Chroma-VAE: Mitigating Shortcut Learning with Generative Classifiers

Wanqian Yang
Department of Computer Science
New York University
New York, NY 10012
wanqian@nyu.edu

Polina Kirichenko
Department of Computer Science
New York University
New York, NY 10012
pk1822@nyu.edu

Micah Goldblum
Department of Computer Science
New York University
New York, NY 10012
mg7285@nyu.edu

Andrew Gordon Wilson
Department of Computer Science
New York University
New York, NY 10012
andrewgw@cims.nyu.edu

Abstract

Deep neural networks are susceptible to shortcut learning, using simple features to achieve low training loss without discovering essential semantic structure. Contrary to prior belief, we show that generative models alone are not sufficient to prevent shortcut learning, despite an incentive to recover a more comprehensive representation of the data than discriminative approaches. However, we observe that shortcuts are preferentially encoded with minimal information, a fact that generative models can exploit to mitigate shortcut learning. In particular, we propose Chroma-VAE, a two-pronged approach where a VAE classifier is initially trained to isolate the shortcut in a small latent subspace, allowing a secondary classifier to be trained on the complementary, shortcut-free latent subspace. In addition to demonstrating the efficacy of Chroma-VAE on benchmark and real-world shortcut learning tasks, our work highlights the potential for manipulating the latent space of generative classifiers to isolate or interpret specific correlations.

1 Introduction

As science fiction writer Robert Heinlein quipped in his 1973 novel *Time Enough for Love*, progress is made not by "early risers", but instead by "lazy men trying to find easier ways to do something". Indeed, we can accuse modern machine learning models of emulating the same behaviour. There are notable examples of models that wind up learning the wrong things. For example, images of cows standing on anything other than grass fields are commonly misclassified, because the combination of cows and grass fields is so prevalent in training data that the model simply learns to rely on the background as a predictive signal [4]. A more concerning example involves predicting pneumonia from chest X-ray scans. Hospitals in the training data have differing rates of diagnosis, a fact that the model easily exploits by learning to detect hospital-specific metal tokens in the scans rather than signals relevant to pneumonia itself [50].

Deep neural networks can learn brittle, unintended signals under empirical risk minimization (ERM) [11, 16, 13], a well-known phenomenon observed and studied by various communities. These signals often possess two key attributes: (i) they are *spuriously correlated* to the label and are therefore strongly predictive [6, 48, 29], despite having no meaningful semantic relationship with the label, and (ii) they are learnt by the neural network as a result of its *inductive biases* [41, 10]. A recent

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

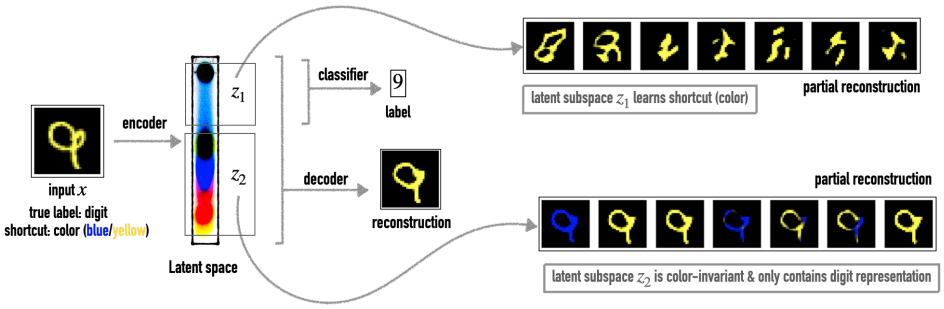


Figure 1: A visual representation of Chroma-VAE. The latent space is partitioned into two subspaces, one of which is incentivized to encode the shortcut under ERM. The complementary subspace learns a shortcut-invariant representation.

unifying effort by Geirhos et al. [10] coins the term *shortcut* to describe such a signal. Networks that learn shortcuts will fail to generalize to relevant or challenging distribution shifts.

Prior work has sought to alleviate this problem under various formalizations, most commonly in the settings of group robustness [e.g. 37, 26, 7] or adversarial robustness [e.g. 5, 43]. In this paper, we motivate a different approach to mitigate undesirable shortcuts, where we seek instead to learn a *shortcut-invariant representation* of the data — that is, "everything but the shortcut". To this end, we first observe empirically that deep neural networks preferentially encode shortcuts as they are the most efficient compression of the data that is strongly predictive of training labels. This helps to explain why shortcut learning is so prevalent amongst discriminative classifiers.

However, this same preference can be exploited in a *generative* classifier to learn a shortcut-invariant compression. Specifically, we consider the Variational Auto-Encoder (VAE) [22], capable of learning latent representations of the data. Our key insight is to back-propagate the classifier's loss through a latent *subspace*, while reconstructing with the *entire latent space*. The model minimizes classification error by encoding the shortcut in this subspace. Since the shortcut representation is isolated, the complementary subspace is free to learn a shortcut-invariant representation through reconstruction. The final classifier is then trained on this invariant representation. We dub our approach **Chroma-VAE**, inspired by the technique of chromatography, which separates the components of a chemical mixture travelling through the mobile phase. Figure 1 summarizes our approach.

Our contributions are as follows:

- We show empirically that: (i) deep neural networks preferentially encode shortcuts as the most efficient compression (Section 3.1), and (ii) the shortcut representation can be sequestered in a latent subspace of the VAE classifier (Section 3.2).
- These observations allow us to propose Chroma-VAE (Section 3.3), an approach designed to learn a classifier on a shortcut-invariant representation of the data. We demonstrate the effectiveness of Chroma-VAE on several benchmark and real-world datasets (Section 5).

2 Background and Notation

Even though shortcuts are present in many domains within deep learning [4, 48, 2, 32, 28, 12], we restrict our discussion to image classification tasks, as (i) VAEs are widely applied to modeling image distributions, and (ii) shortcut learning is ubiquitous in the vision domain.

Let $\mathcal{D}_{tr} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ denote i.i.d. training data. A variational auto-encoder (VAE) [22] (E_ϕ, D_θ) models $\mathbf{x}$ with a latent variable: $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$. The prior $p(\mathbf{z})$ is typically $\mathcal{N}(\mathbf{0}, I)$ and the likelihood $p(\mathbf{x}|\mathbf{z})$ is implicitly modeled by the decoder network $D_\theta(\mathbf{z}) = \big(\mu_\theta(\mathbf{z}), \text{diag}\left(\sigma^2_\theta(\mathbf{z})\right)\big)$ as $\mathcal{N}\big(\mathbf{x}|\mu_\theta(\mathbf{z}), \text{diag}\left(\sigma^2_\theta(\mathbf{z})\right)\big)$. Maximizing $p(\mathbf{x})$ directly is intractable due to the integral over $\mathbf{z}$. Instead, we use an encoder network $E_\phi(\mathbf{x}) = \Big(\mu_\phi(\mathbf{x}), \text{diag}\left(\sigma^2_\phi(\mathbf{x})\right)\Big)$ as an (amortized) variational

approximation $q_\phi\big(\mathbf{z}|\mu_\phi(\mathbf{x}), \text{diag}(\sigma^2_\phi(\mathbf{x}))\big)$ and maximize the Evidence Lower BOund (ELBO):

$$\text{ELBO}(\theta, \phi, \mathbf{x}) := \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \tag{1}$$

by computing unbiased estimates of its gradients.

