using weak baselines can lead to misleading results, as also shown by Gulrajani and Lopez-Paz [31] in the context of domain generalization.

Single-image style transfer does not seem sufficient for adaptation. While in general style transfer helps DASIS both as a pre-processing [22, 45, 76] or learned jointly with the segmentation on the whole dataset [16, 33, 54, 55], single-image style transfer from source to target leads to marginal improvements over the baselines in most of our experiments, and significantly degrades performance on ACDC. For what concerns the latter, we observed that ST can still be effective when adapting to *Night* sequences, probably due to the fact that the appearance change is dras-

tic (Supplementary Sec. C).

Batch norm’s statistics adaptation generally helps. It is known for DA [53] and for model robustness [70] that adapting the BN statistics is a simple yet effective way to improve model performance. This is also confirmed in our case, as shown in Table 1, where BN adaptation always brings improvements over the NA baseline – both when applied independently frame-by-frame (N-BN) or in a cumulative manner when adapting continuously (C-BN). In Supplementary Sec. C, we provide further analyses on the impact of mixing BN statistics at test-time [70].

Vanilla continual adaptation is not always a safe choice. We compare *Naive* (N-PL/N-TENT) adaptation strategies vs. their *Continual* counterparts (C-PL/C-TENT) in Table 1. These results suggest that continual adaptation models may be forgetting important visual features, resulting in degraded predictions over time (*c.f.* blue and orange curves in Fig. 3). We further report qualitative evidence of catastrophic forgetting in Fig. 3 (top).

Using source regularization generally helps. Introducing a regularization term (C-PL-SR/C-TENT-SR) mitigates the degradation in performance that we witnessed in some settings (after switching from *Naive* to *Continual* methods). If we compare the methods with and without SR in Table 1, we can observe that adding source regularization always helps the C- methods (*e.g.*, C-PL-SR vs. C-PL). We further report in Supplementary Sec. C qualitative evidences supporting our claim that SR helps mitigating catastrophic forgetting of important information – while learning continuously.

Using reset methods generally helps. Comparing Class-R- and Oracle-R- methods with their *Naive* (N-) and *Continual* (C-) counterparts in Table 1, we can appreciate how models generally benefit from a reset strategy. As expected, the oracle outperforms all other methods in every benchmark: while it is not necessarily an upper bound, it relies on ground truth information, and hence has significant leverage over the other methods. The gap between Class-R- and Oracle-R- serves as a motivation for future research in reset methods, *i.e.* different instances of Eq. (1). Furthermore, note also that, while both source regularization and model reset aim at overcoming catastrophic forgetting, in general the latter provides stronger improvements. Being complementary methods, they could potentially be combined, yet assessing this is out of scope here. Finally, Fig. 3 compares performance over one ACDC sequence for N-PL, C-PL and Class-R-PL (in orange, blue and green, respectively), showing the catastrophic forgetting avoided with the reset strategy introduced in Section 4.3.

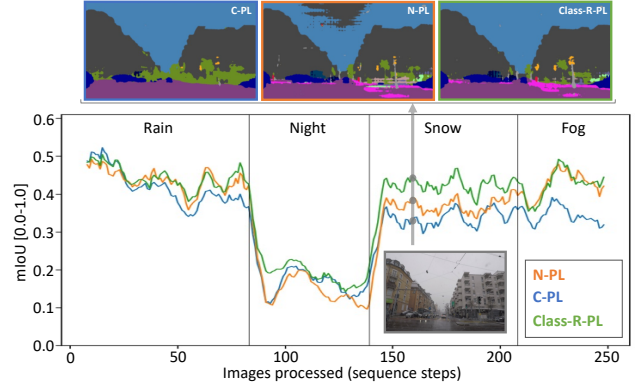


Figure 3. **Naive vs. Continual vs. Reset.** Performance evolution of Naive, Vanilla continual, and Class-reset versions of PL (N-PL, C-PL and Class-R-PL, in orange, blue and green, respectively) – for one ACDC [68] sequence. In the plot, we see that learning continuously without precautions results in sub-optimal performance, and that reset allows maintaining a performance close to the naive counterpart in some parts of the sequence, while significantly improving over it in others. Images on top provide a qualitative comparison between predictions of the methods when processing the reported image; C-PL has catastrophically forgotten the classes *sidewalk*, *pedestrian* and *pole* (best viewed in color and zoomed in).

6. Concluding remarks

In this paper, we formalize the problem of adapting semantic image segmentation models to constantly changing environments. The resulting task better reflects the challenging conditions that a system typically faces when deployed in the real world: it is exposed to a stream of unlabelled samples whose underlying distribution is constantly shifting. To study this specific problem, we introduce **the OASIS benchmark**, which separates the validation step from the deploy step, ensuring that only the best validated model and strategy are evaluated on the test set.

Equipped with this benchmark, we borrow approaches from related fields and extend some of them, to serve as diverse baselines for our task, and extensively analyze their behaviors. The resulting observations allow us to provide a first answer to the research questions raised in Sec 3. (1) First, we observe that carefully choosing the right level of domain randomization allows building more robust models that are resilient to the domain shift present in our test sequences, showing that the quality of pre-training should not be neglected. (2-3) We compare *naive* approaches, which adapt to each frame restarting from the pre-trained model, with *continual* ones, that adapt continuously. Our experiments show that the latter can lead to catastrophic forgetting. (4) As a first attempt to mitigate this issue, we propose a reset strategy that carries promising results.

We believe that addressing the shortcomings of continual, unsupervised adaptation methods is a promising research path – towards the deployment of machine learning systems in the real world. We hope this benchmark will ease the emergence of new research directions in this sense.

References

- [1] Pillow imageenhance module. <https://pillow.readthedocs.io/en/3.0.x/reference/ImageEnhance.html>. 15
- [2] Pillow imageops module. <https://pillow.readthedocs.io/en/3.0.x/reference/ImageOps.html>. 15
- [3] Python imaging library. <https://github.com/python-pillow/Pillow>. 15
- [4] Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo. Ultrafast photorealistic style transfer via neural architecture search. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 5, 19
- [5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: a Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(12):2481–2495, 2017. 2
- [6] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. In *International Conference on Learning Representations (ICLR), Workshops*, 2018. 3
- [7] Wonmin Byeon, Thomas M. Breuel, Federico Raue, and Marcus Liwicki. Scene Labeling with LSTM Recurrent Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [8] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006. 3
- [9] Siddhartha Chandra and Iasonas Kokkinos. Fast, Exact and Multi-Scale Inference for Semantic Image Segmentation with Deep Gaussian CRFs. In *European Conference on Computer Vision (ECCV)*, 2016. 2
- [10] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about Structure: Adapting Structural Information across Domains for Boosting Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [11] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc’Aurelio Ranzato. On tiny episodic memories in continual learning. arXiv:1902.10486, 2019. 6
- [12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(4):834–848, 2017. 2, 4, 6, 15
- [13] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv:1706.05587, 2017. 2
- [14] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to Scale: Scale-aware Semantic Image Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [15] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 2
- [16] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. CrDoCo: Pixel-level Domain Transfer with Cross-Domain Consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 7
- [17] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No More Discrimination: Cross City Adaptation of Road Scene Segmenters. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5
- [18] Zhiyuan Chen and Bing Liu. *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2018. 3
- [19] Anoop Cherian and Alan Sullivan. Sem-gan: Semantically-consistent image-to-image translation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1797–1806, 2019. 5, 18
- [20] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 7, 13, 14
- [21] Gabriela Csurka. Deep Visual Domain Adaptation. arXiv:2012.14176, 2020. 2, 4
- [22] Gabriela Csurka, Fabien Baradel, Boris Chidlovskii, and Stéphane Clinchant. Discrepancy-Based Networks for Unsupervised Domain Adaptation: A Comparative Study. In *IEEE International Conference on Computer Vision (ICCV), Workshops*, 2017. 7
- [23] Gabriela Csurka, Riccardo Volpi, and Boris Chidlovskii. Unsupervised Domain Adaptation for Semantic Image Segmentation: a Comprehensive Surveys. arXiv:2112.03241, 2021. 2
- [24] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. SSF-DAN: Separated Semantic Feature Based Domain Adaptation Network for Semantic Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2
- [25] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual Attention Network for Scene Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [26] Adrien Gaidon and Eleonora Vig. Online Domain Adaptation for Multi-Object Tracking. In *BMVC British Machine Vision Conference (BMVC)*, 2015. 3
- [27] Adrien Gaidon, Gloria Zen, and Jose A. Rodriguez-Serrano. Self-Learning Camera: Autonomous Adaptation of Object Detectors to Unlabeled Video Streams. arXiv:1406.4296, 2014. 3
- [28] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks.

