Adaptive Shortcut Debiasing for Online Continual Learning

Doyoung Kim, Dongmin Park, Yooju Shin, Jihwan Bang, Hwanjun Song, Jae-Gil Lee[*]
KAIST, Daejeon, Republic of Korea
{dodokim, dongminpark, yooju.shin, jihwan.bang, songhwanjun, jaegil}@kaist.ac.kr

Abstract

We propose a novel framework DropTop that suppresses the shortcut bias in online continual learning (OCL) while being adaptive to the varying degree of the shortcut bias incurred by continuously changing environment. By the observed high-attention property of the shortcut bias, highly-activated features are considered candidates for debiasing. More importantly, resolving the limitation of the online environment where prior knowledge and auxiliary data are not ready, two novel techniques—feature map fusion and adaptive intensity shifting—enable us to automatically determine the appropriate level and proportion of the candidate shortcut features to be dropped. Extensive experiments on five benchmark datasets demonstrate that, when combined with various OCL algorithms, DropTop increases the average accuracy by up to 10.4% and decreases the forgetting by up to 63.2%.

Introduction

Deep neural networks (DNNs) often rely on a strong correlation between peripheral features, which are usually easy-to-learn, and target labels during the learning process (Park et al. 2021; Scimeca et al. 2021). Such peripheral features and learning bias are called *shortcut features* and *shortcut bias* (Geirhos et al. 2019; Hendrycks et al. 2021; Scimeca et al. 2021; Shah et al. 2020). For example, DNNs may extract only shortcut features such as color, texture, and local or background cues; when classifying dogs and birds, only the legs (i.e., local cue) and sky (i.e., background cue) could be extracted if they are the easiest to distinguish between the two classes. The shortcut bias is an important problem in most computer vision themes such as image classification (Geirhos et al. 2020).

Online continual learning (OCL) for image classification[1] maintains a DNN to classify images from an online stream of images and tasks, where an upcoming task may include new classes (De Lange et al. 2021; Rolnick et al. 2019; Bang et al. 2021; Buzzega et al. 2020). Due to the characteristics of the online environment, there is not abundant training data (i.e., images), and the computational and memory budgets are typically tight (De Lange et al. 2021). This limited opportunity for learning exacerbates the shortcut bias in OCL, because DNNs tend to learn easy-to-learn features early on (Du et al. 2021). Therefore, we take a step forward to investigate its adverse effects in OCL.

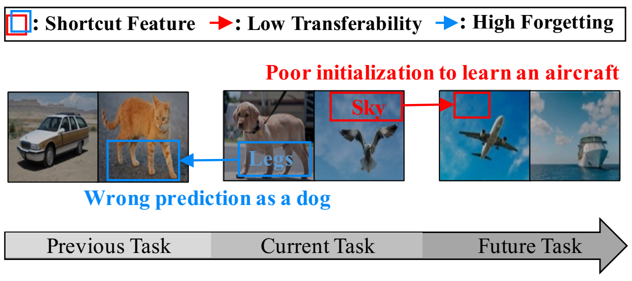


Figure 1: Negative effect of the shortcut bias in OCL: low transferability and high forgetability of shortcut features.

The shortcut bias hinders the main goal of OCL that solves the plasticity and stability dilemma for *high transferability* and *low catastrophic forgetting*. That is, it incurs *low* transferability and *high* forgetting in OCL, because shortcut features do not generalize well to unseen new classes (Geirhos et al. 2020; Park et al. 2021) and are no longer discriminative for all classes. Figure 1 illustrates the negative effect of the shortcut bias in OCL. Regarding low transferability, if an OCL model learns to use the sky as a shortcut feature for the current task, it is faced with a bad initial point for the future task. Regarding high forgetting, if an OCL model learns to use the dog's legs for the current task, it is forced to forget the prior knowledge about the legs due to the misclassification of the animals other than the dog.

A significant amount of research has gone into eliminating an undesirable (e.g., shortcut) bias in *offline* supervised learning. Representative debiasing methods require the prior knowledge of a target task to predefine the undesirable bias for unseen conditions (Bahng et al. 2020; Geirhos et al. 2019; Lee, Kim, and Nam 2019) or leverage auxiliary data such as out-of-distribution (OOD) data (Lee et al. 2021; Park et al. 2021). However, *neither* prior knowledge *nor* auxiliary data is available in OCL, because future tasks are inherently unknown and access

[*]Jae-Gil Lee is the corresponding author.

[1]Unless otherwise specified, OCL is involved with image classification in this paper.

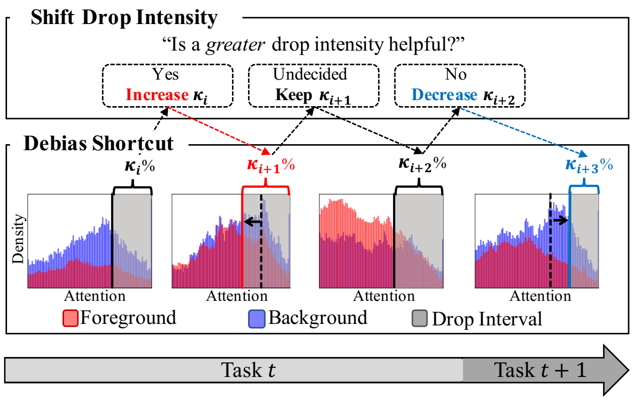


Figure 2: Adaptive intensity shifting: shows how the portion of the high-attention features to be dropped can be adjusted in Split ImageNet, where the ground-truth[2] background (shortcut) annotations are available for evaluation.

to auxiliary data is typically constrained due to a limited memory budget. Therefore, overcoming the lack of prior knowledge and auxiliary data should be the main challenge in debiasing for OCL.

For shortcut debiasing, we suppress the learning of the shortcut features in a DNN. Accordingly, those shortcut features need to be identified accurately and efficiently without help from prior knowledge and auxiliary data. To this end, we pose *two* research questions.

RQ1 "Which level of features is the most useful for identifying shortcut features?": A DNN consists of many layers, and each layer produces its own feature map. Due to the aforementioned uncertainty, we take into account both low-level features from a low-level layer and high-level features from a high-level layer, refining semantic high-level features with structural low-level features via *feature map fusion*. Our experience indicates that picking up the first and last layers is sufficient for the fusion. Meanwhile, it is known that DNNs prefer to learn shortcut features *early* owing to their simplicity (Valle-Perez, Camargo, and Louis 2019; Shah et al. 2020; Scimeca et al. 2021), so they tend to have *high attention scores*. Therefore, we drop the features with top-κ% attention scores on the *fused* feature map.

