Shortcut Learning via Background Bias: A Controlled Counterfactual Study

Bakhodirov Akmal
Student at INHA University in Tashkent
akmal040706@gmail.com

Abstract

Deep neural networks often exploit spurious correlations when such features are predictive of training labels, leading to high in-distribution performance but poor robustness under distribution shift. In visual recognition tasks, background cues are a common source of such shortcut learning. In this work, we present a controlled experimental study that isolates background bias as a single causal factor in binary image classification. Starting from a fixed set of foreground object images and segmentation masks, we construct four datasets that differ only in background statistics: an original dataset, a dataset with perfectly class-correlated backgrounds, a counterfactual dataset with inverted background–label associations, and a dataset with randomized backgrounds. Using identical model architectures, initializations, data splits, and training procedures across all experiments, we show that models trained on biased backgrounds achieve perfect in-distribution accuracy yet fail catastrophically under counterfactual background shifts, while models trained on randomized backgrounds generalize robustly. These results provide a minimal, fully reproducible demonstration of shortcut learning induced by background bias.

1 Introduction

Deep learning models often achieve impressive accuracy on benchmark datasets, yet their success can mask reliance on unintended shortcuts rather than semantically meaningful features. Such shortcut learning arises when spurious correlations between labels and non-causal features are present in the training data. In image classification, background cues are a particularly common source of spurious correlation, as object classes frequently co-occur with characteristic environments.

While prior work has documented shortcut learning in large-scale, real-world datasets, these settings often involve multiple confounding factors, making it difficult to attribute failures in generalization to a single cause. In this paper, we adopt a deliberately minimal and controlled approach. By holding all factors constant except for background color statistics, we isolate background bias as the sole source of shortcut learning.

We design a set of datasets that differ only in how background pixels are assigned, enabling direct causal comparisons. Our experiments reveal a stark contrast between models trained with class-correlated backgrounds and those trained with randomized backgrounds, despite identical architectures, optimization procedures, and data splits. This clarity allows us to draw strong conclusions about the role of background bias in shaping learned representations.

2 Dataset Construction

All datasets are derived from the same set of foreground object images depicting cats and dogs, along with corresponding binary segmentation masks that separate foreground objects from background

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

pixels. Across all datasets, foreground pixels, image resolution, labels, and train/validation/test splits are held fixed. Only background pixel values differ.

2.1 Dataset A: Original Dataset

Dataset A serves as the reference distribution. Images contain the original foreground objects with their original background content. Labels are deterministically assigned based on filename conventions: filenames beginning with an uppercase character correspond to the class *cat*, while filenames beginning with a lowercase character correspond to the class *dog*. The dataset is organized in standard ImageFolder format with separate directories for each class.

2.2 Dataset B: Biased Background Dataset

Dataset B introduces a perfectly predictive spurious correlation between background color and class label. For each image, foreground pixels are preserved exactly, while background pixels are replaced with a constant color determined by the class label: green backgrounds for cats and red backgrounds for dogs. This mapping is deterministic and involves no randomness. As a result, background color alone suffices to predict the label with 100% accuracy.

2.3 Dataset C: Counterfactual Dataset

Dataset C is constructed identically to Dataset B, except that the background–label mapping is reversed. Cats are placed on red backgrounds and dogs on green backgrounds. This dataset represents a counterfactual intervention that preserves all low-level image statistics except for the semantic association between background color and class. Dataset C is never used for training and serves exclusively for evaluation under distribution shift.

2.4 Dataset D: Random Background Dataset

Dataset D removes systematic correlations between background and class by assigning each image a randomly sampled background color. Randomness is controlled using a fixed seed to ensure full reproducibility. As in the other datasets, foreground pixels are unchanged and masks are applied deterministically. Dataset D is intended to encourage models to rely on object features rather than background cues.

3 Experimental Setup

3.1 Model Architecture

All experiments use the same convolutional neural network architecture (SimpleCNN). The architecture is held constant across datasets and seeds to ensure fair comparison.

3.2 Initialization and Training Protocol

To eliminate variability due to random initialization, a single set of initial model weights is generated using a fixed random seed and reused across experiments. Training procedures, optimization hyperparameters, and stopping criteria are identical for all runs.