- In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 19
- [29] Golnaz Ghiasi and Charless C. Fowlkes. Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation. In *European Conference on Computer Vision (ECCV)*, 2016. 2
- [30] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. DLOW: Domain Flow for Adaptation and Generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [31] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021. 7
- [32] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive Pyramid Context Network for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [33] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *International Conference on Machine Learning (ICML)*, 2018. 1, 2, 7
- [34] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. arXiv:1612.02649, 2016. 1, 2, 4
- [35] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 7, 14
- [36] Haoshuo Huang, Qixing Huang, and Philipp Krähenbühl. Domain Transfer Through Deep Activation Matching. In *European Conference on Computer Vision (ECCV)*, 2018. 2
- [37] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 19
- [38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 3, 18
- [39] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Style Normalization and Restitution for Domain Generalization and Adaptation. arXiv:2101.00588, 2020. 2
- [40] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 19
- [41] Yevhen Kuznetsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3
- [42] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning (ICML) Workshops*, 2013. 5
- [43] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid Attention Network for Semantic Segmentation. In *BMVC British Machine Vision Conference (BMVC)*, 2018. 2
- [44] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *European Conference on Computer Vision (ECCV)*, 2018. 5, 19
- [45] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying Neural Style Transfers. In *AAAI International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. 7
- [46] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional Learning for Domain Adaptation of Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4, 5
- [47] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic Object Parsing with Graph LSTM. In *European Conference on Computer Vision (ECCV)*, 2016. 2
- [48] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *International Conference on Learning Representations (ICLR)*, 2015. 2
- [49] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [50] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [51] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 19
- [52] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *International Conference on Learning Representations (ICLR)*, 2019. 5, 18
- [53] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the Wild through Online Domain Adaptation. In *International Conference on Intelligent Robotics and Systems (IROS)*, 2018. 3, 5, 8, 18
- [54] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to Image Translation for Domain Adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7
- [55] Luigi Musto and Andrea Zinelli. Semantically Adaptive Image-to-image Translation for Domain Adaptation of Semantic Segmentation. In *BMVC British Machine Vision Conference (BMVC)*, 2020. 7
- [56] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

- [57] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised Intra-domain Adaptation for Semantic Segmentation through Self-Supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [58] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113:57–71, 2019. 3, 6
- [59] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 19
- [60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, 2019. 6, 15
- [61] Horia Porav, Tom Bruls, and Paul Newman. Don’t worry about the weather: Unsupervised condition-dependent domain adaptation. In *IEEE Intelligent Transportation Systems Conference*, 2019. 3
- [62] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Koltun Vladlen. Playing for Data: Ground Truth from Computer Games. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 4, 13, 15
- [63] Stephan R. Richter, Hassan Abu AlHaija, and Vladlen Koltun. Enhancing photorealism enhancement. arXiv:2105.04619, 2021. 5, 18
- [64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing & Computer Assisted Intervention (MICCAI)*, 2015. 2
- [65] German Ros, Laura Sellart, Joanna Materzyńska, David Vázquez, and Antonio M. López. The SYNTHIA Dataset: a Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 4, 7, 13, 18, 19
- [66] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986. 5
- [67] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *European Conference on Computer Vision (ECCV)*, 2018. 4, 7, 14
- [68] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding. 2021. 4, 7, 8, 13, 19
- [69] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [70] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3, 5, 8
- [71] Alexander G. Schwing and Raquel Urtasun. Fully Connected Deep Structured Networks. arXiv:1506.04579, 2015. 2
- [72] Abhishek Sharma, Abhishek Kumar, Hal Daumé III., and David W. Jacobs. Generalized Multiview Analysis: a Discriminative Latent Space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3
- [73] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 18
- [74] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2
- [75] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020. 3, 4
- [76] Christopher Thomas and Adriana Kovashka. Artistic Object Recognition by Unsupervised Style Adaptation. In *Asian Conference on Computer Vision (ACCV)*, 2019. 7
- [77] Joshua Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *International Conference on Intelligent Robotics and Systems (IROS)*, 2017. 4, 5, 15, 18
- [78] Marco Toldo, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. Unsupervised Domain Adaptation in Semantic Segmentation: a Review. arXiv:2005.10876, 2020. 2, 4, 7
- [79] Alessio Tonioni, Oscar Rahnema, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip H.S. Torr. Learning to adapt for stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [80] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [81] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schuster, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to Adapt Structured Output Space for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4
- [82] Paul Voigtlaender and Bastian Leibe. Online Adaptation of Convolutional Neural Networks for Video Object Segmentation. In *BMVC British Machine Vision Conference (BMVC)*, 2017. 3

- [83] Riccardo Volpi, Diane Larlus, and Gregory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [84] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 5
- [85] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Neural Information Processing Systems (NeurIPS)*, 2018. 2
- [86] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 4
- [87] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021. 3, 4, 5, 20
- [88] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to Unseen Domains: A Survey on Domain Generalization. arXiv:2103.03097, 2020. 2, 4
- [89] Mei Wang and Weihong Deng. Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 312:135–153, 2018. 2, 4
- [90] Xiaoyu Wang, Gang Hua, and Tony X. Han. Detection by Detections: Non-parametric Detector Adaptation for a Video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3
- [91] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. Diversified arbitrary style transfer via deep feature perturbation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 19
- [92] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. DCAN: Dual Channel-wise Alignment Networks for Unsupervised Scene Adaptation. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 5, 18
- [93] Zuxuan Wu, Xin Wang, Joseph E. Gonzalez, Tom Goldstein, and Larry S. Davis. Ace: Adapting to changing environments for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 3
- [94] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *International Conference on Robotics and Automation (ICRA)*, 2018. 3
- [95] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 2
- [96] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M. López. Domain Adaptation of Deformable Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(12):2367–2380, 2014. 3
- [97] Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase Consistent Ecological Domain Adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2
- [98] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 19
- [99] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization without Accessing Target Domain Data. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 4
- [100] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Dong Liu, and Tao Mei. Transferring and Regularizing Prediction for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [101] Zhenyu Zhang, Stéphane Lathuiliere, Andrea Pilzer, Nicu Sebe, Elisa Ricci, and Jian Yang. Online adaptation through meta-learning for stereo depth estimation. arXiv:1904.08462, 2019. 3, 5
- [102] Zhenyu Zhang, Stephane Lathuiliere, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3
- [103] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [104] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2
- [105] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [106] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization: A Survey. arXiv:2103.02503, 2020. 2, 4
- [107] Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 5

Supplementary Material

A. Details of the datasets

We include here additional details related to the datasets we used, namely GTA-5 [62], SYNTHIA [65], Cityscapes [20] and ACDC [68]. In particular, in Fig. A.1 we provide example images from each dataset with the corresponding ground-truth segmentation map. Then, in Sec A.1 we discuss which categories we considered, as different datasets have different annotations available. Finally, we provide details about the sequences we designed for the experiments in Sec A.2.

