Principles of Forgetting in Domain-Incremental Semantic Segmentation in Adverse Weather Conditions

Tobias Kalb
Porsche Engineering Group GmbH
Weissach, Germany
tobias.kalb@porsche-engineering.de

Jürgen Beyerer
Fraunhofer IOSB & Karlsruhe Institute of Technology
Karlsruhe, Germany
juergen.beyerer@iosb.fraunhofer.de

Abstract

Deep neural networks for scene perception in automated vehicles achieve excellent results for the domains they were trained on. However, in real-world conditions, the domain of operation and its underlying data distribution are subject to change. Adverse weather conditions, in particular, can significantly decrease model performance when such data are not available during training. Additionally, when a model is incrementally adapted to a new domain, it suffers from catastrophic forgetting, causing a significant drop in performance on previously observed domains. Despite recent progress in reducing catastrophic forgetting, its causes and effects remain obscure. Therefore, we study how the representations of semantic segmentation models are affected during domain-incremental learning in adverse weather conditions. Our experiments and representational analyses indicate that catastrophic forgetting is primarily caused by changes to low-level features in domain-incremental learning and that learning more general features on the source domain using pre-training and image augmentations leads to efficient feature reuse in subsequent tasks, which drastically reduces catastrophic forgetting. These findings highlight the importance of methods that facilitate generalized features for effective continual learning algorithms.

1. Introduction

Semantic segmentation is widely used for environment perception in automated driving, where it aims at recognizing and comprehending images at the pixel level. One fundamental constraint of the traditional deep learning-based semantic segmentation models is that they are often only trained and evaluated on data collected mostly in clear weather conditions and that they assume that the domain of the training data matches the domain they operate in. However, in the real world, those autonomous driving systems are faced with constantly changing driving environments and variable input data distributions. Specifically, changing weather conditions can have adverse effects on the performance of segmentation models.

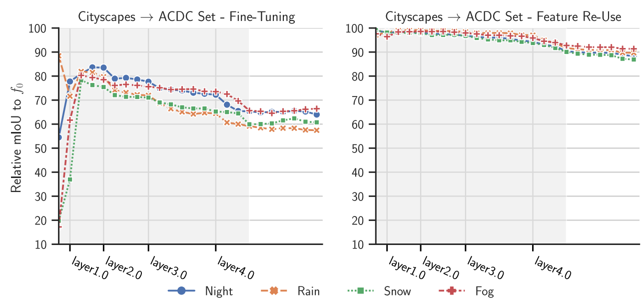

Figure 1. Activation drift between models f_1 to f_0 measured by relative mIoU on the first task of the models stitched together at specific layers (horizontal axis). The layers of the encoder are marked in the gray area, the decoder layers in the white area. Layer-stitching reveals that during domain-incremental learning, changes in low-level features are a major cause of forgetting. With an improved training scheme, combining simple augmentations, exchanging normalization layers and using pre-training, the model is optimized to reuse low-level features during incremental learning, leading to significant reduction of catastrophic forgetting.

Therefore, a semantic segmentation model needs to be adapted to these conditions. A naive solution to this problem would be to incrementally fine-tune the model to new domains with labeled data. However, fine-tuning a neural network to a novel domain will, in most cases, lead to a severe performance drop in previously observed domains. This phenomenon is usually referred to as *catastrophic forgetting* and is a fundamental challenge when training a neural network on a continuous stream of data. Recently proposed methods mostly mitigate this challenge by replaying data from previous domains, re-estimating statistics or even in an unsupervised manner by transferring training images in the style of the novel domain [32, 48, 52]. The focus of our work is to study how the internal representations of semantic segmentation models are affected during domain-incremental learning and how efficient feature reuse can mitigate forgetting without explicit replay of the previous domain. Our main contributions are:

1. We analyze the activation drift that a model's layers are subjected to when adapting from good to adverse weather conditions by stitching them with the previous task's network. We reveal that the major cause of forgetting is a shift of low-level representations in the first convolution layer that adversely affects the population statistics of the following BatchNorm Layer.
2. Using different augmentation strategies to match the target domains in color statistics or in the frequency domain, we reveal that learning color-invariant features stabilizes the representations in early layers, as they don't change when the model is adapted to a new domain.
3. With a combination of pre-training, augmentations and exchanged normalization layers, we achieve an overall reduction of forgetting of ∼20% mIoU compared to fine-tuning without using any form of replay and prove the effectiveness of pre-training and augmentations which are often overlooked in continual learning.

2. Related Work

2.1. Continual Learning

Continual learning research is primarily concerned with developing methods to overcome catastrophic forgetting. It has been extensively studied in incremental classification tasks, where the approaches can be broadly divided into regularization-based methods [2, 14, 25, 29, 58], replay-based methods [19,43,44,47] and parameter-isolation methods [31, 50, 55]. For semantic segmentation, significant progress has been made on addressing class-incremental learning, mostly utilizing knowledge distillation-based approaches [7, 8, 15, 23, 26, 34, 35, 56]. Research on domain-incremental learning for semantic segmentation is relatively sparse. Garg *et al*. [17] propose a dynamic architecture that learns dedicated parameters to capture domain-specific features for each domain. Mirza *et al*. [36] circumvent the issues of biased BatchNorm statistics by re-estimating and saving them for every domain, so that during inference domain-specific statistics can be used. Our experiments in this paper show that the effectiveness of these approaches mostly originates in matching the low-level statistics of the initial block of the network. Recently, the focus shifted towards Continual Unsupervised Domain Adaptation (CDA).

