Online Distillation with Continual Learning for Cyclic Domain Shifts

Joachim Houyon[1,*] Anthony Cioppa[1,2,*] Yasir Ghunaim[2] Motasem Alfarra[2]
Anaïs Halin[1] Maxim Henry[1] Bernard Ghenem[2] Marc Van Droogenbroeck[1]
[1] University of Liège [2] KAUST

Abstract

In recent years, online distillation has emerged as a powerful technique for adapting real-time deep neural networks on the fly using a slow, but accurate teacher model. However, a major challenge in online distillation is catastrophic forgetting when the domain shifts, which occurs when the student model is updated with data from the new domain and forgets previously learned knowledge. In this paper, we propose a solution to this issue by leveraging the power of continual learning methods to reduce the impact of domain shifts. Specifically, we integrate several state-of-the-art continual learning methods in the context of online distillation and demonstrate their effectiveness in reducing catastrophic forgetting. Furthermore, we provide a detailed analysis of our proposed solution in the case of cyclic domain shifts. Our experimental results demonstrate the efficacy of our approach in improving the robustness and accuracy of online distillation, with potential applications in domains such as video surveillance or autonomous driving. Overall, our work represents an important step forward in the field of online distillation and continual learning, with the potential to significantly impact real-world applications.

1. Introduction

Deep Neural Networks (DNNs) have shown remarkable performance on various computer vision tasks thanks in part to the assumption that the training and testing data are identically distributed [21, 27, 37]. However, DNNs' performance degrade significantly when tested on out-of-distribution data, such as testing data that contains domain shifts relative to the training data [22, 23]. Even worse, DNNs tend to forget previously learned distributions when learning continually on a stream of tasks [24]. This performance loss is a major concern because domain shifts are likely to occur in real-world deployments due to changes in brightness between day and night, weather conditions across seasons, and sensor perturbations [35]. Therefore, it is essential to develop algorithms that can enable DNNs to adapt to such domain shifts and maintain high performance in real-world settings.

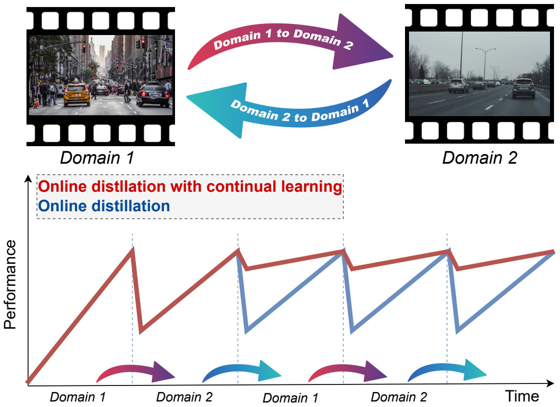

Figure 1. **Online distillation with continual learning.** When cyclic domain shifts occur in long videos, the online distillation framework proposed by Cioppa *et al.* [10] forgets the previously acquired knowledge as it fine-tunes on the current domain. In this work, we study the inclusion of state-of-the-art continual learning methods inside the online distillation framework to mitigate this catastrophic forgetting around the domain shifts.

Continual learning aims at building machine learning models that can learn from a continuous stream of data without forgetting previously learned knowledge [9, 20, 26]. We investigate a practical scenario of online continual learning [7]. Specifically, we consider cyclic domain shifts where a stream of data consistently alternates in revealing new *unlabeled* data from one of two distributions for a period of time. For instance, consider an autonomous driving system that frequently travels between cities and countrysides, where the distribution of instances varies between the two scenes. Such domain variation can cause the online learner to fail in adapting to this distribution shift, raising concerns about the real-world deployment of such systems. While online continual learning has been studied in

(*) Equal contributions
Contacts: joachim.jouyon@student.uliege.be, anthony.cioppa@uliege.be.
Data/code available at github.com/Houyon/online-distillation-cl.

several contexts, such as domain incremental learning [18], unsupervised domain adaptation [39], and test-time adaptation [41], these works typically analyze the more general, and potentially less realistic, setup where domain variations are unconditional. Our focus on cyclic domain shifts enables us to explore a pragmatic setting and develop novel algorithms that can better adapt to these changes.

In this work, we propose a novel approach to address the challenge of adapting to cyclic domain shifts in the context of online domain incremental learning. Specifically, we employ a previously published real-time online distillation technique [10] to learn from the unlabeled cyclic stream of data. Online distillation asynchronously updates a student-teacher based approach on the received data, which enables the model to continually learn from new data. However, we found that the cyclic domain shift can cause the student to forget the previously learned domain, leading to a significant loss in performance. To mitigate this undesirable effect, we combine online distillation with state-of-the-art continual learning as shown in Figure 1, leveraging both regularization- and replay-based approaches from the continual learning literature. Our proposed approach enables the student to effectively adapt to cyclic domain shifts and maintain high performance over time, making it suitable for real-world deployment.

Contributions. We summarize our contributions in two points: **(i)** We define the cyclic online continual learning problem setup and propose corresponding evaluation metrics. **(ii)** We combine online distillation with both regularization- and replay-based continual learning approaches to better learn on cyclic domains. We conduct experiments on the proposed stream where we show that our approach mitigates the forgetting of the original online distillation framework.