RQ2 "How much portion of high-attention features are expected to be shortcut features?": Answering this question is very challenging because the high-attention features include both shortcut and non-shortcut features, though shortcut features are expected to be dominating. Nevertheless, we endeavor to propose a novel, practical solution called *adaptive intensity shifting*. In Figure 2, if the features with top-κ% attention scores include a large number of shortcut (e.g., background) features, it is better to increase κ to drop shortcut features aggressively (see the 1st histogram); for the opposite case possibly resulted from effective debiasing, it is better to decrease κ not to drop useful features (see the 3rd histogram).

[2]The ground-truth information is used for generating the histograms for exposition purposes, but it is *not* used in DropTop.

Although the idea is very intuitive, there is no ground-truth for shortcut features in practice. Instead, we examine the *loss reduction* followed by two shift directions—increment and decrement. A bigger loss reduction can be accomplished if more of shortcut features are dropped. The benefit of this loss-based approach is that estimating the dominance of shortcut features in high-attention features does not require additional overhead.

Overall, answering the two questions, we develop a novel framework, ***DropTop***, for suppressing shortcut features in OCL. It is model-agnostic and thus can be incorporated into any replay-based OCL method; *two* widely-used backbones, a convolutional neural network (CNN) (He et al. 2016) and a Transformer (Dosovitskiy et al. 2021), are adopted for the evaluation. The main contributions of the paper are summarized as follows:

- To the best of our knowledge, this is the *first* work to address the shortcut bias in OCL. Moreover, we theoretically analyze its negative effect in OCL.
- We present feature map fusion and adaptive intensity shifting to get around the uncertainty caused by the absence of prior knowledge and auxiliary data.
- We empirically show that DropTop improves the average accuracy and forgetting of seven representative OCL methods by up to 10.4% and 63.2%, respectively, on conventional benchmarks as well as newly-introduced benchmarks, namely Split ImageNet-OnlyFG and Split ImageNet-Stylized.

Related Work

Online Continual Learning Recent studies have mainly considered an online environment for more practical applications (Rolnick et al. 2019; Buzzega et al. 2020; Prabhu, Torr, and Dokania 2020; Bang et al. 2021; Koh et al. 2022; Chaudhry et al. 2018; Kirkpatrick et al. 2017). Access to the data from the current task is permitted until model convergence in an offline environment, but only once in an online environment. Thus, online continual learning (OCL) is much more challenging than offline continual learning. Representative OCL methods are mainly categorized into two groups: *replay-based* strategies exploiting small episodic memory to consolidate the knowledge of old tasks (Rolnick et al. 2019; Buzzega et al. 2020; Prabhu, Torr, and Dokania 2020; Koh et al. 2022) and *regularization-based* strategies penalizing a model for rapid parameter updates to avoid the forgetting of old tasks (Chaudhry et al. 2018; Kirkpatrick et al. 2017). In general, replay-based approaches outperform regularization-based ones in terms of accuracy and computational efficiency. Thus, we focus on the replay-based approaches in this paper.

Replay-based approaches suggest the policies for keeping more useful instances. ER (Rolnick et al. 2019) and DER++ (Buzzega et al. 2020) propose random sampling and reservoir sampling, respectively. DER++ (Buzzega et al. 2020) additionally utilizes knowledge distillation (Hinton et al. 2015) to better retain previous knowledge. MIR (Aljundi et al. 2019a) selects the samples whose loss increases the most by a preliminary model update.

GSS (Aljundi et al. 2019b) chooses the samples diversifying their gradients. Last, ASER (Shim et al. 2021) retrieves the samples that better preserve latent decision boundaries for known classes while learning new classes.

Debiasing Shorcut Features The negative impact of shortcuts bias has gained a great attention in the deep learning community (Ilyas et al. 2019; Wang et al. 2020). To prevent overfitting to the shortcuts, several methods predefine the type of the target shortcut bias. ReBias (Bahng et al. 2020) removes pixel-level local shortcuts using a set of biased predictions from a bias-characterizing model. Stylized-ImageNet (Geirhos et al. 2019) generates texture-debiased examples. SRM (Lee, Kim, and Nam 2019) synthesizes debiased examples via generative modeling. Another direction is to leverage auxiliary OOD data (Lee et al. 2021; Park et al. 2021), which relies on undesirable features found in the OOD data. However, these existing methods are not directly applicable to our environment, because enforcing prior knowledge or auxiliary data violates the philosophy of OCL.

Moreover, the adverse impact of shortcuts is amplified when training data is insufficient (Lee et al. 2021)—as in OCL involved with an online stream of small tasks. Overall, lack of the prerequisite and rich training data calls for a new debiasing method dedicated for OCL.

Preliminary

Online Continual Learning We consider the online continual learning setting, where a sequence of tasks continually emerges with a set of new classes. Each data instance can be accessed only once unless it is stored in episodic memory. For the t-th task, let $D_t = \{(x_t^i, y_t^i)\}_{i=1}^{m_t}$ be a data stream obtained from a joint data distribution over $\mathcal{X}_t \times \mathcal{Y}_t$, where $\mathcal{X}_t$ is the input space, $\mathcal{Y}_t$ is the label space in a one-hot fashion, and m_t is the number of instances of the t-th task. The goal of OCL is to train a classifier, such that it maximizes the test accuracy of all seen tasks $T = \{1, 2, \ldots, t\}$ at time t, with limited or no access to the data stream $D_{t'}$ of previous tasks $t' < t$.

Shortcut and Non-shortcut Features Let a DNN model trained on $(x, y) \sim \mathcal{X} \times \mathcal{Y}$ comprises a general feature extractor $f^\theta : \mathcal{X} \rightarrow \mathcal{Z} \in \mathbb{R}^d$ and a final classifier layer $g^{\mathrm{w}} : \mathcal{Z} \rightarrow \mathbb{R}^k$, which is the multiplication of a linear weight matrix $\mathrm{w} \in \mathbb{R}^{k \times d}$ and a feature $z \in \mathcal{Z}$ such that $g^{\mathrm{w}}(z) = w \cdot z$. Here, k is the number of classes, and d is the dimensionality of a feature z. Let a *feature* be the function mapping from the input space $\mathcal{X}$ to a real number; formally speaking, for the i-th sub-feature extractor of f^θ where $i \in \{1, 2, \ldots, d\}$, $f_i^\theta : \mathcal{X} \rightarrow \mathbb{R}$. DNNs are known to have simplicity bias toward shortcut features which can easily distinguish between given classes (Geirhos et al. 2019; Shah et al. 2020; Geirhos et al. 2020; Pezeshki et al. 2021). Then, the *shortcut* and *non-shortcut* features are defined by Definitions 1 and 2, respectively.

