Learning With Style: Continual Semantic Segmentation Across Tasks and Domains

Marco Toldo, *Student Member, IEEE*, Umberto Michieli, *Graduate Student Member, IEEE*, and Pietro Zanuttigh, *Member, IEEE*

***Abstract*—Deep learning models dealing with image understanding in real-world settings must be able to adapt to a wide variety of tasks across different domains. Domain adaptation and class incremental learning deal with domain and task variability separately, whereas their unified solution is still an open problem. We tackle both facets of the problem together, taking into account the semantic shift within both input and label spaces. We start by formally introducing continual learning under task and domain shift. Then, we address the proposed setup by using style transfer techniques to extend knowledge across domains when learning incremental tasks and a robust distillation framework to effectively recollect task knowledge under incremental domain shift. The devised framework (LwS, Learning with Style) is able to generalize incrementally acquired task knowledge across all the domains encountered, proving to be robust against catastrophic forgetting. Extensive experimental evaluation on multiple autonomous driving datasets shows how the proposed method outperforms existing approaches, which prove to be ill-equipped to deal with continual semantic segmentation under both task and domain shift.**

***Index Terms*—Continual learning, domain adaptation, semantic segmentation.**

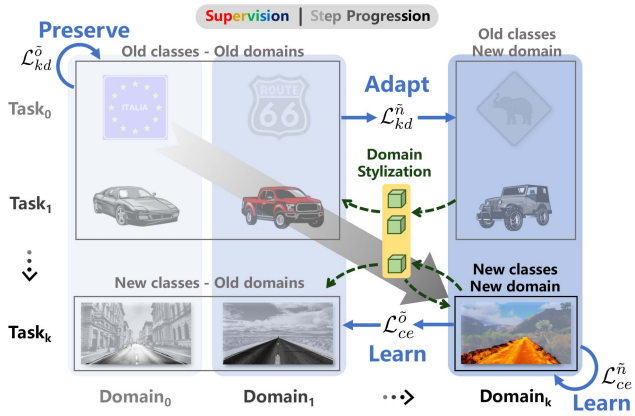

Fig. 1. High-level view of our approach. Transparency decrease (top → down and left → right) indicates progression through learning steps. Colored task icons denote presence of supervision within training data, grayscale ones signal lack of supervision. At each step, we leverage training data to *learn* new classes on the new domain. Domain stylization allows to reiterate old-domain distribution, crucial to *learn* new tasks and *preserve* old ones on former domains, and to *adapt* old-domain old-task knowledge to new domains.

I. Introduction

With the recent rise of deep learning, the computer vision field has witnessed remarkable advances. Challenging tasks, such as image semantic segmentation, are nowadays successfully addressed by well-established deep learning architectures [1], [2], [3]. Nonetheless, the fundamental problem of continuously learning and adapting to novel environments remains open and is actively investigated, with a long way before its definitive solution.

Although capable of remarkable performance in narrow and confined tasks, deep models tend to struggle when confronted with continual learning of dynamic tasks in ever-changing environments. A major issue stands in the tendency to *catastrophically forget* previously acquired knowledge [4], with new information erasing that experienced so far. Furthermore, variable input distribution between supervised training data and target data has been shown to cause performance degradation, giving rise to the need for *domain adaptation*, which targets knowledge transferability across domains. Both constitute critical problems when it comes to deploying deep models in practical applications, as in the real world it is very likely to face distribution variability both in terms of input data and of target tasks.

A thriving research endeavour has been devoted to continual learning (also referred to as incremental learning, IL, or lifelong learning [5]) in vision problems, such as image classification [4], [6], [7], object detection [8], [9], [10] and, more recently, semantic segmentation [11], [12], [13]. The majority of those works, however, are limited to a *class incremental* perspective of the continual learning problem, where the focus is strictly posed on the variable task (e.g., class) supervision and label-space shift experienced throughout the learning process. On the other side, a significant research effort has been directed toward the domain adaptation problem, ranging from a static learning setting [14], [15], [16] to, quite recently, a dynamic perspective [17], [18], [19], taking into account incremental changes in the data distribution.

Nonetheless, the general continual learning problem across both tasks and domains is yet unexplored for the semantic segmentation task. Where class incremental methods usually

Manuscript received 9 September 2022; revised 10 December 2023; accepted 20 April 2024. Date of publication 7 May 2024; date of current version 3 October 2024. This work was supported by the Italian Ministry for Education (MIUR) under the *Departments of Excellence* initiative Law under Grant 232/2016. Recommended for acceptance by G. Farinella. *(Corresponding author: Marco Toldo.)*

The authors are with the Department of Information Engineering, University of Padova, Padova 35131, Italy (e-mail: marco.toldo@dei.unipd.it; umberto.michieli@dei.unipd.it; zanuttigh@dei.unipd.it).

The code is available at https://medialab.dei.unipd.it/paper_data/LwS.

Digital Object Identifier 10.1109/TPAMI.2024.3397461

struggle to cope with domain knowledge transferability, domain incremental methods lack predisposition to address incremental task supervision. We instead propose to tackle continual semantic segmentation with joint incremental shift along class and domain directions. The training process involves multiple steps, each of which carries a new set of classes to learn, along with a training set comprising image samples with a step-distinctive distribution, differing from those experienced in previous steps, and supervision available only on the newly introduced class set. The overall objective is for the incremental segmentation model to deliver satisfactory performance across all the tasks (i.e., class sets) and domains encountered so far, with the class- and domain- wise joint training as the target upper bound.

In this novel problem setup (see Fig. 1), both domain adaptation and recollection of past classes must be performed to achieve satisfactory performance. Under the domain incremental angle, it is required to simultaneously learn new classes over past domains and adapt old-class knowledge to the new domain. From the class incremental perspective, recollection of past knowledge must take into account the variable input distribution characterizing the addressed incremental learning problem.

We therefore devise multiple training objectives to face underlying sub-problems. While to rehearse knowledge of old classes we resort to the old-step segmentation model, which is a common practice among class incremental learning methods [11], to replay information of past-domain input distribution we propose a stylization mechanism. The average style (i.e., a very compact representation) of each encountered domain is computed and stored in a memory bank, to be transferred to novel domains in future steps and reproduce some domain-level information.

The overall optimization framework is made of (i) a standard task loss (i.e., cross-entropy objective) to learn new classes over available training data, (ii) an additional task loss instance to learn new classes in old domains by leveraging stylization, (iii) a knowledge distillation-like objective to infuse adapted information of past classes in the form of hard pseudo-labels to the new domain and finally (iv) an output-level knowledge distillation objective applied on stylized images to retain old-domain old-class performance.

To summarize, our contributions are as follows:

- We investigate a novel comprehensive incremental learning setting that accounts for variable distribution within both input and label spaces.
- We develop a framework to tackle all facets of the class and domain incremental learning problem, based on a stylization mechanism to recall domain knowledge under incremental task supervision and a robust distillation framework to retain task knowledge under incremental domain shift.
- We devise novel experimental setups to simulate the proposed learning setting and conduct an extensive evaluation campaign.
- We show that the proposed method outperforms existing state-of-the-art methods that address the IL problem only from a class or a domain incremental perspective without increasing the computation time at the inference stage.

II. Related Works

Semantic Segmentation: Under the impulse of deep learning, semantic segmentation has witnessed a considerable advance in recent years [20]. Since the introduction of fully convolutional networks (FCNs) [1], which introduced the popular encoder-decoder architecture, huge research efforts have improved the state of the art. Dilated convolutions [2], [21] allow to retain sufficiently large receptive fields limiting the growth in model size. Spatial [22] and feature [23] pyramid pooling extract and aggregate contextual information at different scales to acquire enriched representation for improved dense predictions. At the same time, considerable interest was devoted to the design of lightweight architectures for practical applications typically burdened by strict hardware constraints. MobileNet architectures [24], [25] are built upon the efficient depthwise separable convolution. ErfNet [3] resorts to factorized residual layers to provide real-time accurate segmentation. Recently, transformers have been applied in vision, even for dense prediction tasks such as semantic segmentation [26].

Class Incremental Learning (CIL): Continual learning in the form of incremental classification tasks has been subject of growing research interest in the recent past [5]. Extensive literature can be found targeting image classification [4], [6], [7], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36] and object detection tasks [8], [9], [10], [37] under the incremental learning paradigm. Many of these works [7], [28], [29], [33], [34], [35], [36] rely on exemplars, i.e., a small portion of training data is stored to be replayed in future steps. We instead place ourselves in a totally exemplar-free setup. Among the exemplar-free methods [4], [6], [8], [9], [10], [30], [31], [32], [37] we can identify regularization-based [4], [10], [37], rehearsal-based [6], [8], [9], [30], [31] and structure-based [32]. Even if many works propose techniques which could in principle be generalized to various vision tasks (such as the prosperous knowledge distillation mechanism [6], [8], [38]), when facing the semantic segmentation task, additional complexity, which is not present in case of whole-image classification or object detection, arises [39].

More limited literature can be found for incremental semantic segmentation [11], [12], [13], [40], [41], [42], even though this field has experienced a very recent rise in research consideration [43], [44], [45], [46], [47]. A first direction of study has been oriented toward the adaptation of the knowledge distillation mechanism to incremental semantic segmentation [11], [12], [13], [40], [43], [44], [47]. Michieli et al. [11], [48] have been the first to introduce this technique in CIL for dense classification, proposing both feature- and output- level variants of the distillation objective. In [12] authors address the semantic shift of background regions by proposing a novel distillation formula. Furthermore, [13] improves feature-level distillation by pooling representations to capture spatial relationships. Phan et al. [47] introduce a measure of task similarity as a weighting factor in the distillation objective. Yang et al. [44] resort to a structured self-attention approach for preserve relevant knowledge. Finally, [43] extends the popular contrastive learning paradigm to incremental semantic segmentation to improve class discriminability in the

feature space. Nonetheless, none of the aforementioned works address the distribution shift that could be present across tasks within the input space. We propose to use a distillation objective which is robust to domain incremental gaps, and targets the preservation of old-task knowledge both on the current domain, by distilling through robust hard pseudo-labels, and on the past domains, by leveraging domain stylization to distill knowledge when experiencing old-domain input statistics. Targeting semantic discriminability of latent representations, a clustering-based objective built upon class prototypes is proposed in [42]. Maracani et al. [41] introduce a novel rehearsal approach based on the retrieval of training samples by external sources, i.e., via GAN-based generation or web-crawling. Cermelli et al. [45] further show that it is possible to perform continual training with only image-level annotations in incremental steps and reach high accuracy in some CIL experimental setups. Nonetheless, this approach could be susceptible to the amount of dense supervision provided in the first learning step, and might not scale well to segmentation of images containing objects of different classes. Zhang et al. [46] devise a dynamic incremental framework to decouple the representation learning of old and new tasks. All the aforementioned works assume statistical homogeneity across learning steps in terms of input data distribution. On the other hand, we address the more realistic setup with both input and label spaces undergoing incremental shifts, and we show the superiority in this generalized setup of the proposed incremental approach compared to pure CIL competitors.

Domain Adaptation (DA): Deep models are known to suffer performance degradation when presented with varying input distribution between training and testing phases [49], [50]. Domain adaptation has been extensively investigated to alleviate the aforementioned problem, by safely transferring learned knowledge from label-abundant source domains to label-scarce, or even unsupervised, target ones. Particularly flourishing has been unsupervised domain adaptation (UDA) for the semantic segmentation task [14], [15], [16], [51], [52], [53], as supervision in terms of dense segmentation maps is usually very costly and time expensive to be collected for real-world data. In the standard UDA setting, the task at hand is the same on both source and target domains, while we address a more realistic setup with dynamic task and domain evolution.

More recently, different variations of the static DA have been proposed, relaxing some of the original strict assumptions. One research direction involves distinct tasks between source and target domains, i.e., allows source and target classes to be different. Depending on the relationship between source and target class sets, partial [54], open-set [55] and universal [56], [57] domain adaptation setups have been proposed, even though most research has been confined to the image classification problem [55], [56], [57]. Moreover, these works do not involve class incremental learning, as adaptation is performed with simultaneous access to source and target domains in a single learning phase.

Another line of work has explored diverse setups in terms of domain availability. Some propose to handle multiple source [58], [59] or target [17], [18], [19], [60], [61], [62], [63] domains. This can involve a single adaptation phase [58], [59], or multiple phases where different domains are experienced in different learning steps in an incremental fashion (but still with a fixed class set) [17], [18], [19], [62], [63], in fact, undertaking continual learning under the domain adaptation perspective. Garg et al. [64] develop a multi-domain incremental learning (MDIL) framework that involves classification tasks shifting across multiple domains experienced in an incremental fashion, but the class sets are not disjoint in the incremental steps.

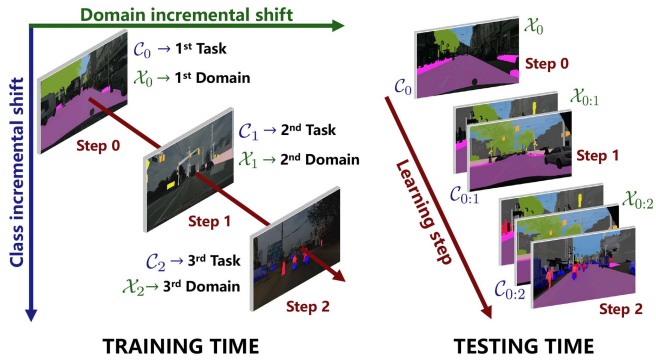

Fig. 2. Overview of the class and domain incremental setup. At each step, training data come from a new domain and is labeled on a new class set. When testing, performance is measured on all domains and classes experienced so far.

