A Whac-A-Mole Dilemma: Shortcuts Come in Multiples Where Mitigating One Amplifies Others

[†]Zhiheng Li[2] [‡]Ivan Evtimov[1] Albert Gordo[1] Caner Hazirbas[1] Tal Hassner[1]
Cristian Canton Ferrer[1] Chenliang Xu[2] [‡]Mark Ibrahim[1]
[1]Meta AI [2]University of Rochester
{ivanevtimov,agordo,hazirbas,thassner,ccanton,marksibrahim}@meta.com
{zhiheng.li,chenliang.xu}@rochester.edu

Abstract

Machine learning models have been found to learn shortcuts—unintended decision rules that are unable to generalize—undermining models' reliability. Previous works address this problem under the tenuous assumption that only a single shortcut exists in the training data. Real-world images are rife with multiple visual cues from background to texture. Key to advancing the reliability of vision systems is understanding whether existing methods can overcome multiple shortcuts or struggle in a Whac-A-Mole game, i.e., where mitigating one shortcut amplifies reliance on others. To address this shortcoming, we propose two benchmarks: 1) UrbanCars, a dataset with precisely controlled spurious cues, and 2) ImageNet-W, an evaluation set based on ImageNet for watermark, a shortcut we discovered affects nearly every modern vision model. Along with texture and background, ImageNet-W allows us to study multiple shortcuts emerging from training on natural images. We find computer vision models, including large foundation models—regardless of training set, architecture, and supervision—struggle when multiple shortcuts are present. Even methods explicitly designed to combat shortcuts struggle in a Whac-A-Mole dilemma. To tackle this challenge, we propose Last Layer Ensemble, a simple-yet-effective method to mitigate multiple shortcuts without Whac-A-Mole behavior. Our results surface multi-shortcut mitigation as an overlooked challenge critical to advancing the reliability of vision systems. The datasets and code are released: `https://github.com/facebookresearch/Whac-A-Mole.git`.

1. Introduction

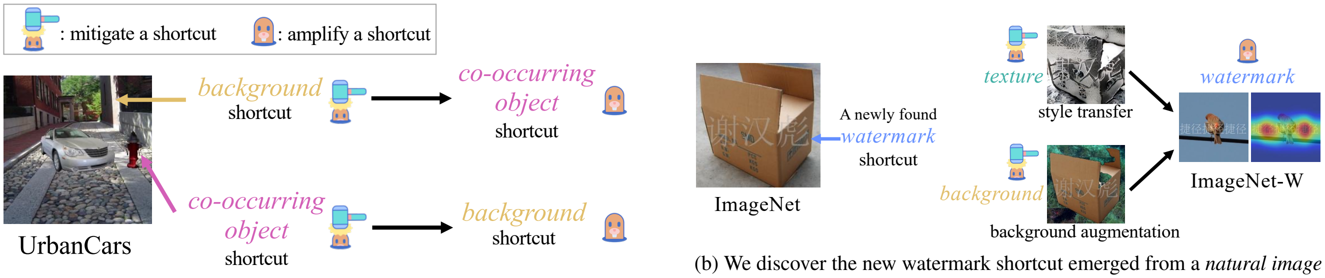
Machine learning often achieves good average performance by exploiting unintended cues in the data [25]. For instance, when backgrounds are spuriously correlated with objects, image classifiers learn background as a rule for object recognition [91]. This phenomenon—called "shortcut learning"—at best suggests average metrics overstate model performance and at worst renders predictions unreliable as models are prone to costly mistakes on out-of-distribution (OOD) data where the shortcut is absent. For example, COVID diagnosis models degraded significantly when spurious visual cues (*e.g.*, hospital tags) were removed [16].

Most existing works design and evaluate methods under the tenuous assumption that a *single shortcut* is present in the data [32,60,72]. For instance, Waterbirds [72], the most widely-used dataset, only benchmarks the mitigation of the background shortcut [7,15,58]. While this is a useful simplified setting, real-world images contain multiple visual cues; models learn multiple shortcuts. From ImageNet [17,80] to facial attribute classification [50] and COVID-19 chest radiographs [16], multiple shortcuts are pervasive. Whether existing methods can overcome multiple shortcuts or struggle in a *Whac-A-Mole* game—where mitigating one shortcut amplifies others—remains a critical open question.

We directly address this limitation by proposing two datasets to study *multi-shortcut* learning: **UrbanCars** and **ImageNet-W**. In UrbanCars (Fig. 1a), we precisely inject two spurious cues—background and co-occurring object. UrbanCars allows us to conduct controlled experiments probing multi-shortcut learning in standard training as well as shortcut mitigation methods, including those requiring shortcut labels. In ImageNet-W (IN-W) (Fig. 1b), we surface a new *watermark* shortcut in the popular ImageNet dataset (IN-1k). By adding a transparent watermark to IN-1k validation set images, ImageNet-W, as a new test set, reveals vision models ranging from ResNet-50 [30] to large foundation models [10] *universally rely on watermark as a spurious cue* for the "carton" class (*cf*. cardboard box in Fig. 1b). When a watermark is added, ImageNet top-1 accuracy drops by 10.7% on average across models. Some, such as ResNet-50, suffer a catastrophic 26.7% drop (from 76.1% on IN-1k to 49.4% on IN-W) (Sec. 2.2)). Along with texture [26,33] and background [91] benchmarks, ImageNet-W allows us to study *multiple shortcuts* emerging in natural images.

We find that across a range of supervised/self-supervised methods, network architectures, foundation models, and shortcut mitigation methods, vision models struggle when multiple shortcuts are present. Benchmarks on UrbanCars and multiple shortcuts in ImageNet (including ImageNet-W)

[†]Work done during the internship at Meta AI. [‡]Equal Contribution.



(a) We construct UrbanCars, a new dataset with multiple shortcuts, facilitating the study of multi-shortcut learning under the *controlled setting*.

(b) We discover the new watermark shortcut emerged from a *natural image* dataset—ImageNet, and create ImageNet-W test set for ImageNet.

Figure 1. Our benchmark results on both datasets reveal the overlooked Whac-A-Mole dilemma in shortcut mitigation, *i.e.*, mitigating one shortcut amplifies the reliance on other shortcuts.

reveal an overlooked challenge in the shortcut learning problem: *multi-shortcut mitigation resembles a Whac-A-Mole game, i.e., mitigating one shortcut amplifies reliance on others*. Even methods specifically designed to combat shortcuts decrease reliance on one shortcut at the expense of amplifying others (Sec. 5). To tackle this new challenge, we propose Last Layer Ensemble (LLE) to mitigate multiple shortcuts jointly without Whac-A-Mole behavior. LLE uses data augmentation based on only the knowledge of the shortcut type without using shortcut labels—making it widely adaptable.

To summarize, our contributions are (1) We create UrbanCars, a dataset with precisely injected spurious cues, to better benchmark multi-shortcut mitigation. (2) We curate ImageNet-W—a new out-of-distribution (OOD) variant of ImageNet benchmarking a pervasive watermark shortcut we discovered— to form a more comprehensive multi-shortcut evaluation suite for ImageNet. (3) Through extensive benchmarks on UrbanCars and ImageNet shortcuts (including ImageNet-W), we uncover that mitigating multiple shortcuts is an overlooked and universal challenge, resembling a Whac-A-Mole game, *i.e.*, mitigating one shortcut amplifies reliance on others. (4) Finally, we propose Last Layer Ensemble for multi-shortcut mitigation without the Whac-A-Mole behavior. We hope our contributions advance research into the overlooked challenge of mitigating multiple shortcuts.

2. New Datasets for Multi-Shortcut Mitigation

While most previous datasets [4,59,60,72] are based on the oversimplified single-shortcut setting, we introduce the UrbanCars dataset (Sec. 2.1) and the ImageNet-Watermark dataset (Sec. 2.2) to benchmark multi-shortcut mitigation.

2.1. UrbanCars Dataset

Overview We construct the UrbanCars dataset with multiple shortcuts: *background* (BG) and *co-occurring object* (CoObj). As shown in Fig. 2, each image in UrbanCars has a car at the center on a natural scene background with a co-occurring object on the right. The task is to classify the car's body type (*i.e.*, target) by overcoming two shortcuts in the training set, which correlate with the target label.

Formally, we denote the dataset as a set of N tuples, $\{(x_i, y_i, b_i, c_i)\}_{i=1}^{N}$, where each image x_i is annotated with three labels: target label y_i for the car body type, *background* label b_i, and *co-occurring object* label c_i. We use a shared label space for all three labels with two classes: `urban` and `country`, *i.e.*, $y_i, b_i, c_i \in \{\texttt{urban}, \texttt{country}\}$. Based on the combination of three labels, the dataset is partitioned into $2^3 = 8$ groups, *i.e.*, $\{\texttt{urban}, \texttt{country}\}$ car on the $\{\texttt{urban}, \texttt{country}\}$ BG with the $\{\texttt{urban}, \texttt{country}\}$ CoObj. We introduce the data distribution and construction below and include details in Appendix A.1.

Data Distribution The training set of UrbanCars has two spurious correlations of BG and CoObj shortcuts, whose strengths are quantified by $P(\mathbf{b} = \mathbf{y} \mid \mathbf{y})$ and $P(\mathbf{c} = \mathbf{y} \mid \mathbf{y})$, respectively. That is, the ratio of common BG (or CoObj) given a target class. We set both to 0.95 by following the correlation strength in [72]. We assume that two shortcuts are independently correlated with the target, *i.e.*, $P(\mathbf{b}, \mathbf{c} \mid \mathbf{y}) = P(\mathbf{b} \mid \mathbf{y})P(\mathbf{c} \mid \mathbf{y})$. As shown in Fig. 2, most urban car images have the urban background (*e.g.*, alley) and urban co-occurring object (*e.g.*, fire plug), and vice versa for country car images. The frequency of each group in the training set is in Fig. 2. The validation and testing sets are balanced without spurious correlations, *i.e.*, ratios are 0.5.

Data Construction The UrbanCars dataset is created from several source datasets. The car objects and labels are from Stanford Cars [49], where the urban cars are formed by classes such as sedan and hatchback. The country cars

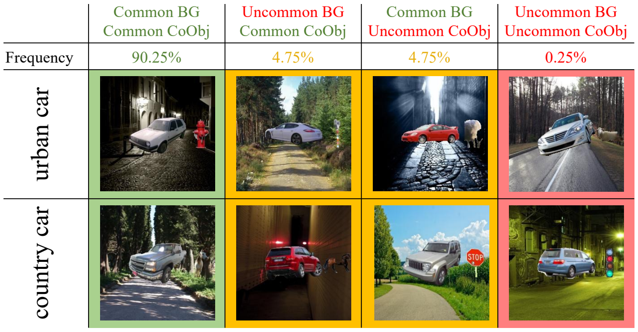


Figure 2. Unbalanced groups in UrbanCars's training set based on two shortcuts: *background* (BG) and *co-occurring object* (CoObj).

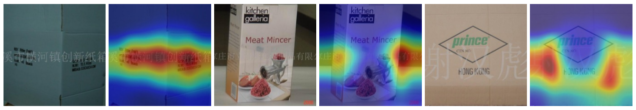


Figure 3. Many carton class images in the ImageNet training set contain the watermark. Saliency maps [76] of ResNet-50 [30] show that the watermark serves as the shortcut for the carton class.

are from classes such as truck and van. The backgrounds are from Places [97]. We use classes such as alley and crosswalk to form the urban background. The country background images are from classes such as forest road and field road. Regarding co-occurring objects, we use LVIS [28] to obtain the urban ones (*e.g*., fireplug and stop sign), and country ones (*e.g*., cow and horse). After obtaining the source images, we paste the car and co-occurring object onto the background.

UrbanCars Metrics We first report the *In Distribution Accuracy* (**I.D. Acc**) on UrbanCars. It computes the weighted average over accuracy per group, where weights are proportional to the training set's correlation strength (*i.e*., frequency in Fig. 2) by following the "average accuracy" [72] to measure the model's performance when no group shift happens.

To measure robustness against the group shift, previous single-shortcut benchmarks [15,58,72] use worst-group accuracy [72], *i.e*., the lowest accuracy among all groups. However, this metric does not capture multi-shortcut mitigation well since it only focuses on groups where both shortcut categories are uncommon (*cf*. the last column in Fig. 2).

To address this shortcoming, we introduce three new metrics: **BG Gap**, **CoObj Gap**, and **BG+CoObj Gap**. BG Gap is the accuracy drop from I.D. Acc to accuracy in groups where BG is uncommon but CoObj is common (*cf*. 1st yellow column in Fig. 2). Similarly, CoObj Gap computes the accuracy drop from I.D. Acc to groups where only CoObj is uncommon (*cf*. 2nd yellow column in Fig. 2). BG+CoObj Gap computes accuracy drop from I.D. Acc to groups where both BG and CoObj are uncommon (*cf*. red column in Fig. 2). The first two metrics measure the robustness against the group shift for each shortcut, and the last metric evaluates the model's robustness when both shortcuts are absent.

2.2. ImageNet-Watermark (ImageNet-W)

In addition to the precisely controlled spurious correlations in UrbanCars, we study naturally occurring shortcuts in the most popular computer vision benchmark: ImageNet [17]. While ImageNet lacks shortcut labels, we can evaluate models' reliance on texture [26] and background [91] shortcuts. We additionally discovered a pervasive watermark shortcut and contribute ImageNet-Watermark (ImageNet-W or IN-W), an evaluation set to expose models' watermark shortcut reliance. Along with texture and background, this forms a comprehensive suite to evaluate reliance on the multiple naturally occurring shortcuts in ImageNet.

Watermark Shortcut in ImageNet In the training set of the *carton* class, many images contain a watermark at the center written in Chinese characters and ImageNet-trained ResNet-50 [30] focuses on the watermark region to predict the carton class (Fig. 3). Since the watermark reads carton factory names or contact person's names of a carton factory, we conjecture that this watermark shortcut originates from the real-world spurious correlation of web images. In the validation set, none of the carton class images contain the watermark, so ResNet-50 underperforms on the carton class (48%) relative to overall accuracy (76%) across 1k classes.

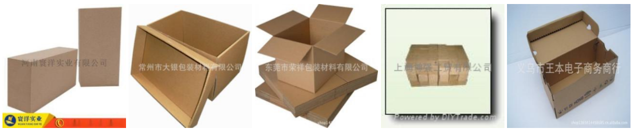


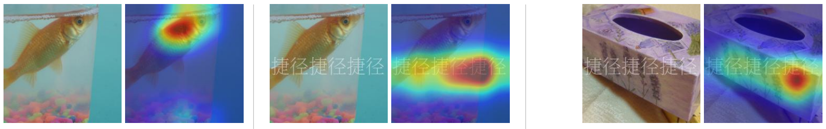
Figure 4. Carton images from LAION [73,74], a large-scale dataset with 400 million to 2 billion images used in CLIP [66] pretraining, also contain watermarks, enabling CLIP's reliance on the watermark shortcut in zero-shot transfer to ImageNet and ImageNet-W.

Data Construction To test the robustness against the watermark shortcut, we create ImageNet-Watermark (ImageNet-W or IN-W) dataset, a new out-of-distribution evaluation set of ImageNet. As shown in Tab. 1, we overlay a transparent watermark written in "捷径捷径捷径" at the center of all images from ImageNet validation set to mimic the watermark pattern in IN-1k, where "捷径" means "shortcut" in Chinese. We do this because we find that models use the watermark even when the content is not identical to the watermark in the training set of carton images, suggesting that it is watermark's presence rather than its content that serves as the shortcut. We evaluate watermark in other contents and languages in Appendix A.2.

ImageNet-W Metrics We mainly use two metrics to measure watermark shortcut reliance: (1) **IN-W Gap** is the accuracy on IN-W minus the accuracy on IN-1k validation set. A smaller accuracy drop indicates less reliance on the watermark shortcut across all 1000 classes. (2) **Carton Gap** is the carton class accuracy increase from IN-1k to IN-W. A smaller Carton Gap indicates less reliance on the watermark shortcut for predicting the carton class.

To demonstrate that the watermark shortcut is used for predicting carton, we also use the following in Tab. 1: (1) $P(\hat{y} = \text{carton})$, the likelihood of predicting carton on all IN-1k validation set images, (2) $\Delta P(\hat{y} = \text{carton})$, the likelihood increase from IN-1k to IN-W on all 1k classes images, and (3) $\Delta P(\hat{y} = \text{carton} \mid y = \text{carton})$, the likelihood increase from IN-1k to IN-W on carton class images.

