Appendix for Enhancing Representation Learning with Deep Classifiers in Presence of Shortcut

Amirhossein Ahmadian and Fredrik Lindsten

A Additional implementation and hyperparameter details

What follows is additional details about the proposed method (*ARE*), for both CIFAR-10 and CelebA experiments. The learning rate used for the classifier network and lens network are 0.0002 and 0.002 respectively, and batch size is 128 (in the CIFAR-10 case, the size is 128×4 in effect due to the rotations). The quite small learning rate of the classifier helps to avoid oscillation and convergence problems. The lens has a larger learning rate since we observed that in practice it cannot keep up with the classifier during the adversarial training (game) if it has the same learning rate and number of iterations as the classifier (possibly due to the longer gradient backpropagation path compared to the classifier). The first 2 iterations (on mini-batches) of the training are considered as warm-up iterations. During these iterations, the lens is ignored, and the classifier is trained only on the real data. It should be emphasized that the ResNet model we train on the upstream data (for our method as well as the other compared ones) is not pre-trained on any dataset, and is initialized randomly.

In the vanilla baseline (ERM) method, the learning rate and batch size is equal to the values used for the classifier in our method, to rule out any effect of learning rate. However, we have also experimented with larger learning rates for this method in the next section. In general, when choosing the hyperparameters for the other compared methods, we try to use the values suggested in their papers or corresponding code when a similar dataset (CIFAR-10 or CelebA) has been used in those papers.

For the *Automatic Shortcut Removal* method, the learning rate is 0.0001 for both of the networks and the datasets, and the batch size is 128. In the arrow shortcut experiment, a suggested λ is already available since this problem is experimented in their paper as well. For the light gradient and CelebA experiments, we found the best λ by probing the downstream test accuracy with $\lambda \in \{1, 10, 160, 320, 640, 1280\}$.

For the *Spectral Decoupling* method, we followed the λ and learning rate values that is used in their paper/code. This means the learning rate 0.001 (with batch size 50) for CIFAR-10 and 0.0001 (with batch size 128) for CelebA experiments.

For the *Diverse Ensemble* method, the ResNet network is used as the shared feature extractor, where its pre-logits output is fed to each head. The heads are feedforward networks with 2 layers of 512 hidden units, ReLU activation functions, and softmax output, which is the same network used in their paper. The best value of λ was determined by probing the downstream test accuracy with $\log_{10} \lambda \in \{-1, 0, 1, 2, 3, 3.6\}$. The learning rate and batch size for this method are 0.0002 and 128.

The downstream classifier is always trained with ADAM optimizer, learning rate 0.001, and batch size 64. For the small training set mode of biased CelebA, the same configuration of each method is used which has resulted in the best performance in the CelebA experiment with full downstream training set.

B Learning curves

Figure 1 shows how the loss values of the lens and classifier networks change during training epochs of the proposed method. In addition, the accuracy of the classifier on real data is shown in this figure. Each point in these plots refers to the loss/accuracy values on the training set averaged over all iterations during the epoch. It should be noted that the classification accuracy on the real data is obtained with discarding the extra (fake class) output neuron of the classifier network.

We observe that the classifier loss can reach a relatively small value in earlier epochs, but that is followed by a maximum, and then it keeps going down. This behavior can be explained by taking into account the nature of adversarial training and the presence of shortcuts. In the early iterations, when the lens is still weak, it is easy for the classifier to detect the fake lens reconstructions. Shortly after, the lens manages to take advantage of the vulnerability of the classifier to shortcuts, and thus the classifier is fooled more easily. In the next phase, the classifier starts to learn the non-shortcut features to improve its performance in both detecting the fake images and classifying real data. The other important point is that the classifier achieves an almost perfect accuracy on real images (upstream classification accuracy). This is reasonable because the shortcuts still exist in the images, and therefore it is always easy to predict the true class. This also shows that the performance of the classifier on the upstream tasks is not impaired by the modifications made by our method, that is, the upstream classifier can be used in the task as usual afterwards, albeit by removing the extra output neuron (in all of the three experiments, the classifier accuracy on the test data ends at a value around 99% as well).