Joint CIL and DA: A very few works address both task incremental learning and domain adaptation. Kalb et al. [65] discuss class and domain incremental learning, but each task is tackled individually by evaluating standard CIL and DA methods. In [66] coarse-to-fine continual learning is explored, but the proposed setup does not involve domain shift across learning steps, as source and target domains are kept fixed. Recently, Simon et al. [67] address continual learning with tasks and domains dynamically evolving. Still, they assume to have task supervision on all the considered domains at each task incremental step, which may not be a realistic assumption in real-world applications. In addition, rehearsal of training exemplars is performed, and the method specifically targets image classification.

III. Problem Setup

In semantic segmentation we aim at labeling every individual spatial location of an image by associating it with a semantic class taken from a predefined collection of candidates $\mathcal{C}$. That is, given an RGB image $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^{H \times W \times 3}$, a segmentation network $S : \mathcal{X} \mapsto \mathcal{Y}$ is exploited to provide its segmentation map $\hat{\mathbf{Y}} \in \mathcal{Y} \subset \mathcal{C}^{H \times W}$. $\hat{\mathbf{Y}}$ should be an accurate prediction of the ground truth map $\mathbf{Y}$, which is available only at training time.

We follow an incremental learning protocol to optimize the segmentation network, as depicted in Fig. 2. Specifically, the predictor is trained in multiple steps $t = 0, \ldots, T-1$ to recognize a progressively increasing set of semantic classes. At step t, a new class set $\mathcal{C}_t$ is introduced, along with training data $\mathcal{D}_t = \{(\mathbf{X}_t, \mathbf{Y}_t)\} \subset \mathcal{X}_t \times \mathcal{Y}_t$ associated to that set, which is available on the current image domain $\mathcal{X}_t$. The supervision provided by $\mathcal{D}_t$ is restricted to $\mathcal{C}_t$, meaning that any pixel within $\mathcal{D}_t$ is tagged in $\mathcal{Y}_t$ with $c \in \mathcal{C}_t$. At the end of the step, all the

currently accessible data is discarded and is not reused again. The procedure is reiterated for multiple learning steps, with a new domain $\mathcal{X}_t$ and class set $\mathcal{C}_t$ being introduced and used for training at each step.

More formally, the objective is to train $S_t : \mathcal{X}_{0:t} \mapsto \mathcal{Y}_{0:t}$

- to recognize all the semantic classes observed up to the current step t:

$$\mathcal{Y}_{0:t} \in \mathcal{C}_{0:t}^{H \times W}, \quad \mathcal{C}_{0:t} = \bigcup_{k=0}^{t} \mathcal{C}_k, \tag{1}$$

- on all the image domains experienced so far:

$$\mathcal{X}_{0:t} = \bigcup_{k=0}^{t} \mathcal{X}_k. \tag{2}$$

We remark that $\{\mathcal{X}_t\}_{t=0}^{T}$ are characterized by diverse statistical properties, i.e., domain shift occurs between them, typically manifested through cross-domain variable visual appearance of scene elements that yet share semantic significance. All $\mathcal{C}_t$ are disjoint sets, except for the *unknown* (u) class, which belongs to each of them. Class u at step t contains all the past and future classes. In other words, u undergoes a semantic shift across subsequent steps and, for this reason, demands special care when being handled [12].

IV. Overview of the Proposed Method

We concurrently face challenges peculiar to both the domain adaptation and the class incremental learning settings.

Domain Adaptation: The segmentation network is trained on data from multiple domains, each holding only a subset of the whole set of the semantic classes. Even so, the model is expected to provide satisfactory prediction performance on all the observed domains and semantic classes.

Class Incremental Learning: The different class supervision available on different domains leads us to a class incremental problem, where semantic categories come across in a continual fashion. Therefore, we are required to address the widely known catastrophic forgetting phenomenon [4], aiming at preserving knowledge from past classes when learning new ones. However, unlike standard CIL, knowledge preservation has to be performed differently depending on the domain in which it is applied.

Hence, it is necessary to transfer knowledge across incremental steps and domains to:

i) learn *new-class* clues shared across the current (supervised) domain and the past ones (where new-class supervision was not available during past steps);
ii) adapt *old-class* knowledge learned in former domains to the novel domain (accounting for the semantic shift within the input space).

We break down the domain shift and class continual learning problems into simpler underlying sub-problems, as indicated above. Our overall learning framework builds upon multiple individual objectives, each focusing on a specific challenge enclosed in the general setup. We simultaneously progress along class and domain incremental directions; at each learning step, after the first one, both classes and domains experienced so far can be arranged into *new* or *old* types, according to whether they are currently available or not. More in detail, we propose a specific learning objective for each of the different combinations of domain and class types (see Table I and Fig. 3), i.e., to:

i) learn new classes on new domains (Section V-A);
ii) learn new classes on old domains (Section V-B);
iii) adapt old-class information to new domains (Section V-C);
iv) preserve old-class information in old domains (Section V-D).

TABLE I
Training Objectives: the *n/o* superscripts Denote the Use of new/old Domain Data, With ˜ Implying Stylization

	New Domain	Old Domains
New Classes	$\mathcal{L}_{ce}^{\tilde{n}}$	$\mathcal{L}_{ce}^{\tilde{o}}$
Old Classes	$\mathcal{L}_{kd}^{\tilde{n}}$	$\mathcal{L}_{kd}^{\tilde{o}}$

A. Domain Stylization

We resort to a style transfer mechanism to recreate image data with statistical properties resembling those of past domains. More specifically, starting from the available image data originating from the input domain accessible at the current step, we transfer the styles extracted from all the previously encountered domains. By doing so, a stylized version of each of the former domains is produced, with image content derived from the novel dataset.

The benefits that originate from domain stylization are manifold: (i) We force the prediction model to experience past input distributions under supervision or pseudo-supervision, tackling domain-level catastrophic forgetting. (ii) We aim at learning new classes on old domains, where supervision was not available when they were directly observed. At the same time, we propose to preserve old-class knowledge on old domains, counteracting class-level catastrophic forgetting. (iii) By encountering a variegate input distribution, the predictor is encouraged to develop the ability to generalize to unseen domains, which is crucial in a continual learning paradigm that involves domain shift.

The style transfer mechanism we adopt is inspired by [16] and involves low computational cost and memory requirements. We also tested alternative options, but they led to lower results. The original algorithm works in the Fourier transform domain: the low frequency portion of the amplitude of the spectral representation from a target image (i.e., the style) is extracted and applied to replace that of a source image (i.e., the content), whose phase component is kept unchanged. The outcome is image data with source semantic information, and target-like low-level appearance.

We enhance the original method to accommodate for the further complexity brought in by the class and domain incremental setting. From each image of the currently available dataset, we extract its style tensor (i.e., the amplitude central window), and

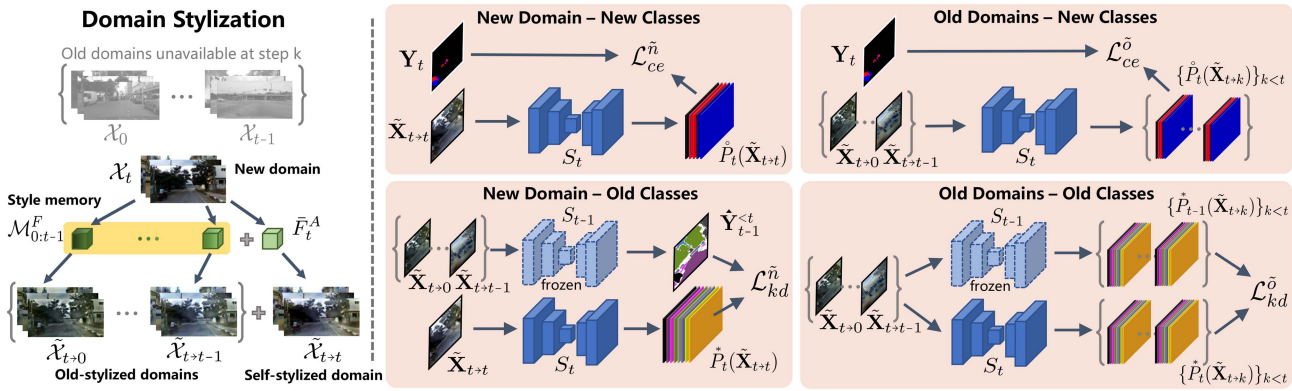

Fig. 3. Model architecture: we decompose class and domain IL into simpler sub-problems, each addressed by a suitable objective (4 panels in the right side); to access no longer available old domain data, we resort to stylization (left side).

we average it over all the samples:

$$\bar{F}_t^A = \frac{1}{|\mathcal{D}_t|} \sum_{\mathbf{X} \in \mathcal{D}_t} \mathcal{F}^A(\mathbf{X})[W_\beta], \tag{3}$$

where $\mathcal{F}^A(\mathbf{X})$ is the amplitude obtained by the FFT applied to image $\mathbf{X}$, and W_β is the style window. By doing so, we are extracting significant knowledge of *domain-dependent* statistical properties, condensed in a compact representation. The domain-specific style $\bar{F}_t^A$ of step t is stored in an incrementally-filled memory bank $\mathcal{M}_{0:t-1}^F = \{\bar{F}_k^A \mid k < t\}$ and preserved across steps. By leveraging the proposed storage mechanism, at each incremental step we can access crucial information of past domain low-level properties (yet minimal if compared to that contained in whole training sets), without requiring direct access to raw image data, which would violate the exemplar-free assumption. We stress that domain shift affects low-level details, while high-level semantic content is mostly shared across domains (e.g., the road serves the same purpose regardless of the dataset, while its appearance in terms of texture or pavement material might vary considerably). To create an oldly-stylized dataset at step t looking back at step $k < t$ (i.e., $\tilde{\mathcal{X}}_k^t$), for each image of the current domain we replace its amplitude window with that of the selected former domain as follows:

$$\tilde{\mathcal{X}}_{t \to k} = \{\mathcal{F}^{-1}([\bar{F}_k^A + \mathcal{F}^A(\mathbf{X})[W_\beta^c], \mathcal{F}^P(\mathbf{X})]) \mid \mathbf{X} \in \mathcal{X}_t\}, \tag{4}$$

where $\mathcal{F}^{-1}$ is the inverse FFT operator and $\mathcal{F}^P(\mathbf{X})$ is the Fourier phase component of $\mathbf{X}$. In addition, we devise a self-stylization mechanism by self-applying domain style to improve generalization toward future steps, promoting forward transfer. As for the dimension of the style window, we experimentally found that the β parameter as defined in [16] (i.e., the parameter controlling the window size) provides satisfactory and robust results when set to $1\mathrm{e}{-2}$.

Finally, we stress that our approach is independent of the style transfer technique used, provided that style information and content can be extracted in two distinct steps.

V. Learning Across Tasks and Domains

A. Learning New Classes Over New Domains

In the proposed class and domain continual learning framework, direct supervision comes uniquely for the newly introduced class set $\mathcal{C}_t$ and image domain $\mathcal{X}_t$ in the form of the training dataset $\mathcal{D}_t \subset \mathcal{X}_t \times \mathcal{Y}_t$. As mentioned before, image pixels not belonging to $\mathcal{C}_t$, i.e., of past or never seen classes, are assigned to a special class *unknown*, whose semantic statistical properties are highly dynamic.

To account for the semantic shift suffered by the *unknown* class at the current step $t > 0$ w.r.t. previous steps, we group the past and unknown class probability channels as follows:

$$\mathring{P}_t(\mathbf{X})[x, y, c] = \begin{cases} P_t(\mathbf{X})[x, y, c], & \text{if } c \neq u \\ \sum_{c' \in \mathcal{C}_{0:t-1}} P_t(\mathbf{X})[x, y, c'], & \text{if } c = u \end{cases} \tag{5}$$

where $P_t(\mathbf{X}) \in \mathbb{R}^{H \times W \times |\mathcal{C}_{0:t}|}$ is the output of S_t prior to the argmax when a generic image $\mathbf{X} \in \mathcal{X}$ is given as input.

We additionally define $\tilde{\mathcal{D}}_{t \to t} \subset \tilde{\mathcal{X}}_{t \to t} \times \mathcal{Y}_t$ as the *self-stylized* training dataset at step t, where the average style (defined above in Section IV) of the current image domain has been applied on top of the $\mathcal{X}_t$ domain itself.

To learn the newly introduced classes over the new domain we optimize:

$$\mathcal{L}_{ce}^{\tilde{n}}(\mathcal{C}_t, \mathcal{X}_t) = -\frac{1}{|\tilde{\mathcal{D}}_{t \to t}|} \sum_{\tilde{\mathbf{X}}, \mathbf{Y} \in \tilde{\mathcal{D}}_{t \to t}} \mathbf{Y} \cdot \log \mathring{P}_t(\tilde{\mathbf{X}}), \tag{6}$$

where we leverage input data with current style and supervision over the new class set. The $\tilde{n}$ superscript indicates the use of self-stylized data on the *new* domain. The purpose of self-stylization is twofold; first, it provides additional robustness and generalization capability to the prediction model, since input data is supplied with more homogeneous low-level statistic across individual samples. Second, it forces the prediction model to experience domain statistics that will be stored and replayed in the future, acting as proxies for the no longer available previous domain statistics.