In a hybrid VAE classifier $(E_\phi, D_\theta, C_\varphi)$, we model $\mathcal{D}_{tr}$ as $p(\mathbf{z}, \mathbf{x}, y) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})p(y|\mathbf{x})$. $p(y|\mathbf{x})$ is modeled by a classifier network C_φ that takes the latent mean $\mu_\phi(\mathbf{x})$ as input and outputs class probabilities, i.e. $p_{\varphi,\phi}(y|\mathbf{x}) = \text{CE}\Big(y, C_\varphi\big(\mu_\phi(\mathbf{x})\big)\Big)$, where CE is the cross-entropy loss. Again, maximizing $p(\mathbf{x}, y)$ directly is intractable and the training objective becomes:

$$\mathcal{L}(\theta, \phi, \varphi) := -\sum_{(\mathbf{x},y)\in\mathcal{D}_{tr}} \text{ELBO}(\theta, \phi, \mathbf{x}) + \lambda \cdot \log p_{\varphi,\phi}(y|\mathbf{x}) \tag{2}$$

where λ is a scalar multiplier widely used in practice to account for the fact that $p(\mathbf{x})$ is typically magnitudes smaller than $p(y|\mathbf{x})$ on high-dimensional image datasets.

Group Terminology Let $s \in \mathcal{S}$ denote the shortcut (spurious) label. The *group* of an input $\mathbf{x}$ is $g = (s, y)$, the combination of its shortcut and true labels. *Majority groups* refer to examples with the dominant correlation between s and y on the training data, while *minority groups* refer to the small number of examples with the opposite correlation that ERM models typically misclassify. *Group robustness* refers to the ability of models to generalize to distribution shifts where shortcuts are no longer predictive. Methods to improve group robustness can make use of group annotations, however, our method *does not assume access* to group labels at train or test time.

3 Chroma-VAE: Separating Shortcuts Generatively

The key intuition behind our approach is to exploit shortcut learning for representation learning, by using a VAE to learn a shortcut-invariant representation of the data. Two central ideas, supported by empirical observations, motivate our method: (i) deep neural networks preferentially encode shortcuts under finite and limited representation capacity, and (ii) the shortcut representation can be sequestered in a latent *subspace* of the VAE when jointly trained with a classifier.

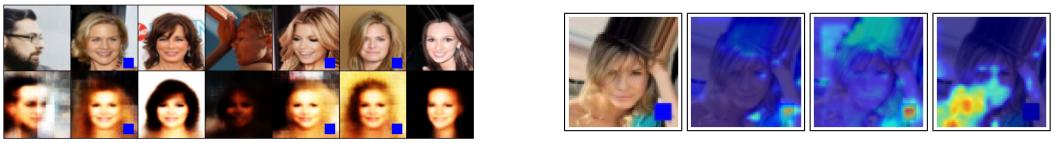
3.1 Shortcuts are preferentially encoded under limited representation capacity

Why do shortcuts exist? As Geirhos et al. [10] note, shortcuts arise because the model's inductive biases (the sum total of interactions between training objective, optimization algorithm, architecture, and dataset) favour learning certain patterns over others, e.g. convolutional neural networks (CNNs) prefer texture over global shape [3, 9].

We approach this from an information-theoretic lens. Deep neural networks are commonly thought of as representation learners that optimize the information bottleneck (IB) trade-off, i.e. they aim to learn a maximally compressed representation (minimally sufficient statistic) that fits the labels well [45, 40]. Our key empirical observation is that in datasets where shortcuts exist, **they are often efficiently compressed, and the compressed information is predictive of labels**. As such, deep models preferentially encode shortcuts, especially under limited representation capacity.

We consider an experiment on the `CelebA` dataset [27], where the task is predicting hair colour ("blond" or not). We inject a synthetic shortcut in the form of a 10×10 blue "patch", superimposed onto the image with 0.9 probability for the positive class (see Figure 2b). We first train a discriminative classifier on the labels, and then *independently* learn a decoder on the (fixed) hidden layer of the trained discriminative model.

Figure 2a shows the reconstructions of the decoder. The shortcut is accurately replicated in sharp detail where it exists, but other regions of the reconstructions are not as faithful to the originals. In particular, note that reconstructions are even less accurate for examples of the positive class (with the shortcut), where mode collapse has happened. Since the decoder was learned separately, this implies that the classifier's latent compression primarily encoded for the shortcut's presence.



(a) (b) (c) (d) (e)

Figure 2: **(Left) (a)** Examples of reconstructions by a decoder trained independently on the discriminative model's hidden layer. The shortcut is sharply captured where it exists. Reconstruction is particularly poor on images *with* the shortcut — the reconstructed faces seem to collapse to the same archetype that does not resemble the original images. **(Right)** Grad-CAM heatmaps on a `CelebA` test image. **(b)** The original image with the shortcut patch. **(c)-(e)** Averaged activations across all training epochs on three separate models, with a bottlenecked hidden layer of sizes 4, 32, and 128 respectively. The smaller the bottleneck, the more likely it is that the model only focuses on the shortcut patch. With more capacity, the model focuses on other regions of the image.

This result is further supported by visualizing the activated regions of the input images during training. We implement Grad-CAM [39], which computes a linear combination of activation maps weighted by the gradients of output logits (or any upstream parameters). This can be visualized as a heatmap superimposed on the original image, showing regions of the image with the greatest positive contributions to the network's prediction.

In Figures 2b-e, we visualized these heatmaps on models with a bottleneck hidden layer of different sizes. As we can see, the smaller the bottleneck size, the stronger the activations on the shortcut patch. With additional model capacity, the model is more likely to encode other relevant (or spurious) features *in addition to* the shortcut itself.

From these experiments, we see that deep neural networks tend to compress input data in a shortcut-preserving manner when such shortcuts are both (i) compressible with little information and (ii) predictive of training labels. This tendency is exacerbated under limited latent capacity and is the reason why discriminative classifiers are susceptible to shortcut learning under ERM.

3.2 VAE classifiers can sequester shortcuts into specific latent dimensions

What presents a problem for discriminative modeling can be exploited into an advantage using a generative model. The key insight here is that we can train a VAE classifier where the latent space is partitioned into two disjoint subspaces — however, only one subspace feeds into a classifier and is backpropagated through using labels. This subspace encodes the shortcut precisely because this information maximizes the classifier's performance (i.e. simultaneously exhibits high mutual information with the labels and small mutual information with the data). **The remaining subspace therefore encodes a partial representation of the image without the shortcut.**

Formally, we modify the standard VAE classifier described in Section 2 as follow: we partition the latent space as $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$. The classifier C_φ uses only $\mathbf{z}_1$ as input, i.e. for a given data point $\mathbf{x}$, the predicted output is $C_\varphi\big(\mu_\phi(\mathbf{x})_1\big)$. As such, $\mathbf{z}_1$ is used for both reconstruction and classification, whereas $\mathbf{z}_2$ is used for reconstruction only.

We visualize the results of this experiment in two ways: (i) Grad-CAM, where we use gradient weights both of $\mu_\phi(\mathbf{x})_1$ and of $\mu_\phi(\mathbf{x})_2$, and (ii) sampling **partial reconstructions** $\tilde{\mathbf{x}}$ from the VAE: given an encoding $\mu_\phi(\mathbf{x})$, we replace each half $\mu_\phi(\mathbf{x})_i$ with standard Gaussian noise and sample from the decoder:

$$\tilde{\mathbf{x}}_1 \sim D_\theta\Big(\big[\mu_\phi(\mathbf{x})_1 \quad N(\mathbf{0}, I)\big]^\top \Big) \tag{3}$$

$$\tilde{\mathbf{x}}_2 \sim D_\theta\Big(\big[N(\mathbf{0}, I) \quad \mu_\phi(\mathbf{x})_2\big]^\top \Big) \tag{4}$$

We use the same predictive task and synthetic shortcut as described in Section 3.1. The visualizations are displayed in Figure 3. As can be seen from Figures 3b-c, Grad-CAM shows that $\mu_\phi(\mathbf{x})_1$ is most sensitive to pixels around the blue patch, whereas $\mu_\phi(\mathbf{x})_2$ is sensitive to pixels spread out in other regions of the image (including certain regions of the celebrity's blonde hair). This highlights that the representation of the patch is largely isolated in $\mathbf{z}_1$.