Definition 1. (SHORTCUT FEATURE). Let x and $\tilde{x}$ be the instances from seen and unseen data distributions $\mathcal{X}$ and $\tilde{\mathcal{X}}$, respectively, where $\mathcal{X} \cap \tilde{\mathcal{X}} = \emptyset$. We call a feature a *shortcut* if it is not only highly activated ($\geq \rho$) for the seen data instance but also undesirably activated ($\geq \epsilon$) for the unseen data instance in expectation. That is,

$$f_s^\theta(\rho, \epsilon) : \mathbb{E}_{x \sim \mathcal{X}}\big[|f_s^\theta(x)|\big] \geq \rho \wedge \mathbb{E}_{\tilde{x} \sim \tilde{\mathcal{X}}}\big[|f_s^\theta(\tilde{x})|\big] \geq \epsilon, \quad (1)$$

where s is the index of the shortcut feature. ρ and ϵ are the thresholds indicating high activation for the seen and unseen data distributions.

Note that a shortcut feature, such as an animal's legs, can be observed from the unseen data instances because it is not an intrinsic feature.

Definition 2. (NON-SHORTCUT FEATURE). Let x and $\tilde{x}$ be the instances from seen and unseen data distributions $\mathcal{X}$ and $\tilde{\mathcal{X}}$, respectively, where $\mathcal{X} \cap \tilde{\mathcal{X}} = \emptyset$. We call a feature a *non-shortcut* if it is highly activated only for the seen instance in expectation. That is,

$$f_n^\theta(\rho, \epsilon) : \mathbb{E}_{x \sim \mathcal{X}}\big[|f_n^\theta(x)|\big] \geq \rho \wedge \mathbb{E}_{\tilde{x} \sim \tilde{\mathcal{X}}}\big[|f_n^\theta(\tilde{x})|\big] < \epsilon. \quad (2)$$

Property of Shortcut Feature Due to a DNN's simplicity bias, the shortcut features tend to have higher activation values than the non-shortcut features (Shah et al. 2020; Pezeshki et al. 2021), such that

$$\mathbb{E}_{s \in \mathcal{S}}\big[|f_s^\theta(x)|\big] > \mathbb{E}_{n \in \mathcal{N}}\big[|f_n^\theta(x)|\big], \quad (3)$$

where $\mathcal{S}$ and $\mathcal{N}$ are the set of the indices of shortcut and non-shortcut features, respectively.

Proposed Method: DropTop

DropTop realizes adaptive feature suppression thorough (a) *attentive debiasing* with *feature map fusion* and (b) *adaptive intensity shifting*. The former drops high-attention features which are expected to be shortcuts, following the guidance from the latter in the form of an adjusted drop intensity. Figure 3 illustrates the detailed procedure of DropTop. While (a) attentive debiasing is executed during the learning process, the drop intensity κ is periodically adjusted by (b) adaptive intensity shifting. DropTop can be easily implemented on top of any existing replay-based OCL methods, and see the implementation details in Appendix A, where the full version (including the appendices) is available at Kim et al. (2023).

Attentive Debiasing

The shortcut features generally exhibit higher attention scores than the non-shortcut feature by Eq. (3). This shortcut bias usually appears as local or background cues (Scimeca et al. 2021). Thus, after fusing the feature maps of different levels, we generate a drop mask to debias the effect of local or background cues on the feature activation.

Feature Map Fusion The high-level features are essential to obtain discriminative signals, but they lack fine-grained details (Zhong, Ling, and Wang 2019; Wei et al. 2021), e.g., boundaries of objects. The low-level features embrace structural information with more details, which is helpful for accurately finding shortcut regions as empirically shown in

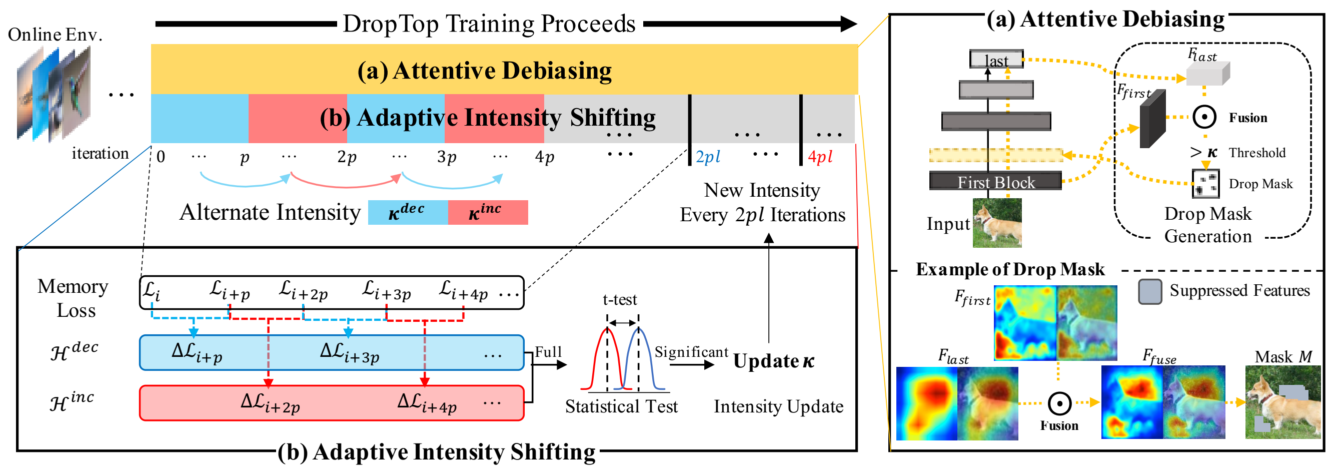


Figure 3: Detailed view of DropTop: (a) drops the highly-activated features on the fused feature map according to the intensity κ; as in the example of the drop mask, the fusion facilitates identifying the background shortcuts from the signal of the high-level features on important parts such as the body of the dog. (b) adjusts the drop intensity κ for the use in attentive debiasing.

Section Ablation Study. Thus, feature map fusion facilitates the reduction of the ambiguity in identifying the shortcut regions. Specifically, as shown in Figure 3(a), given a small memory footprint, we fuse the feature maps $F_{first} \in \mathbb{R}^{h\times w\times c}$ and $F_{last} \in \mathbb{R}^{h'\times w'\times c'}$ respectively from the first and last layers. To cope with different resolutions, i.e., $\langle h,w,c\rangle \neq \langle h',w',c'\rangle$, the last layer's feature map F_{last} is up-sampled to have the same resolution as the first layer's feature map F_{first} such that $h' = h$ and $w' = w$. Then, we conduct average pooling along the channel dimension, compressing the outputs of all c channels to produce the attention map A_{fuse}, which is derived by

$$F_{fuse} = F_{first} \odot \text{Upsample}(F_{last}) \in \mathbb{R}^{h\times w\times c} \text{ and}$$
$$A_{fuse} = \text{ChannelPool}(F_{fuse}) \in \mathbb{R}^{h\times w}, \tag{4}$$

where ChannelPool$(\cdot)$ is the pooling along the channel, and $\odot$ is the element-wise tensor product.

