Self-Supervision on Images and Text Reduces Reliance on Visual Shortcut Features

Anil Palepu [1] **Andrew L Beam** [2]

Abstract

Deep learning models trained in a fully supervised manner have been shown to rely on so-called "shortcut" features. Shortcut features are inputs that are associated with the outcome of interest in the training data, but are either no longer associated or not present in testing or deployment settings. Here we provide experiments that show recent self-supervised models trained on images and text provide more robust image representations and reduce the model's reliance on visual shortcut features on a realistic medical imaging example. Additionally, we find that these self-supervised models "forget" shortcut features more quickly than fully supervised ones when fine-tuned on labeled data. Though not a complete solution, our experiments provide compelling evidence that self-supervised models trained on images and text provide some resilience to visual shortcut features.

1. Introduction

A *shortcut feature* is a recently introduced term to describe the phenomenon of a machine learning model relying on *unstable*, *spurious*, or otherwise unreliable model inputs. Shortcut features have a potentially strong association with the label in the training data, but this association is broken or potentially reversed in testing or deployment environments. Moreover, shortcut features are often non-causal features that humans would not identify as being important to the prediction task. A provocative example was provided in the context of medical imaging by Geirhos et al. (2020). In this example, a deep learning model trained to detect pneumonia in chest X-rays (CXRs) relied on *watermarks* indicating the hospital where the patient was seen instead of lung pathophysiology as a radiologist would. This reliance is a result of the watermarks containing *statistically* relevant information since each hospital treats unique patient populations with different baseline risks for pneumonia (*e.g.*, acute care vs. ambulatory settings). However, this association is broken if deployed to hospitals with new watermarks or hospitals that do not use watermarks at all. Shortcut features thus pose a serious challenge to the safe deployment of these models in high-stakes medical settings.

In this work, we examine the robustness of deep learning models to watermark shortcuts on CXRs. We assess the sensitivity of traditional, fully supervised models commonly used in the medical literature to that of recently introduced self-supervised models trained jointly on images and text. Specifically, we consider text-image alignment models such as CLIP (Radford et al., 2021; Zhang et al., 2020), since we hypothesized that this alignment might result in the model ignoring visual features that lacked a corresponding text anchor.

2. Data

2.1. Datasets

We used the MIMIC-CXR-JPG (Johnson et al., 2019) and CheXpert (Irvin et al., 2019) datasets, which consist of 227,835 and 224,316 chest radiographs respectively, each with clinical labels for Atelectasis, Cardiomegaly, Consolidation, Edema, and Pleural Effusion. The MIMIC-CXR-JPG dataset, which we utilized for model training, also consists of free text radiology reports corresponding to each CXR. We randomly selected 1% of the CheXpert training set to be used for model fine-tuning. The set of 234 CheXpert CXRs that were labeled by the consensus of three radiologists was chosen as our test set. (Irvin et al., 2019)

2.2. Data Preprocessing

At train time, a series of transformations were applied to the images: random cropping, random horizontal flipping, random affine transformations, color jitter, and gaussian

[1]Department of Health Sciences and Technology, Harvard-MIT, Cambridge, MA, USA [2]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA; CAUSALab, Harvard T.H. Chan School of Public Health, Boston, MA, USA; Department of Epidemiology, Harvard T.H. Chan School of Public Health, Boston, MA, USA. Correspondence to: Anil Palepu <apalepu@mit.edu>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

blur. Using these transformations, two data augmentations were produced for each input image. At validation and test time, these transformations were skipped, and the images were instead simply center-cropped to 224 by 224 pixels.

When available, the "Findings" and "Impressions" sections from the MIMIC-CXR-JPG radiology reports were extracted and concatenated to produce shortened clinical text. When these sections were not available, we attempted to extract the "Comparisons" section instead, and in the rare case when that was also missing, we simply extracted the original radiology report and clipped it to 100 words.

2.3. Injection of Synthetic Shortcuts Using Watermarks

The original CXRs were corrupted by synthetic shortcuts by adding label-associated watermarks. A unique symbol was assigned to each of the five labels, and at train time, each image was selected to be corrupted by shortcut features with a probability of 0.9. If selected to be corrupted, the image was given "correct" shortcuts with a probability of 0.9, meaning all positive label symbols were watermarked onto the image. Otherwise (with a probability of 0.1), the corrupted image was given "incorrect" shortcuts and all negative label symbols were watermarked onto the image.