A.1. Classes

In this section, we detail the subset of categories that we considered for our experiments, an important point since some classes are available in some datasets but not in others. For example, lane-marking is available in SYNTHIA but not in GTA-5/Cityscapes/ACDC, and vice-versa the terrain class is available in GTA-5/Cityscapes/ACDC but not in SYNTHIA. We used the 19 classes below from the GTA-5 that are also available in Cityscapes/ACDC: *road, sidewalk, building, fence, pole, light, sign, vegetation, sky, person, car, bicycle, bus, train, motorcycle, wall, terrain, truck, rider*. For GTA-5/Cityscapes/ACDC, we use all of them. For SYNTHIA, we only use the ones highlighted in magenta (the others being unavailable).

A.2. Sequences

In the following, we report the details of the different sequences used in our experiments. This is limited to the SYNTHIA, ACDC, Cityscapes O. and Cityscapes A.W. dataset. In the case of GTA-5, the whole dataset is used for the offline pre-training step; images are randomly sampled from the dataset and the dataset is parsed over several epochs.

SYNTHIA. For clarity, this paragraph uses the sequence IDs as reported in the dataset’s directories: *01* is *Highway*, *04* is *Old European Town* and *05* is *New York-like City*. The weather/daylight/seasonal conditions we used are *Summer, Spring, Fall, Winter, Dawn, Sunset, Night, Rain, Fog, Rain-night, Winter-night*. To indicate a sub-sequence shift, we use arrows (\rightarrow). For each sub-sequence, we indicate the specific environment and the specific weather/daylight condition (e.g., *04/Night*). We use 300 consecutive frames per sub-sequence (to limit the length of the experiments), and build the following sequences – each one totalling 1.5k samples:

- *05/Night (300 frames) \rightarrow 01/Dawn (300 frames) \rightarrow 01/Winter (300 frames) \rightarrow 05/Winternight (300 frames) \rightarrow 04/Soft-rain (300 frames)*

- *04/Night (300 frames) \rightarrow 01/Winter (300 frames) \rightarrow 04/Soft-rain (300 frames) \rightarrow 05/Winternight (300 frames) \rightarrow 05/Fog (300 frames)*
- *01/Winternight (300 frames) \rightarrow 05/Winternight (300 frames) \rightarrow 05/Night (300 frames) \rightarrow 04/Sunset (300 frames) \rightarrow 04/Winter (300 frames)*
- *05/Winternight (300 frames) \rightarrow 05/Dawn (300 frames) \rightarrow 05/Night (300 frames) \rightarrow 05/Sunset (300 frames) \rightarrow 01/Winter (300 frames)*
- *05/Softtrain (300 frames) \rightarrow 01/Night (300 frames) \rightarrow 04/Fog (300 frames) \rightarrow 05/Winter (300 frames) \rightarrow 01/Winternight (300 frames)*
- *01/Night (300 frames) \rightarrow 04/Fog (300 frames) \rightarrow 01/Fall (300 frames) \rightarrow 05/Fall (300 frames) \rightarrow 05/Rain (300 frames)*
- *04/Spring (300 frames) \rightarrow 05/Winter (300 frames) \rightarrow 04/Night (300 frames) \rightarrow 01/Dawn (300 frames) \rightarrow 04/Rain-night (300 frames)*
- *01/Winter (300 frames) \rightarrow 04/Sunset (300 frames) \rightarrow 04/Spring (300 frames) \rightarrow 01/Spring (300 frames) \rightarrow 05/Fog (300 frames)*
- *04/Rainnight (300 frames) \rightarrow 04/Softtrain (300 frames) \rightarrow 05/Winter (300 frames) \rightarrow 05/Fog (300 frames) \rightarrow 01/Dawn (300 frames)*

Cityscapes O. We only use frames from the original Cityscapes dataset (without weather variations) for which “fine-grained” annotation is provided (Cityscapes also provides frames for which some “coarse” annotation is provided). Apart from this, we do not perform any cut to the sub-sequences, and build the following sequences:

- *Aachen (174 frames) \rightarrow Hamburg (248 frames) \rightarrow Frankfurt (267 frames) \rightarrow Munster (174 frames)*
- *Jena (119 frames) \rightarrow Hamburg (248 frames) \rightarrow Zurich (122 frames) \rightarrow Hanover (196 frames)*
- *Hamburg (248 frames) \rightarrow Stuttgart (196 frames) \rightarrow Tübingen (144 frames) \rightarrow Darmstadt (85 frames)*
- *Stuttgart (196 frames) \rightarrow Bochum (96 frames) \rightarrow Monchengladbach (94 frames) \rightarrow Bremen (316 frames)*
- *Lindau (59 frames) \rightarrow Bochum (96 frames) \rightarrow Aachen (174 frames) \rightarrow Stuttgart (196 frames)*
- *Monchengladbach (94 frames) \rightarrow Dusseldorf (221 frames) \rightarrow Jena (119 frames) \rightarrow Strasbourg (365 frames)*
- *Jena (119 frames) \rightarrow Strasbourg (365 frames) \rightarrow Bochum (96 frames) \rightarrow Dusseldorf (221 frames)*
- *Strasbourg (365 frames) \rightarrow Stuttgart (196 frames) \rightarrow Tübingen (144 frames) \rightarrow Monchengladbach (94 frames)*
- *Krefeld (99 frames) \rightarrow Erfurt (109 frames) \rightarrow Tübingen (144 frames) \rightarrow Strasbourg (365 frames)*
- *Monchengladbach (94 frames) \rightarrow Lindau (59 frames) \rightarrow Aachen (174 frames) \rightarrow Jena (119 frames)*