2.2. Continual Unsupervised Domain Adaptation

CDA has the goal of adapting a model that is trained on a supervised source dataset to a sequence of different domains, for which no labels are provided. However, in order to compensate for the missing labels of the target domains, the model has access to the initial source dataset throughout the entire training sequence [32]. Methods in this category work mostly by storing information about the style of the specific domains, so that during training the source images can be transferred into the styles of the different target domains. This can be achieved by storing low-frequency components of the domains [48] or by capturing the style using generative models [32, 52]. Other recent work proposes to use a target-specific memory for every domain to mitigate forgetting [24]. Although this setting is different from the domain-incremental setting in our work, the insights that we gain on the causes and mitigation strategies, especially the importance of the invariance of low-level features, can be transferred to CDA as well.

2.3. Analysing Catastrophic Forgetting

In previous studies, representational analysis techniques such as centered kernel alignment [27] and linear probing were used to analyze the effects of catastrophic forgetting in deep learning for image classification [13, 42]. Their results concluded that deeper layers are disproportionately responsible for forgetting in class-incremental learning. Other work investigates how multi-task- and incremental learning solutions are connected in their loss landscape [38], or how task sequence [40] or task similarity [42] affect catastrophic forgetting. Overall, much of the recent research in analyzing catastrophic forgetting has been primarily focused on classification tasks or class-incremental learning [22, 42]. While much progress has been made on mitigating forgetting in domain-incremental semantic segmentation by utilizing style transfer, replay or matching BN statistics, it remains unclear how these specific changes affect the internal representations of the model.

3. Preliminaries

3.1. Problem Formulation

The task of semantic segmentation is to assign a class, out of a set of pre-defined classes $\mathcal{C}$, to each pixel in a given image. A training task $T = \{(x_n, y_n)\}_{n=1}^{N}$ consists of a set of N images $x \in \mathcal{X}$ with $\mathcal{X} = \mathbb{R}^{H\times W\times 3}$ and corresponding labels $y \in \mathcal{Y}$ with $\mathcal{Y} = \mathcal{C}^{H\times W}$. Given the task T the goal is to learn a mapping $f : \mathcal{X} \mapsto \mathbb{R}^{H\times W\times|\mathcal{C}|}$ from the image space $\mathcal{X}$ to a score vector $\hat{y}$. The final segmentation mask is then computed as $\bar{y}_i = \arg\max_{c\in C} \hat{y}_{i,c}$. In the incremental learning setting, the model f is trained on a sequence of tasks T_k that can introduce new classes or visually distinct instances from the same classes. This study focuses on domain-incremental settings, where the input distribution of the data is changing between tasks while the set of classes is fixed. Specifically, we focus on domain-increments in different adverse weather conditions.

3.2. Normalization Layers

Normalization layers are essential in training Convolutional Neural Networks (CNNs), as they address the internal covariate shift of the network by normalizing the inputs to layers, so that the input distributions to each layer are stable during training [20]. Batch Normalization (BN) [20] is the most common CNN normalization layer, which normalizes layer inputs using moments across the mini-batch dimension. However, to achieve deterministic behavior during inference, the mini-batch variance and mean are replaced by the global population mean and variance, which are obtained during training using an exponential running average. This works for i.i.d.[1] data, but in non-i.i.d. incremental learning, the BN estimates of the population mean and variance are heavily biased towards the most recent task, resulting in a significant loss of performance on old tasks [30]. Continual Normalization (CN) alleviates this discrepancy by combining Group- [51] and Batch Normalization [41].

3.3. Measuring Activation Drift

Accuracy-based evaluation only allows for restrictive insight into the causes of forgetting of a model, specifically if representation shifts happen in the early layers of the networks. As we expect catastrophic forgetting to be the result of a change in weights and activations of the model that are no longer tuned to the most recent task, we aim to measure activation drift in incremental learning. Specifically, we want to measure the activation shift between a model f_0 and f_1. Where f_0 denotes the model trained on T_0 and f_1 is the model initialized with the parameters of f_0 and incrementally trained on T_1. For this, we utilize the layer matching framework introduced by Csiszárik *et al.* [12], without an additional stitching layer [22]. Fig. 2 shows the layer stitching setup. In this setup, we measure the impact of activation drift until layer n by propagating the activations of layer n directly to the layer $n+1$ in f_0. The resulting stitching model is then evaluated on the test set of the initial task T_0 and compared to the initial accuracy of the model f_0. This gives us a proxy measure of how useful the features at a specific layer of the adapted model f_1 are for the initial model f_0.

4. Experiments

Datasets: We conduct our experiments on adapting to adverse weather conditions using an domain-incremental setup that involves adapting from the Cityscapes (CS) [11] dataset to ACDC [46], which is commonly used as a benchmark for unsupervised domain adaptation. The CS dataset is an automotive semantic segmentation dataset, collected during daytime and dry weather conditions in different German, Swiss and French cities. It contains 2975 training and

[1] independent and identically distributed

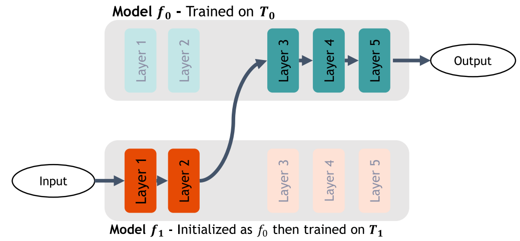

Figure 2. Layer stitching: By directly propagating the activations of f_1 to f_0 we measure the feature reuse between the model f_0 and f_1 [22].

