UNDERSTANDING THE FAILURE MODES OF OUT-OF-DISTRIBUTION GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Empirical studies suggest that machine learning models often rely on features, such as the background, that may be spuriously correlated with the label only during training time, resulting in poor accuracy during test-time. In this work, we identify the fundamental factors that give rise to this behavior, by explaining why models fail this way *even* in easy-to-learn tasks where one would expect these models to succeed. In particular, through a theoretical study of gradient-descent-trained linear classifiers on some easy-to-learn tasks, we uncover two complementary failure modes. These modes arise from how spurious correlations induce two kinds of skews in the data: one *geometric* in nature, and another, *statistical* in nature. Finally, we construct natural modifications of image classification datasets to understand when these failure modes can arise in practice. We also design experiments to isolate the two failure modes when training modern neural networks on these datasets.

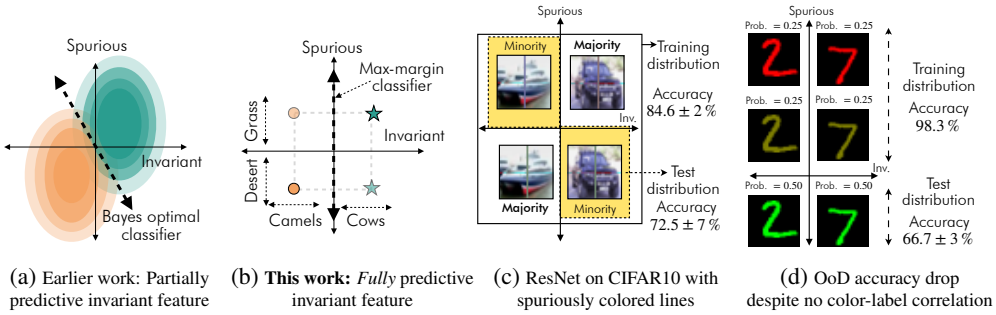
1 INTRODUCTION

A machine learning model in the wild (e.g., a self-driving car) must be prepared to make sense of its surroundings in rare conditions that may not have been well-represented in its training set. This could range from conditions such as mild glitches in the camera to strange weather conditions. This out-of-distribution (OoD) generalization problem has been extensively studied within the framework of the domain generalization setting (Blanchard et al., 2011; Muandet et al., 2013). Here, the classifier has access to training data sourced from multiple "domains" or distributions, but no data from test domains. By observing the various kinds of shifts exhibited by the training domains, we want the the classifier can learn to be robust to such shifts.

The simplest approach to domain generalization is based on the Empirical Risk Minimization (ERM) principle (Vapnik, 1998): pool the data from all the training domains (ignoring the "domain label" on each point) and train a classifier by gradient descent to minimize the average loss on this pooled dataset. Alternatively, many recent studies (Ganin et al., 2016; Arjovsky et al., 2019; Sagawa et al., 2020a) have focused on designing more sophisticated algorithms that do utilize the domain label on the datapoints e.g., by enforcing certain representational invariances across domains.

A basic premise behind pursuing such sophisticated techniques, as emphasized by Arjovsky et al. (2019), is the empirical observation that ERM-based gradient-descent-training (or for convenience, just ERM) fails in a characteristic way. As a standard illustration, consider a cow-camel classification task (Beery et al., 2018) where the background happens to be spuriously correlated with the label in a particular manner only during training – say, most cows are found against a grassy background and most camels against a sandy one. Then, during test-time, if the correlation is completely flipped (i.e., *all* cows in deserts, and *all* camels in meadows), one would observe that accuracy of ERM drops drastically. Evidently, ERM, in its unrestrained attempt at fitting the data, indiscriminately relies on all kinds of informative features, including unreliable *spurious features* like the background. However, an algorithm that carefully uses domain label information can hope to identify and rely purely on *invariant features* (such as the shape of the animal) that are consistently informative across domains.

While the above narrative is an oft-stated motivation behind developing sophisticated OoD generalization algorithms, there has been little formal explanation as to *why* ERM fails in this characteristic way. Existing works (Sagawa et al., 2020b; Tsipras et al., 2019; Arjovsky et al., 2019) provide



(a) Earlier work: Partially predictive invariant feature

(b) **This work:** *Fully* predictive invariant feature

(c) ResNet on CIFAR10 with spuriously colored lines

(d) OoD accuracy drop despite no color-label correlation

Figure 1: **Unexplained OoD failure**: Existing theory can explain why classifiers rely on the spurious feature when the invariant feature is in itself not informative enough (**Fig 1a**). But when invariant features are fully predictive of the label, one would expect a (max-margin) classifier to easily ignore spurious correlations (like in the four-point-dataset of **Fig 1b**; also see Sec 3). Yet, in practice, why do classifiers (including the max-margin) rely on the spurious feature in so many real-world settings where the shapes are perfectly informative of the object label (e.g., **Fig 1c**)? We identify two fundamental factors behind this behavior. In doing so, we also identify (and explain) other kinds of vulnerabilities like in **Fig 1d** (see Sec 4).

valuable answers to this question through concrete theoretical examples; however, their examples critically rely on certain factors (such as invariant features that are only partially predictive of the label – see Fig 1a) to make the task difficult enough for ERM to rely on the spurious features. Surprisingly though, ERM relies on spurious features even in much easier-to-learn tasks where these complicating factors are absent – such as in tasks with fully predictive invariant features e.g., the Fig 1c or the Waterbirds/CelebA examples in Sagawa et al. (2020a) or in any real-world situation where the object shape perfectly determines the label. This failure in easy-to-learn tasks, as we argue later, is not straightforward to explain (see Fig 1b for brief idea). This evidently implies that there must exist factors more general and fundamental than those known so far, that cause ERM to fail.

Our goal in this work is to uncover these fundamental factors behind the failure of ERM. The hope is that this will provide a vital foundation for future work to reason about OoD generalization. Indeed, recent empirical work (Gulrajani & Lopez-Paz, 2020) has questioned whether existing alternatives necessarily outperform ERM on OoD tasks; however, due to a lack of theory, it is not clear how to hypothesize about when/why one algorithm would outperform another here. Through our theoretical study, future work can hope to be better positioned to precisely identify the key missing components in these algorithms, and bridge these gaps to better solve the OoD generalization problem.

Our contributions. To identify the most fundamental factors causing OoD failure, our strategy is to (a) study tasks that are "easy" to succeed at, and (b) to demonstrate that ERM relies on spurious features *despite* how easy the tasks are. More concretely:

1. We formulate a set of constraints on how our tasks must be designed so that they are easy to succeed at (e.g., the invariant feature must be fully predictive of the label). Notably, this class of *easy-to-learn tasks* provides both a theoretical test-bed for reasoning about OoD generalization and also a simplified empirical test-bed. In particular, this class encompasses simplified MNIST and CIFAR10-based classification tasks where we establish empirical failure of ERM.
2. We identify two complementary mechanisms of failure of ERM that arise from how spurious correlations induce two kinds of skews in the data: one that is *geometric* and the other *statistical*. In particular, we theoretically isolate these failure modes by studying linear classifiers trained by gradient descent (on logistic/exponential loss) and its infinite-time-trained equivalent, the max-margin classifier (Soudry et al., 2018; Ji & Telgarsky, 2018) on the easy-to-learn-tasks.
3. We also show that in any easy-to-learn task that does not have these geometric or statistical skews, these models do not rely on the spurious features. This suggests that these skews are not just a sufficient but also a necessary factor for failure of these models in easy-to-learn tasks.
4. To empirically demonstrate the generality of our theoretical insights, we (a) experimentally validate these skews in a range of MNIST and CIFAR10-based tasks, and (b) demonstrate their effects on fully-connected networks (FNNs) and ResNets. We also identify and explain failure in scenarios where standard notions of spurious correlations do not apply (see Fig 1d).

2 RELATED WORK

Understanding failure of ERM. While the fact that ERM relies on spurious correlations has become empirical folk wisdom, only a few studies have made efforts to carefully model this. Sagawa et al. (2020b) explain how the max-margin classifier relies on spurious correlations in a concrete, linear setting with a single invariant feature and a single spurious feature (and many "non-sense" features). Tsipras et al. (2019) similarly explain failure of ERM in a setting with high-dimensional spurious features. Both these works provide concrete upper bounds on OoD accuracy, demonstrating failure under the crucial condition that the invariant feature is sufficiently noisy (i.e., it is only partially informative of the label). In contrast to these works, we study the max-margin classifier in an abstract setting that allows the invariant features to be high-dimensional. But more importantly, in our setting, we show how ERM relies on the spurious feature despite having an invariant feature that is fully informative of the label. Note that we however do not provide concrete OoD accuracy bounds since we study an abstract setting. Finally, we also go beyond the max-margin settings in these works to analyze the dynamics a finite-time gradient-descent trained classifier (although this particular result is only for a specific setting). We also refer the reader to an orthogonal line of theory work (Fawzi et al., 2018; Gilmer et al., 2018) that has investigated the failure of standard classifiers in the adversarial perturbation setting.

Algorithms for OoD generalization. Due to the empirical shortcomings of ERM, a wide range of sophisticated algorithms have been developed for domain generalization. The most popular strategy is to learn useful features while constraining them to have similar distributions across domains (Ganin et al., 2016; Li et al., 2018b). Other works constrain these features in a way that one can learn a classifier that is simultaneously optimal across all domains (Peters et al., 2016; Arjovsky et al., 2019; Krueger et al., 2020). As discussed in Gulrajani & Lopez-Paz (2020), there are also many other existing non-ERM based methods, including that of meta-learning (Li et al., 2018a), parameter-sharing (Sagawa et al., 2020a) and data augmentation (Zhang et al., 2018). Through their extensive empirical survey of many of the above algorithms, Gulrajani & Lopez-Paz (2020) suggest that ERM may be just as competitive as the state-of-the-art. But we must emphasize that this doesn't vindicate ERM of its failures but rather indicates that we are yet to develop a substantial improvement over ERM.

3 EASY-TO-LEARN DOMAIN GENERALIZATION TASKS

Below, we first set up the basic domain generalization setting and the idea of ERM. Then, within this setting, in Section 3.1, we will formulate a class of domain-generalization tasks that are in many aspects "easy" for the learner to succeed at (such as a fully informative invariant feature) – what makes a task "easy" will be more precisely discussed in that subsection. This discussion will prepare the ground for the later sections to show how ERM can fail even in these easy tasks, which will help uncover the fundamental factors behind its failure.

Notations. Consider an input (vector) space $\mathcal{X}$ and label space $\mathcal{Y} \in \{-1, 1\}$. For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, let $p_{\mathcal{D}}(\cdot)$ denote its probability density function (PDF). Let $\mathbb{H}$ denote a class of classifiers $h : \mathcal{X} \to \mathbb{R}$. Let the error of h on $\mathcal{D}$ be denoted as $L_{\mathcal{D}}(h) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[h(\mathbf{x}) \cdot y < 0]$.

The domain generalization setting and ERM. In the domain generalization setting, one considers an underlying class $\mathbb{D}$ of data distributions over $\mathcal{X} \times \mathcal{Y}$ corresponding to different possible domains. The learner is given training data collected from multiple distributions from $\mathbb{D}$. For an ERM-based learner in particular, the training data will be pooled together, so we can model the data as coming from a single (pooled) distribution $\mathcal{D}_{\text{train}}$, which for simplicity, can be assumed to belong to $\mathbb{D}$. Given this data, the learner outputs a hypothesis $\hat{h} \in \mathbb{H}$ that is tested on a new distribution $\mathcal{D}_{\text{test}}$ picked from $\mathbb{D}$. This can be potentially modeled by assuming that all test and training distributions are drawn from a common hyper-distribution over $\mathbb{D}$. However, this assumption becomes pointless in most practical settings (e.g., PACS (Asadi et al., 2019), VLCS (Fang et al., 2013)) where the training domains are not more than three to four in number, and therefore hardly representative of any hyper-distribution. Here, the problem becomes as hard as ensuring good performance on a worst-case test-distribution without any hyper-distribution assumption i.e., this boils down to minimizing $\max_{\mathcal{D}\in\mathbb{D}} L_{\mathcal{D}}(\hat{h})$. Indeed, most works have studied the worst-case setting, both theoretically (Sagawa et al., 2020b) and empirically (Arjovsky et al., 2019; Sagawa et al., 2020a).

Similarly, for this work, we focus on the worst-case setting and define the optimal target function $h^\star$ to be $h^\star = \arg\min_{h\in\mathbb{H}} \max_{\mathcal{D}\in\mathbb{D}} L_{\mathcal{D}}(h)$. Then, we define the features that this "robust" classifier uses as invariant features $\mathcal{X}_{\text{inv}}$ (e.g., the shape of the object), and the rest as spurious features $\mathcal{X}_{\text{sp}}$ (e.g., the background). To formalize this, we assume that there exists a mapping $\Phi : \mathcal{X}_{\text{inv}} \times \mathcal{X}_{\text{sp}} \to \mathcal{X}$ such that each $\mathcal{D} \in \mathbb{D}$ is induced by a distribution over $\mathcal{X}_{\text{inv}} \times \mathcal{X}_{\text{sp}}$ (so we can denote any $\mathbf{x}$ as $\Phi(\mathbf{x}_{\text{inv}}, \mathbf{x}_{\text{sp}})$). With an abuse of notation we will use $p_{\mathcal{D}}(\cdot)$ to also denote the PDF of the distribution over $\mathcal{X}_{\text{inv}} \times \mathcal{X}_{\text{sp}}$. Then, the fact that $\mathcal{X}_{\text{sp}}$ are features that $h^\star$ does not rely on, is mathematically stated as: $\forall \mathbf{x}_{\text{inv}}, \forall\, \mathbf{x}_{\text{sp}} \neq \mathbf{x}'_{\text{sp}},\ h^\star(\Phi(\mathbf{x}_{\text{inv}}, \mathbf{x}_{\text{sp}})) = h^\star(\Phi(\mathbf{x}_{\text{inv}}, \mathbf{x}'_{\text{sp}}))$. Finally, we note that, to make this learning problem tractable, one has to impose further restrictions; we'll detail those when we discuss the class of easy-to-learn domain generalization tasks.