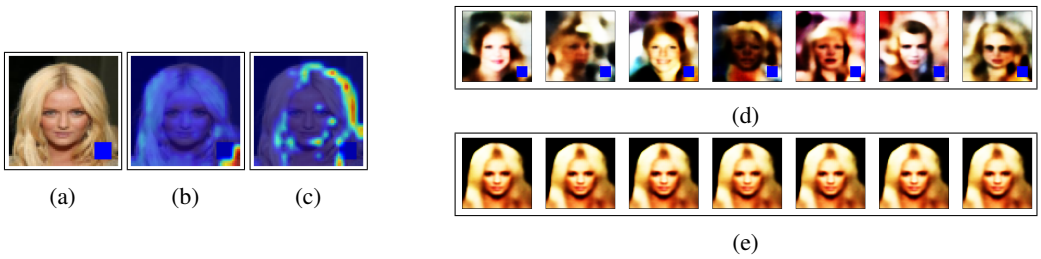


Figure 3: **(a)** Original image containing shortcut. **(b)** Grad-CAM with $\mu_\phi(\mathbf{x})_1$ gradients. **(c)** Grad-CAM with $\mu_\phi(\mathbf{x})_2$ gradients. **(d)** Partial reconstructions $\tilde{\mathbf{x}}_1$. **(e)** Partial reconstructions $\tilde{\mathbf{x}}_2$. While no patches are present in samples of $\tilde{\mathbf{x}}_2$ here, we have observed that they can occur with small probability. Conversely, across our many repeated samplings, the shortcut patch is *always* present in samples of $\tilde{\mathbf{x}}_1$. Note also that $\tilde{\mathbf{x}}_1$ samples contain a variety of hair colours, whereas $\tilde{\mathbf{x}}_2$ samples faithfully reconstruct the celebrity's blonde hair.

Partial reconstructions in Figures 3d-e support this observation. In Figure 3d, where $\mu_\phi(\mathbf{x})_1$ is fixed and we sample only $\mathbf{z}_2 \sim N(\mathbf{0}, I)$, the decoded samples $\tilde{\mathbf{x}}_1$ represent a variety of faces that do not resemble the original image. However, *all* samples contain the shortcut patch in the bottom-right corner. Conversely, in Figure 3e where $\mu_\phi(\mathbf{x})_2$ is fixed, the reconstructed samples $\tilde{\mathbf{x}}_2$ greatly resemble the original image, but may or may not contain the shortcut. In other words, $\mu_\phi(\mathbf{x})_1$ is strongly correlated with the shortcut's presence, whereas $\mu_\phi(\mathbf{x})_2$ is *uncorrelated* with the shortcut.

Together, Grad-CAM and the partial reconstructions suggest that the latent representation of the shortcut patch has largely been sequestered into $\mathbf{z}_1$. Intuitively, this is believable — as the classification loss on C_φ only back-propagates through $\mathbf{z}_1$, it will learn representations that are most useful for prediction. This representation is dominated by the shortcut patch, as we showed in Section 3.1. Since the model is also simultaneously learning to reconstruct the image, most other information describing the image is partitioned into $\mathbf{z}_2$, resulting in $\tilde{\mathbf{x}}_2$ samples being very similar to the original.

3.3 Chroma-VAE

These results suggest a simple approach: we only need to train a separate, secondary classifier $C^{(2)}_{\varphi'}$ on $\mathbf{z}_2$ **after** the initial VAE classifier $(E_\phi, D_\theta, C_\varphi)$ has been trained, i.e. $\mathbf{z}_2$ is a fixed input into $C^{(2)}_{\varphi'}$ that is not back-propagated through to $\mathbf{x}$. Section 3.2 shows that the initial VAE classifier learns $\mathbf{z}_2$ in a way that minimizes shortcut representation but contains other salient features (by virtue of $\mathbf{z}_2$ learning to reconstruct). As such, $C^{(2)}_{\varphi'}$ will be shortcut-invariant while remaining predictive.

We name this approach **Chroma-VAE**. A full description is provided in Algorithm 1. Our method has two hyperparameters: the dimensionality of $\mathbf{z}$, $\dim(\mathbf{z})$, and the partition fraction, $z_p = \frac{\dim(\mathbf{z}_1)}{\dim(\mathbf{z}_1)+\dim(\mathbf{z}_2)} \in (0, 1)$, which controls the relative sizes of $\mathbf{z}_1$ and $\mathbf{z}_2$.

From here on, we will refer to C_φ as the $\mathbf{z}_1$-classifier ($\mathbf{z}_1$`-clf`; *not* used for prediction) and $C^{(2)}_{\varphi'}$ as the $\mathbf{z}_2$-classifier ($\mathbf{z}_2$`-clf`; shortcut-invariant classifier used for prediction). Since $\mathbf{z}_2$ is a latent vector, the $\mathbf{z}_2$-classifier need not be a deep neural net. Indeed, we noticed in our experiments that simpler models like k-nearest neighbors can sometimes perform better; see Section 5 for more details.

$\tilde{\mathbf{x}}_2$-classifier As we observed in Section 3.2, the partial reconstructions $\tilde{\mathbf{x}}_2$ are uncorrelated with the shortcut. As such, instead of training $C^{(2)}_{\varphi'}$ on $\mathbf{z}_2$, we can simply generate partial reconstructions and train a classifier on samples of $\tilde{\mathbf{x}}_2$ directly in $\mathcal{X}$-space. This allows us to exploit deep models since the inputs are now images. We term this the $\tilde{\mathbf{x}}_2$-classifier ($\tilde{\mathbf{x}}_2$`-clf`).

4 Related Work

Generative Classifiers Depending on how we decompose $p_\theta(\mathbf{x}, y)$, generative classifiers can be divided into two categories: (i) *class-conditional models* $p_\theta(y)p_\theta(\mathbf{x}|y)$ or (ii) *hybrid models* $p_\theta(\mathbf{x})p_\theta(y|\mathbf{x})$. The majority of previous work on deep generative classifiers [e.g. 38, 52] focus on

Algorithm 1 Chroma-VAE

Input: data $\mathcal{D}_{tr}$, VAE classifier $(E_\phi, D_\theta, C_\varphi)$, $\mathbf{z}_2$-classifier $C^{(2)}_{\varphi'}$, optimization hyperparameters
// Training initial VAE classifier
initialize (θ, ϕ, φ)
using Adam, minimize Equation (2), noting that C_φ only takes $\mu_\phi(\mathbf{x})_1$ as input
// Training $\mathbf{z}_2$-classifier
initialize φ'
compute the (fixed) input $\mu_\phi(\mathbf{x})_2$ using E_ϕ
using Adam, minimize $\mathcal{L}(\varphi') = \text{CE}\Big(y, C^{(2)}_{\varphi'}\big(\mu_\phi(\mathbf{x})_2\big)\Big)$
return $C^{(2)}_{\varphi'}$

the former category, e.g. for semi-supervised learning [23, 19] or adversarial robustness [38]. The most notable work in the latter category is Nalisnick et al. [30], which applies hybrid models to OOD detection and semi-supervised learning. Our work uses VAEs as the deep generative component, since we require explicit latent representation. We only focus on *hybrid* models as we desire a single model that learns the shortcut representation across all classes.

Group Robustness Out-of-distribution (OOD) generalization is a broad area of study [e.g. 8, 14, 35], depending on what assumptions are made on the relationship between train and test distributions as well as the information known at train time. In this work, we primarily consider distribution shifts arising from the presence of shortcuts or spurious correlations, where signals predictive in the train distribution are no longer correlated to the label in the test distribution. Prior work has generally approached this from the *group robustness* perspective, where the objective is to maximize worst-group accuracy (typically the minority groups) while retaining strong average accuracy. To the best of our knowledge, Chroma-VAE is the first method to tackle shortcut learning from a supervised representation learning approach using generative classifiers.