500 validation images. The ACDC dataset is collected during different adverse weather conditions and divided into four different subsets: *Night*, *Rain*, *Fog* and *Snow*. ACDC and CS share the same 19 classes, so that the changes between the tasks is only based on the domain differences. To study how features are reused and or adapted in each adverse weather condition, we investigate four different scenarios all starting with the same CS model: *CS* $\rightarrow$ *Night*, *CS* $\rightarrow$ *Rain*, *CS* $\rightarrow$ *Snow* and *CS* $\rightarrow$ *Fog*.

Models: We use the widely adopted DeepLabV3+ [9] in our experiments with a ResNet50 backbone, as DeepLabV3+ is a commonly used architecture in domain adaptation. Furthermore, we confirm our findings with different architecture in Appendix F, because the architectural choices can have a significant impact in continual learning [21, 37]. Finally, in Sec. 5.2 we show that the recently introduced SegFormer-B2 [53] is more robust towards low-level feature change than its CNN counterparts. In the majority of the experiments, we use randomly initialized models for training on the first task, as pre-training is known to increase robustness to catastrophic forgetting [16, 33] by enabling low-level feature reuse, as we will see in Sec. 4.3.

Optimization Strategy We train the networks using SGD as an optimizer with momentum of 0.9 and weight decay of 3×10^{-3}. We use a polynomial learning rate schedule with power 0.9, and start CS training with a 0.07 learning rate and use a batch size 8. We use the same learning rate policy for ACDC subset training, starting with 5×10^{-3}. CS and ACDC subsets are trained for 200 and 150 epochs, respectively. We crop the images to 512×1024 and utilize random horizontal flipping and scaling during training. Unless stated otherwise, we do not use any other augmentations. During testing we do not use any scaling or cropping.

Evaluation Metrics: We evaluate the performance of each model on the validation set of each dataset using the mean intersection-over-union (mIoU). We denote the mIoU of the model trained on all tasks up to p and evaluated on task q as $\text{mIoU}_{p,q}$. So that the zero-shot performance of a model trained on task $p = 0$ and evaluated on $q = 1$ is denoted as $\text{mIoU}_{0,1}$. Furthermore, we report *average learning accuracy* and *forgetting*, that measure the learning capability and the severity of forgetting [37].

$$\text{average learning acc.} = \frac{\text{mIoU}_{0,0} + \text{mIoU}_{1,1}}{2} \tag{1}$$

$$\text{forgetting} = \text{mIoU}_{0,0} - \text{mIoU}_{1,0} \tag{2}$$

4.1. Activation Drift after Incremental Adaptation

First, we study the overall activation drift that arises in the CNN models when naively adapting to the different adverse weather conditions. Therefore, we first train DeepLabV3+ on CS and subsequently fine-tune it individually on each of the ACDC subsets. The results are displayed in Tab. 2. As one would expect, the zero-shot performance on the adverse weather conditions is good for visually similar conditions such as *Fog* and *Rain* and significantly worse for *Snow* and *Night*, with *Night* being the worst with only 10.4% $\text{mIoU}_{0,1}$. This can be explained by the apparent differences between the domains, so that the day-to-night shift and snow-covered landscape represents a bigger shift than the wet environment or the foggy conditions [46]. However, after fine-tuning on the adverse subsets, we notice that the better zero-shot performance on *Rain* and *Fog* does not indicate less forgetting compared to *Snow* and *Night*. Most strikingly, forgetting is the lowest after adapting to *Night*. To determine which layers are most affected by the internal activation drift, we measure the activation drift for each layer between the model before and after learning the second task with layer stitching. We use the setup explained in Sec. 3.3. The mIoU relative to the initial performance on the first task is shown in Fig. 1. We observe that activation drift is mainly affecting the network's early layers, which is in contrast to class-incremental learning settings, where early layers remain stable [13, 22, 42]. Specifically, the low-level features of the models tuned to *Fog* and *Snow* cannot be reused by the initial model, indicating that the shallow layers of the network have changed significantly. However, after the initial drop of relative mIoU in Tab. 2 we see a substantial increase in mIoU after *layer1.0*, which indicates later features are in fact reused by the Cityscapes model. We hypothesize that at that point, features are more abstract and therefore more useful for the model trained on CS. After *layer2.0* we observe a steady decrease in relative mIoU until the decoder layers. We note that once the representations are shifted, subsequent layers are unlikely to regain similarity as their representations are based on the output of the previous layer. The changing image distribution is most likely to blame for the initial feature disparity. Thus, we analyze domain image distribution at the pixel level in the next section.

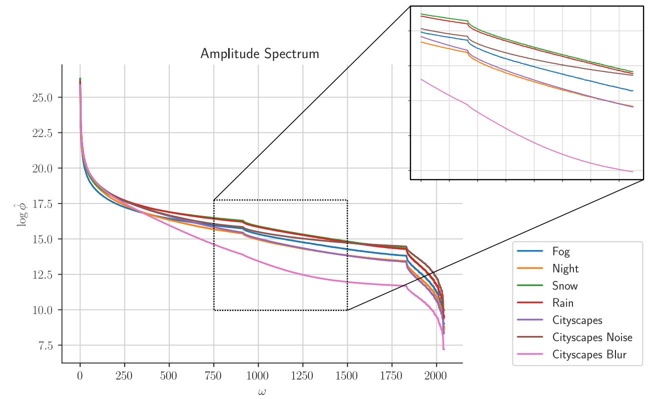

Figure 3. Amplitude Spectra of Cityscapes, augmented CS and the ACDC subsets. In the frequency domain Cityscapes is much more similar to *Night* than to any other of the ACDC subsets, specifically in the high-frequent components of the images. Blur is efficiently removing of high frequency components.

