Regulating Model Reliance on Non-Robust Features by Smoothing Marginal Density of Input

Anonymous authors
Paper under double-blind review

Abstract

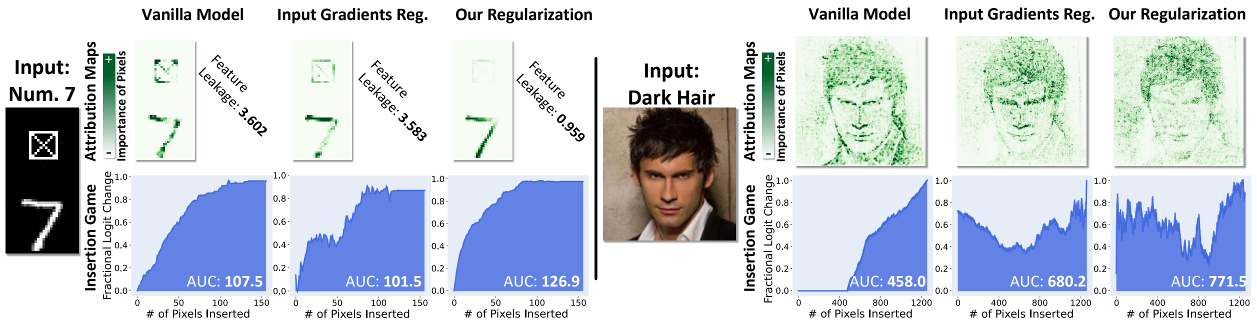
Trustworthy machine learning necessitates meticulous regulation of model reliance on non-robust features. We propose a framework to delineate such features by attributing model predictions to the input. Within our framework, robust feature attributions exhibit a certain consistency, while non-robust feature attributions are susceptible to fluctuations. This feature behavior leads to the identification of correlation between model reliance on non-robust features and smoothness of marginal density of the input samples. Hence, we propose to regularize the gradients of the marginal density w.r.t. the input features for robustness. We also devise an efficient implementation of our regularization to address the potential numerical instability of the underlying optimization process. Moreover, we analytically reveal that, as opposed to our marginal density smoothing, the commonly adopted input gradient regularization smooths conditional or joint density of the input, resulting in limited robustness. Our experiments validate the effectiveness of the proposed method, providing clear evidence of mitigating spurious correlations learned by the model, and addressing the feature leakage problem. We demonstrate that our regularization enables the model to exhibit robustness against perturbations in pixel values, input gradients and density.

1 Introduction

Research on mitigating model reliance on non-robust input features has recently gained increasing attention due to high-stake machine learning applications (Rudin, 2019; Grathwohl et al., 2020; Srinivas & Fleuret, 2021; Dombrowski et al., 2022). In this paper, we advance this direction by introducing a regularization technique that promotes a smooth marginal probability density function of the input to regulate the model's reliance on non-robust features.

To distinguish between robust and non-robust features, we leverage the notion of attributions (Zeiler & Fergus, 2014; Fong & Vedaldi, 2017; Sundararajan et al., 2017). For a model f parameterized by θ, attributions characterize the importance of the i-th feature x_i of the input x for the model prediction by quantifying the output change between $f(x;\theta)$ and $f(x_{[x_i=0]};\theta)$. Since robust input features contribute to model predictions equally well across slight condition variations, their attributions exhibit a certain consistency. On the other hand, non-robust feature attributions fluctuate under such variations. This identifies a correlation between the model's reliance on the non-robust features and the smoothness of the marginal probability density function of the input sample $p_\theta(x)$. For robustness, this offers a possibility of model regularization using the gradients of the marginal density with respect to the input $\nabla_x p_\theta(x)$. Regularizing $\nabla_x p_\theta(x)$ can encourage the model to prioritize the use of robust features and regulate its reliance on non-robust features. However, this can also lead to numerical instability in model optimization. To address that, we further introduce a stable and efficient implementation to estimate the gradient of the marginal density.

We also investigate input gradient norm regularization (Drucker & LeCun, 1992; Ross et al., 2017; Ross & Doshi-Velez, 2018) and reveal that input gradients can be interpreted as input gradients of the log-conditional density $\nabla_x log\ p_\theta(x|y)$ or log-joint density $\nabla_x log\ p_\theta(x,y)$. Input gradient regularization mitigates the model's reliance on non-robust features specific to the class label $y = i$, leading to model blindness to class-specific non-robust features where $y \neq i$. In contrast, our regularization encourages smoothness of the marginal density $p_\theta(x)$ without imposing unintended constraints, providing a comprehensive regularization of the non-robust features. In Figure 1, we employ attribution maps (Shrikumar et al., 2017) and insertion game scores (Petsiuk et al., 2018) to compare the robustness of vanilla models, input gradient regularized models and the models trained with our regularization on BlockMNIST (Shah et al., 2021) and CelebA (Liu et al., 2015) datasets. As the representative examples show, our regularization suppresses both feature leakage and feature spurious correlation, leading to better explainability.



(a) Robustness comparison on BlockMNIST

(b) Robustness comparison on CelebA

Figure 1: Attribution maps (Shrikumar et al., 2017) and insertion game scores (Petsiuk et al., 2018) for representative samples from (a) BlockMNIST and (b) CelebA datasets. As compared to input gradient regularization, our regularization leads to lower feature leakage while also achieving higher AUC for the insertion game.

The effectiveness of our approach is extensively established through a series of experiments. First, using BlockMNIST dataset (Shah et al., 2021), we demonstrate that the model trained with our regularization considerably mitigates the problem of feature leakage. This problem occurs when a model wrongly attributes importance to irrelevant but persistent features in the data to make its predictions, e.g., the *null* block (⊠) in Figure 1. Addressing feature leakage suppresses spurious correlation between the input features and model predictions (Sagawa et al., 2019; Adebayo et al., 2020). Our experiments on CelebA and Waterbirds datasets (Wah et al., 2011; Liu et al., 2015; Sagawa et al., 2019) affirm a significant reduction of this spurious correlation with the proposed regularization. We also establish the robustness of models trained with our regularization against perturbations from adversarial attacks (Goodfellow et al., 2015; Madry et al., 2018), input pixels (Samek et al., 2017) and input density, demonstrating the broad applicability of our approach. Our main contributions are summarized as follows.

1. We identify robust and non-robust features by leveraging attributions, and establish the correlation between model reliance on non-robust features and the smoothness of data marginal density.
2. We propose an efficient technique for regularizing the gradients of log data density, also addressing the numerical instability of the underlying optimization problem.
3. Through extensive experiments, we demonstrate the effectiveness of the proposed regularization. We also establish that our approach exhibits general robustness against perturbations.

2 Related Work

Regularization for Interpretability Robustness: Current deep neural networks lack interpretability in their decision-making process, which is exacerbated by their reliance on non-robust input features. It was demonstrated in (Dombrowski et al., 2019; Zhang et al., 2020) that gradients of the input can be manipulated, which compromises the reliability of predictions. Shah et al. (2021) identified that a standard trained model may rely on non-informative input features, while Adebayo et al. (2020; 2022) showed that deep networks are prone to relying on spurious correlated features. To address that, several regularization techniques are proposed to improve the model's interpretability. In (Schramowski et al., 2020; Erion et al., 2021), the authors incorporated prior knowledge into the model training process to regularize the model behavior. Dombrowski et al. (2019; 2022) found that regularizing the input Hessian using SoftPlus activations or weight decay can boost resilience against manipulated inputs. In addition, Grathwohl et al. (2020) trained a joint energy-based model as a discriminative model for improved robustness. Srinivas and Fleurent (Srinivas & Fleuret, 2021) enhanced the interpretability of the model by improving the alignment between the implicit density model and the data distribution.

Regularization for Adversarial Robustness: In addition to the other sources of prediction unreliability, adversarial attacks can manipulate model outputs with imperceptible perturbations to inputs (Goodfellow et al., 2015; Madry et al., 2018). To address this, adversarial training through data augmentation with adversarial samples is widely employed (Carlini & Wagner, 2017; Madry et al., 2018; Shafahi et al., 2019). Certified adversarial robustness through regularizations (Ross & Doshi-Velez, 2018; Simon-Gabriel et al., 2019; Anil et al., 2019) is another branch of methods to defend against the adversarial perturbations. Inspired by the classic double backpropagation (Drucker & LeCun, 1992), Ross & Doshi-Velez (2018) regularized the input gradient norm for adversarial robustness. In their method, the Frobenius norm of the Jacobian w.r.t. input is included in the training loss. Etmann (2019) further explored the different variants of double backpropagation regularizations for various real-world scenarios. Moosavi-Dezfooli et al. (2019) also proposed regularization to encourage a low curvature for adversarial robustness.

Attribution Methods: Feature attribution methods are used to estimate the importance of input features for a model's prediction and can be categorized as either back-propagation-based or perturbation-based techniques. The former (Simonyan et al., 2014; Sundararajan et al., 2017; Erion et al., 2021) estimate the attribution scores by computing the gradients or integrated gradients with respect to the input features in the backward propagation process. Perturbation-based methods (Zeiler & Fergus, 2014; Fong & Vedaldi, 2017; Zintgraf et al., 2017) calculate the attribution scores by repeatedly perturbing the input features and analyzing the resulting effects on the model prediction. These methods are also extended for evaluating the reliability of the computed attributions (Samek et al., 2017; Petsiuk et al., 2018; Ancona et al., 2018). Our work leverages the attribution framework to distinguish between robust and non-robust features, which enables us to systematically analyze the robustness of input features.

3 Robust and Non-Robust Features by Attributions

We first provide a framework for distinguishing between robust and non-robust features by analyzing feature attributions. Herein, attribution inconsistencies among the features with distinct degrees of robustness identifies a correlation between the model's reliance on non-robust features and the smoothness of output logits.

