Use Perturbations when Learning from Explanations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Machine learning from explanations (MLX) is an approach to learning that uses human-provided explanations of relevant or irrelevant features for each input to ensure that model predictions are *right for the right reasons*. Existing MLX approaches rely on local model interpretation methods and require strong model smoothing to align model and human explanations, leading to sub-optimal performance. We recast MLX as a robustness problem, where human explanations specify a lower dimensional manifold from which perturbations can be drawn, and show both theoretically and empirically how this approach alleviates the need for strong model smoothing. We consider various approaches to achieving robustness, leading to improved performance over prior MLX methods. Finally, we show how to combine robustness with an earlier MLX method, yielding state-of-the-art results on both synthetic and real-world benchmarks.[1]

1 Introduction

Deep neural networks (DNNs) display impressive capabilities, making them strong candidates for real-wold deployment. However, numerous challenges hinder their adoption in practice. Several major deployment challenges have been linked to the fact that labelled data often under-specifies the task (D'Amour et al., 2020). For example, systems trained on chest x-rays were shown to generalise poorly because they exploited dataset-specific incidental correlations such as hospital tags for diagnosing pneumonia (Zech et al., 2018; DeGrave et al., 2021). This phenomenon of learning unintended feature-label relationships is referred to as *shortcut learning* (Geirhos et al., 2020) and is a critical challenge to solve for trustworthy deployment of machine learning algorithms. A common remedy to avoid shortcut learning is to train on diverse data (Shah et al., 2022) from multiple domains, demographics, etc, thus minimizing the underspecification problem, but this may be impractical for many applications such as in healthcare.

Enriching supervision through human-provided explanations of relevant and irrelevant regions/features per example is an appealing direction toward reducing under-specification. For instance, a (human-provided) explanation for chest x-ray classification may highlight scanning artifacts such as hospital tag as irrelevant features. Learning from such human-provided explanations (MLX) has been shown to avoid known shortcuts (Schramowski et al., 2020). Ross et al. (2017) pioneered an MLX approach based on regularizing DNNs, which was followed by several others (Schramowski et al., 2020; Rieger et al., 2020; Stammer et al., 2021; Shao et al., 2021). Broadly, existing approaches employ a model interpretation method to obtain per-example feature saliency, and regularize such that model and human-provided explanations align. Since saliency is unbounded for relevant features, many approaches simply regularize the salience of irrelevant features. In the same spirit, we focus on handling a specification of irrelevant features, which we refer to as an explanation hereafter. We collectively refer to existing MLX methods as *regularization-based*.

[1]Code and data at this anonymous repository: `https://github.com/vps-anonconfs/robust_mlx`

Submitted to the Workshop "XAI in Action: Past, Present, and Future Applications" at the 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Regularization-based approaches suffer from a critical concern stemming from their dependence on a local interpretation method. MLX methods based solely on local, i.e. example-specific, explanations do not have the desired affect of reducing shortcuts globally, i.e. over the entire input domain (see Figure 1). As we demonstrate both analytically and empirically, regularization-based MLX methods require strong model smoothing in order to be globally effective at reducing shortcut learning.

In this work, we explore learning from explanations using various robust training methods with the objective of training models that are robust to perturbations of irrelevant features. We start by framing the provided human explanations as specifications of a local, lower-dimensional manifold from which perturbations are drawn. We then notice that a model whose prediction is invariant to perturbations drawn from the manifold ought also to be robust to irrelevant features. Our perspective yields considerable advantages. Posing MLX as a robustness task enables us to leverage the considerable body of prior work in robustness. Further, we show in Section 4.1 that robust training can provably upper bound the deviation on model value when irrelevant features are perturbed without needing to impose model smoothing. However, when the space of irrelevant features is high-dimensional, robust-training may not fully suppress irrelevant features as explained in Section G. Accordingly, we explore combining both robustness-based and regularization-based methods, which achieves the best results. We highlight the following contributions:

- We theoretically and empirically demonstrate that existing MLX methods require strong model smoothing owing to their dependence on local model interpretation tools.
- We study learning from explanations using robust training methods. To the best of our knowledge, we are the first to analytically and empirically evaluate robust training methods for MLX.
- We distill our insights into our final proposal of combining robustness and regularization-based methods, which consistently out-performs the best regularization method and reduces the error rate by 20-90%.

2 Problem Definition and Background

We assume access to a training dataset with N training examples, $\mathcal{D}_T = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$, with $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and $y^{(i)}$ label. In the MLX setting, a human expert also specifies input mask $\mathbf{m}^{(n)}$ for an example $\mathbf{x}^{(n)}$ where non-zero values of the mask identify *irrelevant* features of the input $\mathbf{x}^{(n)}$. An input mask is usually designed to negate a known shortcut feature that a classifier is exploiting. Figure 2 shows some examples of masks for the datasets that we used for evaluation. For example, a mask in the ISIC dataset highlights a patch that was found to confound with non-cancerous images. With the added human specification, the augmented dataset contains triplets of example, label and mask, $\mathcal{D}_T = \{(\mathbf{x}^{(i)}, y^{(i)}, \mathbf{m}^{(i)})\}_{i=0}^N$. The task therefore is to learn a model $f(\mathbf{x};\theta)$ that fits observations well while not exploiting any features that are identified by the mask $\mathbf{m}$.

The method of Ross et al. (2017) which we call Grad-Reg (short for Gradient-Regularization), and also other similar approaches (Shao et al., 2021; Schramowski et al., 2020) employ an explanation algorithm (E) to assign importance scores to input features: $IS(\mathbf{x})$, which is then regularized with an $\mathcal{R}(\theta)$ term such that irrelevant features are not regarded as important. Their training loss takes the form shown in Equation 1 for an appropriately defined task-specific loss ℓ.

$$
\begin{aligned}
IS(\mathbf{x}) &\triangleq E(\mathbf{x}, f(\mathbf{x};\theta)).\\
\mathcal{R}(\theta) &\triangleq \sum_{n=1}^{N} \|IS(\mathbf{x}^{(n)}) \odot \mathbf{m}^{(n)}\|^2.\\
\theta^* = \arg\min_\theta &\left\{ \sum_n \ell\left(f(\mathbf{x}^{(n)};\theta), y^{(n)}\right) + \lambda\mathcal{R}(\theta) + \frac{1}{2}\beta\|\theta\|^2 \right\}.
\end{aligned}
\tag{1}
$$

We use $\odot$ to denote element-wise product throughout. CDEP (Rieger et al., 2020) is slightly different. They instead use an explanation method that also takes the mask as an argument to estimate the contribution of features identified by the mask, which they minimize similarly.

3 Method

Our methodology is based on the observation that an ideal model must be robust to perturbations to the irrelevant features. Following this observation, we reinterpret the human-provided mask as a specification of a lower-dimensional manifold from which perturbations are drawn and optimize the following objective.

$$\theta^* = \arg\min_\theta \sum_n \left\{ \ell\left(f(\mathbf{x}^{(n)};\theta), y^{(n)}\right) + \alpha \max_{\boldsymbol{\epsilon}:\|\boldsymbol{\epsilon}\|_\infty \leq \kappa} \ell\left(f(\mathbf{x}^{(n)} + (\boldsymbol{\epsilon} \odot \mathbf{m}^{(n)});\theta), y^{(n)}\right) \right\} \quad (2)$$

The above formulation uses a weighting α to trade off between the standard task loss and perturbation loss and $\kappa > 0$ is a hyperparameter that controls the strength of robustness. We can leverage the many advances in robustness in order to approximately solve the inner maximization. We present them below.

Avg-Ex: We can approximate the inner-max with the empirical average of loss averaged over K samples drawn from the neighbourhood of training inputs. Singla et al. (2022) adopted this straightforward baseline for supervising using human-provided saliency maps on the Imagenet dataset. Similar to κ, we use σ to control the noise in perturbations as shown below.

$$\theta^* = \arg\min_\theta \sum_n \left\{ \ell\left(f(\mathbf{x}^{(n)};\theta), y^{(n)}\right) + \frac{\alpha}{K} \sum_{\boldsymbol{\epsilon_j} \sim \mathcal{N}(0,\sigma^2 I)}^{K} \ell\left(f(\mathbf{x}^{(n)} + (\boldsymbol{\epsilon}_j \odot \mathbf{m}^{(n)});\theta), y^{(n)}\right) \right\}$$

PGD-Ex: Optimizing for an estimate of worst perturbation through projected gradient descent (PGD) (Madry et al., 2017) is a popular approach from adversarial robustness. We refer to the approach of using PGD to approximate the second term of our loss as PGD-Ex and denote by $\epsilon^*(\mathbf{x}^{(n)}, \theta, \mathbf{m}^{(n)})$ the perturbation found by PGD at $\mathbf{x}^{(n)}$. Given the non-convexity of this problem, however, no guarantees can be made about the quality of the approximate solution $\mathbf{x}^*$.

$$\theta^* = \arg\min_\theta \sum_n \left\{ \ell\left(f(\mathbf{x}^{(n)};\theta), y^{(n)}\right) + \alpha\ell\left(f(\mathbf{x}^{(n)} + (\boldsymbol{\epsilon}^*(\mathbf{x}^{(n)}, \theta, \mathbf{m}^{(n)}));\theta), y^{(n)}\right) \right\}$$

IBP-Ex: Certified robustness approaches, on the other hand, minimize a certifiable upper-bound of the second term. A class of certifiable approaches known as interval bound propagation methods (IBP) (Mirman et al., 2018; Gowal et al., 2018) propagate input intervals to function value intervals that are guaranteed to contain true function values for any input in the input interval.