Two test sets were generated from the base CheXpert test set for each label. The *Shortcut* test sets had the positive label-specific symbol watermarked onto all label-*positive* images, while the *Adversarial* test sets had the positive label-specific symbol watermarked onto all label-*negative* images. As a result, if a model is overly reliant on the watermarked shortcut features, it should have very high accuracy on the *Shortcut* test sets and very low accuracy on the *Adversarial* test sets. Furthermore, such a model could also perform poorly on the original, uncorrupted CXRs.

3. Model Details

3.1. Contrastive Language-Image Pretraining

We adopted the CLIP architecture (Radford et al., 2021) to jointly train a vision and text encoder on image-text pairs from MIMIC-CXR-JPG. For our vision encoder, we used a ResNet-50 (Ren et al., 2016) backbone that was pretrained on ImageNet (Deng et al., 2009), and for our text encoder, we used Bio_ClinicalBERT (Alsentzer et al., 2019), a text transformer pretrained on clinical discharge summaries. A dense layer of size 512 was added as the output to each encoder. The first eight layers of the text encoder were frozen during training, and the vision model was fully unfrozen.

To train the model, the image and text samples were processed by the encoders to produce their respective embeddings. After computing the cosine similarities between each possible image and text embedding combination, the contrastive loss was computed as the cross entropy between these similarities and the idealized perfect matching of positive image-text pairs. Because we produced two image augmentations from each image, we summed the contrastive loss between each augmentation and the text, as well as between the augmentations themselves (Li et al., 2021). We trained for 50 epochs with a learning rate of 0.0001 and batch size of 32, saving the model version that minimized validation loss.

Using the above methodology, we trained one version of the CLIP model on original MIMIC_CXR images and another on the synthetic data containing watermark-based shortcuts.

3.2. CNN Training and Fine-tuning

For our CNN models, we used the same ResNet-50 architecture as the self-supervised model, except we added an output layer with five classification heads corresponding to each of the five clinical labels. To train these models, we computed and summed the binary cross-entropy loss for each label with the same hyperparameters as the self-supervised model. We again trained one version of this CNN model on the original MIMIC_CXR images and one version on the shortcut-corrupted images.

Afterwards, we fine-tuned each of the models using a randomly selected one percent of CheXpert's training set. This fine-tuning was done on uncorrupted CheXpert radiographs for both CNN models and both CLIP models. For the CNNs, this only required training the models further on this new dataset. For the CLIP models, we detached the vision encoder, added a classification layer with 5 heads (thus constructing an identical architecture to the CNN models), and then trained the models. Using this small subset of CheXpert data, these four models were trained for an additional 100 epochs with a 10-fold reduced learning rate.

4. Model Evaluation

4.1. Zero-shot classification

The vision encoders of the CLIP models were not directly capable of classification, so to produce predictions from these models, we leveraged the text encoder. For each label, we selected a set of 50 MIMIC_CXR reports from samples that were exclusively positive for that label, processed these reports through the text encoder, and averaged the resulting embeddings to create a label-specific embedding. Our zero-shot predictions were determined by computing the cosine similarity of the input image embedding to these label-specific embeddings.

Because cosine similarity is in the range -1 to 1, these predictions could not be interpreted as probabilities. Nevertheless, they could still be utilized to evaluate model discrimination

for each label by computing areas under the curve (AUC). As seen in Table 1 of the appendix, these zero-shot CLIP classifiers did not achieve the efficacy of the supervised CNN models. However, given the lack of task-specific training, they demonstrated impressive predictive power and were more robust to shortcut features present in the shortcut and adversarial test sets than the fully supervised CNNs.