Ubiquitous reliance on the watermark shortcut To study reliance on the watermark shortcut, we use ImageNet-W to benchmark a broad range of State-of-The-Art (SoTA) vision models, including standard supervised training, using different architectures [21,30,67], augmentations and regularizations [26,35,92,93]. We also benchmark foundation models [10] pretrained on larger datasets [27,66,73,74,79]



method	architecture	(pre)training data	IN-1k Acc ↑	$P(\hat{y} = \text{carton})$ (%)	IN-W Gap ↑	$\Delta P(\hat{y} = \text{carton})$ (%) ↓	Carton Gap ↓	$\Delta P(\hat{y} = \text{carton} \mid y = \text{carton})$ (%) ↓
Supervised	ResNet-50 [30]	IN-1k [17]	76.1	0.07	-26.7	+7.56	+40	+42.46
MoCov3 [13] (LP)	ResNet-50	IN-1k	74.6	0.08	-20.7	+2.94	+44	+44.37
Style Transfer [26]	ResNet-50	SIN [26]	60.1	0.10	-17.3	+4.91	+52	+50.06
Mixup [93]	ResNet-50	IN-1k	76.1	0.07	-18.6	+3.43	+38	+39.78
CutMix [92]	ResNet-50	IN-1k	78.5	0.09	-14.8	+1.92	+22	+29.61
Cutout [19,96]	ResNet-50	IN-1k	77.0	0.08	-18.0	+2.93	+32	+38.06
AugMix [35]	ResNet-50	IN-1k	77.5	0.09	-16.8	+2.61	+36	+34.44
Supervised	RG-32gf	IN-1k	80.8	0.09	-14.1	+3.74	+32	+33.43
SEER [27] (FT)	RG-32gf [67]	IG-1B [27]	83.3	0.09	-6.5	+0.56	+18	+24.26
Supervised	ViT-B/32 [21]	IN-1k	75.9	0.09	-8.7	+1.20	+34	+34.31
Uniform Soup [89] (FT)	ViT-B/32	WIT [66]	79.9	0.09	-7.9	+0.32	+24	+23.87
Greedy Soup [89] (FT)	ViT-B/32	WIT	81.0	0.09	-6.5	+0.35	+16	+23.87
Supervised	ViT-L/16	IN-1k	79.6	0.08	-6.2	+0.82	+34	+32.57
CLIP [66] (zero-shot)	ViT-L/14	WIT	76.5	0.06	-4.4	**+0.01**	+12	**+1.75**
CLIP (zero-shot)	ViT-L/14	LAION-400M [74]	72.7	0.05	-4.9	+0.03	+12	+13.76
MAE [29] (FT)	ViT-H/14	IN-1k	86.9	0.08	-3.5	+0.43	+30	+29.59
SWAG [79] (LP)	ViT-H/14	IG-3.6B [79]	85.7	0.09	-4.9	+0.19	**+8**	+12.80
SWAG (FT)	ViT-H/14	IG-3.6B	88.5	0.09	**-3.1**	+0.35	+18	+20.25
CLIP (zero-shot)	ViT-H/14	LAION-2B [73]	77.9	0.06	-3.6	+0.03	+16	+12.01
	average		78.6	0.08	-10.7	+1.74	+26.7	+27.96

Table 1. **Models rely on the watermark as a shortcut for the carton class.** LP and FT denote linear probing and fine-tuning on ImageNet-1k, respectively. Because models exhibit drops (*i.e.*, IN-W Gap) and an increase in accuracy and likelihood on the carton class from IN-1k to IN-W, we conclude that various vision models suffer from the watermark shortcut (more results in Appendices D.1 and D.2).

with different pretraining supervision and transfer learning techniques [13,27,29,66,79,89]. In Tab. 1, we find a considerable IN-W Gap of up to -26.7 and -10.7 on average and a Carton Gap of up to +52 and +26.7 on average. While all models exhibit uniform ($1/1000 = 0.1\%$) likelihood for carton class ($P(\hat{y} = \text{carton})$) on IN-1k, we observe a considerable increase in the likelihood of predicting carton on IN-W ($\Delta P(\hat{y} = \text{carton})$) and a significant likelihood increase in carton class images ($\Delta P(\hat{y} = \text{carton} \mid y = \text{carton})$). Although compared to supervised ResNet-50, some models with larger architectures or extra training data can decrease reliance on the watermark shortcut, none of them fully close the performance gaps. Interestingly, CLIP with zero-shot transfer still suffers from the watermark shortcut with +12 to +16 Carton Gap, which could be explained by many carton images in the pretraining data (*e.g.*, LAION) also containing watermarks (*cf*. Fig. 4). This suggests that even large-scale pretraining datasets (*e.g.*, used for CLIP) can contain shortcuts, confirming findings that it is data's quality, not quantity [24,61], that matters most to CLIP's robustness.

Multi-Shortcut Mitigation Metrics on ImageNet To measure the mitigation of multiple shortcuts, we evaluate models on multiple OOD variants of ImageNet. In this work, we study three shortcuts on ImageNet—background, texture, and watermark. The background shortcut is evaluated on ImageNet-9 (IN-9) [91], and we use **IN-9 Gap** (*i.e.*, BG-Gap in [91]) as the evaluation metric, which is the accuracy drop from Mixed-Same to Mixed-Rand in IN-9, where a lower accuracy drop implies less background shortcut reliance. The texture shortcut is evaluated on Stylized ImageNet (SIN) [26] and ImageNet-R (IN-R) [33], where we use **SIN Gap**, top-1 accuracy drop from IN-1k to SIN, and **IN-R Gap**, the top-1 accuracy drop from IN-200 (*i.e.*, a subset of IN-1k with 200 classes used in IN-R) to IN-R.

3. Benchmark Methods and Settings

On all datasets, we first evaluate standard training that minimizes the empirical risk on the training set (*i.e.*, **ERM** [83]) using ResNet-50 [30] as the network architecture, which serves as the baseline. On ImageNet, we additionally show ERM's results with other architectures, pretraining datasets, and supervision.

In addition to ERM, we comprehensively evaluate shortcut mitigation methods across four categories based on the level of shortcut information required (Tab. 2).

Category 1: Standard Augmentation and Regularization Methods in this category use general data augmentation or regularization without prior knowledge of the shortcut, which are commonly used to improve accuracy on IN-1k,

Category	Summary	Shortcut Information	Methods
1	Standard Augmentation and Regularization	None	Mixup [93], Cutout [19,96], CutMix [92], AugMix [35], SD [63]
2	Targeted Augmentation for Mitigating Shortcuts	Types of shortcuts (w/o shortcut labels)	CF+F Aug [11], Style Transfer (TXT Aug) [26], BG Aug [71,91], WMK Aug
3	Using Shortcut Labels	Image-level ground-truth shortcut label	gDRO [72], DI [87], SUBG [38], DFR [45]
4	Inferring Pseudo Shortcut Labels	Image-level pseudo shortcut label	LfF [60], JTT [58], EIIL [15], DebiAN [53]

Table 2. Existing methods for multi-shortcut mitigation benchmark.

e.g., new training recipes [84,88]. Some works [11,64] show that they can also improve OOD robustness.

Category 2: Targeted Augmentation for Mitigating Shortcuts Other works use data augmentation that modifies shortcut cues. We benchmark CF+F Aug [11] on UrbanCars. On ImageNet, we benchmark texture augmentation (TXT Aug) via Style Transfer [26] and background augmentation (BG Aug) [71,91]. To counter the watermark shortcut, we design watermark augmentation (WTM Aug) that randomly overlays the watermark onto images (*cf*. Appendix B.1).

Category 3: Using Shortcut Labels In this category, methods use shortcut labels for mitigation, which are generally used to reweight [72] or resample training data [38, 45,72]. We only benchmark methods in this category on UrbanCars since ImageNet does not have shortcut labels.

Category 4: Inferring Pseudo Shortcut Labels Following the ideas of methods using shortcut labels, one line of works [15,53,58,60] estimates the pseudo shortcut labels when ground-truth labels are unavailable.

Benchmark Settings We introduce the experiment settings here (details in Appendix B.3). On UrbanCars, we use worst-group accuracy [72] on the validation set to select the early stopping epoch and report test set results. All methods except DFR [45] use end-to-end training on UrbanCars. On ImageNet, following the last layer re-training [45] setting, we only train the last classification layer upon a frozen feature extractor. On both datasets, we use ResNet-50 as the network architecture. On ImageNet, we also benchmark self-supervised and foundation models.

4. Our Approach

Motivation Our multi-shortcut benchmark results (Sec. 5) show that many existing methods suffer from the Whac-A-Mole problem, motivating us to design a better method to mitigate multiple shortcuts simultaneously.

We focus on mitigating multiple *known* shortcuts, *i.e.*, the number and types of shortcuts are given, but shortcut labels are not. Although mitigating unknown numbers and types of shortcuts (*i.e.*, category 1, 4) sounds more desirable, not only do our empirical results show their under-performance but also a concurrent work [57] proves the theoretical impossibility of mitigation without any inductive biases.

We follow methods that use data augmentation to modify the shortcut cues (*i.e.*, category 2). Formally, given a set of K shortcuts $\{\mathrm{s}_i\}_{i=1}^K$ for mitigation, we create a set of augmentations $\mathrm{S}_{\text{aug}} = \{\mathcal{A}_i\}_{i=1}^K \cup \{\mathcal{I}\}$, where the augmentation $\mathcal{A}_i$ (*e.g.*, style transfer [26]) modifies the visual cue of the shortcut s_i (*e.g.*, texture). $\mathcal{I}$ denotes the identity transformation, *i.e.*, no augmentation applied.

Based on the augmentation set S_{aug}, a straightforward way is to minimize the empirical risk [83] over all augmented and original images. However, different augmentations can be incompatible, leading to suboptimal results. That is, augmentation $\mathcal{A}_i$ could be detrimental to mitigating a different shortcut s_j, where $i \neq j$. For example, mitigating the texture shortcut via style transfer [26] augmentation unexpectedly amplifies the saliency of the watermark (Fig. 1b), leading to worse watermark mitigation results (Tab. 1).

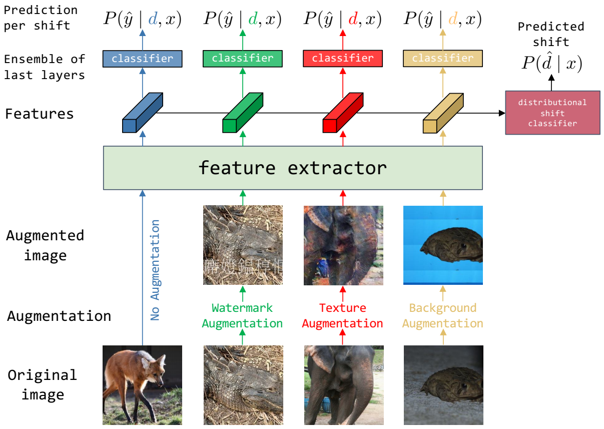


Figure 5. An overview of Last Layer Ensemble (LLE). LLE trains an ensemble of the last classification layers upon a feature extractor, where each last layer is trained with images in one augmentation type. The distributional shift classifier, supervised by the augmentation type, is trained to predict the distributional shift and dynamically aggregates the predictions per shift during testing.

Last Layer Ensemble To address this issue, we propose Last Layer Ensemble (LLE), a new method for mitigating multiple shortcuts simultaneously (Fig. 5). Since it is hard to use a single model to learn the invariance among incompatible augmentations, we instead train an ensemble [20] of classification layers (*i.e.*, last layers) on top of a shared feature extractor so that each classification layer only trains on data from a single type of augmentation that simulates one type of distributional shift d. In this way, each last layer predicts the likelihood of the target $P(\hat{y} \mid d, x)$.

At the same time, we train a *distributional shift classifier*, another classification layer on top of the feature extractor, to predict the type of augmentation that simulates the distributional shift, *i.e.*, $P(\hat{d} \mid x)$. During testing, LLE dynamically aggregates the logits from the ensemble of the last layers based on the predicted distributional shift. *E.g.*, when the testing image contains the texture shift, the *distributional shift classifier* gives higher weights for the logits from the classifier trained with texture augmentation, alleviating the impact from other classification layers trained with incompatible augmentations. In addition, when the weights of the feature extractor are not frozen, we stop the gradient from the *distributional shift classifier* to the feature extractor, preventing the feature extractor from learning the shortcut information. Compared to standard ensemble approaches [20] that train multiple full networks and add significant inference overhead, our method uses minimal additional training parameters and has better computational efficiency.

		shortcut reliance		
	I.D. Acc	BG Gap ↑	CoObj Gap ↑	BG+CoObj Gap ↑
ERM	97.6	-15.3	-11.2	-69.2
Mixup	98.3	-12.6	-9.3	-61.8
CutMix	96.6	-45.0 (×2.94 🐹)	-4.8	-86.5
Cutout	97.8	-15.8 (×1.03 🐹)	-10.4	-71.4
AugMix	98.2	-10.3	-12.1 (×1.08 🐹)	-70.2
SD	97.3	-15.0	-3.6	-36.1
CF+F Aug	96.8	-16.0 (×1.04 🐹)	**+0.4**	-19.4
LfF	97.2	-11.6	-18.4 (×1.64 🐹)	-63.2
JTT (E=1)	95.9	-8.1	-13.3 (×1.18 🐹)	-40.1
EIIL (E=1)	95.5	-4.2	-24.7 (×2.21 🐹)	-44.9
JTT (E=2)	94.6	-23.3 (×1.52 🐹)	-5.3	-52.1
EIIL (E=2)	95.5	-21.5 (×1.40 🐹)	-6.8	-49.6
DebiAN	98.0	-14.9	-10.5	-69.0
LLE (ours)	96.7	**-2.1**	-2.7	**-5.9**

Table 3. **Many methods not using shortcut labels (category 1,2,4) amplify shortcut on UrbanCars.** 🐹: increased reliance on a shortcut relative to ERM. ×2.94: 2.94 times larger than ERM.

5. Experiments

Based on UrbanCars and ImageNet-W datasets, we show results on multi-shortcut mitigation. We first study if standard supervised training (*i.e.*, ERM) relies on multiple shortcuts (Sec. 5.1). Next, we show the multi-shortcut setting is significantly challenging: mitigating one shortcut increases reliance on other shortcuts compared to ERM. We name this phenomenon *Whac-A-Mole*, which is observed in many SoTA methods, including mitigation methods (Sec. 5.2) and self-supervised/foundation models (Sec. 5.3). Finally, we show that our Last Layer Ensemble method can reduce reliance across multiple shortcuts more effectively (Sec. 5.4).

5.1. Standard training relies on multiple shortcuts

On both datasets, we find that standard training (*i.e.*, ERM [83]) relies on multiple shortcuts. On UrbanCars, Tab. 3 shows that ERM achieves near zero in-distributional error (97.6% I.D. Acc.). However, ERM's performance drops when group shift happens. When the background shortcut is absent, ERM's performance drops by 15.3% in BG Gap. Similarly, the accuracy drops by 11.2% in CoObj Gap when the CoObj shortcut is absent. When neither shortcut is present, models suffer catastrophic drops of 69.2% in BG+CoObj Gap. On ImageNet, Tab. 4 shows that ERM achieves good top-1 accuracy of 76.39% on IN-1k. However, it suffers considerable drops in accuracy when watermark, texture, or background cues are altered, *e.g.*, 30% Carton Gap for watermark, 56-69% for texture, and 5.19% for background, suggesting that standard training on natural images from ImageNet leads to reliance on multiple shortcuts.