2. Related Work

Domain shifts. A domain shift is a change in the statistical distribution of data between different domains [15]. This phenomenon is commonly observed at test time in open-world scenarios [4, 19, 28, 42]. In autonomous driving, domain shift can be caused by many diverse factors [40], such as different environments (*e.g*., rural or urban roads), lighting conditions (*e.g*., day or night), weather conditions (*e.g*., sunny or snowy) [35], traffic conditions or even differences in the appearance of roads or traffic signs across different countries [44]. However, it is crucial for autonomous vehicles to have algorithms that are robust to these dynamic domain shifts in order to constantly be able to perceive and understand their surrounding environment to avoid obstacles. Domain adaptation is an active area of research that aims at addressing the domain shift problem, especially in open-world applications such as autonomous vehicles [25, 30, 32, 33, 40], where data is collected in a highly dynamic environment. In this work, we study the particular case of cyclic domain shifts in the field of autonomous driving, where the domains can be represented as a succession of *highway* and *downtown* driving conditions.

Online distillation. In the field of deep neural networks, there is a trade-off between speed, performance, and generalizability across multiple domains. While the best-performing models often exhibit high performance across diverse domains, they tend to be memory-greedy for embedded systems or too slow for use in real-time applications [43, 46, 47]. In contrast, lightweight and fast networks show good performance on smaller domains but lack generalizability [12]. To address this issue, Cioppa *et al*. [10] proposed an online distillation approach for videos, that enables the online training of a lightweight student network using a slower, larger teacher model. At test time, the teacher provides pseudo ground truths to the student, allowing it to specialize in the specific domain being analyzed. The student model therefore adapts to changing video conditions, even matching the performance of the slower teacher. This online distillation approach may be used for different tasks such as semantic segmentation [10] or multi-modal object detection [11]. However, this technique experiences a temporary loss of performance during domain shifts. In this paper, we investigate several continual techniques to mitigate the effects of catastrophic forgetting in online distillation, particularly in cases of cyclic domain shifts. We combine online distillation with both regularization- and replay-based approaches for a better continual learning scheme.

Continual learning. Continual Learning (CL) aims at learning from data arriving as a stream with changing distribution [16, 31]. However, this learning paradigm face the catastrophic forgetting challenge, that is, previously learned knowledge is forgotten when adapting to the newly arriving data samples [9, 24]. One approach of mitigating the forgetting effect is regularizing the training process through constraining the changes of important network parameters [2, 8, 24] or performing knowledge distillation [17, 26, 38]. Alternatively, replay-based methods rehearse previously seen examples by storing a subset of the observed data in a replay buffer [3, 9, 29, 34]. While both approaches were originally proposed for class-incremental setup and classification task, they were recently extended to the more realistic domain incremental setup and the more challenging semantic segmentation task [1, 18]. Nevertheless, prior art assumes fully supervised setups where the stream reveals labeled data for the student learner. To that end, we analyze the domain incremental setup for semantic segmentation under an unsupervised setup.

3. Methodology

In this section, we first describe online distillation in a mathematical framework suited for continual learning. Next we detail the regularization-based and replay-based continual learning methods that we integrate into the online distillation framework. Finally, we explain how to evaluate and benchmark online continual leaning methods under our cyclic stream.

3.1. Online distillation framework

The online distillation framework proposed by Cioppa *et al.* [10] allows a real-time network to adapt to domain shifts at test time. Formally, given a long untrimmed video $\mathcal{V}$ composed of a stream of frames x_i produced at a rate $r_{\mathcal{V}}$ and a task $\mathcal{T}$ (*e.g.*, object detection, semantic segmentation, *etc.*), the objective is to produce a stream of predictions $\hat{y}_i$ for each frame x_i in real time (*i.e.*, at a rate $r_{\mathcal{V}}$). To do so, the authors leverage a student-teacher architecture with a fast and slow route. In the fast route (inference), a student network $\mathbf{S}$ computes $\hat{y}_i = \mathbf{S}(x_i)$ at the rate $r_{\mathcal{V}}$. In parallel in the slow route (training), a slower but high-performance frozen teacher network $\mathbf{T}$ produces pseudo ground-truths $\tilde{y}_{i'} = \mathbf{T}(x_{i'})$ at an asynchronous slower rate $r_{\mathbf{T}}$ on a subset of $\mathcal{V}$. Each new pair $(x_{i'}, \tilde{y}_{i'})$ is then stored through an update function f_U into an online dataset $\mathcal{D}$ of size N that is used to train a copy $\mathbf{S}_c$ of the student network. In the original framework, f_U is chosen as a First In First Out (FIFO) algorithm. Iteratively, $\mathbf{S}_c$ is trained on selected samples extracted from $\mathcal{D}$ by a function f_S, by minimizing the loss:

$$\mathcal{L} = \sum_{n=1}^{N} L(\mathbf{S}_c(x_n), \tilde{y}_n) \ ,$$

where L is a distance function suited to learn task $\mathcal{T}$. In the original framework, f_S selects all pairs in $\mathcal{D}$ one time. The parameters of $\mathbf{S}$ are updated by copying the parameters θ of $\mathbf{S}_c$ at the rate $r_{\mathbf{S}_c}$, corresponding to the inverse of the training time of $\mathbf{S}_c$ on one epoch of $\mathcal{D}$. The complete pipeline may be found in Figure 2.

Thanks to this framework, $\mathbf{S}$ becomes specialized to the last minutes of the particular video it is analyzing. This allows it to adapt to slowly changing domains in $\mathcal{V}$ as long as $\mathbf{T}$ is able to produce reliable predictions. However, this continual fine-tuning makes it forget previously acquired knowledge over time. For instance, when sudden shifts in domain occurs, $\mathbf{S}$ needs several updates to recover good performance even if the same domain already appeared in the video. In the following, we propose to incorporate Continual Learning (CL) techniques in the existing online distillation framework to minimize the catastrophic forgetting of previously acquired knowledge in the case of cyclic domain shifts. In particular, we benchmark several replay-based methods (CL_{Rep}) that act on $\mathcal{D}$ and regularization-based methods (CL_{Reg}) that act on $\mathcal{L}$ as shown in Figure 2.