Methods in this space can be distinguished by the assumptions they make. Some work rely on having group labels for training data [1, 36, 51, 37], for example, Sagawa et al. [36] optimize worst-group accuracy directly. However, as group annotations can be expensive to acquire, other approaches relax this requirement [26, 7, 42, 49, 31]. For example, Liu et al. [26], which we compare to, treat misclassified examples by an initial model as a proxy for minority groups; these samples are up-weighted when training the final model. However, their approach is brittle to hyperparameter choices, requiring group labels on validation data (from the test distribution) for hyperparameter tuning.

Compared to existing work, Chroma-VAE does not require training group labels. Furthermore, we found that Chroma-VAE hyperparameters are not sensitive to groups — results in Section 5 were tuned using *average accuracy* on validation data. We hypothesize the reason for this robustness: the effectiveness of Chroma-VAE depends on (i) the shortcut being successfully isolated in $\mathbf{z}_1$ and (ii) $\mathbf{z}_2$ having sufficient capacity to learn the non-shortcut predictive features. (ii) is achieved by choosing $\dim(\mathbf{z})$ appropriately to ensure good reconstruction — this can be evaluated without group labels. Given such an appropriate $\dim(\mathbf{z})$, most non-extreme values of z_f will ensure that $\mathbf{z}_1$ has enough capacity to encode the shortcut. As such, the $\mathbf{z}_2$-classifier will be shortcut-invariant and can be tuned without needing group labels. Table 1 summarize the group label requirements.

Example	Group-DRO [36]	JTT [26]	Chroma-VAE
Group Labels on Train Data	✓	✗	✗
Group Labels on Validation Data	✓	✓	✓/✗

Table 1: Comparison of group label requirements. While Chroma-VAE can be tuned using validation worst-group accuracy, we have found that tuning for average accuracy suffices.

Method	$\mathcal{D}_{in}$	$\mathcal{D}_{out}$
Theoretical UB	75	75
Invariant	60.8	65.6
Naive-Class	87.8	17.8
Naive-VAE-Class	89.0	13.2
Naive-Independent	89.7	11.4
JTT	63.2	63.8
Chroma-VAE ($\mathbf{z}_1$-clf)	88.8	14.8
Chroma-VAE ($\mathbf{z}_2$-clf)	**70.9**	**70.8**
Chroma-VAE ($\tilde{\mathbf{x}}_2$-clf)	69.3	67.9

Table 2: Predictive accuracies on $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$. Chroma-VAE ($\mathbf{z}_2$-clf) is the best-performing classifier. Note that the best that a digit-only classifier can do is 75% since $p_d = 0.75$ by construction.

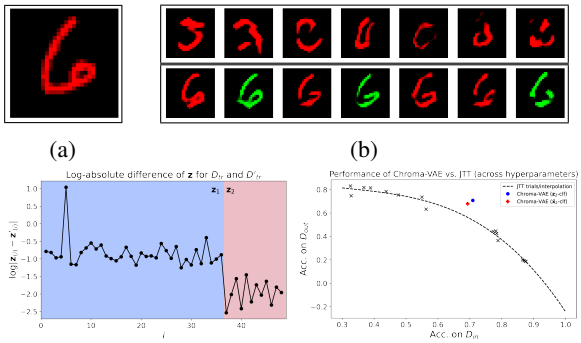


Figure 4: **(a)** Original image. **(b)** Partial reconstructions $\tilde{\mathbf{x}}_1$ *(top row)* and $\tilde{\mathbf{x}}_2$ *(bottom row)*. All samples of $\tilde{\mathbf{x}}_1$ preserve the colour of the original image but not the digit. Conversely, samples of $\tilde{\mathbf{x}}_2$ preserve the digit but can be variably green or red. **(c)** Plot of log-absolute difference for each dimension of $\mathbf{z}$ on $\mathcal{D}_{tr}$ and $\mathcal{D}'_{tr}$. Dimensions of $\mathbf{z}_2$ have smaller differences than $\mathbf{z}_1$. **(d)** Accuracies of various hyperparameter combinations of JTT. We used a polynomial (cubic) regressor to interpolate the accuracy trade-off on $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$; the best point on this curve is reported in Table 2. Both chroma-VAE classifiers lie above this curve.

5 Experiments

In Section 5.1, we present results on the `ColouredMNIST` benchmark, a proof-of-concept which we also use to highlight some key observations and comparisons. In Section 5.2, we present results on two additional benchmark datasets: `CelebA` (gender/hair colour) and `Dominoes` (`MNIST-FashionMNIST`). Finally, in Section 5.3, we demonstrate Chroma-VAE on a real-life problem — namely, pneumonia prediction using chest X-ray scans. In addition, Appendix B.1 contains further results on the `CelebA` synthetic patch proof-of-concept that was presented earlier in Section 3.

Baselines and Metrics Detailed experimental setups are found in Appendix A. For `ColouredMNIST`, we evaluate predictive accuracy on the various test distributions. For the other datasets, we evaluate average and worst-group accuracies. We consider the following baselines:

1. **Naive discriminative classifier** (Naive-Class): ERM classifier
2. **Naive VAE classifier** (Naive-VAE-Class): standard VAE classifier
3. **Naive VAE + classifier** (Naive-Independent): standard VAE is first trained on unlabelled data $\{\mathbf{x}_i\}_{i=1}^N$, the classifier is then separately trained on the latent projection $\{\mu_\phi(\mathbf{x}), y\}_{i=1}^N$
4. **Just Train Twice** (JTT) [26]: classifier is trained with a limited number of epochs, misclassified training points are upweighted and a second classifier is trained again

For fairness, we do not compare to methods that use group labels [36, 34], multiple training environments [1], or counterfactual examples [44].

5.1 ColouredMNIST

We follow the same experimental design as Arjovsky et al. [1]: (i) we first binarize `MNIST` [24] labels $\hat{y}$ (with digits 0-4 and 5-9 as the two classes), (ii) we obtain *actual* labels y (used in training and evaluation) by flipping $\hat{y}$ with probability $p_d = 0.25$, and finally (iii) we obtain colour labels c by flipping y with variable probability p_c, where c is used to colour each image green or red. In the training distribution $\mathcal{D}_{in}$, $p_c = 0.1$, hence colour is the shortcut as c is more strongly correlated to y than $\hat{y}$ is. In the adversarial OOD test distribution $\mathcal{D}_{out}$, $p_c = 0.9$, i.e. every digit is more likely to be shaded the other colour. Note from Table 2 that the invariant classifier (trained on *black-and-white*

	CelebA				MF-Dominoes
	Blond/Gender		Attractive/Smiling		
Method	Ave	Worst	Ave	Worst	Worst
Naive-Class	0.642	0.294	0.625	0.215	0.000
JTT	0.653	0.283	0.647	0.300	
Chroma-VAE ($\mathbf{z}_2$-clf)	**0.820**	**0.544**	**0.669**	**0.531**	**0.840**

Table 3: Average and worst-group test accuracy for the various benchmark tasks. Chroma-VAE beats all the baselines on worst-group accuracy significantly.

images) attains similar performance on both $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$, confirming that degraded performance on $\mathcal{D}_{out}$ is solely due to shortcut learning.

The main takeaway from our results in Table 2 is that **Chroma-VAE vastly outperforms all other methods under the adversarial distribution** $\mathcal{D}_{out}$, suggesting that the $\mathbf{z}_2$-classifier is **learning from a largely shortcut-invariant representation** of the data. Accordingly, the $\mathbf{z}_1$-classifier performs just as *poorly* as the naive methods, suggesting the shortcut is heavily present in $\mathbf{z}_1$. This is further confirmed by the partial reconstructions shown in Figure 4b (further examples can be found in Appendix B). Samples of $\tilde{\mathbf{x}}_1$ are colour-invariant whereas samples of $\tilde{\mathbf{x}}_2$ are digit-invariant, suggesting that the bulk of colour representation has been isolated in $\mathbf{z}_1$.