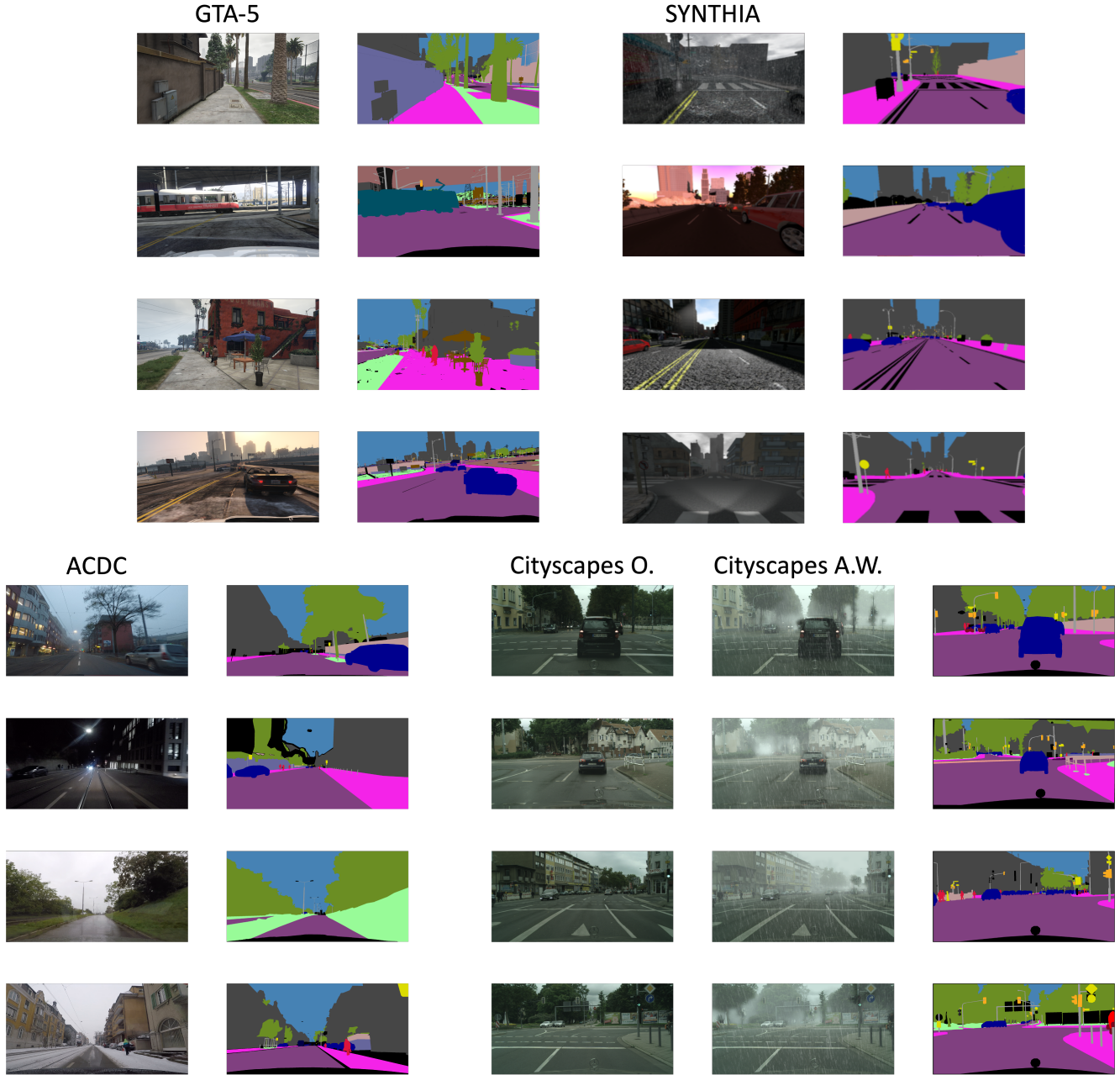


Figure A.1. Sample images of the datasets used for our experiments.

Cityscapes A.W. We define the following sequences by combining sub-sequences (without cut) from the original Cityscapes [20] (Clean), from Cityscapes sequences with artificial *Fog* [67] and with artificial *Rain* [35]. “Clean” indicates that the original sequences are used.

- *Zurich/Clean* (122 frames) → *Darmstadt/Fog* (85 frames) → *Dusseldorf/Rain* (68 frames) → *Jena/Fog* (119 frames)
- *Munster/Rain* (30 frames) → *Hamburg/Fog* (248 frames) →

Cologne/Clean (154 frames) → *Erfurt/Clean* (109 frames)

- *Bremen/Clean* (316 frames) → *Stuttgart/Fog* (196 frames) → *Aachen/Rain* (65 frames) → *Tubingen/Clean* (144 frames)
- *Dusseldorf/Rain* (68 frames) → *Darmstadt/Clean* (85 frames) → *Tubingen/Fog* (144 frames) → *Bremen/Rain* (53 frames)
- *Bremen/Rain* (53 frames) → *Krefeld/Clean* (99 frames) →

Lindau/Fog (59 frames) → Bochum/Clean (96 frames)

- *Cologne/Clean (154 frames) → Munster/Rain (30 frames) → Hanover/Fog (196 frames) → Bremen/Clean (316 frames)*
- *Frankfurt/Fog (267 frames) → Erfurt/Rain (59 frames) → Zurich/Clean (122 frames) → Cologne/Clean (154 frames)*
- *Hanover/Clean (196 frames) → Aachen/Fog (174 frames) → Jena/Fog (119 frames) → Munster/Rain (30 frames)*
- *Bremen/Rain (53 frames) → Ulm/Fog (95 frames) → Zurich/Clean (122 frames) → Darmstadt/Fog (85 frames)*
- *Erfurt/Rain (59 frames) → Ulm/Clean (95 frames) → Aachen/Rain (65 frames) → Lindau/Fog (59 frames)*

ACDC. We do not perform any cut to the sub-sequences, and build the following sequences:

- *GP010476/Fog (41 frames) → GP010402/Rain (31 frames) → GP030176/Snow (22 frames) → GP010376/Night (56 frames)*
- *GP010476/Fog (41 frames) → GOPR0351/Night (149 frames) → GOPR0122/Snow (48 frames) → GP020402/Rain (102 frames)*
- *GOPR0402/Rain (83 frames) → GP010376/Night (56 frames) → GP010607/Snow (69 frames) → GOPR0478/Fog (41 frames)*
- *GP040176/Snow (86 frames) → GP020402/Rain (102 frames) → GP020397/Night (44 frames) → GP010476/Fog (41 frames)*
- *GOPR0122/Snow (48 frames) → GP020475/Fog (37 frames) → GOPR0356/Night (50 frames) → GP010402/Rain (31 frames)*

B. Details of the methods

Hyper-parameter selection. We report details related to the choice of hyper-parameters, expanding on Section 5 from the main manuscript. We train our models with a DeepLab-V2 [12] architecture, implemented in PyTorch [60]. We pre-train our models on GTA-5 [62] for 6 epochs, using SGD optimizer with learning rate $\eta = 2.5 \cdot 10^{-4}$, momentum $\alpha = 0.9$ and weight decay $\lambda = 5 \cdot 10^{-4}$. For GPU-memory constraints, we set the batch size to 1. In the following, we report the hyper-parameters associated with each method.

- **N-BN:** BN momentum $\alpha = 0.1$
- **C-BN:** BN momentum $\alpha = 0.1$
- **N-TENT:** learning rate $\eta = 1.0$
- **C-TENT:** learning rate $\eta = 0.01$
- **C-TENT-SR:** learning rate $\eta = 0.01$, source regularizer weight $\gamma = 1.0$

- **Class-R-TENT:** learning rate $\eta = 0.1$, $K = 1$, $\psi = 1.0$
- **Oracle-R-TENT:** learning rate $\eta = 1.0$
- **N-PL:** learning rate $\eta = 10^{-4}$
- **C-PL:** learning rate $\eta = 10^{-4}$
- **C-PL-SR:** learning rate $\eta = 10^{-4}$, source regularizer weight $\gamma = 2.0$
- **Class-R-PL:** learning rate $\eta = 10^{-4}$, $K = 1$, $\psi = 1.0$
- **Oracle-R-PL:** learning rate $\eta = 10^{-4}$

The hyper-parameters were cross-validated on SYNTHIA sequences, and kept unchanged for ACDC, Cityscapes A.W. and Cityscapes O. sequences.

Domain randomization (DR). To perform domain randomization (DR [77]), we modify the aspect of training samples by applying K different, random image transformations before feeding each sample to the model. Referring to notation in Table 2 in the main paper, we validate $K = 2, 3, 4$ for $\text{DR}\uparrow$, $\text{DR}\uparrow\uparrow$ and $\text{DR}\uparrow\uparrow\uparrow$ respectively. In the following, we report the different transformations we rely on during training and we refer the reader to PIL [1–3] for a more detailed documentation.

- *Identity:* the image remains unchanged.
- *Brightness:* the brightness of the image is perturbed, with intensity in the range $[0.2; 1.8]$.
- *Color:* the color of the image is perturbed, with intensity in the range $[0.2; 1.8]$.
- *Contrast:* the contrast of the image is perturbed, with intensity in the range $[0.2; 1.8]$.
- *RGB perturbations:* a random scalar in the range $[0; 120]$ is added to each of the RGB channels.
- *RGB-to-gray:* the image is converted to grayscale.