3.2. Replay-based methods

This set of methods leverage a replay buffer (*i.e.* a collection of data and corresponding ground-truth labels) of finite size that is accessed by the selection function f_S and updated with new data by an update function f_U at each training epoch. The online distillation framework presented above can be formulated as a replay-based method, where the replay buffer corresponds to $\mathcal{D}$, the labels are the pseudo ground-truth predictions $\tilde{y}_n$, f_S selects all data of the replay buffer to be used during the training epoch, and f_U determines the policy to update samples in the replay buffer. In the original online distillation framework, the size of the replay buffer is also the number of samples, N, passed to the model at each training step. We extend the replay buffer to include $M \geq N$ samples where we sample N samples without replacement from the buffer at each training step. We augment the selected samples with the new incoming data from the stream.

We consider several strategies to modify f_U and f_S to reduce the catastrophic forgetting: FIFO, Uniform, Prioritized, and MIR.

FIFO: f_U stores the most recent samples in the replay buffer while removing oldest ones. This is equivalent to the original framework's update strategy that is used as a baseline for comparison with other methods.

Uniform: f_U stores incoming data at randomly selected replay buffer indices. This strategy leads to an expected remaining lifespan of data to decay exponentially [5], which could avoid forgetting. As for memory selection f_S, it performs a random selection from memory for constructing a training batch.

Prioritized: Adapting the work of Schaul *et al.* [36] on reinforcement learning, we set f_U to assign an importance score $\mathcal{I}$ for each sample in the replay buffer following:

$$\mathcal{I}_n = L(\mathbf{S}(x_n), \mathbf{T}(x_n)) \ .$$

The importance score is then used as a probability of determining which samples to remove from the replay buffer following:

$$p_n = \frac{\mathcal{I}_n^{-1}}{\sum_{n'=1}^{M} \mathcal{I}_{n'}^{-1}} \ .$$

To perform the memory selection f_S operation, prioritized follows the same strategy described above for the update function f_U.

MIR [3]: is a selection function f_S that selects a subset of the replay buffer samples that are maximally interfered by the incoming data in a stream. In other words, it constructs a set of training samples from memory that are negatively affected the most by the next parameter update.

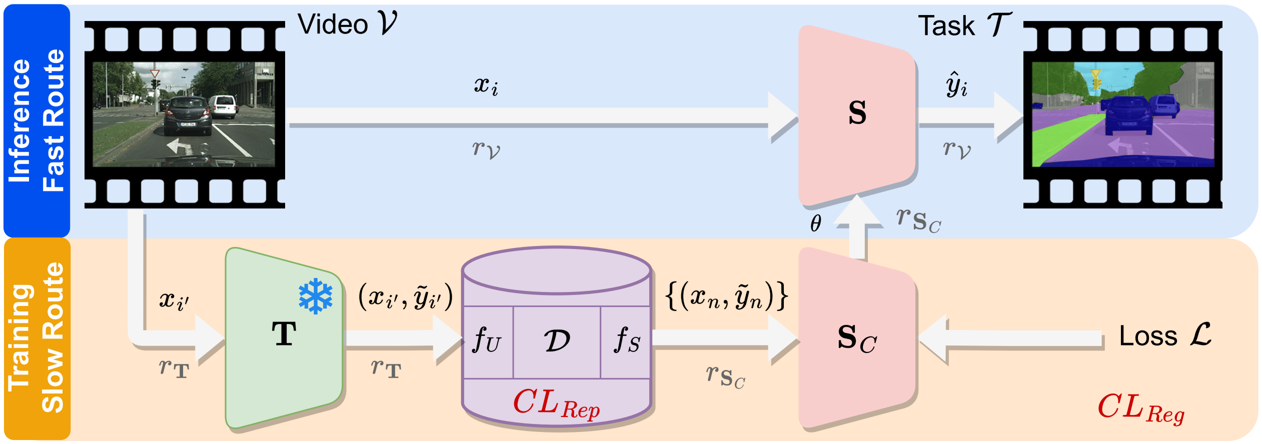

Figure 2. **Online distillation**. The framework is composed of a fast and a slow route. In the fast route (inference), the video stream $\mathcal{V}$ is processed by a student network $\mathbf{S}$ on a task $\mathcal{T}$ (*e.g.*, semantic segmentation for autonomous driving) and produces predictions $\hat{y}_i$ for each frame of the video x_i at the original video rate $r_{\mathcal{V}}$ (*i.e.*, in real time). In parallel in the slow route (training), a frozen teacher $\mathbf{T}$ produces pseudo ground-truths $\tilde{y}_{i'}$ from a subset of frames $x_{i'}$ at a slower rate $r_{\mathbf{T}}$. The pair $(x_{i'}, \tilde{y}_{i'})$ are then stored in an online dataset (or replay buffer) $\mathcal{D}$ through an update function f_U. $\mathcal{D}$ is sampled through a selection function f_S and the selected pairs $(x_n, \tilde{y}_n)$ are used to train a copy of the student network $\mathbf{S}_c$ for one epoch using a loss $\mathcal{L}$. The parameters θ of $\mathbf{S}_c$ are then transferred to $\mathbf{S}$ at a rate $r_{\mathbf{S}_c}$ (corresponding to the inverse of the training time of $\mathbf{S}_c$ on one epoch) so that $\mathbf{S}$ improves on the latest domain of $\mathcal{V}$. One of the contribution of our paper consists in including replay-based Continual Learning (CL) methods, CL_{Rep}, inside $\mathcal{D}$ and regularization-based methods, CL_{Reg}, on $\mathcal{L}$.