Empirical failure of ERM. To guide us in constructing the easy-to-learn tasks, let us ground our study in a concrete empirical setup where an ERM-based linear classifier shows OoD failure. Specifically, consider the following Binary-MNIST based task (the first five digits and the remaining five digits form the two classes). First, we let the Φ mapping be the identity, and so $\mathbf{x} = (\mathbf{x}_{\text{inv}}, \mathbf{x}_{\text{sp}})$. Then, we let $\mathbf{x}_{\text{inv}}$ be a random ReLU features representation of the MNIST digit i.e., if $\mathbf{x}_{\text{raw}}$ represents the MNIST image, then $\mathbf{x}_{\text{inv}} = \text{ReLU}(W\mathbf{x}_{\text{raw}})$ where W is a matrix with Gaussian entries. We make this representation sufficiently high-dimensional so that the data becomes linearly separable. Next, we let the spurious feature take values in $\{+\mathcal{B}, -\mathcal{B}\}$ imitating the two possible background colors in the camel-cow dataset. Finally, on $\mathcal{D}_{\text{train}}$, for any y, we pick the image $\mathbf{x}_{\text{inv}}$ from the corresponding class and independently set the "background color" x_{sp} so that there is some spurious correlation i.e., $\Pr_{\mathcal{D}_{\text{train}}}[x_{\text{sp}} \cdot y > 0] > 0.5$. During test time however, we flip this correlation around so that $\Pr_{\mathcal{D}_{\text{test}}}[x_{\text{sp}} \cdot y > 0] = 0.0$. In this task, we observe in Fig 2a (shown later under Sec 4) that as we vary the train-time spurious correlation from none ($\Pr_{\mathcal{D}_{\text{train}}}[x_{\text{sp}} \cdot y > 0] = 0.5$) to its maximum ($\Pr_{\mathcal{D}_{\text{train}}}[x_{\text{sp}} \cdot y > 0] = 1.0$), the OoD accuracy of a max-margin classifier progressively deteriorates. (We present similar results for a CIFAR10 setting, and all experiment details in Appendix C.1.) Our goal is now to theoretically demonstrate why ERM fails this way (or equivalently, why it relies on the spurious feature) even in tasks as "easy-to-learn" as these.

3.1 Constraints defining easy-to-learn domain-generalization tasks.

To formulate *a class of easy-to-learn tasks*, we enumerate a set of constraints that the tasks must satisfy; notably, this class of tasks will encompass the empirical example described above. The motivation behind this exercise is that restricting ourselves to the constrained set of tasks yields stronger insights – it prevents us from designing complex examples where ERM is forced to rely on spurious features due to a not-so-fundamental factor. Indeed, each constraint here forbids a unique, less fundamental failure mode of ERM from occuring in the easy-to-learn tasks.

Constraint 3.1. (Fully predictive invariant features.) For all $\mathcal{D} \in \mathbb{D}$, $L_{\mathcal{D}}(h^\star) = 0$.
Arguably, our most important constraint is that the invariant features (which is what $h^\star$ purely relies on) are *perfectly informative* of the label. The motivation is that, when this is not the case (e.g., as in noisy invariant features like Fig 1a or in Sagawa et al. (2020b); Tsipras et al. (2019)), failure can arise from the fact that the spurious features provide vital extra information that the invariant features cannot provide (see App A for more formal argument). However this explanation quickly falls apart when the invariant feature in itself is fully predictive of the label.

Constraint 3.2. (Identical invariant distribution.) Across all $\mathcal{D} \in \mathbb{D}$, $p_{\mathcal{D}}(\mathbf{x}_{\text{inv}})$ is identical.
This constraint demands that the (marginal) invariant feature distribution must remain stable across domains (like in our binary-MNIST example). While this may appear to be unrealistic, (the exact distribution of the shapes of cows and camels could vary across domains), we must emphasize that it is easier to make ERM fail when the invariant features are not stable (see example in App A Fig 4a).

Constraint 3.3. (Conditional independence.) For all $\mathcal{D} \in \mathbb{D}$, $\mathbf{x}_{\text{sp}} \perp\!\!\!\perp \mathbf{x}_{\text{inv}} | y$.
This constraint reflects the fact that in the MNIST example, we chose the class label and then picked the color feature independent of the actual hand-written digit picked from that class. This prevents us from designing complex relationships between the background and the object shape to show failure (see example in App A Fig 4b).

Constraint 3.4. (Two-valued spurious features.) We set $\mathcal{X}_{\text{sp}} = \mathbb{R}$ and the support of x_{sp} in $\mathcal{D}_{\text{train}}$ is $\{-\mathcal{B}, +\mathcal{B}\}$.

The constraint[1] captures the simplicity of the cow-camel example where the background color is limited to yellow/green. Notably, this excludes failure borne out of high-dimensional (Tsipras et al., 2019) or (carefully-constructed) continuous-valued (Sagawa et al., 2020b) spurious features.

Constraint 3.5. (Identity mapping.) Φ is the identity mapping i.e., $\mathbf{x} = (\mathbf{x}_{\text{inv}}, \mathbf{x}_{\text{sp}})$.

This final constraint, also implicitly made in Sagawa et al. (2020b); Tsipras et al. (2019), prevents ERM from failing because of a hard-to-disentangle representation (see example in App A Fig 4c).

Before we connect these constraints to our main goal, it is worth mentioning their value beyond that goal. First, as briefly discussed above and elaborated in App A, each of these constraints in itself corresponds to a unique failure mode of ERM, one that is worth exploring in future work. Second, the resulting class of easy-to-learn tasks provides a theoretical (and a simplified empirical) test-bed that would help in broadly reasoning about OoD generalization. For example, any algorithm for solving OoD generalization should at the least hope to solve these easy-to-learn tasks well.

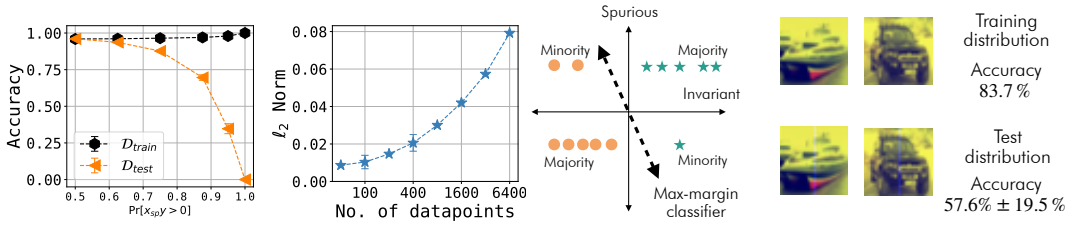
Why is it hard to show that ERM relies on the spurious feature in easy-to-learn tasks? Consider the simplest easy-to-learn 2D task. Specifically, during training we set $x_{\text{inv}} = y$ (and so Constraint 3.1 is satisfied) and x_{sp} to be $y\mathcal{B}$ with probability $p \in [0.5, 1]$ and $-y\mathcal{B}$ with probability $1-p$ (hence satisfying both Constraint 3.3 and 3.4). During test-time, the only shifts allowed are on the distribution of x_{sp} (to respect Constraint 3.2). Observe from Fig 1b that (as long as $p \neq 1$) this distribution has a support of the four points in $\{-1, +1\} \times \{-\mathcal{B}, +\mathcal{B}\}$ and is hence an abstract form of the cow-camel dataset which also has four groups of points: (a majority of) cows/camels against grass/sand and (a minority of) cows/camels against sand/grass. Fitting a max-margin classifier on $\mathcal{D}_{\text{train}}$ leads us to a simple yet key observation: *owing to the geometry of the four groups of points, the max-margin classifier has no reliance on the spurious feature despite the spurious correlation*. In other words, even though this dataset distills what seem to be the core aspects of the cow/camel dataset, we are unable to reproduce the corresponding behavior of ERM. In the next two sections, we will try to resolve this apparent paradox.

4 Failure due to geometric skews

A pivotal piece in solving this puzzle is a particular geometry underlying how real-world distributions are separated. To describe this, consider the random features representation of MNIST (i.e., $\mathbf{x}_{\text{inv}}$) and fit a max-margin classifier on it i.e., the least-norm $\mathbf{w}_{\text{inv}}$ that achieves a margin of $y \cdot (\mathbf{w}_{\text{inv}} \cdot \mathbf{x}_{\text{inv}} + b) \geq 1$ on all data (for some b). Then, we'd observe that *as the number of training points increase, the ℓ_2 norm of this max-margin classifier would grow* (see Figure 2b); similar observations also hold for CIFAR10 (see Appendix C.2). This observation (which stems from the geometry of the dataset) builds on ones that were originally made in Neyshabur et al. (2017); Nagarajan & Kolter (2017; 2019) for norms of overparameterized neural networks. Our contribution here is to empirically establish this for a linear overparameterized model and then to theoretically relate this to OoD failure.

Equipped with this observation, let us go back to study the dataset where every input $\mathbf{x}_{\text{inv}}$ is concatenated with a feature $x_{\text{sp}} \in \{-\mathcal{B}, \mathcal{B}\}$ spuriously correlated with the label in that $\Pr[x_{\text{sp}} \cdot y > 0] > 0.5$. To explain why the max-margin classifier here relies on the spurious feature, we can think of the underlying spurious correlation as inducing two disjoint groups in the whole dataset S, borrowing the terminology of Sagawa et al. (2020b): a *majority group* S_{maj} where $x_{\text{sp}} \cdot y > 0$ e.g., cows/camels with green/yellow backgrounds, and a *minority group* S_{min} where $x_{\text{sp}} \cdot y < 0$ e.g., cows/camels with yellow/green backgrounds. (Although we must emphasize that, as already noted, unlike Sagawa et al. (2020b), our setting is different as it satisfies certain key easy-to-learn constraints.) Then, the crux of the failure mechanism is as follows (visualized in Fig 2c). First, when relying on only $\mathbf{x}_{\text{inv}}$, it is cheaper to separate the small, minority group (i.e., it requires less ℓ_2 norm) than separating the whole dataset – this follows from our empirical observation in Fig 2b. This gap in costs incentivizes the max-margin classifier to use the spurious feature as a short-cut to classify a majority of the data (by setting $w_{\text{sp}} > 0$) and then using the invariant space solely to classify the minority group.

[1]Note that the discrete-value restriction holds only during training. This is so that, in our experiments, we can study two kinds of test-time shifts, one within the support $\{-\mathcal{B}, +\mathcal{B}\}$ and one outside of it (the vulnerability to both of which boils down to the level of reliance on x_{sp}).



(a) OOD failure in Sec 3 Binary-MNIST example

(b) ℓ_2 norm of max-margin

(c) Geometric failure mechanism

(d) CIFAR10 example with a spurious line

Figure 2: **Geometric skews: Fig 2a** demonstrates failure of the max-margin classifier in the easy-to-learn Binary-MNIST task of Sec 3. **Fig 2b** forms the basis for our theory in Sec 4 by showing that the random-features-based max-margin has norms increasing with the size of the dataset. In **Fig 2c**, we visualize the failure mechanism arising from the "geometric skew": with respect to the $\mathbf{x}_{\text{inv}}$-based classifier, the closest minority point is farther away than the closest majority point, which crucially tilts the max-margin classifier towards $w_{\text{sp}} > 0$. **Fig 2d**, as further discussed in Sec 4, stands as an example for how a wide range of OoD failures can be explained geometrically.

To make this more formal, let $\mathbf{w}_{\text{all}} \in \mathcal{X}_{\text{inv}}$ denote the least-norm vector that (a) lies in the invariant space and (b) classifies all of S by a margin of at least 1. Next, let $\mathbf{w}_{\text{min}} \in \mathcal{X}_{\text{inv}}$ similarly denote a least-norm, invariant-space-vector that classifies all of S_{min} by a margin of at least 1. Then, when there is a spurious correlation, we empirically know from Fig 2b that $\|\mathbf{w}_{\text{min}}\|/\|\mathbf{w}_{\text{all}}\| \ll 1$ since the minority group would be quite small (we informally refer to the gap between these values as a geometric skew). Our result below then provides lower and upper bounds on $|w_{\text{sp}}|$ that are larger for smaller values of $\|\mathbf{w}_{\text{min}}\|/\|\mathbf{w}_{\text{all}}\|$. For readability, we'll state only an informal version of our theorem below, and in particular for $\mathcal{B} = 1$. In Appendix B.1, we present the full, precise result, for generic value $\mathcal{B}$, along with the proof.

Theorem 4.1. ***(informal)*** *Let $\mathbb{H}$ be the set of linear classifiers, $h(x) = \mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + w_{\text{sp}}x_{\text{sp}} + b$. Then for any task satisfying all the constraints in Sec 3.1 with $\mathcal{B} = 1$, the max-margin classifier satisfies:*

$$1 - 2\sqrt{\|\mathbf{w}_{\text{min}}\|/\|\mathbf{w}_{\text{all}}\|} \leq w_{\text{sp}} \leq \frac{1}{\|\mathbf{w}_{\text{min}}\|/\|\mathbf{w}_{\text{all}}\|} - 1.$$

A salient aspect of this result is that it explains the varying dynamics between the underlying spurious correlation and spurious-feature-reliance in the classifier. First, as the correlation increases ($Pr_{\mathcal{D}_{\text{train}}}[x_{\text{sp}} \cdot y > 0] \to 1.0$), the size of the minority group decreases to zero. Then, we empirically know that $\|\mathbf{w}_{\text{min}}\|/\|\mathbf{w}_{\text{all}}\|$ progressively shrinks all the way down to 0. Then, we can invoke the lower bound which implies that w_{sp} grows to ≈ 1. This implies serious vulnerability to the test-time shifts: any flip in the sign of the spurious feature can reduce the original margin of ≈ 1 by a value of $2|w_{\text{sp}}x_{\text{sp}}| \approx 2$ (since $\mathcal{B} = 1$ here) making the margin negative (implying misclassification). On the other hand, when spurious correlations diminish ($Pr_{\mathcal{D}_{\text{train}}}[x_{\text{sp}} \cdot y > 0] \to 0.5$) , the value of $\|\mathbf{w}_{\text{min}}\|$ grows comparable to $\|\mathbf{w}_{\text{all}}\|$ (although not exactly as large), and our upper bound suggests that the spurious component must shrink towards ≈ 0, thereby implying robustness to these shifts.