B. Learning New Classes Over Past Domains

To compensate for the lack of available input data for past domains, we generate proxy datasets retaining low-level statistics resembling those of past domains. More precisely, for each style $\bar{F}_k^A \in \mathcal{M}_{0:t}$ of step $k < t$ we build $\tilde{\mathcal{D}}_{t\to k} \subset \tilde{\mathcal{X}}_{t\to k} \times \mathcal{Y}_t$ (as detailed in Section IV), i.e., an *oldly-stylized* training dataset at step t, for which domain-specific visual attributes of step $k < t$ has been applied on domain $\mathcal{X}_t$.

Supervision on the newly introduced classes over the old domains is exploited by optimizing:

$$\mathcal{L}_{ce}^{\tilde{o}}(\mathcal{C}_t, \mathcal{X}_{0:t-1}) = -\frac{1}{t}\sum_{k=0}^{t-1}\frac{1}{|\tilde{\mathcal{D}}_{t\to k}|}\sum_{\tilde{\mathbf{X}},\mathbf{Y}\in\tilde{\mathcal{D}}_{t\to k}} \mathbf{Y}\cdot\log \mathring{P}_t(\tilde{\mathbf{X}}), \tag{7}$$

where we leverage input data with past styles (i.e., with distributions supposedly close[1] to those of no longer available former domains) and the supervision over the new class set. The superscript $\tilde{o}$ indicates the use of *oldly*-stylized data.

By concurrently learning the segmentation task at the present step over an augmented pool of input data distributions from the past, the prediction model should learn more general and shareable clues, overcoming the domain shift inherent in the domain continual learning paradigm.

C. Adapting Old Classes to New Domains

In the addressed class incremental learning scenario, at each new learning step all past class sets are assumed to lack any direct supervision. To recall previously acquired knowledge, we resort to the well-known knowledge distillation objective [38]. Yet, differently from the standard class incremental learning problem as traditionally formalized in the literature [7], we expect to encounter additional challenges:

i) the input data of past domains (i.e., experienced by the segmentation model when previous class sets were learned) are no longer available;
ii) a distribution shift separates the current image data to that available at former steps. Thus, we no longer have access to data distributed as that experienced by the segmentation model saved from the past step, which, in principle, should be leveraged to distill knowledge of old classes.

To replicate the image distribution of data of past steps, we resort to the stylization mechanism (Section IV). Specifically, for each old domain $\mathcal{X}_k$, $k<t$, we build an oldly-stylized dataset $\mathcal{D}_{t\to k}$ starting from that of the current step t.

To access a form of supervision over the past classes we make use of pseudo-labeling via the prediction model from the previous step, which should retain profitable knowledge on the semantic categories learned so far. However, said model might not distill knowledge effectively when fed with input data of an unseen distribution, i.e., originating from the newly introduced domain. Therefore, we exploit oldly-stylized data to enhance pseudo-labeling by mitigating domain shift. We denote with $P_{t-1}^k(\tilde{\mathbf{X}}) \subset \mathbb{R}^{H\times W\times|\mathcal{C}_{t-1}|}$, $\tilde{\mathbf{X}} \in \mathcal{X}_{t\to k}$, the classification probability map from model S_{t-1} over new domain images with the style of step k. We then compute pseudo-labels following:

$$\hat{\mathbf{Y}}_{t-1}^{\mathcal{K}}[x,y] = \underset{c\in\mathcal{C}_{0:t-1}}{\operatorname{argmax}} \max_{k\in\mathcal{K}} P_{t-1}^k(\tilde{\mathbf{X}})[x,y], \tag{8}$$

where we leverage old model predictions over past styles, i.e., we set $\mathcal{K}=\{0,\dots,t-1\}$, while $\max_{k\in\mathcal{K}} P_{t-1}^k(\tilde{\mathbf{X}})[x,y]$ indicates that for each spatial location (x,y) we take the probability vector associated to the style with maximum peak value. We then refine the generated pseudo-labels at each spatial location (we will shorten $\hat{\mathbf{Y}}_{t-1}^{\mathcal{K}=\{0,\dots,t-1\}}$ as $\hat{\mathbf{Y}}_{t-1}^{<t}$ and drop the term $[x,y]$ for ease of notation) as:

$$\overset{\blacktriangle}{\mathbf{Y}}_{t-1}^{<t} = \begin{cases} \hat{\mathbf{Y}}_{t-1}^{<t}, & \text{if } \hat{\mathbf{Y}}_{t-1}^{<t} \text{ confident } \wedge\, \mathbf{Y}_t = u \\ u, & \text{if } \mathbf{Y}_t \neq u \\ \text{ignore}, & \text{elsewhere} \end{cases} \tag{9}$$

where $\mathbf{Y}_t \in \mathcal{Y}_t$. The hard pseudo-label $\hat{\mathbf{Y}}_{t-1}^{<t}[x,y]$ (i.e., after the argmax operation in (8)) is considered to provide a confident prediction if the peak probability value (of the probability map prior to the argmax) is bigger than a threshold τ, or if that value is among the top-K fraction of highest peaks for class $c=\hat{\mathbf{Y}}_{t-1}[x,y]$. We set $\tau=0.9$ and $K=0.66$ as advised in [16]. In addition, we leverage the ground-truth supervision on new classes to correct noisy estimations in pseudo-labels, by marking as *unknown* (i.e., *u*) all the pixels of newly introduced categories. We remark that the employed knowledge distillation is designed to provide insight on previous tasks (where current new classes were assigned to the *u* class), whereas we entrust (6) to instill understanding of the novel task. We experimentally verify that using separate objectives to train on new and old classes leads to improved results, as it forces the model to learn to better discriminate between different incremental class sets, part of which might coexist under the same *unknown* group for one or more learning steps. This is especially true for autonomous driving datasets, where each image can contain several semantically diverse elements, for all of which we may not have supervision from the start of the training.

To infuse adapted information about past classes at the current step without direct access to ground-truth information, we resort to the following objective:

$$\mathcal{L}_{kd}^{\tilde{n}}(\mathcal{C}_{0:t-1}, \mathcal{X}_t) = -\frac{1}{|\mathcal{D}_t|}\sum_{\tilde{\mathbf{X}}\in\mathcal{D}_t} \overset{\blacktriangle}{\mathbf{Y}}_{t-1}^{<t}\cdot\log \overset{*}{P}_t(\tilde{\mathbf{X}}), \tag{10}$$

by which we distill knowledge of past tasks (i.e., recognition of classes in $\mathcal{C}_{0:t-1}$) over the new domain $\mathcal{X}_t$ via the pseudo-labels derived from the old model S_t.

To account for the semantic shift suffered by the *unknown* class of step $t-1$ when moving to a new step $t>0$, we group *new* and *unknown* class probability channels as follows:

$$\overset{*}{P}_t(\mathbf{X})[x,y,c] = \begin{cases} P_t(\mathbf{X})[x,y,c], & \text{if } c \neq u \\ \sum_{c'\in\mathcal{C}_t} P_t(\mathbf{X})[x,y,c'], & \text{if } c = u \end{cases} \tag{11}$$

where $\overset{*}{P}_t(\mathbf{X}) \in \mathbb{R}^{H\times W\times|\mathcal{C}_{0:t-1}|}$. We opt for the use of hard-labels in place of the more common soft-labels in the distillation-like

[1]The *closeness* depends on what the style transfer mechanism is able to transfer in terms of statistical properties. The distribution gap is reduced in terms of low-level properties, while the semantic high-level distribution should already be similar across domains.

loss in order to prevent enforcing an uncertain behavior to S_t. This behaviour could be originated by the mismatch between training and inference input distribution undergone by the old model S_{t-1}, which has been trained over past domains and now is fed with new domain data (the oldly-stylizing operation reduces domain shift but has no guarantees on its complete removal). Experimental data on the pseudo-labeling strategy is provided in Section VIII-B.

D. Preserving old Classes on old Domains

In Section V-C we focused on distilling old-task knowledge on the current novel domain. Nonetheless, our ultimate target is to end up with a segmentation network capable to recognize all the observed classes over all the experienced domains, that is a prediction model robust to both domain and label distribution shifts. For this reason, at every novel incremental step it is required to preserve the task knowledge acquired in the past, that is, on past classes over past domains. To do so, we leverage the output-level knowledge distillation objective in its standard formulation [38], where we force a student model (i.e., the current model) to mimic the predicted classification probability distribution of a teacher model (i.e., the model saved and kept frozen since the end of the previous step). We opted for the objective in its standard fashion [38], as both image and label distributions ideally originate from previous steps, so no domain shift should, in principle, affect the distillation process. In practice, we can not access former incremental datasets. Therefore, to retrieve the missing old-domain data, we resort once more to stylization (Section IV), so that we can leverage oldly-stylized data as proxy for the missing original images. The final objective is of the following form:

$$\mathcal{L}_{kd}^{\tilde{o}}(\mathcal{C}_{0:t-1}, \mathcal{X}_{0:t-1}) = -\frac{1}{t}\sum_{k=0}^{t-1}\frac{1}{|\tilde{\mathcal{D}}_{t\to k}|}\sum_{\tilde{\mathbf{X}}\in\tilde{\mathcal{D}}_{t\to k}} P_{t-1}(\tilde{\mathbf{X}})\cdot\log \overset{*}{P}_t(\tilde{\mathbf{X}}), \tag{12}$$

where $\overset{*}{P}_t(\tilde{\mathbf{X}}) \in \mathbb{R}^{H\times W\times|\mathcal{C}_{0:t-1}|}$ refers to the modified probability distribution from (11), for which *new* and *unknown* categories are incorporated into a single output channel to address the label shift within the u class.

The overall objective is given by:

$$\mathcal{L}_{tot} = \mathcal{L}_{ce}^{\tilde{n}} + \lambda_{ce}^{\tilde{o}}\cdot\mathcal{L}_{ce}^{\tilde{o}} + \lambda_{kd}^{\tilde{n}}\cdot\mathcal{L}_{kd}^{\tilde{n}} + \lambda_{kd}^{\tilde{o}}\cdot\mathcal{L}_{kd}^{\tilde{o}}. \tag{13}$$

VI. Experimental Setup

In this section we provide a detailed description of the experimental setup utilized to validate the proposed framework against multiple competing methods. In Sections VII and VIII we will report the results of the evaluation campaign and extensive ablation studies as additional support.

A. Datasets

To simulate the distribution shift at the input (image) level, we make use of multiple driving data sets, each limited to a specific geographic region or environmental factors, and thus characterized by its distinctive low-level appearance (e.g., road pavement material, type of vehicles, light conditions). On the contrary, the high-level semantic content is mostly consistent across image sets, that is, the road-related or other categories, moving and static obstacles can be found everywhere, and follow similar inter-class structural relations (e.g., the sky will always appear above the road).

Cityscapes: The Cityscapes [68] dataset (CS) is a popular benchmark for autonomous driving applications. Images are collected across 50 cities, all located in Central Europe.

BDD100K: The Berkeley DeepDrive dataset (BDD) [69] is a more diverse collection of road scenes, captured with variable weather conditions at different times of the day. Still, all samples are from 4 restricted localities in the US.

IDD: The Indian Driving Dataset (IDD) [70] includes driving scenes from Indian cities and their outskirts. It offers a diversified set of moving and static road obstacles, as well as a wilder and more natural environment, which breaks away from the typical European or American urban scenarios.

Mapillary Vistas: The Mapillary Vistas dataset [71] contains images collected worldwide, with highly diverse acquisition settings and locations. Unlike previously introduced benchmarks, samples are not limited to a few cities located within quite uniform geographic regions. We leverage the Mapillary dataset to generate continent-wise data splits, as well as to test the domain generalization potential of the proposed class and domain incremental approach.

Shift: The Shift benchmark [72] is a synthetic dataset for autonomous driving, designed to provide a plethora of distribution shifts, simulating the highly variable environmental conditions faced in real-world applications. We exploit it to mimic domain shift due to environmental diversity.

Synscapes: The Synscapes [73] is another synthetic driving dataset, which focuses on realism, and the accurate modeling of illumination and camera processing pipeline.

For BDD, IDD, Synscapes and Mapillary datasets, only the 19 classes available on Cityscapes were used. For Shift, we considered the available 22 semantic categories.

B. Incremental Learning Setup

Domain Incremental Setup: The first domain incremental setup is created by experiencing in succession the CS, BDD and IDD datasets (in different orders) during 3 separate learning steps. Additionally, we propose a further setup, where domain shift across learning steps is achieved by splitting the entire Mapillary dataset into incremental sets based on geographic proximity of samples, i.e., 6 separate data subsets are generated, grouping together pictures taken on the same continent. Finally, we leverage Shift to simulate incrementally variable environmental conditions, by partitioning the whole dataset into 3 groups of samples according to light conditions (i.e., *daytime*, *twilight* and *night*).