5.2. Results: Mitigation Methods

Results: Standard Augmentation and Regularization (Category 1) We first show the results of methods using augmentation and regularization without using inductive biases of shortcuts. On UrbanCars (Tab. 3), we observed that CutMix and Cutout amplify the background shortcut with a larger BG Gap relative to ERM. AugMix increases the reliance on the CoObj shortcut with a larger CoObj Gap (*i.e.*, -12.2%) compared to ERM. Although Mixup and SD do not produce Whac-A-Mole results, they only yield marginal improvement or can only mitigate one shortcut well. On ImageNet, the results in Tab. 4 show that all approaches amplify at least one shortcut. For instance, AugMix achieves a worse Carton Gap to amplify the watermark shortcut compared to ERM. For CutMix, we again observe that it amplifies the BG shortcut on ImageNet. We show more results of CutMix's background reliance in Appendix F.

		shortcut reliance				
		Watermark (WTM)		Texture (TXT)		Background (BG)
	IN-1k	IN-W Gap ↑	Carton Gap ↓	SIN Gap ↑	IN-R Gap ↑	IN-9 Gap ↑
ERM	76.39	-25.40	+30	-69.43	-56.22	-5.19
Mixup	76.17	-24.87	+34 (×1.13 🐹)	-68.18	-55.79	-5.60 (×1.08 🐹)
CutMix	75.90	-25.78 (×1.01 🐹)	+32 (×1.06 🐹)	-69.31	-56.36	-5.65 (×1.09 🐹)
Cutout	76.40	-25.11	+32 (×1.06 🐹)	-69.39	-55.93	-5.35 (×1.03 🐹)
AugMix	76.23	-23.41	+38 (×1.26 🐹)	-68.51	-54.91	-5.85 (×1.13 🐹)
SD	76.39	-26.03 (×1.02 🐹)	+30	-69.42	-56.36	-5.33 (×1.03 🐹)
WTM Aug	76.32	**-5.78**	+14	-69.31	-56.22	-5.34 (×1.03 🐹)
TXT 🔨 Aug	75.94	-25.93 (×1.02 🐹)	+36 (×1.20 🐹)	-63.99	**-53.24**	-5.66 (×1.09 🐹)
BG 🔨 Aug	76.03	-25.01	+36 (×1.20 🐹)	-68.41	-54.51	-4.67
LfF	76.35	-26.19 (×1.03 🐹)	+36 (×1.20 🐹)	-69.34	-56.02	-5.61 (×1.08 🐹)
JTT	76.33	-26.40 (×1.04 🐹)	+32 (×1.06 🐹)	-69.48	-56.30	-5.55 (×1.07 🐹)
EIIL	71.55	-33.48 (×1.31 🐹)	+24	-66.04	-61.35 (×1.09 🐹)	-6.42 (×1.24 🐹)
DebiAN	76.33	-26.40 (×1.04 🐹)	+36 (×1.20 🐹)	-69.37	-56.29	-5.53 (×1.07 🐹)
LLE (ours)	76.25	-6.18	**+10**	**-61.00**	-54.89	**-3.82**

Table 4. **Existing methods fail to combat multiple shortcuts by amplifying at least one shortcut relative to ERM on ImageNet.** All models use ResNet-50 with last layer re-training [45].

> **Takeaway**: Standard augmentation and regularization methods can mitigate some shortcuts (*e.g.*, texture) 🔨 but amplify others 🐹.

Results: Targeted Augmentation for Mitigating Shortcuts (Category 2) Further, we benchmark methods using data augmentation to mitigate a specific shortcut. Compared to methods in category 1, augmentations here use stronger inductive biases about the shortcut by modifying the shortcut visual cue. On UrbanCars, although CF+F Aug achieves good results for the CoObj shortcut, it amplifies the BG shortcut. On ImageNet, texture and background augmentation improve the reliance on the watermark shortcut, which can be explained by the remained or even increased saliency of the watermark in Fig. 1b and Appendix D.3.

> **Takeaway**: Augmentations tackling a specific type of shortcut 🔨 (*e.g.*, style transfer for texture shortcut) can amplify other shortcuts 🐹 (*e.g.*, watermark).

Results: Using Shortcut Labels (Category 3) Then, we show the results of methods using shortcut labels on UrbanCars in Tab. 5. Methods can mitigate multiple shortcuts when labels of both shortcuts are used (*cf*. first section in Tab. 5). However, when using labels of either shortcut, which

	shortcut label			shortcut reliance		
	Train	Val	I.D. Acc	BG Gap ↑	CoObj Gap ↑	BG+CoObj Gap ↑
ERM	✗	BG+CoObj	97.6	-15.3	-11.2	-69.2
gDRO	BG+CoObj	BG+CoObj	91.6	-10.9	-3.6	-16.4
DI	BG+CoObj	BG+CoObj	89.0	**-2.2**	-1.0	**+0.4**
SUBG	BG+CoObj	BG+CoObj	71.1	-4.7	**-0.3**	-6.3
DFR	BG+CoObj	BG+CoObj	89.7	-10.7	-6.9	-45.2
ERM	✗	BG	97.8	-14.6	-11.3	-68.5
gDRO	BG 🔨	BG	96.0	-4.2	-26.9 (×2.39 🐹)	-56.5
DI	BG 🔨	BG	94.7	+2.2	-27.0 (×2.40 🐹)	-25.2
SUBG	BG 🔨	BG	92.6	+1.3	-36.4 (×3.24 🐹)	-35.8
DFR	BG 🔨	BG	97.4	-9.8	-13.6 (×1.21 🐹)	-58.9
ERM	✗	CoObj	97.6	-15.4	-11.0	-68.8
gDRO	CoObj 🔨	CoObj	95.7	-31.4 (×2.03 🐹)	-0.5	-54.9
DI	CoObj 🔨	CoObj	94.2	-36.1 (×2.34 🐹)	+2.8	-35.8
SUBG	CoObj 🔨	CoObj	93.1	-60.2 (×3.90 🐹)	+2.5	-62.4
DFR	CoObj 🔨	CoObj	97.4	-19.1 (×1.24 🐹)	-8.6	-64.9

Table 5. **Methods using shortcut labels (category 3) amplify the unlabeled shortcut when mitigating the labeled shortcut on UrbanCars.** 🔨: mitigate a shortcut, *e.g.*, using shortcut labels.

is the typical situation for in-the-wild datasets where shortcut labels are incomplete, they exhibit a higher performance gap in the other shortcut relative to ERM. *E.g.*, when only using the CoObj labels, models achieve poorer BG Gap results.

> **Takeaway**: Methods using shortcut labels mitigate the labeled shortcut 🔨 but amplifies the unlabeled one 🐹.

Results: Inferring Pseudo Shortcut Labels (Category 4) The Whac-A-Mole problem of methods using shortcut labels motivates us to study whether the problem can be solved by inferring pseudo labels of multiple shortcuts. Here we analyze the results of LfF, JTT, EIIL, and DebiAN. Their key idea is based on ERM's training dynamics of learning different visual cues. LfF infers soft shortcut labels by assuming that the shortcut is learned earlier. Similarly, JTT and EIIL use an under-trained ERM trained with E epochs as the reference model to infer pseudo shortcut labels. We use E=1 and E=2 for JTT and EIIL. Instead of using a fixed reference model, DebiAN jointly trains the reference and mitigation models. The results in Tab. 3 show that LfF, JTT (E=1), and EIIL (E=1) still exhibit Whac-A-Mole results by achieving a larger CoObj Gap than ERM. On the other hand, JTT (E=2) and EIIL (E=2) also show the Whac-A-Mole results by achieving larger BG Gap than ERM. On ImageNet, we observe Whac-A-Mole results produced by LfF, JTT, EIIL, and DebiAN in Tab. 4.

To investigate the reason for their Whac-A-Mole results, we analyze the training dynamics of ERM. In Fig. 6, we plot the accuracy of three visual cues—object (*i.e.*, car body type), background, and co-occurring object on the validation set. The accuracy is computed based on ERM's {`urban`, `country`} predictions against labels of object, BG, and CoObj. We observe a Whac-A-Mole game in ERM's training. At epoch 1, ERM mainly predicts the background (82.6%), suggesting that the background shortcut is learned first. Thus, LfF, JTT (E=1), and EIIL (E=1) can infer the BG shortcut labels well to amplify the CoObj shortcut. As the training continues to epoch 2, the reliance on the BG shortcut decreases (82.6% to 71.2%), but the reliance on the CoObj shortcut is increased (60.6% to 71.8%). It renders JTT (E=2) and EIIL (E=2) better infer CoObj shortcut labels, which, in turn, amplifies the BG shortcut.

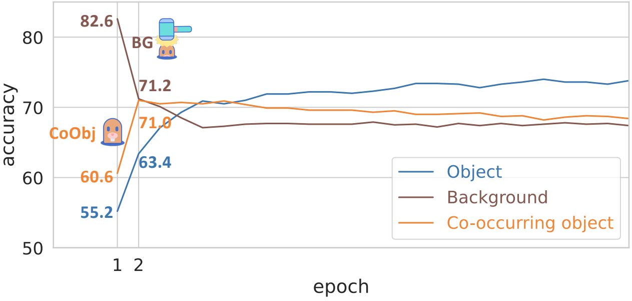


Figure 6. On UrbanCars, **ERM learns BG and CoObj shortcuts at different training epochs, making it difficult to infer pseudo labels (category 4) of multiple shortcuts from ERM.**

> **Takeaway**: Methods inferring pseudo shortcut labels still amplify shortcuts 🐹 because ERM learns different shortcuts asynchronously during training, making it hard to infer labels of all shortcuts 🔨 for mitigation.

5.3. Results: Self-Supervised & Foundation Models

On ImageNet, we further benchmark self-supervised pretraining methods, *i.e.*, MoCov3 [13], MAE [29], SEER [27]. We also benchmark foundation models that use extra training data, *i.e.*, Uniform Soup [89], Greedy Soup [89], CLIP [66], SEER [27], and SWAG [79]. The results in Tab. 6 show that many of them fail to mitigate multiple shortcuts jointly. Regarding self-supervised methods, MoCov3 achieves worse results on all three shortcuts, and MAE achieves a worse SIN Gap for the texture shortcut relative to ERM. Regarding foundation models, although SWAG with linear probing (LP) achieves a much better IN-R Gap (-19.79%), it also has a stronger reliance on the background in BG Gap compared to ERM. Similarly, SEER, Uniform Soup, and Greedy Soup mitigate the watermark shortcut but amplify the background shortcut. When using ViT-L, although CLIP with zero-shot transfer does not produce Whac-A-Mole results, they do not fully close the performance gap. Besides, they also show much lower IN-1k accuracy than other foundation models. We show results using other architectures in Appendix E.2.

> **Takeaway**: Self-supervised and foundation models can mitigate some shortcuts 🔨 but amplify others 🐹.

5.4. Results: Last Layer Ensemble (LLE)

We show that our Last Layer Ensemble (LLE) can better tackle multi-shortcut mitigation. LLE mitigates shortcuts via a set of data augmentations. Specifically, we augment background (BG) and co-occurring object (CoObj) by swapping

		shortcut reliance				
		Watermark		Texture		Background
	IN-1k	IN-W Gap ↑	Carton Gap ↓	SIN Gap ↑	IN-R Gap ↑	IN-9 Gap ↑
arch: RG-32gf						
ERM	80.88	-14.15	+32	**-69.27**	-52.43	**-6.40**
SEER (FT,IG-1B)	83.35	**-6.50**	**+18**	-73.04 (×1.05)	**-50.42**	-7.14 (×1.11)
arch: ViT-B/32						
ERM	75.92	-8.71	+34	**-57.16**	-49.45	**-6.86**
Uniform Soup (FT,WIT)	79.96	-7.90	+24	-59.67 (×1.04)	**-27.51**	-7.78 (×1.13)
Greedy Soup (FT,WIT)	81.01	**-6.47**	**+16**	-59.61 (×1.04)	-30.01	-7.21 (×1.05)
arch: ViT-B/16						
ERM	81.07	-6.69	+26	-62.60	-50.36	-5.36
SWAG (LP,IG-3.6B)	81.89	-7.76 (×1.16)	+18	-67.33 (×1.08)	**-19.79**	-10.39 (×1.94)
SWAG (FT,IG-3.6B)	85.29	-5.43	+24	-66.99 (×1.07)	-29.55	-4.44
MoCov3 (LP)	76.65	-16.0 (×2.39)	+22	-63.36 (×1.01)	-56.86 (×1.12)	-7.80 (×1.45)
MAE (FT)	83.72	-4.60	+24	-65.20 (×1.04)	-47.10	-4.45
MAE+**LLE (ours)**	83.68	**-2.48**	**+6**	**-58.78**	-44.96	**-3.70**
arch: ViT-L/16 or 14						
ERM	79.65	-6.14	+34	-61.43	-53.17	-6.50
SWAG (LP,IG-3.6B)	85.13	-5.73	**+6**	-60.26	-10.17	-7.26 (×1.12)
SWAG (FT,IG-3.6B)	88.07	-3.16	+20	-63.45 (×1.03)	-12.29	-2.92
CLIP (zero-shot,WIT)	76.57	-4.47	+12	-61.27	**-6.26**	-3.68
CLIP (zero-shot,LAION)	72.77	-4.94	+12	-56.85	-8.43	-4.54
MAE (FT)	85.95	-4.36	+22	-62.48 (×1.02)	-36.46	-3.53
MAE+**LLE (ours)**	85.84	**-1.74**	+12	**-56.32**	-34.64	**-2.77**

Table 6. On ImageNet, **many self-supervised and foundation models amplify shortcuts**, whereas **LLE mitigates multiple shortcuts jointly.** (·): transfer learning (and extra data).

BG and CoObj across target classes on UrbanCars (details in Appendix B.5). On ImageNet, we use watermark augmentation (WMK Aug), style transfer [26] (TXT Aug), and background augmentation [71,91] (BG Aug) for watermark, texture, and background shortcuts, respectively. Besides, we show the results of LLE on other OOD variants of ImageNet in Appendices E.3 and E.5.

The results on UrbanCars in Tab. 3 show that LLE beats all other methods in BG Gap and BG+CoObj Gap metrics and achieves second best CoObj Gap to CF+F Aug, a method amplifies the background shortcut. The results of ImageNet with ResNet-50 are in Tab. 4. LLE achieves the best multi-shortcut mitigation results in Carton Gap, SIN Gap, and IN-9 Gap. Regarding IN-W Gap and IN-R Gap, LLE achieves better results than ERM. *I.e.*, no Whac-A-Mole problems. On ImageNet, we further use MAE as the feature extractor, and the results on ImageNet are in Tab. 6. LLE achieves the best results in IN-W Gap, SIN Gap, and IN-9 Gap. LLE also achieves the best results in the remaining metrics comparing to methods not using extra pretraining data.

Ablation Study In Tab. 7, we show the ablation study of LLE: (1) w/o ensemble: training a single last layer. (2) AugMix+w/o ensemble: based on (1) and use JS divergence in AugMix to improve the invariance across augmentations. (3) w/o dist cls.: remove *domain shift classifier* and directly take the mean over the output of ensemble classifiers. Except for IN-R Gap, the full model achieves better results in all other metrics. Although the w/o ensemble achieves a better IN-R Gap, it suffers from reliance on other shortcuts.

6. Related Work

Group Shift Datasets Most previous works use single-shortcut datasets [4,32,43,47,55,59,60,72] to benchmark group shift robustness [72]. Although [8,77,95] use labels of multiple attributes [59] for evaluation, there lacks a sanity check on whether the selected attributes are learned as spurious shortcuts. [53,78] create MNIST-based [52] synthetic datasets with multiple shortcuts, where the shortcuts are unrealistic. In contrast, our UrbanCars dataset is more photo-realistic and contains commonly seen shortcuts. Besides, our ImageNet-W dataset better evaluates shortcut mitigation on the large-scale and real-world ImageNet dataset.

		Shortcut Reliance				
		Watermark		Texture		Background
	IN-1k	IN-W Gap ↑	Carton Gap ↓	SIN Gap ↑	IN-R Gap ↑	IN-9 Gap ↑
w/o ensemble	76.03	-6.71	+18	-66.81	**-52.55**	-5.08
AugMix+w/o ensemble	75.17	-7.27	+22	-66.33	-56.38	-5.38
w/o dist. cls.	75.82	-17.77	+36	-66.45	-53.58	-4.81
LLE (full model)	**76.25**	**-6.18**	**+10**	**-61.20**	-54.89	**-3.82**

Table 7. Ablation study of Last Layer Ensemble on ImageNet.

OOD Datasets of ImageNet While many models achieve great performance on ImageNet [17], they suffer under various distributional shifts, *e.g.*, corruption [34], sketches [85], rendition [33], texture [26], background [91], or unknown distributional shifts [36,68]. In this work, we construct ImageNet-W, where SoTA vision models rely on our newly discovered watermark shortcut.

Shortcut Mitigation and Improving OOD Robustness To address the shortcut learning problem [25], [38,72,87] use shortcut labels for mitigation. With only knowledge of the shortcut type, [5,86] use architectural inductive biases. [26,71,91] use augmentation and [41,45] re-trains the last layer for mitigation. Without knowledge of shortcut types, [3,15,53,58,60,77,82,94] infer pseudo shortcut labels, which is theoretically impossible [57], and we show that they struggle to mitigate multiple shortcuts. Other works suggest that self-supervised pretraining [29,44] and foundation models [10,27,27,40,66,89,90] improve OOD robustness. We show that many of them suffer from the Whac-A-Mole problem or struggle to close performance gaps.