Let us consider an input sample $x \in \mathbb{R}^n$ with label $y \in \mathbb{R}^c$ from a dataset $\mathcal{D}$, and a classifier $f : \mathbb{R}^n \rightarrow \mathbb{R}^c$ parameterized by θ. We denote robust and non-robust features within the input x as $x_{rob}, x_{nrob} \subseteq x$. Consider an attribution method $\phi : \mathbb{R}^c \rightarrow \mathbb{R}^n$ attributing model predictions to input features by estimating their importance, resulting in an attribution map $M = \phi(f(x;\theta))$. Inspired by the success of attribution methods in model explanation, we identify robust and non-robust features by leveraging their attributions.

Without loss of generality, we assume that attributions M of the features can be estimated by calculating the change in output logits when these features are removed from the input, following perturbation-based methods (Zeiler & Fergus, 2014; Ribeiro et al., 2016): $M_{x_{rob}} = f(x;\theta) - f(x_{[x_{rob}=0]};\theta)$ and $M_{x_{nrob}} = f(x;\theta) - f(x_{[x_{nrob}=0]};\theta)$. For the ease of understanding, we use $f(x_{nrob};\theta)$ and $f(x_{rob};\theta)$ to represent attributions M_{rob} and M_{nrob} in the text to follow. We define robust features within the attribution framework as follows.

Definition 1. *A feature x_{feat} shared among different input instances under its domain $\Delta_{x_{feat}}$ is robust if, for a randomly chosen class $y = i$, its attribution $M_{x_{feat}}$ is bounded by a small constant h under a metric $c(\cdot)$, i.e., $c(f(x_{feat};\theta) - f(\tilde{x}_{feat};\theta)) \leq h : \tilde{x}_{feat} \in \Delta_{x_{feat}}$.*

Existence of a robust features is expected to contribute consistently to the model's prediction across different input samples. Non-robust features, on the other hand, are those that contribute to the prediction score inconsistently or only under specific conditions. Our focus here is on distinguishing between robust and non-robust features, without specifying a particular metric. Definition 1 emphasizes on attribution consistency for robust features rather than attribution positivity, thereby allowing for robust features that can also make a negative contribution to the model's prediction. Building further upon the above definition, we make the following remark.

Remark 1. *Robust features are largely **condition-invariant** in that they retain similar attributions despite slight changes to the input. In contrast, non-robust features are **condition-specific** in that their attributions either vary drastically with slightly varying input conditions, or behave robustly only under specific conditions.*

In classification, robust features exhibit stable behavior across the input space, which is observable through consistent output logits $f(x_{rob};\theta)$ regardless of the class y or specific input instance. In contrast, non-robust features rely on specific conditions to exhibit a particular behavior tailored to a specific class $y = i$ or the input instance. Robust modeling aims for a stronger reliance of prediction on robust features. Due to the consistency of output logits $f(x;\theta)$ for robust features, a smooth $f(x;\theta)$ ensures a positive step towards this objective.

4 Regularizing the Gradients of Marginal Density

Here, we establish the correlation between model robustness and the gradients of input marginal density. Then, a robust regularization is proposed for regulating model reliance on non-robust features by smoothing the marginal density.

We commence our analytical analysis with probability density, following Bridle (1990). Given a class $y = i$, a joint probability density function over the input with the output logit $f_i(x;\theta)$ is defined as

$$p_\theta(x, y = i) = e^{f_i(x;\theta)} / Z_\theta, \tag{1}$$

where the constant $Z_\theta = \int e^{f_i(x;\theta)}dx$ is the partition function. Z_θ normalizes the input x to a probability density by integrating over all possible input points x in the input space via the model f. By applying Bayes' rule, we eliminate the condition $y = i$, resulting in the marginal density being defined solely on the input x: $p_\theta(x) = p_\theta(y = i, x)/p_\theta(y = i|x)$. The conditional density function $p_\theta(y = i|x)$ can be further defined as

$$p_\theta(y = i|x) = e^{f_i(x;\theta)}/Z_{f(x;\theta)}, \tag{2}$$

where $Z_{f(x;\theta)} = \sum_{i=1}^{C} e^{f_i(x;\theta)}$ is the partition function for the output logits $f_i(x;\theta)$ defined on all the C classes. For the simplicity of notation, we use $Z_{f(x)}$ to represent $Z_{f(x;\theta)}$ in the subsequent discussion. Exploiting the symmetry property of the joint density defined in Equation 1, i.e., $p_\theta(x, y = i) = p_\theta(y = i, x)$, the marginal density $p_\theta(x)$ can be expressed as

$$p_\theta(x) = \frac{e^{f_i(x;\theta)}/Z_\theta}{e^{f_i(x;\theta)}/Z_{f(x)}} = \frac{Z_{f(x)}}{Z_\theta}, \tag{3}$$

As Remark 1 implies, smoothness of output logits encourages the use of robust features by the model. Hence, we consider the marginal density $p_\theta(x)$ defined on output logits $f(x;\theta)$ across the input space. Promoting a small gradient of the marginal density with respect to the input x, denoted as $\nabla_x p_\theta(x)$, contributes to the smoothness of output logits. Thus, a positive correlation can be established between the use of robust features and the smoothness of the probability marginal density $p_\theta(x)$. In particular, the smooth output logits of robust features across input samples suggest that these features will have small gradients of the density $p_\theta(x)$ with respect to the input values. On the other hand, non-robust features with fluctuating output logits and resulting large gradients of the density will be suppressed during the training process. Therefore, we can conclude with the following remark.

Remark 2. *Model reliance on non-robust features x_{nrob} can be regulated by regularizing the gradients of the marginal density $p_\theta(x)$ with respect to x, and this regularization is achieved through optimizing the model parameters θ^*.*

In the light of Remark 2, we proposed a regularization term for minimizing the gradients of marginal density. However, computing the gradients of the marginal density $\nabla_x p_\theta(x)$ is not feasible because the partition function defined on the entire input space is intractable. We instead compute the gradients with respect to the log-density to avoid the estimation of the intractable Z_θ. This is possible because Z_θ only depends on the model parameter θ and not the input x. Specifically, we compute the gradients with respect to the log-density as $\nabla_x log\, p_\theta(x) = \nabla_x log\ Z_{f(x)} - \nabla_x log\ Z_\theta = \nabla_x log\ Z_{f(x)}$. Expanding the partition function $Z_{f(x)}$, we obtain $\nabla_x log\ p_\theta(x) = \sum_{i=1}^{C} \nabla_x e^{f_i(x;\theta)} / \sum_{i=1}^{C} e^{f_i(x;\theta)}$. The p-norm of this gradient is then computed as the regularization term. In the optimization process, the goal is to find the optimal parameter θ^* by minimizing the loss ℓ as

$$\theta^* = \min_\theta \ell(f(x;\theta), y) + \lambda || \frac{\sum_{i=1}^{C} \nabla_x e^{f_i(x;\theta)}}{\sum_{i=1}^{C} e^{f_i(x;\theta)}} ||_p, \tag{4}$$

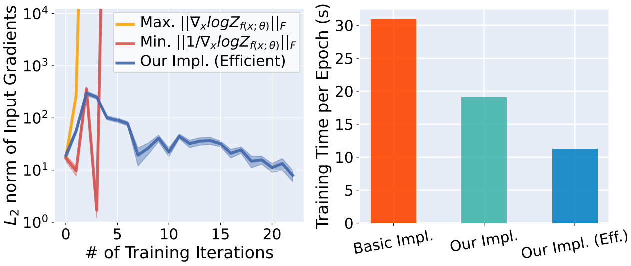
where λ indicates the magnitude of the coefficient for controlling the strength of the regularization.

To regulate model reliance on non-robust features, our regularization encourages the smoothness of marginal density by regularizing its gradients. Since the output logit change in the log-marginal density $log\, p_\theta(x) = log\ Z_{f(x)} - log\ Z_\theta$, and Z_θ is independent of the input x, we can only focus on the first term $Z_{f(x)} = \sum_{i=1}^{C} e^{f_i(x;\theta)}$. Recall, our definition of robust features. Assuming the robust input feature x_{rob} exists in a random input x, the corresponding output logit $f_i(x_{rob})$ will consistently attribute to the model prediction. This property of x_{rob} leads to the smoothness of output change $\sum_{i=1}^{C} e^{f_i(x_{rob};\theta)}$ across different input samples and class labels.

In contrast, non-robust input features x_{nrob} show relatively high attributions for the output logit $f_i(x_{nrob})$ for given a class $y = i$. However, they cannot maintain consistency in attributions across different inputs or labels, leading to the fluctuation in the output change $\sum_{i=1}^{C} e^{f_i(x_{nrob};\theta)}$. It is demonstrated that the magnitude of gradients for input features in the marginal density $p_\theta(x)$ reflects the model's sensitivity to those features. We leverage this relation to mitigate model reliance on the non-robust features by smoothing the marginal density of the input samples.

5 Stable and Efficient Implementation for Regularization

From the implementation perspective, the gradient computation of marginal density involves multiple exponential operations in both the numerator and denominator of Equation 4 which can introduce numerical instability in the optimization process, leading to gradient vanishing and explosion problems. Such issues can potentially hinder the application of our regularization to large non-linear models or wide-distribution data. For instance, batch normalization (BN) (Ioffe & Szegedy, 2015) layer solving internal covariate shifts with learnable scaling and shifting parameters can amplify the errors caused by the exponential operations during the backpropagation, leading to gradient vanishing or explosion. Figure 2(a) shows the L_2 norm change of input gradients as the number of training iterations increases for ResNet-34. Two implementations, corresponding to yellow and red lines, for minimizing the gradients of the marginal density cause gradient vanishing and explosion in the training process. Therefore, it is crucial to address these through careful implementation to ensure the feasibility of our regularization.



(a) Numurical Stability. **(b) Training Efficiency.**

Figure 2: Numerical stability and training efficiency comparison of ResNet-34 trained on CIFAR-10 dataset.