3.3. Regularization-based methods

Regularization-based methods mitigate forgetting by adding a regularization term to the training loss function $\mathcal{L}$. Generally, this can be formulated as:

$$\mathcal{L} = \sum_{n=1}^{N} L(\mathbf{S}_c(x_n), \tilde{y}_n) + \mathcal{R} \ ,$$

where $\mathcal{R}$ is a method-specific regularization term. In this paper, we consider four different regularization-based continual learning methods: ER-ACE [6], LwF [26], MAS [2], and RWalk [8]. We summarize these methods hereafter.

ER-ACE [6] aims at reducing the changes in the learned representation when training on samples from a new class. It does so by applying an asymmetric parameter update on the incoming data and the previously seen data that are sampled from a replay buffer. Specifically, ER-ACE restricts the loss computation on classes presented in the incoming data while ignoring remaining classes. We note that ER-ACE only works on incoming data while keeping the original loss on the data sampled from replay buffer.

The following methods were originally proposed for settings with clear task boundaries. We adopt them to work on online streams without task boundaries by using two properties: **(i)** warmup and **(ii)** update frequency. The warmup defines a time period for the network to be initialized during the warmup phase, we set $\mathcal{R} = 0$. The update frequency simulates an artificial task boundary after every k steps, where k is a fixed hyperparameter for all methods.

LwF [26] uses knowledge distillation to encourage the current network's output to resemble that of a network trained on data from previous time steps. In our setup, LwF keeps a previous version of our student network $\mathbf{S}_c$ to guide the future parameter updates of this network. Maintaining a previous network that is potentially more tailored to previous domains could help in preserving learned knowledge.

MAS [2] assigns an importance weight for each network parameter by approximating the sensitivity of the network output to a parameter change. When training on new distributions, it penalizes large changes to important parameters and, thus, preserves previously learned knowledge.

RWalk [8] is a generalized formulation that combines a modified version of the two popular importance-based methods: EWC [24] and PI [45]. RWalk computes importance scores for network parameters, similar to MAS, and regularizes over the network parameters.

3.4. Evaluation methodology

To evaluate the adaption to new domains and the forgetting of past domains, we propose several evaluation metrics. Following the work of Cioppa *et al.* [10], the performance of the student network $\mathbf{S}_c$ (equivalent to $\mathbf{S}$) over time is defined as follows: given a task-specific metric $\mathcal{M}$ (*e.g.*, $mIoU$ for semantic segmentation or $accuracy$ for classification), a set of size I of frames $X'_i = \{x_{i'}, ..., x_{i'+I}\}$ and pseudo ground truths $\tilde{Y}'_i = \{\tilde{y}_{i'}, ..., \tilde{y}_{i'+I}\}$, the performance of the student network at time i' is given by:

$$\mathcal{M}(\mathbf{S}_c(X_{i'};\theta_{i'}), \tilde{Y}_{i'}) \ ,$$

where $\theta_{i'}$ are the parameters of $\mathbf{S}_c$ at time i', which may be asynchronous with the training of $\mathbf{S}_c$ and update of $\mathbf{S}$ as it operates at a the different rate $r_{\mathbf{S}_c}$.

Backward Transfer (BWT): Motivated by the discrete implementation of backward transfer [14], we propose a modified version for online streams that measures forgetting of the current student network with respect to previous data, which corresponds to the previous domain in our case:

$$\text{BWT}(i') = \mathcal{M}(\mathbf{S}_c(X_{i'-h};\theta_{i'}), \tilde{Y}_{i'-h}) \ ,$$

where h refers to the backward time shift.

In addition, we report the **Final Backward Transfer (Final BWT)**. Given a stream of length K, we evaluate the backward transfer of the final model θ_K on the entire stream, *i.e*. setting $h = 0$ in BWT. This metrics allows to evaluate the final student model on all previous domains, rather than only one specific past domain.

Forward Transfer (FWT): Similar to the backward transfer, we adapt the discrete version [14] of forward transfer for our online setup as follows:

$$\text{FWT}(i') = \mathcal{M}(\mathbf{S}_c(X_{i'+h};\theta_{i'}), \tilde{Y}_{i'+h}) \ .$$

Forward transfer measures the model's performance on future unseen data. In our case, this metric is useful in evaluating the current model on the next domain.

4. Experiments

In this section, we first describe the experimental setup on which we benchmark our continual online distillation framework. Next, we provide quantitative results including a comparative study, of our framework using our proposed evaluation methodology. Finally, we display some qualitative results to show the practical impact for autonomous driving applications.

4.1. Experimental setup

Our online continual learning framework is agnostic to the task, metric, and training parameters. In this section, we provide the technical details describing our experiments in various settings.

Task. We benchmark our framework on the outdoor semantic segmentation task, which consists in assigning a class label to each pixel of a frame. We study the particular case of videos taken behind the windshield of vehicles, which is the typical study-case for autonomous driving applications.