Broader empirical implications. While our theorem explains failure in linear, easy-to-learn settings, informally, the underlying geometric argument is more generally applicable. To demonstrate this, we identify and enumerate below a variety of unique settings where we can intuitively understand the failure of a (neural network) model through Theorem 4.1 (we defer these explanations and experimental details to Appendix C.3). Notably, through these examples, we also explain the other failure modes that arise from breaking the easy-to-learn constraints, as discussed in Sec 3.1.

1. In Fig 1c, we add a line to CIFAR10 images with its color spuriously correlated with the class only during training. The $\gtrsim 10\%$ OoD accuracy drop of a ResNet here, we argue, arises from the fact that it takes greater norms for the ResNet to fit larger proportions CIFAR10 (see Appendix C.1).
2. In Fig 1d, we fade the first channel of a two-channel Binary-MNIST dataset during training, and the second channel during testing, thereby creating a color shift. Crucially, even though there is

no correlation between the label and the color of the images, an FNN drops by $\gtrsim 30\%$ in OoD accuracy. To explain this, we identify an "implicit", non-visual kind of spurious correlation here (one between the label and a particular component of the difference between the two channels).

3. In Fig 2d, we add a line to the last channel of CIFAR10 images (regardless of the label), and make the line brighter during testing (resembling a camera glitch), which results in a $\gtrsim 27\%$ drop in a ResNet's accuracy. We geometrically argue how this failure arises from breaking Constraint 3.5.
4. In App C.3.4, we consider an MNIST setting inspired by Tsipras et al. (2019), where failure arises (geometrically) due to high-dimensional spurious features (breaking Constraint 3.4).

We hope that these examples, described in greater detail in App C.3 (with explanations based off of the other failure modes discussed App A), provide (a) a broader way to think about how spurious correlations manifest, and (b) how a variety of resulting failure modes can be reasoned geometrically.

5 Failure due to statistical skews

Having theoretically studied max-margin classifiers, let us now turn our attention to studying linear classifiers trained by gradient descent on logistic/exponential loss. Under some conditions, on linearly separable datasets, these classifiers would converge to the max-margin classifier given infinite time (Soudry et al., 2018; Ji & Telgarsky, 2018). So it is reasonable to say that even these classifiers would suffer from the geometric skews, even if stopped in some finite time. However, are there any other failure modes that would arise here?

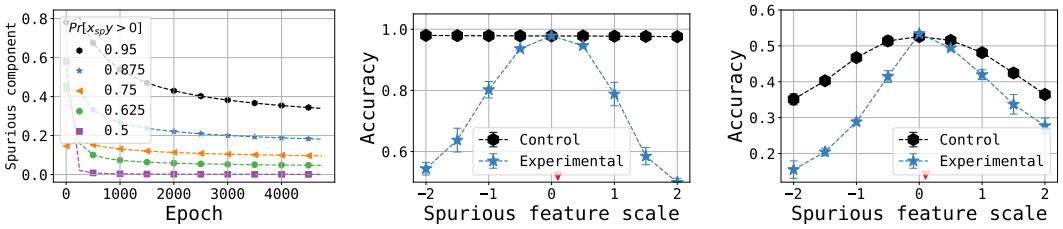
To answer this, let us dial back to the easiest-to-learn task: the setting with four points $\{-1,+1\} \times \{-\mathcal{B},+\mathcal{B}\}$, where in the training distribution (call it $\mathcal{D}_{\text{2-dim}}$), we have $x_{\text{inv}} = y$ and x_{sp} to be $y\mathcal{B}$ with probability $p \in [0.5, 1]$ and $-y\mathcal{B}$ with probability $1-p$. Even though the max-margin classifier here does not rely on x_{sp} for any level of spurious correlation $Pr_{\mathcal{D}_{\text{train}}}[x_{\text{sp}} \cdot y > 0] = p$ (provided $p \neq 1$), the story is more complicated when we empirically evaluate via gradient descent (stopped in finite time). Specifically, for various values of p we plot $w_{\text{sp}}/\sqrt{w_{\text{inv}}^2 + w_{\text{sp}}^2}$ vs. t (here, looking at w_{sp} alone does not make sense since the weight norm grows unbounded). We observe in Fig 3a that the spurious component appears to stagnate around a value proportional to p, even after sufficiently long training, and even though it is supposed to converge to 0.

To explain this behavior, a partial clue already exists in Soudry et al. (2018); Ji & Telgarsky (2018): gradient descent can have a frustratingly slow logarithmic rate of convergence to the max-margin i.e,. the ratio $|w_{\text{sp}}/w_{\text{inv}}|$ could decay to zero as slow as $O(1/\ln t)$. However, this bound is a distribution-independent one that does not explain why the behavior varies with the spurious correlation. Unfortunately, obtaining a distribution-specific bound is technically quite challenging. Therefore we will tackle this problem under a few simplifications. The key simplification is that we will focus solely on this 2D setting and with $\mathcal{B} = 1$ (unlike the general setting studied in the previous failure mode). We then show that the rate of convergence of $w_{\text{sp}}/w_{\text{inv}}$ to 0 is tightly characterized by $1/\ln t$, crucially with an extra factor that monotonically increases in $[0, \infty)$ with respect to the spurious correlation, $p \in [0.5, 1]$ (thus capturing slower convergence for larger spurious correlation). Another notable aspect of the following result is that when there is no spurious correlation ($p = 0.5$), both the upper and lower bound reduce to 0, indicating quick convergence. Below we state this bound specifically for the exponential loss $\exp(-yh(\mathbf{x}))$. The proof is in Appendix B.2, where we also provide similar bounds for the logistic loss in Theorem B.3 (although only for sufficiently large t).

Theorem 5.1. *Under the exponential loss with infinitesimal learning rate, a linear classifier $w_{\text{inv}}(t)x_{\text{inv}} + w_{\text{sp}}(t)x_{\text{sp}}$ initialized to the origin and trained on $\mathcal{D}_{\text{2-dim}}$ with $\mathcal{B} = 1$ satisfies $\forall t$:*

$$\Omega\left(\frac{\ln\left((1+2p)/(3-2p)\right)}{\ln(t)}\right) \leq \frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} \leq O\left(\frac{\ln\left(p/(1-p)\right)}{\ln(t)}\right), \quad \textit{where } p := Pr_{\mathcal{D}_{\text{2-dim}}}[x_{\text{sp}} \cdot y > 0] \in [0.5, 1].$$

While the above result is only for a very specific setting, the proof provides valuable intuition for why similar results may hold in a more general setting too – indeed, we will shortly isolate this failure mode in more general empirical settings. The rough intuition in a general setting is that in the initial epochs, when the loss $\exp(-y\mathbf{w} \cdot \mathbf{x})$ on all points are more or less the same, the updates $1/|S| \cdot \sum_{(\mathbf{x},y)\in S} y\mathbf{x} \exp(-y\mathbf{w} \cdot \mathbf{x})$ roughly push along the direction, $1/|S| \cdot \sum_{(\mathbf{x},y)\in S} y\mathbf{x}$. This is the precise (mis)step where gradient descent "absorbs" the spurious correlation, as this step pushes w_{sp} along $p\mathcal{B} - (1-p)\mathcal{B} = (2p-1)\mathcal{B}$. While this update would be near-zero when there is only little



(a) Failure of gradient-descent (b) FNN + MNIST (c) ResNet + CIFAR10

Figure 3: **Statistical skews:** In **Fig 3a**, we plot the slow convergence of $w_{\text{sp}}/\sqrt{w_{\text{inv}}^2+w_{\text{sp}}^2} \in [0,1]$ under logistic loss with learning rate 0.001 and a training set of 2048 from $\mathcal{D}_{\text{2-dim}}$. In **Fig 3b** and **Fig 3c**, we demonstrate the effect of statistical skews. Here, during test-time we shift both the scale of the spurious feature (from its original scale of 0.1 as marked by the red triangle) and its correlation. Observe that the network trained on the statistically-skewed S_{exp} is more vulnerable to these shifts compared to no-skew S_{con}.[2]

spurious correlation ($p \approx 0.5$), it takes larger values for larger levels of spurious correlation ($p \approx 1$). Unfortunately, under exponential-type losses, the gradients decay with time, and so the future gradients, even if they eventually get rid of this absorbed spurious component, take an exponentially long time to do so.

Broader empirical implications. In practice, it is challenging to validate the effect of statistical skews because any gradient-descent-trained model is likely to be hurt by both statistical skews and geometric skews. We handle this by designing the following experiment. We first create a control dataset S_{con} where there are no geometric or statistical skews. For this, we take a set of images S_{inv} (with no spurious features), and create two copies of it, S_{maj} and S_{min} where we add spurious features, positively and negatively aligned with the label, respectively, and define $S_{\text{con}} = S_{\text{maj}} \cup S_{\text{min}}$. Next, we create an experimental dataset S_{exp} with a statistical skew in it. We do this by taking S_{con} and duplicating S_{maj} in it so that the ratio $|S_{\text{maj}}| : |S_{\text{min}}|$ becomes $10 : 1$. Importantly, this dataset has no geometric skews, since merely replicating points does not affect the geometry/max-margin norms. Then, if we were to observe that training gradient descent on S_{exp} results in greater spurious-feature-reliance than S_{con}, we would have isolated the effect of statistical skews. Indeed, we demonstrate this in Fig 3 for Binary-MNIST using FNNs (where we concatenate a spurious channel) and for (multiclass) CIFAR10 (Fig 1c) using ResNet, validating the effect of statistical skews in practice. More details are provided in App C.4.

6 Conclusions and Future Work

We identify that spurious correlations during training can induce two distinct skews in the training set, one geometric and another statistical. These skews result in two complementary ways by which empirical risk minimization (ERM) via gradient descent is guaranteed to rely on those spurious correlations. At the same time, our theoretical results (in particular, the upper bounds on the spurious component of the classifier) show that when these skews do disappear, there is no failure within the considered tasks. This suggests that within the class of easy-to-learn tasks and for gradient-descent-trained linear models, the above discussion likely captures all possible failure modes.

However, when we do venture into the real-world to face more complicated tasks and use non-linear, deep models, many other kinds of failure modes would crop up (such as the ones we enumerate in Sec 3.1, besides also the above fundamental ones). Indeed, the central message of our work is that there is no one unique mechanism by which classifiers fail under spurious correlations, even in the simplest of tasks. This in turn has a key practical implication: in order to improve our solutions to OoD generalization, it would be valuable to figure out whether or not a unified solution approach is sufficient to tackle all these failure mechanisms. We hope that the foundation we have laid in this study helps future work in tackling these challenges.

[2]The overall accuracy on CIFAR10 is low because even though $|S_{\text{con}}| = 50k$ and $|S_{\text{exp}}| = 455k$, the number of unique samples here is just $5k$. See App C.4.2 for more explanation.

REFERENCES

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. 2019. URL http://arxiv.org/abs/1907.02893.

Nader Asadi, Mehrdad Hosseinzadeh, and Mahdi Eftekhari. Towards shape biased unsupervised representation learning for domain generalization. abs/1909.08245, 2019.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Proceedings, Part XVI*, 2018.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems 24*, 2011.

Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *IEEE International Conference on Computer Vision, ICCV 2013*, 2013.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 2016.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian J. Goodfellow. Adversarial spheres. In *6th International Conference on Learning Representations, ICLR 2018, Workshop Track Proceedings*. OpenReview.net, 2018.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. 2020. URL https://arxiv.org/abs/2007.01434.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. abs/1803.07300, 2018.

David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). abs/2003.00688, 2020.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*. AAAI Press, 2018a.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Computer Vision - ECCV 2018 - 15th European Conference*, 2018b.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, 2013.

Vaishnavh Nagarajan and J. Zico Kolter. Generalization in deep networks: The role of distance from initialization. 2017.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems 32*, 2019.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. 2017.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B*, 2016.

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. 2020a.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. 2020b.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19, 2018.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998. ISBN 978-0-471-03003-4.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

A CONSTRAINTS AND OTHER FAILURE MODES

Here, we elaborate on how each of the constraints we impose on the easy-to-learn tasks corresponds to eliminating a particular complicated kind of failure of ERM from happening. In particular, for each of these constraints, we'll construct a task that betrays that constraint (but obeys all the others) and that causes a unique kind of failure of ERM. We hope that laying these out concretely can provide a useful starting point for future work to investigate these various failure modes. Finally, it is worth noting that all the failure modes here can be explained via a geometric argument, and furthermore the failure modes for Constraint 3.5 and Constraint 3.4 can be explained as a special case of the argument in Theorem 4.1.

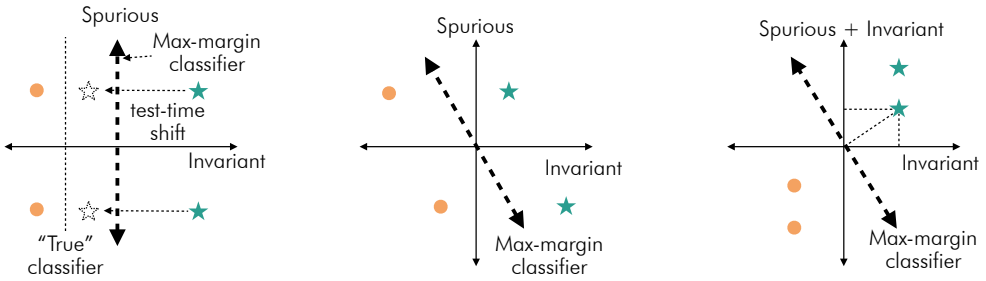
Failure due to weakly predictive invariant feature (breaking Constraint 3.1). We enforced in Constraint 3.1 that the invariant feature be fully informative of the label. For a setting that breaks this constraint, consider a 2D task with *noisy* invariant features, where across all domains, we have a Gaussian invariant feature of the form $x_{\text{inv}} \sim \mathcal{N}(y, \sigma_{\text{inv}}^2)$ – this sort of a noisy invariant feature was critically used in Tsipras et al. (2019); Sagawa et al. (2020b); Arjovsky et al. (2019) to explain failure of ERM. Now, assume that during training we have a spurious feature $x_{\text{sp}} \sim \mathcal{N}(y, \sigma_{\text{sp}}^2)$ (say with relatively larger variance, while positively correlated with y). Then, observe that the Bayes optimal classifier on $\mathcal{D}_{\text{train}}$ is $\text{sgn}\left(x_{\text{inv}}/\sigma_{\text{inv}} + x_{\text{sp}}/\sigma_{\text{sp}}\right)$ i.e., it must rely on the spurious feature.