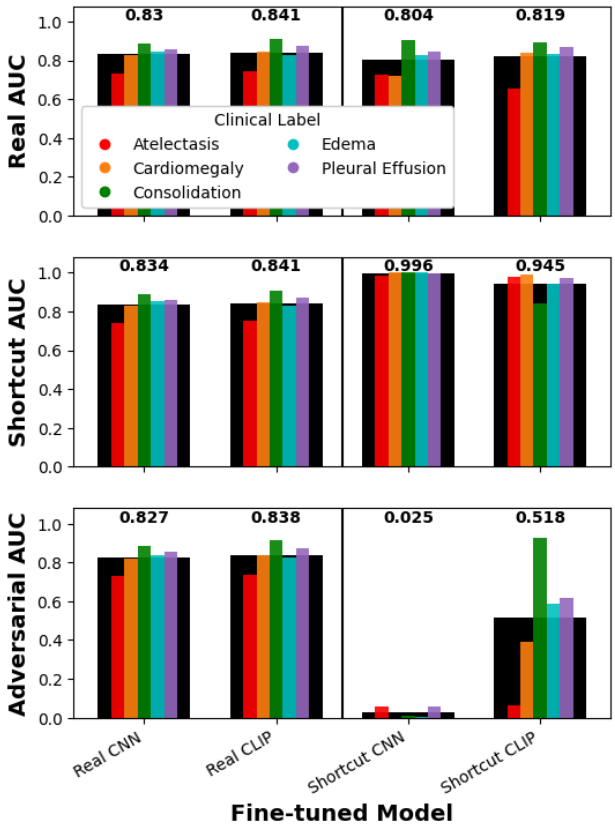

Figure 1. AUC results for the four fine-tuned models on our test sets. These are directly comparable, as the fine-tuned models have identical architectures. The top panel is on the original CXRs, the middle panel is with shortcuts added to the positive label samples, and the bottom panel is with shortcuts added to the negative label samples. These AUCs were computed for each of the five clinical labels, which were averaged to compute the black bars.

4.2. Classification Performance

After fine-tuning, we had four identical CNN architectures that differed only in the initial training strategy (CNN vs CLIP) and the presence of shortcuts in the training data (Real vs Shortcut). We evaluated each of these models by computing the AUC for the five clinical label predictions on the original, shortcut, and adversarial CheXpert test sets. These results can be seen in Figure 1.

As expected, the models that were trained on real data were largely unaffected by the presence of watermarks in the testing data. In terms of AUC, the fine-tuned CLIP models performed comparably to or slightly better than the fine-tuned CNN models.

The models that were trained on shortcut data performed slightly worse on real data than their uncorrupted counterparts. On the label-associated shortcut test data, the CLIP and CNN models both saw a drastic rise in performance, while on adversarial test data, they both saw a drastic drop in performance. This relationship was especially striking in the CNN model, which achieved near-perfect discrimination (AUC=0.996) on associated shortcuts, and completely failed (AUC=0.025) on adversarial shortcuts. While the CLIP model did learn to use these shortcuts, our results suggest it was relatively less reliant on them and was often able to make predictions in conflict with the shortcuts present.

4.3. Integrated Gradient Maps

Integrated gradients are a model attribution method that we leveraged to visualize the features being used by our models (Sundararajan et al., 2017). The integrated gradients were computed by generating a linear interpolation between a black baseline and the input and accumulating the gradients of the output with respect to these interpolated inputs. The integrated gradient maps on all test images were computed for the models prior to fine-tuning. To do so, we used Google's PAIR saliency package to implement *Integrated Gradient + SmoothGrad*. For each input image, *SmoothGrad* produces several noisy copies of that image and then averages the resulting integrated gradients from each copy (Smilkov et al., 2017).

A key axiom that integrated gradients are designed to fulfill is "implementation invariance," meaning that functionally equivalent networks should have identical attributions (Sundararajan et al., 2017). As seen in Figure 2(b), the fact that the CNN models had poor invariance to training and testing data suggests that these models were far from functionally equivalent. For a CNN model trained on shortcuts, the presence of shortcuts drastically altered that model's behavior. In particular, as demonstrated qualitatively by Figure 2(a), the shortcut-trained CNN appeared to exclusively utilize the watermarked shortcuts (when available) to make its predictions.

On the other hand, the CLIP model exhibited far more consistency. The shortcut-trained CLIP had highly similar integrated gradients regardless of whether or not a shortcut was present at test time. Additionally, its integrated gradients were quite similar to those produced by its real-trained counterpart, indicating that these models were functionally similar despite one being trained on shortcut-corrupted data.