7. Conclusion

In this work, we propose novel benchmarks to evaluate multi-shortcut mitigation. The results show that state-of-the-art models fail to mitigate multiple shortcuts in a Whac-A-Mole game. We show that this challenge pervades modern vision models, ranging from shortcut mitigation methods to foundation models. To tackle this challenge, we propose Last Layer Ensemble (LLE) to mitigate multiple shortcuts simultaneously with knowledge of the shortcut types. We leave to future work the setting with no knowledge of shortcut types (see Appendix G). We hope our work inspires future research into the overlooked challenge of multi-shortcut mitigation.

Acknowledgment This work has been partially supported by the National Science Foundation (NSF) under Grant 1909912 and by the Center of Excellence in Data Science, an Empire State Development-designated Center of Excellence. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

References

[1] Whack A Mole image is obtained from Flaticon.com.

[2] Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating Example Difficulty Using Variance of Gradients. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 26

[3] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2021. 8

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization, 2020. 2, 8

[5] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning De-biased Representations with Biased Representations. In *International Conference on Machine Learning*, 2020. 8

[6] Yujia Bao and Regina Barzilay. Learning to Split for Automatic Bias Detection. *arXiv:2204.13749 [cs]*, 2022. 26

[7] Yujia Bao, Shiyu Chang, and Dr Regina Barzilay. Learning Stable Classifiers by Transferring Unstable Features. In *International Conference on Machine Learning*, 2022. 1

[8] Yujia Bao, Shiyu Chang, and Regina Barzilay. Predict then Interpolate: A Simple Algorithm to Learn Stable Classifiers. In *International Conference on Machine Learning*, 2021. 8

[9] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, 2019. 24, 26

[10] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models, 2022. 1, 3, 8

[11] Chun-Hao Chang, George Alexandru Adam, and Anna Goldenberg. Towards Robust Classification Model by Counterfactual and Invariant Data Generation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4, 5, 13, 17

[12] Hila Chefer, Idan Schwartz, and Lior Wolf. Optimizing Relevance Maps of Vision Transformers Improves Robustness. *Advances in Neural Information Processing Systems*, 2022. 19, 22

[13] Xinlei Chen, Saining Xie, and Kaiming He. An Empirical Study of Training Self-Supervised Vision Transformers. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 4, 7, 19

[14] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 2021. 13

[15] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment Inference for Invariant Learning. In *International Conference on Machine Learning*, 2021. 1, 3, 4, 5, 8, 17

[16] Alex J. DeGrave, Joseph D. Janizek, and Su-In Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 2021. 1

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1, 3, 4, 8, 19

[18] Greg d'Eon, Jason d'Eon, James R. Wright, and Kevin Leyton-Brown. The Spotlight: A General Method for Discovering Systematic Errors in Deep Learning Models. In *ACM Conference on Fairness, Accountability, and Transparency*, 2022. 26

[19] Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout, 2017. 4, 19

[20] Thomas G. Dietterich. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, 2000. 5

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 3, 4, 19

[22] Elias Eulig, Piyapat Saranrittichai, Chaithanya Kumar Mummadi, Kilian Rambach, William Beluch, Xiahan Shi, and Volker Fischer. DiagViB-6: A Diagnostic Benchmark Suite for Vision Models in the Presence of Shortcut and Generalization Opportunities. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 26

[23] Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Re. Domino: Discovering Systematic Errors with Cross-Modal Embeddings. In *International Conference on Learning Representations*, 2022. 26

[24] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan,

Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data Determines Distributional Robustness in Contrastive Language Image Pre-training (CLIP). *International Conference on Machine Learning*, 2022. 4

[25] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020. 1, 8

[26] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 1, 3, 4, 5, 8, 17, 19, 21, 22

[27] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision Models Are More Robust And Fair When Pretrained On Uncurated Images Without Supervision, 2022. 3, 4, 7, 8, 19

[28] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 14

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 7, 8, 16, 19

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3, 4, 19

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In *The European Conference on Computer Vision (ECCV)*, 2016. 19, 22

[32] Yue He, Zheyan Shen, and Peng Cui. Towards Non-I.I.D. image classification: A dataset and baselines. *Pattern Recognition*, 2021. 1, 8

[33] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 4, 8

[34] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2019. 8

[35] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *International Conference on Learning Representations*, 2020. 3, 4, 19

[36] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural Adversarial Examples. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8, 24

[37] Mark Ibrahim, Quentin Garrido, Ari Morcos, and Diane Bouchacourt. The Robustness Limits of SoTA Vision Models to Natural Variation, 2022. 26

[38] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. *Conference on Causal Learning and Reasoning*, 2022. 4, 5, 8, 13

[39] Badr Youbi Idrissi, Diane Bouchacourt, Randall Balestriero, Ivan Evtimov, Caner Hazirbas, Nicolas Ballas, Pascal Vincent, Michal Drozdzal, David Lopez-Paz, and Mark Ibrahim. ImageNet-X: Understanding Model Mistakes with Factor of Variation Annotations, 2022. 26

[40] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. 8

[41] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. On Feature Learning in the Presence of Spurious Correlations. In *Advances in Neural Information Processing Systems*, 2022. 8

[42] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling Model Failures as Directions in Latent Space, 2022. 26

[43] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. BiaSwap: Removing Dataset Bias With Bias-Tailored Swapping Augmentation. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 8

[44] Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning Debiased Classifier with Biased Committee. In *Advances in Neural Information Processing Systems*, 2022. 8

[45] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations, 2022. 4, 5, 6, 8, 16, 22

[46] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic Segmentation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 13

[47] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 8

[48] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *The European Conference on Computer Vision (ECCV)*, 2020. 19, 22

[49] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *The IEEE International Conference on Computer Vision Workshops*, 2013. 2, 13

[50] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. Ex-

plaining in Style: Training a GAN to explain a classifier in StyleSpace. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[51] Guillaume Leclerc, Hadi Salman, Andrew Ilyas, Sai Vemprala, Logan Engstrom, Vibhav Vineet, Kai Xiao, Pengchuan Zhang, Shibani Santurkar, Greg Yang, Ashish Kapoor, and Aleksander Madry. 3DB: A Framework for Debugging Computer Vision Models, 2021. 26

[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 8

[53] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover and Mitigate Unknown Biases with Debiasing Alternate Networks. In *The European Conference on Computer Vision (ECCV)*, 2022. 4, 5, 8, 17

[54] Zhiheng Li and Chenliang Xu. Discover the Unknown Biased Attribute of an Image Classifier. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 26

[55] Weixin Liang and James Zou. MetaShift: A Dataset of Datasets for Evaluating Contextual Distribution Shifts and Training Conflicts. In *International Conference on Learning Representations*, 2022. 8

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *The European Conference on Computer Vision (ECCV)*, 2014. 13

[57] Yong Lin, Shengyu Zhu, Lu Tan, and Peng Cui. ZIN: When and How to Learn Invariance Without Environment Partition? In *Advances in Neural Information Processing Systems*, 2022. 5, 8, 25, 26

[58] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just Train Twice: Improving Group Robustness without Training Group Information. *International Conference on Machine Learning*, 2021. 1, 3, 4, 5, 8, 17

[59] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 8

[60] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from Failure: De-biasing Classifier from Biased Classifier. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 4, 5, 8

[61] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality Not Quantity: On the Interaction between Dataset Design and Robustness of CLIP. In *Advances in Neural Information Processing Systems*, 2022. 4

[62] Zoe Papakipos and Joanna Bitton. AugLy: Data Augmentations for Robustness. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 16

[63] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient Starvation: A Learning Proclivity in Neural Networks. *Advances in Neural Information Processing Systems*, 2021. 4

[64] Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip H. S. Torr, and Puneet K. Dokania. RegMixup: Mixup as a Regularizer Can Surprisingly Improve Accuracy and Out Distribution Robustness. In *Advances in Neural Information Processing Systems*, 2022. 5

[65] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection. In *The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020. 23

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *International Conference on Machine Learning*, 2021. 3, 4, 7, 8, 19

[67] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing Network Design Spaces. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 4, 19

[68] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, 2019. 8, 18, 24

[69] Evgenia Rusak, Steffen Schneider, Peter Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Adapting ImageNet-scale models to complex distribution shifts with self-learning, 2021. 24

[70] Evgenia Rusak, Steffen Schneider, Peter Vincent Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. ImageNet-D: A new challenging robustness dataset inspired by domain adaptation. In *ICML 2022 Shift Happens Workshop*, 2022. 24

[71] Chaitanya K. Ryali, David J. Schwab, and Ari S. Morcos. Characterizing and Improving the Robustness of Self-Supervised Learning through Background Augmentations, 2021. 4, 5, 8, 16, 22

[72] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 4, 5, 8, 13, 16, 17, 26

[73] Christoph Schuhmann, Romain Beaumont, Cade W. Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R. Kundurthy, Katherine Crowson, Mitchell Wortsman, Richard Vencu, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 3, 4, 19

[74] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. In *Advances in Neural Information Processing Systems Workshops*, 2021. 3, 4, 19

[75] Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which Shortcut Cues Will DNNs Choose? A Study from the Parameter-Space Perspective. In *International Conference on Learning Representations*, 2022. 26

[76] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek

Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 3, 18

[77] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Unsupervised Learning of Debiased Representations With Pseudo-Attributes. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[78] Robik Shrestha, Kushal Kafle, and Christopher Kanan. An Investigation of Critical Issues in Bias Mitigation Techniques. In *The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022. 8

[79] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting Weakly Supervised Pre-Training of Visual Perception Models. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 4, 7, 19

[80] Sahil Singla and Soheil Feizi. Salient ImageNet: How to discover spurious features in Deep Learning? In *International Conference on Learning Representations*, 2022. 1

[81] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding Failures of Deep Networks via Robust Feature Extraction. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 26

[82] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No Subclass Left Behind: Fine-Grained Robustness in Coarse-Grained Classification Problems. In *Advances in Neural Information Processing Systems*, 2020. 8

[83] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1999. 4, 5, 6

[84] Vasilis Vryniotis. How to Train State-Of-The-Art Models Using TorchVision's Latest Primitives. https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives, 2021. 5, 16

[85] Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary C. Lipton. Learning Robust Global Representations by Penalizing Local Predictive Power. In *Advances in Neural Information Processing Systems*, 2019. 8, 23

[86] Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. Learning Robust Representations by Projecting Superficial Statistics Out. *International Conference on Learning Representations*, 2019. 8

[87] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4, 8

[88] Ross Wightman, Hugo Touvron, and Hervé Jégou. ResNet strikes back: An improved training procedure in timm, 2021. 5

[89] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022. 4, 7, 8, 19

[90] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[91] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or Signal: The Role of Image Backgrounds in Object Recognition. In *International Conference on Learning Representations*, 2021. 1, 3, 4, 5, 8, 16, 22

[92] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 4, 19, 25

[93] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*, 2018. 3, 4, 19

[94] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich Feature Construction for the Optimization-Generalization Dilemma. In *International Conference on Machine Learning*, 2022. 8

[95] Eric Zhao, De-An Huang, Hao Liu, Zhiding Yu, Anqi Liu, Olga Russakovsky, and Anima Anandkumar. Scaling Fair Learning to Hundreds of Intersectional Groups. In *Submitted to International Conference on Learning Representations*, 2022. 8

[96] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. *AAAI Conference on Artificial Intelligence*, 2020. 4, 19

[97] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 3, 14

Appendix

A. More Details of Datasets

A.1. UrbanCars Details

Here we present more details of the UrbanCars dataset.

Number of Images Regarding the number of images, each target class contains 4000 images in the training set, *i.e.*, 8000 images in total. That is, our training set is balanced regarding the target label and only imbalanced with shortcut labels. Therefore, UrbanCars does not have a target class imbalance issue [38] in Waterbirds dataset [72], where 76.8% of images are waterbird, and 23.3% of images are landbird. In validation and testing sets of UrbanCars, each split contains 500 images.

Data Annotation As mentioned in Sec. 2.1, each image is annotated with three image-level labels—car body type, background, and co-occurring object. Besides, following Waterbirds [72] dataset, the dataset also contains the mask annotation of the car object and the co-occurring object, which enables shortcut mitigation via targeted augmentation (category 2), *i.e.*, CF+F Aug [11] (*cf*. Appendix B.4) and our proposed LLE approach (*cf*. Appendix B.5).

Details of Data Construction Here, we present the details of collecting the data from source datasets based on three visual cues—main object (*i.e.*, car), background shortcut, and co-occurring object shortcut.

First, to obtain car images, we use MaskFormer [14] pretrained on MS-COCO [56] dataset's panoptic segmentation [46] task to segment cars from Stanford Cars [49] dataset. In each image from Stanford Cars, we choose the predicted car instance mask that has the largest IoU with the bounding box annotation provided in Stanford Cars. After segmentation, we run MaskFormer on foreground-only images to detect humans. Images with humans detected are filtered out.

When pasting the car object to the background, we first compute its square bounding box, which is the bounding box whose side length is the longer side of the actual bounding box of the car object based on the predicted segmentation mask. Then, we resize the square bounding box such that the side length is 50% of the final image size, which is smaller than the size of the car object.

We merge the original 196 classes in Stanford Cars into urban cars (*e.g.*, sedan, hatchback, *etc.*) and country cars (*e.g.*, pickup truck, van, *etc.*). The mapping from the original 196 classes in Stanford Cars to *urban cars* and *country cars* is as follows:

- *urban* cars: Acura RL Sedan 2012, Acura TL Sedan 2012, Acura TL Type-S 2008, Acura TSX Sedan 2012, Acura Integra Type R 2001, Acura ZDX Hatchback 2012, Aston Martin V8 Vantage Coupe 2012, Aston Martin Virage Convertible 2012, Aston Martin Virage Coupe 2012, Audi RS 4 Convertible 2008, Audi A5 Coupe 2012, Audi TTS Coupe 2012, Audi R8 Coupe 2012, Audi V8 Sedan 1994, Audi 100 Sedan 1994, Audi 100 Wagon 1994, Audi TT Hatchback 2011, Audi S6 Sedan 2011, Audi S5 Convertible 2012, Audi S5 Coupe 2012, Audi S4 Sedan 2012, Audi S4 Sedan 2007, Audi TT RS Coupe 2012, BMW ActiveHybrid 5 Sedan 2012, BMW 1 Series Convertible 2012, BMW 1 Series Coupe 2012, BMW 3 Series Sedan 2012, BMW 3 Series Wagon 2012, BMW 6 Series Convertible 2007, BMW M3 Coupe 2012, BMW M5 Sedan 2010, BMW M6 Convertible 2010, BMW Z4 Convertible 2012, Bentley Continental Supersports Conv. Convertible 2012, Bentley Arnage Sedan 2009, Bentley Mulsanne Sedan 2011, Bentley Continental GT Coupe 2012, Bentley Continental GT Coupe 2007, Bentley Continental Flying Spur Sedan 2007, Bugatti Veyron 16.4 Convertible 2009, Bugatti Veyron 16.4 Coupe 2009, Buick Regal GS 2012, Buick Verano Sedan 2012, Cadillac CTS-V Sedan 2012, Chevrolet Corvette Convertible 2012, Chevrolet Corvette ZR1 2012, Chevrolet Corvette Ron Fellows Edition Z06 2007, Chevrolet Camaro Convertible 2012, Chevrolet Impala Sedan 2007, Chevrolet Sonic Sedan 2012, Chevrolet Cobalt SS 2010, Chevrolet Malibu Hybrid Sedan 2010, Chevrolet Monte Carlo Coupe 2007, Chevrolet Malibu Sedan 2007, Chrysler Sebring Convertible 2010, Chrysler 300 SRT-8 2010, Chrysler Crossfire Convertible 2008, Chrysler PT Cruiser Convertible 2008, Daewoo Nubira Wagon 2002, Dodge Caliber Wagon 2012, Dodge Caliber Wagon 2007, Dodge Magnum Wagon 2008, Dodge Challenger SRT8 2011, Dodge Charger Sedan 2012, Dodge Charger SRT-8 2009, Eagle Talon Hatchback 1998, FIAT 500 Abarth 2012, FIAT 500 Convertible 2012, Ferrari FF Coupe 2012, Ferrari California Convertible 2012, Ferrari 458 Italia Convertible 2012, Ferrari 458 Italia Coupe 2012, Fisker Karma Sedan 2012, Ford Mustang Convertible 2007, Ford GT Coupe 2006, Ford Focus Sedan 2007, Ford Fiesta Sedan 2012, Geo Metro Convertible 1993, Honda Accord Coupe 2012, Honda Accord Sedan 2012, Hyundai Veloster Hatchback 2012, Hyundai Sonata Hybrid Sedan 2012, Hyundai Elantra Sedan 2007, Hyundai Accent Sedan 2012, Hyundai Genesis Sedan 2012, Hyundai Sonata Sedan 2012, Hyundai Elantra Touring Hatchback 2012, Hyundai Azera Sedan 2012, Infiniti G Coupe IPL

2012, Jaguar XK XKR 2012, Lamborghini Reventon Coupe 2008, Lamborghini Aventador Coupe 2012, Lamborghini Gallardo LP 570-4 Superleggera 2012, Lamborghini Diablo Coupe 2001, Lincoln Town Car Sedan 2011, MINI Cooper Roadster Convertible 2012, Maybach Landaulet Convertible 2012, McLaren MP4-12C Coupe 2012, Mercedes-Benz 300-Class Convertible 1993, Mercedes-Benz C-Class Sedan 2012, Mercedes-Benz SL-Class Coupe 2009, Mercedes-Benz E-Class Sedan 2012, Mercedes-Benz S-Class Sedan 2012, Mitsubishi Lancer Sedan 2012, Nissan Leaf Hatchback 2012, Nissan Juke Hatchback 2012, Nissan 240SX Coupe 1998, Plymouth Neon Coupe 1999, Porsche Panamera Sedan 2012, Rolls-Royce Phantom Drophead Coupe Convertible 2012, Rolls-Royce Ghost Sedan 2012, Rolls-Royce Phantom Sedan 2012, Scion xD Hatchback 2012, Spyker C8 Convertible 2009, Spyker C8 Coupe 2009, Suzuki Aerio Sedan 2007, Suzuki Kizashi Sedan 2012, Suzuki SX4 Hatchback 2012, Suzuki SX4 Sedan 2012, Tesla Model S Sedan 2012, Toyota Camry Sedan 2012, Toyota Corolla Sedan 2012, Volkswagen Golf Hatchback 2012, Volkswagen Golf Hatchback 1991, Volkswagen Beetle Hatchback 2012, Volvo C30 Hatchback 2012, Volvo 240 Sedan 1993, smart fortwo Convertible 2012.