Drop Mask Generation Given a drop intensity $\kappa\%$ for each class[3], we generate a *drop mask* $M \in \mathbb{R}^{h\times w}$ based on the attention map A_{fuse} in Eq. (4). The mask at (i,j) in M is determined by

$$M_{i,j} = \begin{cases} 0 & \text{if } (i,j) \in \text{top-}\kappa(A_{fuse}) \\ 1 & \text{otherwise,} \end{cases} \tag{5}$$

where top-$\kappa(A_{fuse})$ returns the set of the top-$\kappa\%$ elements of A_{fuse}. While this *hard* drop mask is simple and effective, we also test a *soft* drop mask with continuous values in Appendix E. Last, we apply the drop mask to the first feature map F_{first} after the stem layer of a backbone network, e.g., a ResNet or a Vision Transformer (ViT),

$$\tilde{F}_{first} = M \odot F_{first}. \tag{6}$$

[3]The class is used only during the training process for debiasing. We do not perform such feature suppression during testing.

This masking process affects all succeeding layers along the forward process. The masked feature map $\tilde{F}_{first}$ accomplishes debiasing by using only the attention scores which can be computed on the fly.

Adaptive Intensity Shifting

The extent of the shortcut bias naturally varies depending on the incoming tasks and the learning progress of a DNN model. Thus, *adaptive intensity shifting* aims to guide attentive debiasing by adaptively adjusting the drop intensity κ in a timely manner. The drop intensity is maintained separately for each class to capture diverse sensitivity to shortcut bias. (See Appendix E for an additional test of sharing the drop intensity across classes.) It is *decreased* if the removal of features is expected to lose important non-shortcut features which have high predictive power, *increased* if the removal of features is expected to help emphasize important non-shortcut features, and *unchanged* in neither of the cases,

$$\kappa \leftarrow \begin{cases} \kappa' * \alpha & \text{if decrement} \\ \kappa' * (1/\alpha) & \text{if increment} \\ \kappa' & \text{otherwise,} \end{cases} \tag{7}$$

where κ' is the preceding value, and $\alpha\,(< 1.0)$ is a hyperparameter that indicates the shift step.

Loss Collection To classify each case, we focus on the contribution to the reduction of the training loss because a better choice of the drop intensity leads to a better training performance (Scimeca et al. 2021; Geirhos et al. 2020). To determine whether a decrement or an increment is preferable, two potential values $\kappa = \kappa' * \alpha$ and $\kappa = \kappa' * (1/\alpha)$ alternate every p iterations. p is set to be long enough to observe *stable* behavior with respect to each κ option, as shown in Appendix D. The loss reductions are computed at the end of every p iterations and maintained in the set $\mathcal{H}^{dec}$ when $\kappa' * \alpha$ is used and in the set $\mathcal{H}^{inc}$ when $\kappa' * (1/\alpha)$ is used, as shown in Figure 3(b),

$$\mathcal{H}^{dec} = \{\Delta\mathcal{L}_0, \Delta\mathcal{L}_{2p}, \ldots, \Delta\mathcal{L}_{2p(l-1)}\}\text{and}$$
$$\mathcal{H}^{inc} = \{\underbrace{\Delta\mathcal{L}_p, \Delta\mathcal{L}_{3p}, \ldots, \Delta\mathcal{L}_{2pl}}_{\text{Collect } \Delta\mathcal{L} \text{ every } 2p \text{ iterations}}\}, \tag{8}$$

where $\Delta\mathcal{L}_{(q+1)\cdot p} = \mathcal{L}_{q\cdot p} - \mathcal{L}_{(q+1)\cdot p}$, and $\mathcal{L}_{(q+1)\cdot p}$ is the expected cross-entropy loss on the samples of a specific class in the memory buffer at the iteration $(q+1)\cdot p$. Note that we compute the loss using the samples in the memory buffer rather than in a batch to obtain a more generalizable training loss. Those loss reductions are recorded until $\mathcal{H}^{dec}$ and $\mathcal{H}^{inc}$ have l elements. Only a single model is needed to compare two shift directions, preserving the memory constraint of OCL; the validity of using a single model is empirically confirmed in Section Evaluation.

Statistical Test and Update When $\mathcal{H}^{dec}$ and $\mathcal{H}^{inc}$ become full every $2 \cdot p \cdot l$ iterations, we conduct t-test to evaluate if the difference between the two directions in the loss reduction is statistically significant, i.e., either p-value ≤ 0.05 or p-value ≥ 0.95. If a better direction exists, the preceding value is updated to the better direction; otherwise, it remains the same. Appendix A describes the pseudocode of adaptive intensity shifting, which is self-explanatory.

Training Data Stabilization Furthermore, we consider the side effect of shifting the drop intensity during training. If it changes over time, the distribution of input features could be swayed inconsistently. As widely claimed in Ioffe and Szegedy; Zhou et al., this problem often causes training instability and eventually deteriorates the overall performance. In this sense, we decide to drop $\gamma\%$ of the features *constantly*. Thus, κ cannot grow beyond γ; if $\kappa < \gamma$, $(\gamma - \kappa)\%$ of the features are additionally chosen uniformly at random among those whose drop mask is not 1.

Understanding of Shortcut Bias in OCL

From a theoretical standpoint, we clarify two detrimental effects of learning shortcut features in OCL. See Appendix B for relevant empirical evidences. Let $g_i^{\mathrm{w}} : \mathbb{R} \to \mathbb{R}^k$ be the multiplication of an i-th column vector of the weight matrix w and an i-th feature f_i^θ. Then, $g_i^{\mathrm{w}}(f_i^\theta(x))$ denotes the contribution of the i-th feature to a prediction, i.e., how much the i-th feature $f_i^\theta(x)$ contributes to the prediction $g^{\mathrm{w}}(f^\theta(x))$, since $g^{\mathrm{w}}(f^\theta(x)) = \sum_{i=1}^d g_i^{\mathrm{w}}(f_i^\theta(x))$.

Property 1 (LOW TRANSFERABILITY)**.** A shortcut feature of the current task hinders the model from learning the knowledge of the next task.

Proof. Let's consider training a classifier for the (t+1)-th task, given a shortcut biased model θ_t as the initial model. By Eq. (1), $\mathbb{E}_{x_{t+1}\sim\mathcal{X}_{t+1}}\big[|f_s^{\theta_t}(x_{t+1})|\big] \geq \epsilon$ if x_{t+1} shares the same shortcut features with x_t. Then, when training a new model θ_{t+1} from the model θ_t, the shortcut feature $f_s^{\theta_t}$ is no longer discriminative since it can appear in both $\mathcal{X}_t$ and $\mathcal{X}_{t+1}$. Thus, the model θ_{t+1} should learn a totally new discriminative feature, starting from a poor initialization. As a result, the training convergence becomes slower compared with a non-biased model θ_t^*. ■

Property 2 (HIGH FORGETABILITY)**.** A shortcut feature of the current task makes the model forget the knowledge of the previous tasks.