However, also observe that if one were to eliminate noise in the invariant feature by setting $\sigma_{\text{inv}} \to 0$ (thus making the invariant feature *perfectly informative* of the label like in our MNIST example), the Bayes optimal classifier approaches $\text{sgn}\left(x_{\text{inv}}\right)$, thus succeeding after all.

Failure due to "unstable" invariant feature (breaking Constraint 3.2). If during test-time, we push the invariant features closer to the decision boundary (e.g., partially occlude the shape of every camel and cow), test accuracy will naturally deteriorate, and this embodies a unique failure mode.

Concretely, consider a domain generalization task where across all domains $x_{\text{inv}} \geq -0.5$ determines the true boundary (see Fig 4a). Now, assume that in the training domains, we see $x_{\text{inv}} = 2y$ or $x_{\text{inv}} = 3y$. This would result in learning a max-margin classifier of the form $x_{\text{inv}} \geq 0$. Now, during test-time, if one were to provide "harder" examples that are closer to the true boundary in that $x_{\text{inv}} = -0.5 + 0.1y$, then all the positive examples would end up being misclassified.

Failure due to complex conditional dependencies (breaking Constraint 3.3). This constraint imposed $x_{\text{inv}} \perp\!\!\!\perp x_{\text{sp}} | y$. Stated more intuitively, given that a particular image is that of a camel, this constraint captures the fact that knowing the background color does not tell us too much about the precise shape of the camel. An example where this constraint is broken is one where x_{inv} and



(a) Breaking Constraint 3.2: Unstable invariant feature

(b) Breaking Constraint 3.3: Conditionally dependent features

(c) Breaking Constraint 3.5: Hard-to-disentangle features

Figure 4: **Other failure modes:** We visualize the different (straightforward) ways in which a max-margin classifier can be shown to fail in tasks where one of the Constraints in Sec 3.1 is disobeyed. More discussion in Sec A.

x_{sp} share a neat geometric relationship during training, but not during testing, which then results in failure as illustrated below.

Consider the example in Fig 4b where for the positive class we set $x_{\text{inv}}+x_{\text{sp}} = 1$ and for the negative class we set $x_{\text{inv}}+x_{\text{sp}} = -1$, thereby breaking this constraint. Now even though the invariant feature in itself is fully informative of the label, while the spurious feature is not, the max-margin here is parallel to the line $x_{\text{inv}} + x_{\text{sp}} = c$ and is therefore reliant on the spurious feature.

Failure mode due to high-dimensional spurious features (lack of Constraint 3.4). Akin to the setting in Tsipras et al. (2019), albeit without noise, consider a task where the spurious feature has D different co-ordinates, $\mathcal{X}_{\text{sp}} = \{-1, 1\}^D$ and the invariant feature just one $\mathcal{X}_{\text{inv}} = \{-1, +1\}$. Then, assume that the ith spurious feature $x_{\text{sp},i}$ independently the value y with probability p_i and $-y$ with probability $1 - p_i$, where without loss of generality $p_i > 1/2$. Here, with high probability, all datapoints in S can be separated simply by summing up the spurious features (given D is large enough). Then, we argue that the max-margin classifier would provide some non-zero weight to this direction because it helps maximize its margin.

One way to see why this is true is by invoking a special case of Theorem 4.1. In particular, if we define $\sum x_{\text{sp},i}$ to be a single dimensional spurious feature x_{sp}, this feature satisfies $x_{\text{sp}} \cdot y > 0$ on all training points. In other words, this is an extreme scenario with no minority group. Then, Theorem 4.1 would yield a positive lower bound on the weight given to x_{sp}, explaining why the classifier relies on the spurious pixels. For the sake of completeness, we formalize all this discussion below:

Proposition A.1. *Let c be a constant such that for all i, $p_j > \frac{1}{2} + \frac{c}{2}$. Let D be sufficiently large so that $D \geq \frac{1}{2c}\sqrt{2 \ln \frac{m}{\delta}}$ where m is the number of training datapoints in S. Then, w.h.p. of $1 - \delta$ over the draws of S, the max-margin classifier corresponds is of the form $w_{\text{inv}} x_{\text{inv}} + \mathbf{w}_{\text{sp}} \mathbf{x}_{\text{sp}}$ where:*

$$\frac{\|\mathbf{w}_{\text{sp}}\|}{w_{\text{inv}}} \geq \frac{c\sqrt{D}}{2}.$$

Proof. First, we'll show that on S there exists a classifier that relies purely on the spurious features to separate the data. In particular, consider $\mathbf{w}'_{\text{sp}}$ where the ith dimension is $1/\sqrt{D}$ if $p_i > 1/2$, and $-1/\sqrt{D}$ otherwise. By the Hoeffding's inequality, we have that with high probability $1 - \delta$, on all the m training datapoints, $y\mathbf{w}'_{\text{sp}} \cdot \mathbf{x}_{\text{sp}} \geq \frac{1}{\sqrt{D}} \sum (2p_i - 1) - \sqrt{\frac{2}{D} \ln \frac{m}{\delta}} \geq \frac{c}{2}\sqrt{D}$.

Now, for the max-margin classifier, assume that $w_{\text{inv}}^2 = \alpha$. Further, assume that the margin contributed by $\mathbf{w}_{\text{sp}} x_{\text{sp}} + b$ equals $\sqrt{1 - \alpha^2} m$ for some m. Observe that m must satisfy $m \geq \frac{c}{2}\sqrt{D}$ (as otherwise, we can replace $\mathbf{w}_{\text{sp}}$ with $\sqrt{1 - \alpha^2}\mathbf{w}'_{\text{sp}}$ to achieve a better margin). Now, for the resulting margin to be maximized, α must satisfy $\frac{\alpha}{\sqrt{1-\alpha^2}} = \frac{1}{m}$.

□

Failure mode due to hard-to-separate features (lack of Constraint 3.5) Here we assumed that the feature space can be orthogonally decomposed into invariant and spurious features. Now let us imagine a 2D task, visualized in Fig 4c where this is not respected in that each datapoint is written as $(x_{\text{inv}}, x_{\text{inv}} + x_{\text{sp}})$, assuming that $x_{\text{inv}} = y$ and $x_{\text{sp}} \in \{-0.5, 0.5\}$. A practical example of this sort of structure is the example in Fig 2d where we add a line to the last channel of CIFAR10, and then vary the brightness of the line during test-time.

To understand why failure occurs in this 2D example, observe that, regardless of the correlation between x_{sp} and y, we'd have that $(x_{\text{inv}} + x_{\text{sp}}) \cdot y > 0$. In other words, the second co-ordinate is fully informative of the label. The max-margin classifier, due to its bias, relies on both the first and the second co-ordinate to maximize its margin i.e., the classifier would be of the form $w_1 x_{\text{inv}} + w_2(x_{\text{inv}} + x_{\text{sp}})$ where $w_2 > 0$. (Again, like in our discussion of the failure mode of Constraint 3.4, we can argue this via Thm 4.1 by considering $x_{\text{inv}} + x_{\text{sp}}$ itself as a spurious feature, and by observing that there's no minority group here.) Hence, by assigning a positive weight to the second co-ordinate it inadvertently becomes susceptible to the spurious feature that may shift during training.

B PROOFS

B.1 PROOF OF THEOREM 4.1 ON FAILURE DUE GEOMETRIC SKEWS

Below we provide a proof for our result analyzing the failure mode arising from geometric skews in the data.

Recall that given a dataset S, where the spurious feature can take only values in $\{-\mathcal{B}, +\mathcal{B}\}$, we partitioned S into two subsets S_{maj} and S_{min} where in S_{maj} the points satisfy $x_{\text{sp}} \cdot y > 0$ and in S_{min} the points satisfy $x_{\text{sp}} \cdot y < 0$.

Next, we define two key notations. First, for any dataset $T \subseteq S$, let $\mathbf{v}(T) \in \mathcal{X}_{\text{inv}}$ denote a least-norm vector (purely in the invariant space) that achieves a margin of at least 1 on all datapoints in T (see Appendix B.1 for clearer definition). Similarly, let $\tilde{\mathbf{v}}(T) \in \mathcal{X}_{\text{inv}}$ denote a least-norm vector that achieves a margin of at least 1 on T, and a margin of at least 0 on $S \setminus T$ (see full definition below). While by definition, $\|\mathbf{v}(T)\| \leq \|\tilde{\mathbf{v}}(T)\|$, we can informally treat these quantities as the same since empirically $\|\mathbf{v}(T)\| \approx \|\tilde{\mathbf{v}}(T)\|$. We show these plots in Sec C.1 for both MNIST and CIFAR10. But importantly, both these quantities grow with $|T|$ as noted earlier. Then, by virtue of the small size of the minority group S_{min}, we can say, for example that $\|\mathbf{v}(S_{\text{min}})\|$ is smaller than both $\|\mathbf{v}(S_{\text{maj}})\|$ and $\|\mathbf{v}(S)\|$. We refer to this gap as a geometric skew. When this skew is prominent enough (e.g., $\|\mathbf{v}(S_{\text{min}})\|/\|\mathbf{v}(S)\| \approx 0$), our result below argues that the spurious component in the overall max-margin classifier must be sufficiently large (and positive). On the flip side, we also show that when the skew is negligible enough (e.g., $\|\mathbf{v}(S_{\text{min}})\|/\|\mathbf{v}(S)\| \approx 1$), then the spurious component has to be sufficiently small.

To be able to better visualize these bounds, we write these as bounds on $|\mathcal{B} w_{\text{sp}}|$ (i.e., $|w_{\text{sp}} x_{\text{sp}}|$ rather than $|w_{\text{sp}}|$). Then we can think of a lower bound of the form $|\mathcal{B} w_{\text{sp}}| \gtrsim 1$ as demonstrating serious failure as a shift in the correlation can adversely reduce the original margin of ≈ 1.

Before we state the result, for clarity, we state the full mathematical definition of $\mathbf{v}$ and $\tilde{\mathbf{v}}$ as follows. For any $T \subseteq S$:

$$
\begin{aligned}
\mathbf{v}(T), b(T) = \underset{\mathbf{w}_{\text{inv}} \in \mathcal{X}_{\text{inv}}, b}{\arg\min} \quad & \|\mathbf{w}_{\text{inv}}\|^2 \\
\text{s.t.} \quad & y(\mathbf{w}_{\text{inv}} \cdot \mathbf{x}_{\text{inv}}) + b) \geq 1 \qquad \forall\, ((\mathbf{x}_{\text{inv}}, x_{\text{sp}}), y) \in T
\end{aligned}
$$

$$
\begin{aligned}
\tilde{\mathbf{v}}(T), \tilde{b}(T) = \underset{\mathbf{w}_{\text{inv}} \in \mathcal{X}_{\text{inv}}, b}{\arg\min} \quad & \|\mathbf{w}_{\text{inv}}\|^2 \\
\text{s.t.} \quad & y(\mathbf{w}_{\text{inv}} \cdot \mathbf{x}_{\text{inv}}) + b) \geq 1 \qquad \forall\, ((\mathbf{x}_{\text{inv}}, x_{\text{sp}}), y) \in T \\
& y(\mathbf{w}_{\text{inv}} \cdot \mathbf{x}_{\text{inv}}) + b) \geq 0 \qquad \forall\, ((\mathbf{x}_{\text{inv}}, x_{\text{sp}}), y) \in S \setminus T
\end{aligned}
$$

Using these notations, we state our full theorem and provide its proof below.

Theorem B.1. *Let $\mathbb{H}$ be the set of linear classifiers, $h(x) = \mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + w_{\text{sp}}x_{\text{sp}} + b$. Let the geometric skews in a dataset S be quantified through the terms $\kappa_1 = \|\mathbf{v}(S_{\text{min}})\|/\|\mathbf{v}(S)\|$, $\kappa_2 = \|\mathbf{v}(S_{\text{min}})\|/\|\mathbf{v}(S_{\text{maj}})\|$ and $\tilde{\kappa}_1 := \|\tilde{\mathbf{v}}(S_{\text{min}})\|/\|\tilde{\mathbf{v}}(S)\|$, $\tilde{\kappa}_2 := \|\tilde{\mathbf{v}}(S_{\text{min}})\|/\|\tilde{\mathbf{v}}(S_{\text{maj}})\|$. Then for any task satisfying all the constraints in Sec 3.1 the max-margin classifier satisfies the inequalities (where for readability, we will use $c_1 := 1/(2\|\tilde{\mathbf{v}}(S)\|\mathcal{B})$, $c_2 := 1/(2\|\tilde{\mathbf{v}}(S_{\text{maj}})\|\mathcal{B})$):*

$$\mathcal{B}w_{\text{sp}} \geq \max\left(1 - 2\sqrt{\tilde{\kappa}_1 + c_1^2}, 0\right) \qquad \textit{if} \quad \tilde{\kappa}_2 \leq \sqrt{1/4 - c_2^2}, \quad \textit{and}$$

$$|\mathcal{B}w_{\text{sp}}| \leq 1/\kappa_1 - 1 \qquad \textit{if} \quad \kappa_2 \leq 1.$$

In the above result one can think of c_1 and c_2 as small constants as that would be the case when either $\mathcal{B}$ is large or when $\mathbf{v}(S)$ is large. Furthermore, for readability, one can also imagine that all the κ terms are similar to each other.

Proof. We present the proof of lower bound first, followed by the upper bound.

Proof of lower bound. First, we will show that there exists a classifier of norm 1 that relies on the spurious feature to create a sufficiently large margin. We'll let this classifier be of the form $\alpha\frac{\tilde{\mathbf{v}}(S_{\text{min}})}{\|\tilde{\mathbf{v}}(S_{\text{min}})\|} \cdot \mathbf{x}_{\text{inv}} + \sqrt{1-\alpha^2}x_{\text{sp}} + \alpha b_{\text{min}}$. By the definition of $\tilde{\mathbf{v}}(S_{\text{min}})$, the margin of this classifier on any datapoint in S_{min} is at least $\frac{\alpha}{\|\tilde{\mathbf{v}}(S_{\text{min}})\|} - \sqrt{1-\alpha^2}\mathcal{B}$. Again, by the definition of $\tilde{\mathbf{v}}(S_{\text{min}})$, the margin on S_{maj} is at least $\sqrt{1-\alpha^2}\mathcal{B}$. Let us pick an α such that these two quantities are equal. Such an α would satisfy $\frac{\alpha}{\sqrt{1-\alpha^2}} = 2\|\tilde{\mathbf{v}}(S_{\text{min}})\|\mathcal{B}$. By plugging this back, we get that the resulting margin of this classifier on the whole dataset S is at least $\frac{\mathcal{B}}{\sqrt{1+4\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2\mathcal{B}^2}}$. In other words, this also means the least norm classifier with a margin of at least 1 on S has a norm of at most $\frac{\sqrt{1+4\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2\mathcal{B}^2}}{\mathcal{B}}$.