3.3 Data Splits

A single train/validation/test split is generated once using a fixed random seed and reused across all datasets. This guarantees that performance differences arise solely from dataset properties rather than sample composition.

3.4 Evaluation Metrics

Models are evaluated using accuracy, precision, recall, F1 score, and confusion matrices. All metrics are computed on held-out test sets.

4 Results

4.1 Overview of Train–Test Performance

Table 1 summarizes classification performance across all evaluated train–test dataset pairs, reported as mean ± standard deviation over three random seeds. Accuracy, precision, recall, and F1-score are included for completeness. The results reveal strong asymmetries in generalization behavior depending on background statistics.

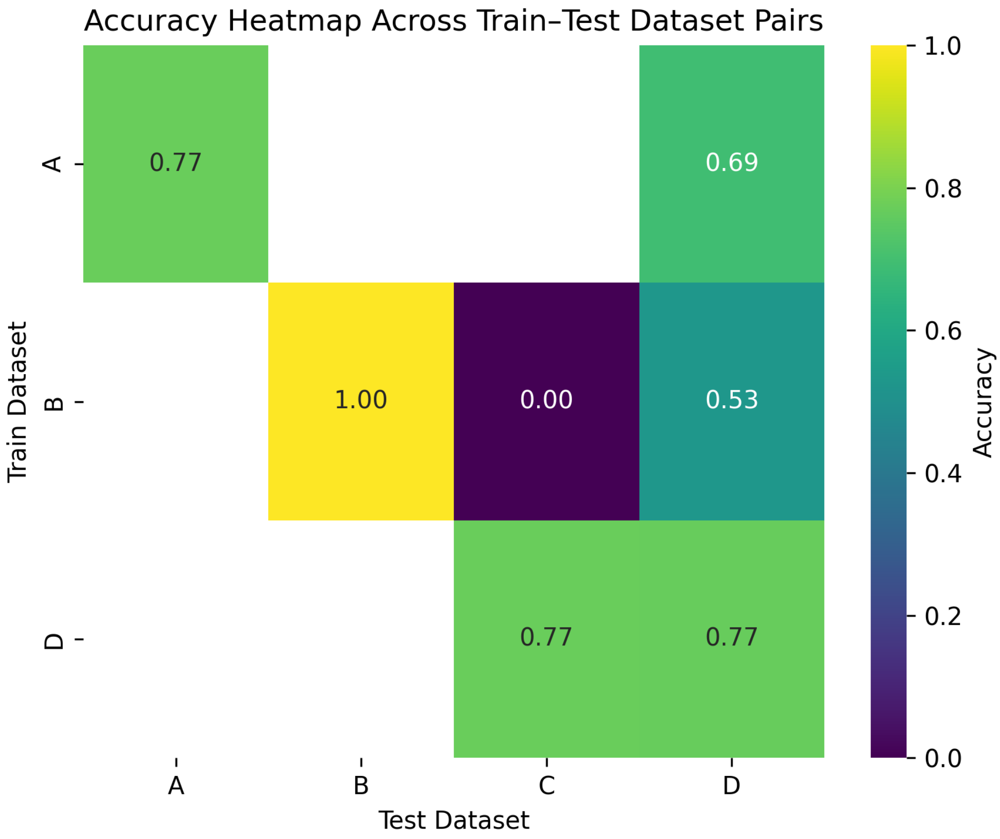

Figure 1: Train–test accuracy heatmap across dataset pairs. Models trained on Dataset B fail completely when evaluated on Dataset C, indicating reliance on spurious background features rather than object shape.

Table 1: Classification performance (mean ± std over 3 seeds).

Train	Test	Accuracy	Precision	Recall	F1
A	A	0.771 ± 0.011	0.806 ± 0.012	0.862 ± 0.006	0.833 ± 0.007
A	D	0.695 ± 0.010	0.791 ± 0.008	0.731 ± 0.026	0.760 ± 0.012
B	B	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
B	C	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
B	D	0.531 ± 0.008	0.655 ± 0.002	0.611 ± 0.020	0.632 ± 0.012
D	C	0.772 ± 0.023	0.793 ± 0.012	0.887 ± 0.040	0.837 ± 0.019
D	D	0.769 ± 0.011	0.798 ± 0.010	0.870 ± 0.008	0.833 ± 0.008

Figure 1 visualizes these results as a heatmap of mean accuracy across train–test combinations, highlighting systematic generalization failures under counterfactual background shifts.