- *country* cars: AM General Hummer SUV 2000, Aston Martin V8 Vantage Convertible 2012, BMW X5 SUV 2007, BMW X6 SUV 2012, BMW X3 SUV 2012, Buick Rainier SUV 2007, Buick Enclave SUV 2012, Cadillac SRX SUV 2012, Cadillac Escalade EXT Crew Cab 2007, Chevrolet Silverado 1500 Hybrid Crew Cab 2012, Chevrolet Traverse SUV 2012, Chevrolet HHR SS 2010, Chevrolet Tahoe Hybrid SUV 2012, Chevrolet Express Cargo Van 2007, Chevrolet Avalanche Crew Cab 2012, Chevrolet TrailBlazer SS 2009, Chevrolet Silverado 2500HD Regular Cab 2012, Chevrolet Silverado 1500 Classic Extended Cab 2007, Chevrolet Express Van 2007, Chevrolet Silverado 1500 Extended Cab 2012, Chevrolet Silverado 1500 Regular Cab 2012, Chrysler Aspen SUV 2009, Chrysler Town and Country Minivan 2012, Dodge Caravan Minivan 1997, Dodge Ram Pickup 3500 Crew Cab 2010, Dodge Ram Pickup 3500 Quad Cab 2009, Dodge Sprinter Cargo Van 2009, Dodge Journey SUV 2012, Dodge Dakota Crew Cab 2010, Dodge Dakota Club Cab 2007, Dodge Durango SUV 2012, Dodge Durango SUV 2007, Ford F-450 Super Duty Crew Cab 2012, Ford Freestar Minivan 2007, Ford Expedition EL SUV 2009, Ford Edge SUV 2012, Ford Ranger SuperCab 2011, Ford F-150 Regular Cab 2012, Ford F-150 Regular Cab 2007, Ford E-Series Wagon Van 2012, GMC Terrain SUV 2012, GMC Savana Van 2012, GMC Yukon Hybrid SUV 2012, GMC Acadia SUV 2012, GMC Canyon Extended Cab 2012, HUMMER H3T Crew Cab 2010, HUMMER H2 SUT Crew Cab 2009, Honda Odyssey Minivan 2012, Honda Odyssey Minivan 2007, Hyundai Santa Fe SUV 2012, Hyundai Tucson SUV 2012, Hyundai Veracruz SUV 2012, Infiniti QX56 SUV 2011, Isuzu Ascender SUV 2008, Jeep Patriot SUV 2012, Jeep Wrangler SUV 2012, Jeep Liberty SUV 2012, Jeep Grand Cherokee SUV 2012, Jeep Compass SUV 2012, Land Rover Range Rover SUV 2012, Land Rover LR2 SUV 2012, Mazda Tribute SUV 2011, Mercedes-Benz Sprinter Van 2012, Nissan NV Passenger Van 2012, Ram C/V Cargo Van Minivan 2012, Toyota Sequoia SUV 2012, Toyota 4Runner SUV 2012, Volvo XC90 SUV 2007.

Second, regarding the background images for the background shortcut, we use images from the Places [97] dataset, where the *urban* background images are from alley, crosswalk, downtown, gas station, garage (outdoor), driveway classes, and the *country* background images are forest road, field road, desert road. We use MaskFormer mentioned above to detect humans, cars, and co-occuring objects (*e.g.*, fireplug) on Places images. Images with the aforementioned object categories detected will be filtered out. When used as the background image in UrbanCars, we resize each image to 256×256.

Lastly, the co-occurring objects are from LVIS [28] based on its ground-truth instance segmentation mask, where *urban* co-occurring object images are from fireplug, stop sign, street sign, parking meter, traffic light and *country* co-occurring object images are from farm animals—cow, horse, sheep. We filter out instance masks with more than one connected component (*e.g.*, instances with more than one connected component are usually occluded by other objects). When pasting to the background, the square bounding box (see above) of the co-occurring object is resized such that the side length is 25% of the final image size.

Dataset Release Since Places dataset (the source dataset for the background) does not own the copyright of images, we cannot directly release the final images in UrbanCars. Instead, we release the code that creates the UrbanCars from source datasets.

A.2. ImageNet-Watermark (ImageNet-W) Details

Here we show more details about creating the ImageNet-Watermark dataset. Regarding the position, we paste the watermark at the center of the image. More specifically, the XY-position of the top-left corner of the watermark is $(0.01W, 0.4H)$, where W and H are the width and height of the input image for the models. Regarding the font size, we use 36 for the 224×224 sized images, which is the most common input size for most vision models. For large foundation models using larger input sizes, we use $62, 82, 84$ for 384×384, 512×512, 518×518 sized images, respectively, where the font sizes are approximately 0.16 times smaller to the image size. The font color for the watermark is (255, 255, 255, 128) in RGBA, which is a transparent white color. We use the open-sourced “SourceHanSerifSC-ExtraLight”[†] as the font family.

Content of Watermark As mentioned in Sec. 2.2, we use “捷径捷径捷径” as the content of the watermark. We show the results of using other contents or languages in Tab. 8. When using other content in Simplified Chinese (*e.g.*, “一二三四五六”) or other languages (*i.e.*, Japanese, Korean, English, and Arabic), we observe smaller IN-W Gap and Carton Gap. We conjecture this is due to the simpler shape of other contents compared to “捷径捷径捷径” used in the ImageNet-W. Nevertheless, the accuracy drops across different contents suggest that it is the presence of the watermark rather than its content that causes the watermark shortcut reliance. Besides, the watermark shortcut reliance is stronger when the watermark’s content looks more visually similar to the pattern of the watermark in carton class images in ImageNet-1k training set, *e.g.*, Simplified Chinese characters with complex shapes (*cf*. Fig. 7).

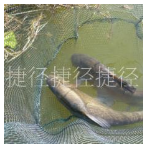
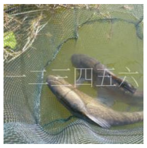
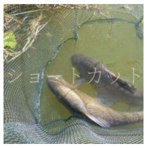
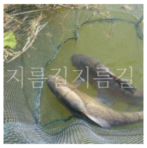
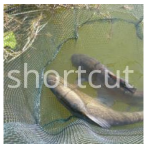
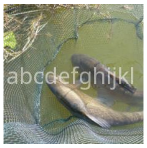
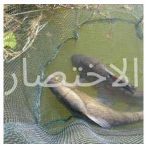
watermark content	language	English translation	Example Image	IN-W Gap	Carton Gap
捷径捷径捷径	Simplified Chinese	shortcut shortcut shortcut		-26.64	+40
一二三四五六	Simplified Chinese	one two three four five six		-6.12	+22
ショートカット	Japanese	shortcut		-2.66	+18
지름길지름길	Korean	shortcut shortcut		-12.30	+34
shortcut	English	N/A		-6.39	+8
abcdefghijkl	English	N/A		-5.54	+4
الاختصار	Arabic	shortcut		-7.79	+4

Table 8. Ablation study on ResNet-50’s reliance on the watermark shortcut in different content and languages. Watermarks of various contents can cause shortcut reliance. We choose the content shown in the first row for creating ImageNet-W as it causes a larger IN-W Gap and Carton Gap and is more visually similar to the watermark that appears in the IN-1k training set.

[†] https://source.typekit.com/source-han-serif/

Dataset Release We release the code of adding watermarks instead of directly releasing the final images. We follow AugLy [62] to implement the code of adding watermarks, which is encapsulated as a function similar to PyTorch's transforms API. It is easy to use and can evaluate vision models on the fly by simply adding the watermark transform function with ImageNet-1k validation set downloaded, *i.e*., no need to save images with watermarks to the disk in advance.

B. Implementation Details

Here we present more details of the benchmark methods and our Last Layer Ensemble approach.

B.1. Watermark Augmentation (WMK Aug)

To mitigate the watermark shortcut on ImageNet, we propose simple-yet-effective watermark augmentation (WMK Aug). Concretely, we overlay a random watermark onto the training images in ImageNet-1k. The watermark is random in terms of (1) position, (2) font size, and (3) content, where we use random CJK (Chinese, Japanese, and Korean) characters in a random number of characters. The randomness of watermark augmentation in training avoids being identical to the watermark used for evaluation on ImageNet-W.

B.2. Background Augmentation (BG Aug)

To mitigate the background shortcut on ImageNet, we follow [71,91] and use background augmentation (BG Aug). Concretely, we use unsupervised saliency segmentation developed by Ryali *et al.* [71] to separate the foreground object from the backgrounds in each image. Then "tiled" background images are created by repeating the procedure of pasting the largest rectangular of the background onto the foreground region to cover the foreground object (more details in [91]). Finally, to augment the background, we paste the segmented foreground object from class A onto a tiled background from class B (A≠B).

B.3. Detailed Experiment Settings

UrbanCars On UrbanCars, we follow the standard regularization setting in [72]. Concretely, we use stochastic gradient descent (SGD) optimizer with 10^{-3} learning rate and 10^{-4} weight decay (*i.e*., ℓ_2 penalty). We use 128 for the batch size. All models are trained with 300 epochs, and we use the early stopped epoch that achieves the best validation set worst-group accuracy to report the final results on the testing set. Specifically, for methods that do not use ground-truth shortcut labels (*i.e*., category 1, 2, 4), the worst-group accuracy is computed based on labels of both shortcuts, *i.e*., lowest accuracy among all eight groups. Methods using shortcut labels (*i.e*., category 3) may encounter the issue in which one or a subset of shortcuts remain unlabeled or even unknown. To simulate the situation, besides standard setting using labels of both shortcuts, we additionally create two settings—(1) only using BG label; (2) only using CoObj label (*cf*. bottom two sections in Tab. 5). In both cases, the worst-group accuracy on the validation set also only considers the label of one shortcut, *i.e*., the lowest accuracy among four groups based on the combination of the target label and the single shortcut label. Each experiment on UrbanCars is repeated six times using different random seeds, and we report the average results over six runs.

ImageNet On ImageNet, we use last layer re-training [45] to only train the last classification layer upon a frozen feature extractor to benchmark methods in Tab. 4 and our Last Layer Ensemble (LLE) method in Tab. 6. Note that we directly evaluate self-supervised approaches and foundation models in Tab. 6 without using last layer re-training. When using ResNet-50 network architecture with last layer re-training (*i.e*., methods in Tab. 4), we use SGD optimizer with 10^{-4} weight decay. For all models, we tune the learning rate over $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and choose the one with the best top-1 accuracy on IN-1k. We use 1024 for the batch size. Unlike the detailed implementation in [45], we do not train the last classification layer from scratch but initialize it by the weights of ERM's last layer because we find that the latter way converges faster. Note that ERM's last layer is also re-trained (*e.g*., ERM in Tab. 4). When applying our LLE approach with the MAE feature extractor, we follow MAE [29] to use 0 weight decay.

B.4. Details of Benchmark Methods

We introduce more details (*e.g*., hyperparameters) of benchmark methods in each category.

Category 1: Standard Augmentation and Regularization Following PyTorch's new training recipe [84], we use $\alpha = 0.2$ for Mixup, $p = 0.1$ for Cutout, and $\alpha = 1.0$ for CutMix on both UrbanCars and ImageNet experiments. For AugMix, we use all default hyperparameters in the original implementation. For the co-efficient of ℓ_2 penalty of logits in SD, we use 0.1 on UrbanCars and 10^{-4} on ImageNet (we find that SD using 0.1 on ImageNet achieves poor results).

Category 2: Targeted Augmentation for Mitigating Shortcuts For CF+F Aug [11], based on the ground-truth masks (*cf*. Appendix A.1), we use CF(Grey) and F(Random) for generating counterfactual and factual augmentations because they achieve the best results on Waterbirds when not using external generative models. Concretely, CF(Grey) infills the grey color to the bounding box area of the object to generate the counterfactual image, and F(Random) uses random noises to replace the background area—outside of the bounding box of the car object (more details in [11]).

For style transfer [26] (*i.e*., texture augmentation or TXT Aug), we use the official code to generate Stylized ImageNet (SIN) for training. The details of BG Aug and WTM Aug are introduced in Appendix B.2 and Appendix B.1, respectively. Note that WTM Aug, TXT Aug, and BG Aug shown in Tab. 4 jointly use augmented images and original IN-1k images for training.

Category 3: Using Shortcut Labels We follow the original GroupDRO (gDRO)'s implementation to use 0.01 step size and $\gamma = 0.1$. For Domain Independent (DI), its number of domains is decided based on the usage of shortcut labels, *i.e*., 2 when using labels of only one shortcut and 4 for using labels of both shortcuts. We follow SUBG's implementation to subsample the training data to rebalance the data, where each group has (fewer but) the same number of images. For DFR, we use its DFR$^{Tr}_{Tr}$ variant where ERM's last layer is re-trained on a balanced sub-sampled training set (*i.e*., SUBG).

Category 4: Inferring Pseudo Shortcut Labels For LfF, we follow the original implementation to set $q = 0.7$. As discussed in Sec. 5.2, JTT and EIIL use an early-stopped ERM as the reference model to infer the pseudo shortcut labels, where we use E to denote the number of training epochs of the reference ERM model. For JTT, we use E=1 and E=2 on UrbanCars. Since JTT [58] use E=40,50,60 on Waterbirds, we also show their results on UrbanCars in Appendix C.1. We use $\lambda_{\text{up}} = 100$ for JTT on UrbanCars. On ImageNet, we use E=1 and $\lambda_{\text{up}} = 5$ because we found $\lambda_{\text{up}} = 100$ (*i.e*., sampling wrongly predicted examples 100 times) is not scalable on the larger ImageNet dataset. For EIIL, we use E=1 and E=2 on UrbanCars and E=1 on ImageNet. We use gDRO as the invariant learner for EIIL (more details in [15]). While DebiAN uses a full network as the shortcut "discoverer" (more details in [53]), we use a single fully-connected layer on top of the feature extractor for its experiments on ImageNet under the last layer re-training setting.

B.5. Details of Last Layer Ensemble (LLE)

On UrbanCars, we augment background and co-occurring object visual cues to mitigate multiple shortcuts based on ground-truth masks (*cf*. Appendix A.1). Concretely, we use ground-truth masks of the car object and co-occurring object to (1) segment car object; (2) segment co-occurring object; (3) create the tiled background, a background-only image where the regions of the object and co-occurring object are tiled (*cf*., Appendix B.2). To augment the background, we sample segmented car object and co-occurring object from class A and tiled background from class B (A≠B), which are used to form the background-augmented images—pasting car object and co-occurring object on the tiled background. Similarly, to augment the co-occurring object, we sample the segmented car object and tiled background from class A and sample the segmented co-occurring object from class B (A≠B) to create the augmented images. Note that we only use the target label of the car body type for augmentation. In other words, neither the BG shortcut labels nor the CoObj shortcut labels are used. After obtaining two types of augmented images, LLE uses three last classification layers as an ensemble—two layers for two shortcuts and one layer for the original images. The distributional shift classifier predicts three shift categories: (1) no shift (*i.e*., original images), (2) background shift (*i.e*., background-augmented images), (3) co-occurring object shift (*i.e*., co-occurring object augmented images).