Dataset. The online distillation framework requires long untrimmed videos, in our case containing cyclic domain shifts. Additionally, these videos must be relevant to highlight the task's objectives. Since most datasets for semantic segmentation are composed of frames or small video clips (*e.g*., CityScapes [13], BDD100K [44], *etc*.), they cannot be used in our context of online continual learning. Hence, we follow the same strategy to simulate long videos with domain shifts as in [10] and propose to artificially construct a video $\mathcal{V}$ by concatenating sequences from 2 different domains, D^A and D^B, alternating in cycle from one domain to the other. The resulting video is therefore an ordered set $\mathcal{V} = \{\mathcal{V}_1^A, \mathcal{V}_1^B, \mathcal{V}_2^A, \mathcal{V}_2^B, ...\}$, where the $\mathcal{V}_i^A$ and $\mathcal{V}_i^B$ are sequences from domain D^A and domain D^B, respectively. In our autonomous driving case, we define the two domains D^A and D^B as a highway environment and a downtown environment, which differ from the priors on the semantic classes (*e.g*., there should be fewer persons in highways than downtown) or the background (*e.g*., there are more buildings in downtown and more empty spaces in highways). We extract several clips from each domain and alternatively concatenate them to build $\mathcal{V}$. To consider clips of different time lengths, we construct two video $\mathcal{V}$ streams where the extracted clips are 20 minutes and 40 minutes long respectively.

Evaluation metric. Following the standards in semantic segmentation, we use $\mathcal{M} = mIoU$ to evaluate the segmentation masks of each frame as described in Section 3. Following the work of Cioppa *et al*. [10], since ground-truth data is unavailable for our dataset, we evaluate the performance of the student with respect to the pseudo ground truths produced by the teacher. This evaluates the capacity of the student to imitate the teacher. We provide the $mIoU$, FWT, BWT, Final BWT metrics either during the video or averaged over the entire video (referred as mean). We choose $I = 1$ minute and $h = 20$ minutes or $h = 40$ minutes depending on the domain sequences length to evaluate the forgetting on the previous or future domains. Finally, we also compute the average across a time window of ± 2 minutes of each domain shift occurrence. We call this metric $mIoU$ Near Domain Shifts ($mIoU$ NDS).

Networks and training parameters. For the teacher network $\mathbf{T}$, we chose SegFormer [43] trained on the CityScapes dataset, which is the state of the art in semantic segmentation on this dataset. For the student networks $\mathbf{S}$ and $\mathbf{S}_c$, we chose TinyNet [10, 12], a lightweight segmentation network that only needs a few training samples to specialize on a particular domain, that is fast to train, and operates in real time (at least 30 frames per second for full-HD videos on a Nvidia 1080 GPU). The student network $\mathbf{S}_c$ is trained from scratch at the beginning of the video using a learning rate of 10^{-4} and ADAM optimizer for online learning following [10]. The replay buffer size is set to $M = 250$ and the number of selected frames to $N = 100$ frames. Given the chosen video, networks, and replay buffer size,

Table 1. **Quantitative results.** We compare several memoryless and replay-based methods with the original baseline framework proposed by Cioppa *et al*. [10]. For each category, we benchmark several selection functions f_S, update functions f_U, and regularizers $\mathcal{R}$. The performance is provided for our proposed evaluation metrics for the 20/40 concatenated sequences. The replay-based methods generally outperform the baseline and the memoryless methods. The LwF and MAS regularization methods decrease the performance, while ACE and RWalk increase the performance. The best results are obtained with a uniform replay buffer, MIR, MIR+ACE, and MIR+RWalk. We compare the temporal evolution of the performance of the Baseline with one of the best performing method MIR+RWalk in Figure 3.

Methods	**Parameters**			**Metrics (mean %)**				
	f_S	f_U	$\mathcal{R}$	$mIoU$	$mIoU$ NDS	FWT	BWT	Final BWT
Memoryless	/	/	/	18.4/19.4	14.9/15.1	6.8/4.8	7.8/7.5	14.9/15.0
	/	/	MAS	14.0/14.0	13.0/13.3	11.1/11.1	12.9/12.9	14.2/14.2
	/	/	LwF	15.7/15.9	12.0/11.0	9.7/6.8	11.3/8.9	14.7/12.9
	/	/	RWalk	18.3/19.3	14.6/14.7	7.5/4.7	8.6/6.5	15.1/14.2
Baseline	All	FIFO	/	23.4/24.2	19.8/18.2	14.5/9.5	17.7/13.9	21.9/19.9
Replay Buffer	Uniform	Uniform	/	25.5/25.0	23.6/21.1	**22.2**/17.3	30.6/28.8	29.4/28.4
	Prioritized	Prioritized	/	25.1/25.1	23.2/20.8	21.3/17.3	29.2/28.4	29.2/28.9
	MIR	Uniform	/	25.2/25.2	23.7/**24.5**	21.9/**22.5**	30.5/28.6	29.5/29.7
	MIR	Uniform	MAS	14.5/14.9	13.4/14.7	12.1/13.6	13.9/15.2	15.1/15.4
	MIR	Uniform	LwF	18.7/18.1	17.6/15.7	17.4/13.9	21.0/20.2	22.4/21.1
	MIR	Uniform	ACE	**25.6/25.5**	**24.2**/21.8	22.0/17.5	**30.8**/29.4	28.8/28.5
	MIR	Uniform	RWalk	25.2/25.4	23.4/22.0	21.8/18.0	30.0/**30.8**	**30.1/30.8**

the rates are: $r_{\mathcal{V}} = 30$ frames per second, $r_{\mathbf{T}} = 3$ seconds per frame, and $r_{\mathbf{S}_c} = 60$ seconds per epoch.