Now, assume that the least norm classifier with a margin of 1 on S is of the form $\mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + w_{\text{sp}}x_{\text{sp}} + b$. First, we derive a lower bound on $|w_{\text{sp}}|$, and then show that $w_{\text{sp}} > 0$. For this we'll consider two cases, one where $|w_{\text{sp}}| \geq \frac{1}{\mathcal{B}}$ (and so we already have a lower bound) and another case where $|w_{\text{sp}}| < \frac{1}{\mathcal{B}}$. In the latter case, we will need the invariant part of the max-margin classifier, namely $\mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + b$, to have to have a margin of at least $1 - |w_{\text{sp}}|\mathcal{B}$ on S; if it were any lesser, the contribution from spurious component which is at most $|w_{\text{sp}}x_{\text{sp}}|$ will be unable to bump this margin up to 1. Now, for the invariant part of the max-margin classifier to have a margin of at least $1 - |w_{\text{sp}}|\mathcal{B}$ (a non-negative quantity since $|w_{\text{sp}}| \leq 1/\mathcal{B}$) on S, the norm of the classifier must be at least $(1 - |w_{\text{sp}}|\mathcal{B})\|\tilde{\mathbf{v}}(S)\|$ (which follows from the definition of $\tilde{\mathbf{v}}(S)$). Therefore, for the overall classifier to be the max-margin classifier, we need $(1 - |w_{\text{sp}}|\mathcal{B})\|\mathbf{v}(S)\| \leq \frac{\sqrt{1+4\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2\mathcal{B}^2}}{\mathcal{B}}$. Rearranging this gives us the bound that $\mathcal{B}|w_{\text{sp}}| \geq 1 - 2\sqrt{\frac{\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2}{\|\tilde{\mathbf{v}}(S)\|^2} + \frac{1}{4\mathcal{B}^2\|\tilde{\mathbf{v}}(S)\|^2}}$. (Note that throughout the discussion, the term $\tilde{\mathbf{v}}(S)$ is identical to $\mathbf{v}(S)$, but we chose to use the former to make our final result more readable.)

What remains now is to show that $w_{\text{sp}} > 0$. For this we do the same argument as above but with a slight modification. First, if $w_{\text{sp}} > 1/\mathcal{B}$, we are done. So, assume that $w_{\text{sp}} \leq 1/\mathcal{B}$. Then, we can say that the invariant part of the max-margin classifier i.e., $\mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + b$, must achieve a margin of $1 - w_{\text{sp}}\mathcal{B}$ (a non-negative quantity since $w_{\text{sp}} \leq 1/\mathcal{B}$) specifically on S_{maj}. Then, by the definition of $\mathbf{v}(S_{\text{maj}})$, it follows that the norm of our overall max-margin classifier must be at least $(1 - w_{\text{sp}}\mathcal{B})\|\mathbf{v}(S_{\text{maj}})\|$. Again, for the overall classifier to be the max-margin classifier, we need $(1 - |w_{\text{sp}}|\mathcal{B})\|\mathbf{v}(S_{\text{maj}})\| \leq \frac{\sqrt{1+4\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2\mathcal{B}^2}}{\mathcal{B}}$, which when rearranged gives us $\mathcal{B}w_{\text{sp}} \geq 1 - \frac{\sqrt{4\mathcal{B}^2\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2+1}}{\mathcal{B}\|\tilde{\mathbf{v}}(S_{\text{maj}})\|}$. The R.H.S is at least 0 when $\frac{1}{4}\left(1 - \frac{1}{\|\tilde{\mathbf{v}}(S_{\text{maj}})\|^2\mathcal{B}^2}\right) \geq \frac{\|\tilde{\mathbf{v}}(S_{\text{min}})\|^2}{\|\tilde{\mathbf{v}}(S_{\text{maj}})\|^2}$ (i.e., $\sqrt{\frac{1}{4} - c_2^2} \geq \tilde{\kappa}_2$). In other words when $\tilde{\kappa}_2 \leq \sqrt{\frac{1}{4} - c_2^2}$, we have $w_{\text{sp}} \geq 0$.

Proof of upper bound. The spurious component of the classifier $\mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + w_{\text{sp}}x_{\text{sp}} + b$ positively contributes to the margin of one of the groups, and negatively contributes to the other group. On whichever group the spurious component negatively contributes to the margin, the invariant part

of the classifier, $\mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + b$, must counter this and achieve a margin of $1 + |w_{\text{sp}}|\mathcal{B}$. To manage this, we'd require that $\|\mathbf{w}_{\text{inv}}\| \geq (1 + |w_{\text{sp}}|\mathcal{B}) \min(\|\mathbf{v}(S_{\text{min}})\|, \|\mathbf{v}(S_{\text{maj}})\|)$. However, since the overall classifier $\mathbf{w}_{\text{inv}}\mathbf{x}_{\text{inv}} + w_{\text{sp}}x_{\text{sp}} + b$ is a min-norm classifier, if $\|\mathbf{w}_{\text{inv}}\|$ was larger than $\|\mathbf{v}(S)\|$ we'd run into a contradiction (as that would result in $\mathbf{v}(S)$ being the min-norm classifier). Therefore, $(1 + |w_{\text{sp}}|\mathcal{B}) \min(\|\mathbf{v}(S_{\text{min}})\|, \|\mathbf{v}(S_{\text{maj}})\|) \leq \|\mathbf{v}(S)\|$. Since we are given $\kappa_2 \leq 1$ that $\min(\|\mathbf{v}(S_{\text{min}})\|, \|\mathbf{v}(S_{\text{maj}})\|) = \|\mathbf{v}(S_{\text{min}})\|$, this simplifies to $(1 + |w_{\text{sp}}|\mathcal{B})\|\mathbf{v}(S_{\text{min}})\| \leq \|\mathbf{v}(S)\|$, which when rearranged reaches the desired result. □

B.2 Proof of Theorem 5.1 on failure due to statistical skews

Here we state the full form of Theorem 5.1 and its proof.

Theorem B.2. *Under the exponential loss with infinitesimal learning rate, a linear classifier $w_{\text{inv}}(t)x_{\text{inv}} + w_{\text{sp}}(t)x_{\text{sp}}$ initialized to the origin and trained on $\mathcal{D}_{\text{2-dim}}$ with $\mathcal{B} = 1$ satisfies:*

$$\frac{\ln\left({}^{(1+2p)}/_{(3-2p)}\right)}{\ln(1 + 3\max(t, 1))} \leq \frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} \leq \frac{\ln\left({}^{p}/_{(1-p)}\right)}{\ln(1 + 2t)}, \quad \textit{where } p := Pr_{\mathcal{D}_{\text{2-dim}}}[x_{\text{sp}} \cdot y > 0] \in [0.5, 1].$$

Proof. Throughout the proof, we'll drop the argument t from $w_{\text{inv}}(t)$ and $w_{\text{sp}}(t)$ for convenience.

The loss function that is being minimized in this setting corresponds to:

$$L(w_{\text{inv}}, w_{\text{sp}}) = pe^{-(w_{\text{inv}}+w_{\text{sp}})} + (1-p)e^{-(w_{\text{inv}}-w_{\text{sp}})},$$

where $p \geq 0.5$. Here, the first term is the loss on the majority dataset (where $x_{\text{sp}} = y\mathcal{B}$) and the second term is the loss on the minority dataset (where $x_{\text{sp}} = -y\mathcal{B}$).

Now the updates on w_{inv} and w_{sp} are given by:

$$\begin{aligned}
\dot{w}_{\text{inv}} &= pe^{-(w_{\text{inv}}+w_{\text{sp}})} + (1-p)e^{-(w_{\text{inv}}-w_{\text{sp}})} \\
\dot{w}_{\text{sp}} &= pe^{-(w_{\text{inv}}+w_{\text{sp}})} - (1-p)e^{-(w_{\text{inv}}-w_{\text{sp}})},
\end{aligned}$$

which means:

$$\begin{aligned}
\frac{d(w_{\text{inv}} + w_{\text{sp}})}{dt} &= 2pe^{-(w_{\text{inv}}+w_{\text{sp}})} \\
\frac{d(w_{\text{inv}} - w_{\text{sp}})}{dt} &= 2(1-p)e^{-(w_{\text{inv}}+w_{\text{sp}})}
\end{aligned}$$

Thus, by rearranging and integrating we get:

$$\begin{aligned}
w_{\text{inv}} + w_{\text{sp}} &= \ln(1 + 2pt) \\
w_{\text{inv}} - w_{\text{sp}} &= \ln(1 + 2(1-p)t) \\
w_{\text{inv}} &= 0.5(\ln(1 + 2pt) + \ln(1 + 2(1-p)t)) \\
w_{\text{sp}} &= 0.5(\ln(1 + 2pt) - \ln(1 + 2(1-p)t)).
\end{aligned}$$

Now let us define $\beta(t) = w_{\text{sp}}/w_{\text{inv}}$:

$$\beta(t) := \frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} = \frac{\ln(1 + 2pt) - \ln(1 + 2(1-p)t)}{\ln(1 + 2pt) + \ln(1 + 2(1-p)t)}. \tag{1}$$

To bound this quantity, we'll consider two cases, $t \geq 1$ and $t < 1$. First let us consider $t \geq 1$. We begin by noting that the numerator $w_{\text{sp}}(t)$ is increasing with time t. This is because,

$$
\begin{aligned}
w_{\text{sp}}(t) &= \ln \frac{1+2pt}{1+2(1-p)t} \\
&= \ln \left(1 + \frac{2(2p-1)t}{1+2(1-p)t}\right) \\
&= \ln \left(1 + \frac{2(2p-1)}{\frac{1}{t}+2(1-p)}\right).
\end{aligned}
$$

Here, the term $\frac{2(2p-1)}{\frac{1}{t}+2(1-p)}$ is increasing with t due to the fact that the numerator is non-negative ($p \geq 0.5$) and the denominator is decreasing with t. So, given that $w_{\text{sp}}(t)$ is increasing, we can say that for all $t \geq 1$:

$$
\beta(t) \geq \frac{w_{\text{sp}}(1)}{w_{\text{inv}}(t)} = \frac{\ln \frac{1+2p}{3-2p}}{\ln(1+2(1-p)t) + \ln(1+2pt)} \geq \frac{\ln \frac{1+2p}{3-2p}}{\ln(1+3t)}.
$$

Here we have used the fact that the denominator can be upper bounded as $\ln(1+2(1-p)t)+\ln(1+2pt) \leq \ln(1+2t+4(1-p)pt) \leq \ln(1+3t)$.

Now, for any $t \leq 1$, we can show that $\beta(t) \geq \beta(1) = \frac{\ln \frac{1+2p}{3-2p}}{\ln(3+4(p-p^2))} \geq \frac{\ln \frac{1+2p}{3-2p}}{\ln 4}$. This follows if we can show that $\beta(t)$ is decreasing for $t \geq 0$. Taking its derivative with respect to time, we get:

$$
\begin{aligned}
\dot{\beta} =& \frac{(\ln(1+2pt) + \ln(1+2(1-p)t)) \left(\frac{2p}{1+2pt} - \frac{2(1-p)}{(1+2(1-p)t)}\right)}{(\ln(1+2pt) + \ln(1+2(1-p)t))^2} \\
&- \frac{(\ln(1+2pt) - \ln(1+2(1-p)t)) \left(\frac{2p}{(1+2pt)} + \frac{2(1-p)}{1+2(1-p)t}\right)}{(\ln(1+2pt) + \ln(1+2(1-p)t))^2} \\
=& 2 \cdot \frac{\ln(1+2(1-p)t)\frac{2p}{1+2pt} - \ln(1+2pt)\frac{2(1-p)}{1+2(1-p)t}}{(\ln(1+2pt) + \ln(1+2(1-p)t))^2} \\
=& 2 \cdot \frac{\ln(1+2(1-p)t)\frac{1}{\frac{1}{2p}+t} - \ln(1+2pt)\frac{1}{\frac{1}{2(1-p)}+t}}{(\ln(1+2pt) + \ln(1+2(1-p)t))^2}
\end{aligned}
$$

The sign of the above quantity is equal to the sign of:

$$
\begin{aligned}
&\ln(1+2(1-p)t)\left(\frac{1}{2(1-p)}+t\right) - \ln(1+2pt)\left(\frac{1}{2p}+t\right) \\
&= \underbrace{\ln\left(\frac{1}{2(1-p)}+t\right)\left(\frac{1}{2(1-p)}+t\right) + \ln(\frac{1}{2(1-p)})\left(\frac{1}{2(1-p)}+t\right)}_{:=f\left(\frac{1}{2(1-p)}\right)} \\
&- \underbrace{\ln\left(\frac{1}{2p}+t\right)\left(\frac{1}{2p}+t\right) - \ln\frac{1}{2p}\left(\frac{1}{2p}+t\right)}_{:=f\left(\frac{1}{2p}\right)}
\end{aligned}
$$

Now, we show that $f(x) = (x+t)\ln(x+t) - (x+t)\ln x = (x+t)\ln\left(1+\frac{t}{x}\right)$ is a non-increasing function:

$$
\begin{aligned}
f'(x) &= \ln\left(1+\frac{t}{x}\right) + \frac{x+t}{1+\frac{t}{x}} \cdot \frac{-t}{x^2} \\
&= \ln\left(1+\frac{t}{x}\right) - \frac{t}{x} \\
&\leq \frac{t}{x} - \frac{t}{x} \leq 0.
\end{aligned}
$$

Now since $p \geq 0.5$, and f is non-increasing, $f\left(\frac{1}{2(1-p)}\right) - f\left(\frac{1}{2p}\right) \leq 0$. Subsequently, $\dot{\beta} \leq 0$. Therefore, $\beta(t) \geq \beta(1)$ for any $t \in [0, 1]$.