On ImageNet, LLE uses style transfer [26] (*i.e*., TXT Aug) to mitigate the texture shortcut, BG Aug (details in Appendix B.2) to mitigate the background shortcut, and WMK Aug (details in Appendix B.1) to mitigate the watermark shortcut. LLE jointly trains four last classification layers as an ensemble—three layers for three shortcuts and one layer for original images in IN-1k. The distributional shift classifier predicts four categories: (1) no shift (original images from IN-1k), (2) texture shift (*i.e*., texture augmented images), (3) background shift (*i.e*., background augmented images), (4) watermark shift (*i.e*., watermark augmented images).

C. More Results on UrbanCars

C.1. More Results of JTT

In Sec. 5.2, we show the result of JTT when E=1 and E=2. Since JTT tunes E over $\{40, 50, 60\}$ on Waterbirds [72] (more epochs for training the reference ERM models to infer pseudo shortcut labels). Here, we also show the results of JTT when E

		shortcut reliance		
	I.D. Acc	BG Gap	CoObj Gap	BG+CoObj Gap
ERM	97.9	-16.3	-11.9	-69.5
JTT (E=1)	95.9	-8.1	-13.3 (×1.18)	-37.6
JTT (E=2)	94.6	-23.3 (×1.52)	-5.3	-52.1
JTT (E=40)	97.7	-15.8 (×1.03)	-10.7	-69.3
JTT (E=50)	97.6	-14.8	-11.0	-67.9
JTT (E=60)	97.2	-15.1	-10.7	-70.5

Table 9. Results of JTT when using ERM trained with other epochs (E $\in \{40, 50, 60\}$) as the reference model to infer pseudo shortcut labels.

		Shortcut Reliance		
	I.D. Acc	BG Gap	CoObj Gap	BG+CoObj Gap
ERM	97.9	-16.3	-11.9	-69.5
w/o stop gradient	97.3	-3.3	-2.6	-7.7
w/ frozen feature extractor	97.3	-11.2	-9.0	-43.3
LLE	96.7	-2.1	-2.7	-5.9

Table 10. Results of ablation study of LLE on UrbanCars dataset.

$\in \{40, 50, 60\}$ in Tab. 9, where JTT either exhibits Whac-A-Mole results by amplifying the background shortcut or barely mitigates either shortcut compared to ERM.

C.2. Ablation Study of LLE on UrbanCars

As mentioned in Sec. 4, when training the distributional shift classifier, we stop the gradient from the distributional shift classifier to the feature extractor under the end-to-end training setting on UrbanCars. Here we show the ablation study in Tab. 10, where the variant without stopping the gradient achieves suboptimal results. The results demonstrate the necessity of stopping the gradient to prevent the feature extractor from learning the shortcut information used in the distributional shift classifier's supervision.

While we use the end-to-end training setting for experiments on UrbanCars, we also show the results of LLE with the last layer re-training setting (*cf*. frozen feature extractor in Tab. 10), which shows that using a frozen feature extractor can also improve the results over ERM, but the results are also suboptimal compared to end-to-end training.

D. More Results on ImageNet-W

D.1. Results of More Methods on ImageNet-W

The results of more methods in addition to methods in Tab. 1 are shown in Tab. 11. We observe a pervasive watermark shortcut reliance across network architecture, pretraining datasets, supervision, mitigation methods, *etc*.

D.2. ImageNetV2-W: ImageNet-W with ImageNetV2

To further verify the pervasiveness of watermark shortcut reliance, we also overlay the watermark on ImageNetV2 [68] dataset to construct the ImageNet-W test set. We denote this ImageNet-W variant as **ImageNetV2-W**. The results are shown in Tab. 12, which is comparable to results on ImageNet-W shown in Tab. 1. Note that some models show +0 Carton Gap results (*e.g*., CLIP pretrained on WIT and LAION-400M). We conjecture that it is due to the small number (*i.e*., ten) of carton class images in ImageNetV2. Nevertheless, they still show a considerable likelihood increase of carton class images ($\Delta P(\hat{y} = \text{carton} \mid y = \text{carton})$). Therefore, the results on ImageNetV2-W strengthen our claim of the watermark shortcut for predicting the carton class learned by various vision models.

D.3. More Qualitative Examples of Watermark Shortcut

Many Carton Class Images in ImageNet-1k Training Set Contain Watermark We show more watermark examples of carton class images in ImageNet-1k training set. As shown in Fig. 7, these images contain the watermark written in Chinese characters. We also show ResNet-50's saliency maps [76] for predicting the carton class. While they highlight the watermark region, it may still be hard to interpret because the watermark and the carton object share similar spatial locations. This could be one of the reasons why previous works did not discover this shortcut.

Adding Watermark to Carton Class Images in IN-1k Validation Set (*i.e*., IN-W) Leads to Carton Class Predictions Our ImageNet-W can better address the difficulty of interpreting the watermark shortcut by providing the counterfactual explanations. In Fig. 8a, we first show carton class images in ImageNet-1k validation set that are predicted incorrectly by ResNet-50 (*e.g*., cradle, paper towel, *etc*.). By adding the watermark to the images, we show that not only are the predictions altered to carton but also the highlighted regions of the saliency maps are shifted to the watermark.

method	architecture	(pre)training data	IN-1k Acc ↑	$P(\hat{y} = \text{carton})$ (%)	IN-W Gap ↑	$\Delta P(\hat{y} = \text{carton})$ (%) ↓	Carton Gap ↓	$\Delta P(\hat{y} = \text{carton} \mid y = \text{carton})$ (%) ↓
Supervised	ResNet-50 [30]	IN-1k [17]	76.1	0.07	-26.7	+7.56	+40	+42.46
MoCov3 [13] (LP)	ResNet-50	IN-1k	74.6	0.08	-20.7	+2.94	+44	+44.37
Style Transfer [26]	ResNet-50	SIN [26]	60.1	0.10	-17.3	+4.91	+52	+50.06
Mixup [93]	ResNet-50	IN-1k	76.1	0.07	-18.6	+3.43	+38	+39.78
CutMix [92]	ResNet-50	IN-1k	78.5	0.09	-14.8	+1.92	+22	+29.61
Cutout [19,96]	ResNet-50	IN-1k	77.0	0.08	-18.0	+2.93	+32	+38.06
AugMix [35]	ResNet-50	IN-1k	77.5	0.09	-16.8	+2.61	+36	+34.44
BiT-M [48]	ResNet-50v2 [31]	IN-21k	82.3	0.09	-8.6	+0.60	+28	+29.73
Supervised	RG-32gf	IN-1k	80.8	0.09	-14.1	+3.74	+32	+33.43
SEER [27] (FT)	RG-32gf [67]	IG-1B [27]	83.3	0.09	-6.5	+0.56	+18	+24.26
SWAG [79] (LP)	RG-32gf	IG-3.6B [79]	84.6	0.08	-6.5	+0.36	+22	+20.56
SWAG (FT)	RG-32gf	IG-3.6B	86.8	0.08	-4.5	+0.49	+30	+26.03
Supervised	ViT-B/32 [21]	IN-1k	75.9	0.09	-8.7	+1.20	+34	+34.31
Uniform Soup [89] (FT)	ViT-B/32	WIT [66]	79.9	0.09	-7.9	+0.32	+24	+23.87
Greedy Soup [89] (FT)	ViT-B/32	WIT	81.0	0.09	-6.5	+0.35	+16	+23.87
Supervised	ViT-B/16	IN-1k	81.0	0.08	-6.7	+0.73	+26	+31.28
RobustViT [12]	ViT-B/16	IN-1k	80.3	0.08	-7.3	+0.44	+34	+37.06
MoCov3 (LP)	ViT-B/16	IN-1k	76.6	0.09	-16.0	+1.97	+22	+38.34
MAE [29] (FT)	ViT-B/16	IN-1k	83.7	0.09	-4.6	+0.67	+24	+22.46
SWAG (LP)	ViT-B/16	IG-3.6B	81.8	0.08	-7.7	+0.46	+18	+19.74
SWAG (FT)	ViT-B/16	IG-3.6B	85.2	0.09	-5.4	+0.45	+24	+25.95
Supervised	ViT-L/16	IN-1k	79.6	0.08	-6.2	+0.82	+34	+32.57
MAE (FT)	ViT-L/16	IN-1k	85.9	0.09	-4.4	+0.50	+22	+22.70
SWAG (LP)	ViT-L/16	IG-3.6B	85.1	0.08	-5.7	+0.23	**+6**	+9.72
SWAG (FT)	ViT-L/16	IG-3.6B	88.0	0.09	-3.2	+0.24	+20	+19.14
CLIP [66] (zero-shot)	ViT-L/14	WIT [66]	76.5	0.06	-4.4	**+0.01**	+12	**+1.75**
CLIP (zero-shot)	ViT-L/14	LAION-400M [74]	72.7	0.05	-4.9	+0.03	+12	+13.76
MAE (FT)	ViT-H/14	IN-1k	86.9	0.08	-3.5	+0.43	+30	+29.59
SWAG (LP)	ViT-H/14	IG-3.6B	85.7	0.09	-4.9	+0.19	+8	+12.80
SWAG (FT)	ViT-H/14	IG-3.6B	88.5	0.09	**-3.1**	+0.35	+18	+20.25
CLIP (zero-shot)	ViT-H/14	LAION-2B [73]	77.9	0.06	-3.6	+0.03	+16	+12.01
CLIP (zero-shot)	ViT-G/14	LAION-2B	76.6	0.06	-3.8	+0.02	+12	+5.61

Table 11. Results of more methods (also include the methods in Tab. 1). LP and FT stand for linear probing and fine-tuning on ImageNet-1k, respectively.

method	architecture	(pre)training data	IN-1k Acc ↑	$P(\hat{y} = \text{carton})$ (%)	IN-W Gap ↑	$\Delta P(\hat{y} = \text{carton})$ (%) ↓	Carton Gap ↓	$\Delta P(\hat{y} = \text{carton} \mid y = \text{carton})$ (%) ↓
Supervised	ResNet-50	IN-1k	63.19	0.09	-26.07	+9.29	+70	+53.50
MoCov3 (LP)	ResNet-50	IN-1k	61.98	0.09	-19.83	+3.33	+40	+44.43
Style Transfer	ResNet-50	SIN	48.63	0.09	-15.88	+5.16	+40	+40.28
Supervised	RG-32gf	IN-1k	69.67	0.10	-16.59	+5.21	+40	+34.09
SEER (FT)	RG-32gf	IG-1B	72.48	0.08	-9.00	+0.76	+30	+31.03
SWAG (LP)	RG-32gf	IG-3.6B	75.51	0.10	-7.48	+0.45	+20	+17.57
SWAG (FT)	RG-32gf	IG-3.6B	78.18	0.09	-5.67	+0.74	+30	+27.15
Supervised	ViT-B/32	IN-1k	62.99	0.07	-8.45	+1.39	+30	+20.97
Uniform Soup (FT)	ViT-B/32	WIT	68.58	0.08	-8.57	+0.42	+60	+47.84
Greedy Soup (FT)	ViT-B/32	WIT	69.54	0.08	-7.43	+0.44	+50	+40.78
Supervised	ViT-B/16	IN-1k	69.55	0.09	-7.55	+0.92	+40	+22.66
MoCov3 (LP)	ViT-B/16	IN-1k	65.25	0.09	-16.32	+2.40	+50	+41.75
MAE (FT)	ViT-B/16	IN-1k	73.20	0.10	-6.12	+1.05	+50	+38.12
SWAG (LP)	ViT-B/16	IG-3.6B	72.87	0.10	-8.66	+0.55	+10	+20.01
SWAG (FT)	ViT-B/16	IG-3.6B	75.57	0.09	-6.51	+0.66	+40	+32.34
Supervised	ViT-L/16	IN-1k	67.49	0.07	-7.37	+0.99	+30	+37.09
MAE (FT)	ViT-L/16	IN-1k	76.65	0.10	-6.57	+0.87	+40	+33.43
SWAG (LP)	ViT-L/16	IG-3.6B	76.64	0.09	-6.71	+0.30	+30	+12.46
SWAG (FT)	ViT-L/16	IG-3.6B	80.39	0.10	-4.14	+0.36	+20	+30.21
CLIP (zero-shot)	ViT-L/14	WIT	70.87	0.09	-5.29	+0.02	+0	+4.20
CLIP (zero-shot)	ViT-L/14	LAION-400M	65.43	0.06	-5.90	+0.02	+0	+9.44
MAE (FT)	ViT-H/14	IN-1k	78.46	0.10	-5.26	+0.71	+30	+31.43
SWAG (LP)	ViT-H/14	IG-3.6B	77.38	0.10	-6.46	+0.23	+0	+10.74
SWAG (FT)	ViT-H/14	IG-3.6B	81.06	0.09	-4.39	+0.46	+10	+21.45
CLIP (zero-shot)	ViT-H/14	LAION-2B	70.92	0.08	-4.44	+0.02	+30	+19.09
CLIP (zero-shot)	ViT-G/14	LAION-2B	69.65	0.09	-5.16	+0.02	+20	+9.96

Table 12. Results of watermark shortcut with ImageNet-V2.

Adding Watermark to Non-Carton Class Images in IN-1k Validation Set (*i.e.*, IN-W) Leads to Carton Class Predictions Similarly, we also the qualitative results for non-carton class images in Fig. 8b. While ResNet-50 makes correct predictions for non-carton class images (*e.g.*, indigo bunting, brambling, hen, *etc.*) on IN-1k, the predictions are switched to carton class after adding watermarks to the images. Besides, the saliency maps show that the ResNet-50 shifts its attention from the object to the watermark shortcut.