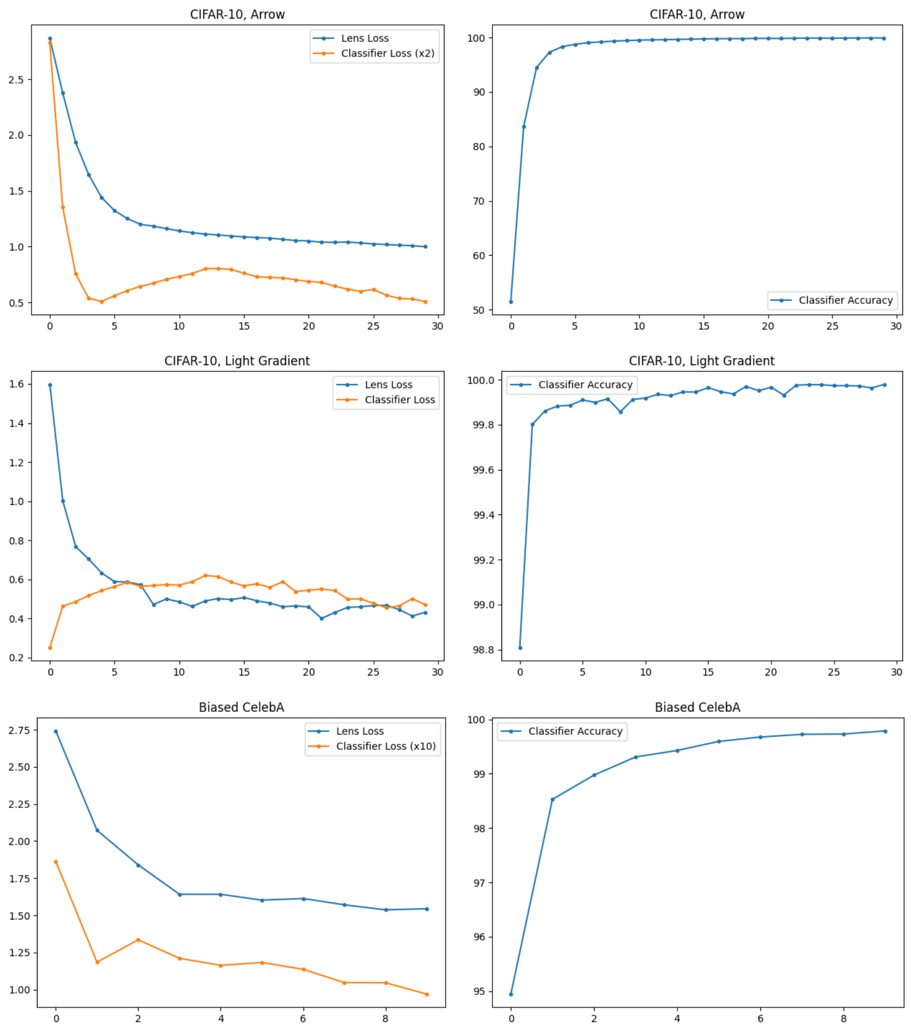


Figure 1: Loss curves of the classifier and lens, and accuracy in real images classification, in our proposed method. The values are per epoch and on training data.

C More on the vanilla method

Although changing learning rate is not reported as a solution for shortcut learning in previous studies, it is reasonable to ask whether increasing the learning rate beyond the order of 10^{-4} can improve the representations obtained by the basic vanilla method. Thus, we ran the vanilla experiments on CIFAR-10 arrow/gradient, and biased CelebA with learning rate 0.001. We find that on CIFAR-10 data, the method performs even worse than the small learning rate mode. But there is about 0.7% improvement on CelebA, which is still behind the best method in this problem.

Problem	Default LR (0.0002)	Larger LR (0.001)
CIFAR-10 Arrow	53.2	51.1
CIFAR-10 Light Gradient	45.5	43.9
Biased CelebA	85.6	86.3

Table 1: Downstream accuracy of the vanilla method with different learning rates.

It can be also interesting to compare the results of the vanilla method on shortcut biased data with the clean (without shortcut) case. As verified by the accuracy curves in figure 2, the classifier rapidly discovers the arrow or gradient shortcut for classifying the rotation during the early training epochs, and reaches a much higher performance than when it is trained on the normal CIFAR-10 images.