To address this challenge, we transform the computation from the summation of exponential operations to softmax. By subtracting a constant value η from the logits before exponentiation, softmax can prevent numerical overflow during the exponential operations (i.e., $e^{f_i(x;\theta)}/\sum_j e^{f_j(x;\theta)} = e^{f_i(x;\theta)-\eta}/\sum_j e^{f_j(x;\theta)-\eta}$). Thus, we incorporate the softmax function into the computation of log density gradient $\nabla_x logz_{f(x)}$ as

$$\begin{aligned} \nabla_x logz_{f(x)} &= \nabla_x f_i(x;\theta) - \nabla_x(loge^{f_i(x;\theta)} - logz_{f(x)}) \\ &= \nabla_x f_i(x;\theta) - \nabla_x(log(f_i(x;\theta)/z_{f(x)})), \end{aligned} \tag{5}$$

where $f_i(x)$ indicates a logit of a random i-th class, and $z_{f(x)}$ equals $\sum_j e^{f_j(x;\theta)}$. Thus, the gradients with respect to the log-marginal density $p_\theta(x)$ can be replaced by computing the difference between the gradient $\nabla_x f_i(x;\theta)$ and the gradient of a log-softmax output $\nabla_x log(e^{f_i(x;\theta)}/Z_{f(x)})$.

Our technique improves upon the common approach (Bridle, 1989; Blanchard et al., 2021) to achieve numerical stability in log exponential sum calculations, which typically employs the formula $log(\sum_{i\in\{1,\dots,n\}} e^{x_i}) = \eta + log(\sum_{i\in\{1,\dots,n\}} e^{x_i-\eta})$, with η being the maximum value of inputs $\{x_1,\dots,x_n\}$. We address the numerical instability by employing softmax, avoiding computationally expensive comparisons of the maximum gradient values. Specifically, the basic stable implementation involves finding the maximum gradient within the gradients of various classes, leading to a time complexity of $O(n)$, where n represents the number of classes. In contrast, our method eliminates numerical instabilities associated with a randomly selected class, eliminating the need for maximum value comparisons and leading to a more favorable time complexity of $O(1)$.

While our solution helps avoid numerical instability, it requires twice the gradient computations as compared to the sole calculation of density gradients. Hence, we further propose an efficient mechanism for estimating the difference in the gradients. Specifically, we compute the gradients of the difference between two outputs to approximate the difference between the two gradients of outputs using Taylor series as

$$\nabla_x f_i(x;\theta) - \nabla_x log(e^{f_i(x;\theta)}/Z_{f(x)}) \approx \nabla_x(f_i(x;\theta) - log(e^{f_i(x;\theta)}/Z_{f(x)})). \tag{6}$$

The proof of Equation 6 is provided in Appendix A.1. As such, the proposed approach enables stable and efficient model optimization. The blue curve in Figure 2(a) shows that our efficient implementation can effectively avoid numerical instability in the gradient computation. Figure 2(b) provides a comparison of training efficiency between the basic implementation with numerical stability and our two implementations. It is evident that our method substantially improves the per-epoch training time.

6 Limited Robustness in Input Gradient Regularization

Input gradient regularization (InputGrad Reg.) (LeCun et al., 1998; Ross & Doshi-Velez, 2018) computes the Frobenius norm of input gradients $||\nabla_x f_i(x;\theta)||_F$ for a given class label $y = i$, which is a baseline robust regularization for model optimization. Existing works (LeCun et al., 1998; Ross et al., 2017; Ross & Doshi-Velez, 2018) explain InputGrad Reg. as the prediction stability to gain robustness against perturbations. In this section, we reveal that the input gradient norm potentially regularizes the gradients of implicit data density. Moreover, we provide an understanding of how InputGrad Reg. encourages robustness as well as its limitations.

Suppose all the classes have equal probability, i.e., $p_\theta(y = i) = 1/C$. We can express the class-conditional density $p_\theta(x|y)$ by using Bayes's rule as

$$p_\theta(x|y = i) = \frac{p_\theta(x, y = i)}{p_\theta(y = i)} = \frac{e^{f_i(x;\theta)}}{Z_\theta/C}. \tag{7}$$

Now, we compute the gradients of the log density defined in Equation 7 with respect to the input x as

$$\nabla_x log\, p_\theta(x, y = i) = \nabla_x log\, p_\theta(x|y = i) = \nabla_x f_i(x; \theta). \tag{8}$$

Thus, Equation 8 demonstrates that the input gradients can be interpreted as the gradients of either the log joint density or the log conditional density with respect to the input x.

To simplify the expression, we consider the exponential of the output logits, i.e., $e^{f_i(x;\theta)}$. This formulation highlights that InputGrad Reg. encourages consistent attributions of input features for the model's predictions. However, it is important to note that the condition $y = i$ in this context refers to the label of the input x in InputGrad Reg. The regularization is assumed under the specific condition $y = i$, which limits its effectiveness in resolving inconsistencies when predicting a different class $y = j$, where $j \neq i$.

Although input features consistently contribute to the model's prediction under a given class, InputGrad Reg. fails to consider inconsistent attributions of these features across different classes, thereby allowing model non-robust behavior to exist. Consequently, a model trained with input gradient regularization may exhibit spurious robustness relying on specific conditions.

Remark 3. *Input gradient regularization smooths the joint and conditional density of the input x under a specific label $y = i$, compromising its ability to resist the class-specific non-robust features.*

In contrast to the existing contributions (Ross & Doshi-Velez, 2018; Srinivas & Fleuret, 2021) that highlight the reasons of InputGrad Reg. efficacy, we reveal a weakness of this technique. Unlike regularizing gradients based on joint or conditional densities, our approach allows for more effective regularization without imposing the condition $y = i$. Our method focuses on regularizing the gradients of the marginal density $\nabla_x p_\theta(x)$, thereby smoothing the output logits across the input samples.

7 Experiments

We perform extensive experiments to validate the efficacy of our regularization and the newly established correlation between the smooth marginal density and model reliance on non-robust features. We present measurement results for two applications to demonstrate the robustness of models trained with our regularization. Firstly, we evaluate our method on the BlockMNIST dataset (Shah et al., 2021) and observe significant suppression of the model's reliance on non-robust features. This dataset allows for a controlled and straightforward robustness assessment, allowing a faithful analysis of different methods in terms of their robustness. Moreover, we extend our analysis to natural images and investigate the mitigation of spurious correlations (Sagawa et al., 2019). Our regularization technique effectively mitigates spurious correlations learned by the models on both CelebA and WaterBrid datasets (Liu et al., 2015; Wah et al., 2011; Zhou et al., 2017; Sagawa et al., 2019), demonstrating its effectiveness in handling real-world scenarios. In addition to these specific applications, more quantitative tests on the robustness against perturbations in input pixels, gradients and density (Samek et al., 2017; Srinivas & Fleuret, 2021) demonstrate the desirability of our regularization. More details about the datasets and the models used in our experiments can be found in Appendix A.8.

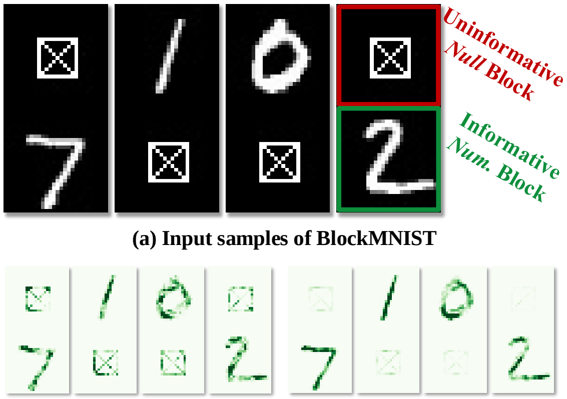
7.1 Efficacy against Feature Leakage and Adversarial Attacks

In Shah et al. (2021); Adebayo et al. (2020), it is demonstrated that deep models also end up assigning importance to irrelevant input features. The issue is mitigated in adversarially trained models due to their better interpretabilty. Here, we extend the analysis of (Shah et al., 2021) on the topic. Shah et al. (2021) used BlockMNIST in their experiments, which is a synthetic dataset extended from MNIST (LeCun et al., 1998). To each MNIST sample, BlockMNIST attaches a *null* block (an irrelevant pattern) randomly at the top or bottom of the image. Representative samples of BlockMNIST are shown in Figure 3(a). Shah et al. (2021) observed that the explaining tool InputGrad (Simonyan et al., 2014) attributes importance to both the informative number block and the uninformative null block in the standard trained model. This phenomenon is termed as *feature leakage* by the authors.

Reproducibility and Quantitative Measurement of Feature Leakage. Owing to the unreliability of InputGrad caused by model saturation (Shrikumar et al., 2017), we employ Integrated Gradients (IG) (Sundararajan et al., 2017), an axiomatic model explanation tool, to re-investigate the feature leakage phenomenon. In Figure 3(b)-(c), we show the attributions computed by IG for both the standard and adversarially trained models given the input samples in Figure 3(a). Originally, Shah et al. (2021) showed that the informative pattern of the number could be leaked to the null block regions by observing attributions computed by Input Gradients. In contrast, IG can reproduce the leakage problem but confines the attributions within the null block. We employ explanations provided by IG to serve as the basis for our subsequent analysis, given its higher reliability compared to Input Gradients. Feature leakage, an important phenomenon in the context of model robustness, lacks a quantitative metric in the current literature to quantify its extent. We use attributions to define the metric $M_{leakage}$ to address the gap. Mathematically,

$$M_{leakage} = \mathop{\mathbb{E}}_{x_{nrob}\sim D} ||x_{nrob} \times \int_{\alpha=0}^{1} \frac{\partial f(\alpha \cdot x_{nrob};\theta))}{\partial x_{nrob}} \mathrm{d}\alpha||_F, \tag{9}$$

where α indicates the step from the absence to the presence of the input features, and x_{nrob} represents the non-robust features in the null block. Since the attributions of x_{nrob} for the model's prediction are expected to be zero, estimating the attribution norm can be used to measure the leakage of features.