Class Incremental Setup: We start by following [40] to identify 3 separate groups within the 19 Cityscapes' classes, i.e., (i) *background regions*, (ii) *moving elements*, (iii) *static elements*, which are observed incrementally under various arrangements. Then, we extend the aforementioned 3-way class splitting to

TABLE II
SPLIT OF CITYSCAPES'S (CS) AND SHIFT'S CLASS SETS FOLLOWING THE CRITERION PROPOSED BY [40]

		$\mathcal{C}_{bgr}$	$\mathcal{C}_{stat}$	$\mathcal{C}_{mov}$
CS	$\mathcal{C}^0$	{road, sidewalk}	{build., wall, fence}	{person, rider, motorcycle, bicycle}
	$\mathcal{C}^1$	{veg., terr., sky}	{pole, t. light, t. sign}	{car, truck, bus, train}
Shift	$\mathcal{C}^s$	{r.line, road, veg., ground, water, s.walk, terr., sky}	{build., wall, fence, pole, t. light, bridge, r.track, g.rail, t. sign, static}	{pedestrian, vehicles, dynamic}

TABLE III
CLASS AND DOMAIN INCREMENTAL SETS

	Class sets	Domains
Urban	$\{\mathcal{C}_{bgr}, \mathcal{C}_{stat}, \mathcal{C}_{mov}\}$	{CS, BDD, IDD}
Worldwide	$\{\mathcal{C}^0_{bgr}, \mathcal{C}^1_{bgr}, \mathcal{C}^0_{stat}, \mathcal{C}^1_{stat}, \mathcal{C}^0_{mov}, \mathcal{C}^1_{mov}\}$	{EU, NA, AS, OC, AF, SA}
Environmental	$\{\mathcal{C}^s_{bgr}, \mathcal{C}^s_{stat}, \mathcal{C}^s_{mov}\}$	{Daytime, Twilight, Night}

$\mathcal{C}^s$ indicates that the class subset is derived from Shift's original set.

Shift in a similar fashion to [40], this time on the 22 classes offered by the synthetic benchmark. All the class incremental sets are detailed in Table II.

By merging class and domain individual settings, we devise each class and domain incremental setup reported in Table III. The first (i.e., *urban*) is generated using CS, BDD and IDD datasets, together with the 3-way class split from [40]. Formally, we set the total number of learning steps $T = 3$, and at each step $0 \leq t < T$:

$$\mathcal{D}_t \subset (\mathcal{X}_t, \mathcal{C}_t) \in \{\text{CS}, \text{BDD}, \text{IDD}\} \times \{\mathcal{C}_{bgr}, \mathcal{C}_{stat}, \mathcal{C}_{mov}\}, \tag{14}$$

where each dataset and class split is observed once. We further propose an incremental setup (i.e., *worldwide*) based on continent-wise splitting of the Mapillary dataset. To match the increase in domain set size to 6 elements, we divide each class group [40] in half, for a total of 6 class splits (Table II). We set $T = 6$, and at each step $0 \leq t < T$:

$$\mathcal{D}_t \subset (\mathcal{X}_t, \mathcal{C}_t) \in \{\text{EU}, \text{NA}, \text{AS}, \text{OC}, \text{AF}, \text{SA}\} \times \{\mathcal{C}^0_{bgr}, \mathcal{C}^1_{bgr}, \mathcal{C}^0_{stat}, \mathcal{C}^1_{stat}, \mathcal{C}^0_{mov}, \mathcal{C}^1_{mov}\}, \tag{15}$$

where each class set and each domain appears only in a single step. Among the large number of possible incremental sequences, we perform the experimental evaluation in the EU $\rightarrow$ NA $\rightarrow$ AS $\rightarrow$ OC $\rightarrow$ AF $\rightarrow$ SA and $\mathcal{C}^0_{bgr} \rightarrow \mathcal{C}^1_{bgr} \rightarrow \mathcal{C}^0_{stat} \rightarrow \mathcal{C}^1_{stat} \rightarrow \mathcal{C}^0_{mov} \rightarrow \mathcal{C}^1_{mov}$ setups.

Finally, the last setup (i.e., *environmental*) combines the environmental partitioning chosen for Shift with the 3-way class splitting from [40].

C. Implementation Details

We built our framework in PyTorch. Due to the complexity of the investigated problem, in most experiments we use a lightweight segmentation model, i.e., ErfNet [3]. We argue that a smaller network complies more realistically to deployment-related constraints in real-word applications, e.g., in terms of memory occupation and inference speed. Yet, for comparison purposes we report additional results with the heavier and better performing DeeplabV3 architecture [74] with ResNet101 backbone [75]. In all experiments, the segmentation model is pre-trained on ImageNet [76].

With ErfNet, we use the Adam optimizer [77] and learning rate set to $5\text{e}{-4}$. With DeeplabV3, we use the SGD optimizer and learning rate set to $1\text{e}{-3}$. Weight decay is fixed to $1\text{e}{-4}$, and we employ a polynomial decay of power 0.9 for learning rate scheduling. We train for 100 and 50 epochs at each learning step, with ErfNet and DeeplabV3 respectively (except in Shift, where we set the number of epochs to 10). With ErfNet we use a batch size of 6, with DeeplabV3 we reduce its value to 2 due to GPU memory constraints.

When experimentally evaluating on Cityscapes-BDD-IDD and Shift setups, images are resized to 512×1024 resolution. When using Mapillary for training, inputs are first resized to 1024 width (fixed aspect ratio), and then cropped to 512×1024. This pre-processing is done to accommodate for the highly variable aspect ratios of Mapillary's samples.

The β parameter controlling the size of the style window is empirically set to $1\text{e}{-2}$ and fixed in all experiments. Plus, we experimentally fix $\lambda^{\tilde{o}}_{ce} = \lambda^{\tilde{n}}_{kd} = \lambda^{\tilde{o}}_{kd} = 10$, and keep them unchanged in every incremental setup. This shows that our approach is robust to change of experimental setting, and requires minimal hyper-parameter tuning. Ablation studies on the impact of β and loss weights are in Section VIII.

D. Competitors

To the best of our knowledge, this is the first work explicitly modeling and addressing class and domain incremental learning in semantic segmentation. For this reason, we compare with other methods targeting class (CIL) or domain (DIL) incremental learning as individual problems.

Among class-incremental methods, we consider ILT [11] and MiB [12], along with state-of-the-art PLOP [13] and UCD [43]. When using PLOP with ErfNet, we apply the *LocalPOD* loss [13] on embeddings extracted at the end of the first and second blocks, as well as at the output of the encoder. For UCD, we modify the contrastive distillation loss so that the maximum number of positives and negatives is set to 3000 each (which are randomly selected among the whole sets as defined in the original work). We perform this adjustment to meet GPU memory limitations. All experiments were performed on a RTX Titan GPU with 24 GB of memory. We believe that a fair comparison should involve comparable GPU resources for all the competitors.

On the domain-incremental side, we compare with [64]. Differently from our setup, they assume to have full task supervision on all the domains incrementally encountered. We adapt their framework to a class-incremental setup by replacing the standard cross-entropy loss with the unbiased version from [12], to prevent the background shift from erasing the task-knowledge learned in past steps.

TABLE IV
EXPERIMENTAL RESULTS ON CS → BDD → IDD DOMAIN SETUP AND $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ CLASS SETUP

CS → BDD → IDD $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$	Step 0 **CS** ($\mathcal{X}_0$)			Step 1 **BDD** ($\mathcal{X}_1$)		CS ($\mathcal{X}_0$)			Step 2 **IDD** ($\mathcal{X}_2$)		BDD ($\mathcal{X}_1$)		CS ($\mathcal{X}_0$)		
	$\text{mIoU}_0^0\uparrow$	$\Delta_0^0\downarrow$	$\bar{\Delta}_0\downarrow$	$\text{mIoU}_1^1\uparrow$	$\Delta_1^1\downarrow$	$\text{mIoU}_1^0\uparrow$	$\Delta_1^0\downarrow$	$\bar{\Delta}_1\downarrow$	$\text{mIoU}_2^2\uparrow$	$\Delta_2^2\downarrow$	$\text{mIoU}_2^1\uparrow$	$\Delta_2^1\downarrow$	$\text{mIoU}_2^0\uparrow$	$\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
FT ($\mathcal{L}_{ce}^{n}$)	79.67	5.32	5.32	24.38	61.35	18.11	74.06	67.71	26.27	61.48	10.47	81.72	12.10	81.18	74.79
FT w/ self-style ($\mathcal{L}_{ce}^{\tilde{n}}$)	79.19	5.89	5.89	20.41	67.65	19.08	72.67	70.16	27.12	60.24	11.51	79.91	13.68	78.72	72.95
MDIL [64]	80.35	4.51	**4.51**	26.12	58.59	23.65	66.13	62.36	28.10	58.80	12.46	78.25	13.22	79.44	72.16
ILT [11]	79.67	5.32	5.32	22.21	64.80	44.70	35.99	50.39	26.69	60.87	16.70	70.85	29.76	53.71	61.81
MiB [12]	79.67	5.32	5.32	34.35	45.55	49.24	29.48	37.51	42.58	37.57	26.36	53.98	36.58	43.10	44.88
PLOP [13]	79.67	5.32	5.32	36.78	41.70	50.05	28.32	35.01	43.15	36.73	27.24	52.44	36.84	42.70	43.96
UCD [43]	79.67	5.32	5.32	35.45	43.80	50.38	27.85	35.83	43.19	36.67	27.38	52.19	37.34	41.91	43.59
LwS w/o $\mathcal{L}_{kd}^{\tilde{o}}$	79.19	5.89	5.89	44.41	29.60	50.77	27.29	28.44	50.70	25.66	34.86	39.14	43.04	33.05	32.62
LwS	79.19	5.89	5.89	44.47	29.51	53.31	23.65	**26.58**	51.20	24.93	35.73	37.62	44.17	31.29	**31.28**
Oracle	84.15	-	-	63.08	-	69.82	-	-	68.20	-	57.28	-	64.29	-	-

E. Metrics

Inspired by [64], to provide a valuable measure of prediction performance across multiple tasks and domains, we resort to a domain average relative performance w.r.t. a fully-supervised *oracle* reference (the smaller the better) defined at any step t as:

$$\bar{\Delta}_t = \underbrace{\frac{1}{t+1}\sum_{k=0}^{t}}_{\text{domain avg}} \underbrace{\frac{A_{\mathcal{X}_k|S_t}^{\mathcal{C}_{0:t}} - A_{\mathcal{X}_k|S^*}^{\mathcal{C}_{0:t}}}{A_{\mathcal{X}_k|S^*}^{\mathcal{C}_{0:t}}}}_{\substack{\Delta_t^k\text{: relative acc. gap w.r.t.} \\ \text{oracle on step-k domain}}}, \tag{16}$$

where $A_{\mathcal{X}|S}^{\mathcal{C}}$ is the class-average accuracy (we make use of the commonly employed mIoU metric [20]) attained by segmentation network S on domain $\mathcal{X}$ and class set $\mathcal{C}$. S^* is the oracle segmentation model, i.e., trained with full supervision on the entire pool of classes and domains (even classes and domains that will be observed after step t).

We further provide a measure of generalization aptitude (the higher the better), expressed as the accuracy (i.e., in terms of mIoU) achieved over the entire class set observed so far on a novel dataset never experienced before. At step t, the metric follows:

$$\Gamma_t^{gen} = A_{\mathcal{X}_{ext}|S_t}^{\mathcal{C}_{0:t}} = \frac{1}{|\mathcal{C}_{0:t}|}\sum_{c\in\mathcal{C}_{0:t}} A_{\mathcal{X}_{ext}|S_t}^{c}, \tag{17}$$

where $\mathcal{X}_{ext}$ is the unseen domain.

VII. EXPERIMENTAL RESULTS

A. Evaluation on Urban Scenes

The first experimental setup we explore entails incrementally transitioning between urban and suburban areas of different regions around the world. High- and low- level image contents undergo distribution shifts of different extent: although it might be reasonable to assume that the basic semantic structure of road images is invariant to geographic location, scene elements are likely to change appearance significantly when travelling around the world.

1) Study on Domain Ordering: To reproduce class and domain distribution shifts, we train on the Cityscapes, BDD and IDD datasets in an incremental fashion. The class incremental protocol is instead the one proposed in [40] (i.e., $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$). As detailed in Section VI-B, we define a total of 3 learning steps. In Tables IV, V and VI we report experimental results following 3 different dataset orders, so that each dataset is viewed at all the 3 possible learning steps, considering all experiments performed.

We report results in terms of mIoU computed over all classes excluding the *unknown* one, as typically done in the literature. The mIoU is computed for each domain $\mathcal{X}_k$ (i.e., dataset) experienced up to a current step t (i.e., mIoU_t^k, $k \leq t$), $\forall t < T$. In addition, we provide a measure of relative performance w.r.t. a supervised reference, both for individual domains Δ_t^k, and as a global quantity $\bar{\Delta}_t$ ((16)). The supervised reference, denoted as *Oracle*, corresponds to the joint training over both class sets and domains, i.e., to a multi-dataset training with all classes labeled in all samples.

We compare with methods addressing class incremental learning (ILT [11], MiB [12], PLOP [13] and UCD [43]) and with a recent domain incremental method (MDIL [64]). We also include a simple baseline, activating only the task loss on the new classes and new domain (6). This approach is usually referred to as *fine-tuning*, as the focus is just posed on learning the new task. Two variants are reported for this baseline, i.e., with or without self-stylization applied on input images, indicated respectively as $\mathcal{L}_{ce}^{\tilde{n}}$ and $\mathcal{L}_{ce}^{n}$. As for our approach, we evaluate its final form (13), complete of all the training objectives detailed in Section V, as well as a simpler configuration without the $\mathcal{L}_{kd}^{\tilde{o}}$ loss (12).

By inspecting results in Tables IV, V and VI, we notice that the performance achieved by different methods at the end of the **initial learning step** are comparable. This is due to the similar objectives employed so far, to learn just the first class set ($\mathcal{C}_{bgr}$) on the first domain, regardless of the domain order. We remark that the proposed self-stylization is not detrimental when learning the current task. We will provide some ablation studies on the impact of stylization in Section VIII.

When progressing to the **first incremental step**, catastrophic forgetting has to be addressed to retain good performance. We observe that the $\mathcal{L}_{ce}^{n}$ and $\mathcal{L}_{ce}^{\tilde{n}}$ losses alone are not sufficient to achieve satisfactory results, being focused on the new task and providing no constraints to preserve past knowledge. MDIL [64] performs poorly as well, since the proposed dynamic architecture is not suitable to address partial class incremental supervision, which in our setup is present along with domain incremental shift.