We define an input interval for $\mathbf{x}^{(n)}$ as $[\mathbf{x}^{(n)} - \kappa\mathbf{m}^{(n)}, \mathbf{x}^{(n)} + \kappa\mathbf{m}^{(n)}]$ where κ is defined in Eqn. 2. We then use bound propagation techniques to obtain function value intervals for the corresponding input interval: $\mathbf{l}^{(n)}, \mathbf{u}^{(n)}$, which are ranges over class logits. Since we wish to train a model that correctly classifies an example irrespective of the value of the irrelevant features, we wish to maximize the minimum probability assigned to the correct class, which is obtained by combining minimum logit for the correct class with maximum logit for incorrect class: $\tilde{f}(\mathbf{x}^{(n)}, y^{(n)}, \mathbf{l}^{(n)}, \mathbf{u}^{(n)};\theta) \triangleq \mathbf{l}^{(n)} \odot \bar{\mathbf{y}}^{(n)} + \mathbf{u}^{(n)} \odot (\mathbf{1} - \bar{\mathbf{y}}^{(n)})$ where $\bar{\mathbf{y}}^{(n)} \in \{0,1\}^c$ denotes the one-hot transformation of the label $y^{(n)}$ into a c-length vector for c classes. We refer to this version of the loss as IBP-Ex, summarized below.

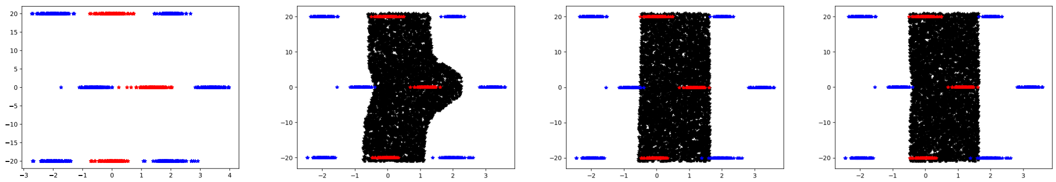
$$\begin{aligned}
\mathbf{l}^{(n)}, \mathbf{u}^{(n)} &= IBP(f(\bullet;\theta), [\mathbf{x}^{(n)} - \kappa \times \mathbf{m}^{(\mathbf{n})}, \mathbf{x}^{(n)} + \kappa \times \mathbf{m}^{(\mathbf{n})}]) \\
\tilde{f}(\mathbf{x}^{(n)}, y^{(n)}, \mathbf{l}^{(n)}, \mathbf{u}^{(n)};\theta) &\triangleq \mathbf{l}^{(n)} \odot \bar{\mathbf{y}}^{(n)} + \mathbf{u}^{(n)} \odot (\mathbf{1} - \bar{\mathbf{y}}^{(n)}) \\
\theta^* &= \arg\min_\theta \sum_n \ell\left(f(\mathbf{x}^{(n)};\theta), y^{(n)}\right) + \alpha\ell\left(\tilde{f}(\mathbf{x}^{(n)}, y^{(n)}, \mathbf{l}, \mathbf{u};\theta), y^{(n)}\right)
\end{aligned} \quad (3)$$

Combined robustness and regularization: PGD-Ex+Grad-Reg, IBP-Ex+Grad-Reg. We combine robustness and regularization by simply combining their respective loss terms. We show the objective for IBP-Ex+Grad-Reg below, PGD-Ex+Grad-Reg follows similarly.

$$\theta^* = \arg\min_\theta \sum_n \ell\left(f(\mathbf{x}^{(n)};\theta), y^{(n)}\right) + \alpha\ell\left(\tilde{f}(\mathbf{x}^{(n)}, y^{(n)}, \mathbf{l}, \mathbf{u};\theta), y^{(n)}\right) + \lambda\mathcal{R}(\theta). \quad (4)$$

$\lambda\mathcal{R}(\theta)$ and $\alpha, \tilde{f}$ are as defined in Eqn. 5 and Eqn. 3 respectively. In Section 4, F.1, we demonstrate the complementary strengths of robustness and regularization-based methods.

4 Theoretical Motivation



(a) Toy data (b) Grad-Reg with β=0 (c) Grad-Reg with β=1 (d) IBP-Ex with β=0

Figure 1: Illustration of the uneasy relationship between Grad-Reg and smoothing strength. (b) The decision boundary is nearly vertical (zero gradient wrt to nuisance y-axis value) for all training points and yet varies as a function of y value when Grad-Reg fitted using $\beta = 0$. (c) Grad-Reg requires strong model smoothing ($\beta = 1$) in order to translate local insensitivity to global robustness to x-coordinate. (d) IBP-Ex fits vertical pair of lines without any model smoothing.

In this section, we motivate the merits and drawbacks of robustness-based over regularization-based methods. Through non-parametric analysis in Theorems 1, 2, we argue that (a) regularization methods are robust to perturbations of irrelevant features (identified by the mask) only when the underlying model is sufficiently smoothed, thereby potentially compromising performance, (b) robust training upper-bounds deviation in function values when irrelevant features are perturbed, which can be further suppressed by using a more effective robust training. Although our analysis is restricted to nonparametric models for the ease of analysis, we empirically verify our claims with parametric neural network optimized using a gradient-based optimizer. We then highlight a limitation of robustness-based methods when the number of irrelevant features is large through Proposition 2.

4.1 Merits of Robustness-based methods

Consider a two-dimensional regression task, i.e. $\mathbf{x}^{(n)} \in \mathcal{X}$ and $y \in \mathbb{R}$. Assume that the second feature is the shortcut that the model should not use for prediction, and denote by $\mathbf{x}_j^{(n)}$ the j^{th} dimension of n^{th} point. We infer a regression function f from a Gaussian process prior $f \sim GP(f; 0, K)$ with a squared exponential kernel where $k(x, \tilde{x}) = \exp(-\sum_i \frac{1}{2}\frac{(x_i - \tilde{x}_i)^2}{\theta_i^2})$. As a result, we have two hyperparameters θ_1, θ_2, which are length scale parameters for the first and second dimensions respectively. Further, we impose a Gamma prior over the hyperparameters: $\mathcal{G}(\theta_i^{-2}; \alpha, \beta)$.

Theorem 1 (Grad-Reg)**.** *We infer a regression function f from a GP prior as described above with the additional supervision of $[\partial f(\mathbf{x})/\partial x_2]|_{\mathbf{x}^{(i)}} = 0, \quad \forall i \in [1, N]$. Then the function value deviations to perturbations on irrelevant feature are lower bounded by a value proportional to the perturbation strength δ as shown below.*

$$f(\mathbf{x} + [0, \delta]^T) - f(\mathbf{x}) \geq \frac{2\delta\alpha}{\beta}\Theta(x_1^2 x_2^6 + \delta x_1^2 x_2^5) \tag{5}$$

Full proof of Theorem 1 is in Appendix A, we provide the proof outline below.

We observe from Theorem 1 that if we wish to infer a function that is robust to irrelevant feature perturbations, we need to set $\frac{\alpha}{\beta}$ to a very small value. Since the expectation of gamma distributed inverse-square length parameter is $\mathbb{E}[\theta^{-2}] = \frac{\alpha}{\beta}$, which we wish to set very small, we are, in effect, sampling functions with very large length scale parameter i.e. strongly smooth functions. This result brings us to the intuitive takeaway that regularization using Grad-Reg, or any local-interpretation methods that is closed under linear operation, applies globally only when the underlying family of functions is sufficiently smooth. One could also argue that we can simply use different priors for different dimensions, which would resolve the over-smoothing issue. However, we do not have access to parameters specific to each dimension in practice and especially with DNNs, therefore only overall smoothness may be imposed such as with parameter norm regularization in Eqn. 1.

We now look at properties of a function fitted using robustness methods and argue that they bound deviations in function values better. In order to express the bounds, we introduce a numerical quantity called coverage (C) to measure the effectiveness of a robust training method. We first define a notion of inputs covered by a robust training method as $\hat{\mathcal{X}} \triangleq \{\mathbf{x} \mid \mathbf{x} \in \mathcal{X}, \ell(f(\mathbf{x};\theta), y) < \phi\} \subset \mathcal{X}$ for a small positive threshold ϕ on loss. We define coverage as the maximum distance along second coordinate between any point in $\mathcal{X}$ and its closest point in $\hat{\mathcal{X}}$, i.e. $C \triangleq \max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{x}\in\hat{\mathcal{X}}} |\mathbf{x}_2 - \hat{\mathbf{x}}_2|$. We observe that C is small if the robust training is effective. In the extreme case when training minimizes the loss for all points in the input domain, i.e. $\hat{\mathcal{X}} = \mathcal{X}$, then C=0.

Theorem 2. *When we use a robustness algorithm to regularize the network, the fitted function has the following property.*

$$|f(\mathbf{x} + [0, \delta]^T) - f(\mathbf{x})| \leq 2C\frac{\alpha}{\beta}\delta_{max} f_{max}. \tag{6}$$

δ_{max} and f_{max} are maximum values of Δx_2 and $f(\mathbf{x})$ in the input domain ($\mathcal{X}$) respectively.

Full proof is in Appendix B. The statement shows that deviations in function values are upper bounded by a factor proportional to C, which can be dampened by employing an effective robust training method. We can therefore control the deviations in function values without needing to regress $\frac{\alpha}{\beta}$ (i.e. without over-smoothing).