4.2. Quantitative results

We compare the performance of the original framework with the proposed continual learning approaches. As a naive approach, we also study a memoryless online distillation framework, in which the online dataset does not store any frame. In this setup, the pairs produced by the teacher are used only once for training and are then deleted. As can be seen from Table 1, the memoryless approaches perform worse than the original framework for all metrics, showing that retaining some information in an online dataset (or replay buffer) improves the performance. Interestingly, all replay-based methods without regularizers improve compared to the baseline, with the best performance obtained by MIR overall. Adding a regularizer is however not always beneficial. For instance, MAS and LwF systematically decrease the performance, while ACE and RWalk slightly increase the performance. We hypothesize that this can be attributed to the fact that MAS and LwF were proposed in the offline setup with the aim of reducing the elasticity of the model towards adapting to new information. While this approach was proven to be useful in several scenarios, it could hinder the student from quickly adapting to new domains in the online setup. The biggest improvement is therefore mainly due to the replay buffer method with MIR.

In Figure 3, we show the evolution of the performance over time for the baseline and on one of the best method (MIR+RWalk) for cycles of 20. As can be seen from the $mIoU$ plot, during the two first cycles, both methods have similar results. This is expected as they both discover the new domains. The first difference can be seen at the second transition, where the first domain is seen once again. The baseline method has a huge drop, while the continual learning method shows good performance. At each other transition, MIR+RWalk does not suffer from the drop in performance caused by the forgetting of the previous domain. We conduct a comparison between the MIR+RWalk method and the original online distillation framework (baseline) by analyzing the performance evolution of the $mIoU$, BWT, Final-BWT, and FWT metrics. When evaluated on the previous domain, MIR+RWalk significantly outperforms the baseline in BWT, indicating its ability to retain information about the previous domain on frames it has been trained on. In the case of Final-BWT, the baseline quickly forgets past knowledge, while MIR+RWalk is able to maintain high performance for both domains across many cycles. Finally, when evaluated on the future domain, MIR+RWalk also shows significant performance improvements compared to the baseline in FWT, indicating its ability to generalize on new frames from a previous domain.

4.3. Qualitative results

We qualitatively demonstrate the effect of the best performing continual learning methods on the catastrophic forgetting. To do so, we investigate the quality of the segmentation masks right after the second transition from high-

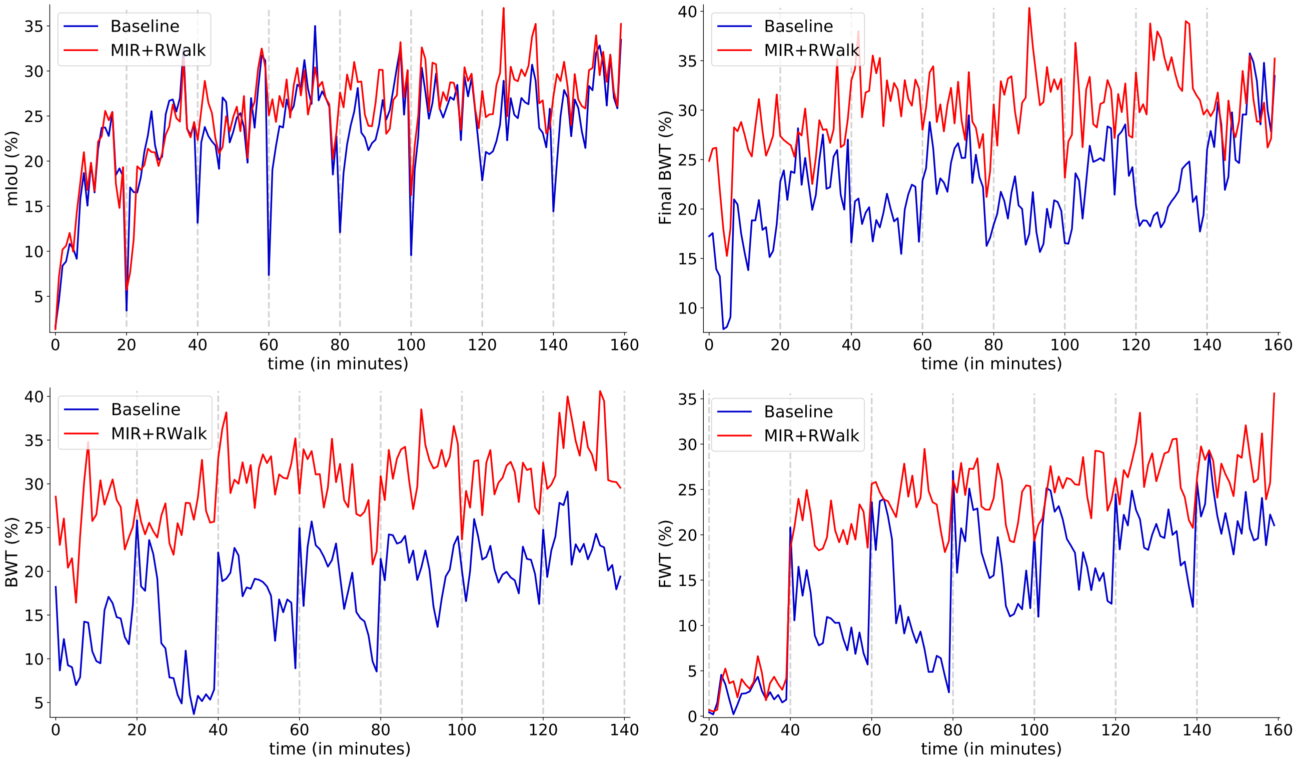

Figure 3. **Evolution of the performance over time.** We compare the evolution with respect to $mIoU$, BWT, Final-BWT, and FWT of the MIR+RWalk method with the original online distillation framework (baseline). (Top-left) $mIoU$: the performances are mostly similar within the domain, but around the domain shifts (from the second cycle), the baseline suffers from forgetting while MIR+RWalk keeps high performance. (Bottom-left) BWT: when evaluating on the previous domain, MIR+RWalk clearly outperforms the baseline, showing that it is able to retain information about the previous domain, on frames it has trained on. (Top-right) Final-BWT: the baseline quickly forgets past knowledge, while MIR+RWalk is able to retain high performance for both domains across many cycles. (Bottom-right) FWT: when evaluating on the future domain, MIR+RWalk also significantly outperforms the baseline, showing that it is able to generalize on new frames of a particular domain using information from a previous domain it has seen before.