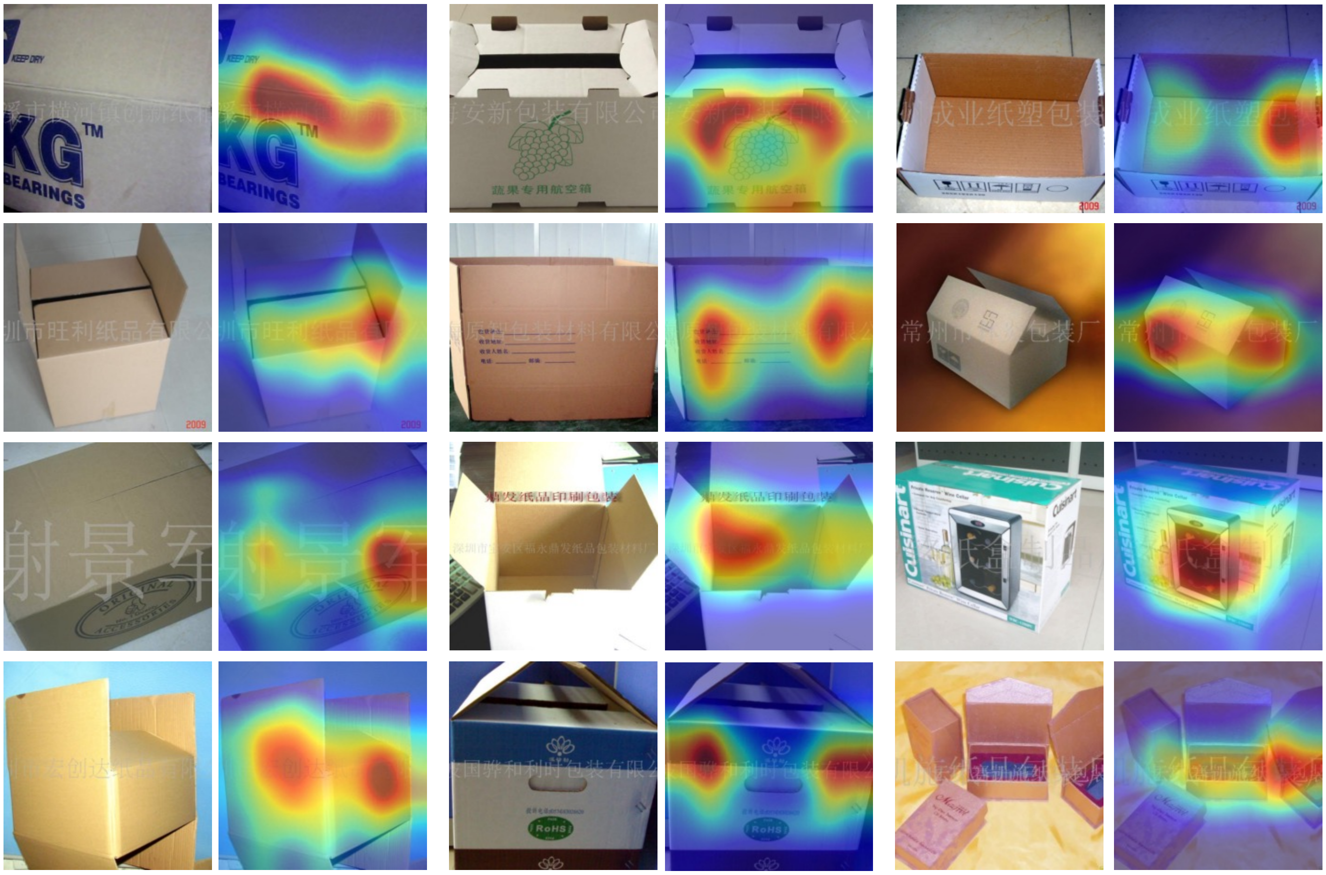
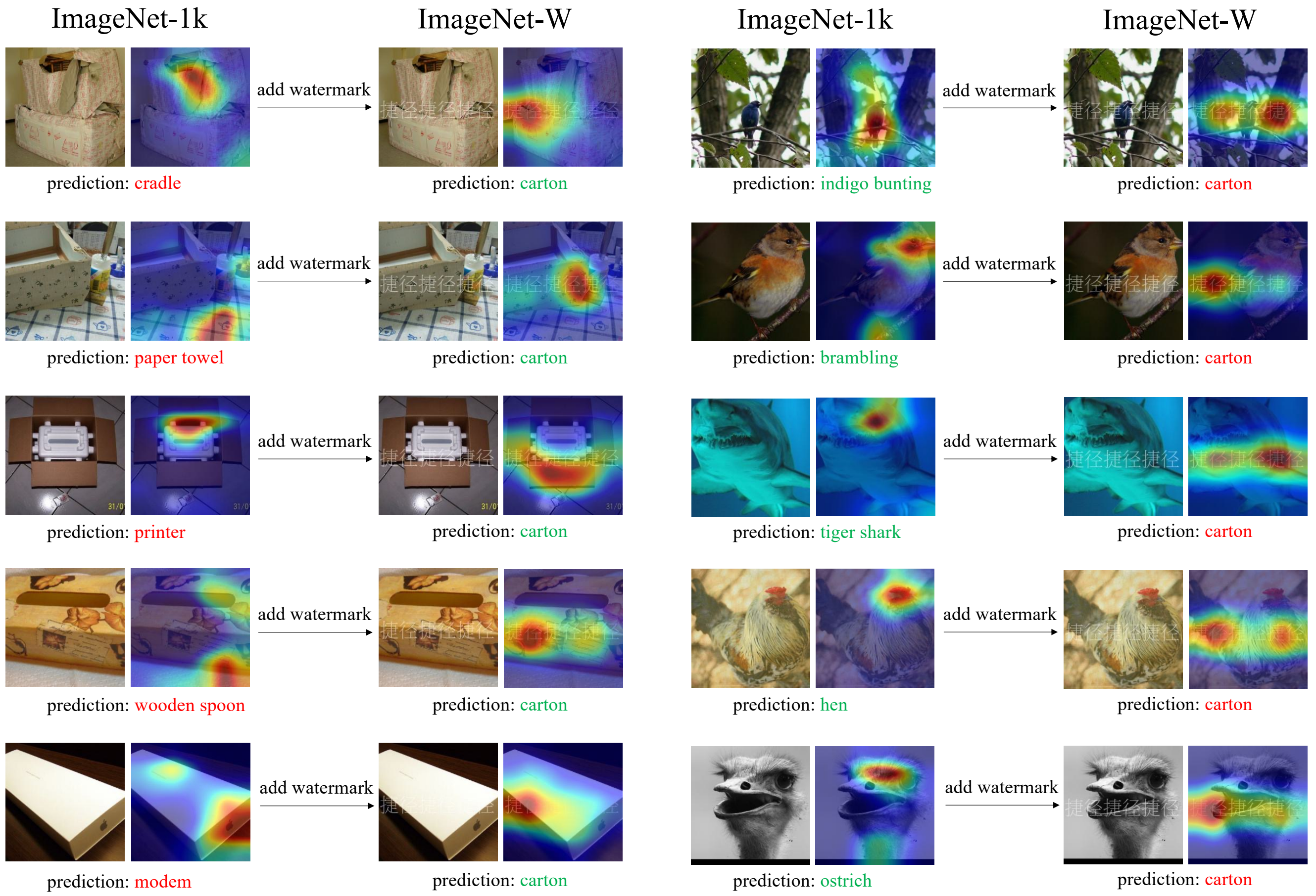


Figure 7. More examples of carton class images with watermark in ImageNet-1k training set. The saliency maps show that ResNet-50 relies on the watermark shortcut to predict carton.



(a) More examples of carton class images. ResNet-50 mispredicts many of them on ImageNet-1k validation set (left column). On ImageNet-W, after adding watermarks to carton class images from ImageNet-1k, ResNet-50 uses watermark as the shortcut to achieve correct predictions (right column).

(b) More examples of images that are not from the carton class. ResNet-50 predicts many of them correctly on ImageNet-1k's validation set (left column). On ImageNet-W, after adding watermarks to carton class images from ImageNet-1k, ResNet-50 uses the watermark as the shortcut to make incorrect predictions as the carton class (right column).

Figure 8. Adding watermarks alters the prediction and focused region of ResNet-50.

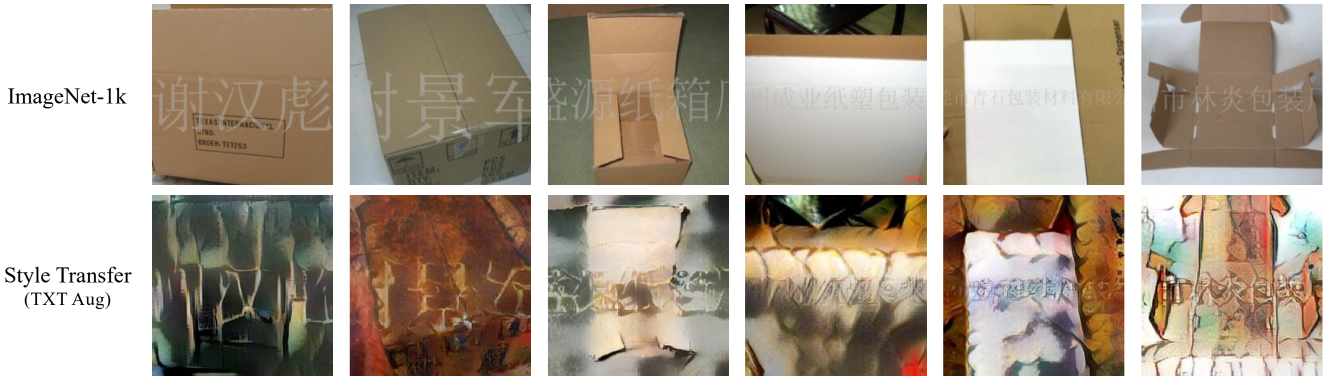


Figure 9. Examples of Style Transfer augmentation [26] (TXT Aug) on carton class images from ImageNet-1k training set. Although the augmentation is designed to mitigate the texture shortcut by increasing the "shape bias," it unexpectedly preserves or amplifies the shape of the watermark.

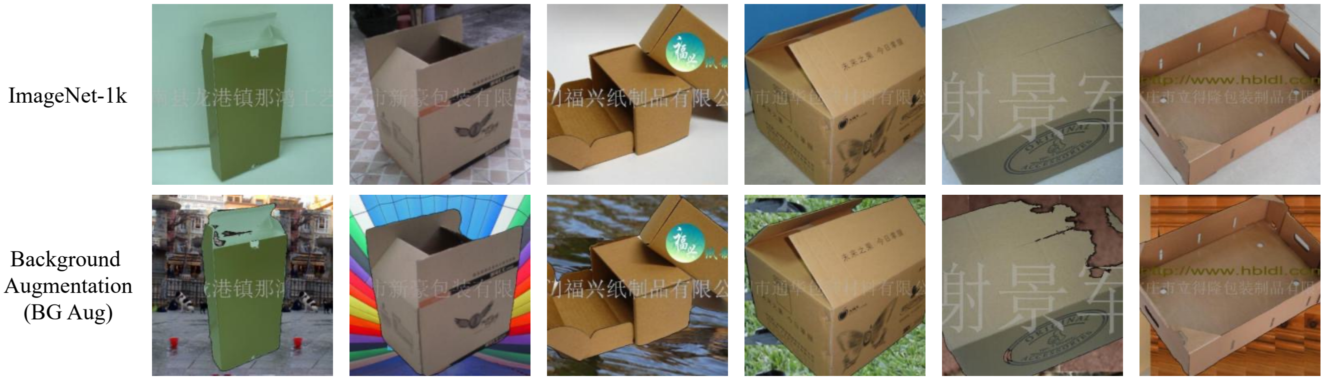


Figure 10. Examples of background augmentation (BG Aug) [71,91] on carton class images from ImageNet-1k training set. BG Aug is designed to mitigate the background shortcut. However, it preserves the watermarks, leading models to pivot to the watermark shortcut.

Style Transfer (TXT Aug) Preserves or Amplifies the Shape of Watermark In addition to Fig. 1b, we show more examples of style transfer [26] augmentation for carton class images with watermark in Fig. 9. While the technique was originally targeted at mitigating the texture shortcut by randomizing the texture information to increase the shape bias towards the object, the shape of the watermark shortcut, as shown in Fig. 9, is preserved or even amplified. Watermarks in large font sizes (*cf*. first three images in Fig. 9) are still legible after style transfer. The pattern of watermarks in small font size is still retained or even more salient, *e.g*., the pattern of the transparent watermarks becomes more salient after style transfer when the background is white. This can explain why style transfer (*i.e*., TXT Aug) amplifies the watermark shortcut results in Tabs. 4 and 13.

Background Augmentation (BG Aug) Preserves the Watermark Shortcut Besides Fig. 1b, we show more examples of background augmentation (BG Aug) [71,91] preserving the watermark shortcut in Fig. 10. Since the watermark is located over the main object, watermarks are still visible when replacing the background with a random one, which explains why BG Aug amplifies the watermark shortcut in Tab. 4. More recently, RobustViT [12] uses the object mask to regularize the model to focus on the object region in the objective function, aiming to mitigate the background shortcut. Although it does not use masks to modify the input image as BG Aug does, we show that it also amplifies the watermark shortcut in Tab. 13 (*cf*. Appendix E.1), which can be explained by the shared spatial locations between watermark and carton object.

E. More Results of Multi-Shortcut Mitigation on ImageNet

E.1. Benchmark More Existing Approaches

End-to-End Training In Sec. 5.2 and Tab. 4, we benchmark existing methods using last layer re-training [45]. Here we show the results of those methods (*i.e*., Mixup, Cutout, CutMix, AugMix, SD, Style Transfer, LfF, JTT, EIIL, DebiAN) using end-to-end training in Tab. 13. We show that most of them still exhibit the Whac-A-Mole problem by achieving worse shortcut mitigation results. Although Mixup does not amplify shortcuts, its improvement over ERM is still small.

Big Transfer (BiT) We also show the results of Big Transfer (BiT-M) [48], a foundation model pretrained on ImageNet-21k (*i.e*., excluding 1k classes of ImageNet-1k from the full ImageNet with 22k classes) using ResNet-50v2 [31] architecture. Tab. 13 shows that BiT-M achieves a larger SIN Gap than ERM and barely mitigates the background shortcut.

RobustViT Mitigates Background Shortcut but Amplifies Other Shortcuts RobustViT [12] is a recent work designed to mitigate the background shortcut by optimizing the relevance map against the object mask. The results in Tab. 13 show that it mitigates the background shortcut but amplifies the watermark shortcut. Besides, it also achieves a worse SIN Gap result for the texture shortcut.

E.2. Results: LLE Using Other Feature Extractors

We further show the results of models using the large ViT-H architecture in Tab. 14. We observed that there is no clear winner among these methods for achieving the best mitigation results on all shortcuts. Our method (LLE) can improve shortcut

			shortcut reliance				
			Watermark		Texture		Background
		IN-1k	IN-W Gap ↑	Carton Gap ↓	SIN Gap ↑	IN-R Gap ↑	IN-9 Gap ↑
ERM	ResNet-50	76.13	-26.64	+40	-69.03	-55.96	-5.53
Mixup	ResNet-50	76.11	**-12.30**	+38	-66.81	-53.03	**-5.06**
CutMix	ResNet-50	78.58	-19.50	**+22**	-72.86 (×1.06 🐹)	-58.51 (×1.05 🐹)	-6.25 (×1.13 🐹)
Cutout	ResNet-50	77.06	-16.29	+32	-69.95 (×1.01 🐹)	-57.32 (×1.02 🐹)	-5.90 (×1.07 🐹)
AugMix	ResNet-50	77.53	-16.76	+36	-66.38	-51.83	-6.42 (×1.16 🐹)
SD	ResNet-50	70.19	-16.12	+30	-63.63	-59.32 (×1.06 🐹)	-10.89 (×1.97 🐹)
Style Transfer (Texture 🔨)	ResNet-50	60.18	-17.31	+52 (×1.30 🐹)	**-4.32**	**-40.76**	-7.81 (×1.41 🐹)
LfF	ResNet-50	70.26	-17.57	+40	-64.34	-56.54 (×1.01 🐹)	-8.10 (×1.46 🐹)
JTT	ResNet-50	75.64	-15.74	+32	-69.04	-55.70	-6.75 (×1.22 🐹)
EIIL	ResNet-50	65.42	-19.71	+42 (×1.05 🐹)	-61.27	-57.43 (×1.03 🐹)	-8.66 (×1.57 🐹)
DebiAN	ResNet-50	74.05	-20.00	+30	-67.54	-56.70 (×1.01 🐹)	-7.29 (×1.32 🐹)
BiT-M (IN-21k)	ResNet-50v2	82.32	-8.63	+28	-73.69 (×1.07 🐹)	-51.19	-5.25
ERM	ViT-B/16	81.07	**-6.69**	**+26**	**-62.67**	-50.36	-5.36
RobustViT (Background 🔨)	ViT-B/16	80.33	-7.35 (×1.10 🐹)	+30 (×1.15 🐹)	-64.06 (×1.02 🐹)	**-45.64**	**-5.01**

Table 13. More multi-shortcut mitigation results on ImageNet. Note that methods from ERM to DebiAN use end-to-end training, which is different from the last layer re-training setting in Tab. 4. BiT-M is a foundation model pretrained on ImageNet-21k (IN-21k). RobustViT fine-tunes an ERM to mitigate the background shortcut.

			shortcut reliance				
			Watermark		Texture		Background
	train data	IN-1k	IN-W Gap	Carton Gap	SIN Gap	IN-R Gap	IN-9 Gap
SWAG (LP)	IG-3.6B	85.74	-4.89	**+8**	-59.99	-8.80	-7.86
SWAG (FT)	IG-3.6B	88.54	-3.09	+18	-62.22	-9.37	-3.19
CLIP (zero-shot)	LAION-2B	77.90	-3.61	+16	-59.47	**-5.61**	-3.71
MAE (FT)	IN-1k	86.89	-3.48	+30	-62.29	-33.15	-3.24
MAE+**LLE (ours)**	IN-1k	86.84	**-1.11**	+28	**-55.69**	-30.95	**-2.35**

Table 14. Multi-shortcut mitigation results on ImageNet with ViT-H network architecture. LP and FT stand for linear probing and fine-tuning on ImageNet-1k, respectively. Note that there is no ERM (supervised training) available with ViT-H on ImageNet-1k.

mitigation results over MAE in all metrics. Our method can even beat methods using extra pretraining data (*i.e.*, SWAG and CLIP) in IN-W Gap, SIN Gap, and IN-9 Gap.

Besides, we also show the results of LLE using SWAG (FT) in ViT-B/16 architecture in Tab. 15. While SWAG (LP) and SWAG (FT) suffer the Whac-A-Mole dilemma, LLE consistently mitigates multiple shortcuts jointly over ERM and SWAG (FT). Besides, we also show SWAG (FT) + LLE with edge augmentation (Edge Aug) and the results on ImageNet-Sketch. More details are introduced below (*cf*. Appendix E.3).

