Supporting material for "Use Perturbations when Learning from Explanations"

A Proof of Theorem 1

We restate the result of Theorem 1 for clarity.

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.

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

Proof outline. We first show that the posterior mean of the function estimates marginalised over hyperparameters with Gamma prior has the following closed form where $d(x, y) = (x - y)^2/2$ and $\tilde{y}$ denotes original observations y augmented with observations on gradients, which is described in more detail further below.

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

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

We then derive the following lower bound on the function value deviations and finally use simple inequalities to arrive at the final result.

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

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

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

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

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.

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

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 $\hat{K}_{XX}^{-1}\hat{y}$.

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

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

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

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 $\Delta\mathbf{x} = [0, \delta]^T$.

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

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

$$\left(\frac{1}{1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta}}\right)^{\alpha}-\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}$$

$$=\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}\left[\left(\frac{\beta+d(x_2,x_2^{(n)})}{\beta+d(x_2+\delta,x_2^{(n)})}\right)^{\alpha}-1\right]$$

$$\geq\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}-\alpha\left[\frac{\beta+d(x_2+\delta,x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}-1\right]$$

$$=\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}\alpha\left[\frac{d(x_2,x_2^{(n)})-d(x_2+\delta,x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right]$$

$$\text{Assuming } |x_2-x_2^{(n)}|\gg\delta \quad \forall n\in[N] \tag{8}$$

$$\approx\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}\alpha\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right] \tag{9}$$

Similarly,

$$\frac{\frac{\alpha}{\beta}(x_2+\delta-x_2^{(n)})}{(1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta})^{\alpha+1}}-\frac{\frac{\alpha}{\beta}(x_2-x_2^{(n)})}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}$$

$$\geq\frac{\alpha}{\beta}(x_2-x_2^{(n)})\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha+1}(\alpha+1)\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right]+\frac{\delta\frac{\alpha}{\beta}}{(1+\frac{d(x_2+\delta,x_2^{(n)})}{\beta})^{\alpha+1}}$$

$$\geq\frac{\alpha}{\beta}(x_2-x_2^{(n)})\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha+1}(\alpha+1)\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}\right]$$

$$+\frac{\delta\frac{\alpha}{\beta}}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}(\alpha+1)\left[\frac{-2\delta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}+1\right]$$

$$=\frac{\alpha+1}{(1+\frac{d(x_2,x_2^{(n)})}{\beta})^{\alpha+1}}\left[\frac{-2\delta(x_2-x_2^{(n)})^2\alpha/\beta-2\delta^2\alpha/\beta(x_2-x_2^{(n)})}{\beta+d(x_2,x_2^{(n)})}+\frac{\delta\alpha}{\beta}\right]$$

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

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

$$f(\mathbf{x}+\Delta\mathbf{x})-f(\mathbf{x})\geq\sum_n\left(\frac{1}{1+\frac{d(x_1,x_1^{(n)})}{\beta}}\right)^{\alpha}\left(\frac{1}{1+\frac{d(x_2,x_2^{(n)})}{\beta}}\right)^{\alpha}$$

$$\left[\frac{-2\delta\alpha\tilde{y}_n}{\beta+d(x_2,x_2^{(n)})}+\frac{-2\delta\alpha(\alpha+1)\tilde{y}_{n+N}}{\beta+d(x_2,x_2^{(n)})}\left(\frac{-2(x_2-x_2^{(n)})[x_2+\delta-x_2^{(n)}]}{\beta+d(x_2,x_2^{(n)})}+1\right)\right]$$

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

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

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

□

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.

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

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 1

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 2. *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

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

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

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 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.2 Metrics

Relative Core Sensitivity (RCS) (Singla et al., 2022). The metric measures the relative dependence of the model on core features and is normalised such that the best value is 100. Higher value of RCS imply that the model is exploiting core features more than the spurious features.

$$RCS = 100 \times \frac{acc^{(C)} - acc^{(S)}}{2\min(\bar{a}, 1-\bar{a})}, \text{ where } \bar{a} = \frac{acc^{(C)} + acc^{(S)}}{2}$$

$$acc^C \triangleq \frac{1}{N}\sum_n \mathbb{1}\left(f\left(\mathbf{x}^{(n)} + \sigma(\mathbf{z} \odot \mathbf{m}^{(n)}); \theta\right) = y^{(n)}\right)$$

$$acc^S \triangleq \frac{1}{N}\sum_n \mathbb{1}\left(f\left(\mathbf{x}^{(n)} + \sigma(\mathbf{z} \odot (1 - \mathbf{m}^{(n)})); \theta\right) = y^{(n)}\right)$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\sigma = 0.25$ for all experiments. The interpretation of $acc^{(C)}$ is the accuracy when noise is added outside of core region, and $acc^{(S)}$ is the accuracy when noise is added outside of spurious region.