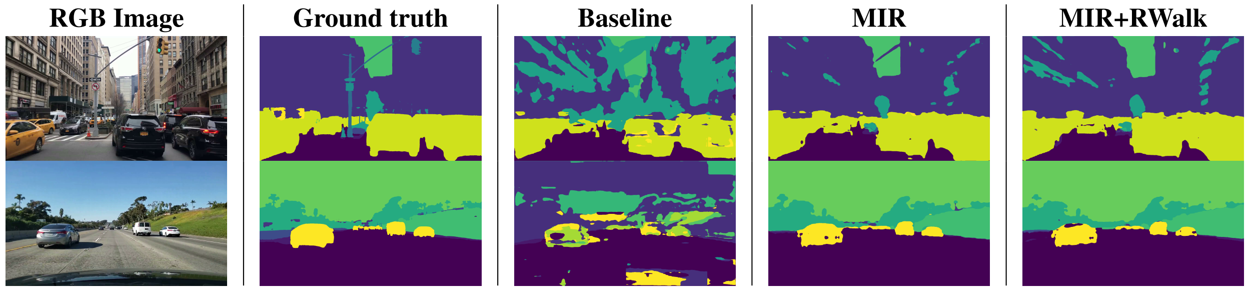

Figure 4. **Qualitative results.** Comparison of the segmentation masks obtained by different online continual learning methods: (top row) a frame taken right after second transition between highway and downtown, and (bottom row) a frame taken right after seventh transition between downtown and highway. The baseline method predicts poor segmentation masks after the domain shift, even though it has already seen this domain before. In contrast, MIR and MIR+RWalk produce better segmentation masks.

way to downtown (the student has seen the downtown only once before), and the seventh's transition from downtown to highway (the student has already seen the highway domain 6 times before). Figure 4 compares the segmentation masks obtained by the baseline method, MIR, and MIR+RWalk with the ground-truth mask. As shown, even though the student has already seen the domain previously, the segmentation masks of the baseline right after the domain shift are very poor. In practice, this could lead to hazardous situations for the autonomous vehicle and its passengers. On the contrary, the segmentation masks obtained with MIR and MIR+RWalk are much closer to the ground-truth masks. The quantitative results demonstrate that incorporating continual learning algorithms into the online distillation framework considerably enhances the quality of the predictions, rendering it more viable for real-world applications.

5. Conclusion

In conclusion, the development of online distillation has brought new opportunities for adapting deep neural networks in real time, making them more suitable for practical applications such as autonomous driving. However, the issue of catastrophic forgetting when the domain shifts has been a major challenge in the implementation of this technique. In this paper, we proposed a novel solution to this issue by incorporating continual learning methods. Through our experimentation, we evaluated several state-of-the-art continual learning methods and demonstrated their effectiveness in reducing catastrophic forgetting. We also conducted a detailed analysis of our proposed solution in the case of cyclic domain shifts. The results highlight that our approach improves the robustness and accuracy of online distillation, making it a promising technique for real-world applications. This work represents a significant step forward in the field of online distillation and continual learning, with the potential to have a meaningful impact on various fields such as autonomous driving.

Acknowledgement. A. Cioppa is funded by the F.R.S.-FNRS. A. Halin is funded by the Walloon region (Service Public de Wallonie Recherche, Belgium) under grant No. 2010235 (ARIAC by DIGITALWALLONIA4.AI). M. Henry is funded by PIT MecaTech, Belgium, under grant No. C8650 (ReconnAIssance). This work was partially supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2021-4648.

References

[1] Motasem Alfarra, Zhipeng Cai, Adel Bibi, Bernard Ghanem, and Matthias Müller. SimCS: Simulation for online domain-incremental continual segmentation. *CoRR*, abs/2211.16234, 2022. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 11207 of *Lect. Notes Comput. Sci.*, pages 144–161. Springer Int. Publ., 2018. 2, 4

[3] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019. 2, 3

[4] Fatemeh Azimi, Sebastian Palacio, Federico Raue, Jorn Hees, Luca Bertinetto, and Andreas Dengel. Self-supervised test-time adaptation on video data. In *IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pages 2603–2612, Waikoloa, HI, USA, Jan. 2022. Inst. Electr. Electron. Eng. (IEEE). 2

[5] Olivier Barnich and Marc Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.*, 20(6):1709–1724, Jun. 2011. 3

[6] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *Int. Conf. Learn. Represent. (ICLR)*, 2022. 4

[7] Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 8261–8270, Montréal, Can., Oct. 2021. Inst. Electr. Electron. Eng. (IEEE). 1

[8] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 11215 of *Lect. Notes Comput. Sci.*, pages 556–572. Springer Int. Publ., 2018. 2, 4

[9] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. In *Int. Conf. Mach. Learn. (ICML)*, 2019. 1, 2

[10] Anthony Cioppa, Adrien Deliege, Maxime Istasse, Christophe De Vleeschouwer, and Marc Van Droogenbroeck. ARTHuS: Adaptive real-time human segmentation in sports through online distillation. In *IEEE Int. Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW), CVsports*, pages 2505–2514, Long Beach, CA, USA, Jun. 2019. Inst. Electr. Electron. Eng. (IEEE). 1, 2, 3, 4, 5, 6