Proof. Given the previous tasks at $t' \in \{1, \ldots, t-1\}$ and the current task at t, let's assume that we trained a shortcut biased model θ_t. Then, the shortcut feature s of f^{θ_t} contributes to predict the class $y_t \in \mathcal{Y}_t$ such that $\operatorname{argmax} g_s^{\mathrm{w}_t}(f_s^{\theta_t}(x_t)) = y_t \in \mathcal{Y}_t$. However, by Eq. (1), $\mathbb{E}_{x_{t'}\sim\mathcal{X}_{t'}}\big[|f_s^{\theta_t}(x_{t'})|\big] \geq \epsilon$ if $x_{t'}$ shares the same shortcut features with x_t. Then, even for the previous instance $x_{t'}$, the shortcut feature s contributes to predict the class $y_t \in \mathcal{Y}_t$ such that $\operatorname{argmax} g_s^{\mathrm{w}_t}(f_s^{\theta_t}(x_{t'})) = y_t \notin \mathcal{Y}_{t'}$, which is a wrong prediction. Due to the high activation of shortcut features in Eq. (3), this wrong feature contribution gives a significant impact to the final prediction $g^{\mathrm{w}}(f^\theta(x_{t'})) = \sum_{i=1}^d g_i^{\mathrm{w}}(f_i^\theta(x_{t'}))$, so that the model highly forgets the knowledge of the previous tasks. ■

Evaluation

Experiment Setting

Dataset Preparation We validate the debiasing efficacy of DropTop on *biased* and *unbiased* setups. A biased setup measures the OCL performance on biased test datasets, where the test dataset shares the same distribution of shortcut features with the training dataset, e.g., the sky background in the bird class. On the other hand, an unbiased setup (Bahng et al. 2020) does not contain the shortcut features in the test dataset, so that we can more clearly measure the debiasing efficacy.

Biased Setup: We use the *Split CIFAR-10* (Krizhevsky, Hinton et al. 2009), *Split CIFAR-100* (Krizhevsky, Hinton et al. 2009), and *Split ImageNet-9* (Xiao et al. 2020) for the biased setup. Split CIFAR-10 and Split CIFAR-100 consist of five different tasks with non-overlapping classes, and thus each task contains two and 20 classes, respectively. The details of Split ImageNet-9 are provided in Appendix C.

Unbiased Setup: The evaluation of unbiased setup is conducted after being trained on *Split ImageNet-9*. We introduce two *Split ImageNet-9* variants: *Split ImageNet-OnlyFG* and *Split ImageNet-Stylized*, which are generated by debiasing two realistic shortcuts, the background and the local texture cue, respectively. In ImageNet-OnlyFG (Xiao et al. 2020), the background is removed to evaluate the dependecy on the background in image recognition; in ImageNet-Stylized (Geirhos et al. 2019), the local texture is shifted by style-transfer, and the reliance of a model on the local texture cue is removed. Performance improvements on these two datasets indicate that DropTop helps untie undesirable dependency on the realistic shortcuts.

Algorithms and Evaluation Metrics We apply DropTop to seven popular algorithms including ER (Rolnick et al. 2019) and DER++ (Buzzega et al. 2020), MIR (Aljundi et al. 2019a), GSS (Aljundi et al. 2019b), ASER (Shim et al.

	Biased Setup						**Unbiased Setup**			
	Split CIFAR-100		**Split CIFAR-10**		**Split IN-9**		**Split IN-OnlyFG**		**Split IN-Stylized**	
Method	A_{avg}	F_{last}	A_{avg}	F_{last}	A_{avg}	F_{last}	A_{avg}	F_{last}	A_{avg}	F_{last}
ER	25.3 (±0.2)	19.2 (±0.3)	55.4 (±0.3)	39.9 (±2.4)	46.1 (±0.2)	55.8 (±1.8)	32.8 (±0.4)	**45.0** (±1.5)	39.3 (±0.3)	49.7 (±0.9)
+**DropTop**	**26.5** (±0.2)	**18.3** (±1.3)	**61.0** (±0.6)	**37.6** (±1.9)	**48.9** (±0.4)	**44.0** (±3.8)	**35.9** (±0.9)	47.8 (±3.7)	**41.3** (±0.1)	**38.9** (±2.6)
Rel. Improv.	4.4%	4.9%	10.2%	5.7%	6.1%	21.1%	9.4%	-6.2%	5.1%	21.8%
DER++	23.6 (±0.3)	33.4 (±0.7)	59.8 (±0.8)	28.2 (±1.5)	44.2 (±1.0)	63.0 (±4.6)	33.2 (±0.5)	61.6 (±2.1)	37.8 (±0.7)	60.3 (±3.2)
+**DropTop**	**25.1** (±0.3)	**31.1** (±1.2)	**62.6** (±0.7)	**23.7** (±1.0)	**45.4** (±0.5)	**60.7** (±5.1)	**34.5** (±0.7)	**54.9** (±3.2)	**39.7** (±0.4)	**58.9** (±4.1)
Rel. Improv.	6.4%	6.9%	4.7%	16.0%	2.8%	3.6%	3.7%	11.0%	4.9%	2.4%
MIR	20.8 (±0.6)	25.3 (±1.7)	51.5 (±0.5)	58.4 (±1.3)	39.3 (±1.2)	41.1 (±2.0)	32.1 (±0.4)	38.9 (±3.1)	34.0 (±1.0)	40.7 (±1.6)
+**DropTop**	**22.4** (±0.1)	**24.8** (±1.0)	**51.6** (±0.6)	**56.3** (±1.6)	**42.0** (±0.9)	**39.3** (±4.5)	**35.4** (±1.4)	**36.0** (±3.1)	**36.1** (±0.6)	**38.3** (±3.4)
Rel. Improv.	8.1%	1.9%	0.2%	3.7%	6.8%	4.4%	10.4%	7.5%	6.0%	6.0%
GSS	21.8 (±0.3)	**27.9** (±0.8)	46.0 (±0.4)	69.7 (±0.7)	40.3 (±0.6)	71.5 (±2.0)	34.5 (±0.9)	65.6 (±0.7)	35.3 (±0.7)	64.2 (±1.9)
+**DropTop**	**23.7** (±0.2)	28.4 (±1.0)	**49.1** (±0.5)	**69.2** (±1.5)	**41.3** (±0.6)	**68.5** (±1.9)	**36.5** (±0.7)	**61.6** (±1.8)	**35.9** (±0.4)	**62.3** (±2.3)
Rel. Improv.	8.8%	-1.7%	6.5%	0.6%	2.3%	4.2%	5.9%	6.1%	1.8%	2.9%
ASER	32.7 (±0.1)	41.8 (±0.4)	50.3 (±0.5)	57.8 (±3.3)	39.7 (±0.7)	62.6 (±2.1)	30.2 (±0.2)	60.2 (±0.9)	33.6 (±0.5)	56.8 (±1.1)
+**DropTop**	**33.4** (±0.4)	**39.5** (±0.8)	**51.3** (±0.3)	**57.5** (±0.8)	**41.6** (±0.8)	**57.5** (±3.2)	**31.2** (±0.2)	**49.6** (±2.8)	**35.5** (±0.4)	**52.6** (±2.5)
Rel. Improv.	2.3%	5.3%	2.0%	0.4%	4.8%	8.2%	3.4%	17.7%	5.7%	7.3%