(a) Input samples of BlockMNIST

(b) Integrated Gradients on standard trained model

(c) Integrated Gradients on adversarially trained model

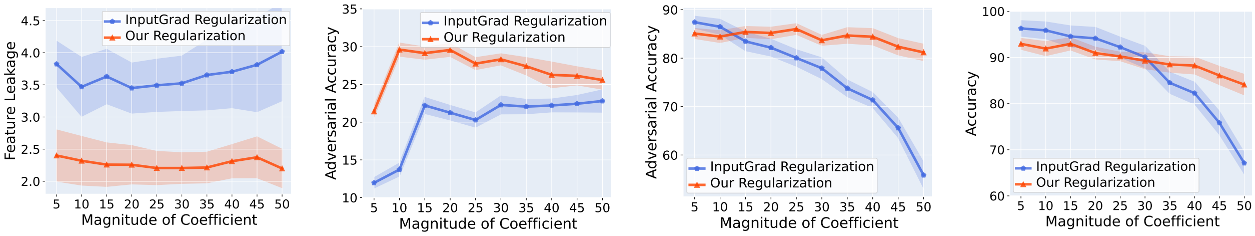
Figure 3: BlockMNIST samples and feature leakage problem. **(a)** BlockMNIST randomly appends a null block at the top or bottom of MNIST samples. **(b&c)** Attribution maps are calculated by IG on the standard and adversarially trained models.

Robustness against Feature Leakage. Table 1 presents the experimental results on the BlockMNIST dataset. We compare our method with other robust regularizations and techniques including InputGrad (Ross & Doshi-Velez, 2018), SoftPlus activations (Dombrowski et al., 2019) and Hessian (Dombrowski et al., 2022). InputGrad and Hessian regularize the first-order and second-order gradients w.r.t. the input. Models trained with SoftPlus activations and Hessian regularization fail to suppress the leakage problem, which indicates that feature leakage is not caused by the geometry of the model output manifold or high curvature (Dombrowski et al., 2019; Zhang et al., 2020). InputGrad regularization demonstrates robustness against both L_2 and L_∞ adversarial attacks, yet it still fails to address the leakage problem. The result aligns well with Remark 3 which highlights the allowance of non-robust features across different classes in InputGrad regularization. These results further reveal that adversarial robustness is not a sufficient condition for suppressing feature leakage. Our method demonstrates a considerable improvement over other techniques for feature leakage, while maintaining superiority in adversarial robustness. In our experiments, we take the L_2 norm for all compared regularization terms for a fair comparison. It is worth noting that our use feature of attribution enables us to select different normalizations for feature sparsity in various data distributions. By exploring optimal normalization, our regularization enables a more favorable trade-off for model robustness. Additional experiments with different norms are provided in Appendix A.2.

Table 1: Experimental results on BlockMNIST. Standard accuracy, feature leakage, and adversarial accuracy under L_2 and L_∞ attacks are reported. ST: Standard Training, AT: Adversarial Training.

Method	Feature Leakage ↓	PGD-20 (L_2) ↑	PGD-20 (L_∞) ↑	Accuracy ↑
AT-FGSM	3.324	87.48	0.00	99.02
AT-PGD	2.313	92.75	28.05	98.97
ST	3.657	73.57	0.00	**99.12**
ST + SoftPlus Activations	3.533	67.95	0.02	98.52
ST + Hessian Reg.	4.258	80.06	0.00	98.48
ST + InputGrad Reg.	3.461	83.46	21.14	94.56
ST + Our Reg.	**2.259**	**85.41**	**29.36**	93.05

Feature Leakage in Adversarially Trained Model. The FGSM adversarially trained model (Goodfellow et al., 2015) augments the training samples by adversarial examples $x + \epsilon \cdot sign(\nabla_x f_i(x;\theta))$. Notably minimizing the loss of the perturbed input $x + \epsilon \cdot sign(\nabla_x f_i(x;\theta))$ is similar to the InputGrad regularization. Thus, training with FGSM is still limited in its ability to suppress the leakage problem. On the other hand, PGD attack (Madry et al., 2018) weakens the effect of the given condition $y = i$ by iteratively searching for the perturbations from a random starting point, leading to a substantial enhancement in suppressing feature leakage.



(a) Feature Leakage.

(b) L_∞ Attack.

(c) L_2 Attack.

(d) Accuracy.

Figure 4: Performance comparison between our method and InputGrad regularization under varying regularization coefficient for (a) Feature leakage, (b)-(c) Adversarial Accuracy under L_∞ and L_2 PGD-20 attacks, and (d) Accuracy.

Magnitude of Coefficient for Regularization. Figure 4(a)-(d) present a comparison of results for feature leakage and adversarial accuracy under L_∞ and L_2 PGD attacks, as well as the standard accuracy across varying magnitudes of the coefficient for the regularization strength. The results affirm that our method effectively regulates feature leakage by imposing a penalty on non-robust features. Moreover, our regularization enables the model to defend against both L_2 and L_∞ attacks while maintaining high accuracy, showing an outstanding trade-off across four criteria. More adversarial robustness comparisons on CIFAR dataset (Krizhevsky et al., 2009) are also reported in Appendix A.3.

7.2 Efficacy for Spurious Correlation

Recent research has highlighted the susceptibility of neural models in learning spurious correlations that enhance performance on training samples but fail to generalize (Buolamwini & Gebru, 2018; Sagawa et al., 2019; Adebayo et al., 2020). For instance, in CelebA dataset (Liu et al., 2015), which commonly consists of samples containing female celebrities with blond hair and male celebrities with dark hair, models heavily rely on the spurious correlated *gender* feature to predict the target *hair color* (Hashimoto et al., 2018; Sagawa et al., 2019). Consequently, accuracy tends to be lower for samples containing male celebrities with blond hair, see Figure 5.

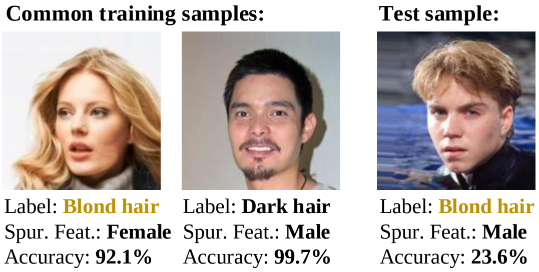


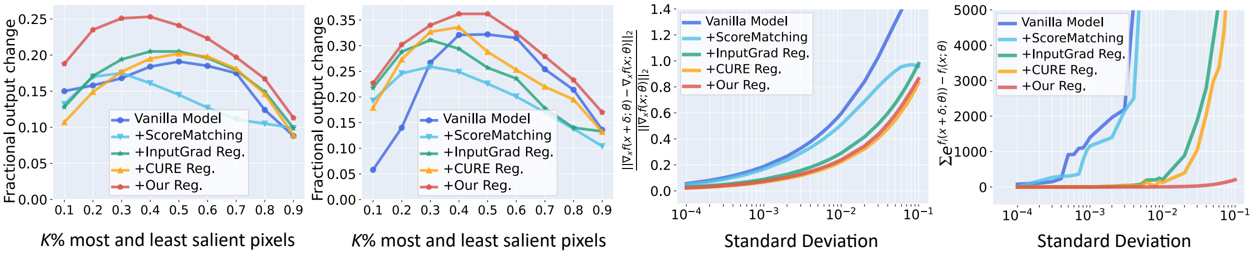
Figure 5: Spurious correlation in ResNet-34 trained on CelebA dataset. The model fails to classify the male celebrity (spurious correlated feature) with blond hair (target label).

To mitigate spurious correlations, distributional robust optimization (DRO) techniques have been proposed to re-weight the training loss of input samples from different groups (Hu et al., 2018; Sagawa et al., 2019). In our regularization, an additional penalty is imposed to penalize the model's reliance on these spurious correlated features because of their inconsistent attributions. This encourages the use of robust features while suppressing the model's reliance on the spurious correlated features. Although quantitatively evaluating the model's robustness on natural images remains challenging, the attribution map (Sundararajan et al., 2017) and the insertion game (Petsiuk et al., 2018) demonstrate the effectiveness of our regularization in suppressing the use of the spurious correlated *gender* feature and promoting the use of the robust *hair* feature, as shown in Figure 1(b). This superiority leads to performance improvements on worst-case samples in the model trained with our regularization.

Table 2 presents comparison of accuracy on worst-group samples and overall samples from the CelebA and Waterbirds datasets (Liu et al., 2015; Sagawa et al., 2019). The Waterbirds dataset consists of synthetic bird images from CUB-200-2011 (Wah et al., 2011) and Places (Zhou et al., 2017) datasets, incorporating spurious background features, such as land and water scenes, to confuse true labels of bird categories. We compare the proposed regularization method with InputGrad and Score-Matching regularizations (Srinivas & Fleuret, 2021), as well as Group DRO (Sagawa et al., 2019). Score-Matching regularization is proposed to enhance the interpretability of the model by improving the alignment of implicit density models. Experimental results clearly demonstrate the effectiveness of our method in enhancing worst-group accuracy while maintaining overall sample accuracy. Furthermore, the performance gains can be further enhanced by incorporating our regularization technique into Group DRO. This enhancement highlights the applicability and effectiveness of our regularization technique in real-world scenario applications. More results of attribution maps and corresponding insertion games on both CelebA and Waterbirds datasets are reported in Appendix A.7. Moreover, out-of-distribution detection Hendrycks & Gimpel (2016) is also performed on CIFAR (Krizhevsky et al., 2009) and SVHN (Netzer et al., 2011) in Appendix A.4.

Table 2: Worst-group accuracy and overall accuracy comparisons between Vanilla ResNet-34, group DRO, Score-Matching, InputGrad and our regularization on CelebA and Waterbirds datasets.