C. Additional results

We provide in this section additional results, to extend the main ones included in the manuscript. To provide a roadmap, in Sec. C.1 we analyze the effect of BN momentum η on the N-BN method; in Sec. C.2 we show adaptation results obtained when starting from the ERM source model; in Sec. C.3 we analyze how the results vary in the iterative methods N-PL and N-TENT when increasing the number of adaptation iterations; in Sec. C.4 we provide additional ACDC results, for sequences where only the urban environment change, but the weather/daylight condition is fixed; and finally, in Sec. C.5 we provide more continual learning curves such as the one shown in Fig. 3 in the main paper.

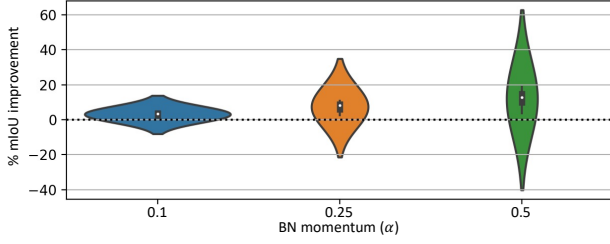


Figure C.2. SYNTHIA results for N-BN – varying BN momentum (α).

C.1. BN adaptation

In Figure C.2 we showcase the distribution of N-BN results for the SYNTHIA sequences, when we vary the BN momentum ($\alpha = 0.1, 0.25, 0.5$ in blue, orange and green, respectively). The general trend is that increasing the momentum α – that means, increasing the impact of the target sample’s statistics when mixing those with the source statistics – leads to higher average results at the price of a significantly larger spread. These results indicate that the algorithm will perform significantly better on some sequences, and significantly worse in others. Since this approach is versatile and could be applied in tandem with any other method (for example, one could perform continual adaptation with a reset mechanism, and also adapt BN statistics with increased BN momentum on each sample), we believe that a more thorough understanding of its behavior on the proposed task represents an interesting research direction.

C.2. Adapting from ERM v.s. adapting from DR

In Table C.1 we compare the adaptation results on ACDC between the models started from ERM with the models started from DR $\uparrow\uparrow$. We can observe that none of the models started from ERM achieves the performance of the baseline DR $\uparrow\uparrow$ – apart the Oracle-R, which starting from ERM performs on par with the DR $\uparrow\uparrow$ NA baseline. This confirms the crucial importance of the initial \mathcal{M}_{θ_0} . A second observation we can make is that while the numbers are lower for models that start the adaptation from ERM (first column), the overall improvement trend is consistent in general; this emphasizes that the conclusions made concerning the different adaptation strategies hold even if we change the initial \mathcal{M}_{θ_0} .

C.3. Adaptation iterations for PL and TENT

In Table C.2 we compare N-PL and N-TENT results as we increase the number of iterations. For example, in the case of N-PL we iterate several times between pseudo-labeling and updating the model. We can observe that, in general, increasing the number of iterations yields better results for N-PL. In the case of N-TENT, in the multi-iteration

Comparing adaptation from ERM and DR

Method	Pre-trained model	
	ERM	DR $\uparrow\uparrow$
<i>No adaptation</i>		
	29.5 \pm 2.5	33.6 \pm 2.5
<i>Style transfer</i>		
N-ST (NN)	27.4 \pm 2.2	31.9 \pm 2.3
N-ST (rand)	25.5 \pm 1.6	31.1 \pm 1.7
<i>Naive adapt.</i>		
N-BN	30.2 \pm 2.6	34.4 \pm 2.5
N-PL	30.4 \pm 2.6	34.6 \pm 2.5
N-TENT	31.5 \pm 2.7	35.3 \pm 2.7
<i>CL adapt.</i>		
C-BN	31.7 \pm 2.2	35.9 \pm 2.3
C-PL	27.8 \pm 3.4	29.7 \pm 3.6
C-TENT	31.2 \pm 2.8	34.5 \pm 3.4
<i>CL+SR adapt.</i>		
C-PL-SR	31.3 \pm 2.8	34.5 \pm 2.7
C-TENT-SR	31.1 \pm 2.9	35.6 \pm 2.8
<i>Adaptive-reset adapt.</i>		
Class-N-PL	32.4 \pm 2.3	36.3 \pm 2.3
Class-N-TENT	32.1 \pm 2.3	36.0 \pm 2.3
<i>Oracle-reset adapt.</i>		
Oracle-N-PL	33.6 \pm 2.4	37.5 \pm 2.3
Oracle-N-TENT	33.6 \pm 2.4	37.2 \pm 2.3

Table C.1. Comparison between models adapted starting from ERM or DR $\uparrow\uparrow$ pre-training. Results reported in mIoU.

case, results can be significantly improved by reducing the learning rate (except in the case of Cityscapes O.). Naturally, this improvement comes with an increased computational cost, which can be prohibitive according to specific applications – *e.g.*, autonomous driving.

C.4. Sequences with multiple cities and fixed weather/daylight conditions

We report in Table C.3 results associated with ACDC sequences where the condition (*Fog, Night, Rain, Snow*) is fixed, and only the urban environment change. For each condition, results are averaged over the following sequences:

Fog:

- GP020475 \rightarrow GOPR0478 \rightarrow GOPR0476 \rightarrow GP010476
- GOPR0477 \rightarrow GP020478 \rightarrow GP010476 \rightarrow GOPR0475
- GP020475 \rightarrow GOPR0476 \rightarrow GOPR0478 \rightarrow GP010476
- GOPR0477 \rightarrow GOPR0479 \rightarrow GOPR0476 \rightarrow GP020475

Varying the number of iterations for N-PL and N-TENT						
Method	Adapt iter.	Learn. rate	Sequence type			
			SYNTHIA	ACDC	Cityscapes A.W.	Cityscapes O.
N-PL	1	0.0001	+3.5% \pm 1.0	+2.9% \pm 0.6	+2.4% \pm 1.0	+1.4% \pm 0.2
N-PL	3	0.0001	+7.8% \pm 2.7	+6.5% \pm 1.7	+5.2% \pm 2.6	+2.4% \pm 0.4
N-PL	5	0.0001	+9.5% \pm 3.7	+8.4% \pm 2.7	+6.5% \pm 3.7	+2.2% \pm 0.6
N-TENT	1	1.0	+8.5% \pm 3.1	+4.9% \pm 2.0	+3.1% \pm 3.6	−1.2% \pm 0.7
N-TENT	3	1.0	+8.4% \pm 4.2	+4.0% \pm 3.1	+3.4% \pm 5.3	−3.0% \pm 1.0
N-TENT	5	1.0	+6.8% \pm 4.6	+2.9% \pm 3.5	+2.8% \pm 5.9	−4.5% \pm 1.2
N-TENT	1	0.1	+4.0% \pm 1.1	+3.1% \pm 0.7	+2.6% \pm 1.2	+1.4% \pm 0.2
N-TENT	3	0.1	+8.9% \pm 2.9	+6.9% \pm 1.9	+5.4% \pm 3.0	+2.1% \pm 0.4
N-TENT	5	0.1	+10.7% \pm 3.8	+8.5% \pm 2.9	+6.4% \pm 4.2	+1.5% \pm 0.6

Table C.2. Results (relative performance gain in %) obtained on N-PL and N-Tent when increasing the number of training iterations.