In addition, both the $\mathbf{z}_2$-classifier and $\tilde{\mathbf{x}}_2$-classifier perform well. In particular, we noticed that simpler models work better as the $\mathbf{z}_2$-classifier — MLPs with smaller hidden layers have better accuracies, and our top performance is a k-nearest neighbor classifier with $k = \sqrt{N}$.

As further evidence that the shortcut is represented by $\mathbf{z}_1$, we create $\mathcal{D}'_{tr}$, a dataset that is identical to the training set $\mathcal{D}_{tr}$ except that every image has its colour flipped. We then measure $\log |\mathbf{z}_{(i)} - \mathbf{z}'_{(i)}|$ for every dimension i of the latent space. In Figure 4c, observe that the dimensions of $\mathbf{z}_2$ have a smaller difference than dimensions of $\mathbf{z}_1$, suggesting colour (shortcut) invariance is preserved in $\mathbf{z}_2$.

Next, this experiment highlights our finding that **generative models alone are insufficient.** Neither the VAE classifier (Naive-VAE-Class) nor the independent VAE + classifier (Naive-Independent) improved on the ERM model (Naive-Class). While our work motivated VAEs specifically from observations about the information bottleneck, we note that the community at-large has hoped [10] that *simply* having a generative component might incentivize the model to learn a comprehensive representation of the data-generating factors, and not just the minimum compression necessary for small training loss. Our results show that this is not true. This failure is not limited to hybrid models — further results in Section B.1 show that class-conditional generative classifiers fare just as poorly.

Detailed comparison with JTT JTT has two hyperparameters: (a) the number of epochs T with which the initial classifier was trained, and (b) the factor α by which initially mis-classified examples are upweighted. We tested various combinations of these two hyperparameters (see Appendix A) and Figure 4d shows the results of this ablation. In particular, we can see that while it is possible to choose hyperparameters such that JTT performs better than Chroma-VAE on *one* of the two test distributions, JTT does not outperform Chroma-VAE on *both*.

5.2 Benchmark Datasets

We consider two benchmark `CelebA` tasks: (1) predicting **Blond** where **Gender** is the spurious feature [36, 26], and (ii) predicting **Attractive** where **Smiling** is the spurious feature [25]. We also consider the `MF-Dominoes` dataset [33], where input images consist of `MNIST` digits (0 or 1) concatenated with `FashionMNIST` objects (coat or dress), with the FashionMNIST object being the true label. In all of these experiments, we use the *harder* setting from Lee et al. [25], where the spurious feature is *completely correlated* with the label at train time (by filtering out the two minority groups). Chroma-VAE is naturally suited for this regime, as complete correlation allows the spurious feature to be simultaneously compressible and highly discriminative.

The results are summarized in Table 3. All results were tuned on average accuracy. **Chroma-VAE outperforms the naive discriminative model, as well as JTT,** on both average and worst-group

(a) (b)

Figure 5: **(a)** `CelebA` example. The original example is **Attractive** but not **Smiling**. All samples of $\tilde{\mathbf{x}}_1$ *(top right)* are not smiling, whereas samples of $\tilde{\mathbf{x}}_2$ *(bottom right)* may or may not be smiling, suggesting that $\mathbf{z}_2$ learns "smile-invariance". **(b)** `MNIST-FashionMNIST` example. $\mathbf{z}_1$ learns the shortcut ("1" digit) and $\mathbf{z}_2$ retains the core feature (shirt image).

Method	Ave. Acc.	Worst Acc.
Naive-Class	**85.0**	10.3
JTT	64.2	52.3
Chroma-VAE ($\mathbf{z}_1$-`clf`)	60.2	47.3
Chroma-VAE ($\mathbf{z}_2$-`clf`)	59.9	**57.8**

Table 4: Average and worst-group accuracy on the test set.

accuracy. Similarly, the fact that the $\mathbf{z}_1$-classifier does *worse* than a naive discriminative model suggests that the shortcut has been preferentially encoded by $\mathbf{z}_1$. Partial reconstructions for a test example from either dataset, shown in Figure 5, confirm these results.

Detailed Comparison with JTT On the `CelebA` dataset, Table 3 shows that JTT underperformed. This is primarily due to the fact that these experiments are performed in the complete correlation setting (no minority groups). As such, JTT is unable to leverage on misclassification of minority examples to train the final model. This experiment highlights a significant difference between group robustness methods and Chroma-VAE. Group robustness methods like JTT tend to perform more poorly when there fewer minority examples, as they rely on these examples to teach the model what to learn. In contrast, Chroma-VAE leverages representation learning of shortcuts by the classifier. Stronger correlation between the shortcut and true signals means that it is more likely for the shortcut to be preferentially encoded by the VAE classifier, since the shortcut is now a perfect predictor of the label. As such, Chroma-VAE will do better in learning a shortcut-invariant representation.

5.3 Chest X-Rays

We consider the prediction of pneumonia from frontal chest X-ray scans, a well-known example of shortcut learning [50, 34, 20]. The shortcut is the background of the image, which is spuriously correlated with the hospital as different hospitals have different machines, lightning conditions, etc. — resulting in different background artifacts. This poses a problem for datasets containing X-rays from multiple hospitals with differing rates of pneumonia diagnosis.

There is no benchmark dataset and different authors have created their own datasets from existing public or private collections of X-ray repositories. Likewise, we make a training set ($N = 20$K) with roughly equal numbers of X-rays from the National Institutes of Health Clinical Center (NIH) dataset [46] and the Chexpert dataset [18], such that 90% of NIH images are from the negative class and 90% of Chexpert images are from the positive class.

Table 4 shows the results. Examples of partial reconstructions and a qualitative analysis can be found in Appendix B. **The main takeaway from this experiment is that our method is viable on a natural, real-world data set**. Compared to the other approaches, Chroma-VAE has the highest worst-group accuracy. However, we note that it does suffer from lower average accuracy.

6 Discussion and Conclusion

We propose Chroma-VAE, inspired by the empirical observation that shortcuts, being the most efficient and predictive compression of the data, are often preferentially encoded by deep neural networks. Our model makes use of a generative model to train a second classifier on the latent representation of the image where the shortcut is abstracted away. We demonstrate the efficacy of Chroma-VAE on several benchmark and real-world shortcut learning tasks. Finally, we conclude with a few promising avenues for future work:

Learning diverse ensembles with ≥ 2 classifiers In this work, we only considered separating two sets of features (shortcut and true signal). However, Chroma-VAE could be extended to settings where there are multiple relevant predictive features. To this end (and truer to the chromatograph metaphor), Chroma-VAE can be extended to iteratively learn a diverse ensemble of predictors.

Prior on $\mathbf{z}$ Imposing a prior on $\mathbf{z}$ can help to further disentangle the features learnt by $\mathbf{z}_1$ and $\mathbf{z}_2$. One suggestion is to enforce an orthogonal loss component on $\mathbf{z}$, i.e. by minimizing $\langle \nabla_{\mathbf{z}_1} y, \nabla_{\mathbf{z}_2} y \rangle$.

Acknowledgments and Disclosure of Funding

We would like to thank Marc Finzi, Pavel Izmailov, and Wesley Maddox for helpful comments. This research is supported by NSF CAREER IIS-2145492, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, NSF 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science, Meta Core Data Science, Google AI Research, BigHat Biosciences, Capital One, and an Amazon Research Award.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. arXiv preprint arXiv:1907.02893, 2019.