Empirical verification with a toy dataset. For empirical verification of our results, we fit a 3-layer feed-forward network on a two-dimensional data shown in Figure 1 (a), where color indicates the label. We consider fitting a model that is robust to changes in the second feature shown on y-axis. In Figures 1 (b), (c), we show the Grad-Reg fitted classifier using gradient ($\partial f/\partial x_2$ for our case) regularization for two different strengths of parameter smoothing (0 and 1 respectively). With weak smoothing, we observe that the fitted classifier is locally vertical (zero gradient along y-axis), but curved overall (Figure 1 (b)), which is fixed with strong smoothing (Figure 1 (c)). On the other hand, IBP-Ex fitted classifier is nearly vertical without any parameter regularization as shown in (d). This example illustrates the need for strong model smoothing when using a regularization-based method.

5 Experiments

We evaluate different methods on three datasets: one synthetic and two real-world. The synthetic dataset is similar to decoy-MNIST of Ross et al. (2017) with induced shortcuts and is presented in Section F.1. For evaluation on practical tasks, we evaluated on a plant phenotyping (Shao et al., 2021) task in Section F.2 and skin cancer detection (Rieger et al., 2020) task presented in Section 5.3 All the datasets contain a known spurious feature, and were used in the past for evaluation of MLX methods. Figure 2 summarises the three datasets, notice that we additionally require in the training dataset the specification of a mask identifying irrelevant features of the input; the patch for ISIC dataset, background for plant dataset, and decoy half for Decoy-MNIST images.

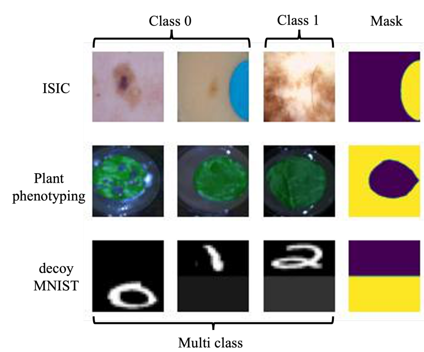


Figure 2: Sample images and masks for different datasets.

More details about experimental setup including metrics, network architecture, datasets, data splits, computing specs, and hyperparameters can be found in Appendix E.

5.1 Decoy-MNIST

Decoy-MNIST dataset is similar to MNIST-CIFAR dataset of Shah et al. (2020) where a very simple label-revealing color based feature (decoy) is juxtaposed with a more complex feature (MNIST image) as shown in Figure 1. We also randomly swap the position of decoy and MNIST parts, which makes ignoring the decoy part more challenging. We then validate and test on images where decoy part is set to correspond with random other label.

We make the following observations from Decoy-MNIST results presented in Table 1. ERM is only slightly better than a random classifier confirming the simplicity bias observed in the past (Shah et al., 2020). Grad-Reg, PGD-Ex and IBP-Ex perform comprably and better than ERM, but when combined (IBP-Ex+Grad-Reg,PGD-Ex+Grad-Reg) they far exceed their individual performances.

Dataset→	Decoy-MNIST		Plant		ISIC	
Method↓	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
ERM	15.1 ± 1.3	10.5 ± 5.4	71.3 ± 2.5	54.8 ± 1.3	77.3 ± 2.4	55.9 ± 2.3
G-DRO	64.1 ± 0.1	28.1 ± 0.1	74.2 ± 5.8	58.0 ± 4.6	66.6 ± 5.4	58.5 ± 10.7
Grad-Reg	72.5 ± 1.7	46.2 ± 1.1	72.4 ± 1.3	68.2 ± 1.4	76.4 ± 2.4	60.2 ± 7.4
CDEP	14.5 ± 1.8	10.0 ± 0.7	67.9 ± 10.3	54.2 ± 24.7	73.4 ± 1.0	60.9 ± 3.0
Avg-Ex	29.5 ± 0.3	19.5 ± 1.4	76.3 ± 0.3	64.5 ± 0.3	77.1 ± 2.1	55.2 ± 6.6
PGD-Ex	67.6 ± 1.6	51.4 ± 0.3	79.8 ± 0.3	78.5 ± 0.3	**78.7 ± 0.5**	**64.4 ± 4.3**
IBP-Ex	68.1 ± 2.2	47.6 ± 2.0	76.6 ± 3.5	73.8 ± 1.7	75.1 ± 1.2	64.2 ± 1.2
P+G	**96.9 ± 0.3**	**95.8 ± 0.4**	79.4 ± 0.5	76.7 ± 2.8	**79.6 ± 0.5**	**67.5 ± 1.1**
I+G	**96.9 ± 0.2**	**95.0 ± 0.6**	**81.7 ± 0.2**	**80.1 ± 0.3**	**78.4 ± 0.5**	**65.2 ± 1.8**

Table 1: Macro-averaged (Avg) accuracy and worst group (Wg) accuracy on (a) decoy-MNIST, (b) plant dataset, (c) ISIC dataset. Results are averaged over three runs and their standard deviation is shown after ±. I+G is short for IBP-Ex+Grad-Reg and P+G for PGD-Ex+Grad-Reg. See text for more details.

5.2 Plant Phenotyping

Plant phenotyping is a real-world task of classifying images of a plant leaf as healthy or unhealthy. Schramowski et al. (2020) discovered that standard models exploited unrelated features from the nutritional solution in the background in which the leaf is placed, thereby performing poorly when evaluated outside of the laboratory setting. Thus, we aim to regulate the model not to focus on the background of the leaf using binary specification masks indicating where the background is located. More detailed analysis of the dataset can be found in Schramowski et al. (2020).

Table 1 contrasts different algorithms on the plant dataset. We visualize the interpretations of models obtained using Smooth-Grad (Smilkov et al., 2017) trained with five different methods for three sample images from the train split in Figure 3. IBP-Ex draws features from a wider region and has more diverse pattern of active pixels, leading to higher Wg and Avg.

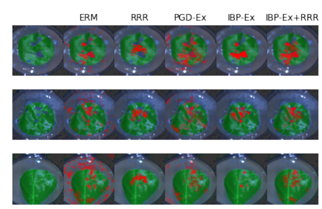


Figure 3: Visual heatmap of salient features for different algorithms on Plant data using SmoothGrad (Smilkov et al., 2017).

5.3 ISIC: Skin Cancer Detection

ISIC is a dataset of skin lesion images, which are to be classified cancerous or non-cancerous. Since half the non-cancerous images in the dataset contains a colorful patch as shown in Figure 2, standard DNN models depend on the presence of a patch for classification while compromising the accuracy on non-cancerous images without a patch (Codella et al., 2019; Tschandl et al., 2018).

We observe that Avg-Ex performed no better than ERM whereas PGD-Ex, IBP-Ex, IBP-Ex+Grad-Reg, and PGD-Ex+Grad-Reg significantly improved Wg accuracy over other baselines. The reduced accuracy gap between NPNC and C when using combined methods is indicative of reduced dependence on patch. Detailed results with error bars are shown in Table 4 of Appendix F.

Method	NPNC	PNC	C
ERM	55.9	96.5	79.6
Grad-Reg	67.1	99.0	63.2
CDEP	72.1	98.9	62.2
Avg-Ex	62.3	97.8	71.0
PGD-Ex	65.4	99.0	71.7
IBP-Ex	68.4	98.5	67.7
I+G	66.6	99.6	68.9
P+G	69.6	98.8	70.4

Table 2: Per-group accuracies on ISIC. Non-cancerous images without patch (NCNP) and with patch (NCP), and cancerous images (C).

6 Conclusions

By casting MLX as a robustness problem and using human explanations to specify the manifold of perturbations, we have shown that it is possible to alleviate the need for strong parameter smoothing of earlier approaches. Borrowing from the well-studied topic of robustness, we evaluated two strong approaches, one from adversarial robustness (PGD-Ex) and one from certified robustness (IBP-Ex).

Limitations. Detecting and specifying irrelevant regions per-example by humans is a laborious and non-trivial task. Hence, it is interesting to see the effects of learning from incomplete explanations , which we leave for future work.

References

Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

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

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 2020.

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

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Hennig, P., Osborne, M. A., and Kersting, H. P. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022. doi: 10.1017/9781316681411.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pp. 3578–3586. PMLR, 2018.

Piratla, V., Netrapalli, P., and Sarawagi, S. Focus on the common good: Group distributional robustness follows. *arXiv preprint arXiv:2110.02619*, 2021.

Rieger, L., Singh, C., Murdoch, W., and Yu, B. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pp. 8116–8126. PMLR, 2020.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Schramowski, P., Stammer, W., Teso, S., Brugger, A., Herbert, F., Shao, X., Luigs, H.-G., Mahlein, A.-K., and Kersting, K. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Shah, R., Varma, V., Kumar, R., Phuong, M., Krakovna, V., Uesato, J., and Kenton, Z. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint arXiv:2210.01790*, 2022.

Shao, X., Skryagin, A., Stammer, W., Schramowski, P., and Kersting, K. Right for better reasons: Training differentiable models by constraining their influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9533–9540, 2021.

Singla, S., Moayeri, M., and Feizi, S. Core risk minimization using salient imagenet. *arXiv preprint arXiv:2203.15566*, 2022.

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

Stammer, W., Schramowski, P., and Kersting, K. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3619–3629, 2021.

Tschandl, P., Rosendahl, C., and Kittler, H. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

Supporting material for "Use Perturbations when Learning from Explanations"

A Proof of Theorem 1

We restate the result of Theorem 1 for clarity.