Night

- $GP0R351 \rightarrow GP010376 \rightarrow GP0R356 \rightarrow GP020397$
- $GP020397 \rightarrow GP0R356 \rightarrow GP0R376 \rightarrow GP0R351$
- $GP010397 \rightarrow GP010376 \rightarrow GP0R351 \rightarrow GP0R356$
- $GP0R356 \rightarrow GP0R351 \rightarrow GP0R376 \rightarrow GP010376$

Rain

- $GP0R400 \rightarrow GP020400 \rightarrow GP020402 \rightarrow GP0R402$
- $GP010400 \rightarrow GP020402 \rightarrow GP0R400 \rightarrow GP0R402$
- $GP010400 \rightarrow GP0R400 \rightarrow GP010402 \rightarrow GP020400$
- $GP0R402 \rightarrow GP0R400 \rightarrow GP010400 \rightarrow GP010402$

Snow

- $GP010607 \rightarrow GP0R0604 \rightarrow GP0R0606 \rightarrow GP0R122$
- $GP0R0606 \rightarrow GP010122 \rightarrow GP050176 \rightarrow GP0R0607$
- $GP0R0607 \rightarrow GP010122 \rightarrow GP0R0604 \rightarrow GP030176$
- $GP010607 \rightarrow GP030176 \rightarrow GP0R0604 \rightarrow GP0R0606$

Discussion. When we analyze the results in the Table C.3, on **Fog** and **Snow** sequences we can observe a behavior similar to the one that we have observed for the ACDC dataset obtained with the multi-environment multi-condition sequences (Table 1). On the other hand, **Night** and **Rain** sequences represent an interesting case study, with some discrepancies. For example, we can observe significant improvements when adapting with Style Transfer (ST) to Night sequences due to the fact that the ST can effectively increase the overall brightness and contrast of the image (see Fig. C.8), making it easier for the model to process such samples. Furthermore, note that due to a strong domain gap

ACDC results (multi environment / single condition)				
No adapt. (NA)	Sequence type			
	Fog	Night	Rain	Snow
	41.4 \pm 1.5	15.6 \pm 0.9	41.9 \pm 0.2	38.9 \pm 0.8
Method	Improvements			
<i>Style transfer</i>				
N-ST (NN)	−8.8% \pm 2.1	−3.3% \pm 0.5	−6.4% \pm 0.6	−2.8% \pm 0.6
N-ST (rand)	−13.9% \pm 1.3	+20.6% \pm 1.6	−9.9% \pm 0.5	−10.3% \pm 0.5
<i>Naive adaptation</i>				
N-BN	+2.7% \pm 0.8	+4.5% \pm 0.3	+1.6% \pm 0.2	+2.1% \pm 0.4
N-PL	+3.5% \pm 1.1	+5.1% \pm 0.3	+1.9% \pm 0.2	+2.7% \pm 0.4
N-TENT	+9.1% \pm 3.7	+5.5% \pm 0.7	+1.2% \pm 0.5	+7.1% \pm 1.2
<i>CL adaptation</i>				
C-BN	+8.4% \pm 5.1	+23.7% \pm 2.1	+0.0% \pm 0.8	+6.8% \pm 2.1
C-PL	+0.8% \pm 6.2	−41.2% \pm 21.1	−15.8% \pm 2.3	−12.2% \pm 6.3
C-TENT	+8.7% \pm 5.7	−8.1% \pm 2.0	−3.0% \pm 1.0	+4.6% \pm 2.1
<i>CL+SR adaptation</i>				
C-PL-SR	+8.2% \pm 2.6	−10.8% \pm 1.7	−2.0% \pm 0.9	+4.3% \pm 2.0
C-TENT-SR	+6.1% \pm 3.0	+0.0% \pm 3.2	+0.6% \pm 0.1	+3.5% \pm 1.3
<i>Adaptive-reset adaptation</i>				
Class-N-PL	+9.8% \pm 5.6	+25.0% \pm 2.2	+0.2% \pm 0.9	+7.3% \pm 2.0
Class-N-TENT	+9.4% \pm 6.0	+24.6% \pm 2.1	−0.4% \pm 0.9	+6.8% \pm 2.3
<i>Oracle-reset adaptation</i>				
Oracle-N-PL	+13.0% \pm 5.9	+32.2% \pm 2.7	+2.7% \pm 0.8	+10.4% \pm 1.9
Oracle-N-TENT	+13.6% \pm 6.4	+31.1% \pm 2.9	+1.6% \pm 0.8	+10.1% \pm 2.1

Table C.3. Results (relative performance gain in %) on ACDC sequences built with multiple cities but fixed weather/daylight conditions.

between GTA-5 (containing mainly day images) and images all over these sequences (all night images), the NA baseline yields to a poor performance even with DR $\uparrow\uparrow$. The con-

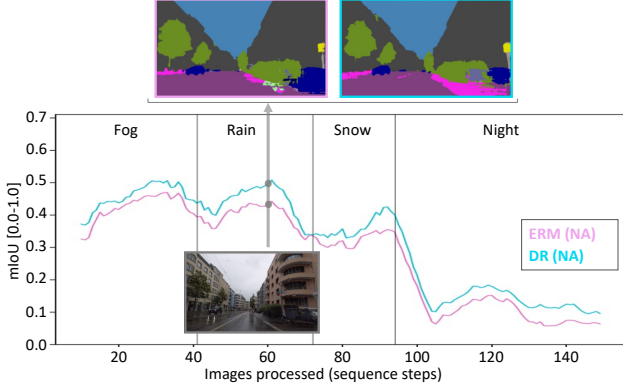


Figure C.3. **ERM vs DR** Performance evolution of ERM and DR non-adapted (NA) models, in pink and light blue, respectively – for one ACDC [65] sequence. In the plot, it can be observed that DR brings consistent improvements with respect to ERM. Top: qualitative comparison between predictions of the two methods when processing the reported image; notice the *sidewalk* on the bottom-right corner.

tinual learning methods C-PL and C-TENT fail in this condition, and SR does not carry the same improvements than it did in other setups. The best performing strategies are C-BN and the reset methods. For the **Rain** sequences, we observe a similar behavior to the one already observed for Cityscapes O. (Table 1). Starting from a strong DR $\uparrow\uparrow$ baseline model, none of the continual methods, excluding the Oracle-R approach, brings significant improvements, and the gain of Oracle-R over the baseline is relatively modest. Surprisingly, there is a significant improvement for the Fog sequence, a condition for which the NA model performs as well as for Rain. Properly understanding different condition-dependent behaviors represents an interesting research question for real-world applications.

C.5. Qualitative plots

We finally report additional qualitative plots, showing the performance evolution as in Figure 3 in the main paper. We report them in Figures C.3–C.6, extending Figure 3 from the main manuscript. Apart from models reported in Figure C.3, results reported in all other figures assume DR pre-training. We summarize the content of each figure below, and report the details in the different captions.

- **Figure C.3:** We compare models with and without DR [77] pre-training, in *light blue* and *pink*, respectively.
- **Figure C.4:** We compare models trained with and without BN [38] statistics online adaptation [53], in *blue* and *pink*, respectively.
- **Figure C.5:** We compare models trained via N-PL, C-PL and Class-R-PL, in *blue*, *orange* and *green*, respectively.

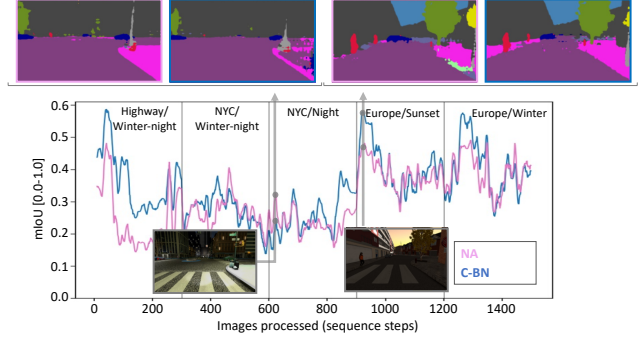


Figure C.4. **C-BN vs NA.** Performance evolution of a non-adapted model (NA) and a model whose BN statistics is updated continuously [53], in pink and blue, respectively – for one SYNTHIA [65] sequence. In the plot, it can be observed that C-BN generally brings consistent improvements with respect to NA. Top: qualitative comparison between predictions of the two methods when processing the reported image (on the left, we report failure cases of C-BN w.r.t. NA, from the limited number of frames in which C-BN underperforms.).