Method	Worst-Group Accuracy (%)		Overall Accuracy (%)	
	CelebA	Waterbirds	CelebA	Waterbirds
Vanilla Model	49.90 ± 8.69	62.90 ± 0.10	94.90 ± 0.39	87.70 ± 0.08
Group DRO	59.44 ± 5.98	63.60 ± 0.17	**94.96** ± 0.21	87.60 ± 0.05
Score-Matching Reg.	59.78 ± 7.56	58.19 ± 1.55	93.46 ± 0.71	85.64 ± 0.38
InputGrad Reg.	82.66 ± 3.63	58.18 ± 1.22	92.12 ± 2.18	85.50 ± 0.27
Our Reg.	**85.62** ± 5.36	63.78 ± 2.83	92.30 ± 1.38	86.48 ± 0.38
Group DRO + Our Reg.	82.98 ± 4.69	**73.82** ± 2.27	93.62 ± 0.74	**90.52** ± 0.17



(a) Pixel Perturbation on CIFAR-10 and CIFAR-100.

(b) Robustness of Gradients and Density.

Figure 6: Robustness comparison. **(a)** Pixel perturbation experimental results on ResNet-18 trained on CIFAR-10 and CIFAR-100. Higher curves indicate better results. **(b)** Robustness of relative gradients and absolute density on CIFAR-10. Lower curves indicate better results.

7.3 Efficacy for Perturbation in Pixels, Gradients and Density

In this part, we employ pixel perturbation (Samek et al., 2017; Yang et al., 2023) to quantitatively compare the robustness of different models following Srinivas & Fleuret (2021) who iteratively removed the most and least important input pixels identified by attribution maps for model robustness evaluation (Zeiler & Fergus, 2014; Sundararajan et al., 2017). Robust models are expected to exhibit increased sensitivity when removing the most important pixels and decreased sensitivity when removing the least important ones. We assess the difference in fractional output logit change between the images with the top and bottom $k\%$ most salient pixels using SmoothGrad (Smilkov et al., 2017) on ResNet-18 (He et al., 2016) trained on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), as depicted in Figure 6(a). It can be observed that the model trained with our regularization significantly outperforms different robust regularizations including Score-Matching, InputGrad and CURE regularizations (Moosavi-Dezfooli et al., 2019).

We also test the robustness of the relative input gradients $||\nabla_x f(x+\delta;\theta) - \nabla_x f(x;\theta)||_2 / ||\nabla_x f(x;\theta)||_2$ and the absolute density through $\sum_{i=1}^{C} e^{f_i(x+\delta;\theta) - f_i(x;\theta)}$ on the input with Gaussian noise δ in increasing standard deviation on CIFAR-10 dataset, as shown in Figure 6(b). We can observe that our regularization can achieve competitive robustness for the relative gradient in comparison with regularizing the hessian norm in CURE. Moreover, our regularization naturally leads to density robustness, which is associated with a strong generative ability of models (Srinivas & Fleuret, 2021). More robustness tests on CIFAR-100 are provided in Appendix A.5. Appendix A.6 further supports visualization examples calculated by activation optimization. These results affirm that our regularization improves both discriminative and generative abilities of models.

8 Conclusion

In this paper, we define robust and non-robust features from a feature attribution perspective, and establish a correlation between the smoothness of input marginal density and model reliance on non-robust features. This connection motivates us to propose a regularization that targets the gradients of the marginal density, aiming to regulate the reliance on non-robust features. Extensive experiments demonstrate the effectiveness of our regularization in boosting model robustness across different applications. It is essential to note that our approach does not advocate for the complete removal of model reliance on non-robust features, but instead seeks to achieve a balance between model performance and robustness through appropriate regularization strength.

Societal Impact

Accurate attribution of features in a trained model helps reduce bias and inaccuracy by enabling a deeper understanding of how the model arrives at its predictions. By identifying the most influential features, we can assess their potential biases and correct them during model training. This process allows us to prioritize relevant information while reducing the impact of biased or irrelevant features, leading to fairer and more accurate predictions. Additionally, accurate feature attribution aids in interpreting the model's decisions, making it easier to detect and address potential biases, improve transparency, and ensure the model's robustness against adversarial attacks or data drift.

Reproducibility Statement

The implementation code for our regularization method is provided in the supplementary material. Furthermore, detailed information regarding the experimental setup, including datasets, models, the experimental platform, and corresponding hyperparameter choices, can be found in Appendix A.8.

References

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *International Conference on Learning Representations, ICLR*, 2022.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations, ICLR*, 2018.

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning, ICML*, 2019.

Pierre Blanchard, Desmond J Higham, and Nicholas J Higham. Accurately computing the log-sum-exp and softmax functions. *IMA Journal of Numerical Analysis*, 41(4):2311–2330, 2021.

John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in Neural Information Processing Systems, NeurIPS*, 1989.

John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, Architectures and Applications*, pp. 227–236. Springer, 1990.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM Conference on Fairness, Accountability, and Transparency, FAccT*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy, EuroS&P*, 2017.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

Ann-Kathrin Dombrowski, Christopher J Anders, Klaus-Robert Müller, and Pan Kessel. Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194, 2022.

Harris Drucker and Yann LeCun. Improving generalization performance using double backpropagation. *IEEE Trans. Neural Networks*, 1992.

Gabriel G. Erion, Joseph D. Janizek, Pascal Sturmfels, Scott M. Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 2021.

Christian Etmann. A closer look at double backpropagation. *CoRR*, 2019.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *International Conference on Computer Vision, ICCV*, 2017.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations, ICLR*, 2015.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations, ICLR*, 2020.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning, ICML*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision, European Conference on Computer Vision, ECCV*, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, 2016.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning, ICML*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning, ICML*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *CoRR*, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision, ICCV*, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations, ICLR*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *Advances in Neural Information Processing Systems, NeurIPSW*, 2011.

Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference, BMVC*, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD*, 2016.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI Conference on Artificial Intelligence, AAAI*, 2018.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *International Joint Conference on Artificial Intelligence, IJCAI*, 2017.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations, ICLR*, 2019.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *TNNLS*, 28(11):2660–2673, 2017.

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2020.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? In *Advances in Neural Information Processing Systems, NeurIPS*, 2021.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning, ICML*, 2017.

Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning, ICML*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations, ICLR*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations, ICLR*, 2014.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017.

Suraj Srinivas and François Fleuret. Rethinking the role of gradient-based attribution methods for model interpretability. In *International Conference on Learning Representations, ICLR*, 2021.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning, ICML*, 2017.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

Peiyu Yang, Naveed Akhtar, Zeyi Wen, and Ajmal Mian. Local path integration for attribution. In *AAAI Conference on Artificial Intelligence, AAAI*, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *British Machine Vision Conference, BMVC*, 2016.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision, ECCV*, 2014.

Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable deep learning under fire. In *USENIX Security*, 2020.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2017.

Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *International Conference on Learning Representations, ICLR*, 2017.

A Appendix

Contents

A.1 Proof

In this section, we provide proof of our proposed approach for estimating the gradient difference using the gradients of the output difference.

Proof of Equation 7. Assuming that two functions, f and g, are continuously differentiable with respect to $x \in \mathbb{R}^n$, we can express them using their Taylor series expansions, as

$$f(x) = f(a) + f'(a)(x-a) + o((x-a)^2), \tag{10}$$

and

$$g(x) = g(a) + g'(a)(x-a) + o((x-a)^2). \tag{11}$$

By subtracting Equation 11 from Equation 10, we have

$$f(x) - g(x) = f(a) - g(a) + (f'(a) - g'(a))(x-a) + o((x-a)^2). \tag{12}$$

Next, we compute the gradients of both sides of Equation 12 with respect to x as

$$\nabla_x(f(x) - g(x)) = f'(a) - g'(a) + o((x-a)^3). \tag{13}$$

Then, we can set $x = a$ in Equation 13 as

$$\nabla_a(f(a) - g(a)) \approx f'(a) - g'(a). \tag{14}$$

Since we have assumed that f and g are differentiable, we can estimate the difference between the gradients of the two functions as the gradient of the difference between the functions as

$$\nabla_x(f(x) - g(x)) \approx \nabla_x f(x) - \nabla_x g(x). \tag{15}$$

Since the model f_θ parameterized with θ is assumed as continuously differentiable, we can substitute the model output logit $f_i(x;\theta)$ and log-softmax output logit $log(\frac{e^{f_i(x;\theta)}}{Z_{f(x)}})$ into the functions f and g in Equation 15 as

$$\nabla_x f_i(x;\theta) - \nabla_x log(\frac{e^{f_i(x;\theta)}}{Z_{f(x)}}) \approx \nabla_x(f_i(x;\theta) - log(\frac{e^{f_i(x;\theta)}}{Z_{f(x)}})). \tag{16}$$

Thus, the gradients of the difference between two outputs can be used to approximate the difference between the two gradients of the outputs. □

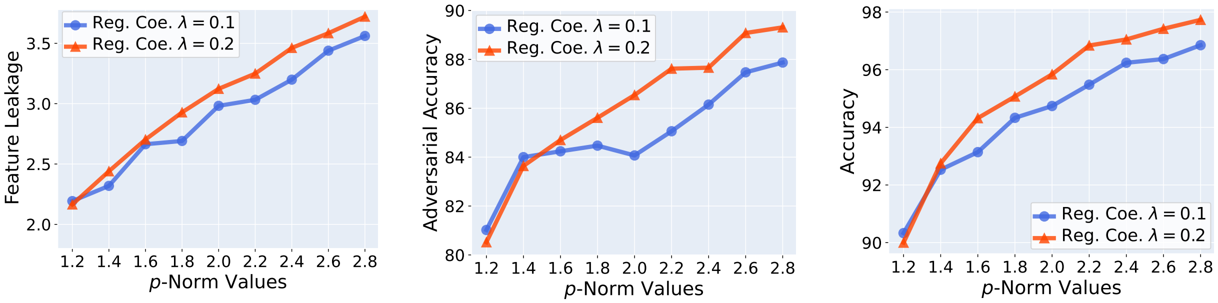
A.2 Norm and Implementation Comparison

In this section, we evaluate the impact of different norms and implementations on our regularizations.