Upper bound. For an upper bound on $\beta(t)$, we note that since $w_{\text{sp}}(t)$ is always increasing $w_{\text{sp}}(t) \leq \lim_{t\to\infty} w_{\text{sp}}(t) = \ln\left(\frac{p}{1-p}\right)$. On the other hand $w_{\text{inv}}(t) = \ln(1 + 2t + 4p(1-p)t^2) \geq \ln(1 + 2t)$. Combining these inequalities, we get:

$$
\beta(t) \leq \frac{\ln\left(\frac{1}{p} - 1\right)}{\ln(1+2t)}.
$$

□

B.2.1 THEOREM 5.1 UNDER LOGISTIC LOSS

While Theorem 5.1 was concerned with bounding the spurious component when trained under the exponential loss, here we state a similar result for logistic loss $\ln(1 + e^{-yh(\vec{x})})$. Here again both the upper and lower bounds increase as the spurious correlations varies from none ($Pr_{\mathcal{D}_{\text{2-dim}}}[x_{\text{sp}} \cdot y > 0] = 0.5$) to its maximum ($Pr_{\mathcal{D}_{\text{2-dim}}}[x_{\text{sp}} \cdot y > 0] = 1.0$). Also, here too, when we set $p = 0.5$, the bounds resolve to zero indicating quick convergence. However we must note that our bounds hold only for sufficiently large t.

Theorem B.3. *Under the logistic loss with infinitesimal learning rate, a linear classifier $w_{\text{inv}}(t)x_{\text{inv}} + w_{\text{sp}}(t)x_{\text{sp}}$ initialized to the origin and trained on $\mathcal{D}_{\text{2-dim}}$ with $\mathcal{B} = 1$ satisfies for a sufficiently large t (where $p := Pr_{\mathcal{D}_{\text{2-dim}}}[x_{\text{sp}} \cdot y > 0] \in [0.5, 1]$):*

$$
\min\left(1, \frac{\frac{1}{2}\ln\left(\frac{2}{3-2p}\right)}{\ln(t+1)}\right) \leq \frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} \leq \frac{\frac{1}{2}\ln\frac{1-p}{p}}{\ln(0.5t+1)}.
$$

Proof. Here, the loss function is of the form:

$$
L(w_{\text{inv}}, w_{\text{sp}}) = p\log(1 + e^{-(w_{\text{inv}}+w_{\text{sp}})}) + (1-p)\log(1 + e^{-(w_{\text{inv}}-w_{\text{sp}})})
$$

where $p \geq 0.5$. Now the updates on w_{inv} and w_{sp} are:

$$
\begin{aligned}
\dot{w}_{\text{inv}} &= p\frac{e^{-(w_{\text{inv}}+w_{\text{sp}})}}{1+e^{-(w_{\text{inv}}+w_{\text{sp}})}} + (1-p)\frac{e^{-(w_{\text{inv}}-w_{\text{sp}})}}{1+e^{-(w_{\text{inv}}-w_{\text{sp}})}} \\
\dot{w}_{\text{sp}} &= p\frac{e^{-(w_{\text{inv}}+w_{\text{sp}})}}{1+e^{-(w_{\text{inv}}+w_{\text{sp}})}} - (1-p)\frac{e^{-(w_{\text{inv}}-w_{\text{sp}})}}{1+e^{-(w_{\text{inv}}-w_{\text{sp}})}},
\end{aligned}
$$

which means:

$$
\frac{d(w_{\text{inv}} + w_{\text{sp}})}{dt} = 2p\frac{e^{-(w_{\text{inv}}+w_{\text{sp}})}}{1+e^{-(w_{\text{inv}}+w_{\text{sp}})}} = 2p\frac{1}{1+e^{(w_{\text{inv}}+w_{\text{sp}})}}
$$

$$\frac{d(w_{\text{inv}} - w_{\text{sp}})}{dt} = 2(1-p)\frac{e^{-(w_{\text{inv}}-w_{\text{sp}})}}{1+e^{-(w_{\text{inv}}-w_{\text{sp}})}} = 2(1-p)\frac{1}{1+e^{(w_{\text{inv}}-w_{\text{sp}})}}.$$

Solving for this, we get:

$$w_{\text{inv}} + w_{\text{sp}} + e^{w_{\text{inv}}+w_{\text{sp}}} = 2pt + 1 \tag{2}$$
$$w_{\text{inv}} - w_{\text{sp}} + e^{w_{\text{inv}}-w_{\text{sp}}} = 2(1-p)t + 1. \tag{3}$$

We first derive some useful inequalities.

First, we argue that forall t,

$$w_{\text{sp}}(t) \geq 0. \tag{4}$$

This is because at the point where $w_{\text{sp}}(t) = 0$, $\dot{w}_{\text{sp}}(t) \geq \frac{2p-1}{1+e^{w_{\text{inv}}}} \geq 0$ (since $p \geq 0.5$). Hence, the system can never reach values of $w_{\text{sp}} < 0$.

Next, we have for all t,

$$w_{\text{inv}}(t) \in [0, \ln(t+1)]. \tag{5}$$

We can show this by summing up Eq 2 and 3

$$\begin{aligned} & 2w_{\text{inv}} + e^{w_{\text{inv}}+w_{\text{sp}}} + e^{w_{\text{inv}}-w_{\text{sp}}} = 2t+2 \\ \implies \quad & 2w_{\text{inv}} + 2\sqrt{e^{w_{\text{inv}}+w_{\text{sp}}} \cdot e^{w_{\text{inv}}-w_{\text{sp}}}} \leq 2t+2 \\ \implies \quad & 2w_{\text{inv}} + 2e^{w_{\text{inv}}} \leq 2t+2 \end{aligned}$$

and since $\dot{w}_{\text{inv}} \geq 0$, and $w_{\text{inv}}(t) = 0$, $w_{\text{inv}}(t) \geq 0$,

$$2e^{w_{\text{inv}}} \leq 2t+2$$

Next, we show:

$$w_{\text{sp}}(t) \leq \frac{1}{2}\ln\frac{2(1-p)t+1}{2pt+1} \leq \frac{1}{2}\ln\frac{(1-p)}{p}. \tag{6}$$

To show this, we divide Eq 2 by Eq 3, to get:

$$\begin{aligned} & \frac{w_{\text{inv}} + w_{\text{sp}} + e^{w_{\text{inv}}+w_{\text{sp}}}}{w_{\text{inv}} - w_{\text{sp}} + e^{w_{\text{inv}}-w_{\text{sp}}}} = \frac{2(1-p)t+1}{2pt+1} \\ \implies \quad & (2(2p-1)t)w_{\text{inv}} + (2(2p-1)t)w_{\text{sp}} + (2pt+1)e^{w_{\text{inv}}+w_{\text{sp}}} = e^{w_{\text{inv}}-(2(1-p)t+1)w_{\text{sp}}} \end{aligned}$$

since by $p \geq 0.5$, Eq 4 and Eq 5 the first two terms are positive,

$$\begin{aligned} \implies \quad & (2pt+1)e^{w_{\text{inv}}+w_{\text{sp}}} \leq (2(1-p)t+1)e^{w_{\text{inv}}-w_{\text{sp}}} \\ \implies \quad & e^{2w_{\text{sp}}} \leq \frac{2(1-p)t+1}{2pt+1}. \end{aligned}$$

This proves the first inequality. The second inequality follows from the fact that $\frac{2(1-p)t+1}{2pt+1}$ is increasing with t so applying $\lim t \to \infty$ gives us an upper bound.

Finally, we rewrite Equation 2 and Equation 3 to get:

$$w_{\text{inv}} + w_{\text{sp}} = \ln(2pt + 1 - (w_{\text{inv}} + w_{\text{sp}})) \tag{7}$$

$$w_{\text{inv}} - w_{\text{sp}} = \ln(2(1-p)t + 1 - (w_{\text{inv}} - w_{\text{sp}}). \tag{8}$$

Adding and subtracting these, we get a different form for the dynamics of these quantities:

$$w_{\text{inv}} = 0.5(\ln(2pt + 1 - (w_{\text{inv}} + w_{\text{sp}})) + \ln(2(1-p)t + 1 - (w_{\text{inv}} - w_{\text{sp}})) \tag{9}$$

$$w_{\text{sp}} = 0.5(\ln(2pt + 1 - (w_{\text{inv}} + w_{\text{sp}})) - \ln(2(1-p)t + 1 - (w_{\text{inv}} - w_{\text{sp}})). \tag{10}$$

Lower bound. To prove a lower bound on $w_{\text{sp}}(t)/w_{\text{inv}}(t)$, we'll first lower bound w_{sp}. Observe that:

$$\begin{aligned} w_{\text{sp}}(t) &= \frac{1}{2}\ln\frac{2pt + 1 - (w_{\text{inv}} + w_{\text{sp}})}{2(1-p)t + 1 - (w_{\text{inv}} - w_{\text{sp}})} \\ &= \frac{1}{2}\ln\left(1 + \frac{2(2p-1)t - 2w_{\text{sp}}}{2(1-p)t + 1 - (w_{\text{inv}} - w_{\text{sp}})}\right) \end{aligned}$$

Now, since w_{sp} is upper bounded by a constant (Eq 6), for sufficiently large t, the numerator of the second term inside the $\ln$ will be positive, and can be lower bounded by $(2p-1)t$. Then, let us consider two scenarios. Either that $w_{\text{sp}} > w_{\text{inv}}$, in which case we already have a lower bound on $\beta(t)$, or that $w_{\text{sp}} \leq w_{\text{inv}}$. In the latter case, we can lower bound the above as:

$$w_{\text{sp}}(t) \geq \frac{1}{2}\ln\left(1 + \frac{(2p-1)t}{2(1-p)t + 1}\right)$$

Since the right hand side is an increasing in t, we can say that for sufficiently large $t \geq 1$,

$$\begin{aligned} w_{\text{sp}}(t) &\geq \frac{1}{2}\ln\left(1 + \frac{(2p-1)}{2(1-p) + 1}\right) \\ &\geq \frac{1}{2}\ln\left(\frac{2}{3-2p}\right). \end{aligned}$$

Combining this with Eq 5 we get for sufficiently large t, either

$$\frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} \geq \frac{\frac{1}{2}\ln\left(\frac{2}{3-2p}\right)}{\ln(t+1)}.$$

or $\frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} \geq 1$.

Upper bound. To upper bound $w_{\text{sp}}(t)/w_{\text{inv}}(t)$, we'll lower bound $w_{\text{inv}}(t)$:

$$w_{\text{inv}} = 0.5(\ln(2pt + 1 - (w_{\text{inv}} + w_{\text{sp}})) + \ln(2(1-p)t + 1 - (w_{\text{inv}} - w_{\text{sp}}))$$

Since by Eq 5 $w_{\text{inv}}(t) \in [0, \ln(t+1)]$ and by Eq 6, $w_{\text{sp}}(t) \in [0, \frac{1}{2}\ln\frac{1-p}{p}]$, for a sufficiently large t, the linear terms within the $\ln$ terms dominate and so, for large t

$$w_{\text{inv}} \geq (\ln(0.5 \cdot 2pt + 1) = \ln(0.5t + 1)$$

Combining this with Eq 6, we get, for large t:

$$\frac{w_{\text{sp}}(t)}{w_{\text{inv}}(t)} \leq \frac{\frac{1}{2}\ln\frac{1-p}{p}}{\ln(0.5t+1)}.$$

□

C MORE ON EXPERIMENTS

Common details: In all our MNIST-based experiments, we consider the Binary-MNIST classification task (Arjovsky et al., 2019) where the first five digits (0 to 4) need to be separated from the rest (5 to 9). Unless specified otherwise, for this we train a fully-connected three-layered ReLU network with a width of 400 and using SGD with learning rate 0.1 for 50 epochs. In all our CIFAR10-based experiments, unless stated otherwise, we consider the 10-class classification problem, and train a ResNetV1 with a depth of 20 for 200 epochs. [3] All values are averaged at least over fives runs. Finally, when we describe our datasets, we'll adopt the convention that all pixels lie between 0 and 1,

C.1 RANDOM FEATURE EXPERIMENTS FROM SECTION 3.1

For the random features based experiment on Binary MNIST in Section 3.1, we consider $50k$ random ReLU features i.e., $\mathbf{x}_{\text{inv}} = \text{ReLU}(W\mathbf{x}_{\text{raw}})$ where W is a $50k \times 784$ matrix (and so this is well overparameterized for dataset sizes upto 6400). Each entry here is drawn from the normal distribution. We set the spurious feature support to be $\{-100, 100\}$, which is about $1/10$th the magnitude of $\|\mathbf{x}_{\text{inv}}\|$. We also conduct similar experiments on a two-class CIFAR10 and report similar OoD accuracy drops in Fig 5a. Here, we use the first two classes of CIFAR10, as against grouping five classes together into each of the classes. This is because on the latter dataset, the random features representation has poor in-distribution accuracy to begin with.

C.2 INCREASING NORM EXPERIMENTS FROM SECTION 4

The main premise behind our geometric skews argument is that as we increase the number of datapoints, it requires greater norm for the model to fit those points. We verify this for random features models on Binary-MNIST and two-class CIFAR10 in Figs 5b, 5c.

While the theory is solely focused on linear classifiers, we can verify this premise intuitively for neural network classifiers. However, for neural networks, the notion of margin is not well-defined. Nevertheless, as a proxy measure, we look at how much distance the weights travel from their initialization in order to classify the dataset completely. Such plots have already been considered in Neyshabur et al. (2017); Nagarajan & Kolter (2017; 2019) (although in the completely different context of understanding why deep networks succeed at in-distribution generalization). We present similar plots for completeness.

Fig 5d shows this for Binary-MNIST on an FNN. For CIFAR10, we conduct two experiments. Fig 5f uses a ResNet with Adam and decaying learning rate. Here, we observe that the norms saturate after a point, which is because of the learning rate decay. Since this does not make a fair comparison between the geometries of larger datasets and smaller datasets, we also plot this for SGD with fixed learning rate in Fig 5e to recover the increasing norms observation. Here, sometimes the model sometimes saturates at an accuracy of 99% (rather than 100%); in those cases, we report the value of the weight norm at the final epoch (namely, at the 200th epoch).