[11] Anthony Cioppa, Adrien Deliège, Noor Ul Huda, Rikke Gade, Marc Van Droogenbroeck, and Thomas B. Moeslund. Multimodal and multiview distillation for real-time player detection on a football field. In *IEEE Int. Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW), CVsports*, pages 3846–3855, Seattle, WA, USA, Jun. 2020. 2

[12] Anthony Cioppa, Adrien Deliège, and Marc Van Droogenbroeck. A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games. In *IEEE Int. Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW), CVsports*, pages 1846–1855, Salt Lake City, UT, USA, Jun. 2018. 2, 5

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3213–3223, Las Vegas, NV, USA, Jun. 2016. Inst. Electr. Electron. Eng. (IEEE). 5

[14] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning. In *Continual learning W., Neural Inf. Process. Syst. (NeurIPS)*, 2018. 5

[15] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation. *Trans. Comput. Sci. Comput. Intell.*, pages 877–894, 2021. 2

[16] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.*, 3(4):128–135, Apr. 1999. 2

[17] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-DFCIL: Relation-guided representation learning for data-free class incremental learning. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 13683 of *Lect. Notes Comput. Sci.*, pages 423–439. Springer Nat. Switz., 2022. 2

[18] Prachi Garg, Rohit Saluja, Vineeth N. Balasubramanian, Chetan Arora, Anbumani Subramanian, and C. V. Jawahar. Multi-domain incremental learning for semantic segmentation. In *IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pages 2080–2090, Waikoloa, HI, USA, Jan. 2022. Inst. Electr. Electron. Eng. (IEEE). 2

[19] Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip H. S. Torr, and Bernard Ghanem. Real-time evaluation in online continual learning: A new hope. *CoRR*, abs/2302.01047, 2023. 2

[20] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 13923–13932, Seattle, WA, USA, Jun. 2020. Inst. Electr. Electron. Eng. (IEEE). 1

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 770–778, Las Vegas, NV, USA, Jun. 2016. Inst. Electr. Electron. Eng. (IEEE). 1

[22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. 1

[23] Oguzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3D common corruptions and data augmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 18941–18952, New Orleans, LA, USA, Jun. 2022. Inst. Electr. Electron. Eng. (IEEE). 1

[24] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proc. National Acad. Sci. (PNAS)*, 114(13):3521–3526, Mar. 2017. 1, 2, 4

[25] Jinlong Li, Runsheng Xu, Jin Ma, Qin Zou, Jiaqi Ma, and Hongkai Yu. Domain adaptive object detection for autonomous driving under foggy weather. In *IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pages 612–622, Waikoloa, HI, USA, Jan. 2023. Inst. Electr. Electron. Eng. (IEEE). 2

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, Dec. 2018. 1, 2, 4

[27] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3367–3375, Boston, MA, USA, Jun. 2015. Inst. Electr. Electron. Eng. (IEEE). 1

[28] Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. TTN: A domain-shift aware batch normalization in test-time adaptation. In *Int. Conf. Learn. Represent. (ICLR)*, pages 1–19, 2023. 2

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017. 2

[30] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2502–2511, Long Beach, CA, USA, Jun. 2019. Inst. Electr. Electron. Eng. (IEEE). 2

[31] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychol. Learn. Motiv.*, pages 109–165, 1989. 2

[32] Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 13694 of *Lect. Notes Comput. Sci.*, pages 128–146. Springer Nat. Switz., 2022. 2

[33] Sébastien Piérard, Anthony Cioppa, Anaïs Halin, Renaud Vandeghen, Maxime Zanella, Benoît Macq, Saïd Mahmoudi, and Marc Van Droogenbroeck. Mixture domain adaptation to improve semantic segmentation in real-world surveillance. In *IEEE/CVF Winter Conf. Appl. Comput. Vis. Work. (WACVW)*, pages 22–31, Waikoloa, HI, USA, Jan. 2023. Inst. Electr. Electron. Eng. (IEEE). 2

[34] Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. GDumb: A simple approach that questions our progress in continual learning. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 12347 of *Lect. Notes Comput. Sci.*, pages 524–540. Springer Int. Publ., 2020. 2

[35] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 10745–10755, Montreal, QC, Canada, Oct. 2021. Inst. Electr. Electron. Eng. (IEEE). 1, 2

[36] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2015. 3

[37] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, Jan. 2015. 1

[38] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 9354–9364, Montreal, QC, Canada, Oct. 2021. Inst. Electr. Electron. Eng. (IEEE). 2

[39] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Eur. Conf. Comput. Vis. (ECCV)*, volume 9915 of *Lect. Notes Comput. Sci.*, pages 443–450. Springer Int. Publ., 2016. 2

[40] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: A synthetic driving dataset for continuous multi-task domain adaptation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 21339–21350, New Orleans, LA, USA, Jun. 2022. Inst. Electr. Electron. Eng. (IEEE). 2

[41] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *CoRR*, abs/2006.10726, 2020. 2

[42] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7191–7201, New Orleans, LA, USA, Jun. 2022. Inst. Electr. Electron. Eng. (IEEE). 2

[43] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 34, pages 12077–12090, 2021. 2, 5

[44] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2633–2642, Seattle, WA, USA, Jun. 2020. Inst. Electr. Electron. Eng. (IEEE). 2, 5

[45] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Int. Conf. Mach. Learn. (ICML)*, 2017. 4

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6230–6239, Honolulu, HI, USA, Jul. 2017. Inst. Electr. Electron. Eng. (IEEE). 2

[47] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6877–6886, Nashville, TN, USA, Jun. 2021. Inst. Electr. Electron. Eng. (IEEE). 2