In terms of downstream performance, the vanilla method (with learning rate 0.001) leads to the accuracy 68.3% on the clean CIFAR-10 data, which is at least around 17% higher than the corresponding accuracy in the shortcut-biased modes of the problem. The downstream accuracy of our proposed method in this problem with clean data is 66.1% if we use the same hyperparameters as before. However, by increasing the learning rate of the classifier to 0.001 in this particular case, the obtained downstream accuracy of our method rises to 71.5%, that is again higher than the vanilla baseline within this experimental setup.

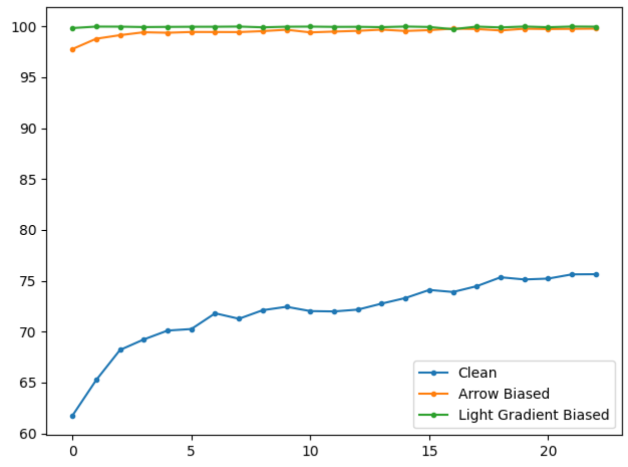


Figure 2: Accuracy of the basic vanilla classifier in rotation prediction (upstream test accuracy) versus training epochs, on different modes of the CIFAR-10 dataset.

D Lens reconstruction plots

In the following, some more examples of the images from the three studied problems, and their corresponding lens output obtained by our method, have been shown. In most of the light gradient reconstructions, a stripe with different colors is seen in the bottom of the image. This is perhaps because this part of the image is always zero in input data (contains no useful information for the classifier), which means the lens can replace it with any arbitrary value.

In the arrow shortcut case, the shortcut mark (arrow) is not very obvious in the second epoch reconstructions, unlike the case of light gradient, though it becomes more noticeable in the third epoch. One reason can be simply that the arrow is a small local artifact, whose effect can be simulated by modifying a few pixels in the image (similar to adversarial attacks). Moreover, according to the classifier accuracy curves in figure 1, in the arrow problem, the classifier seems to take the shortcut in later epochs compared to the light gradient. Nonetheless, the arrow appears very sharp in the later reconstructions shown in table 3.

For the biased CelebA problem, there are several interesting observations to comment on. First, we see that the effect of shortcut is amplified in some cases. Specifically, the black hair can be much expanded compared to the original image, such as the first row in table 4, and the color of blond hair might be spread over the image, such as the second row. In addition, we notice that even in the last epoch, although the reconstructed image looks close to a natural face overall, its details do not exactly match the original image.

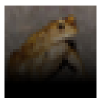
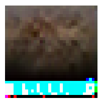
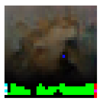
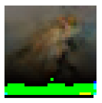
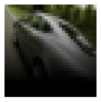
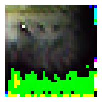
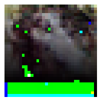
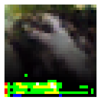
Input	Lens Reconstructions		
			
			

Table 2: CIFAR-10 Light Gradient Examples. Reconstructions (from left to right) obtained in epochs 2, 10, 20 .

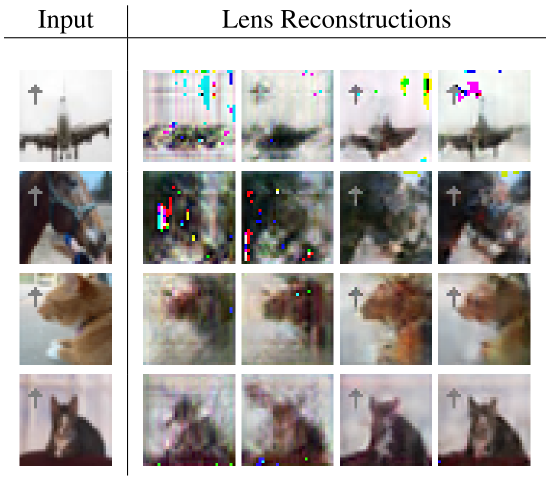


Table 3: CIFAR-10 Arrow Examples. Reconstructions (from left to right) obtained in epochs 2, 3, 10, 20 .