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, (0.65, 0.1, 0.25) for Plant (similar to Schramowski et al. (2020)) and (0.6, 0.15, 0.25) for Salient-Imagenet. 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.

Salient-Imagenet. The six classes we considered are *Rhinoceros Beetle, Dowitcher, Alaskan Tundra Wolf, Dragonfly, Gorilla, Snoek Fish.*

E.5 Computational cost

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 5, 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
×2.3	×4.9	×2.2	×3.5	× 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 Additional Results

F.1 Standard deviations

We repeated all our experiments on Decoy-MNIST, Plant and ISIC dataset three times and report the mean and standard deviation in Table 4. Similarly, we report in Table 5, the standard deviations for the corresponding Table 2 of the main paper.

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 4: 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.

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 5: 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 ±. Note that the worst group for each run can be different

F.2 Additional results on Salient-Imagenet

In Table 6, we show average accuracy and accuracy when noise (drawn from standard normal) is added to spurious or irrelevant regions (N-Acc column). We observe that for PGD-Ex + Grad-Reg, the accuracy did not diminish by much when noise is added to the spurious region.

F.3 Comparison of PGD-Ex and IBP-Ex

In Table 5, 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

Method	Accuracy	N-Acc ↑	RCS ↑
ERM	96.4	87.5	47.9
Grad-Reg	88.3	82.2	52.5
PGD-Ex	93.8	90.2	58.7
PGD-Ex + Grad-Reg	**94.6**	**93.8**	**65.0**

Table 6: The columns in that order are the average accuracy, accuracy when noise is added to spurious (or irrelevant) regions and RCS value for our Salient-Imagenet data setup.

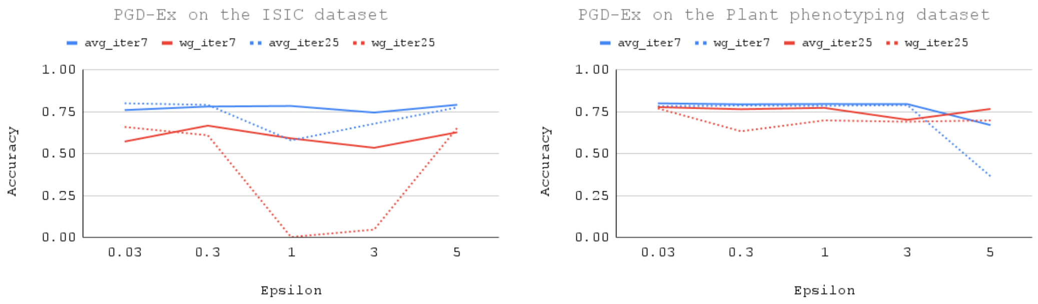
the accuracy drop when colorful patches are removed from images in the PNC group in Table 7. We replace the colorful patch of the image with its mean value, making it looks like a background skin color. Note that we evaluate the robustness to concept-level perturbations rather than pixel-level perturbations, 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	**82.5** ± **9.5**

Table 7: 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.

F.4 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.

F.5 Out-of-distribution scenarios on the Plant data

In the main paper, we follow the dataset construction from Schramowski et al. (2020) to replace background with the average pixel value, which is obtained from train split. Here, in Table 8 we evaluated on a test set obtained by adding varying magnitude of noise to the background to test methods under out-of-distribution scenarios. We observe that robustness and regularization methods when combined led to a model that is far more robust to noise in the background, aligning with our original results on the plant dataset.

	Noise ($\mathcal{N}(0,1)$)		Noise ($\mathcal{N}(0,10)$)		Noise ($\mathcal{N}(0,30)$)	
	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
ERM	59.8 ± 11.9	43.5 ± 2.0	57.4 ± 7.6	38.1 ± 4.8	55.8 ± 1.7	22.0 ± 3.7
Grad-Reg	71.6 ± 2.0	66.1 ± 1.8	68.7 ± 6.2	53.4 ± 4.3	56.1 ± 3.3	34.8 ± 1.8
PGD-Ex+Grad-Reg	69.8 ± 1.8	67.2 ± 2.1	69.5 ± 3.7	60.6 ± 4.8	67.5 ± 4.5	50.8 ± 2.4

Table 8: Out-of-distribution scenarios on the Plant data

F.6 Generality to new explanation methods: Integrated-gradient and CDEP

In Table 9, we introduced evaluation using Integrated-gradient (Sundararajan et al., 2017) based regularization and also added evaluation with PGD-Ex+CDEP that we did not originally include on Decoy-MNIST dataset.