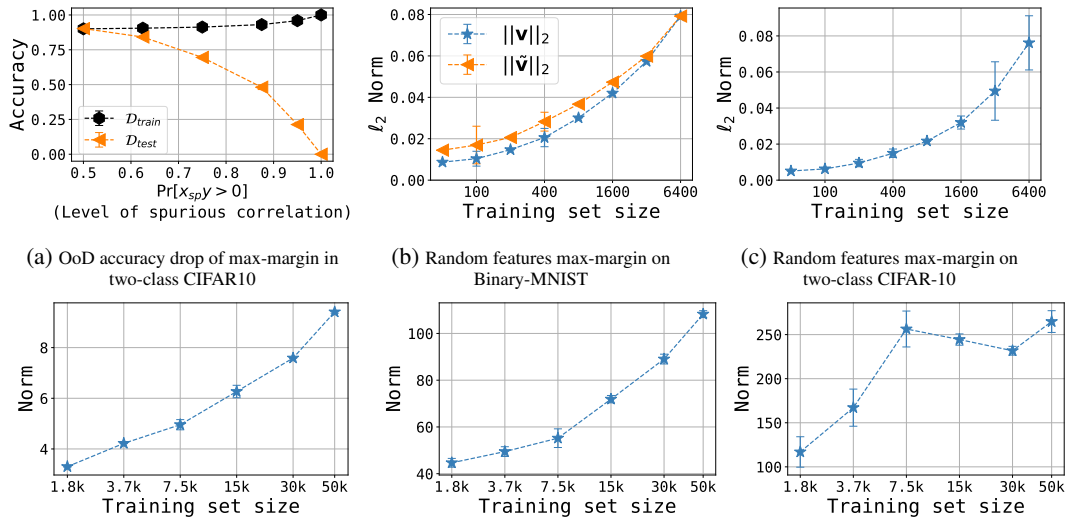
C.3 BROADER EXAMPLES OF GEOMETRIC FAILURE

We elaborate on the multiple datasets we showcased in the paper as examples where ERM fails due to a geometric skew. We first discuss the two CIFAR10 datasets, and then discuss the two Binary-MNIST datasets.

C.3.1 CIFAR10 EXAMPLE WITH SPURIOUSLY COLORED LINE

Here we provide more details about the CIFAR-10 dataset we presented in Sec 4 and in Fig 1c. This dataset can be thought of as an equivalent of the cow-camel classification tasks but for 10 classes. For this, we use ten different values of the spurious feature, one for each class. We argue that the failure here arises from the fact that the ResNet requires greater norms to fit larger datapoints (see Fig 5e), and so a similar argument as Theorem 4.1 should explain failure here.

[3] Borrowing the implementation in `https://github.com/keras-team/keras/blob/master/examples/cifar10_resnet.py`, without data augmentation.



(a) OoD accuracy drop of max-margin in two-class CIFAR10

(b) Random features max-margin on Binary-MNIST

(c) Random features max-margin on two-class CIFAR-10

(d) FNN on Binary-MNIST

(e) ResNet on CIFAR10 with SGD, fixed learning rate

(f) ResNet on CIFAR10 with Adam, (plot saturates due to decaying learning rate)

Figure 5: **Validating geometric skews in MNIST and CIFAR10:** In **Fig 5a**, we show the OOD accuracy drop of a random features based max-margin model trained to classify two classes in CIFAR10. In the next few images, we demonstrate that MNIST and CIFAR10 datasets have the property that the more the datapoints in the dataset, the larger the norm required to fit them. Specifically, in **Fig 5b** and **Fig 5c**, we plot the max-margin norms of a random features representation (see Sec B.1 for the definitions of two plotted lines). In **Fig 5d**, **Fig 5e**, **Fig 5f**, we plot the distance from initialization of neural network models (presented for the sake of completeness).

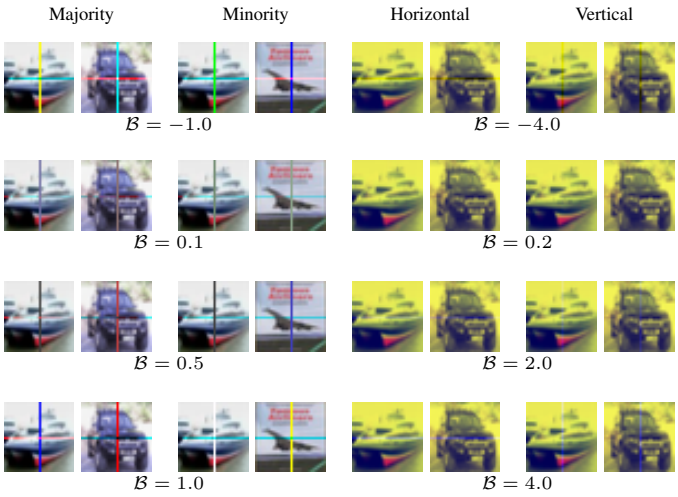
Dataset details. To create the ten-valued spurious feature, we consider a vertical line passing through the middle of each channel, and also additionally the horizontal line through the first channel. Next, we let each of these four lines take a constant value of either $(0.5 \pm 0.5\mathcal{B})$ where $\mathcal{B} \in [-1, 1]$ denotes a "spurious feature scale". Since each of these lines can take two configurations, it allows us to instantiate 16 different configurations. We'll however use only 10 of these configurations, and arbitrarily fix a mapping from those configurations to the 10 classes. For convenience let us call these ten configurations $\mathbf{x}_{\text{sp},1}, \ldots, \mathbf{x}_{\text{sp},10}$. Then, for any datapoint in class i, we denote the probability of the spurious feature taking the value $\mathbf{x}_{\text{sp},j}$, conditioned on y, as $p_{i,j}$.

To induce a spurious correlation, we set $p_{i,i} > 0.1$, and set all other $p_{i,j} := {}^{(1-p_{i,i})}/_{10}$. Thus, every value of the spurious feature $\mathbf{x}_{\text{sp},j}$ is most likely to occur with its corresponding class j. Finally, note that to incorporate the spurious pixel, we zero out the original pixels in the image, and replace them with the spurious pixels.

For the observation reported in Fig 1c, we use the value of $\mathcal{B} = 0.5$ during training and testing.

C.3.2 CIFAR10 EXAMPLE WITH A LINE IN THE THIRD CHANNEL

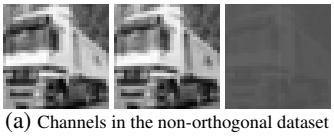
Here, we elaborate on the discussion regarding the dataset in Fig 2d. In this dataset, we add a line to the last channel of CIFAR10 (regardless of the label), and vary its brightness during testing. We argue that one way to understand the failure is via the fact that the "linear mapping" Constraint 3.5 is broken. In particular, if we imagine that each channel contains the same invariant feature $\mathbf{x}_{\text{inv}}$ (which is almost the case as can be seen in Fig 7), then for simplicity, we can imagine this dataset to be of the form $(\mathbf{x}_{\text{inv}}, \mathbf{x}_{\text{inv}} + \mathbf{x}_{\text{sp}})$ i.e., $\mathbf{x}_{\text{inv}}$ and $\mathbf{x}_{\text{sp}}$ are not orthogonal to each other. In this scenario, the second co-ordinate can still be fully predictive of the label, and therefore the max-margin classifier would rely on both the first and the second co-ordinate to maximize its margins e.g., $\mathbf{w}_1 \cdot \mathbf{x}_{\text{inv}} + \mathbf{w}_2(\mathbf{x}_{\text{inv}} + \mathbf{x}_{\text{sp}})) + b$ where $\mathbf{w}_1, \mathbf{w}_2 \neq 0$. Crucially, since the classifier has not quite disentangled $\mathbf{x}_{\text{sp}}$ from the invariant part of the second channel, this makes the classifier vulnerable to test-time shifts on $\mathbf{x}_{\text{sp}}$. In Sec A we detail this under the failure mode of Constraint 3.5, and visualize this failure in Fig 4c, and also connect it to Theorem 4.1.



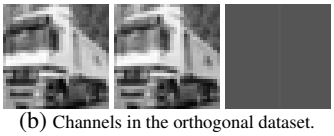
(a) Images with lines whose colors are spuriously correlated with the label.

(b) Images with lines in the third channel added regardless of label.

Figure 6: **Our CIFAR10 examples**: In **Fig 6a** we visualize the dataset discussed in Sec C.3.1. Each row corresponds to a different value of the "scale" of the spurious feature. The left two images correspond to datapoints where the spurious feature maps to the corresponding value for that label. The right two images correspond to datapoints where the spurious feature maps to one of the 9 other values. In **Fig 6b**, we visualize the dataset discussed in Sec C.3.2. The left two images correspond to adding a horizontal line to the last channel, and the right corresponds to a vertical line.



(a) Channels in the non-orthogonal dataset

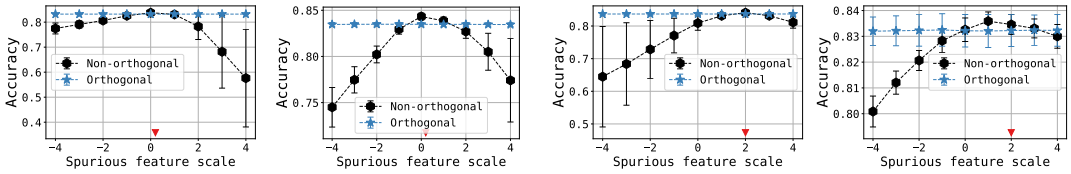


(b) Channels in the orthogonal dataset.

Figure 7: **The channels in the CIFAR-10 dataset from Section C.3.2**: In the top image, is our main dataset, where one can see that the third channel has a faint copy of the original "invariant feature" and an extra vertical line added onto it. In the bottom image, is the control dataset where the last channel only contains the vertical line (thus making the spurious feature orthogonal to the invariant features).

Dataset details. In Fig 6b, we visualize this dataset for multiple values of a spurious feature scale parameter, $\mathcal{B} \in [-4, 4]$. In particular, we take the last channel of CIFAR10 and add $\mathcal{B}$ to the middle line of the channel. Since this can result in negative pixels, we add a value of 4 to all pixels in the third channel, and then divide all those pixels by a value of $1 + 8$ so that they lie in between 0 and 1. (This normalization is the reason the color of the images differ from the original CIFAR10 dataset; as such this normalization is not crucial to our discussion.)

More experimental results. We run two kinds of experiments: one where we add only a vertical line to all images, and another where we add a horizontal line to 50% of the images (essentially simulating data from two different domains). We also run experiments for two different values of $\mathcal{B}$ during training, 0.2 and 1.0. As a "control" experiment, we completely fade out the original CIFAR image in the third channel. Then, according to our explanation, the model should not fail in this setting as data is of the form $(\mathbf{x}_{\text{inv}}, \mathbf{x}_{\text{sp}})$ i.e., the two features are orthogonally decomposed. We summarize the key observations from these experiments (plotted in Fig 8) here:



(a) Only vertical line, $\mathcal{B} = 0.2$ during training (b) Vertical or horizontal lines, $\mathcal{B} = 0.2$ during training (c) Only vertical line, $\mathcal{B} = 2$ during training (d) Vertical or horizontal lines, $\mathcal{B} = 2.0$ during training

Figure 8: **More experiments on CIFAR10 example from Sec C.3.2**: Here the red triangle corresponds to the value of the scale of spurious feature during training. 'Non-orthogonal' corresponds to our main setting, and 'Orthogonal' corresponds to the control setting where the original image in the third channel is zeroed out.

1. As predicted, we observe that when trained on the "orthogonal" dataset (where the original CIFAR10 image is zerod out in the third channel), the OoD performance of the neural network is unaffected. This provides evidence for our explanation of failure in the "non-orthogonal" dataset.
2. We observe that training on ERM on the "multidomain dataset" (Figs 8b, 8d) that has both horizontal and vertical lines, makes it more robust to test-time shifts compared to the dataset with purely vertical lines (Figs 8a, 8c).
3. Remarkably, even though the third channel has only a very faint copy of the image (Fig 7), the classifier still learns a significant enough weight on the channel that makes it susceptible to shifts in the middle line.
4. We note that introducing a different kind of test-time shift such as a Gaussian shift, is not powerful enough to cause failure since such a shift does not align with the weights learned, $\mathbf{w}_2$. However, shifts such as the line in this case, are more likely to be aligned with the classifier's weights, and hence cause a drop in the accuracy.

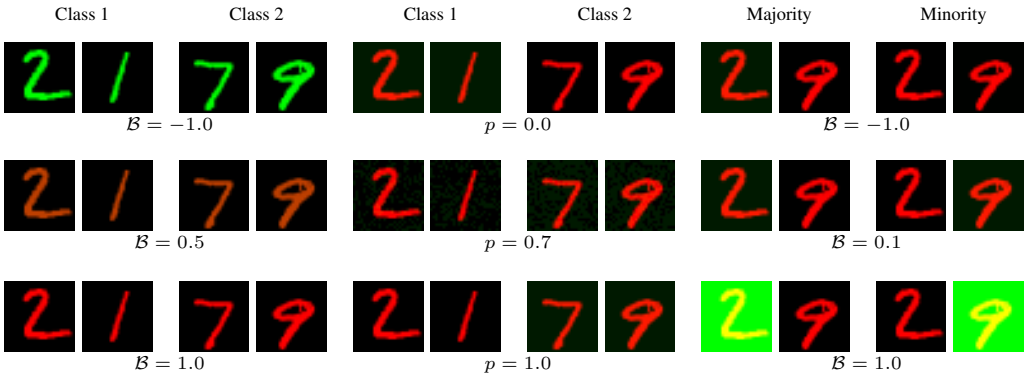
C.3.3 BINARY-MNIST EXAMPLE WITH COLOR SHIFTS

Recall that the dataset from Fig 1d consists of different domains where the Binary-MNIST digits are all given a particular color in each domain. To construct this, for each domain we pick a value $\mathcal{B} \in [-1, 1]$, and then set the first channel of every image to be $0.5 \cdot (1 + \mathcal{B})$ times the original MNIST image, and the second channel to be $0.5 \cdot (1 - \mathcal{B})$ times the original MNIST image. During test time we set $\mathcal{B}$ to have positive values so that the image is concentrated more towards the first channel, and during test-time, we flip the mass completely over to the other channel.