4.2 Biased Backgrounds and Counterfactual Failure

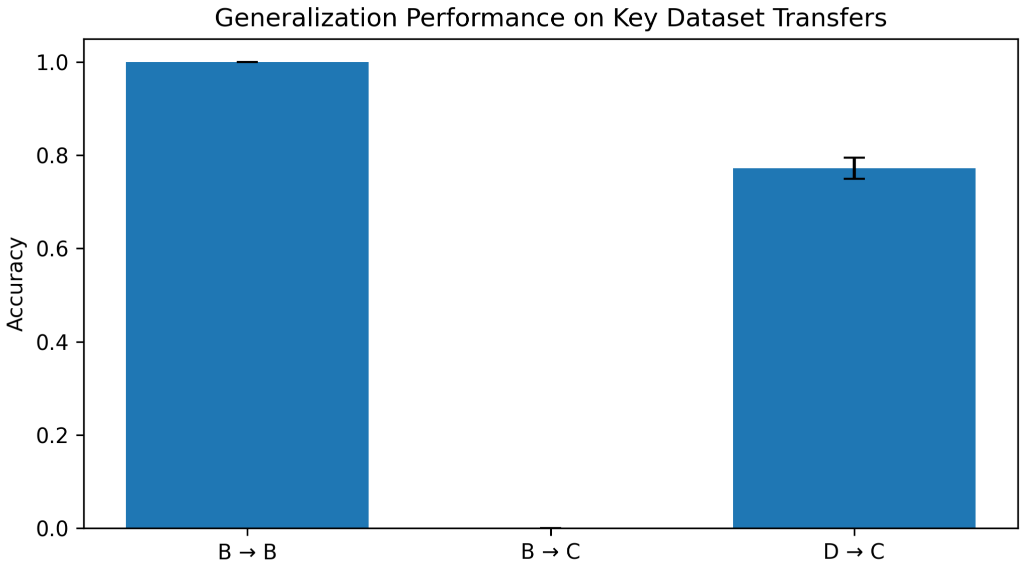

Figure 2: Generalization performance on key dataset transfers. Models trained on the biased Dataset B achieve perfect accuracy in-distribution (B→B) but fail catastrophically under counterfactual background shifts (B→C). Training on Dataset D mitigates this failure.

As shown in Table 1 and Figure 2, models trained on the biased Dataset B achieve perfect in-distribution accuracy on Dataset B. However, when evaluated on the counterfactual Dataset C, accuracy drops to zero for all seeds, with confusion matrices indicating complete label inversion. This behavior demonstrates exclusive reliance on background color as a decision cue.

4.3 Random Backgrounds Improve Robustness

Models trained on Dataset D exhibit substantially more stable behavior. As shown in Table 1 and Figure 2, performance on D→C remains comparable to D→D, indicating robustness to background color changes. Figure 3 further shows accuracy on Dataset D when training on A, B, and D. Training on randomized backgrounds yields the highest robustness, while training on biased backgrounds degrades performance even on random-background test data.

Together, these results demonstrate that removing systematic background–label correlations during training is sufficient to prevent catastrophic shortcut learning.

5 Discussion

The experimental results demonstrate that shortcut learning can arise even in extremely simple settings when spurious correlations are perfectly predictive. More importantly, they show that controlling background statistics during training is sufficient to substantially improve robustness under counterfactual interventions. The stark contrast between models trained on Datasets B and D highlights how dataset design choices directly shape the features that neural networks learn.

Our controlled setup complements prior studies of shortcut learning by removing many of the confounding factors present in large-scale datasets. Because foreground pixels, labels, splits, and initialization are all held constant, the observed effects can be causally attributed to background bias alone.