Alg.	Avg Acc	Wg Acc
Integrated-Grad	26.7 ±1.3	17.6 ±1.2
CDEP	14.5 ±1.8	10.0 ±0.7
PGD-Ex	67.6 ±1.6	51.4 ±0.3
PGD-Ex+Integrated-Grad	80.5 ±2.1	**62.1 ± 6.8**
PGD-Ex+CDEP	**84.8 ±0.8**	**64.2 ±1.6**

Table 9: PGD-Ex+Integrated-Grad and PGD-Ex+CDEP on the Decoy-MNIST data

F.7 Generality to new model architecture: Attention map-based (ViT)

Using a Visual transformer architecture (of depth 3 and width 128), we evaluated regularization using local explanations obtained using an attention map – regularization based on attention maps was used to supervise prior knowledge in Miao et al. (2022) and is called SPAN. We obtained saliency explanations on inputs using the procedure proposed in Miao et al. (2022), shown in Table 10

Alg.	Avg Acc	Wg Acc
ERM	10.0 ±0.3	8.1 ±0.3
SPAN	19.0 ±0.3	8.1 ±0.3
PGD-Ex	**64.6 ±4.7**	**37.4 ±3.5**
PGD-Ex+SPAN	**63.1 ±2.6**	**39.4 ± 2.9**

Table 10: Attention Map based local explanations with ViT on the Decoy-MNIST data

F.8 Sensitivity to hyperparameters on Decoy-MNIST dataset

In Table 11, we show sensitivity to hyperparameters on Decoy-MNIST dataset. In summary, results are broadly stable with the choice of hyperparameters, and we did not extensively search for the best hyperparameters.

Decoy-MNIST	Lambda (Grad-Reg)	Eps (PGD-Ex)	Avg -Acc	Wg -Acc
PGD-Ex + Grad-Reg	**1**	**3**	**96.9 ± 0.3**	**95.8 ± 0.4**
	0.1	3	96.8 ± 0.8	94.2 ± 0.2
	1.5	3	95.6 ± 1.0	93.0 ± 1.0
	5	3	91.6 ± 0.9	87.6 ± 2.3
	0.0001	3	75.5 ± 0.9	57.2 ± 3.6
	1	1	95.1 ± 2.0	91.3 ± 3.9
	1	2	95.4 ± 1.8	92.0 ± 1.6
	1	4	97.5 ± 0.2	95.4 ± 0.8
	1	5	93.8 ± 1.6	86.3 ± 3.1
	1	0.1	58.5 ± 7.9	30.0 ± 2.7
	1	0.0001	59.5 ± 1.1	40.1 ± 2.0
PGD-Ex	**0**	**3**	**67.6 ± 1.6**	**51.4 ± 0.3**
	0	0.0001	16.5 ± 1.1	15.4 ± 2.7
	0	0.1	19.1 ± 0.7	13.6 ± 0.6
	0	1	62.5 ± 1.0	40.1 ± 2.2
	0	2	74.6 ± 5.6	52.8 ± 4.8
	0	4	71.9 ± 8.5	58.6 ± 12.9
	0	5	57.0 ± 3.1	42.6 ± 4.2
Grad-Reg	**10**	**0**	**64.1 ± 0.1**	**28.1 ± 0.1**
	5	0	39.2 ± 2.2	21.5 ± 0.6
	20	0	49.1 ± 2.7	33.2 ± 4.3
	100	0	50.1 ± 0.4	35.2 ± 2.3
	500	0	48.0 ± 0.9	36.2 ± 1.7
	1000	0	49.6 ± 1.7	30.5 ± 6.1

Table 11: Sensitivity to hyperparameters on Decoy-MNIST dataset

G Discussion on poor CDEP performance

Regarding ISIC dataset discrepancy: In Table 5, 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.

Regarding DecoyMNIST dataset discrepancy: Note that our Decoy-MNIST setting is inspired from decoy-mnist of CDEP (Rieger et al., 2020), but not the same. All the methods were found to be equally good on the original decoy-mnist dataset (Rieger et al., 2020), which is why we had to alter the dataset to be more challenging. A key difference is that the volume of spurious/simple features in our version of decoy-mnist dataset is much higher, making it harder to remove dependence of a model on decoy/spurious features. This explains why there is the performance gap on this dataset reported in our paper and CDEP (Rieger et al., 2020).

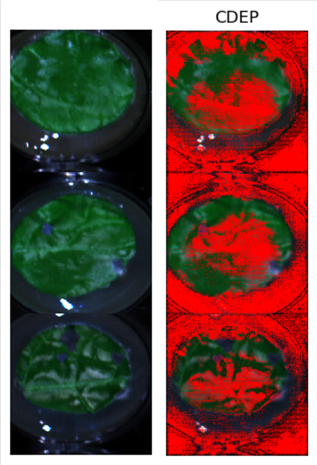


Figure 5: Visual heatmap of salient features for CDEP on three sample images from the train split of Plant phenotyping data. Importance score from SmoothGrad Smilkov et al. (2017) method is normalized between 0 to 1 and visualized with a threshold 0.6.