Dataset	Mean			Standard Deviation		
	Hue	Saturation	Value	Hue	Saturation	Value
Rain	86	32	110	62	35	78
Snow	104	19	132	55	21	62
Night	64	122	60	66	64	46
Fog	93	20	131	60	23	64
Cityscapes	59	49	83	18	22	49

Table 1. The Mean and standard deviations for the HSV-channels of each subset. There is a severe color shift between the domains in overall brightness of the images.

4.2. Analysis of image statistics

To interpret the low-level feature change within the network, we compare the image statistics of each domain to explain the significant representation changes in the early layers. Therefore, we compare the domains by their corresponding pixel mean and standard deviation for each HSV-channel in Tab. 1. We observe that there is a substantial difference between the domains, specifically *Rain*, *Snow* and *Fog* being notably brighter than Cityscapes. Furthermore, as the capability of CNNs to generalize can be adversely affected by exploiting mid- to high-frequency components of images [1, 49], we also analyze the amplitude spectra in the frequency domain of the different training tasks in Fig. 3. We can see that the domains are similar in low- to mid-frequency ranges, but *Snow* and *Rain* contain much more high-frequency components. This could lead to overfitting to high-frequency features when the model is trained on these domains and magnify forgetting.

4.3. Adjusting Low-Level Features

Previous work suggests that low-level feature reuse is important for successful transfer learning [39] and that pretraining can mitigate forgetting for class-incremental classification tasks [16, 33]. In a set of experiments, we show that low-level feature reuse is not only important for success-

Task 2	Task 1 mIoU$_{CS}$	Task 1 mIoU$_{T2}$	Task 2 mIoU$_{CS}$	Task 2 mIoU$_{T2}$	Task 2 Forgetting
Rain	72.0	30.4	38.8	57.7	33.2
Night	72.0	10.5	45.9	43.6	26.1
Snow	72.0	23.1	42.2	62.3	29.8
Fog	72.0	33.4	44.0	69.0	28.0

Table 2. Results on *CS* $\rightarrow$ *ACDC* in mIoU (%) for each subset of ACDC. While the zero-shot performance for *Night* is the worst, after the fine-tuning to *Night*, it is least affected by forgetting.

ful transfer of knowledge to downstream tasks, but is also vital to reduce catastrophic forgetting. Therefore, we investigate different pre-training and augmentation protocols to initialize the model f_0 on CS. We then fine-tune the model f_0 on the different ACDC subsets without any augmentations or any continual learning algorithm. Thus, we can examine how different methods improve the model's feature reuse and reduce forgetting. To study the impact on pre-trained backbones, we use ResNet50 weights trained on ImageNet1k either fully-supervised or using the self-supervised methods DINO [6], MoCo v3 [10], SwAV [5] and BarlowTwins [57]. For the augmentation experiments, we use the following strategies:

- Using AutoAlbument (AutoAlbum) [4], to learn an image augmentation policy from the CS dataset using Faster AutoAugment [18].
- Learning color-invariant features by intensive Color Jittering and randomly rearranging input image channels. We denote this combination as *Distortion* (Distort).
- Learning features tuned for mid- to low-frequencies by Gaussian blurring or adding Gaussian noise to remove high-frequency information. Changes to the spectrum are displayed in Fig. 3

We also add an offline pre-trained model that is trained jointly on the CS and ACDC subsets and then fine-tuned on the target task. We use offline pre-training to infer an upper bound on a model's feature reuse, as the model should have learned features that are the joint optimum on both tasks. The results are displayed in Tab. 3. The overall trend we see is that both pre-training and augmentations during training on CS lead to better transfer to the subsequent tasks and reduced forgetting on CS. However, while pre-training improves the transfer to the new tasks, it only moderately improves zero-shot capabilities when compared to the model without pre-training. Color-based augmentation and AutoAlbum improve zero-shot performance but perform worse on the ACDC tasks than the pre-trained models, indicating that better zero-shot performance does not always lead to better transfer performance. Still, these augmentations are the most effective at mitigating forgetting for all tasks. Augmentations based on removing high-frequency components reduce forgetting less significantly or, in the case of *Fog*, lead to a further decrease in mIoU on the previous task. We note that for the domains *Rain* and *Snow* that contain more high-frequency components in their images than CS, adding noise and blurring is more effective than it is for *Fog* and *Night*. So far, the results indicate two things: 1) pre-trained features are less susceptible to forgetting and lead to a better transfer to future tasks, 2) augmentations significantly improve generalization and produce more general features in the early layers.

We also investigate these findings using layer stitching and display the plots in Fig. 4. The pre-trained models (bottom row) have a lower initial drop in similarity than randomly initialized models and remain higher throughout later layers. Noticeable is that even the offline pre-trained model is affected by a severe drop in similarity for *Snow*, *Fog* and also a moderate drop in similarity for *Rain* and *Fog*. We will later confirm that this is largely due to the biased population mean and standard deviation of the BN layers. Most surprising is that the models that used color augmentation during training are not affected by this initial drop, even though the optimization process on task 2 is the same for the offline and pre-trained models, so they should be affected by the same change in population statistics. The fact that this does not occur indicates that when training with augmentations, the first convolutional layers extract features that are more domain-invariant than the features of the pre-trained models, resulting in BN layer population statistics that are less biased to the previous task [41]. We validate this claim in the next section.