[2] Marcus A Badgeley, John R Zech, Luke Oakden-Rayner, Benjamin S Glicksberg, Manway Liu, William Gale, Michael V McConnell, Bethany Percha, Thomas M Snyder, and Joel T Dudley. Deep learning predicts hip fracture using confounding patient and healthcare variables. NPJ digital medicine, 2(1):1–10, 2019.

[3] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. PLoS computational biology, 14(12): e1006613, 2018.

[4] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In Proceedings of the European conference on computer vision (ECCV), pages 456–473, 2018.

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Joint European conference on machine learning and knowledge discovery in databases, pages 387–402. Springer, 2013.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Conference on fairness, accountability and transparency, pages 77–91. PMLR, 2018.

[7] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In International Conference on Machine Learning, pages 2189–2200. PMLR, 2021.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In International conference on machine learning, pages 1180–1189. PMLR, 2015.

[9] 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. arXiv preprint arXiv:1811.12231, 2018.

[10] 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, 2(11):665–673, 2020.

[11] Patrick J Grother, Patrick J Grother, P Jonathon Phillips, and George W Quinn. Report on the evaluation of 2D still-image face recognition algorithms. Citeseer, 2011.

[12] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. arXiv preprint arXiv:1803.02324, 2018.

[13] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In International Conference on Machine Learning, pages 1929–1938. PMLR, 2018.

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261, 2019.

[15] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations, 2017.

[16] Dirk Hovy and Anders Søgaard. Tagging performance correlates with author age. In Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers), pages 483–488, 2015.

[17] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869, 2014.

[18] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, 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, pages 590–597, 2019.

[19] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In International Conference on Machine Learning, pages 4615–4630. PMLR, 2020.

[20] Sarah Jabbour, David Fouhey, Ella Kazerooni, Michael W Sjoding, and Jenna Wiens. Deep learning applied to chest x-rays: Exploiting and preventing shortcuts. In Machine Learning for Healthcare Conference, pages 750–782. PMLR, 2020.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[23] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Advances in neural information processing systems, pages 3581–3589, 2014.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[25] Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Learning from underspecified data. arXiv preprint arXiv:2202.03418, 2022.

[26] Evan Z 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. In International Conference on Machine Learning, pages 6781–6792. PMLR, 2021.

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15(2018):11, 2018.

[28] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

[29] Mazda Moayeri, Phillip Pope, Yogesh Balaji, and Soheil Feizi. A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. arXiv preprint arXiv:2201.10766, 2022.

[30] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In International Conference on Machine Learning, pages 4723–4732. PMLR, 2019.

[31] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. arXiv preprint arXiv:2007.02561, 2020.

[32] Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In Proceedings of the ACM conference on health, inference, and learning, pages 151–159, 2020.

[33] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. arXiv preprint arXiv:2202.04414, 2022.

[34] Aahlad Puli, Lily H Zhang, Eric K Oermann, and Rajesh Ranganath. Predictive modeling in the presence of nuisance-induced spurious correlations. arXiv preprint arXiv:2107.00520, 2021.

[35] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? arXiv preprint arXiv:1806.00451, 2018.

[36] 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. arXiv preprint arXiv:1911.08731, 2019.

[37] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In International Conference on Machine Learning, pages 8346–8356. PMLR, 2020.

[38] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. arXiv preprint arXiv:1805.09190, 2018.

[39] 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 Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

[40] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017.

[41] Fabian H Sinz, Xaq Pitkow, Jacob Reimer, Matthias Bethge, and Andreas S Tolias. Engineering a less artificial intelligence. Neuron, 103(6):967–979, 2019.

[42] Nimit S Sohoni, Jared A Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. arXiv preprint arXiv:2011.12945, 2020.

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[44] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In European Conference on Computer Vision, pages 580–599. Springer, 2020.

[45] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In 2015 IEEE Information Theory Workshop (ITW), pages 1–5. IEEE, 2015.

[46] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2097–2106, 2017.

[47] Mika Westerlund. The emergence of deepfake technology: A review. Technology Innovation Management Review, 9(11), 2019.

[48] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. arXiv preprint arXiv:2006.09994, 2020.

[49] Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet, Timothy J Hazen, and Alessandro Sordoni. Increasing robustness to spurious correlations using forgettable examples. arXiv preprint arXiv:1911.03861, 2019.

[50] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. PLoS medicine, 15(11):e1002683, 2018.

[51] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In 32nd Conference on Neural Information Processing Systems (NeurIPS), 2018.

[52] Roland S Zimmermann, Lukas Schott, Yang Song, Benjamin A Dunn, and David A Klindt. Score-based generative classifiers. arXiv preprint arXiv:2110.00473, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The experiments in Section 3 and Section 5 support the claims we made.
   (b) Did you describe the limitations of your work? [Yes] Limitations are addressed in experimental findings in Appendix 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Potential societal impact of our work is discussed in Section C.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Section 5 and Appendix A describe the experimental details necessary to reproduce our results. We will release our code accompanying the public release of our paper.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We have generally found our results to be robust to repeated random runs.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We note that the computational resources we used are typical of most research using the same models and datasets.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] All datasets and models used were cited in Section 5 or Appendix A.
   (b) Did you mention the license of the assets? [No] All these datasets and models are well-known and broadly used. We refer readers to the original papers for licenses.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] This only applies to `CelebA`, discussed in the original paper.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] This only applies to `CelebA`, discussed in the original paper.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Experimental Details

Section 3: CelebA (Synthetic Patch) We use standard CNN architecture. The encoder contains 4 convolutional layers, followed by fully-connected layers to μ_ϕ and σ^2_ϕ. The decoder contains a fully-connected layer, followed by 4 deconvolutional layers. The Adam optimizer [21] is used with a learning rate of 0.0005. The hyperparameters are $\dim(z) = 128$ and $z_p = 0.5$.

Section 5: ColouredMNIST We use standard CNN architecture. The encoder contains 4 convolutional layers, followed by fully-connected layers to μ_ϕ and σ^2_ϕ. The decoder contains a fully-connected layer, followed by 4 deconvolutional layers. The Adam optimizer is used with a learning rate of 0.001. The hyperparameters are $\dim(z) = 48$ and $z_p = 0.75$. We performed a hyperparameter sweep for JTT with $T \in \{1, 3, 5, 10\}$ and $\alpha \in \{2, 5, 50, 100\}$.

Section 5: CelebA and MF-Dominoes We use standard CNN architecture. The encoder contains 4 convolutional layers, followed by fully-connected layers to μ_ϕ and σ^2_ϕ. The decoder contains a fully-connected layer, followed by 4 deconvolutional layers. The Adam optimizer is used with a learning rate of 0.0005. For `CelebA`, the hyperparameters are $\dim(z) = 128$ and $z_p = 0.5$. For `MF-Dominoes`, the hyperparameters are $\dim(z) = 8$ and $z_p = 0.5$. We performed a hyperparameter sweep for JTT with $T \in \{1, 3, 5, 10\}$ and $\alpha \in \{2, 5, 50, 100\}$.

Section 5: Chest X-rays Following prior work [50], we use a Densenet (pretrained on ImageNet) [17] for the encoder, along with a standard CNN decoder (containing a fully-connected layer, followed by 4 deconvolutional layers). The Adam optimizer is used with a learning rate of 0.0001. The hyperparameters are $\dim(z) = 128$ and $z_p = 0.5$.

B Additional Results

Method	$\mathcal{D}_{tr}$	$\mathcal{D}_{neut}$	$\mathcal{D}_{anti}$
Invariant	95.58	94.57	92.07
Naive-Class	**99.83**	90.90	4.45
Naive-VAE-Class	99.65	89.47	4.69
Naive-Independent	99.64	91.49	9.86
Class-Conditional	93.52	61.19	3.95
JTT	99.62	**92.76**	6.52
Chroma-VAE ($\mathbf{z}_1$-`clf`)	99.64	92.33	6.03
Chroma-VAE ($\mathbf{z}_2$-`clf`)	95.57	91.82	**87.24**

Table 5: Predictive accuracy on the test set. On $\mathcal{D}_{anti}$, Chroma-VAE improves on the naive discriminative classifier by a factor of 20, showing that it is learning from a shortcut-invariant representation.