An implicit spurious correlation. Peculiarly, even though there is no explicit, visual spurious correlation between the color and the label here, we can still identify a different kind of non-visual spurious correlation here. To reason about this, first observe that, if the two channels correspond to $(\mathbf{x}_1, \mathbf{x}_2)$, then $\mathbf{x}_1 + \mathbf{x}_2$ is a constant across all domains (and is fully informative of the label). Hence $\mathbf{x}_1 + \mathbf{x}_2$ can hence be thought of as an invariant feature. On the other hand, consider the feature $\mathbf{x}_{\text{diff}} = \mathbf{x}_1 - \mathbf{x}_2$. During training time, this feature would correspond to a version of the MNIST digit scaled by a positive factor of $2\mathcal{B}$. In other words, this is a fully informative feature i.e., informally there must $\mathbf{w}_{\text{diff}}$ such that $y \cdot (\mathbf{w}_{\text{diff}} \cdot \mathbf{x}_{\text{diff}}) > 0$ for all the training datapoints. Then, we can imagine a single-dimensional spurious feature that corresponds to the component of the data along this direction i.e., $x_{\text{sp}} := \mathbf{w}_{\text{diff}} \cdot \mathbf{x}_{\text{diff}}$.

Notably, for all datapoints in the training set, since $x_{\text{sp}} \cdot y > 0$, we can think of this as a setting with no minority datapoints in the training set (see Fig 10a). Then, as a special case of Theorem 4.1, we can derive a positive lower bound on the component of the classifier along x_{sp}. However, during training time, since $\mathbf{x}_{\text{diff}}$ is no longer a positively scaled version of the MNIST digits, the value of $\mathbf{w}_{\text{diff}} \cdot \mathbf{x}_{\text{diff}}$ would no longer be informative of the label. This leads to an overall drop in the accuracy.

Thus, even though there is no color-label correlation in this dataset, there is still a spurious correlation in the dataset, although manifest in a way that does not visually stand out. While this is just one such novel kind of spurious correlation that we identify, the key message here is that we must not limit ourselves to thinking of spurious correlations as straightforward co-occurences between the label and a spurious object in the image.



(a) Colored identically regardless of the label.

(b) Each pixel in second channel independently drawn correlated with label.

(c) All pixels in the second channel of the same value, correlated with the label.

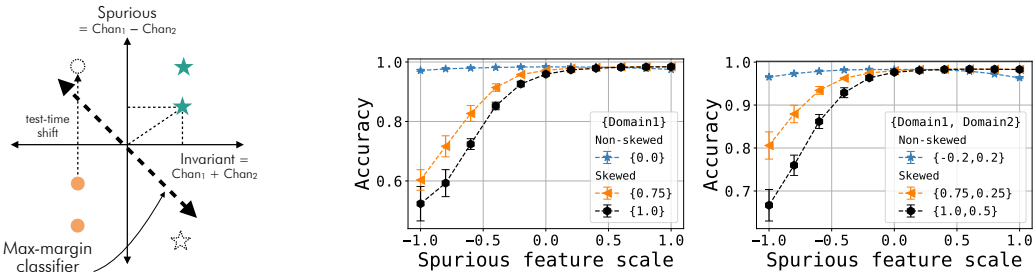
Figure 9: **Our Binary-MNIST examples**: In **Fig 9a** we present the dataset from Sec C.3.3 for various values of $\mathcal{B}$ which determines how much of the image resides in the first channel. **Fig 9b** presents the dataset from Sec C.3.4 where the second channel pixels are individually picked to align with the label with probability p (the difference is imperceptible because the pixels are either 0 or 0.1). **Fig 9c** presents the dataset for Sec C.4.1, where all pixels in the second channel are either 0 or $\mathcal{B}$.

Remark. It is worth distinguishing this spurious correlation with the one in the Colored MNIST dataset (Arjovsky et al., 2019). In Colored MNIST, the color is correlated with the label, and one can again think of $\mathbf{x}_{\text{diff}}$ as giving rise to the spurious feature. However, the manner in which this is spuriously correlated is different. More precisely, in Colored MNIST, the sign of each co-ordinate in $\mathbf{x}_{\text{diff}}$ is indicative of the true label (if it is positive, it means the image resides in the first channel, and is red, which is correlated with the label). Mathematically, if we define $\tilde{\mathbf{w}}_{\text{diff}}$ to be the vector of all ones, then we can think of $\tilde{\mathbf{w}}_{\text{diff}} \cdot \mathbf{x}_{\text{diff}}$ as a single-dimensional spurious feature here. In our case however, the vector $\mathbf{w}_{\text{diff}}$ that yields the single-dimensional spurious feature is different, and is given by the direction of separation between the two classes.

More experimental results. We conduct two sets of experiments, one where we use only a single domain to train and another with two domains. In both, settings we also consider a control setting where the data is not skewed: in the single-domain experiment, this means that we set $\mathcal{B} = 0$ (both channels have the same mass); in the two-domain experiment this means that we set $\mathcal{B}$ to be positive in one domain and negative in another. According to our explanation above, in the single-domain control setting Fig 10, since $\mathbf{x}_{\text{inv}} = 0$, the classifier is likely to not rely on this direction and should be robust to test-time shifts in $\mathbf{x}_{\text{diff}}$. In the two-domain control setting, since the classifier also sees negatively scaled images in $\mathbf{x}_{\text{diff}}$, it would be relatively robust to such negative scaling during test-time. Indeed, we observe in Fig 10, that the performance on the non-skewed, control datasets are robust. On the other hand, on the skewed datasets, we observe greater drops in OoD accuracy when $\mathcal{B}$ is flipped, thus validating our hypothesis.

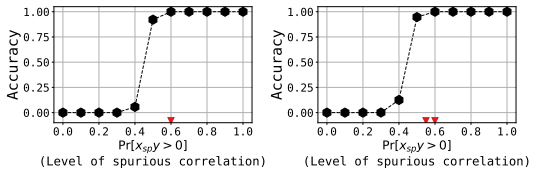
C.3.4 Binary-MNIST example with high-dimensional spurious features

In this dataset, we consider a two-channel MNIST setting where the second channel is a set of spurious pixels, each independently picked to be either 0 or 0.1 with probability $1 - p$ and p for positively labeled points and p and $1 - p$ for negatively labeled points. During training we set $p = 0.55$ and $p = 0.60$ corresponding to two training domains, and during testing, we flip this to $p = 0.0$. We visualize this dataset in Fig 9b. In Fig 11, we observe that the classifier drops to 0 accuracy during testing. To explain this failure, we can think of the sum of the pixels in the second channel as a spurious feature: since these features are independently picked, with high probability, their sum becomes fully informative of the label. As discussed in Section C.3.3, this boils down to the setting of Theorem 4.1 when there are no minority datapoints. In Appendix A, we make this argument more formal (see under the discussion of Constraint 3.4 in that section).



(a) Understanding failure (b) Single domain (c) Two domains

Figure 10: **Failure on the Binary-MNIST dataset in Sec C.3.3.** In **Fig 10a**, we visualize the failure observed on this dataset. Here we can think of the spurious feature as the difference between the two channels (projected along a direction $\mathbf{w}_{\text{diff}}$ where they are informative of the label) and the invariant feature as the sum of the two channels. **Fig 10b** and **Fig 10c** show the OoD performance under different shifts in the scale of the spurious feature $\mathcal{B}$. During training time this is set to the value given by 'Domain1' and/or 'Domain2'.



(a) Single domain (b) Two domains

Figure 11: **Experiments for the high-dimensional spurious features dataset in Sec C.3.4.** The red triangles here denote the values of p used during training.

C.4 Experiments from Sec 5 on statistical skews

Experiment on $\mathcal{D}_{\text{2-dim}}$. For the plot in Fig 3a, we train a linear model with no bias on the logistic loss with a learning rate of 0.001, batch size of 32 and training set size of 2048.

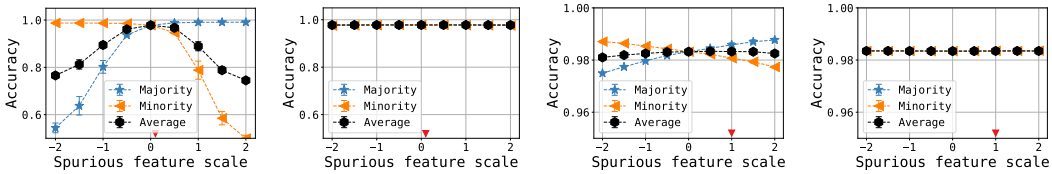
C.4.1 Binary-MNIST experiments validating the effect of statistical skews.

For this experiment, we consider a Binary-MNIST dataset where the second channel is set to be either all-0 or all-constant pixels. We visualize this dataset in Fig 9c.

To generate our control and experimental datasets, we do the following. First, we sample a set of Binary-MNIST images S_{inv} and their labels from the original MNIST distribution. Next we create two datasets S_{maj} and S_{min} by taking each of these invariant images, and appending a spurious feature to it. More precisely, we let $S_{\text{min}} = \{((x_{\text{inv}}, x_{\text{sp}}), y)$ where $x_{\text{sp}} = \mathcal{B}(y+1) | x_{\text{inv}} \in S_{\text{inv}}\}$ and $S_{\text{maj}} = \{((x_{\text{inv}}, x_{\text{sp}}), y)$ where $x_{\text{sp}} = \mathcal{B}(-y+1) | x_{\text{inv}} \in S_{\text{inv}}\}$.

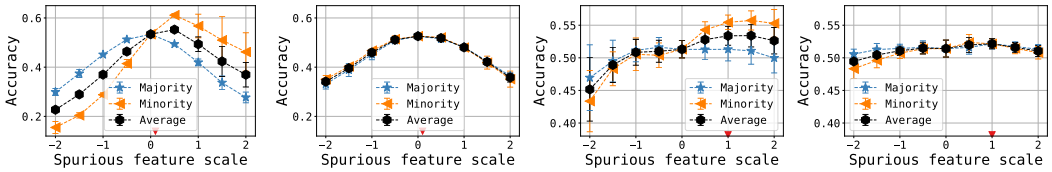
We then define a "control" dataset $S_{\text{con}} := S_{\text{maj}} \cup S_{\text{min}}$, which has a 1:1 split between the two groups of points. Next, we create an experimental "duplicated" dataset $S_{\text{exp}} := S_{\text{maj}} \cup S_{\text{min}} \cup S_{\text{dup}}$ where S_{dup} is a *large* dataset consisting of datapoints randomly chosen from S_{maj}, thus creating a spurious correlation between the label and the spurious feature. The motivation in creating datasets this way is that neither S_{con} and S_{exp} have geometric skews; however S_{exp} does have statistical skews, and so any difference in training on these datasets can be attributed to those skews.

Observations. In our experiments, we let S_{inv} be a set of $30k$ datapoints, and so S_{con} has $60k$ datapoints. We duplicate S_{min} nine times so that S_{exp} has $330k$ datapoints and has a $10:1$ ratio between the two groups. We consider two different settings, one where $\mathcal{B} = 0.1$ during training and $\mathcal{B} = 1.0$ during training. During testing, we report the accuracy under two kinds of shifts: shifting the value of $\mathcal{B}$, and also shifting the correlation completely to one direction (i.e., by concentrating



(a) Statistically skewed, with $\mathcal{B} = 0.1$ (b) No skew, with $\mathcal{B} = 0.1$ (c) Statistically skewed, with $\mathcal{B} = 1.0$ (d) No skew, with $\mathcal{B} = 1.0$

Figure 12: **Experiments validating the effect of statistical skews on the MNIST dataset in Sec C.4.1**. The red triangle denotes the value of $\mathcal{B}$ during training.



(a) Statistically skewed, with $\mathcal{B} = 0.1$ (b) No skew, with $\mathcal{B} = 0.1$ (c) Statistically skewed, with $\mathcal{B} = 1.0$ (d) No skew, with $\mathcal{B} = 1.0$

Figure 13: **Experiments validating the effect of statistical skews on the MNIST dataset in Sec C.4.1**. The red triangle denotes the value of $\mathcal{B}$ during training.

all mass on the minority/majority group). As reported in Fig 12, the statistically skewed dataset does suffer more during test-time when compared to the unskewed dataset.

C.4.2 CIFAR10 EXPERIMENTS VALIDATING THE EFFECT OF STATISTICAL SKEWS.

For this experiment, we consider the same CIFAR-10 dataset that we design in Section C.3.1 i.e., we introduce lines in the dataset, that can take 10 different color configurations each corresponding to one of the 10 different classes. Here, the scale $\mathcal{B}$ of the spurious feature varies from $[-1, 1]$ (see Section C.3.1 for more details on this).

The way we construct the control and experimental dataset requires a bit more care here since we have 10 classes. Specifically, we replicate S_{inv} ten times, and to each copy, we attach a spurious feature of a particular configuration. This creates S_{con} which has no geometric or statistical skews. Also, the has a size of $|S_{\text{con}}| = 10|S_{\text{inv}}|$. Then, we consider the $1/10$th fraction of points in S_{con} where the spurious feature has the correct configuration corresponding to the label. We create a large duplicate copy of this subset that is 81 times larger than it (we do this by randomly sampling from that subset). We add this large duplicate set to S_{con} to get S_{exp}. This gives rise to a dataset where for any label, there is a $10:1$ ratio[4] between whether the spurious feature matches the label or where it takes one of the other nine configurations.

Observations. We run experiments by setting $|S_{\text{inv}}| = 5k$ and so $|S_{\text{control}}| = 50k$ and $S_{\text{exp}} = 455k$. During training we try two different values of $\mathcal{B}$, 0.1 and 1 respectively. During testing, as in the previous section, we vary both the scale of the spurious feature and also its correlation by evaluating individually on the minority dataset (where the spurious features do not match the label) and majority datasets (where the spurious features match the label). Here, again we observe that the model trained on the non-skewed dataset is less robust, evidently due to the statistical skews.

It is worth noting that even though there is no statistical or geometric skew in the control dataset, we observe in Fig 13 (b) that the classifier is not completely robust to shifts in the spurious feature. We suspect that this may point to other causes of failure specific to how neural network models train and learn representations.

[4]Here's the calculation: in S_{control}, we have a subset of size $|S_{\text{inv}}|$ where the spurious feature is aligned with the label and in the remaining $9|S_{\text{inv}}|$ datapoint, the spurious feature is not aligned. So, if we add $81|S_{\text{inv}}|$ datapoints with matching spurious features, we'll have a a dataset where $90|S_{\text{inv}}|$ datapoints have matching spurious features while $9|S_{\text{inv}}|$ don't, thus creating the desired $10:1$ ratio.