TABLE V
EXPERIMENTAL RESULTS ON BDD → IDD → CS DOMAIN SETUP AND $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ CLASS SETUP

BDD → IDD → CS $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$	Step 0 **BDD** ($\mathcal{X}_0$)			Step 1 **IDD** ($\mathcal{X}_1$)		BDD ($\mathcal{X}_0$)			Step 2 **CS** ($\mathcal{X}_2$)		IDD ($\mathcal{X}_1$)		BDD ($\mathcal{X}_0$)		
	$\text{mIoU}_0^0\uparrow$	$\Delta_0^0\downarrow$	$\bar{\Delta}_0\downarrow$	$\text{mIoU}_1^1\uparrow$	$\Delta_1^1\downarrow$	$\text{mIoU}_1^0\uparrow$	$\Delta_1^0\downarrow$	$\bar{\Delta}_1\downarrow$	$\text{mIoU}_2^2\uparrow$	$\Delta_2^2\downarrow$	$\text{mIoU}_2^1\uparrow$	$\Delta_2^1\downarrow$	$\text{mIoU}_2^0\uparrow$	$\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
FT ($\mathcal{L}_{ce}^n$)	72.22	6.61	6.61	33.37	52.43	20.49	67.52	59.98	23.36	65.60	7.09	89.60	5.45	89.02	81.41
FT w/ self-style ($\mathcal{L}_{ce}^{\tilde{n}}$)	72.12	6.74	6.74	33.27	52.58	21.12	66.52	59.55	28.52	55.64	15.44	77.36	14.24	75.14	69.38
MDIL [64]	72.44	6.33	**6.33**	26.78	61.83	15.44	75.52	68.68	25.52	60.30	11.61	82.98	10.77	81.20	74.83
ILT [11]	72.22	6.61	6.61	42.10	39.98	43.00	31.84	35.91	33.33	48.15	26.93	60.52	29.68	48.19	52.29
MiB [12]	72.22	6.61	6.61	52.18	25.62	45.28	28.22	26.92	48.22	25.00	33.57	50.77	30.94	45.98	40.58
PLOP [13]	72.22	6.61	6.61	53.15	24.24	44.25	29.85	27.05	47.21	26.56	35.36	48.15	32.02	44.10	39.60
UCD [43]	72.22	6.61	6.61	52.42	25.28	45.20	28.35	26.81	48.40	24.72	32.60	52.19	28.95	49.47	42.13
LwS w/o $\mathcal{L}_{kd}^{\tilde{o}}$	72.12	6.74	6.74	54.34	21.07	41.36	34.44	27.75	52.56	18.24	36.70	46.19	32.33	43.56	36.00
LwS	72.12	6.74	6.74	54.53	20.80	43.98	30.28	**25.54**	52.63	18.14	38.14	44.08	34.03	40.59	**34.27**
Oracle	77.33	-	-	70.16	-	63.08	-	-	68.20	-	57.28	-	64.29	-	-

TABLE VI
EXPERIMENTAL RESULTS ON IDD → CS → BDD DOMAIN SETUP AND $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ CLASS SETUP

IDD → CS → BDD $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$	Step 0 **IDD** ($\mathcal{X}_0$)			Step 1 **CS** ($\mathcal{X}_1$)		IDD ($\mathcal{X}_0$)			Step 2 **BDD** ($\mathcal{X}_2$)		CS ($\mathcal{X}_1$)		IDD ($\mathcal{X}_0$)		
	$\text{mIoU}_0^0\uparrow$	$\Delta_0^0\downarrow$	$\bar{\Delta}_0\downarrow$	$\text{mIoU}_1^1\uparrow$	$\Delta_1^1\downarrow$	$\text{mIoU}_1^0\uparrow$	$\Delta_1^0\downarrow$	$\bar{\Delta}_1\downarrow$	$\text{mIoU}_2^2\uparrow$	$\Delta_2^2\downarrow$	$\text{mIoU}_2^1\uparrow$	$\Delta_2^1\downarrow$	$\text{mIoU}_2^0\uparrow$	$\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
FT ($\mathcal{L}_{ce}^n$)	78.80	8.52	**8.52**	9.66	47.37	8.64	93.07	70.22	9.66	83.14	8.64	86.56	7.09	89.60	86.43
FT w/ self-style ($\mathcal{L}_{ce}^{\tilde{n}}$)	78.78	8.55	8.55	42.11	39.69	19.81	71.76	55.73	14.05	75.47	12.63	80.35	11.01	83.86	79.89
MDIL [64]	78.72	8.62	8.62	34.87	50.06	11.70	83.32	66.69	8.90	84.46	8.22	87.21	6.70	90.18	87.28
ILT [11]	78.80	8.52	**8.52**	44.44	36.35	43.32	38.26	37.30	24.48	57.26	30.00	53.34	27.88	59.12	56.57
MiB [12]	78.80	8.52	**8.52**	56.23	19.47	23.59	66.37	42.92	23.62	58.76	33.24	48.30	20.57	69.84	58.97
PLOP [13]	78.80	8.52	**8.52**	57.05	18.29	24.74	64.74	41.51	24.18	57.79	34.23	46.76	21.42	68.59	57.71
UCD [43]	78.80	8.52	**8.52**	56.29	19.38	26.45	62.29	40.84	24.88	56.57	34.72	45.99	22.35	67.24	56.60
LwS w/o $\mathcal{L}_{kd}^{\tilde{o}}$	78.78	8.55	8.55	59.61	14.63	43.30	38.28	26.45	34.84	39.18	39.11	39.17	36.13	47.03	41.79
LwS	78.78	8.55	8.55	59.26	15.13	43.95	37.35	**26.24**	37.94	33.76	42.10	34.51	36.60	46.34	**38.21**
Oracle	86.14	-	-	69.82	-	70.16	-	-	68.20	-	57.28	-	64.29	-	-

By analyzing class incremental learning methods, we note that they are able to preserve previously acquired knowledge to some extent, while allowing some plasticity for learning the new task. Still, the domain shift between previous and current datasets has a negative impact on the prediction accuracy of the incrementally trained predictor. All the considered CIL methods, in fact, rely on the ability of a segmentation model frozen from the previous step to preserve knowledge of the past. Yet, because of the domain discrepancy between past and new data, this distillation mechanism could introduce unreliable guidance on former tasks, as the frozen model is subject to a shift in the experienced distribution at the input level when fed with new domain data. At the same time, the distribution gap may hinder the transferability of new-class knowledge to old domains, which are no longer available as training data.

These drawbacks are revealed by results of Table VI (IDD → CS → BDD): the significant domain shift between the Cityscapes and IDD datasets prevents CIL methods from effectively preserving and learning task-related clues on IDD, which was experienced at step 0. On the contrary, our approach addresses domain shift by leveraging the stylization scheme and applying carefully designed objectives to suitably tackle the general class and domain incremental learning. In particular, the proposed objectives $\mathcal{L}_{ce}^{\tilde{o}}$ (7) and $\mathcal{L}_{kd}^{\tilde{o}}$ (12) are specifically designed to address the aforementioned problems affecting CIL methods and allow to achieve superior accuracy on former domains. As a result, LwS improves accuracy by more than 17 mIoU points on IDD at step 1 w.r.t. the best competitor (i.e., UCD [43]).

We also remark that, even with alternative domain orders (Tables IV and V), LwS shows the best stability-plasticity trade-off, retaining the best overall accuracy in terms of $\bar{\Delta}_1$. Furthermore, we can see that, for both CS → BDD → IDD and BDD → IDD → CS orders, the addition of the $\mathcal{L}_{kd}^{\tilde{o}}$ objective in LwS leads to a boost in performance on the past domain, which coincides with the design purpose of the objective.

In the **final learning step**, the struggle to handle the class and domain incremental training is exacerbated for all the competitors. Baselines and MDIL still provide inferior results, with the latter performing even worse than naïve fine-tuning with self-stylization in some setups.

As for CIL methods, PLOP [13] and UCD [43] are the best performing. Both combines output and feature level objectives, which prove to be somewhat robust to domain shift. Even so, the simpler MiB [12] approach shows very competitive results, suggesting that strategies taking into account only a class incremental perspective may not be so effective when incremental domain shift is also occurring. Our method in its complete form greatly outperforms all CIL competitors by a large margin regardless of domain order, going from 5% (BDD → IDD → CS) to 12% (CS → BDD → IDD) and even 16% (IDD → CS → BDD) in terms of $\bar{\Delta}_2$ gap.

Furthermore, in Table VII we investigate the generalization performance (i.e., Γ_t^{gen} from (17)) achieved by the considered methods. To do so, we compute the accuracy at each incremental step on the *unseen* Mapillary dataset for the sets of classes observed so far. Notice that this study is very relevant for a generalization assessment, since inference is performed on a

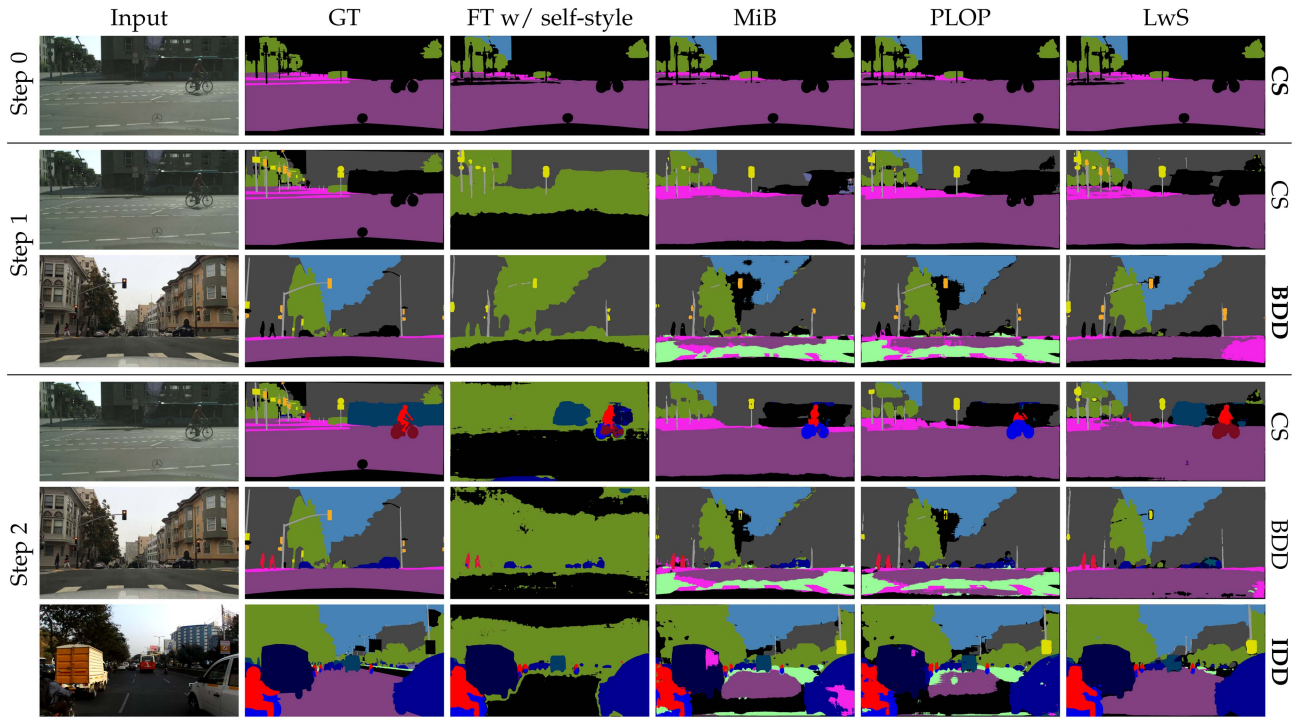

Fig. 4. Qualitative results on CS → BDD → IDD domain setup and $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ class setup.

TABLE VII
GENERALIZATION PERFORMANCE (Γ_t^{gen}) AS MIOU COMPUTED ON MAPILLARY'S TEST SET ($\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ SETUP)

	CS → BDD → IDD			BDD → IDD → CS			IDD → CS → BDD		
	Step 0	Step 1	Step 2	Step 0	Step 1	Step 2	Step 0	Step 1	Step 2
FT ($\mathcal{L}_{ce}^{n}$)	36.27	22.03	13.71	66.74	25.27	6.60	**59.56**	7.81	8.52
FT† ($\mathcal{L}_{ce}^{n}$)	**58.09**	19.83	14.99	**66.83**	25.77	16.34	59.19	23.40	11.97
MDIL	44.60	24.77	16.05	66.36	18.40	11.01	56.41	14.86	8.55
ILT	36.27	26.80	20.69	66.74	41.32	28.97	**59.56**	39.27	27.96
MiB	36.27	37.68	32.36	66.74	45.99	33.01	**59.56**	23.61	24.23
PLOP	36.27	39.62	33.69	66.74	45.45	34.01	**59.56**	25.29	25.04
UCD	36.27	38.46	34.07	66.74	**46.22**	29.92	**59.56**	27.08	25.89
LwS	**58.09**	**46.36**	**40.43**	**66.83**	44.99	**37.33**	59.19	**43.15**	**39.16**
Oracle	83.96	73.77	65.42	83.96	73.77	65.42	83.96	73.77	65.42

† indicates the presence of self-stylization.