E.3. Results of LLE on ImageNet-Sketch

Results: ImageNet-Sketch We further show the results of LLE on ImageNet-Sketch [85] (IN-Sketch), another OOD variant of ImageNet containing sketch images in 1000 ImageNet classes. We use IN-Sketch Gap, the accuracy drop from IN-1k to IN-Sketch, to measure mitigation of color and texture shortcuts. The results in Tab. 15 show that our LLE method consistently improves the results over ERM, MAE, and SWAG (FT).

Edge Augmentation While style transfer augmentation could be suboptimal for mitigating the color and texture shortcuts measured by IN-Sketch, we propose edge augmentation (Edge Aug) to improve the results further. Concretely, we use [65] to detect edges on images from ImageNet-1k training set. The examples are shown in Fig. 11, where we observe that color and texture information is successfully removed via edge detection. Similar to style transfer and background augmentation (*cf*. Fig. 1b), we still observe the amplified or preserved saliency of the watermark (*cf*. carton class image in Fig. 11). The edge augmentation is used to train an additional last layer in the classifier ensemble. The results in Tab. 15 show that using

			shortcut reliance					
			Watermark		Texture		Background	Color and Texture
	train data	IN-1k	IN-W Gap	Carton Gap	SIN Gap	IN-R Gap	IN-9 Gap	IN-Sketch Gap
arch: ResNet-50								
ERM	IN-1k	76.39	-25.40	+30	-69.43	-56.22	-5.19	-52.32
LLE (ours)	IN-1k	76.25	-6.18	**+10**	**-61.02**	-54.89	**-3.82**	-51.56
LLE (ours) + Edge Aug	IN-1k	76.24	**-6.18**	**+10**	-61.52	**-53.69**	-3.95	**-48.25**
arch: ViT-B/16								
ERM	IN-1k	81.07	-6.69	+26	-62.60	-50.36	-5.36	-51.67
SWAG (LP)	IG-3.6B	81.89	-7.76 (×1.16)	+18	-67.33 (×1.08)	**-19.79**	-10.39 (×1.94)	**-32.22**
SWAG (FT)	IG-3.6B	85.29	-5.43	+24	-66.99 (×1.07)	-29.55	-4.44	-42.58
SWAG (FT) + **LLE (ours)**	IG-3.6B	85.37	-2.50	**+8**	**-60.92**	-28.37	**-3.19**	-41.52
SWAG (FT) + **LLE (ours)** + Edge Aug	IG-3.6B	85.31	**-2.48**	+12	-61.24	-27.78	-3.28	-38.37
MAE (FT)	IN-1k	83.72	-4.60	+24	-65.20 (×1.04)	-47.10	-4.45	-47.77
MAE + **LLE (ours)**	IN-1k	83.68	**-2.48**	**+6**	**-58.78**	-44.96	**-3.70**	-46.70
MAE + **LLE (ours)** + Edge Aug	IN-1k	83.69	-2.54	**+6**	-59.04	**-43.97**	**-3.70**	**-43.17**
arch: ViT-L/16								
ERM	IN-1k	79.65	-6.14	+34	-61.43	-53.17	-6.50	-52.40
MAE (FT)	IN-1k	85.95	-4.36	+22	-62.48 (×1.02)	-36.46	-3.53	-40.29
MAE + **LLE (ours)**	IN-1k	85.84	**-1.74**	**+12**	**-56.32**	-34.64	**-2.77**	-39.14
MAE + **LLE (ours)** + Edge Aug	IN-1k	85.84	-1.76	+16	-56.52	**-33.76**	-2.94	**-36.45**

Table 15. Ablation study of adding edge augmentation (Edge Aug) to LLE. Edge Aug further improves the results on ImageNet-Sketch.

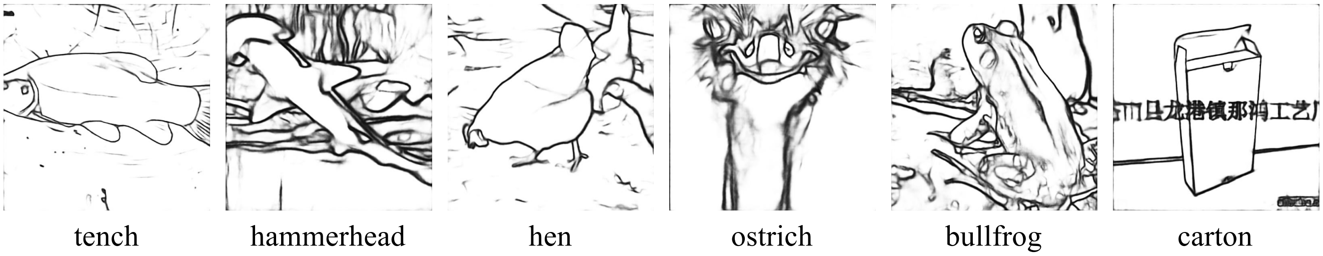

tench hammerhead hen ostrich bullfrog carton

Figure 11. Example images of edge augmentation for ImageNet-1k training set to mitigate color and texture shortcuts. The ground-truth class name is shown below each image.

Edge Aug can further close the In-Sketch Gap and IN-R Gap—IN-R also contains sketch images. The results demonstrate the effectiveness of designing targeted augmentation to tackle the known type of shortcut.

E.4. Top-1 Accuracy of LLE on OOD Variant of ImageNet

In this work, we mainly use the gap of accuracy between IN-1k to OOD variants of ImageNet as the metric. We also show the results of LLE in top-1 accuracy on OOD variants of ImageNet in Tab. 16, which can help future research to compare with LLE in top-1 accuracy.

Note that we do not include the top-1 accuracy on ImageNet-W. Although existing models suffer a performance drop from IN-1k to IN-W, an increased IN-W accuracy over IN-1k, which future works may achieve, also indicates the watermark shortcut reliance. Because of the counterfactual nature between IN-1k and IN-W, we encourage future works to use IN-W Gap and Carton Gap to report the watermark shortcut mitigation results, where closer to zero gaps indicate better results.

E.5. Results of LLE on Other OOD Variants of ImageNet

We also show the results of LLE on other OOD variants of ImageNet, including ImageNet-A [36] (IN-A), ImageNetV2 [68] (IN-V2), ObjectNet [9], and ImageNet-D [69,70] (IN-D). IN-D has rendition images similar to IN-R except for having additional domain annotations, *e.g.*, clipart, infograph, *etc*. Besides, IN-D also has real-domain images (*i.e.*, IN-D real). We report the top-1 accuracy on IN-A, IN-V2, ObjectNet, and IN-D clipart to IN-D sketch. Regarding the types of shortcut reliance, ObjectNet measures the robustness against unusual background, viewpoint, and rotation. The results from IN-D clipart to IN-D sketch measure the robustness against the texture shortcut. The remaining results, *i.e.*, IN-A, IN-V2, IN-D real, do not explicitly measure the robustness against specific shortcuts. Therefore, we denote their shortcut reliance type as “unknown.”

				shortcut reliance			
				Texture		Background	Color and Texture
	arch	train data	IN-1k	SIN	IN-R	Mixed-Rand	IN-Sketch
LLE	ResNet-50	IN-1k	76.25	15.25	37.31	84.40	24.67
LLE + Edge Aug	ResNet-50	IN-1k	76.24	14.72	38.43	84.30	27.99
SWAG (FT) + LLE	ViT-B/16	IG-3.6B	85.37	24.45	68.14	90.12	43.85
SWAG (FT) + LLE + Edge Aug	ViT-B/16	IG-3.6B	85.31	24.07	68.70	89.98	46.94
MAE + LLE	ViT-B/16	IN-1k	83.68	24.90	50.84	89.41	36.98
MAE + LLE + Edge Aug	ViT-B/16	IN-1k	83.69	24.65	51.85	89.36	40.52
MAE + LLE	ViT-L/16	IN-1k	85.84	29.52	62.24	91.58	46.70
MAE + LLE + Edge Aug	ViT-L/16	IN-1k	85.84	29.32	63.13	91.41	49.39
MAE + LLE	ViT-H/14	IN-1k	86.84	31.15	66.21	93.01	50.60
MAE + LLE + Edge Aug	ViT-H/14	IN-1k	86.84	30.94	66.89	92.86	53.39

Table 16. Top-1 accuracy results of Last Layer Ensemble (LLE) on OOD variants of ImageNet.

			shortcut reliance									
			unknown		background, viewpoint, rotation	texture					unknown	
	arch	(pre)training data	IN-A	IN-V2	ObjectNet	IN-D clipart	IN-D infograph	IN-D painting	IN-D quickdraw	IN-D sketch	IN-D real	IN-D (mDE) ↓
ERM	ResNet-50	IN-1k	0.02	**63.48**	36.10	23.94	10.69	34.83	0.83	17.77	59.86	88.27
LLE	ResNet-50	IN-1k	**0.12**	63.34	**36.67**	25.86	**11.35**	**36.86**	0.85	19.57	**60.60**	86.79
LLE + Edge Aug	ResNet-50	IN-1k	0.09	63.05	**36.67**	**26.31**	11.29	36.82	**0.92**	**20.72**	60.57	**86.50**
ERM	ViT-B/16	IN-1k	20.88	69.56	39.89	29.87	13.62	41.37	1.13	21.86	62.75	83.53
SWAG (FT)	ViT-B/16	IG-3.6B	53.01	75.58	53.90	49.54	20.09	52.88	2.53	39.34	68.17	70.99
SWAG (FT) + LLE	ViT-B/16	IG-3.6B	53.71	**75.75**	54.48	51.18	**21.63**	54.88	3.19	41.09	69.12	69.25
SWAG (FT) + LLE + Edge Aug	ViT-B/16	IG-3.6B	**53.75**	75.68	**54.55**	**51.69**	21.43	**54.93**	**3.59**	**41.95**	**69.20**	**68.93**
MAE (FT)	ViT-B/16	IN-1k	35.81	**73.20**	47.30	34.11	15.27	44.30	1.17	27.14	64.92	80.15
MAE (FT) + LLE	ViT-B/16	IN-1k	36.88	73.06	47.63	35.25	**16.37**	45.90	1.25	28.66	65.47	78.93
MAE (FT) + LLE + Edge Aug	ViT-B/16	IN-1k	**37.00**	72.94	**47.79**	**35.73**	16.10	**45.97**	**1.34**	**29.65**	**65.52**	**78.66**
ERM	ViT-L/16	IN-1k	16.64	67.49	36.79	27.68	12.45	39.47	0.58	19.40	62.04	85.32
MAE (FT)	ViT-L/16	IN-1k	**57.07**	76.65	55.31	42.64	18.05	50.14	3.12	36.87	66.66	74.10
MAE (FT) + LLE	ViT-L/16	IN-1k	56.65	**76.74**	55.46	43.95	**19.31**	51.67	3.27	38.05	**67.29**	72.87
MAE (FT) + LLE + Edge Aug	ViT-L/16	IN-1k	56.77	76.66	**55.65**	**44.24**	19.06	**51.81**	**3.44**	**38.88**	**67.29**	**72.65**
MAE (FT)	ViT-H/14	IN-1k	68.17	**78.46**	60.47	43.69	19.10	51.29	3.89	39.17	67.61	72.63
MAE (FT) + LLE	ViT-H/14	IN-1k	68.27	78.34	60.61	45.40	**20.80**	52.94	4.24	40.75	68.20	71.12
MAE (FT) + LLE + Edge Aug	ViT-H/14	IN-1k	**68.35**	78.32	**60.78**	**45.76**	20.66	**53.06**	**4.40**	**41.60**	**68.23**	**70.86**

Table 17. Results of LLE on other OOD variants of ImageNet, *i.e.*, ImageNet-A (IN-A), ImageNetV2 (IN-V2), ObjectNet, and ImageNet-D (IN-D). Except for IN-D overall results (*i.e.*, last column), all other results are in top-1 accuracy. The overall IN-D results are reported in mDE, where lower numbers indicate better results (↓).

The results are shown in Tab. 17. On both ObjectNet and IN-D datasets, LLE consistently improves the results over various baselines (*i.e.*, ERM, SWAG (FT), and MAE (FT)) in different network architectures. When the shortcut type is unknown, LLE achieves comparable results against the baselines with slight performance improvement or drop depending on the architectures and pretraining datasets. Note that LLE is designed for mitigating multiple *known* shortcuts (*cf*. Sec. 4). Therefore, it may not improve the results when the types of shortcuts remain unknown. However, due to the theoretical impossibility of inferring shortcut labels [57] and the practical difficulty of mitigating multiple unknown shortcuts, we encourage future research to tackle this problem by first interpreting the distributional shift on IN-A or IN-V2 before performing mitigation (more discussion in Appendix G).

F. CutMix Amplifies Background Shortcut

Results of CutMix on Waterbirds On UrbanCars (*cf*. Tab. 5) and ImageNet (*cf*. Tabs. 4 and 13), we observe that CutMix [92] amplifies the background shortcut. We further show its background shortcut reliance on Waterbirds dataset. We use the following metrics on Waterbirds: (1) Average Group Accuracy: the unweighted average results over four groups ($\{\text{waterbird}, \text{landbird}\} \times \{\text{water background}, \text{land background}\}$); (2) Worst Group Accuracy: the lowest per group accuracy result. For this experiment on Waterbirds, we use the experiment setting on UrbanCars (*cf*. Appendix B.3). Tab. 18 shows that CutMix achieves worse results of mitigating the background shortcut than ERM. Other techniques, *i.e.*, Mixup and Cutout, slightly mitigates background shortcut on Waterbirds.

	Average Group Accuracy (%)	Worst Group Accuracy (%)
ERM	87.19	73.88
Mixup ($\alpha = 0.05$)	87.76	75.73
Cutout ($p = 0.1$)	88.57	74.87
CutMix ($\alpha = 1.0$)	74.51 (-12.68)	47.38 (-26.50)

Table 18. Results of standard augmentation and regularization on Waterbirds dataset. CutMix amplifies the background shortcut on Waterbirds [72] dataset. $(\cdot)$: hyperparameter used in each approach.

Explaining the Background Shortcut Reliance of CutMix Since CutMix consistently amplifies the background shortcut on three datasets (*i.e.*, UrbanCars, Waterbirds, and ImageNet), we take a closer look at its augmentation and regularization strategy. In terms of augmentation, CutMix crops a rectangular patch from one image and pastes it to the other to create the augmented image. In the regularization, the ground-truth label for the augmented image is the linear interpolation of ground-truth labels of two source images, where the interpolation co-efficient (*i.e.*, called combination ratio λ in CutMix) is proportional to the area of the patch. In this way, the network is regularized to predict the likelihood over classes that is proportional to the area in the image. Therefore, when the background takes the larger area in the image, the model predicts more on the background class instead of the smaller foreground object, leading to an amplified background shortcut reliance.

G. Discussion

We discuss the recommendations and limitations based on our multi-shortcut benchmarking results.

G.1. Recommendation and Future Direction

To future shortcut mitigation practitioners, we recommend the community drop the unrealistic single-shortcut assumption and be aware of the multiple-shortcut problem by having a sanity check on various inductive biases in model design, such as the usage of shortcut labels, assumption of shortcut learning during training, data augmentation, regularization, *etc*.

For future shortcut mitigation dataset creators, a broader range of factors of variations (FoV) needs to be studied since some FoVs could serve as multiple shortcuts learned by models. This can be achieved by (1) manually choosing various FoVs under the controlled setting [9,22,37,39,51,75] or (2) developing better approaches to detect and interpret shortcuts [2,6,18,23,42,54, 81] on in-the-wild datasets.

Although our work mainly focuses on the shortcut mitigation task, the importance and challenge of multiple shortcuts also apply to the shortcut detection task. For example, Eyuboglu *et al.* [23] design a shortcut detection benchmark based on CelebA, where only a single shortcut exists. Specifically, they achieve this by amplifying the correlation strength of the spurious correlation between the target attribute and the shortcut attribute. Therefore, whether or not existing shortcut detection approaches can detect multiple shortcuts is underexplored and is a promising future direction.

G.2. Limitations

Admittedly, our work has limitations. For example, Last Layer Ensemble (LLE) does not address the problem of unknown types of shortcuts, which LLE may amplify. However, since mitigating unknown types of shortcuts without any inductive biases is still a theoretical [57] and practical challenge, we advocate a human-in-the-loop solution. That is, detecting and interpreting shortcuts at the first stage. Then, LLE can be applied to mitigate the detected shortcuts.