- **Figure C.6:** We compare models trained via N-PL, C-PL and C-PL-SR, in *blue*, *orange* and *red*, respectively.

Figure C.7 (top) provides a qualitative view on improvements led by adapting the model with C-TENT-SR (top-right) with respect to the non-adapted baseline (top-middle). In this specific example, the improvement in classifying the sidewalk’s pixels is very significant. Figure C.7 (bottom) provides qualitative evidence of “catastrophic forgetting”, which we discussed about in the main manuscript. When the model is trained continuously and without regularization, it can forget classes if it does not encounter them for a while. In this specific example, the C-TENT model has forgotten the pedestrian class almost completely (bottom-middle). By regularizing with a term that optimizes the cross-entropy loss on source samples, the C-TENT-SR model is continuously exposed to the source classes; hence, it does not forget about them (bottom-right).

D. Style transfer

One of the most popular approaches for unsupervised domain adaptation is to apply a photorealistic transformation to the source images (usually the training set) so that they resemble the *style* of the target images (usually the test or deployment data) [19, 52, 63, 73, 92]. However these methods assume that, even if unlabeled, both the source and target image distributions are known at train time. In this work, we are proposing a benchmark where test images are unknown a priori and we receive them as a continuous stream. Thus, we can not apply these techniques directly. On the other hand, a related line of work has focused on performing single-image style transfer: Given a *content* image and a *style* image, transform the content image to mimic

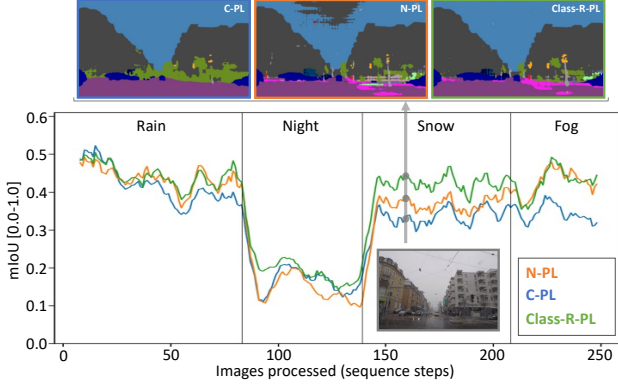


Figure C.5. **Naive vs Continual vs Reset.** Performance evolution of Naive, “Vanilla” continual, and Class-reset versions of PL (N-PL, C-PL and Class-R-PL, is orange, blue and green, respectively) – for one ACDC [68] sequence. In the plot, it can be observed that learning continuously without precautions results in sub-optimal performance, and that reset can allow maintaining a performance close to the naive counterpart in some parts of the sequence, while significantly improving over it in others. Top: qualitative comparison between predictions of different methods when processing the reported image; C-PL has catastrophically forgotten *sidewalks, pedestrians* and *poles* (best in color zooming in).

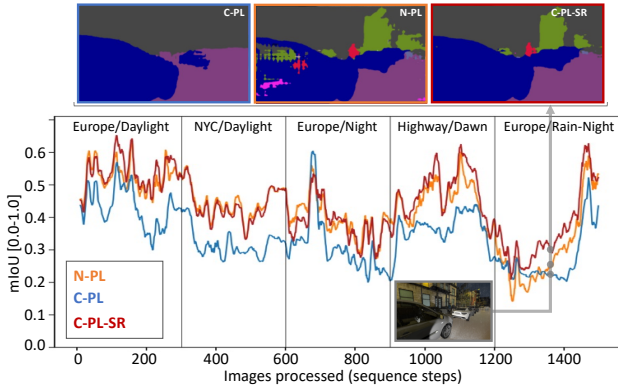


Figure C.6. **Naive vs Continual vs Source regularized.** Performance evolution of Naive, “Vanilla” continual, and Source-regularized versions of PL (N-PL, C-PL and Class-R-PL, is orange, blue and red, respectively) – for one SYNTHIA [65] sequence. In the plot, it can be observed that learning continuously without precautions results in sub-optimal performance, and that source regularization helps mitigating such negative impact, leading to performance often Naive. Top: qualitative comparison between predictions of different methods when processing the reported image; C-PL has catastrophically forgotten *pedestrians* and *vegetation* (best in color zooming in).

the high-level appearance of the style image. Several works in this area have focused on artistic style transfer, where they provide images with painting styles e.g. [28, 37, 40, 91]. However, when applying these methods between two images they create artifacts that yield unrealistic results. This motivated other works to focus on photorealistic style trans-

fer [4, 44, 51, 59, 98]. Although these works were generally motivated from an aesthetics perspective, we note photorealistic style transfer can be used to perform online unsupervised domain adaptation, where for each image received at test time we apply style transfer as a pre-processing step to mimic the appearance of *some* image from the training set. In our work, we use the method described in [98].

D.1. Experimental details

We use the public implementation⁸ from [98] with default parameters and stylize at all modules (encoder, decoder and skip connections). We do not use segmentation masks since the content image is unlabeled and using our prediction as a segmentation mask might induce unwanted artifacts. In our experiments, we apply style transfer to every test (or validation) image independently as a pre-processing step. For every content image (test or validation) we choose a corresponding style image (from the training set) to apply the style transfer. We compare two strategies:

Random selection: We select images from the training set uniformly, regardless of the appearance and semantic content of images. This selection method is indeed fast, however, it might pair style and content images which are very different and lead to somewhat artificial image styles.

Nearest neighbour selection: In this case, the idea is to match each content image with the closest style image. We compare images using the cosine similarity between the features extracted after applying a CNN encoder. We note that we can pre-compute the features of the training images offline. Moreover, we use the same encoder used for style transfer as feature extractor for further efficiency.

Regardless of the selection method, style transfer is a rather expensive procedure which takes of the order of 1-2 seconds per image on a GPU NVIDIA V100, depending on the image size. Therefore, in order to apply it in a real-time scenario, further work should be dedicated into speeding-up the style transfer process. In Figure C.8 we provide illustrative samples of content, style and stylized images for each dataset and sampling method.

E. Code and Streamlit web app

Our code to replicate the experiments provided in this work is attached to the submission, see the files in 2419_code.zip.

We report in Fig. E.9 a screenshot of the Web application we will release to explore results and models (the file used to generate it is `streamlit_app.py`). On the left, one can select the specific model and sequence; on the right, the results are reported; in the middle, the ground truth, image and predicted masks are reported – the selection can be made via the slider on the bottom-left. We find this app

⁸<https://github.com/clovaai/WCT2>

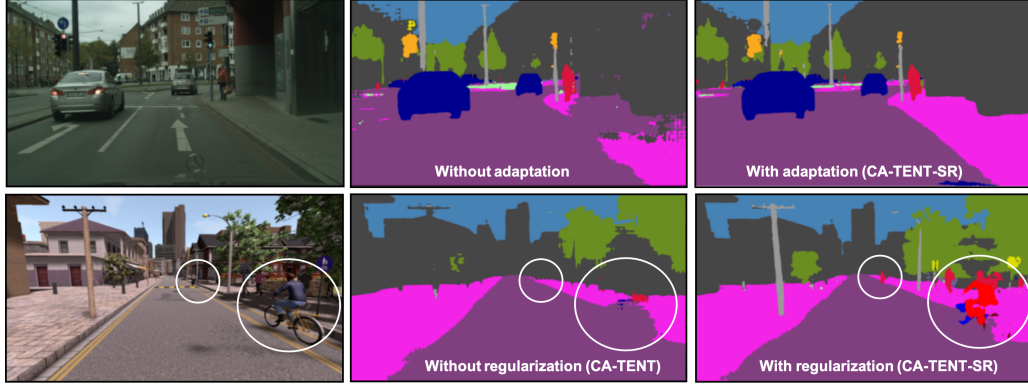


Figure C.7. *Top*: We compare the predictions of a non-adapted model (*top-middle*) and a model adapted via CA-TENT-SR (*top-right*). This image shows how the non-adapted model struggles in classifying the sidewalk’s pixels, while the adapted model improves in this regard. *Bottom*: We compare the predictions of models trained via two different versions of the TENT [87] algorithm, when provided with an image (*left*) CA-TENT (*bottom-middle*) and CA-TENT-SR (*bottom-right*). This comparison shows how models trained continuously but without regularization can catastrophically forget some classes – in this case, the pedestrian one.