Table 1: Average accuracy (higher is better) and forgetting (lower is better) on Split CIFAR-100, Split CIFAR-10, and Split ImageNet-9 (IN-9) for a biased setup and on Split ImageNet-OnlyFG (IN-OnlyFG) and Split ImageNet-Stylized (IN-Stylized) for an unbiased setup. The best values are highlighted in bold.

2021), L2P (Wang et al. 2022b), and DualPrompt (Wang et al. 2022a). They are all actively being cited. We adopt widely-used performance measures (Aljundi et al. 2019b; Shim et al. 2021; Koh et al. 2022): (1) *average accuracy* $A_{avg} = \frac{1}{T}\Sigma_{i=1}^{T} A_i$, where A_i is the accuracy at the end of the i-th task, and (2) *forgetting* $F_{last} = \frac{1}{T-1}\Sigma_{j=1}^{T-1} f_{T,j}$, where $f_{i,j}$ indicates how much the model forgets about the j-th task after learning the i-th task ($j < i$). To ensure the reliability of the experimental results, we repeat every experiment *five* times with different random seeds and report the average value with the standard error. We report the relative improvements (denoted as "Rel. Improv.") by DropTop for each metric.

Implementation Details For all algorithms and datasets, the size of a minibatch from the data stream and the replay memory is set to 32, following (Buzzega et al. 2020). The size of episodic memory is set to 500 for Split CIFAR-10 and Split ImageNet-9 and 2,000 for Split CIFAR-100 depending on the total number of classes. In order to add DropTop's debiasing capability to L2P and DualPrompt, which are originally rehearsal-free, we attach small replay memory whose size is reduced to 200 and 500, respectively. We train ResNet18 using SGD with a learning rate of 0.1 (Buzzega et al. 2020; Shim et al. 2021) for all ResNet-based algorithms. We optimize L2P and DualPrompt with a pretrained ViT-B/16 using Adam with a learning rate of 0.05, β_1 of 0.9, and β_2 of 0.999. All algorithms are implemented using PyTorch 1.12.1 and tested on a single NVIDIA RTX 2080Ti GPU, and the source code is available at https://github.com/kaist-dmlab/DropTop. More implementation details are presented in Appendix E.

Hyperparameters There are a few hyperparameters introduced by DropTop. For attentive debiasing, we fix the total drop ratio γ to 5.0% and set the initial drop intensity κ_0 to 5.0% for ER, DER++, and MIR and to 0.5% for GSS and ASER, differently depending on the sampling method. For L2P and DualPrompt, we set γ and κ_0 to 2.0% and 1.0%, respectively, owing to the difference of the backbone network. For adaptive intensity shifting, we fix the history length l to 10, which is approximately the minimum sample size for one-sided t-test with the significance level 0.05 (Heckert et al. 2002). The alternating period $p = 3$ and the shifting step size $\alpha = 0.9$ are adequate across the algorithms and datasets. Please see Appendix D for the sensitivity analysis of the hyperparameters and Appendix E for the hyperparameters for the other algorithms.

Method	IN-9 A_{avg}	IN-9 F_{last}	IN-OnlyFG A_{avg}	IN-OnlyFG F_{last}	IN-Stylized A_{avg}	IN-Stylized F_{last}
L2P	95.3 (±0.7)	12.6 (±1.9)	91.0 (±0.7)	20.6 (±2.2)	87.7 (±0.5)	25.8 (±1.8)
+DropTop	**97.1** (±0.4)	**5.6** (±1.1)	**92.9** (±0.4)	**12.7** (±0.6)	**89.7** (±0.8)	**18.3** (±1.0)
Rel. Improv.	1.9%	55.4%	2.1%	38.3%	2.3%	28.9%
DualPrompt	96.1 (±0.6)	9.9 (±1.4)	90.8 (±1.2)	20.0 (±2.0)	86.3 (±1.5)	29.2 (±2.4)
+DropTop	**97.5** (±0.3)	**3.7** (±0.8)	**93.5** (±0.5)	**9.8** (±0.5)	**89.9** (±0.6)	**16.6** (±0.8)
Rel. Improv.	1.5%	63.2%	2.9%	51.0%	4.2%	43.1%

Table 2: Performance of DropTop on top of pretrained ViT-based CL algorithms, L2P and DualPrompt, on Split ImageNet-9, Split ImageNet-OnlyFG, and Split ImageNet-Stylized. The highest values are marked in bold.

Improvements through Debiasing

Biased Setup Table 1 summarizes the performance of five replay-based OCL methods *with* and *without* applying DropTop under the biased and unbiased setups. Overall, DropTop consistently improves their performances with respect to A_{avg} and F_{last}. Quantitatively, A_{avg} and F_{last} are improved by 5.3% and 6.5%, respectively, on average across all baselines and datasets. That is, the debiasing by DropTop is very effective in OCL regardless of the method and dataset. In addition, the high A_{avg} and the low F_{last} demonstrate that the proposed debiasing approach helps expedite the training convergence of OCL models (higher transferability) and retain the knowledge of the previous tasks well (lower forgetability).

In detail, the effect of our method is the most prominent for ER, increasing A_{avg} by 7.0% on average. In particular, DropTop improves the performances of ER and DER++ on Split CIFAR-10 and Split CIFAR-10 by 10.2% and 9.4% in terms of A_{avg} and by 5.7% and 16.0% in terms of F_{last}, respectively. In contrast, ASER is improved relatively less. Since ASER is designed for keeping the samples with high diversity in the memory, we conjecture that the adverse effect of shortcut features is naturally smaller than random sampling as in ER and DER++.