4.4. Impact of Batch Normalization on Forgetting

The results in Sec. 4.3 suggest that changing BN population statistics are a major cause of early layer representation changes. To verify this, we re-estimate the BN Layer population statistics on the combined dataset of CS and the specific ACDC subset without changing any parameters.[2] We then evaluate the model on the CS dataset and report the re-estimated mIoU$_R$ and increased performance as ΔmIoU in Tab. 5. Most methods benefit significantly from re-estimation of population statistics, with the Fine-Tuning (FT) model benefiting the most. Furthermore, we observe that pre-trained models improve only moderately compared to FT, meaning that they are less influenced by biased population statistics. Finally, we discover that models trained with augmentation only slightly improve after BN re-estimation, and even decrease in the *CS* $\rightarrow$ *Fog* setting. We explain this effect by the invariance to low-level properties of the images such as hue, saturation and brightness that the first CNN layer of the models trained with augmentations has to learn in order to cope with the augmentation

[2]This is achieved by doing a forward pass over the dataset while the BN Layers update their population statistics

	Method	Cityscapes Test mIoU	Night Zero Shot	Night Test mIoU	Night Forgetting	Rain Zero Shot	Rain Test mIoU	Rain Forgetting	Fog Zero Shot	Fog Test mIoU	Fog Forgetting	Snow Zero Shot	Snow Test mIoU	Snow Forgetting
Augment.	FT	72.0	10.5	43.6	26.1	30.4	57.7	33.2	33.4	69.0	28.0	23.1	62.3	33.2
	AutoAlb.	72.2	24.2	47.3	15.1	42.8	59.4	10.7	50.1	68.2	14.7	37.2	63.8	10.7
	Distort	71.7	19.8	46.5	16.2	38.9	60.9	19.0	46.5	68.3	15.2	32.8	62.3	19.0
	Gaus	69.1	8.1	46.3	23.0	26.9	60.0	26.7	26.9	65.4	28.3	15.8	64.7	26.5
	Noise	69.8	9.6	46.7	21.3	27.8	60.6	25.2	27.5	69.3	30.2	21.3	63.2	30.2
Pre-Training	ImageNet	73.9	6.3	47.5	19.1	27.3	60.9	22.5	21.8	68.8	23.8	25.8	66.2	28.8
	MOCO	75.2	14.0	48.3	17.3	32.6	63.5	26.8	41.9	72.3	22.3	30.3	66.3	33.8
	DINO	75.0	11.6	49.7	18.7	28.2	64.4	23.4	35.2	72.4	19.6	27.3	67.0	27.1
	BarlowT.	73.9	14.3	47.3	17.1	34.8	65.7	22.5	41.6	71.4	14.4	31.8	65.4	22.0
	SwAV	76.4	15.8	48.1	17.8	31.0	62.4	24.5	40.9	71.7	32.8	28.0	66.4	27.8
	Offline	-	46.1	47.4	3.2	59.0	59.1	0.5	67.5	66.9	3.2	62.4	62.9	3.3

Table 3. Results on *CS* $\rightarrow$ *ACDC* in mIoU (%) for each subset of ACDC: *Night*, *Rain*, *Snow* and *Fog* using different pre-training and augmentation strategies (Augment.). While pre-training significantly improves learning accuracy, it does not improve zero-shot performance, but still gives moderate improvements in reducing forgetting. Augmentations lead to slightly improved performance on the target set, improved zero-shot performance and significantly reduced forgetting for all weather conditions.

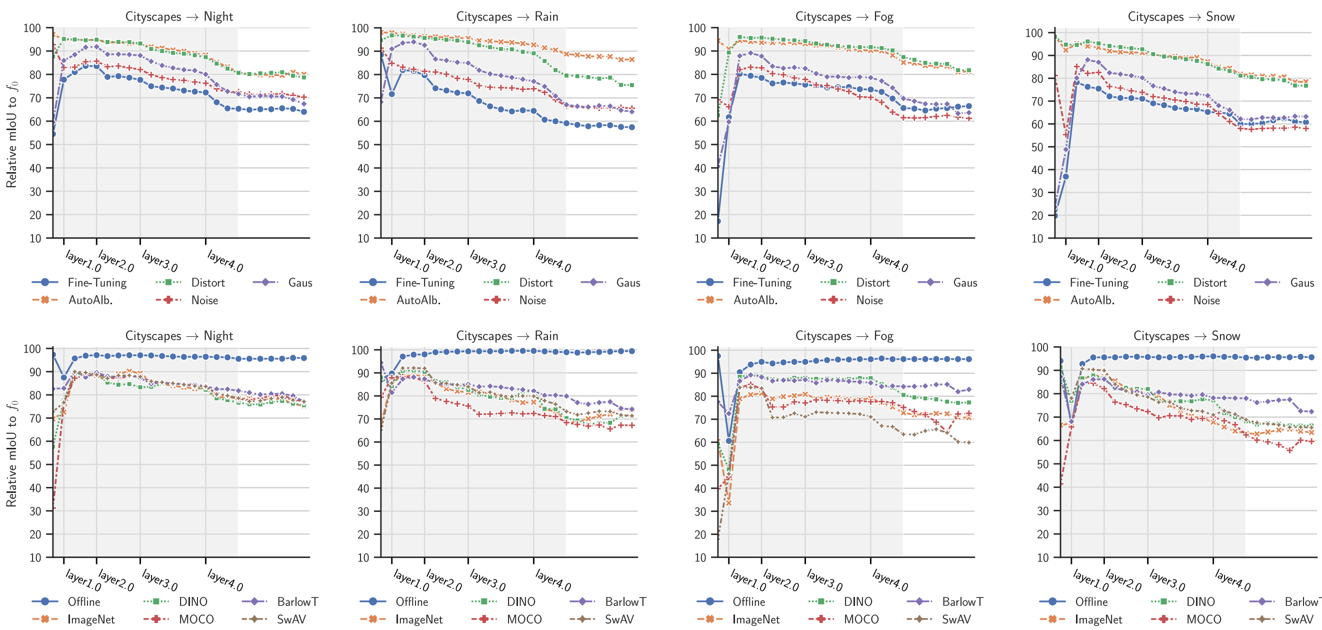

Figure 4. Activation drift between f_1 to f_0 measured by relative mIoU on the first task of the model stitched together at specific layers (horizontal axis). The layers of the encoder are marked in the gray area, the decoder layers in the white area. The activations up until *layer1.0* undergo drastic changes, specifically for *Snow* and *Fog*. After *layer1.0* the activations can again be reused by f_0 leading to an mIoU increase. However, throughout the remaining encoder layers of the network the activations of f_1 further deviate from f_0.