B.1 Synthetic Patch on CelebA

We consider the same task as in Section 3.1, where we predict blond hair, with a synthetic shortcut (blue patch superimposed onto the positive class with probability 0.9). We evaluate performance on three test environments:

- **Training Distribution** ($\mathcal{D}_{tr}$): the patch is added to all members of the positive class
- **Neutral Distribution** ($\mathcal{D}_{neut}$): no patches are added
- **Adversarial Distribution** ($\mathcal{D}_{anti}$): the patch is added to all members of the *negative* class

Table 5 summarizes the results. **Chroma-VAE vastly outperforms all other methods under the adversarial distribution**. The $\mathbf{z}_2$-classifier has the same performance as the invariant classifier on the in-distribution $\mathcal{D}_{tr}$, implying that it did not benefit (unfairly) from shortcut representations, unlike the naive baselines. However, the fact that the $\mathbf{z}_2$-classifier is still outperformed by the invariant classifier on $\mathcal{D}_{anti}$ and $\mathcal{D}_{neut}$ suggests that not all desired features were captured by $\mathbf{z}_2$. This implies

that the maximally-useful compression of the data in $\mathbf{z}_1$ primarily encodes the shortcut, but also other useful correlations.

Figure 6 shows partial reconstruction samples. The first image represents an example of the negative class (non-blonde hair), the second image is an example of the positive class with the shortcut, and the third image is an example of the positive class *without* the shortcut (which happens with probability 0.1 in the training set). Similar to Figure 3, samples of $\tilde{\mathbf{x}}_1$ do not resemble the original image, but always encode the shortcut patch if it exists in the original image. Conversely, samples of $\tilde{\mathbf{x}}_2$ bear great resemblance to the original image but may or may contain the shortcut — confirming that they are shortcut-invariant.

Comparison with JTT While JTT does well on $\mathcal{D}_{neut}$, it performs just as poorly as the naive methods on $\mathcal{D}_{anti}$. This highlights a key difference in the approaches of JTT and Chroma-VAE. JTT relies on having sufficient examples in the minority group to upweigh and correct the second classifier. In the case of the adversarial distribution, doing so is not possible because there are simply no examples in the training set with a negative label and the shortcut present. Conversely, Chroma-VAE relies on learning a shortcut-invariant representation of the data, and is therefore impervious to adversarial distribution shifts as such.

Ablations on β and relationship to disentanglement Our method relates to the idea of learning disentangled latent representations. β-VAE, proposed by Higgins et al. [15], introduces the hyperparameter β (the coefficient on the Kullback-Liebler (KL) divergence term in (1)) as a means of learning disentangled representations. We ablate for this by learning the $\mathbf{z}_2$-classifier with different values of β. Table 6 shows the results of this ablation. Increasing $\beta > 1$ result in degraded performance on $\mathcal{D}_{anti}$ similar to naive baselines, which seems counterintuitive as further restricting the latent capacity of the VAE should encourage it to encode the shortcut. We postulate that this could be because increasing $\beta > 1$ restricts *both* $\mathbf{z}_1$ and $\mathbf{z}_2$, and might instead have encouraged $\mathbf{z}_2$ to encode the shortcut when it normally would not have.

Method	$\mathcal{D}_{tr}$	$\mathcal{D}_{neut}$	$\mathcal{D}_{anti}$
$\beta = 100$	99.90	88.53	3.09
$\beta = 50$	99.74	88.58	14.98
$\beta = 100$	99.76	90.67	7.60
$\beta = 1$	95.57	91.82	**87.24**
$\beta = 0.5$	95.78	91.88	84.88

Table 6: Performance of the $\mathbf{z}$-classifier with different values of β. With large β values, the $\mathbf{z}$-classifier does not perform better than naive baselines. Conversely, even when $\beta < 1$, $\mathbf{z}_2$ still learns a shortcut-invariant representation of the data.

Figure 6: **For each row, left to right: (a)** original image, **(b)** samples of partial reconstructions $\tilde{\mathbf{x}}_1$, **(c)** samples of partial reconstructions $\tilde{\mathbf{x}}_2$.

B.2 ColouredMNIST: Partial Reconstructions

Figure 7 shows partial reconstruction samples on three additional images. We observe the same patterns as in Figure 4, where samples of $\tilde{\mathbf{x}}_1$ preserve the colour of the original image but not the digit, whereas samples of $\tilde{\mathbf{x}}_2$ preserve the digit of the original shape but is colour-invariant.

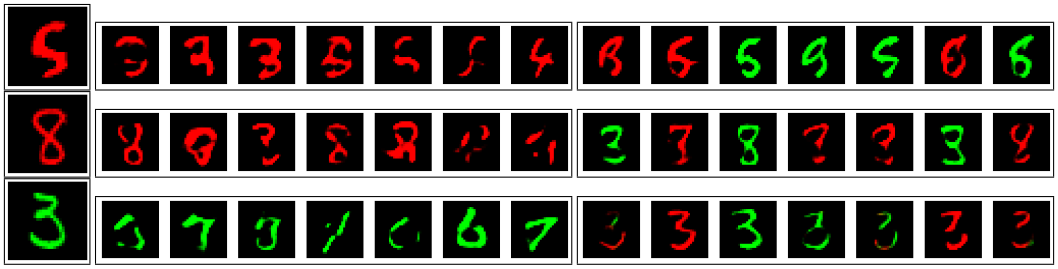


Figure 7: **For each row, left to right: (a)** original image, **(b)** samples of partial reconstructions $\tilde{\mathbf{x}}_1$, **(c)** samples of partial reconstructions $\tilde{\mathbf{x}}_2$.

B.3 Chest X-Ray: Partial Reconstructions

Figure 8 shows the reconstruction as well as partial reconstruction samples on a test example. Similar to partial reconstructions on the `CelebA` dataset, samples of $\tilde{\mathbf{x}}_1$ are more diverse and less likely to resemble the original image, whereas samples of $\tilde{\mathbf{x}}_2$ resemble the original image more, especially around the center region of the image (where true predictive signals for pneumonia are located). This confirms that $\mathbf{z}_1$ is more likely to have encoded the shortcut (X-ray background) whereas $\mathbf{z}_2$ is more likely to have retained true predictive signals for pneumonia.

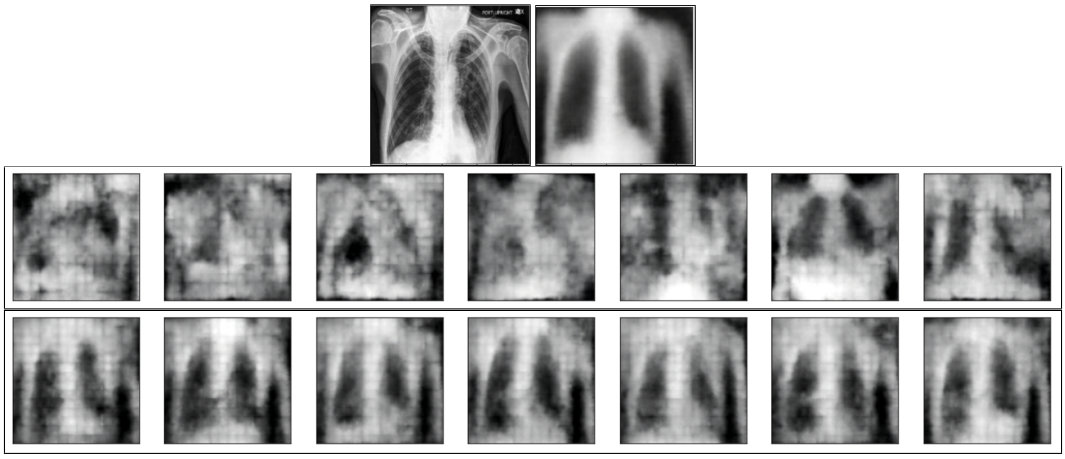


Figure 8: **Top to bottom, left to right: (a)** original image, **(b)** reconstruction, **(c)** samples of partial reconstructions $\tilde{\mathbf{x}}_1$, **(d)** samples of partial reconstructions $\tilde{\mathbf{x}}_2$.