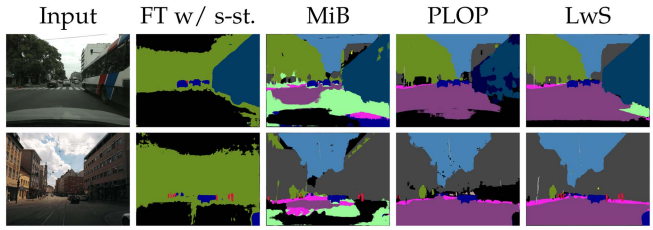

Fig. 5. Qualitative results on test images from the Mapillary dataset at the end of step 2 (CS → BDD → IDD domain setup and $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ class setup).

domain totally disjoint from the training ones. We notice that simple fine-tuning and MDIL offer poor generalization results, which is expected due to the low accuracy they already provide on datasets directly observed. On the other hand, CIL methods reach more competitive results, even if none of them proves to be superior in all setups. Still, our approach outperforms all competitors, getting significantly closer to the *Oracle* upper-bound (i.e., the supervised training on the entire Mapillary), specially in the IDD → CS → BDD setup. Also, we remark how we get similar generalization results with different domain incremental orders, demonstrating how our approach is able to learn and preserve generalizable task-related clues regardless of the training environment.

Finally, qualitative results in the form of segmentation maps are provided in Fig. 4. We stress how the proposed approach yields better **backward** and **forward transfer** throughout the incremental learning. In particular, moving classes like *bicycle* and *bus* appear to be recognized more effectively by our method on the Cityscapes (CS) dataset at the end of the incremental training, even though CS was experienced only along with background-class supervision during the first step. On the other hand, MiB and PLOP fail to provide satisfactory **backward transfer** of those classes to the past CS domain. A similar reasoning can be done regarding the **forward transfer** aptitude. Our approach is able to deliver good segmentation accuracy on the *road* and *sidewalk* background classes even on BDD and IDD datasets, despite them being experienced when $\mathcal{C}_{bgr}$ supervision is no longer available. Contrarily, MiB and PLOP suffer from the domain statistical gap across learning steps, struggling to maintain satisfactory segmentation accuracy on first-step classes by forward transferring knowledge to future steps. Some visual results for the evaluation on Mapillary are instead shown in Fig. 5. Additional analyses will be provided in Section VIII-D.

2) Study on Class Ordering: We further investigate the impact of a permutation of the class incremental arrangement.

TABLE VIII
EXPERIMENTAL RESULTS ON CS → BDD → IDD DOMAIN SETUP AND $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{mov} \rightarrow \mathcal{C}_{stat}$ CLASS SETUP

CS → BDD → IDD $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$			Method $\mathcal{L}^{n}_{ce}$	$\mathcal{L}^{\bar{n}}_{ce}$	MiB	PLOP	UCD	LwS	Oracle
Step 0	$\text{mIoU}_0\uparrow$	**CS**	79.83	79.39	79.83	79.83	79.83	79.39	84.15
	$\bar{\Delta}_0\downarrow$		**5.13**	5.65	**5.13**	**5.13**	**5.13**	5.65	-
Step 1	$\text{mIoU}_1\uparrow$	**BDD**	15.79	19.43	26.15	27.69	23.73	40.92	63.08
		CS	14.26	17.22	40.38	42.44	39.76	49.70	69.82
	$\bar{\Delta}_1\downarrow$		76.26	71.03	48.21	45.40	50.68	**28.99**	-
Step 2	$\text{mIoU}_2\uparrow$	**IDD**	13.47	14.82	31.01	31.89	30.72	43.54	68.20
		BDD	6.94	8.21	23.40	25.21	22.83	32.34	57.28
		CS	7.45	10.49	33.60	33.81	32.85	39.76	64.29
	$\bar{\Delta}_2\downarrow$		85.52	82.54	53.81	52.21	54.67	**39.29**	-

TABLE IX
EXPERIMENTAL RESULTS WITH DEEPLABV3-RESNET101

CS → BDD → IDD $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$			Method $\mathcal{L}^{n}_{ce}$	$\mathcal{L}^{\bar{n}}_{ce}$	MiB	PLOP	UCD	LwS	Oracle
Step 0	$\text{mIoU}_0\uparrow$	**CS**	78.1	77.13	78.1	78.1	78.1	77.13	84.30
	$\bar{\Delta}_0\downarrow$		**7.35**	8.50	**7.35**	**7.35**	**7.35**	8.50	-
Step 1	$\text{mIoU}_1\uparrow$	**BDD**	31.97	29.44	28.38	29.07	30.84	50.36	64.60
		CS	31.82	55.13	45.10	45.15	45.51	54.75	71.24
	$\bar{\Delta}_1\downarrow$		52.92	56.19	46.38	45.81	44.19	**22.60**	-
Step 2	$\text{mIoU}_2\uparrow$	**IDD**	30.68	30.29	35.93	33.98	38.24	51.92	70.94
		BDD	17.48	17.27	24.18	23.57	26.14	44.98	61.48
		CS	19.46	17.92	33.22	34.38	34.93	47.08	69.17
	$\bar{\Delta}_2\downarrow$		66.73	67.77	53.99	54.68	51.02	**28.53**	-

Table VIII reports experimental results with the CS → BDD → IDD progression, but a modified class order with moving categories $\mathcal{C}_{mov}$ experienced before static ones $\mathcal{C}_{stat}$. We notice a similar trend to that observed in Table IV (i.e., same domain order, but different class order), with baselines and MDIL [64] performing poorly, and the improved accuracy achieved by CIL methods still being largely outperformed by the proposed approach.

In addition, we observe that the absolute results are decreased by applying the new class order. The performance of our approach, in fact, drops from 31.28% to 39.29% of $\bar{\Delta}_2$. This discrepancy might be due to class sets observed on domains where it is harder to learn them, and, at the same time, to generalize to the other domains. For instance, we note that IDD provides a lower overall percentage of pixels of $\mathcal{C}_{stat}$ w.r.t. the BDD (11% vs 17%), while for $\mathcal{C}_{mov}$ numbers are similar between them (both around 10% of total pixels). Still, the performance loss is similar for CIL methods, with the gap w.r.t. the best competitor rising from 12 to 13 points of $\bar{\Delta}_2$ (compared to the previous class order).

3) Study on Model Architecture: We finally evaluate the considered methods when a more complex segmentation network is used, moving from the lightweight ErfNet to the heavier DeeplabV3 with ResNet101 backbone. For comparison purposes, the setup analyzed is again that involving CS → BDD → IDD and $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ orders (Table IX). For what concerns our approach, we observe an improved relative performance, raising from 31.28% to 28.53% in terms of $\bar{\Delta}_2$. We emphasize that the $\bar{\Delta}$ measure already takes into account the better oracle results; the accuracy boost, then, shows that our method is able to capitalize the increased capacity offered by the segmentation model.

On the other hand, the CIL competitors are unable to take advantage of the growth in network capacity, which could indicate a tendency to overfit on the currently observed domain distribution. The best competitor (i.e., UCD), in fact, is significantly outperformed by more than 20% in terms of $\bar{\Delta}$ at both steps 1 and 2. We remark that no additional parameter tuning is performed in this experimental setup concerning method-specific parameters.

B. Evaluation With Synthetic Data

To evaluate the capability of our approach to tackle the large domain shift between synthetic and real data, we perform an additional experiment with the synthetic Synscapes dataset [73] observed at the initial step, followed by Cityscapes and BDD in the other two steps. From Table XI it is clear how LwS is able to outperform competitors by a large margin, even in this setting.

We additionally conduct an analysis on the generalization aptitude of trained models similar to that of Table VII, using the same 3 datasets for training (with Synscapes at step 0) and testing on Mapillary. Our approach reaches a final mIoU of 37.97% w.r.t. the 21.25% achieved by the best competitor, i.e., results are aligned with those of Table VII, with an even greater gap compared to the other methods. We also analyze how trained models generalize from real to synthetic data. We evaluate models learned under the incremental settings of Table IV on Synscapes: LwS obtains a mIoU of 46.09% w.r.t. the 38.69% of the best competitor.

Finally, we consider a multi-dataset setup. In particular, we employ Synscapes at step 0 as a sort of synthetic pre-training, followed by two real-world datasets (Cityscapes and BDD) jointly observed at step 1, and another real-world dataset (IDD) experienced in the last step. We achieve a final $\bar{\Delta}_2$ score of 37.66%, while the best competitors stand at around 50%.

C. Evaluation With Larger Geographic Diversity

The second experimental class and domain incremental setup we explore is derived from the Mapillary dataset. Domain shift is once more induced by the variable geographic origin of image samples collected worldwide, i.e., we identify data partitions associated to 6 different continents, corresponding to 6 incremental steps. However, the Mapillary dataset contains variegate data distribution, even considering intra-continent samples, providing a more robust support for training segmentation models. Data richness in turns promotes generalization across steps, in fact lessening the domain gap between different domains. We report experimental results in Table X. In the first steps, when the domain shift is small (e.g., between Europe, EU, and North America, NA), the different methods achieve similar performance. Nonetheless, when progressing to the last steps and experiencing increased statistical gap (e.g., when introducing Africa's images, AF), we note that our approach outperforms CIL competitors

TABLE X
EXPERIMENTAL RESULTS ON THE MAPILLARY DATASET

EU→NA→AS→OC→AF→SA $\mathcal{C}_{bgr}^{0\to1} \to \mathcal{C}_{stat}^{0\to1} \to \mathcal{C}_{mov}^{0\to1}$			Method $\mathcal{L}_{ce}^{n}$	$\mathcal{L}_{ce}^{\tilde{n}}$	MiB	PLOP	UCD	LwS	Oracle
Step 0	$\text{mIoU}_0\uparrow$	**EU**	73.12	73.07	73.12	73.12	73.12	73.07	79.53
	$\bar{\Delta}_0\downarrow$		**8.06**	8.13	**8.06**	**8.06**	**8.06**	8.13	-
Step 1	$\text{mIoU}_1\uparrow$	**NA**	51.80	51.63	81.28	80.82	81.70	81.85	87.51
		EU	47.67	47.52	76.05	75.76	75.26	74.80	82.34
	$\bar{\Delta}_1\downarrow$		41.46	41.65	**7.38**	7.82	7.62	7.82	-
Step 2	$\text{mIoU}_2\uparrow$	**AS**	25.18	26.09	65.40	65.98	65.70	65.36	74.70
		NA	23.61	23.82	69.28	69.63	68.66	69.77	79.40
		EU	23.98	24.10	66.66	66.86	65.79	65.87	76.62
	$\bar{\Delta}_2\downarrow$		68.42	67.87	12.73	**12.24**	13.23	12.89	-
Step 3	$\text{mIoU}_3\uparrow$	**OC**	16.53	16.74	61.29	60.58	60.53	63.07	76.46
		AS	14.31	14.22	57.95	57.60	57.61	58.13	70.96
		NA	17.10	17.20	62.41	63.29	61.91	64.04	75.77
		EU	14.94	14.94	59.78	60.01	59.34	61.15	72.97
	$\bar{\Delta}_3\downarrow$		78.79	78.72	18.47	18.46	19.15	**16.82**	-
Step 4	$\text{mIoU}_4\uparrow$	**AF**	8.98	7.77	38.48	39.97	40.54	43.93	66.54
		OC	6.03	5.95	40.17	43.52	42.15	47.43	72.30
		AS	7.23	7.31	39.15	41.09	42.03	46.13	69.87
		NA	7.78	7.07	43.10	45.12	45.00	50.07	74.22
		EU	5.45	5.41	38.99	41.52	41.28	46.37	70.22
	$\bar{\Delta}_4\downarrow$		89.91	90.48	43.40	40.20	40.24	**33.77**	-
Step 5	$\text{mIoU}_5\uparrow$	**SA**	9.61	9.18	39.64	41.79	41.48	45.36	64.45
		AF	11.35	9.63	40.76	41.98	41.44	45.25	63.03
		OC	6.78	7.42	36.52	39.08	37.21	41.76	60.82
		AS	8.97	8.17	37.90	40.15	38.76	43.63	64.74
		NA	8.97	9.54	41.40	43.45	43.05	47.08	66.88
		EU	8.18	7.63	38.37	40.53	38.93	43.51	64.04
	$\bar{\Delta}_5\downarrow$		85.98	86.58	38.90	35.67	37.28	**30.57**	-

TABLE XI
EXPERIMENTAL RESULTS ON SYNSCAPES → CS → BDD DOMAIN SETUP AND $\mathcal{C}_{bgr} \to \mathcal{C}_{stat} \to \mathcal{C}_{mov}$ CLASS SETUP

Syn→CS→BDD $\mathcal{C}_{bgr} \to \mathcal{C}_{stat} \to \mathcal{C}_{mov}$	**Syn** $\bar{\Delta}_0\downarrow$	**CS** $\Delta_1^1\downarrow$	Syn $\Delta_1^0\downarrow$	$\bar{\Delta}_1\downarrow$	**BDD** $\Delta_2^2\downarrow$	CS $\Delta_2^1\downarrow$	Syn $\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
FT ($\mathcal{L}_{ce}^{n}$)	**-0.40**	49.49	88.43	68.96	78.06	83.02	90.63	83.90
FT w/ self-style ($\mathcal{L}_{ce}^{\tilde{n}}$)	-0.32	49.18	67.07	58.13	79.20	85.48	90.45	85.04
MiB [12]	**-0.40**	44.90	57.53	51.21	63.02	64.89	69.40	65.77
PLOP [13]	**-0.40**	45.24	61.24	53.24	64.13	65.28	70.61	66.67
UCD [43]	**-0.40**	43.66	59.47	51.57	65.41	67.64	69.53	67.52
LwS w/o $\mathcal{L}_{kd}^{\tilde{o}}$	-0.32	18.44	26.51	22.47	41.12	44.31	52.92	46.12
LwS	-0.32	17.97	24.34	**21.15**	36.93	40.16	50.22	**42.44**

by a considerable margin, which is of 5 points of $\bar{\Delta}$ w.r.t. the best competitor (PLOP) at the end of incremental training. Also, superior performance in later steps is attained on both new and old domains, confirming the better plasticity-stability trade-off provided by our method. Overall, the improved results LwS reaches w.r.t. state-of-the-art CIL competitors, even when training data is collected to ensure some statistical diversity (as in the experimental setup just considered), further suggests that CIL methods are likely to be inadequate to deal with distribution shift in the input space.