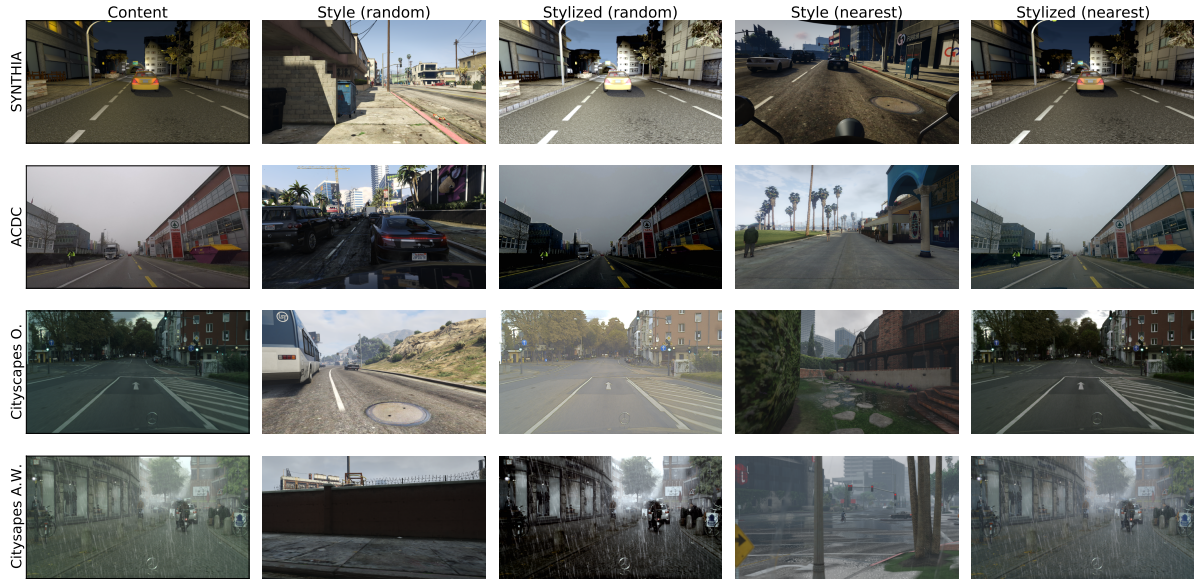


Figure C.8. Sample images generated with the style transfer baseline from each dataset. The original *content* image is associated either a random or a nearest neighbour *style* image from the GTA-5 dataset. Then we apply a style transfer method on the content-style pair to obtain a *stylized* image. Note how when choosing style images randomly we can obtain somewhat unrealistic stylized images while choosing them with a simple nearest neighbour search helps mitigate this issue.

very useful for research on semantic segmentation, where exploring qualitative results is as important as assessing final performance (*e.g.*, final mIoU values). The same app can be used to generate the plots shown in Fig. C.3–C.6

F. Limitations

We conclude by highlighting the limitations of ideas and methods detailed in this work.

For what concerns the OASIS benchmark we introduced (our core contribution), we tried, as much as possible, to mimic conditions that one may face when deploying a machine learning system in the real world – in particular, test-

ing on samples/sequences significantly different from the ones on which the models have been trained and validated. Yet, it is important to remember that the real world can expose our models to a variety of conditions significantly broader than the ones a benchmark can contain. Thus, it is important to avoid having a false sense of security before deploying a system in the real world; for example, if we consider an outdoor robot, there may be combinations of weather/visual conditions and urban environments, not considered in the benchmark, that might significantly alter its performance.

For what concerns the methodology, it is important to

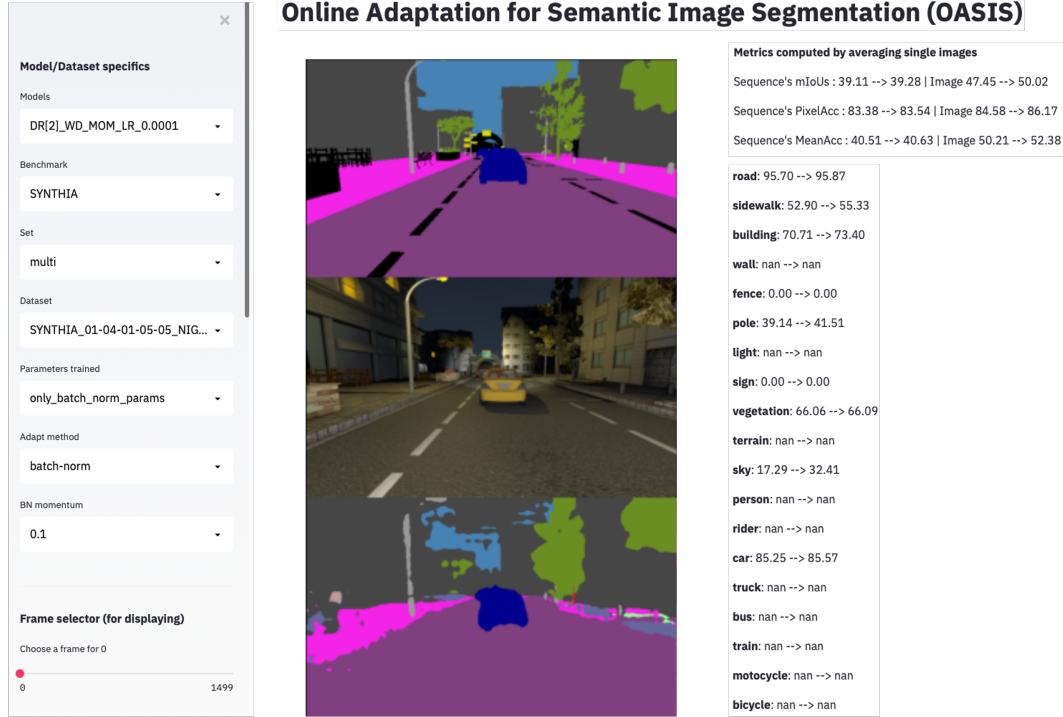


Figure E.9. Screenshot of our Streamlit web app.

gain familiarity with the failure cases of unsupervised domain adaptation. As we tried to convey with the experiments reported in Table C.1, the starting point is extremely important to foster good adaptation results; if the pre-trained model is significantly under-performing in some conditions, domain adaptation can hardly improve on such performance (in some cases, it can even deteriorate the performance even more). Finally, concerning the implications of continual learning – and, more specifically, of continual unsupervised adaptation – catastrophic forgetting is a severely limiting factor. While in Sec. 4.3 we have proposed a family of methods based on a reset mechanism; this is a very simple heuristic, far from solving a very broad research problem. Still, we hope that raising these limitations will encourage the community to consider the problem and devise more advanced solutions to tackle it.