The posterior mean of the function estimates marginalised over hyperparameters with Gamma prior has the following closed form.

$$f(x) \triangleq \mathbb{E}_\theta[m_x] = \int\int m_x \mathcal{G}(\theta_1^{-2};\alpha,\beta)\mathcal{G}(\theta_2^{-2};\alpha,\beta)d\theta_1^{-2}d\theta_2^{-2}$$

$$f(\mathbf{x}) = \sum_{n=1}^{N}\left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^{\alpha}\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}\left[\tilde{y}^{(n)} + \frac{\frac{\alpha}{\beta}(x_2 - x_2^{(n)})}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\tilde{y}^{(n+N)}\right]$$

$$f(\mathbf{x}+[0,\delta]^T]) - f(\mathbf{x}) \geq \frac{2\delta\alpha}{\beta}\sum_n \left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^{\alpha}\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha+1}$$
$$\left[(\alpha+1)\tilde{y}_{n+N}\left(\frac{2(x_2-x_2^{(n)})[x_2+\delta-x_2^{(n)}]}{\beta+d(x_2,x_2^{(n)})} - 1\right) - \tilde{y}_n\right]$$

Proof. We first derive the augmented set of observations ($\hat{y}$) and $\hat{K}$ explained in the main section.

$$\hat{y} = [y_1, y_2, \ldots, y_N, \partial f(\mathbf{x}^{(1)})/\partial x_2, \partial f(\mathbf{x}^{(2)})/\partial x_2, \ldots, \partial f(\mathbf{x}^{(N)})/\partial x_2]^T$$

$$k(x^{(i)}, x^{(j)}) = \begin{cases} \exp(-\frac{1}{2}\sum_{k=1}^{2}\frac{(x_k^{(i)}-x_k^{(j)})^2}{\theta_k^2}) & \text{when i, j} \leq \text{N} \\ \frac{(x_2^{(i)}-x_2^{(j)})}{\theta_2^2}\exp(-\frac{1}{2}\sum_{k=1}^{2}\frac{(x_k^{(i)}-x_k^{(j)})^2}{\theta_k^2}) & \text{when i} \leq \text{N, j>N} \\ -\frac{(x_2^{(i)}-x_2^{(j)})}{\theta_2^2}\exp(-\frac{1}{2}\sum_{k=1}^{2}\frac{(x_k^{(i)}-x_k^{(j)})^2}{\theta_k^2}) & \text{when j} \leq \text{N, i>N} \\ -2\frac{(x_2^{(i)}-x_2^{(j)})^2}{\theta_2^4}\exp(-\frac{1}{2}\sum_{k=1}^{2}\frac{(x_k^{(i)}-x_k^{(j)})^2}{\theta_k^2}) & \\ \quad +\frac{1}{\theta_2^2}\exp(-\frac{1}{2}\sum_{k=1}^{2}\frac{(x_k^{(i)}-x_k^{(j)})^2}{\theta_k^2}) & \text{when i, j > N} \end{cases}$$

These results follow directly from the results on covariance between observations of f and its partial derivative below (Hennig et al., 2022).

$$\text{cov}(f(x), \frac{\partial f(\tilde{x})}{\partial \tilde{x}}) = \frac{\partial k(x,\tilde{x})}{\partial \tilde{x}}$$
$$\text{cov}(\frac{\partial f(x)}{x}, \frac{\partial f(\tilde{x})}{\tilde{x}}) = \frac{\partial^2 k(x,\tilde{x})}{\partial x \partial \tilde{x}}$$

The posterior value of the function at an arbitrary point $\mathbf{x}$ would then be of the form $p(f(\mathbf{x}) \mid \mathcal{D}) \sim \mathcal{N}(f(\mathbf{x}); m_x, k_x)$ where m_x and k_x are have the following closed form for Gaussian prior and Gaussian likelihood in our case.

$$m_x = k(x,X)K_{XX}^{-1}\hat{y}$$
$$k_x = k(x,x) - k(x,X)K_{XX}^{-1}k(X,x)$$

Since m_x, k_x are functions of the parameters θ_1, θ_2, we obtain the closed form for posterior mean by imposing a Gamma prior over the two parameters. For brevity, we denote by $d(x,\tilde{x}) = (x-\tilde{x})^2/2$ and $\tilde{y}^{(i)}$ is the $i^{(th)}$ component of $\hat{K}_{XX}^{-1}\hat{y}$.

$$f(x) \triangleq \mathbb{E}_\theta[m_x] = \int\int m_x \mathcal{G}(\theta_1^{-2};\alpha,\beta)\mathcal{G}(\theta_2^{-2};\alpha,\beta)d\theta_1^{-2}d\theta_2^{-2}$$
$$= \int\int\left[\sum_{n=1}^{N} k(x,x^{(n)})\tilde{y}_n + \sum_{n=1}^{N}\frac{(x_2-x_2^{(n)})}{\theta_2^2}k(x,x^{(n)})\tilde{y}_{n+N}\right]\mathcal{G}(\theta_1^{-2};\alpha,\beta)\mathcal{G}(\theta_2^{-2};\alpha,\beta)d\theta_1^{-2}d\theta_2^{-2}$$

$$
\begin{aligned}
&\int\int k(\mathbf{x},\mathbf{x}^{(n)})\tilde{y}_n \mathcal{G}(\theta_1^{-2};\alpha,\beta)\mathcal{G}(\theta_2^{-2};\alpha,\beta)d\theta_1^{-2}d\theta_2^{-2}\\
&=\int\int \exp\left(-\frac{\theta_1^{-2}(x_1-x_1^{(n)})^2}{2}+\frac{\theta_2^{-2}(x_2-x_2^{(n)})^2}{2}\right)\frac{\beta^\alpha}{\Gamma(\alpha)}\theta_1^{-2\alpha+2}\exp\left(-\beta\theta_1^{-2}\right)\\
&\qquad\qquad\frac{\beta^\alpha}{\Gamma(\alpha)}\theta_2^{-2\alpha+2}\exp\left(-\beta\theta_2^{-2}\right)\tilde{y}_n d\theta_1^{-2}d\theta_2^{-2}\\
&=\left(\frac{\beta}{\beta+\frac{(x_1-x_1^{(n)})^2}{2}}\right)^\alpha\left(\frac{\beta}{\beta+\frac{(x_2-x_2^{(n)})^2}{2}}\right)^\alpha\tilde{y}_n
\end{aligned}
$$

$$
\begin{aligned}
&\int\int \frac{x_2-x_2^{(n)}}{\theta_2^2}k(\mathbf{x},\mathbf{x}^{(n)})\tilde{y}_{n+N}\mathcal{G}(\theta_1^{-2};\alpha,\beta)\mathcal{G}(\theta_2^{-2};\alpha,\beta)d\theta_1^{-2}d\theta_2^{-2}\\
&=(x_2-x_2^{(n)})\left(\frac{\beta}{\beta+\frac{(x_1-x_1^{(n)})^2}{2}}\right)^\alpha\frac{\beta^\alpha/\Gamma(\alpha)}{(\beta+\frac{(x_2-x_2^{(n)})^2}{2})^{\alpha+1}/\Gamma(\alpha+1)}\tilde{y}_{n+N}\\
&=\left(\frac{\beta}{\beta+\frac{(x_1-x_1^{(n)})^2}{2}}\right)^\alpha\frac{\alpha(x_2-x_2^{(n)})}{\beta+\frac{(x_2-x_2^{(n)})^2}{2}}\left(\frac{\beta}{\beta+\frac{(x_2-x_2^{(n)})^2}{2}}\right)^\alpha\tilde{y}_{n+N}
\end{aligned}
$$

Overall, we have the following result.

$$
f(x)=\sum_{n=1}^{N}\left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^\alpha\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha\left[\tilde{y}_n+\frac{\frac{\alpha}{\beta}(x_2-x_2^{(n)})}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\tilde{y}_{n+N}\right]
$$

We now derive the sensitivity to perturbations on the second dimension for $\Delta\mathbf{x}=[0,\delta]^T$.

$$
\begin{aligned}
f(\mathbf{x}+\Delta\mathbf{x})-f(\mathbf{x})=\sum_{n=1}^{N}\left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^\alpha&\left\{\left[\left(\frac{1}{1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta}}\right)^\alpha-\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha\right]\tilde{y}_n\right.\\
&\left.\left[\frac{\frac{\alpha}{\beta}(x_2+\delta-x_2^{(n)})}{(1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta})^{\alpha+1}}-\frac{\frac{\alpha}{\beta}(x_2-x_2^{(n)})}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}\right]\tilde{y}_{n+N}\right\} \qquad (7)
\end{aligned}
$$

Using Bernoulli inequality, $(1+x)^r\geq 1+rx$ if $r\leq 0$, we derive the following inequalities.

$$
\begin{aligned}
&\left(\frac{1}{1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta}}\right)^\alpha-\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha\\
&=\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha\left[\left(\frac{\beta+d(x_2,x_2^{(n)})}{\beta+d(x_2+\delta,x_2^{(n)})}\right)^\alpha-1\right]\\
&\geq\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha-\alpha\left[\frac{\beta+d(x_2+\delta,x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}-1\right]\\
&=\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha\alpha\left[\frac{d(x_2,x_2^{(n)})-d(x_2+\delta,x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right]\\
&\text{Assuming } |x_2-x_2^{(n)}|\gg\delta\quad\forall n\in[N] \qquad (8)\\
&\approx\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^\alpha\alpha\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right] \qquad (9)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&\frac{\frac{\alpha}{\beta}(x_2+\delta-x_2^{(n)})}{(1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta})^{\alpha+1}}-\frac{\frac{\alpha}{\beta}(x_2-x_2^{(n)})}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}} \\
\geq&\frac{\alpha}{\beta}(x_2-x_2^{(n)})\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha+1}(\alpha+1)\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right]+\frac{\delta\frac{\alpha}{\beta}}{(1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta})^{\alpha+1}} \\
\geq&\frac{\alpha}{\beta}(x_2-x_2^{(n)})\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha+1}(\alpha+1)\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right] \\
&+\frac{\delta\frac{\alpha}{\beta}}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}(\alpha+1)\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}+1\right] \\
&=\frac{\alpha+1}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}\left[\frac{-2\delta(x_2-x_2^{(n)})^2\alpha/\beta-2\delta^2\alpha/\beta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}+\frac{\delta\alpha}{\beta}\right] \\
&=\frac{-2\delta\alpha(\alpha+1)}{\beta(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}\left[\frac{-2(x_2-x_2^{(n)})[x_2+\delta-x_2^{(n)}]}{\beta+d(x_2,x_2^{(n)})}+1\right]
\end{aligned} \tag{10}
$$