Unbiased Setup The improvement of DropTop becomes more noticeable in the unbiased setup than in the biased setup, because this unbiased setup does not contain the shortcut features in the test datasets. For example, in Table 1, the relative improvement for MIR on Split ImageNet-OnlyFG is 10.4%, while that on Split ImageNet-9 is comparatively lower, amounting to 6.8%. In addition, DropTop is more effective in reducing the background bias than the local cue bias; A_{avg} improves by 6.6% on average for all methods in Split ImageNet-OnlyFG, which is larger than 4.7% increase observed in Split ImageNet-Stylized. Achieving high accuracy for the unbiased setup requires robustly generalizing to more intrinsic and complex features beyond less generalizable shortcut features. Obviously, attentive debiasing coordinated by adaptive intensity shifting supports the requirement by suppressing the undesirable reliance on shortcut features.

Method	No MF	No Drop Int. Shift: Rand Drop	No Drop Int. Shift: Fixed Drop	DropTop
ER	43.8	43.9	44.1	**45.5**
DER++	42.0	41.6	43.4	**44.4**
MIR	38.0	37.8	**39.3**	38.7
GSS	34.7	34.8	37.0	**38.0**
ASER	40.5	41.4	40.5	**42.1**
Degrade	**-4.9%**	**-4.5%**	**-2.1%**	-

Table 3: Ablation study on multi-level feature fusion and drop intensity shifting with respect to the accuracy A_{avg} averaged over the biased datasets: Split CIFAR-100, Split CIFAR-10, and Split ImageNet-9. The highest values are marked in bold.

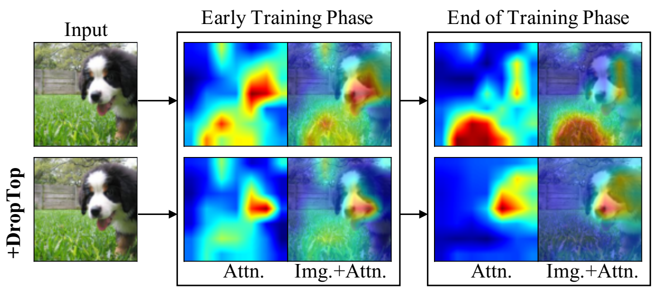
Debiasing Pretrained ViT-based CL

We test DropTop's adaptability to a pretrained ViT since it has been used frequently in recent CL studies, by injecting DropTop into L2P and DualPrompt in an online setting. Table 2 shows the performance of L2P and DualPrompt *with* and *without* DropTop on the Split ImageNet datasets. We note that both versions of L2P or DualPrompt are modified to perform experience replay (Rolnick et al. 2019) using the small replay buffer for a fair comparison. Overall, DropTop consistently exhibits significant improvements in both forgetting and accuracy, where the accuracy is initially high due to pretraining. Quantitatively, F_{last} and A_{avg} are improved by 46.7% and 2.5%, respectively, on average across the datasets and algorithms. This result clearly demonstrates the universal need for mitigating the shortcut bias in different backbone networks as well as DropTop's excellent adaptability. Please refer to Appendix F for the result on more datasets.

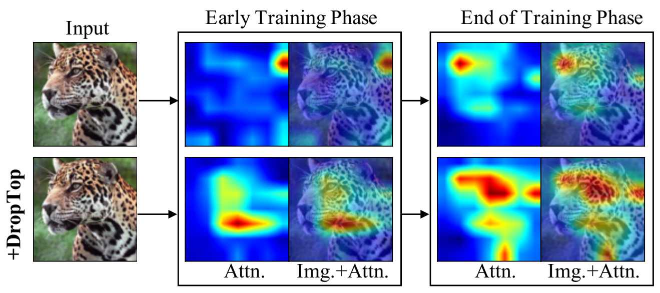
Ablation Study

We conduct an ablation study to examine the general effectiveness of multi-level feature fusion in Eq. (4) and adaptive intensity shifting in Eq. (7). Table 3 compares DropTop with its three variants with respect to the average accuracy across the biased datasets (Split CIFAR-100, Split CIFAR-10, and Split ImageNet-9). The results for the unbiased datasets are presented in Appendix E. *No MF* drops only high-level features without multi-level fusion, *Rand Drop* drops randomly chosen features, and *Fixed Drop* drops the features with the highest attention scores without adaptive intensity shifting.

Multi-level Feature Fusion The first variant, No MF, omits multi-level feature fusion for generating the drop masks, which solely relies on the high-level features. As a result, it shows the worst average accuracy among the variants. In particular, compared to DropTop which relies on the first and last layers for multi-level feature fusion, the average accuracy drops significantly by 4.9% on average. Therefore, refining semantic high-level features



(a) Debiasing the background shortcut bias.



(b) Debiasing the local cue bias.

Figure 4: Visualization of the gradual debiasing process by DropTop. The results are the activation maps from ER and ER+DropTop trained on Split ImageNet-9: (a) and (b) are respectively related to debiasing the background and local cue bias during training, where the bias is the reliance on the grass background in (a) and on a local part of a leopard (e.g., part of the face without the body) in (b).

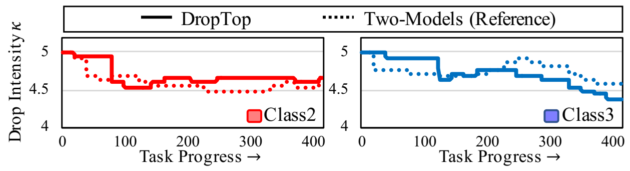


Figure 5: Comparison between the original and two-model versions of DropTop regarding the adjustments of the drop intensity κ for Split CIFAR-10 on top of ER.

with structural low-level features via feature map fusion is essential to precisely identify the shortcut features.

Adaptive Intensity Shifting The other two variants, Rand Drop and Fixed Drop, do not apply adaptive intensity shifting while dropping the same proportion of the features as DropTop. Based on their accuracy worse than DropTop, we conclude that drop intensity shifting makes further improvements. Quantitatively, without drop intensity shifting, they face the average degradation of 4.5% and 2.1% compared with the complete DropTop. Therefore, adaptively adjusting the drop intensity is needed for effective debiasing to catch the continuously varying proportion of the shortcut features in a timely manner. Furthermore, the superior performance of DropTop as well as Fixed Drop over Rand Drop verifies that debiasing based on highly activated features cannot be easily replaced by the regularization effect of the random drop (Srivastava et al. 2014).

Qualitative Analysis for Debiasing Effect

Figure 4 visualizes the activation maps of ER *with* and *without* DropTop on Split ImageNet-9 to qualitatively analyze the debiasing effect. In summary, we observe that DropTop gradually alleviates the shortcut bias as the training proceeds whereas the original ER increasingly relies on the shortcut bias—the background and local cue. In Figure 4(a) for the background bias, as the training proceeds, ER+DropTop successfully debiases the grass background and, eventually, focuses on the intrinsic shapes of the dog at the end of training whereas ER rather intensifies the reliance on the background. In Figure 4(b) for the local cue bias, ER+DropTop expands the extent of the discriminative regions to cover the overall shapes of the object, resulting in more comprehensive recognition of the leopard compared with ER. Please see Appendix H for more visualizations.