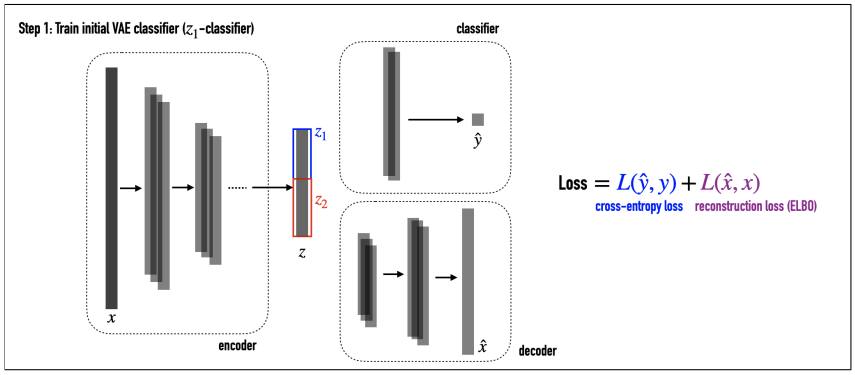
C Broader Societal Impact

Positive Sources of Impact By learning shortcut-invariant predictive signals, Chroma-VAE is robust to distribution shifts and performs well when deployed on OOD test cases, especially in high-stakes domains where errors are costly, such as the pneumonia prediction application in Section 5.3 [50]. Furthermore, the partial reconstructions produced by Chroma-VAE can also indicate if shortcut learning is occurring in the first place, as they can be examined to identify traits or features that are invariant.

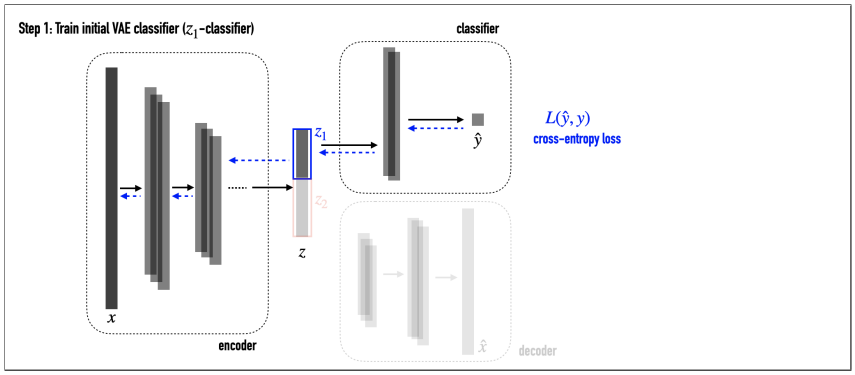
Potential for Abuse Deep generative models can be used to produce *deepfakes* [47], which are samples generated by the model that are realistic enough to resemble real-life examples, e.g. fake faces produced by a generative model trained on `CelebA`. Chroma-VAE can be abused in a similar (and potentially more dangerous) way, as partial reconstructions can be generated to keep certain traits while changing others, by concatenating $\mathbf{z}_1$ and $\mathbf{z}_2$ from different samples. For example, partial reconstructions produced by Chroma-VAE trained on gender labels might be used to generate "counterfactual deepfakes", e.g. a person with the opposite gender. Combating deepfakes is an active area of research, and such techniques can be used to detect Chroma-VAE deepfakes as well.

D Chroma-VAE: Diagram of Model

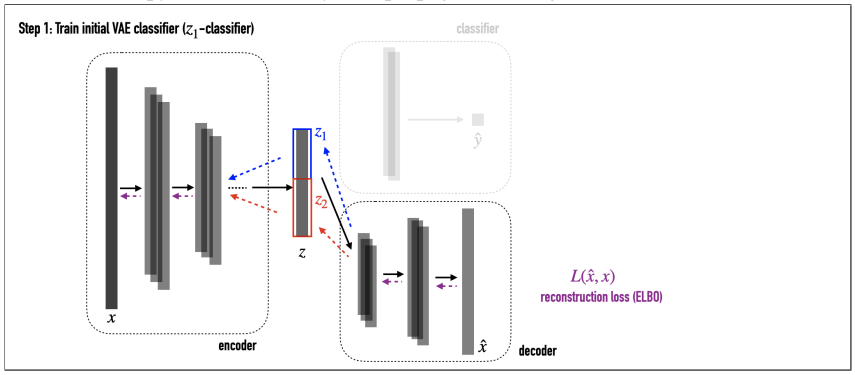
Figure 9 depicts Chroma-VAE and its training procedure. In the diagrams below, the forward-facing solid arrows represent the forward pass and the backwards-facing dotted arrows represent the backpropagation of gradients.



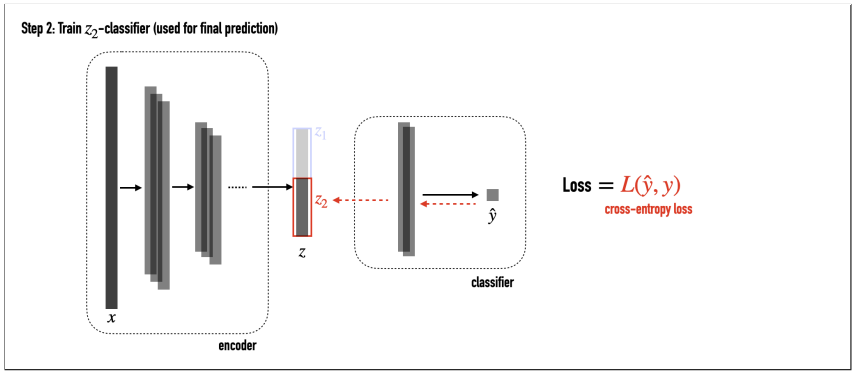
(a) In the first stage of the training procedure, the encoder, decoder, and classifier are all trained. The loss function consists of two terms: the cross-entropy loss and the reconstruction loss.



(b) The classifier head is attached to $\mathbf{z}_1$ only (*not* the entire latent embedding $\mathbf{z}$). As such, the cross-entropy loss term is only backpropagated through $\mathbf{z}_1$.



(c) The decoder is attached to the entire latent embedding $\mathbf{z}$. As such, the reconstruction loss term is backpropagated through both $\mathbf{z}_1$ and $\mathbf{z}_2$. Consequently, putting both terms of the loss function together, $\mathbf{z}_1$ learns to encode shortcut features whereas $\mathbf{z}_2$ encodes salient non-shortcut features of the input image.



(d) In the second stage of the training procedure, we discard the original classifier head (attached to $\mathbf{z}_1$) and attach a new one to $\mathbf{z}_2$ only instead. We train this classifier using the same cross-entropy loss, except that gradients are now being backpropagated to $\mathbf{z}_2$. Note that the encoder and decoder are *fixed* at this stage, and so $\mathbf{z}$ itself is fixed. Only the parameters of the new classifier head are being updated. Since $\mathbf{z}_2$ was trained to encode non-shortcut representations, the classifier learns to predict the label using non-shortcut features only.

Figure 9: Diagram of Chroma-VAE's training procedure, showing loss terms and backpropagated gradients.