D. Evaluation With Variable Environmental Conditions

We evaluate the proposed method when incremental domain shift is due to changing environmental factors, i.e., variable light conditions experienced at different times during the day. In this setting, we employed the Shift synthetic benchmark. We consider the $Daytime \to Twilight \to Night$ domain sequence.

TABLE XII
EXPERIMENTAL RESULTS ON THE SHIFT DATASET

Daytime→Twilight→Night $\mathcal{C}_{bgr} \to \mathcal{C}_{stat} \to \mathcal{C}_{mov}$	**Night** $\text{mIoU}_2^2\uparrow$	$\Delta_2^2\downarrow$	Twilight $\text{mIoU}_2^1\uparrow$	$\Delta_2^1\downarrow$	Daytime $\text{mIoU}_2^0\uparrow$	$\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
FT ($\mathcal{L}_{ce}^{n}$)	10.54	85.82	9.62	87.21	4.61	94.06	89.03
FT w/ self-style ($\mathcal{L}_{ce}^{\tilde{n}}$)	10.12	86.39	8.50	88.70	7.56	90.26	88.45
MiB	48.07	35.35	52.71	29.92	48.29	37.77	34.34
PLOP	48.58	34.67	53.66	28.66	51.11	34.13	32.48
LwS	60.27	18.94	62.57	16.81	59.78	22.97	**19.57**
Oracle	74.35	-	75.21	-	77.60	-	-

Class incremental scheduling follows the $\mathcal{C}_{bgr} \to \mathcal{C}_{stat} \to \mathcal{C}_{mov}$ arrangement of [40], with the only difference from [40] being that the starting class pool to be split corresponds to the 22 Shift's categories in place of the 19 Cityscape's ones. Results are reported in Table XII, where we compare with MiB and PLOP as CIL competitors, along with fine-tuning baselines. We verify the superiority of our approach in jointly handling class and domain incremental training, as we surpass PLOP by 13 points of $\bar{\Delta}_2$. We once more point out the better stability-plasticity balance reached by our method, which achieves improved performance simultaneously over novel and former domains. Overall, results show that the proposed method is effective under domain shifts of different nature. On the other hand, CIL methods prove to be greatly penalized just from the variable scene illumination in different tasks. We argue that in many real-world applications, such as autonomous driving, it is unrealistic to assume that a continual learner will not experience any sort of alteration in input data distribution, making our continual learning approach much more applicable.

E. Analysis of the Computation Time

Finally, we evaluate the computational requirements of our method. At training time, the additional provisions used and the style transfer slow down the training step rate. Each step requires around 2 s w.r.t. 0.5 s when performing just naïve fine-tuning (the data refers to the ResNet101 backbone on a NVIDIA RTX 3090 GPU). However, notice how the inference time is basically the same of the backbone model and similar to the main competitors: the inference requires only 12 ms when using ErfNet and 60 ms with the ResNet101 backbone. In conclusion, while the approach introduces some training overhead, the trained backbone can be directly used without any additional cost.

VIII. ABLATION STUDIES

In this section, we provide extensive ablation studies to investigate key features of our approach. We will consider the *urban* experimental setup, with CS → BDD → IDD domain and $\mathcal{C}_{bgr} \to \mathcal{C}_{stat} \to \mathcal{C}_{mov}$ class orders, unless otherwise stated.

A. Contribution of Individual Optimization Objectives

We investigate the impact of each of the proposed learning objectives in the overall optimization framework in Table XIII. Just leveraging the currently available training data by fine-tuning (first two rows) yields unsatisfactory results (even

TABLE XIII
ABLATION STUDY ON THE CONTRIBUTION OF LOSS COMPONENTS

CS → BDD → IDD	**IDD**		BDD		CS		
$\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$	$\text{mIoU}_2^2\uparrow$	$\Delta_2^2\downarrow$	$\text{mIoU}_2^1\uparrow$	$\Delta_2^1\downarrow$	$\text{mIoU}_2^0\uparrow$	$\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
$\mathcal{L}_{ce}^{n}$	26.27	61.48	10.47	81.72	12.10	81.18	74.79
$\mathcal{L}_{ce}^{\tilde{n}}$	27.12	60.24	11.51	79.91	13.68	78.72	72.95
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{ce}^{\tilde{o}}$	28.09	58.81	13.32	76.75	16.32	74.61	70.06
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd}^{\tilde{o}}$	40.63	40.43	24.95	56.44	34.14	46.90	47.92
$\mathcal{L}_{ce}^{n}+\mathcal{L}_{kd}^{n}$	43.33	36.47	26.62	53.53	37.36	41.89	43.96
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd}^{\tilde{n}}$	48.12	29.45	32.40	43.44	40.57	36.89	36.59
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd}^{\tilde{n}}+\mathcal{L}_{kd}^{\tilde{o}}$	19.23	71.80	17.68	69.13	24.15	62.44	67.79
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	50.08	26.57	34.16	40.36	42.86	33.33	33.42
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd}^{\tilde{n}}+\mathcal{L}_{ce}^{\tilde{o}}$	50.70	25.66	34.86	39.14	43.04	33.05	32.62
$\mathcal{L}_{ce}^{n}+\mathcal{L}_{kd}^{n}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	46.59	31.69	30.51	46.74	40.44	37.10	38.51
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd}^{\tilde{n}}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	51.20	24.93	35.73	37.62	44.17	31.29	**31.28**
Oracle	68.20	-	57.28	-	64.29	-	-

The $\mathcal{L}_{kd}^{n}$ notation here implies that pseudo-labels are generated leveraging new-domain input samples.

TABLE XIV
ABLATION STUDY ON PSEUDO-LABELING SCHEMES

CS → BDD → IDD	**IDD**		BDD		CS		
$\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$	$\text{mIoU}_2^2\uparrow$	$\Delta_2^2\downarrow$	$\text{mIoU}_2^1\uparrow$	$\Delta_2^1\downarrow$	$\text{mIoU}_2^0\uparrow$	$\Delta_2^0\downarrow$	$\bar{\Delta}_2\downarrow$
$\mathcal{L}_{ce}^{n}+\mathcal{L}_{kd,n}^{n}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	46.59	31.69	30.51	46.74	40.44	37.10	38.51
$\mathcal{L}_{ce}^{n}+\mathcal{L}_{kd,o}^{n}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	40.09	41.22	25.55	55.40	34.90	45.71	47.44
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd,n}^{\tilde{n}}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	51.11	25.06	34.01	40.63	43.96	31.62	32.44
$\mathcal{L}_{ce}^{\tilde{n}}+\mathcal{L}_{kd,o}^{\tilde{n}}+\mathcal{L}_{ce}^{\tilde{o}}+\mathcal{L}_{kd}^{\tilde{o}}$	51.20	24.93	35.73	37.62	44.17	31.29	**31.28**
Oracle	68.20	-	57.28	-	64.29	-	-

We added $\{n,o\}$ to the loss notation to indicate if pseudo-labels are generated leveraging new-domain ($\mathcal{L}_{kd,n}^{d}$) or oldlystylized ($\mathcal{L}_{kd,o}^{d}$) input samples.

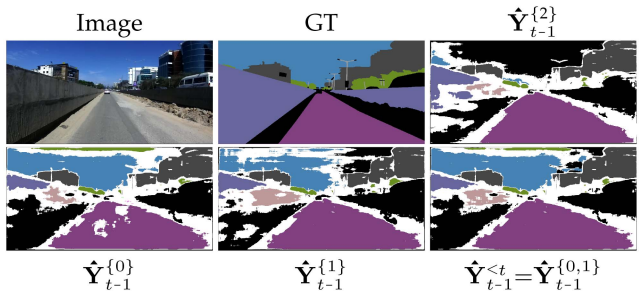

Fig. 6. Different ways of pseudo-labeling ($t=2$). White regions correspond to the *ignore* label.

with self-stylization), leading to catastrophic forgetting of class and domain knowledge. Yet, $\mathcal{L}_{ce}^{n}$ (or $\mathcal{L}_{ce}^{\tilde{n}}$) is essential to learn new tasks, so it will be kept in the following analyses to test multi-term objectives.

By adding a second term in the overall objective (second block of rows) we improve results, especially if the supplemental objective is focused on retaining old-class knowledge. We reach, in fact, the best performance with a 2-term configuration when $\mathcal{L}_{kd}^{\tilde{n}}$ is introduced. This suggests that old-class knowledge preservation is effective even when applied on the new domain, which is directly experienced by means of the available training data. At the same time, the $\mathcal{L}_{kd}^{\tilde{n}}$ objective allows to retain good accuracy w.r.t. past domains, thanks to the improved generalization aptitude promoted by the stylization mechanism, without which (i.e., third row of the block) multiple accuracy points are lost.

When analyzing 3-term objectives (third block of rows), we see noticeable gain with different combinations, except for $\mathcal{L}_{kd}^{\tilde{n}}$ and $\mathcal{L}_{kd}^{\tilde{o}}$ jointly active, where the excessive focus on past-class knowledge preservation generates training instability. In the last row of the block, we clearly see that, by adding the $\mathcal{L}_{ce}^{\tilde{o}}$ loss on top of the best two-term configuration, the incremental learning becomes more robust, with improved final results on all domains.

Finally, we remark that the full framework (last block) yields the best overall performance, with stylization once more playing a substantial role. The overall performance is, in fact, strongly degraded if stylization is turned off, as showed in the second last row.

B. Pseudo-Label Generation

We further analyze the influence exerted by pseudo-labeling in Table XIV. We remark that the proposed enhanced labeling mechanism (described in Section V-C) exploits oldly-stylized images to mitigate the domain shift endured by the frozen segmentation model distilling knowledge from the past.

We notice that when self-stylization is disabled (first two rows) the efficacy of our method is reduced, while the beneficial effect offered by the self-stylizing module can be appreciated in the last two rows. This occurs because self-stylization better prepares the segmentation model for future steps, in which the stylizing mechanism leverages old-domain styles to inject old-domain knowledge into the ongoing learning step. In other words, when self-stylizing images, what will be experienced as an *old* style will have already been experienced as a *new* style before. Therefore, the undesired visual artifacts generated by style transfer are experienced by the network from the very first step in which each domain is introduced. This, in turn, ensures greater robustness over the incremental learning process. Furthermore, in setups with self-stylization, as opposed to what occurs without it, pseudo-labeling performed on top of oldly-stylized images yields the best overall performance, if compared to the same labeling process executed over image samples with new-domain style. This happens because the network (frozen from the past step) used to generate pseudo-labels is better equipped to face input distributions of old domains, while it may suffer from domain shift when presented with new unseen input distributions.

In Fig. 6 we report pseudo-labels generated according to different criteria, to provide visual confirmation of the improved pseudo-supervision achieved on top of the oldly stylization. The considered setup involves CS → BDD → IDD and $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$ progressions, and maps are retrieved at the last step (i.e., $t=2$). We observe that the segmentation model taken from step $t-1$ (i.e., second last step) is not detecting the sky region of the new-domain image, i.e., $\hat{\mathbf{Y}}_{t-1}^{\{2\}}$ provides unreliable supervision by labeling the top portion of the picture as *unknown* (when the true *sky* class is among those already seen). On the other hand, when leveraging oldly-stylized images to generate

TABLE XV
ABLATION STUDY ON STYLIZATION ($\beta = 0.01$ CORRESPONDS TO THE DEFAULT CONFIGURATION)

CS → BDD → IDD $\mathcal{C}_{bgr} \rightarrow \mathcal{C}_{stat} \rightarrow \mathcal{C}_{mov}$			No stylization	β 0.001	β 0.01	β 0.1
Step 0	$\text{mIoU}_0\uparrow$	**CS**	79.67	79.8	79.19	78.54
	$\bar{\Delta}_0\downarrow$		5.32	**5.17**	5.89	6.66
Step 1	$\text{mIoU}_1\uparrow$	**BDD**	33.67	35.06	44.47	44.79
		CS	49.20	43.75	53.31	50.45
	$\bar{\Delta}_1\downarrow$		38.08	40.88	**26.58**	28.37
Step 2	$\text{mIoU}_2\uparrow$	**IDD**	43.33	48.60	51.20	50.03
		BDD	26.62	27.77	35.73	34.84
		CS	37.36	37.61	44.17	43.01
	$\bar{\Delta}_2\downarrow$		43.96	40.59	**31.28**	32.97

pseudo-supervision ($\mathbf{Y}^{\blacktriangle <t}_{t-1}$), more reliable old-domain guidance ($\overset{\blacktriangle}{\mathbf{Y}}{}^{\{0\}}_{t-1}$ and $\overset{\blacktriangle}{\mathbf{Y}}{}^{\{1\}}_{t-1}$) is exploited, with individual positive contributions successfully merged in the final map (e.g., in *sky* and *road* regions). Thus, we end up with $\overset{\blacktriangle}{\mathbf{Y}}{}^{<1}_{t-1}$ being more accurate than each domain-specific alternative $\overset{\blacktriangle}{\mathbf{Y}}{}^{\{k\}}_{t-1} k \leq t$.