Firstly, we investigate the effect of different p-norm values on our regularization approach. Given a variable $p \in \mathbb{R}$, p-norm of input $x \in \mathbb{R}^n$ is defined as

$$||x||_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}. \tag{17}$$

The L_p norm allows us to measure the magnitude of a vector using different p values. Different p values exhibit different properties. Smaller p values promote sparsity, while larger p values emphasize the maximum value. Hence, selecting an appropriate p value that strikes a balance between these characteristics is crucial when applying the regularization method to models. In Figure 7, we test the effect of p norm values from $p = 1.2$ to $p = 2.8$ on models using our regularization with two regularization coefficients $\lambda = 0.1$ and $\lambda = 0.2$ on BlockMNIST (Shah et al., 2021). In Figure 7(a), lower p values effectively suppress feature leakage, indicating that encouraging sparse features reduces reliance on non-informative features. In Figure 7(b) and Figure 7(c), larger p values lead to enhanced adversarial robustness and higher accuracy, suggesting that models are susceptible to perturbations caused by large gradients. The results reveal that models are easily perturbed from large gradients. The results demonstrate that our regularization enables models to regulate their reliance on non-robust features by adjusting the norm value p and regularization coefficient λ. In our experiments, we employ $p = 2$ for all regularizations to ensure a fair comparison. However, exploring alternative norms in addition to the p norm is expected to further enhance robustness.



(a) Feature Leakage.

(b) Adv. Accuracy.

(c) Accuracy.

Figure 7: Results of feature leakage, adversarial accuracy (%) and standard accuracy (%) under different p-norm values. **(a)** Smaller p-norm values lead to better results in suppressing the feature leakage problem. **Lower** curves indicate **better** results. **(b-c)** Larger p-norm values lead to enhanced adversarial robustness and higher accuracy. **Higher** curves indicate **better** results.

More experimental results are presented to compare the three implementations of our regularization method on the BlockMNIST dataset. Table 3 shows the results for three different models: MLP, VGG11 (Simonyan & Zisserman, 2015), and ResNet-18. Experimental results of models trained with our regularization, including variations with stable and efficient implementations are reported. For MLP models, we observe that both the stable implementation and the efficient implementation of our method achieve outstanding performance compared to the original implementation. This demonstrates the effectiveness of our proposed alternative implementations in enhancing the robustness and performance of MLPs. However, when applied to VGG11 and ResNet-18 models, our regularization compromises their robustness in terms of feature leakage and vulnerability to adversarial perturbations. It can be also observed that ResNet-18, which contains batch normalization (BN) layers (Ioffe & Szegedy, 2015), exhibits additional performance degradation. This is because BN layers not only introduce the non-linearity operation in the model but also compute gradients with respect to a batch of input samples. This can lead to inaccuracies in the computation of the density gradients. Nevertheless, our implementation still demonstrates robustness compared to the vanilla model and the model using InputGrad regularization. These results suggest that finding a more effective approach to address numerical stability issues and extend the robustness of our regularization method from small to large models is a promising direction for future work.

Table 3: Experimental results on BlockMNIST. Feature leakage, standard accuracy (%) and adversarial accuracy (%) under L_2 PGD-20 threat models are reported.

Method	Feature Leakage ↓	Adv. Accuracy ↑	Accuracy ↑
MLP	3.657	73.57	**99.12**
MLP + InputGrad Reg.	3.461	83.46	**94.56**
MLP + Our Reg. (Stable & Efficient)	2.483	84.36	94.22
MLP + Our Reg. (Stable)	**2.289**	**85.71**	93.45
MLP + Our Reg.	**2.259**	**85.41**	93.05
VGG11	3.418	79.54	**99.18**
VGG11 + InputGrad Reg.	3.792	84.72	97.35
VGG11 + Our Reg. (Stable & Efficient)	**2.899**	**87.79**	**97.74**
VGG11 + Our Reg. (Stable)	**2.878**	**84.75**	96.63
ResNet-18	4.113	39.46	**99.48**
ResNet-18 + InputGrad Reg.	3.969	**89.69**	97.94
ResNet-18 + Our Reg. (Stable & Efficient)	**3.408**	86.67	96.29
ResNet-18 + Our Reg. (Stable)	**3.057**	**89.93**	**99.16**

Table 4: Adversarial robustness comparison on ResNet-18. Adversarial accuracy (%) of the standard trained model (ST) and models with various regularizations under PGD-20 L_2 attack, along with the standard accuracy (%), are reported.

Method	Accuracy	$\|\|\epsilon\|\|_2 = 0.1$	$\|\|\epsilon\|\|_2 = 0.3$	$\|\|\epsilon\|\|_2 = 0.5$
ST	58.62	37.40	11.54	4.92
ST + Score-Matching Reg.	56.66	39.78	13.88	5.29
ST + InputGrad Reg.	57.94	40.94	14.88	5.70
ST + Our Reg.	**58.62**	**47.03**	**26.76**	**15.19**

Table 5: Adversarial robustness comparison on ResNet-18. Adversarial accuracy (%) of the standard trained model (ST) and models with various regularizations under PGD-20 L_∞ attack are reported.

Method	$\|\|\epsilon\|\|_\infty = 1/255$	$\|\|\epsilon\|\|_\infty = 2/255$	$\|\|\epsilon\|\|_\infty = 3/255$
ST	29.06	11.13	4.35
ST + Score-Matching Reg.	30.05	12.85	4.85
ST + InputGrad Reg.	30.95	13.49	5.16
ST + Our Reg.	**41.10**	**26.02**	**13.38**

A.3 Adversarial Robustness Comparison

In this section, we present additional results for the comparison of adversarial robustness. Specifically, we evaluate the performance of ResNet-18 (He et al., 2016) trained on the CIFAR-100 dataset (Krizhevsky et al., 2009), considering both standard accuracy and adversarial accuracy with the varying perturbation budget ϵ. In Table 4 and Table 5, a comprehensive comparison of the adversarial robustness for different models is presented. The first table shows the performance under L_2 adversarial PGD-20 (Madry et al., 2018) attacks, while the second table focuses on the models' performance under L_∞ attacks. We compare the standard trained model and three robust models trained with Score-Matching regularization (Srinivas & Fleuret, 2021), InputGrad regularization (Ancona et al., 2018), and our proposed method. Our results clearly demonstrate the superiority of models trained with our regularization technique. The performance gap is significant, indicating that our approach outperforms the other methods in terms of adversarial robustness under both L_∞ and L_2 attacks. These experimental findings provide compelling evidence that our regularization technique effectively enhances the model's robustness to adversarial attacks.

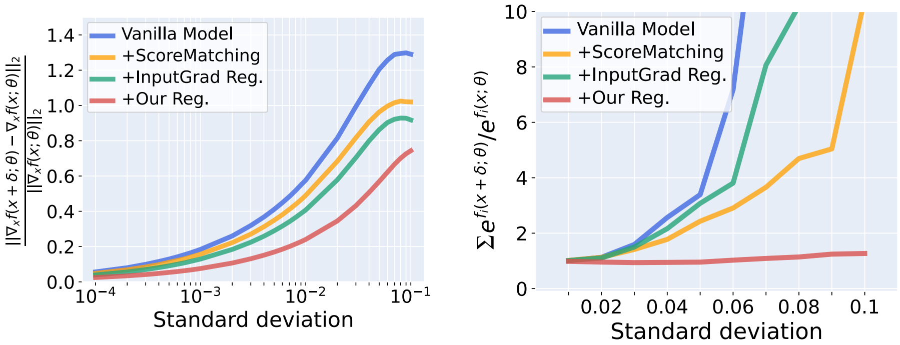
A.4 Out-of-Distribution Detection

Out-of-distribution (OOD) detection (Hendrycks & Gimpel, 2016; Liang et al., 2017; Grathwohl et al., 2020) is a binary classification problem that aims to identify samples that do not belong to the in-distribution dataset. A robust model is expected to generate discriminative outputs capable of distinguishing between samples from in-distribution and out-of-distribution data. In this section, we evaluate the performance of models trained with robust regularizations in detecting OOD samples. To assess the performance of OOD detection in models, we employ the area under the receiver-operating curve (AUROC) as the metric, following the recommendation by Hendrycks & Gimpel (2016).

Table 6 presents the OOD detection results obtained using ResNet-18 trained on the CIFAR-10 dataset (Krizhevsky et al., 2009). In our experiments, we employ the ResNet-18 models trained with different regularizations to detect out-of-distribution samples from both CIFAR-100 and SVHN (Netzer et al., 2011) datasets. The results include the output logits $f(x;\theta)$ as well as the output logit $f_i(x;\theta)$ for a specific label $y = i$. Results of both our regularization and the corresponding efficient implementation are reported. The results clearly demonstrate that our regularization technique, as well as the proposed efficient approach, significantly enhances the model's performance in detecting out-of-distribution samples. Our approach achieves high accuracy and demonstrates the model's capability to effectively discriminate between in-distribution and out-of-distribution samples.

Table 6: AUROC results for OOD detection on ResNet-18. Models with different regularizations trained on CIFAR-10 are employed to detect the out-of-distribution samples from CIFAR-100 and SVHN datasets. Both output logit $f(x;\theta)$ and output logit $f_i(x;\theta)$ for a specific label $y=i$ are compared.

Method	CIFAR-100		SVHN	
	$f_i(x;\theta)$	$f(x;\theta)$	$f_i(x;\theta)$	$f(x;\theta)$
Vanilla Model	0.218	0.511	0.163	0.531
Score-Matching Reg.	0.203	0.523	0.322	0.496
InputGrad Reg.	0.345	0.538	**0.419**	0.570
Our Reg.	**0.372**	**0.557**	0.339	**0.570**
Our Reg. (Efficient)	**0.378**	**0.554**	**0.470**	**0.590**



(a) Relative Gradient Robustness.