scheme. Thus, extracted features are invariant to these properties and subsequent BN layers are less affected by the distribution shift. The fact that only re-estimating BN layers without changing the initial layers leads to such a significant improvement, demonstrates that the adjusted population statistics can normalize the variance between the domains.[3] Overall, this means that BN is a major contributor to forgetting in the domain-incremental setting, but forgetting is also precipitated by low-level features that are tuned to their specific domain, which lead to a major change in population statistics in the BN layers. We validate these findings by exchanging all BN layers with Continual Normalization layers [41]. The results are shown in Fig. 5 and Tab. 4. We see that CN greatly reduces forgetting on CS, while we still observe a similar discrepancy in low-level features as for the models with BN. However, due to the combination of Group and Batch Normalization the changing low-level features are normalized across the channel dimensions before affecting the population statistics of BN.

[3] In Appendix C, we investigate which BN layers are affected.

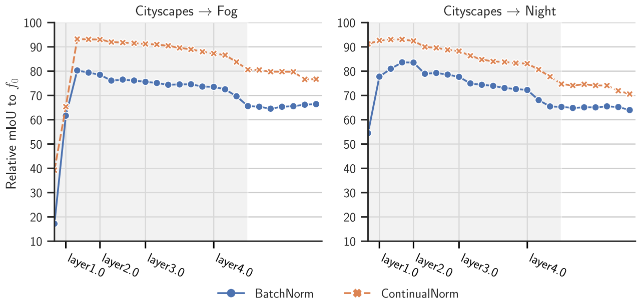

Figure 5. Comparing the activation drift between models trained with Batch Normalization and Continual Normalization on *Night* and *Fog*. Continual Normalization effectively reduces forgetting by mitigating the biased population in statistics in early BN layers.

Normalization	CS	Night			Fog		
	Test mIoU	*Zero Shot*	*Test mIoU*	*Average Forgetting*	*Zero Shot*	*Test mIoU*	*Average Forgetting*
BatchNorm	72.0	10.5	43.6	26.1	33.4	69.0	28.0
Cont. Norm	71.2	10.3	44.2	21.2	34.4	67.1	19.1

Table 4. Results on *CS* $\rightarrow$ *ACDC* in mIoU (%) with Batch- and Continual Normalization. By reducing the biased population statistics in early BN layers, CN effectively reduces forgetting.

Method	Night		Rain		Fog		Snow	
	$mIoU_R \uparrow$	$\Delta mIoU$	$mIoU_R \uparrow$	$\Delta mIoU$	$mIoU_R \uparrow$	$\Delta mIoU$	$mIoU_R \uparrow$	$\Delta mIoU$
FT	58.6	12.7	58.2	19.4	49.7	5.8	51.3	9.1
AutoAlb.	59.8	2.7	62	0.5	54.4	-3.1	53.9	-2.1
Distort	59.2	3.7	59.1	6.5	52	-4.5	54	0.0
ImageNet	61.7	7.4	59.6	8.1	51.7	2.1	54.7	9.7
MOCO	63.5	5.6	62.5	14.0	55.1	2.2	57.4	16,0
DINO	64	7.7	63.8	12.2	57.8	2.4	61.9	14.0
BarlowT	65.4	8.6	65.2	13.9	60.6	8.7	59.4	0.0
Img+Dis	62.5	4.3	63	-0.6	55.3	-4.4	59	0.3
Offline	68.3	-0.2	70.3	-0.8	68.8	-0.4	69.2	0.4

Table 5. Performance in mIoU [%] on *CS* of the model f_1 after re-estimating all BN layer population statistics. FT is most affected by changing population statistics, while models trained with augmentation are least affected.

4.5. Combining our Findings

We run the same experiments making incremental changes to the training process in CS by sequentially adding pre-training with DINO, then AutoAlbum and replacing BN with CN layers. Results in Tab. 6 demonstrate that these changes complement each other as they dramatically reduce forgetting on CS. This is apparent when comparing the layer stitching plots in Fig. 6, where we see that pre-training with DINO alone increases feature reuse only after *layer1.0* compared to fine-tuning. However, when combined with augmentations and CN, the representation drift before *layer1.0* is significantly reduced as well. This indicates that pre-training and training with augmentations enable feature reuse at different layers of the network depending on the task at hand. Therefore, when combining pre-training, CN and augmentation, we increase the feature

Method	Night		Rain		Fog		Snow	
	lrn. acc.	*forg.*	*lrn. acc.*	*forg.*	*lrn. acc.*	*forg.*	*lrn. acc.*	*forg.*
FT	57.8	26.1	64.9	33.2	70.5	28.0	67.2	29.8
+ DINO	62.3	18.7	69.7	23.4	73.7	19.6	71.0	27.1
+ AutoAlb.	62.2	13.5	71.2	8.2	73.9	11.7	71.2	13.1
+ CN	61.3	9.1	71.0	8.0	75.4	7.3	70.9	10.5

Table 6. Forgetting and Learning accuracy on *CS* $\rightarrow$ *ACDC* with incremental additions that increase the feature reuse, significantly reduces forgetting.