Using inequalities 9, 10 in Equation 7, we have the following.

$$
\begin{aligned}
f(\mathbf{x}+\Delta\mathbf{x})-f(\mathbf{x})\geq\sum_n&\left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^{\alpha}\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha} \\
&\left[\frac{-2\delta\alpha\tilde{y}_n}{\beta+d(x_2,x_2^{(n)})}+\frac{-2\delta\alpha(\alpha+1)\tilde{y}_{n+N}}{\beta+d(x_2,x_2^{(n)})}\left(\frac{-2(x_2-x_2^{(n)})[x_2+\delta-x_2^{(n)}]}{\beta+d(x_2,x_2^{(n)})}+1\right)\right]
\end{aligned}
$$

312

$$
\begin{aligned}
f(\mathbf{x}+\Delta\mathbf{x})-f(\mathbf{x})\geq\frac{2\delta\alpha}{\beta}\sum_n&\left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^{\alpha}\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha+1} \\
&\left[(\alpha+1)\tilde{y}_{n+N}\left(\frac{2(x_2-x_2^{(n)})[x_2+\delta-x_2^{(n)}]}{\beta+d(x_2,x_2^{(n)})}-1\right)-\tilde{y}_n\right]
\end{aligned} \tag{11}
$$

Using the inequality $(1+x)^r\geq 1+rx$ if $r\leq 0$, we have

$$
\begin{aligned}
f(\mathbf{x}+\Delta\mathbf{x})-f(\mathbf{x})\geq&\frac{2\delta\alpha}{\beta}\sum_n\left\{\left(1-\frac{\alpha}{\beta}d(x_1,x_1^{(n)})\right)\left(1-\frac{\alpha+1}{\beta}d(x_2,x_2^{(n)})\right)\right. \\
&\left.\left[\frac{\alpha+1}{\beta}\tilde{y}_{n+N}\left(2(x_2-x_2^{(n)})[x_2+\delta-x_2^{(n)}](1-d(x_2,x_2^{(n)}))-1\right)-\tilde{y}_n\right]\right\} \\
=&\frac{2\delta\alpha}{\beta}\Theta(x_1^2x_2^6+\delta x_1^2x_2^5)
\end{aligned}
$$

314 □

B Proof of Theorem 2

We restate the result of Theorem 2 for clarity.

When we use an adversarial robustness algorithm to regularize the network, the fitted function has the following property.

$$
\begin{aligned}
&|f(\mathbf{x}+[0,\delta]^T) - f(\mathbf{x})| \leq \frac{\alpha}{\beta}\delta_{max} f_{max} C \\
&\text{where } C = \max_{\mathbf{x}\in\mathcal{X}} \min_{\hat{\mathbf{x}}\in\hat{\mathcal{X}}} |\mathbf{x}_2 - \hat{\mathbf{x}}_2|
\end{aligned}
$$

δ_{max} and f_{max} are maximum value of Δx_2 and $f(\mathbf{x})$ in the input domain ($\mathcal{X}$) respectively. $\hat{\mathcal{X}}$ denotes the subset of inputs covered by the robustness method. C therefore captures the maximum gap in coverage of the robustness method.

Proof. We begin by estimating the Lipschitz constant of a GP with squared exponential kernel.

$$
\begin{aligned}
&f(\mathbf{x}) = K_{xX}K_{XX}^{-1}y \\
&\frac{\partial f(x)}{\partial x_2} = \frac{\partial K_{xX}K_{XX}^{-1}y}{\partial x_2} = \tilde{K}_{xX}K_{XX}^{-1}y \\
&\text{where } [\tilde{K}_{xX}]_n = \frac{\partial}{\partial x_2}\exp(-\frac{((x_1 - x_1^{(n)})^2 + (x_2 - x_2^{(n)})^2)}{2\theta^2}) \\
&= -\frac{(x_2 - x_2^{(n)})}{\theta^2}[K_{xX}]_n \\
&\implies \frac{\partial f(x)}{\partial x_2} = -[\sum_{n=1}^{N}\frac{(x_2 - x_2^{(n)})}{\theta^2}[K_{xX}]_n]K_{XX}^{-1}y
\end{aligned}
$$

We denote with δ_{max} the maximum deviation of any input from the training points, i.e. we define δ_{max} as $\max_{\mathbf{x}\in\mathcal{X}}\min_{n\in[N]}|x_2 - x_2^{(n)}|$. Also, we denote by f_{max} the maximum function value in the input domain, i.e. $f_{max} \triangleq \max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$. We can then bound the partial derivative wrt second dimension as follows.

$$
\frac{\partial f(\mathbf{x})}{\partial x_2} \leq \frac{\delta_{max} f(\mathbf{x})}{\theta^2} \leq \frac{\delta_{max} f_{max}}{\theta^2}
$$

For any arbitrary point $\mathbf{x}$, the maximum function deviation is upper bounded by the product of maximum slope and maximum distance from the closest point covered by the adversarial distance method.

$$
|f([x_1, x_2]^T) - f([x_1, \hat{x}_2]^T)| \leq \frac{\delta_{max} f_{max}}{\theta^2} \max_{\mathbf{x}\in\mathcal{X}} \min_{\hat{\mathbf{x}}\in\tilde{\mathcal{X}}} |x_2 - \hat{x}_2| = \frac{\delta_{max} f_{max}}{\theta^2} C
$$

Therefore,

$$
|f(\mathbf{x}+[0,\delta]^T) - f(\mathbf{x})| \leq 2\frac{\delta_{max} f_{max}}{\theta^2} C
$$

Marginalising θ^{-2} with the Gamma prior leads to the final form below.

$$
|f(\mathbf{x}+[0,\delta]^T) - f(\mathbf{x})| \leq 2C\frac{\alpha}{\beta}\delta_{max} f_{max}
$$

□

C Proof of Proposition 2

We restate the result here for clarity.

Consider a regression task with $D+1$-dimensional inputs $\mathbf{x}$ where the first D dimensions are irrelevant, and assume they are $x_d = y, d \in [1, D]$ while $x_{D+1} \sim \mathcal{N}(y, 1/K)$. The MAP estimate of linear regression parameters $f(\mathbf{x}) = \sum_{d=1}^{D+1} w_d x_d$ when fitted using Avg-Ex are as follows: $w_d = 1/(D+K), \quad d \in [1, D]$ and $w_{D+1} = K/(K+D)$.

Proof. Without loss of generality, we assume α, σ^2 parameters of Avg-Ex are set to 1. In effect, our objective is to fit parameters that predict well for inputs sampled using standard normal perturbations, i.e. $\mathbf{x}^{(n)} + \mathbf{m}\epsilon, \forall n \in [1, N], \epsilon \sim \mathcal{N}(0, 1), \mathbf{m} = [1, 1, \ldots, 1, 0]^T \in \{0, 1\}^{D+1}$. The original problem therefore is equivalent to fitting on transformed input $\hat{\mathbf{x}}$ such that $\hat{\mathbf{x}}_i^{(n)} \sim \mathcal{N}(y, \sigma_i^2)$ where $\sigma_i^2 = 1$ for all $i \leq D$ and is $1/K$ when $i = D + 1$.

Likelihood of observations for the equivalent problem is obtained as follows.

$$
\begin{aligned}
P(y \mid \hat{x}_1, \hat{x}2, \ldots, \hat{x}_{D+1}) &= \prod_{i=1}^{D+1} P(y \mid \hat{x}_i) \propto \prod_{i=1}^{D+1} P(\hat{x}_i \mid y) P(y) \\
&= \prod_i \mathcal{N}(\hat{x}_i; y, \sigma_i^2) \propto \exp(-\sum_i \frac{(y - \hat{x}_i)^2}{2\sigma_i^2}) \\
&= \exp\left\{ -y^2 (\sum_i \frac{1}{2\sigma_i^2}) + y(\sum_i \frac{\hat{x}_i}{\sigma_i^2}) + \sum_i \frac{\hat{x}_i^2}{2\sigma_i^2} \right\} \\
&\propto \mathcal{N}(y; \sum_i \frac{\hat{x}_i}{\sigma_i^2} P, P) \\
\text{where } P = \frac{1}{\sum_i 1/\sigma_i^2}
\end{aligned}
$$

Substituting, the value of σ_i defined as above, we have P=D+K and the MLE estimate for the linear regression parameters are as shown in the statement. The MAP estimate also remains the same since we do not impose any informative prior on the regression weights. □

D Parametric Model Analysis

In this section we show that a similar result to what is shown for non-parametric models also holds for parametric models. We will analyse the results for a two-layer neural networks with ReLU activations. We consider a more general case of D dimensional input where the first d dimensions identify the spurious features. We wish to fit a function $f : \mathbb{R}^D \to \mathbb{R}$ such that $f(\mathbf{x})$ is robust to perturbations to the spurious features. We have the following bound when training a model using gradient regularization of Ross et al. (2017).