(b) Relative Density Robustness.

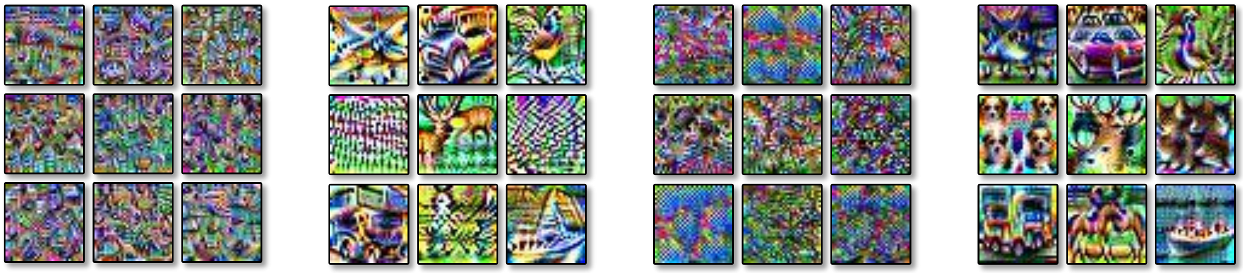
Figure 8: Robustness comparison against perturbation in input gradients and density on ResNet-18 trained on CIFAR-100. **(a)-(b)** Relative gradient robustness and relative density robustness against uniform noise with increasing deviation are shown respectively. **Lower** curves indicate **better** results.

A.5 Robustness against Input Gradient and Density Perturbations

In this section, we present additional results regarding the robustness of models against Gaussian noise δ with the increasing standard deviation in both input gradients and density on the CIFAR-100 dataset. We compare the robustness of the vanilla ResNet-18 model and models trained with three regularizations: Score-Matching regularization, InputGrad regularization and our proposed regularization. Figure 8(a) shows the relative gradient robustness, denoted as $||\nabla_x f(x+\delta;\theta) - \nabla_x f(x;\theta)||_2/||\nabla_x f(x;\theta)||_2$. We can observe that our proposed regularization method not only achieves comparable results but also surpasses other methods by a significant margin. In Figure 8(b), we present the robustness comparison of the relative density through $\sum_{i=1}^{C} e^{f(x+\delta;\theta)}/e^{f(x;\theta)}$. Notably, our regularization technique leads to a high level of robustness against perturbations in density. The ability to maintain relative density robustness is closely associated with the strong generative capabilities of the models. Moreover, more visualization results by maximizing the activations are provided in Appendix A.6.

A.6 Activation Visualization

In this section, we present visualization samples generated by applying gradient ascent on random inputs. We compare the visualization results of different regularizations on both WideResNet-28 (Zagoruyko & Komodakis, 2016) and ResNet-18 models trained on CIFAR-10 and CIFAR-100 respectively. Figure 9 displays the visualization results for WideResNet-28, while Figure 10 presents the results for ResNet-18. The visualization results obtained from the model using our regularization exhibit reduced noise and present more interpretable patterns. The improvement in visualization quality serves as evidence that our technique enhances the interpretability of the underlying features learned by the models. Our method is effective in enhancing the interpretability and clarity of the learned representations. The reduction of noise and the emergence of more interpretable patterns contribute to a better understanding of the model's decision-making process and aid in capturing relevant features for the respective classes.



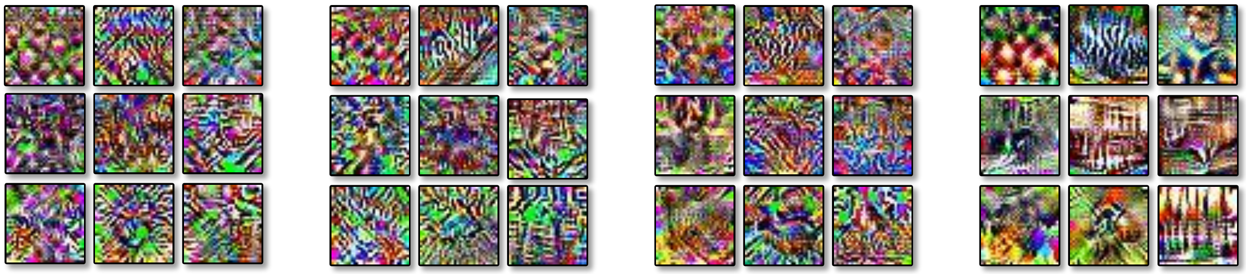
(a) WideResNet-28

(b) WideResNet-28 with InputGrad Reg.

(c) WideResNet-28 with Score-Matching Reg.

(d) WideResNet-28 with Our Reg.

Figure 9: Visualization samples are generated by applying gradient ascent on random inputs using WideResNet-28 trained on CIFAR-10. The visualization results show nine different classes in CIFAR-10. Our method demonstrates superior performance by exhibiting reduced noise and more interpretable patterns in the visualization results.



(a) ResNet-18

(b) ResNet-18 with InputGrad Reg.

(c) ResNet-18 with Score-Matching Reg.

(d) ResNet-18 with Our Reg.

Figure 10: Visualization samples are generated by applying gradient ascent on random inputs using ResNet-18 trained on CIFAR-100. The visualization results show nine different classes in CIFAR-100. Our method demonstrates superior performance by exhibiting reduced noise and more interpretable patterns in the visualization results.

A.7 Attribution Maps and Insertion Games

In this section, we present additional results of attribution maps and insertion games. In Figure 11, Figure 12 and Figures 13, we generate attribution maps using the Integrated Gradients method (Sundararajan et al., 2017) and compute the area under the curve (AUC) of the insertion games for representative samples from BlockMNIST, CelebA and Waterbirds datasets.

The attribution maps demonstrate the effectiveness of our regularization method in suppressing the feature leakage problem on BlockMNIST. The lower values of feature leakage observed in the attribution maps indicate that our approach successfully mitigates the issue of irrelevant or misleading features being attributed to certain classes. Similarly, the attribution maps generated for input from CelebA and Waterbirds dataset show improved interpretability. Moreover, corresponding insertion games are performed to evaluate the model robustness of their highly attributed features. Specifically, the pixels will be interactively inserted in a zero input by their attributions computed by Integrated Gradients. The AUC of the fractional output change with increasing inserted pixels is calculated. For ease of comparison, we sort the output changes of samples in CelebA and Waterbirds datasets.

These results highlight the benefits of our regularization technique, both in terms of improving interpretability and enhancing the model's performance in detecting objects. The results provide additional evidence of the efficacy of our approach in achieving superior performance and robustness.

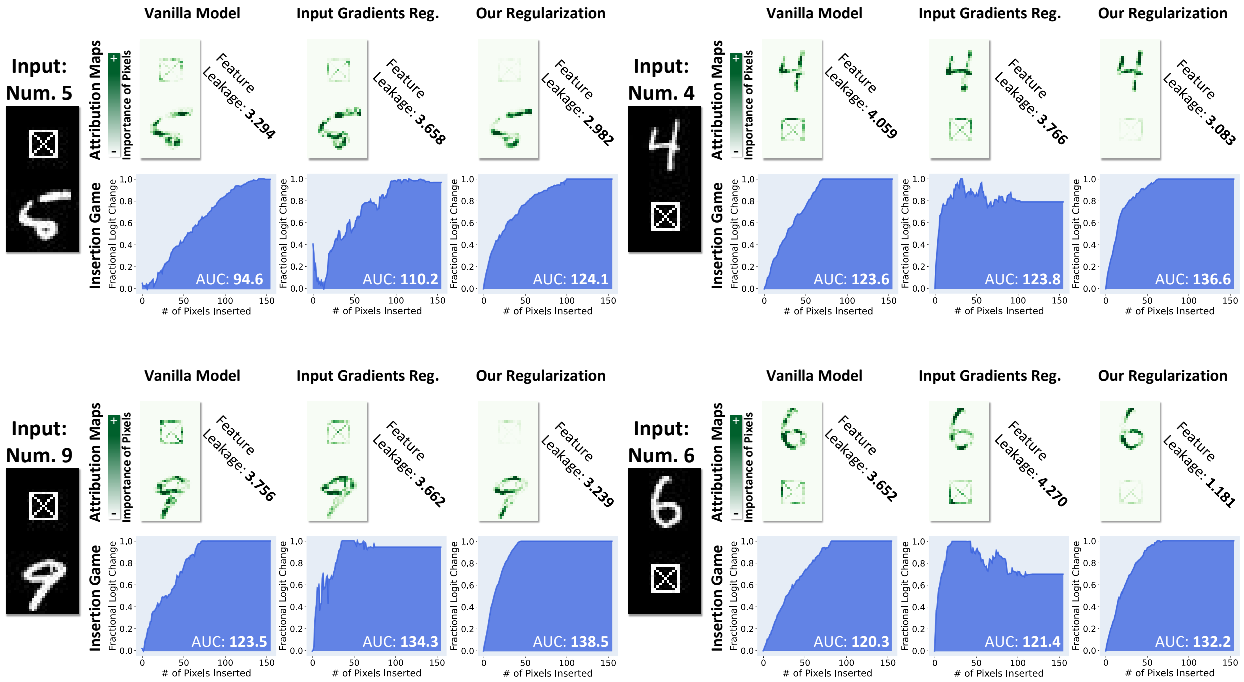


Figure 11: Attribution maps generated by Integrated Gradients and the area under the curve (AUC) of the insertion games for representative samples from BlockMNIST. As compared to the vanilla model and the model with Input Gradient regularization, our regularization leads to interpretable attribution maps with reduced feature leakage and fewer spurious correlations, while also achieving higher AUC for the insertion game.

A.8 Experimental Setup

In this section, we provide details regarding the datasets, models, and the experimental platform employed in our experiments.