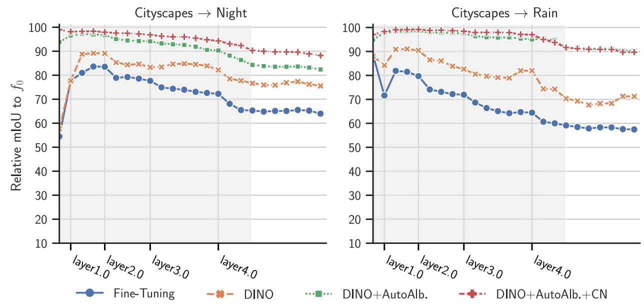

Figure 6. Comparing the activation drift between models trained on *Night* and *Fog* with sequentially adding pre-training with DINO, AutoAlb. and CN.

reuse in all domains, as we see in Fig. 1, reducing forgetting without any continual learning algorithm.

5. Ablation Studies

5.1. Comparison to Continual Learning Algorithms

So far, no explicit continual learning strategies like regularization or replay were used. Therefore, we show in Tab. 7 that achieving low-level feature reuse outperforms regularization methods even when using the same initialization. However, our training regime is still outperformed by naive replay. While EWC significantly improves when combined with CN and AutoAlbum, we see that Replay does not benefit from these adjustments. For Replay only pre-training leads to a significant increase in learning accuracy and a minor reduction of forgetting. A likely explanation is that the model is able to learn features that are invariant to domain differences due to the batch construction during replay, in which half of the mini-batch consists of replay samples from previous tasks. This could explain why replay is reported to be sample efficient in domain incremental learning [23].

5.2. Architectures

We confirm our findings by repeating our experiments on SegFormer [53], DeepLabV3+, and ERFNet [45], all of which were pre-trained on ImageNet. We choose DeepLabV3+ with a ResNet50 backbone and SegFormer-

Method	Night lrn. acc.	Night avg. forg.	Rain lrn. acc.	Rain avg. forg.
EWC	52.8	18.0	61.4	21.2
+ DINO	58.1	10.0	66.0	11.3
+ AutoAlb + CN	58.5	4.8	69.1	4.9
Replay	58.2	5.6	64.2	2.9
+ DINO	62.4	4.2	70.8	2.1
+ AutoAlb + CN	61.8	4.1	69.9	3.8
DINO + AutoAlb + CN	62.1	7.9	71.0	8.0

Table 7. Forgetting and Learning accuracy on *Night* and *Rain* with EWC and Replay. Our training scheme outperforms EWC on this benchmark and can be added to existing CL methods to improve learning accuracy and reduce forgetting.

Model	Night lrn. acc.	Night forg.	Rain lrn. acc.	Rain forg.	Fog lrn. acc.	Fog forg.	Snow lrn. acc.	Snow forg.
Segfromer-B2 [54]	59.4	16.2	69.1	11.1	71.6	11.8	70.4	14.3
DeepLabv3+ [9]	60.5	19.1	67.4	22.5	71.1	23.8	70	28.8
ERFNet [45]	56.6	29	63.2	36.2	67.5	30.1	64.5	59.5

Table 8. Forgetting and Learning accuracy of different Segformer-B2, DeepLabv3+ and ERFNet trained on *CS* $\rightarrow$ *ACDC* in mIoU (%) for each subset of ACDC.

Augment.	Night lrn. acc.	Night forg.	Rain lrn. acc.	Rain forg.	Fog lrn. acc.	Fog forg.	Snow lrn. acc.	Snow forg.
	59.4	16.2	69.1	11.1	71.6	11.8	70.4	14.3
Distort.	59.6	14.0	68.8	9.6	72.6	10.1	70.3	10.9
AutoAlb.	59.7	14.2	68.3	8.5	72.2	7.9	70.8	10.5

Table 9. Forgetting and learning accuracy in mIoU (%) of SegFormer-B2 with different augmentations SegFormer-B2 improves less than DeepLabV3+ using augmentations, suggesting it is less affected by color-dependent features.

B2 as they have a similar number of parameters, so that the results are comparable. The much smaller ERFNet is known to be more susceptible to forgetting due to its size [22]. The results in Tab. 8 show that transformer-based SegFormer is much less affected by catastrophic forgetting than its CNN counterpart, even without any augmentations. A likely explanation is that SegFormer learns more general features and is thus more robust to distribution changes. This would also explain why SegFormer-B2 does not improve as significantly as DeepLabV3+ with the addition of pixel-level augmentations, as shown Tab. 9. However, it is unclear whether this is an inherent feature of the self-attention mechanism, a result of training recipes [3], or of architectural choices such as using Layer- instead of Batch Normalization. We leave this for future studies.