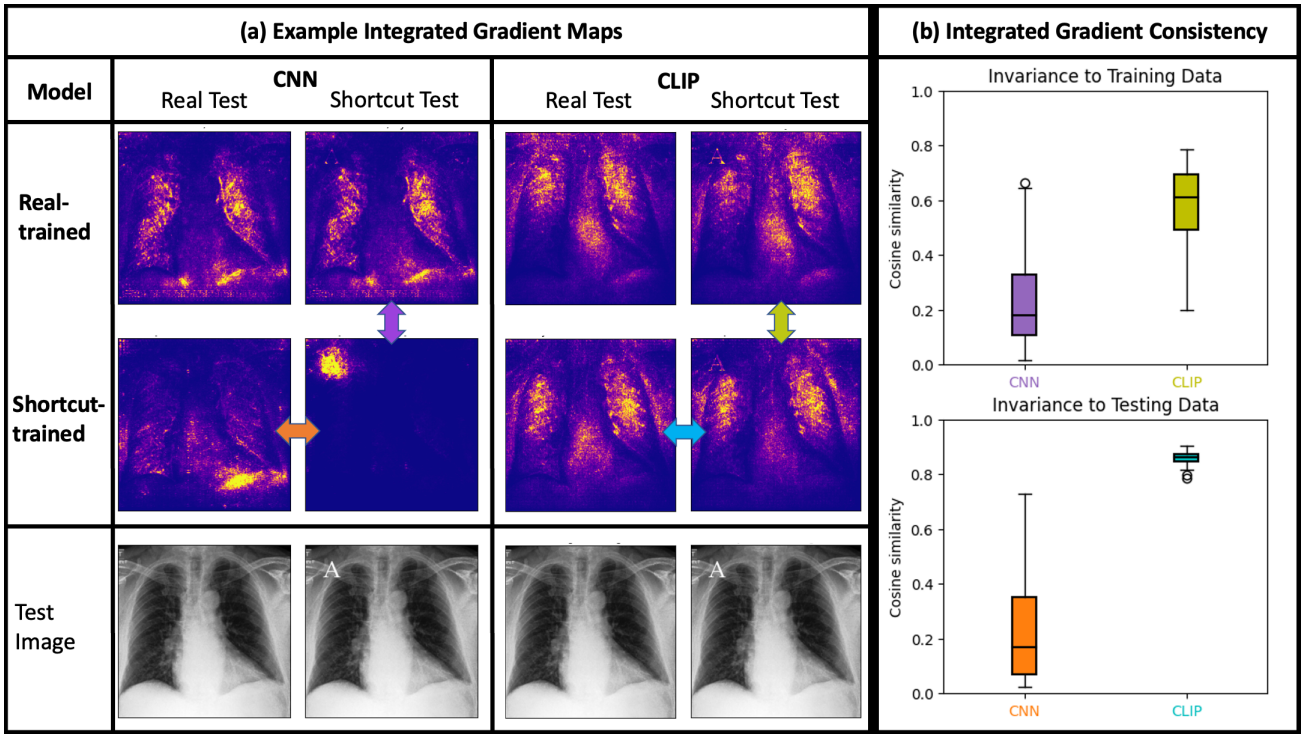

Figure 2. (a) On the left is an example of the integrated gradient maps for Atelectasis prediction using the CNN models prior to fine-tuning. On the right are the corresponding maps using the CLIP models. The colored arrows correspond to the comparisons made in Figure 2(b). (b) After computing the maps in (a) for all CheXpert test images, we computed the pixel-level cosine similarities between maps from the real-trained and shortcut-trained models (top plot) as well as between real and shortcut test images (bottom plot). A higher cosine similarity indicates more consistency between the integrated gradient maps. These plots demonstrate that the CLIP models are more consistent regardless of the shortcuts present at train or test time.

5. Discussion

5.1. Conclusions

Our results suggest that the natural language supervision provided by the CLIP architecture reduced model reliance on shortcut features. Unlike the CLIP model, the CNN trained on label-associated shortcuts was completely impaired by adversarially manipulating these shortcuts, and was unable to unlearn the shortcuts even with further training on uncorrupted data. The consistency of the integrated gradient maps further corroborate the notion that pretraining with the CLIP architecture allowed the vision encoder to remain relatively robust to shortcuts.

We can explain these results by considering the source of shortcut learning. Shortcuts are learned due to associations that happen to be present in the training data. By forcing the image features to align with a more nuanced text embedding instead of simple binary labels, we can diminish these spurious associations that give rise to shortcut learning and learn more robust visual features.

5.2. Limitations

In this study, we generated shortcut datasets by watermarking label-associated symbols onto the images. This is fairly realistic for CXRs; prior studies have shown that watermarked laterality markers (DeGrave et al., 2021) and hospital tokens (Geirhos et al., 2020) posed risks for shortcut learning in CXRs. However, the associations we constructed were likely stronger than would be seen in a typical dataset. Nevertheless, using these strongly-correlated shortcuts enabled us to better visualize and manipulate shortcut usage, and we believe our findings will generalize to scenarios with less exaggerated associations.

Even after fine-tuning, the shortcut-trained CLIP model is still ineffective on adversarial examples, failing about half of the time. Furthermore, the real CNN saw a minor drop in performance after fine-tuning, which may be indicative of over-fitting. Additionally, while the integrated gradient results suggest that the CLIP model is more consistent in the presence of shortcuts, more investigation is needed to evaluate if these learned features are clinically relevant.