Proposition 1. *We assume that the model is parameterised as a two-layer network with ReLU activations such that* $f(\mathbf{x}) = \sum_j \beta_j \phi(\sum_i w_{ji} x_i + b_j)$ *where* $\vec{\beta} \in \mathbb{R}^F, \vec{w} \in \mathbb{R}^{F \times D}, \vec{b} \in \mathbb{R}^F$ *are the parameters, and* $\phi(z) = \max(z, 0)$ *is the ReLU activation. For any function such that gradients wrt to the first d features is exactly zero, i.e.* $\frac{\partial f}{\partial x_i}|_{\mathbf{x}_i^{(n)}} = 0 \quad \forall i \in [1, d], n \in [1, N]$*, we have the following bound on the function value deviations for input perturbations from a training instance* $\mathbf{x}$*:* $\tilde{x} - x = \Delta\mathbf{x} = [\Delta\mathbf{x}_{1:d}^T, \mathbf{0}_{d+1:D}^T]^T$.

$$
|f(\tilde{x}) - f(x)| = \Theta((\|\vec{\beta}\|^2 + \|\vec{w}\|_F^2)\|\Delta\mathbf{x}\|) \tag{12}
$$

For a two-layer network trained to regularize gradients wrt first d dimensions on training data, the function value deviation from an arbitrary point $\tilde{\mathbf{x}}$ from a training point $\mathbf{x}$ such that $\tilde{\mathbf{x}} - \mathbf{x} = \Delta\mathbf{x} = [\Delta\mathbf{x}_{1:d}^T, \mathbf{0}_{d+1:D}^T]^T$ is bounded as follows.

$$
|f(\tilde{x}) - f(x)| = \Theta((\|\vec{\beta}\|^2 + \|\vec{w}\|_F^2)\|\Delta\mathbf{x}\|)
$$

Proof. Recall that the function is parameterised using parameters $\vec{w}, \vec{b}, \vec{\beta}$ such that $f(\mathbf{x}) = \sum_j \beta_j \phi(\sum_i w_{ji} x_i + b_j)$ where $\vec{\beta} \in \mathbb{R}^F, \vec{w} \in \mathbb{R}^{F \times D}, \vec{b} \in \mathbb{R}^F$ are the parameters, and $\phi(z) = \max(z, 0)$ is the ReLU activation.

Since we train such that $\frac{\partial f(\mathbf{x})}{\partial x_i} = 0, \quad i \in [1, d]$, we have that $\frac{\partial f(\mathbf{x})}{x_i} = \sum_j \beta_j \hat{\phi}(\sum_i w_{ij} x_i + b_i) w_{ij}$ where $\hat{\phi}(a) = \max(\frac{a}{|a|}, 0)$.

We now bound the variation in the function value for changes in the input when moving from $\mathbf{x} \to \tilde{\mathbf{x}}$ where $\mathbf{x}$ is an instance from the training data. We define four groups of neurons based on the sign of $\sum_i w_{ji}x_i + b_j$ and $\sum_i w_{ji}\tilde{x}_i + b_j$. g_1 is both positive, g_2 is negative and positive, g_3 is positive and negative, g_4 is both negative. By defining groups, we can omit the ReLU activations as below.

$$\begin{aligned}
f(\tilde{\mathbf{x}}) - f(\mathbf{x}) &= \sum_j \beta_j \phi(\sum_i w_{ji}\tilde{x}_i + b_j) - \sum_j \beta_j \phi(\sum_i w_{ji}x_i + b_j) \\
&= \sum_{j\in g_1} \beta_j \sum_i w_{ji}(\tilde{x}_i - x_i) + \sum_{j\in g_2} \beta_j(\sum_i w_{ji}\tilde{x}_i + b_j) - \sum_{j\in g_3} \beta_j(\sum_i w_{ji}x_i + b_j) \\
&= \sum_{j\in g_1} \beta_j \sum_{i=1}^{d} w_{ji}(\tilde{x}_i - x_i) + \sum_{j\in g_2} \beta_j(\sum_{i=1}^{D} w_{ji}\tilde{x}_i + b_j) - \sum_{j\in g_3} \beta_j(\sum_{i=1}^{D} w_{ji}x_i + b_j)
\end{aligned}$$

Since we have that $\sum_{j\in g_1\cup g_3} \beta_j w_{ij} = 0, \forall i \in [1, d]$, we have

$$\begin{aligned}
&= \sum_{j\in g_1} \beta_j \sum_{i=1}^{d} w_{ji}\tilde{x}_i + \sum_{j\in g_2} \beta_j(\sum_{i=1}^{d} w_{ji}\tilde{x}_i + \sum_{i=d+1}^{D} w_{ji}x_i + b_j) - \sum_{j\in g_3} \beta_j(\sum_{i=d+1}^{D} w_{ji}x_i + b_j) \\
&\qquad \underbrace{- \sum_{j\in g_1} \beta_j \sum_{i=1}^{d} w_{ji}x_i - \sum_{j\in g_3} \beta_j \sum_{i=1}^{d} w_{ji}x_i}_{=\sum_{i=1}^{d} x_i \sum_{j\in g_1\cup g_3} \beta_j w_{ji} = 0}
\end{aligned}$$

$$= \sum_{j\in g_1\cup g_2} \beta_j \sum_{i=1}^{d} w_{ji}\tilde{x}_i + \sum_{j\in g_2} \beta_j(\sum_{i=d+1}^{D} w_{ji}x_i + b_j) - \sum_{j\in g_3} \beta_j(\sum_{i=d+1}^{D} w_{ji}x_i + b_j)$$

retaining only the terms that depend on $\Delta x = \tilde{x} - x$, the expression is further simplified as a term that grows with $\Delta\mathbf{x}$ and a constant term that depends on the value of $\mathbf{x}$

$$\begin{aligned}
&= \sum_{j\in g_1\cup g_2} \beta_j \sum_{i=1}^{d} w_{ji}\Delta x_i + \text{constant} \\
\implies &= \Theta(\|\beta\|\|\vec{w}\|_F\|\Delta\mathbf{x}\|) \quad \text{Cauchy-Schwartz inequality} \\
&= \Theta((\|\beta\|^2 + \|\vec{w}\|_F^2\|)\|\Delta\mathbf{x}\|)
\end{aligned}$$

□

E Further Experiment Details

E.1 Setup

E.1.1 Baselines

We denote by ERM the simple minimization of cross-entropy loss (using only the first loss term of Equation 1). We also compare with G-DRO(Sagawa et al., 2019), which also has the objective of avoiding to learn known irrelevant features but is supervised through group label (see Section **??**). Although the comparison is unfair toward G-DRO because MLX methods use richer supervision of per-example masks, it serves as a baseline that can be slightly better than ERM in some cases.

Regulaization-based methods. Grad-Reg and CDEP, which were discussed in Section 2. We omit comparison with Shao et al. (2021) because their code is not publicly available and is non-trivial to implement the influence-function based regularization.

Robustness-based methods. Avg-Ex, PGD-Ex, IBP-Ex along with **combined robustness and regularization methods**. IBP-Ex+Grad-Reg, PGD-Ex+Grad-Reg that are described in Section 3.

E.1.2 Metrics

We report performance using two metrics that indicate if the model is using irrelevant features (Wg Acc) without compromising the average accuracy (Avg Acc).

Avg Acc. Since the two real-world datasets contain imbalanced class populations, we only report accuracy macro-averaged over labels, simply denoted as "Avg Acc".

Wg Acc. Worst accuracy among groups where groups are appropriately defined. Different labels define the groups for decoy-MNIST and plant dataset, which therefore have ten and two groups respectively. In ISIC dataset, different groups are defined by the cross-product of label and presence or absence of the patch. We denote this metric as "Wg Acc", which is a standard metric when evaluating on datasets with shortcut features (Sagawa et al., 2019; Piratla et al., 2021).

E.1.3 Training and Implementation details

Choice of the best model. We picked the best model using the held-out validation data. We then report the performance on test data averaged over three seeds corresponding to the best hyperparameter.

Network details. We use four-layer CNN followed by three-fully connected layers for binary classification on ISIC and plant dataset following the setting in Zhang et al. (2019), and three-fully connected layers for multi classification on decoy-MNIST dataset.

E.2 Hyperparameters.

We picked the learning rate, optimizer, weight decay, and initialization for best performance with ERM baseline on validation data, which are not further tuned for other baselines unless stated otherwise. We picked the best λ for Grad-Reg and CDEP from [1, 10, 100, 1000]. Additionally, we also tuned β (weight decay) for Grad-Reg from [1e-4, 1e-2, 1, 10]. For Avg-Ex, perturbations were drawn from 0 mean and σ^2 variance Gaussian noise, where σ was chosen from [0.03, 0.3, 1, 1.5, 2]. In PGD-Ex, the worst perturbation was optimized from ℓ_∞ norm ϵ-ball through seven PGD iterations, where the best ϵ is picked from the range 0.03-5. We did not see much gains when increasing PGD iterations beyond 7, Appendix F contains some results when the number of iterations is varied. In IBP-Ex, we follow the standard procedure of Gowal et al. (2018) to linearly dampen the value of α from 1 to 0.5 and linearly increase the value of ϵ from 0 to ϵ_{max}, where ϵ_{max} is picked from 0.01 to 2. We usually just picked the maximum possible value for ϵ_{max} that converges. For IBP-Ex+Grad-Reg, we have the additional hyperparameter λ (Eqn. 4), which we found to be relatively stable and we set it to 1 for all experiments.