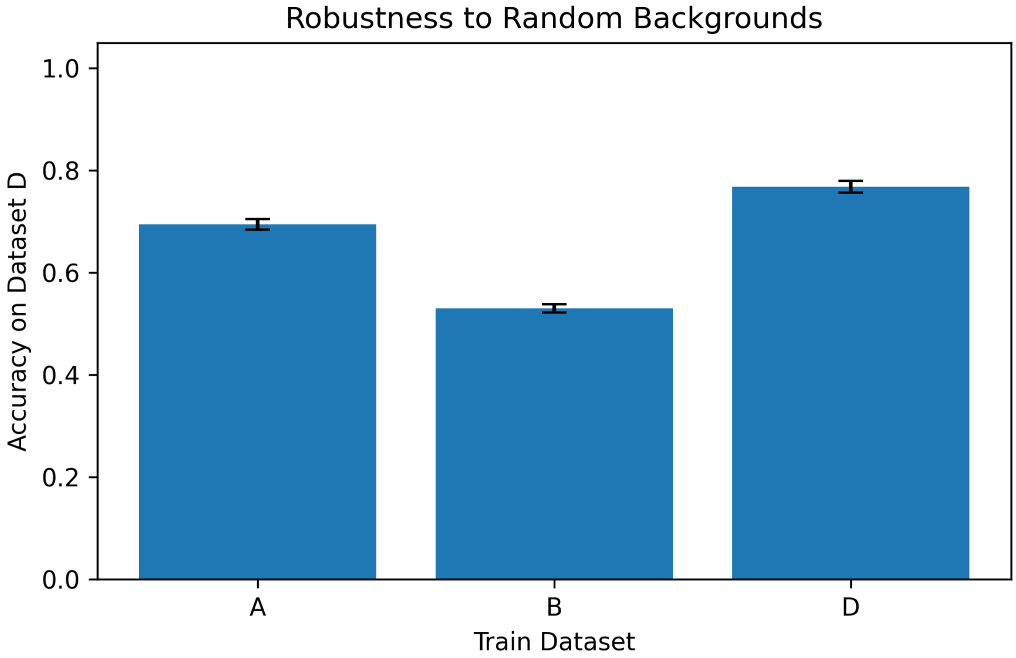

Figure 3: Accuracy on Dataset D when models are trained on different datasets. Training on randomized backgrounds (Dataset D) yields higher robustness compared to training on biased backgrounds (Dataset B).

6 Conclusion

We presented a fully controlled, reproducible study of shortcut learning induced by background bias in image classification. By constructing counterfactual and randomized background datasets from the same underlying images, we showed that models trained with class-correlated backgrounds fail catastrophically under background shifts, while models trained with randomized backgrounds generalize robustly. These findings underscore the importance of dataset design and motivate further work on training strategies that mitigate shortcut learning.

7 Related Work

Shortcut learning in deep neural networks has been widely documented across vision and language tasks. Geirhos et al. (2020) introduced the term *shortcut learning* to describe the tendency of models to rely on spurious, non-semantic correlations when such features are predictive of the training labels. In image classification, background cues have been shown to act as particularly strong shortcuts, often dominating object shape and texture.

Ilyas et al. (2019) demonstrated that many features learned by modern classifiers are *non-robust*, meaning they are highly predictive yet brittle under small distribution shifts. Subsequent work has explored dataset bias arising from contextual correlations, including object–background co-occurrence and scene-level cues, showing that models often fail when such correlations are altered or removed.

Several studies have proposed mitigation strategies, such as data augmentation, background randomization, and causal interventions, to reduce reliance on spurious features. However, many existing benchmarks involve multiple confounding factors simultaneously, making it difficult to isolate the precise cause of shortcut behavior.

In contrast, our work adopts a deliberately minimal and fully controlled setup in which background color is the sole varying factor across datasets. By holding foreground objects, labels, splits, and initialization constant, we provide a clear causal demonstration of how background bias alone can induce shortcut learning and catastrophic counterfactual failure.

8 References

- Geirhos, R., Jacobsen, J. H., Michaelis, C., et al. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*.
- Ilyas, A., Santurkar, S., Tsipras, D., et al. (2019). Adversarial examples are not bugs, they are features. *NeurIPS*.

9 Appendix A: Reproducibility Details

All dataset generation scripts, random seeds, model initialization procedures, and data splits are deterministic and provided with the codebase. Given the released scripts and fixed seeds, all results reported in this paper can be reproduced exactly.