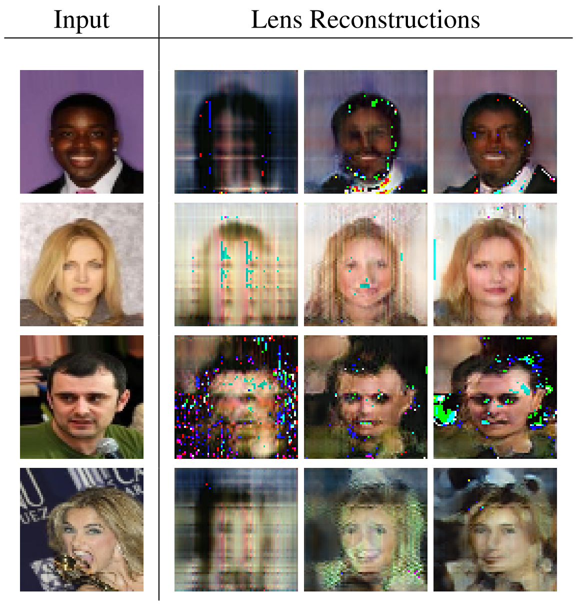


Table 4: Biased CelebA Examples. Reconstructions (from left to right) obtained in epochs 2, 5, 10 .

E More results of Automatic Shortcut Removal

Table 5 shows the effect of the lens network on images after many iterations, that is obtained by our implementation of the Automatic Shortcut Removal method. For the arrow shortcut problem, we see reconstructed images in which the arrow is almost completely removed as well as many cases where it is partially faded. More interestingly, in some cases (e.g., the rightmost example) an arrow is added to the image in a direction misleading for the classifier (which suggests the method is not merely a 'remover' of shortcuts but can also add them). With more training epochs, the ASR lens gets better at removing the shortcut gradually, as discussed in the paper, and supported by the performance reported in table 6.

For the light gradient biased images, we almost could not observe any obvious modifications by the lens with 30 training epochs. But after training for much longer (670 epochs), it looks as the lens makes the image brightness more uniform, mostly by modifying the bottom part of images.

The results on the CelebA faces are even more intuitive. In this case, the ASR lens clearly changes the hair color from black to blond for most of the men. This is often accompanied by changing some other attributes (e.g., eyes) towards a more feminine face (which is basically not intended in this problem as a shortcut removal). The effect on the female faces is less intense, although their hair color looks quite darker in many reconstructions.

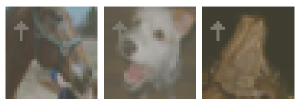
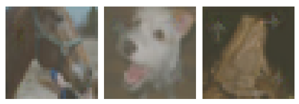
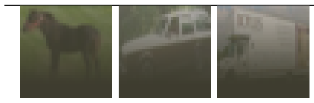
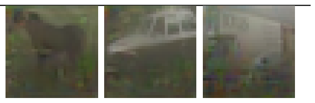
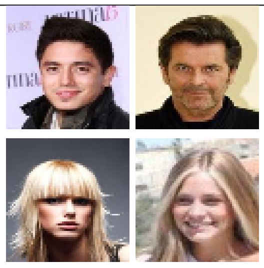
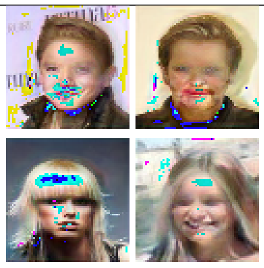
Inputs	ASR Lens Reconstructions
	
	
	

Table 5: Examples of reconstructions obtained by the lens in the Automatic Shortcut Removal method, near the end of training. The results are shown for the arrow biased, light gradient biased, and biased CelebA data, with 30, 670, and 10 training epochs respectively.

Training Epochs	Downstream Accuracy (%)
100	57.1
200	59.3
500	61.7
670	62.6

Table 6: Downstream performance of the ASR method in the arrow-biased CIFAR-10 problem after different numbers of training epochs