E.3 Data splits

We randomly split available labelled data in to training, validation, and test sets in the ratio of (0.75, 0.1, 0.15) for ISIC and (0.65, 0.1, 0.25) for Plant (similar to Schramowski et al. (2020)). We use the standard train-test splits on MNIST.

E.4 Datasets

ISIC dataset The ISIC dataset consists of 2,282 cancerous (C) and 19,372 non-cancerous (NC) skin cancer images of 299 by 299 size, each with a ground-truth diagnostic label. We follow the standard setup and dataset released by Rieger et al. (2020), which included masks with patch segmentations. In half of the NC images, there is a spurious correlation in which colorful patches are only attached next to the lesion. This group is referred to as patch non-cancerous (PNC) and the other half is referred to as not-patched non-cancerous (NPNC) Codella et al. (2019). Since trained models tend to learn easy-to-learn and useful features, they tend to take a shortcut by learning spurious features instead of understanding the desired diagnostic phenomena. Therefore, our goal is to make the model invariant to such colorful patches by providing a human specification mask indicating where they are.

decoy-MNIST dataset The MNIST dataset consists of 70,000 images of handwriting digit from 0 to 9. Each class has about 7,000 images of 28 by 28 size. We use three-fully connected layers for multi classification with 512 hidden dimension and 3 channels.

E.5 Computing

Run time and memory usage Table 3 presents the computation costs, including run time and memory usage, for each method using GTX 1080 Ti. It is worth noting that IBP-Ex has significantly less run time and memory usage compared to PGD-Ex, with a 10-fold reduction in run time and a 2.5-fold reduction in memory usage. Considering that PGD-Ex and IBP-Ex have similar performance in terms of worst group accuracy, as shown in Table 4, IBP-Ex+Grad-Reg appears to be comparably effective and efficient for model modification. Additionally, the combined method IBP-Ex+Grad-Reg, which presents the best performance in terms of averaged and worst group accuracy compared to PGD-Ex, also has a 3-fold reduction in run time and a 2-fold reduction in memory usage compared to PGD-Ex.

Grad-Reg	PGD-Ex	IBP-Ex	IBP-Ex+Grad-Reg	PGD-Ex+Grad-Reg
$\times 2.3$	$\times 4.9$	$\times 2.2$	$\times 3.5$	$\times$ 7.0

Table 3: Running time in comparison to ERM on the ISIC dataset

E.6 Network Architecture

Model architecture on the decoy-MNIST dataset

```
Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(32, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
    (3): ReLU()
    (4): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): ReLU()
    (6): Conv2d(64, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
    (7): ReLU()
    (8): Flatten(start_dim=1, end_dim=-1)
    (9): Linear(in_features=200704, out_features=1024, bias=True)
    (10): ReLU()
    (11): Linear(in_features=1024, out_features=1024, bias=True)
    (12): ReLU()
    (13): Linear(in_features=1024, out_features=2, bias=True)
  )
```

Model architecture on the ISIC dataset

```
Sequential(
    (0): Flatten(start_dim=1, end_dim=-1)
    (1): Linear(in_features=2352, out_features=512, bias=True)
    (2): ReLU()
    (3): Linear(in_features=512, out_features=512, bias=True)
    (4): ReLU()
    (5): Linear(in_features=512, out_features=512, bias=True)
    (6): ReLU()
    (7): Linear(in_features=512, out_features=10, bias=True)
    )
```

Model architecture on the Plant phenotyping dataset

```
Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(32, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
    (3): ReLU()
    (4): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): ReLU()
    (6): Conv2d(64, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
```

```
    (7): ReLU()
    (8): Flatten(start_dim=1, end_dim=-1)
    (9): Linear(in_features=200704, out_features=1024, bias=True)
    (10): ReLU()
    (11): Linear(in_features=1024, out_features=1024, bias=True)
    (12): ReLU()
    (13): Linear(in_features=1024, out_features=2, bias=True)
  )
```

F Addition Results

Method	NPNC	PNC	C	Avg	Wg
ERM	55.9 ± 2.3	96.5 ± 2.4	79.6 ± 6.6	77.3 ± 2.4	55.9 ± 2.3
G-DRO	72.4 ± 4.0	63.2 ± 14.8	64.1 ± 5.6	66.6 ± 5.4	58.5 ± 10.7
Grad-Reg	67.1 ± 4.8	99.0 ± 1.0	63.2 ± 11.3	76.4 ± 2.4	60.2 ± 7.4
CDEP	72.1 ± 5.4	98.9 ± 0.7	62.2 ± 4.7	73.4 ± 1.0	60.9 ± 3.0
Avg-Ex	62.3 ± 11.7	97.8 ± 0.8	71.0 ± 16.7	77.1 ± 2.1	55.2 ± 6.6
PGD-Ex	65.4 ± 5.4	99.0 ± 0.3	71.7 ± 6.7	**78.7 ± 0.5**	64.4 ± 4.3
IBP-Ex	68.4 ± 3.4	98.5 ± 1.0	67.7 ± 4.8	75.1 ± 1.2	64.2 ± 1.2
P+G	69.6 ± 2.8	98.84 ± 0.6	70.4 ± 4.1	**79.6 ± 0.5**	**67.5 ± 1.1**
I+G	66.6 ± 3.1	99.6 ± 0.2	68.9 ± 4.7	**78.4 ± 0.5**	**65.2 ± 1.8**

Table 4: Macro-averaged (Avg) accuracy and worst group (Wg) accuracy on ISIC dataset. Also shown are the average precision scores for each of the three groups. All the results are averaged over three runs and their standard deviation is shown after $\pm$. Note that the worst group for each run can be different

F.1 Decoy-MNIST

Decoy-MNIST dataset is similar to MNIST-CIFAR dataset of Shah et al. (2020) where a very simple label-revealing color based feature (decoy) is juxtaposed with a more complex feature (MNIST image) as shown in Figure 1. We also randomly swap the position of decoy and MNIST parts, which makes ignoring the decoy part more challenging. We then validate and test on images where decoy part is set to correspond with random other label.

We make the following observations from Decoy-MNIST results presented in Table 1. ERM is only slightly better than a random classifier confirming the simplicity bias observed in the past (Shah et al., 2020). Grad-Reg, PGD-Ex and IBP-Ex perform comprably and better than ERM, but when combined (IBP-Ex+Grad-Reg,PGD-Ex+Grad-Reg) they far exceed their individual performances.

In order to understand the surprising gains when combining regularization and robustness methods, we draw insights from gradient explanations on images from train split for Grad-Reg and IBP-Ex. We looked at $s_1 = \mathcal{M}\left[\left\|\mathbf{m}^{(n)} \times \frac{\partial f(\mathbf{x}^{(n)})}{\partial \mathbf{x}^{(n)}}\right\|\right]$ and $s_2 = \mathcal{M}\left[\left\|\mathbf{m}^{(n)} \times \frac{\partial f(\mathbf{x}^{(n)})}{\partial \mathbf{x}^{(n)}}\right\| \Big/ \left\|(\mathbf{1} - \mathbf{m}^{(n)}) \times \frac{\partial f(\mathbf{x}^{(n)})}{\partial \mathbf{x}^{(n)}}\right\|\right]$, where $\mathcal{M}[\bullet]$ is the median function. For an effective algorithm, we expect both s_1, s_2 to be close to zero. However, the values of s_1, s_2 is 2.3e-3, 0.26 for the best model fitted using Grad-Reg and 6.7, 0.05 for IBP-Ex. We observe that Grad-Reg has lower s_1 while IBP-Ex has lower s_2, which shows that Grad-Reg is good at dampening the contribution of decoy part but also dampened contribution of non-decoy likely due to over-smoothing. IBP-Ex improves the contribution of the non-decoy part but did not fully dampen the decoy part likely because high dimensional space of irrelevant features, i.e. half the image is irrelevant and each pixel is indicative of the label. When combined, IBP-Ex+Grad-Reg has low s_1, s_2, which explains the increased performance when they are combined.

F.2 Plant Phenotyping

Plant phenotyping is a real-world task of classifying images of a plant leaf as healthy or unhealthy. About half of leaf images are infected with a Cercospora Leaf Spot (CLS), which are the black spots on leaves as shown in the first image in the second row of Figure 2. Schramowski et al.

(2020) discovered that standard models exploited unrelated features from the nutritional solution in the background in which the leaf is placed, thereby performing poorly when evaluated outside of the laboratory setting. Thus, we aim to regulate the model not to focus on the background of the leaf using binary specification masks indicating where the background is located. Due to lack of out-of-distribution test set, we evaluate with in-domain test images but with background pixels replaced by a constant pixel value, which is obtained by averaging over all pixels and images in the training set. We replace with an average pixel value in order to avoid any undesired confounding from shifts in pixel value distribution. More detailed analysis of the dataset can be found in Schramowski et al. (2020).

Table 1 contrasts different algorithms on the plant dataset. All the algorithms except CDEP improve over ERM, which is unsurprising given our test data construction; any algorithm that can divert focus from the background pixels can perform well. Wg accuracy of robustness (except Avg-Ex) and combined methods far exceed any other method by 5-12% over the next best baseline and by 19-26% accuracy point over ERM. Surprisingly, even Avg-Ex has significantly improved the performance over ERM likely because spurious features in the background are spiky or unstable, which vanish under normal perturbation.

We visualize the interpretations of models obtained using SmoothGrad (Smilkov et al., 2017) trained with five different methods for three sample images from the train split in Figure 3. As expected, ERM has strong dependence on non-leaf background features. Although Grad-Reg features are all on the leaf, they appear to be localized to a small region on the leaf, which is likely due to over-smoothing effect of its loss. IBP-Ex, IBP-Ex+Grad-Reg on the other hand draws features from a wider region and has more diverse pattern of active pixels.