C. Degree of Stylization

We propose an additional analysis on the stylization mechanism. Table XV shows the results of the our method (complete with all objectives) under different degrees of stylization, which are determined by the β parameter (see Section IV). We notice that disabling stylization or operating it in a more conservative manner (i.e., with $\beta = 0.001$) yields low results, with the latter configuration still outperforming the no stylization approach, as the statistical properties captured and transferred are not sufficient to successfully retain old-domain information. On the other hand, if the stylization is raised to an excessive extent (i.e., with $\beta = 0.1$), we observe performance degradation on the overall $\bar{\Delta}_2$ score. In this scenario, artifacts are more likely to be introduced on oldly-stylized images, thus hindering the segmentation task.

D. Knowledge Transfer Across Tasks and Domains

We propose further ablation studies to evaluate the knowledge transfer aptitude of our method, both under task and domain perspectives. Fig. 7 presents a comparative of multiple CIL competitors in terms of predisposition towards *domain-knowledge transfer;* we report the mIoU achieved on individual domains only on classes experienced so far across multiple steps in matrix form. We consider multiple incremental setups, with urban datasets and variable domain order. We observe that our approach, right from the first learning step, achieves better *forward transfer* to future domains, as indicated by per-domain mIoU values in the top triangular sections, regardless of the setup considered. At the same time, this translates into superior performance on current domains (represented by diagonal mIoU values), as they benefit from a better forward-adaptability acquired before. Plus, improved *backward transfer* to former domains is testified by higher mIoU values in the bottom triangular part of matrices.

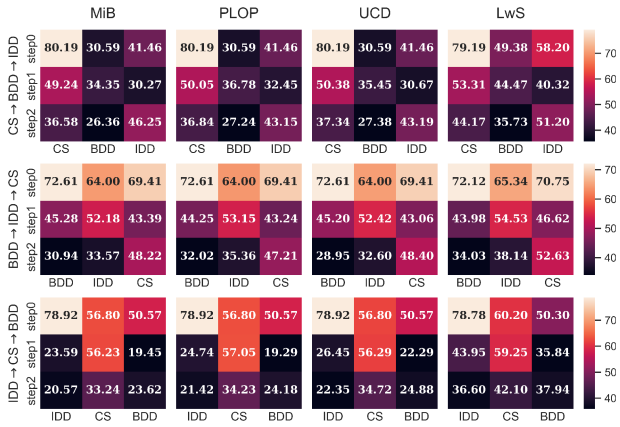

Fig. 7. Domain-knowledge transfer (mIoU↑ (%)).

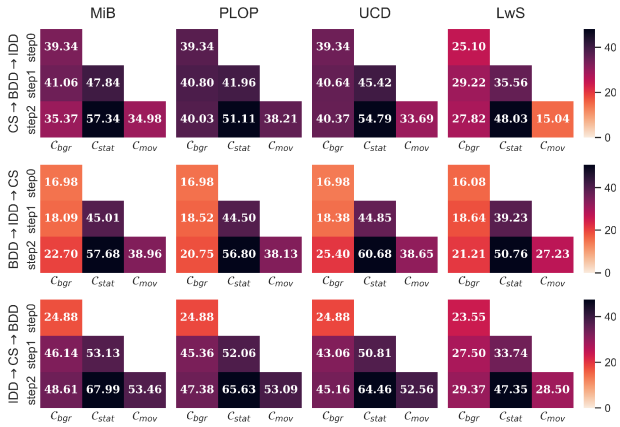

Fig. 8. Task-knowledge transfer ($\bar{\Delta}\downarrow$).

To provide an insight on *task-knowledge transfer* proneness of different incremental methods, in Fig. 8 we report a comparative in terms of $\bar{\Delta}$ results at multiple learning steps; values are computed on *single* incremental sets of classes and represent an average score across all domains (both experienced and future ones). The experimental setups are the same considered when studying domain transfer, and results are arranged in matrix form. We observe that our $\bar{\Delta}$ scores in the bottom triangular part of matrices are lower than competitors, suggesting that our method yields better *backward transfer* in terms of task knowledge. At the same time, the smaller $\bar{\Delta}$ diagonal elements indicate improved performance on current tasks, confirming the better stability-plasticity compromise offered by our approach.

IX. CONCLUSION

In this paper, we formalized a general setting for continual learning, where both domains and tasks to be learned incrementally change over time. We addressed this under-explored learning setting targeting the semantic segmentation task by breaking it down into underlying sub-problems, each tackled with a specific learning objective. Leveraging a stylization mechanism, domain knowledge is replayed over time, whereas a

robust distillation mechanism allows to retain and adapt old-task information. Overall, the proposed learning framework enables learning new tasks, while preserving performance on old ones and spreading task knowledge across all the encountered domains. We achieved significant results outperforming state-of-the-art competitors on multiple challenging benchmarks. Further research will tackle even more application-oriented settings, i.e., where task and domain shifts happen in a continuous fashion rather than in discrete steps and distinct overlapping sets of classes are introduced in different domains. We will also perform a more in-depth investigation of alternative style transfer techniques. Finally, the extension of the framework to applications beyond driving scenarios will be considered.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[3] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[4] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, pp. 3521–3526, 2017.

[5] M. De Lange et al., "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[6] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[7] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5533–5542.

[8] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3420–3429.

[9] C. Peng, K. Zhao, and B. C. Lovell, "Faster ILOD: Incremental learning for object detectors based on faster RCNN," *Pattern Recognit. Lett.*, vol. 140, pp. 109–115, 2020.

[10] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5826–5836.

[11] U. Michieli and P. Zanuttigh, "Incremental learning techniques for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3205–3212.

[12] F. Cermelli, M. Mancini, S. R. Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9230–9239.

[13] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "PLOP: Learning without forgetting for continual semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4039–4049.

[14] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3723–3732.

[15] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2512–2521.

[16] Y. Yang and S. Soatto, "FDA: Fourier domain adaptation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4084–4094.

[17] R. Volpi, D. Larlus, and G. Rogez, "Continual adaptation of visual representations via domain randomization and meta-learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4441–4451.

[18] R. Volpi, P. De Jorge, D. Larlus, and G. Csurka, "On the road to online adaptation for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19162–19173.

[19] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7191–7201.

[20] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, Jul. 2022.

[21] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2016, *arXiv: 1511.07122*.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.

[23] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv: 1805.10180*.

[24] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, arXiv: *1704.04861*.

[25] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[26] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, 2021, pp. 87–110.

[27] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 831–839.

[28] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 86–102.

[29] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3013–3022.

[30] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5867–5876.

[31] M. Toldo and M. Ozay, "Bring evanescent representations to life in lifelong class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16711–16720.

[32] K. Zhu, W. Zhai, Y. Cao, J. Luo, and Z. Zha, "Self-sustaining representation expansion for non-exemplar class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9286–9295.

[33] T.-Y. Wu et al., "Class-incremental learning with strong pre-trained models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9591–9600.

[34] J. Xie, S. Yan, and X. He, "General incremental learning with domain-aware categorical representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14331–14340.

[35] Y.-M. Tang, Y.-X. Peng, and W.-S. Zheng, "Learning to imagine: Diversify memory for incremental learning using unlabeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9539–9548.

[36] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "DyTox: Transformers for continual learning with dynamic token expansion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9275–9285.

[37] B. Yang et al., "Continual object detection via prototypical task correlation guided gating mechanism," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9245–9254.

[38] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, arXiv: *1503.02531*.

[39] U. Michieli, M. Toldo, and P. Zanuttigh, "Domain adaptation and continual learning in semantic segmentation," in *Proc. Adv. Methods Deep Learn. Comput. Vis.*, 2022, pp. 275–303.

[40] M. Klingner, A. Bär, P. Donn, and T. Fingscheidt, "Class-incremental learning for semantic segmentation re-using neither old data nor old labels," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–8.

[41] A. Maracani, U. Michieli, M. Toldo, and P. Zanuttigh, "RECALL: Replay-based continual learning in semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 7006–7015.

[42] U. Michieli and P. Zanuttigh, "Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1114–1124.

[43] G. Yang et al., "Uncertainty-aware contrastive distillation for incremental semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2567–2581, Feb. 2023.

[44] G. Yang et al., "Continual attentive fusion for incremental learning in semantic segmentation," *IEEE Trans. Multimedia*, vol. 25, pp. 3841–3854, 2022.

[45] F. Cermelli, D. Fontanel, A. Tavera, M. Ciccone, and B. Caputo, "Incremental learning in semantic segmentation from image labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4361–4371.
[46] C.-B. Zhang, J.-W. Xiao, X. Liu, Y.-C. Chen, and M.-M. Cheng, "Representation compensation networks for continual semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7043–7054.
[47] M. H. Phan, S. L. Phung, L. Tran-Thanh, and A. Bouzerdoum, "Class similarity weighted knowledge distillation for continual semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16845–16854.
[48] U. Michieli and P. Zanuttigh, "Knowledge distillation for incremental learning in semantic segmentation," *Comput. Vis. Image Understanding*, vol. 205, 2021, Art. no. 103167.
[49] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 137–144.
[50] P. Testolina, F. Barbato, U. Michieli, M. Giordani, P. Zanuttigh, and M. Zorzi, "SELMA: SEmantic large-scale multimodal acquisitions in variable weather, daytime and viewpoints," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7012–7024, Jul. 2023.
[51] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation: A review," *Technol.*, vol. 8, no. 2, 2020, Art. no. 35.
[52] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12409–12419.
[53] L. Hoyer, D. Dai, and L. Van Gool, "DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9914–9925.
[54] Y. Tian and S. Zhu, "Partial domain adaptation on semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3798–3809, Jun. 2022.
[55] T. Jing, H. Liu, and Z. Ding, "Towards novel target discovery through open-set domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9302–9311.
[56] K. Saito and K. Saenko, "OVANet: One-vs-all network for universal domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8980–8989.
[57] X. Ma, J. Gao, and C. Xu, "Active universal domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8948–8957.
[58] J. He, X. Jia, S. Chen, and J. Liu, "Multi-source domain adaptation with collaborative learning for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11003–11012.
[59] R. Gong, D. Dai, Y. Chen, W. Li, and L. Van Gool, "mDALU: Multi-source domain adaptation and label unification with partial datasets," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8856–8865.
[60] Z. Liu et al., "Open compound domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12403–12412.
[61] T. Isobe et al., "Multi-target domain adaptation with collaborative consistency learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8183–8192.
[62] Y. Zhao, Z. Zhong, Z. Luo, G. H. Lee, and N. Sebe, "Source-free open compound domain adaptation in semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 7019–7032, Oct. 2022.
[63] R. A. Marsden, F. Wiewel, M. Döbler, Y. Yang, and B. Yang, "Continual unsupervised domain adaptation for semantic segmentation using a class-specific transfer," in *Proc. Int. Joint Conf. Neural Netw.*, 2023, pp. 1–8.
[64] P. Garg, R. Saluja, V. N. Balasubramanian, C. Arora, A. Subramanian, and C. V. Jawahar, "Multi-domain incremental learning for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2080–2090.
[65] T. Kalb, M. Roschani, M. Ruf, and J. Beyerer, "Continual learning for class- and domain-incremental semantic segmentation," in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 1345–1351.
[66] D. Shenaj, F. Barbato, U. Michieli, and P. Zanuttigh, "Continual coarse-to-fine domain adaptation in semantic segmentation," *Image Vis. Comput.*, vol. 121, 2022, Art. no. 104426.
[67] C. Simon et al., "On generalizing beyond domains in cross-domain continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9255–9264.
[68] M. Cordts et al., "The CityScapes dataset for semantic urban scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.
[69] F. Yu et al., "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2633–2642.
[70] G. Varma, A. Subramanian, A. M. Namboodiri, M. Chandraker, and C. V. Jawahar, "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1743–1751.
[71] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The Mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5000–5009.
[72] T. Sun et al., "SHIFT: A synthetic driving dataset for continuous multi-task domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 21339–21350.
[73] M. Wrenninge and J. Unger, "Synscapes: A photorealistic synthetic dataset for street scene parsing," 2018, arXiv: *1810.08705*.
[74] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, arXiv: *1706.05587*.
[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[76] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv: 1412.6980*.

Marco Toldo (Student Member, IEEE) received the PhD degree in information engineering from the University of Padova, in 2022. In 2021, he interned for eight months as Research Engineer with Samsung Research U.K. His research concerns transfer learning applied to computer vision, with a focus on domain adaptation, continual learning, and federated learning.

Umberto Michieli (Graduate Student Member, IEEE) received the PhD degree in information engineering from the University of Padova, in 2021. He spent research periods with TU Dresden and Samsung Research, U.K. He is currently a post-doctoral researcher and an adjunct professor with the University of Padova. His research interests include the intersection of foundation AI problems applied to semantic understanding. In particular, he focuses on domain adaptation, continual learning, coarse-to-fine learning, and federated learning.

Pietro Zanuttigh (Member, IEEE) received the PhD degree from the University of Padova, Italy, in 2007 where he is currently an associate professor with the Department of Information Engineering. His research interests include domain adaptation and continual learning in semantic segmentation, federated learning, computational imaging, Time-oF-Flight sensors, and hand gesture recognition.