6. Conclusion

Our study has shown that a major cause of catastrophic forgetting in domain-incremental learning is the shift of low-level representations, particularly in the first convolution layer. This shift affects the population statistics of subsequent BN layers and results in forgetting when adapting to new domains. To address this problem, we investigated the use of various pre-training schemes and pixel-level augmentations to facilitate features in early layers that can be reused in upcoming tasks. Our experiments showed that these methods were effective in reducing representation shift, with pre-training stabilizing the first layers, and augmentations primarily stabilizing the representations after the first BN layer. We hypothesize that training with augmentation strategies like *Distortion* or *AutoAlbum* encourages the model to learn features that are invariant to low-level image statistics such as hue, saturation and brightness that vary between the domains. So that during optimization on the new domain those features are not affected, leading to a significant reduction in forgetting. Interestingly, we found that pre-trained models struggle to learn such features in the early layers, but they still reduce forgetting notably compared to randomly initialized models. Leading us to believe that pre-training on ImageNet leads to more generalized features throughout the network. In our experiments, self-supervised pre-training outperformed supervised ImageNet pre-training on all domains, which suggests that SSL pre-training might not only be a vital tool for classification [16], but also for continual semantic segmentation.
The effectiveness of these approaches varies across the domains but is consistent for different CNN architectures that use BN layers[4]. Our findings are related to the research on spurious features in continual learning [28], as our training scheme is reducing the emergence of spurious features for the first task. Overall, we hope that our results highlight that an important component of continual learning can be found in methods that extract generalized features from the initial task, instead of only mitigating the effects of catastrophic forgetting during training on new data.
However, we note that even with improved low-level feature reuse, the model is still susceptible to catastrophic forgetting in later layers due to more abstract domain changes. As a result, we consider low-level feature reuse in incremental learning to be an important component of continual learning, but not the sole solution.

ACKNOWLEDGMENT

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project “KI Delta Learning“ (Förderkennzeichen 19A19013T).

[4]Experiments on architectures are displayed in Appendix F

References

[1] Antonio A. Abello, Roberto Hirata, and Zhangyang Wang. Dissecting the high-frequency bias in convolutional neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 863–871, 2021. 4

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 9 2018. 2

[3] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26831–26843. Curran Associates, Inc., 2021. 8

[4] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. 5

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 5

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 5

[7] Fabio Cermelli, Dario Fontanel, Antonio Tavera, Marco Ciccone, and Barbara Caputo. Incremental learning in semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4371–4381, June 2022. 2

[8] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9230–9239, 2 2020. 2

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3, 8

[10] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 5

[11] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[12] Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5656–5668. Curran Associates, Inc., 2021. 3

[13] MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16712–16721, June 2022. 2, 4

[14] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2

[15] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[16] Jhair Gallardo, Tyler L. Hayes, and Christopher Kanan. Self-Supervised Training Enhances Online Continual Learning. In *British Machine Vision Conference (BMVC)*, 2021. 3, 4, 8

[17] Prachi Garg, Rohit Saluja, Vineeth N Balasubramanian, Chetan Arora, Anbumani Subramanian, and C.V. Jawahar. Multi-domain incremental learning for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 761–771, January 2022. 2

[18] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 1–16. Springer, 2020. 5

[19] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. 3

[21] Tobias Kalb, Niket Ahuja, Jingxing Zhou, and Jürgen Beyerer. Effects of architectures on continual semantic segmentation. *arXiv preprint arXiv:2302.10718*, 2023. 3

[22] Tobias Kalb and Jürgen Beyerer. Causes of catastrophic forgetting in class-incremental semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 56–73, December 2022. 2, 3, 4, 8

[23] Tobias Kalb, Masoud Roschani, Miriam Ruf, and Jürgen Beyerer. Continual learning for class- and domain-

incremental semantic segmentation. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 1345–1351, 2021. 2, 7

[24] Joonhyuk Kim, Sahng-Min Yoo, Gyeong-Moon Park, and Jong-Hwan Kim. Continual unsupervised domain adaptation for semantic segmentation, 2020. 2

[25] 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. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

[26] Marvin Klingner, Andreas Bär, Philipp Donn, and Tim Fingscheidt. Class-incremental learning for semantic segmentation re-using neither old data nor old labels. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020. 2

[27] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 6156–6175, 5 2019. 2

[28] Timothée Lesort. Continual feature selection: Spurious features in continual learning. *arXiv preprint arXiv:2203.01012*, 2022. 8

[29] Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. 2

[30] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. Rehearsal-free continual learning over small non-i.i.d. batches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 3

[31] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 2

[32] Robert A. Marsden, Felix Wiewel, Mario Döbler, Yang Yang, and Bin Yang. Continual unsupervised domain adaptation for semantic segmentation using a class-specific transfer, 2022. 1, 2

[33] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning, 2021. 3, 4

[34] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pages 3205–3212, 7 2019. 2

[35] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1124, 6 2021. 2

[36] M Jehanzeb Mirza, Marc Masana, Horst Possegger, and Horst Bischof. An efficient domain-incremental learning approach to drive in all weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3001–3011, 2022. 2

[37] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning, 2022. 3, 4

[38] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021. 2

[39] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc., 2020. 4

[40] Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091*, 2019. 2

[41] Quang Pham, Chenghao Liu, and HOI Steven. Continual normalization: Rethinking batch normalization for online continual learning. In *International Conference on Learning Representations*, 2022. 3, 5, 6

[42] Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021. 2, 4

[43] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: incremental classifier and representation learning. In *CVPR*, 2017. 2

[44] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[45] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018. 7, 8

[46] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 3, 4

[47] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 2

[48] Jan-Aike Termöhlen, Marvin Klingner, Leon J. Brettin, Nico M. Schmidt, and Tim Fingscheidt. Continual unsupervised domain adaptation for semantic segmentation by online frequency domain style transfer. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2881–2888, 2021. 1, 2

[49] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 4

[50] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020. 2

[51] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3

[52] Zuxuan Wu, Xin Wang, Joseph E. Gonzalez, Tom Goldstein, and Larry S. Davis. Ace: Adapting to changing environments for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2

[53] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3, 7

[54] 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 M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12077–12090. Curran Associates, Inc., 2021. 8

[55] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 2

[56] Lu Yu, Xialei Liu, and Joost van de Weijer. Self-training for class-incremental semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2022. 2

[57] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 5

[58] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995, International Convention Centre, Sydney, Australia, 8 2017. PMLR. 2