F.3 ISIC skin cancer dataset

ISIC is a dataset of skin lesion images, which are to be classified cancerous or non-cancerous. Since half the non-cancerous images in the dataset contains a colorful patch as shown in Figure 2, standard DNN models depend on the presence of a patch for classification while compromising the accuracy on non-cancerous images without a patch (Codella et al., 2019; Tschandl et al., 2018). We follow the standard setup and dataset released by Rieger et al. (2020), which include masks highlighting the patch.

We identify three groups in the dataset, non-cancerous images without patch (NCNP) and with patch (NCP), and cancerous images (C). In Table 2, we report on per-group accuracies for different algorithms. Detailed results with error bars are shown in Table 4 of Appendix F. The Wg accuracy (of Table 1) may not match with the worst of the average group accuracies in Table 2 because we report average of worst accuracies. We now make the following observations. ERM performs the worst on the NPNC group confirming that predictions made by a standard model depend on the patch. The accuracy on the PNC group is high overall perhaps because PNC group images are at a lower scale (see middle column of Figure 2 for an example) are systematically more easier to classify even when the patch is not used for classification. Although human-explanations for this dataset, which only identifies the patch if present, do not full specify all spurious correlations, we still saw gains when learning from them. Grad-Reg and CDEP improved NPNC accuracy at the expense of C's accuracy while still performing relatively poor on Wg accuracy. Avg-Ex performed no better than ERM whereas PGD-Ex, IBP-Ex, IBP-Ex+Grad-Reg, and PGD-Ex+Grad-Reg significantly improved Wg accuracy over other baselines. The reduced accuracy gap between NPNC and C when using combined methods is indicative of reduced dependence on patch.

F.4 Overall results

Among the regularization-based methods, Grad-Reg performed the best while also being simple and intuitive. CDEP surprisingly performed worse than ERM on Decoy-MNIST and Plant datasets despite our best efforts, which are elaborated in Appendix H.

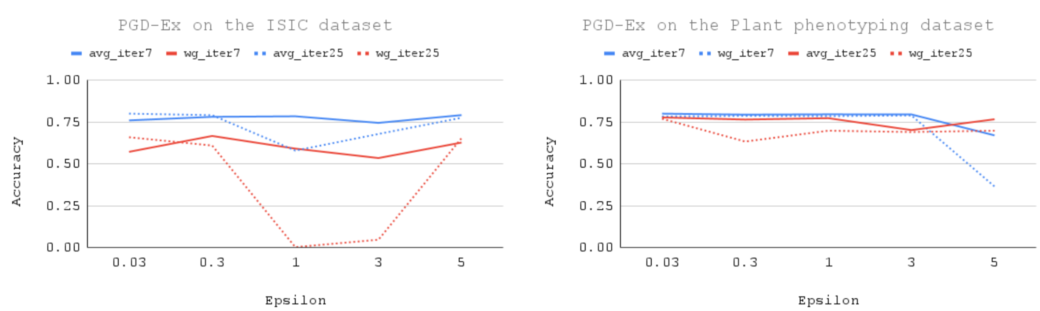
Robustness-based methods except Avg-Ex are consistently and effortlessly better or comparable to regularization-based methods on all the benchmarks with an improvement to Wg accuracy by 3-10% on the two real-world datasets. Combined methods are better than their constituents on all the datasets readily without much hyperparameter tuning.

Comparison of PGD-Ex and IBP-Ex It is difficult to compare the worst group accuracy of IBP-Ex (64.2) and PGD-Ex (64.4) due to the comparably high standard deviation of PGD-Ex (4.3). Therefore, we additionally compare the accuracy drop when colorful patches are removed from images in the PNC group in Table 5. We replace the colorful patch of the image with its mean value, making it looks like a background skin color. Note that we evaluate the robustness to concept-level perturbations rather than pixel-level perturbations, as our focus is on avoiding spurious concept features rather than robustness to adversarial attacks. Interestingly, the accuracy drops about 17% and 37% in IBP-Ex and PGD-Ex, respectively, showing that IBP-Ex is more robust to concept perturbations. This can be explained by the effectiveness of robustness methods in covering the epsilon ball with the center of each input point defined in a low-dimensional manifold annotated in the human specification mask. IBP guarantees robustness on any possible pixel combination within the epsilon ball while PGD only considers the worst case in the epsilon ball. When the inner maximization to find the PGD attack is non-convex, an inappropriate local worst case is found instead of the global one. Thus, IBP-Ex shows better robustness when spurious concepts are removed, which involves large perturbations on irrelevant parts within the defined epsilon ball. The combined method IBP-Ex+Grad-Reg, where Grad-Reg compensates for the practical limitations of the training procedure of IBP-Ex, shows about 1% higher worst group accuracy than IBP-Ex alone.

Method	PNC	PNC (Remove patch)
PGD-Ex	99.0 ± 0.3	62.2 ± 17.0
IBP-Ex	98.5 ± 1.0	81.6 ± 16.5
IBP-Ex+Grad-Reg	99.6 ± 0.2	$\mathbf{82.5 \pm 9.5}$

Table 5: Comparison between robustness based methods. Macro-averaged accuracy and regval loss before and after removing color patch part of images in PNC group on ISIC dataset.

Results of PGD-Ex with different epsilon and iteration number. We experimented with different values of epsilon and iteration numbers on the ISIC and Plant phenotyping datasets. The epsilon values tested were 0.03, 0.3, 1, 3, and 5, and the iteration numbers were 7 and 25. In Figure 4, the results on the ISIC dataset showed that using an iteration of 7 with different epsilon values resulted in stable results, but using an iteration of 25 resulted in unstable worst group accuracy. However, in the Plant phenotyping dataset, we found that both average and worst group accuracy were similar regardless of the epsilon and iteration values used.



(a) PGD-Ex on the ISIC data

(b) PGD-Ex on the Plant phenotying data

Figure 4: PGD-Ex results on the ISIC and Plant phenotyping dataset with different epsilon and iteration numbers in (a) and (b), respectively.

G Drawbacks of Robustness-based methods

Although robust training is appealing in low dimensions, their merits do not transfer well when the space of irrelevant features is high-dimensional owing to difficulty in solving the inner maximization of Eqn. 2. Sub-par estimation of the maximization term may learn parameters that still depend on the irrelevant features. We demonstrate this below with a simple exercise.

Proposition 2. *Consider a regression task with $D+1$-dimensional inputs $\mathbf{x}$ where the first D dimensions are irrelevant, and assume they are $x_d = y, d \in [1, D]$ while $x_{D+1} \sim \mathcal{N}(y, 1/K)$. The*

MAP estimate of linear regression parameters $f(\mathbf{x}) = \sum_{d=1}^{D+1} w_d x_d$ when fitted using Avg-Ex are as follows: $w_d = 1/(D+K), \quad d \in [1, D]$ and $w_{D+1} = K/(K+D)$.

We present the proof in Appendix C. We observe that as D increases, the weight of the only relevant feature (x_{D+1}) diminishes. On the other hand, the weight of the average feature: $\frac{1}{D}\sum_{d=1}^{D} x_d$, which is $D/(D+K)$ approaches 1 as D increases. This simple exercise demonstrates curse of dimensionality for robustness-based methods. For this reason, we saw major empirical gains when combining robustness methods with a regularization method especially when the number of irrelevant features is large such as in the case of Decoy-MNIST dataset, which is described in the next section.

Further remarks on sources of over-smoothing in regularization-based methods. We empirically observed that the term $\mathcal{R}(\theta)$ (of Eqn. 1), which supervises explanations, also has a smoothing effect on the model when the importance scores (IS) are not well normalized, which is often the case. This is because reducing IS($\mathbf{x}$) everywhere will also reduce saliency of irrelevant features.

H Discussion on poor CDEP performance

In Table 4, CDEP demonstrates better performance in worst group accuracy compared to ERM on the ISIC dataset. However, it fails to surpass RRR, which contradicts results from previous research in Rieger et al. (2020) where CDEP was found to perform better than RRR. This discrepancy may be attributed to the fact that Rieger et al. (2020) used different metrics (F1 and AUC) and employed a pretrained VGG model to estimate the contribution of mask features, whereas in our study we used worst group accuracy and employed a four-layer CNN followed by three fully connected layers without any pretraining. We do not use a pre-trained model for CDEP in order to make a fair comparison to other methods. As a result, CDEP also fails to improve worst group accuracy over ERM on the Plant Phenotyping and Decoy-MNIST datasets. We further illustrate the interpretations of CDEP on the Plant Phenotyping dataset using Smooth Gradient in Figure 5. In comparison to the interpretations of other methods shown in Figure 3 in the main paper, CDEP appears to focus primarily on the spurious agar part instead of the main leaf part.

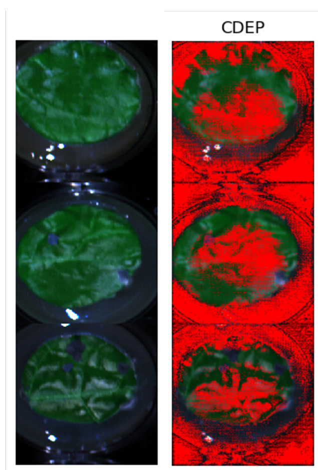


Figure 5: Visual heatmap of salient features for CDEP on three sample images from the train split of Plant phenotyping data. Importance score from SmoothGrad Smilkov et al. (2017) method is normalized between 0 to 1 and visualized with a threshold 0.6.