Software and Data

The datasets used are readily available:
MIMIC-CXR-JPG: https://physionet.org/content/mimic-cxr-jpg/2.0.0/,
CheXpert: https://stanfordmlgroup.github.io/competitions/chexpert/. Generation of the synthetic data is done at train/test time and these methods are fully described in the Data section.

Our code is available here: https://github.com/apalepu13/CNN_CLIP_Shortcut_Feature_Reliance

Acknowledgements

Research reported in this workshop was supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB), of the National Institutes of Health. The content is solely the responsibility of the authors and does not represent the official views of the National Institutes of Health. ALB received funding from the National Institutes of Health (K01 HL141771).

References

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., and McDermott, M. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 590–597, 2019.

Johnson, A. E., Pollard, T. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Peng, Y., Lu, Z., Mark, R. G., Berkowitz, S. J., and Horng, S. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., and Yan, J. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ren, S., Sun, J., He, K., and Zhang, X. Deep residual learning for image recognition. In *CVPR*, volume 2, pp. 4, 2016.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., and Langlotz, C. P. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

A. Appendix

Below are the AUCs for the eight models: (CNN/CLIP) trained on (Real/Shortcut) data (prior to/after) fine-tuning. The CLIP models prior to fine-tuning are performing zero-shot classification as described in the paper, while the rest are directly predicting the five clinical labels. Results are reported on the Real, Shortcut, and Adversarial test sets.

Real Test AUCs	Average	Atelectasis/Cardiomegaly/Consolidation/Edema/Pleural Effusion
Real CNN	0.866	0.807 / 0.840 / 0.895 / 0.902 / 0.886
Zero-shot Real CLIP	0.749	0.556 / 0.841 / 0.776 / 0.853 / 0.720
Fine-tuned Real CNN	0.830	0.735 / 0.825 / 0.888 / 0.846 / 0.858
Fine-tuned Real CLIP	0.841	0.744 / 0.846 / 0.911 / 0.830 / 0.872
Shortcut CNN	0.872	0.867 / 0.838 / 0.922 / 0.864 / 0.868
Zero-shot Shortcut CLIP	0.686	0.565 / 0.766 / 0.567 / 0.884 / 0.649
Fine-tuned Shortcut CNN	0.804	0.728 / 0.720 / 0.902 / 0.829 / 0.843
Fine-tuned Shortcut CLIP	0.819	0.657 / 0.841 / 0.891 / 0.834 / 0.872

Shortcut Test AUCs	Average	Atelectasis/Cardiomegaly/Consolidation/Edema/Pleural Effusion
Real CNN	0.866	0.806 / 0.843 / 0.886 / 0.905 / 0.887
Zero-shot Real CLIP	0.765	0.568 / 0.843 / 0.818 / 0.866 / 0.728
Fine-tuned Real CNN	0.834	0.740 / 0.829 / 0.887 / 0.855 / 0.859
Fine-tuned Real CLIP	0.841	0.751 / 0.846 / 0.905 / 0.832 / 0.872
Shortcut CNN	1.000	0.999 / 1.000 / 1.000/ 1.000 / 1.000
Zero-shot Shortcut CLIP	0.946	0.893 / 0.956 / 0.979 / 0.998 / 0.903
Fine-tuned Shortcut CNN	0.996	0.984 / 1.000 / 1.000 / 1.000/ 0.997
Fine-tuned Shortcut CLIP	0.945	0.978 / 0.988 / 0.844 / 0.941 / 0.973

Adversarial Test AUCs	Average	Atelectasis/Cardiomegaly/Consolidation/Edema/Pleural Effusion
Real CNN	0.865	0.807 / 0.830 / 0.902 / 0.897 / 0.885
Zero-shot Real CLIP	0.737	0.547 / 0.837 / 0.752 / 0.837 / 0.713
Fine-tuned Real CNN	0.827	0.729 / 0.820 / 0.887 / 0.840 / 0.857
Fine-tuned Real CLIP	0.838	0.735 / 0.841 / 0.915 / 0.827 / 0.872
Shortcut CNN	0.001	0.002 / 0.000 / 0.001 / 0.000/ 0.000
Zero-shot Shortcut CLIP	0.165	0.111 / 0.356 / 0.036 / 0.224 / 0.098
Fine-tuned Shortcut CNN	0.025	0.060 / 0.000 / 0.008 / 0.003 / 0.056
Fine-tuned Shortcut CLIP	0.518	0.065 / 0.392 / 0.929 / 0.585 / 0.619