A.8.1 Datasets

BlockMNIST. BlockMNIST dataset (Shah et al., 2021) is an extension of the MNIST dataset (LeCun et al., 1998). Each sample in BlockMNIST is derived from an original MNIST sample by adding a *null* block, which contains non-informative features, randomly positioned at the top or bottom of the image. During the training process, each BlockMNIST sample is generated by randomly attaching the null block to MNIST samples. In the testing process, models are evaluated on the same test samples with fixed-placed null blocks.

CelebA. Liu et al. (2015) introduced the CelebA dataset for facial attribute recognition. Sagawa et al. (2019) further constructed the training set consisting of 162,770 training samples. The smallest group within this dataset comprises male celebrities with blond hair, containing 1,387 samples. In our experiment, we adopt the same dataset configuration, with hair color (blond & dark) as the target attribute and gender (male & female) as the spurious correlated features.

Waterbirds. Waterbirds dataset (Sagawa et al., 2019) is constructed by combining the CUB-200-2011 (Wah et al., 2011) and Places datasets (Zhou et al., 2017). Specifically, the bird images from CUB-200-2011 are cropped using segmentation annotations and then positioned on backgrounds from the Places dataset, which consists of land or water scenes. The placement of the bird images on the backgrounds is determined by the category of the birds, i.e., whether they are land or water birds. Consistent with the settings in (Sagawa et al., 2019), we follow the same approach of placing 95% of all waterbirds against a water background and the remaining 5% against a land background.

CIFAR-10 and CIFAR-100. CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009) are widely utilized for evaluating the recognition capabilities of various models. The CIFAR-10 dataset consists of 60,000 images, each with dimensions of 32×32×3, and is divided into 10 different classes, with 6,000 images per class. The dataset is further

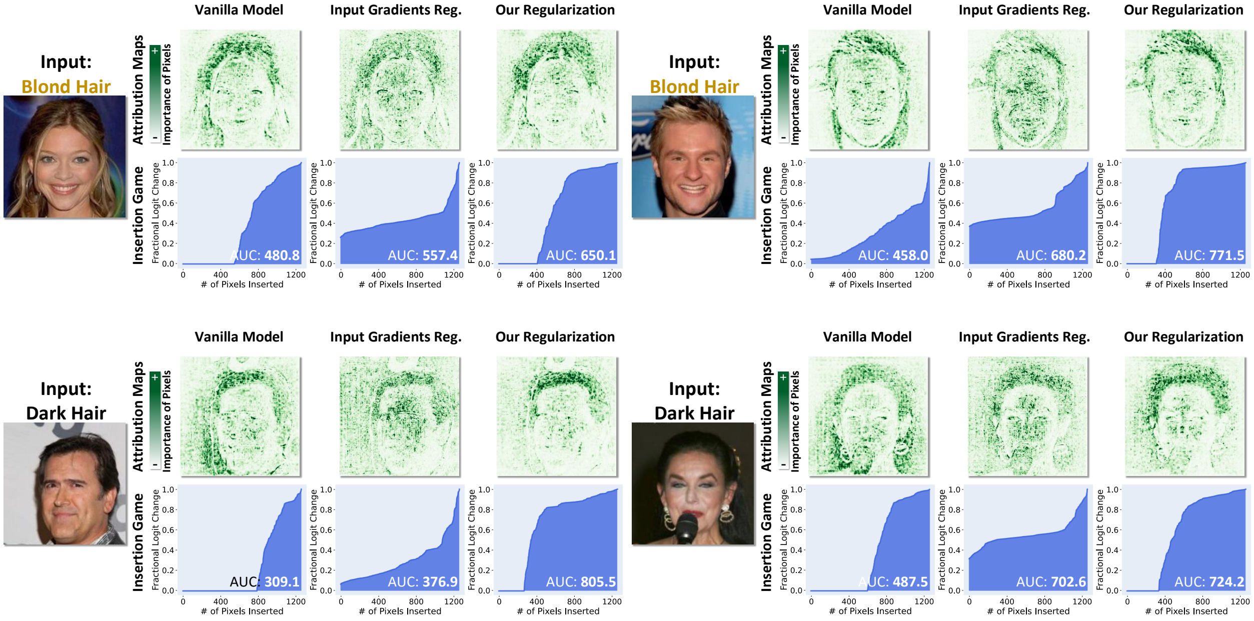


Figure 12: Attribution maps generated by Integrated Gradients and the area under the curve (AUC) of the insertion games for representative samples from CelebA. As compared to the vanilla model and the model with Input Gradient regularization, our regularization leads to lower feature leakage while also achieving higher AUC for the insertion game.

partitioned into a training set containing 50,000 samples and a separate test set comprising 10,000 samples. Similarly, CIFAR-100 also comprises 60,000 images, but it offers a more fine-grained classification task with 100 distinct classes.

SVHN. SVHN dataset (Netzer et al., 2011) is a collection of real-world images depicting house numbers captured from street views. It contains a training set of 73,257 images and a test set of 26,032 images. The dataset exhibits diverse variations in lighting conditions, viewpoints, and digit appearances, reflecting the challenges encountered in real-world scenarios.

A.8.2 Models

MLP. We train two-hidden-layer MLPs using different techniques and regularizations on the BlockMNIST dataset for 80 epochs with a learning rate of 0.0001. In our experiments, two L_∞ adversarially trained models are compared. To augment the training samples, we generated perturbations using PGD-3 (Madry et al., 2018) and FGSM (Goodfellow et al., 2015) attacks separately. For PGD adversarial accuracy, we test all models under L_∞ and L_2 threats. We employ steps of $\alpha = 0.2$ and $\alpha = 0.01$, within the perturbation budgets of $\epsilon = 0.3$ and $\epsilon = 1.0$. Regarding FGSM accuracy, we generated L_2 adversarial perturbations using a step size of $\alpha = 0.25$ and a budget of $\epsilon = 0.2$.

ResNet. In CelebA and Waterbirds datasets, we train ResNet-34 models (He et al., 2016) with different regularizations for 50 epochs and 300 epochs separately. Vanilla ResNet-34 and ResNet-34 with GroupDRO (Sagawa et al., 2019) are trained with a learning rate of 0.0001 and all compared models with different regularizations are trained with a learning rate decayed by 10. In the training process, each batch of training samples is re-weighted to have the same number of samples in each group. In CIFAR-10 and CIFAR-100 dataset, we train ResNet-18 for 200 epochs with a learning rate of 0.01 decayed by 10 in the 100-th and 175-th epochs.

WideResNet. To perform activation visualization, WideResNet-28 (Zagoruyko & Komodakis, 2016) is also trained in CIFAR-10 for 200 epochs with a learning rate of 0.01 decayed by 10 in the 100-th and 175-th epochs.

VGGNet. We trained VGG11 models (Simonyan & Zisserman, 2015) on the BlockMNIST dataset using various techniques and regularizations. The training process involved 80 epochs with a fixed learning rate of 0.0001.

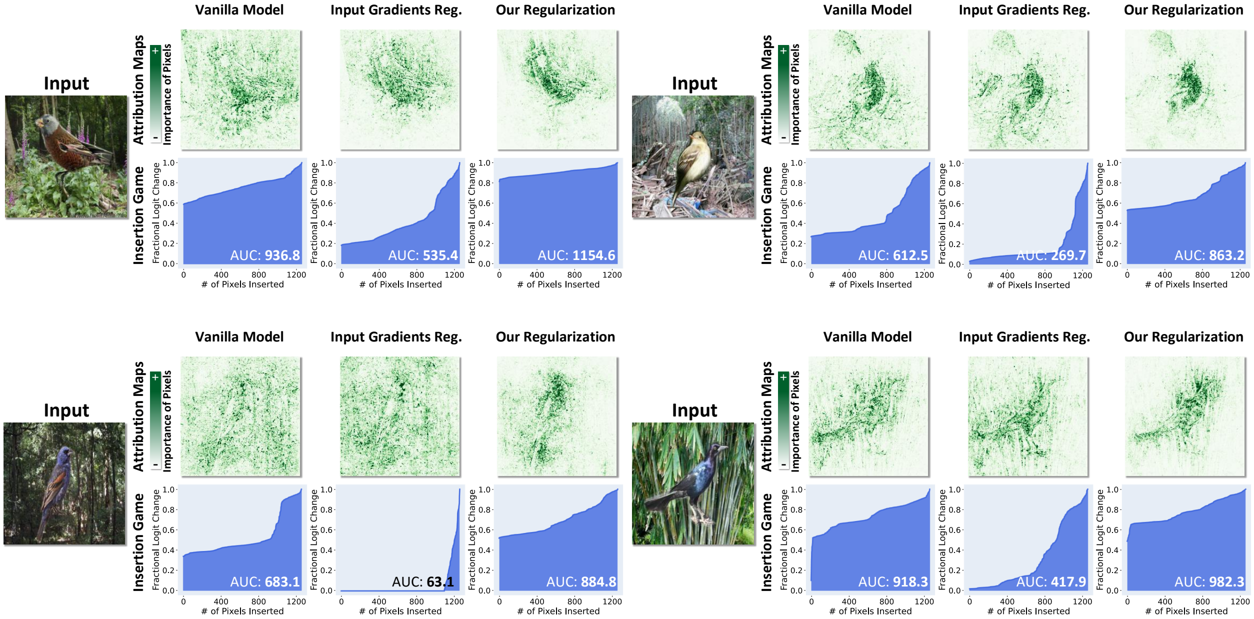


Figure 13: Attribution maps generated by Integrated Gradients and the area under the curve (AUC) of the insertion games for representative samples from Waterbirds. As compared to the vanilla model and the model with Input Gradient regularization, our regularization leads to lower feature leakage while also achieving higher AUC for the insertion game.

A.8.3 Experimental Platform

All experiments were performed on a Linux machine equipped with an NVIDIA GTX 3090Ti GPU featuring 24GB of memory. The machine also consisted of a 16-core 3.9GHz Intel Core i9-12900K CPU and 128GB of main memory. The models were tested and trained using the PyTorch deep learning framework (v1.12.1) in the Python programming language.