Validity of Drop Intensity Adjustments

One might think that alternating the increment and decrement of κ with a single model is invalid because the effect of one option may influence that of the other option. Using only a single model is inevitable because of the memory restriction in OCL. Thus, by assuring the period for each option to be long enough, we aim to reduce undesirable influence between the two options. To verify the validity of our approach, we implement a two-model "reference" version that maintains two separate models, each for increment and decrement, but does violate the memory restriction. As shown in Figure 5, the original version adjusts κ very similarly to the reference version. Quantitatively, 82.1±0.6% of the adjustments of κ coincide with each other, and their accuracy is nearly identical, 61.0±0.6% and 61.5±0.2% for the test.

Conclusion

We propose a debiasing OCL method called **DropTop**, introducing two novel solutions for debiasing shortcut features, *attentive debiasing* with *feature map fusion* and *adaptive intensity shifting*. Without relying on prior knowledge and auxiliary data, DropTop determines the appropriate level and proportion of possible short features and drops them from the feature map for debiasing. It can easily be built on top of any existing replay-based OCL methods. The evaluation confirms that DropTop significantly improves the existing state-of-the-art OCL methods. Overall, we believe that our work sheds the light on the importance of debiasing shortcuts in OCL.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00157, Robust, Fair, Extensible Data-Centric Continual Learning).

References

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019a. Online Continual Learning with Maximal Interfered Retrieval. In *NeurIPS*, 11849–11860.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019b. Gradient Based Sample Selection for Online Continual Learning. In *NeurIPS*.

Bahng, H.; Chun, S.; Yun, S.; Choo, J.; and Oh, S. J. 2020. Learning De-biased Representations with Biased Representations. In *ICML*, 528–539.

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.-W.; and Choi, J. 2021. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In *CVPR*, 8218–8227.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. *NeurIPS*, 15920–15930.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *ECCV*, 532–547.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366–3385.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Du, M.; Manjunatha, V.; Jain, R.; Deshpande, R.; Dernoncourt, F.; Gu, J.; Sun, T.; and Hu, X. 2021. Towards Interpreting and Mitigating Shortcut Learning Behavior of NLU Models. *arXiv preprint arXiv:2103.06922*.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, 2(11): 665–673.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2019. ImageNet-trained CNNs Are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Heckert, N. A.; Filliben, J. J.; Croarkin, C. M.; Hembree, B.; Guthrie, W. F.; Tobias, P.; Prinz, J.; et al. 2002. Handbook 151: NIST/SEMATECH e-handbook of Statistical Methods. *NIST Interagency/Internal Report (NISTIR)*.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021. Natural Adversarial Examples. In *CVPR*, 15262–15271.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial Examples are Not Bugs, They are Features. In *NeurIPS*.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 448–456. PMLR.

Kim, D.; Park, D.; Shin, Y.; Bang, J.; Song, H.; and Lee, J.-G. 2023. Adaptive Shortcut Debiasing for Online Continual Learning. *arXiv preprint arXiv:2312.08677*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Koh, H.; Kim, D.; Ha, J.-W.; and Choi, J. 2022. Online Continual Learning on Class Incremental Blurry Task Configuration with Anytime Inference. In *ICLR*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning Multiple Layers of Features from Tiny Images. *Technical Report*.

Lee, H.; Kim, H.-E.; and Nam, H. 2019. Srm: A Style-based Recalibration Module for Convolutional Neural Networks. In *ICCV*, 1854–1862.

Lee, S.; Park, C.; Lee, H.; Yi, J.; Lee, J.; and Yoon, S. 2021. Removing Undesirable Feature Contributions Using Out-of-Distribution Data. In *ICLR*.

Park, D.; Song, H.; Kim, M.; and Lee, J.-G. 2021. Task-Agnostic Undesirable Feature Deactivation Using Out-of-Distribution Data. In *NeurIPS*, 4040–4052.

Pezeshki, M.; Kaba, O.; Bengio, Y.; Courville, A. C.; Precup, D.; and Lajoie, G. 2021. Gradient Starvation: A Learning Proclivity in Neural Networks. *NeurIPS*, 1256–1272.

Prabhu, A.; Torr, P. H.; and Dokania, P. K. 2020. Gdumb: A Simple Approach that Questions Our Progress in Continual Learning. In *ECCV*, 524–540.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; and Wayne, G. 2019. Experience Replay for Continual Learning. *NeurIPS*.

Scimeca, L.; Oh, S. J.; Chun, S.; Poli, M.; and Yun, S. 2021. Which Shortcut Cues Will Dnns Choose? A Study from the Parameter-space Perspective. *arXiv preprint arXiv:2110.03095*.

Shah, H.; Tamuly, K.; Raghunathan, A.; Jain, P.; and Netrapalli, P. 2020. The Pitfalls of Simplicity Bias in Neural Networks. *NeurIPS*, 9573–9585.

Shim, D.; Mai, Z.; Jeong, J.; Sanner, S.; Kim, H.; and Jang, J. 2021. Online Class-Incremental Continual Learning with Adversarial Shapley Value. In *AAAI*, 9630–9638.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958.

Valle-Perez, G.; Camargo, C. Q.; and Louis, A. A. 2019. Deep Learning Generalizes Because the Parameter-function Map is Biased towards Simple Functions. In *ICLR*.

Wang, H.; Wu, X.; Huang, Z.; and Xing, E. P. 2020. High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks. In *CVPR*, 8684–8694.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; et al. 2022a. Dualprompt: Complementary Prompting for Rehearsal-free Continual Learning. In *ECCV*, 631–648.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to Prompt for Continual Learning. In *CVPR*, 139–149.

Wei, J.; Wang, Q.; Li, Z.; Wang, S.; Zhou, S. K.; and Cui, S. 2021. Shallow Feature Matters for Weakly Supervised Object Localization. In *CVPR*, 5993–6001.

Xiao, K.; Engstrom, L.; Ilyas, A.; and Madry, A. 2020. Noise or Signal: The Role of Image Backgrounds in Object Recognition. *arXiv preprint arXiv:2006.09994*.

Zhong, G.; Ling, X.; and Wang, L.-N. 2019. From Shallow Feature Learning to Deep Learning: Benefits from the Width and Depth of Deep Architectures. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9: e1255.

Zhou, S.; Zhao, H.; Zhang, S.; Wang, L.; Chang, H.; Wang, Z.; and Zhu, W. 2022. Online Continual Adaptation with Active Self-Training. In *AISTATS*, 8852–8883.