Attack-free Evaluating and Enhancing Adversarial Robustness on Categorical Data

Yujun Zhou [*1] **Yufei Han** [*2] **Haomin Zhuang** [1] **Hongyan Bao** [3] **Xiangliang Zhang** [1]

Abstract

Research on adversarial robustness has predominantly focused on continuous inputs, leaving categorical inputs, especially tabular attributes, less examined. To echo this challenge, our work aims to evaluate and enhance the robustness of classification over categorical attributes against adversarial perturbations through efficient attack-free approaches. We propose a robustness evaluation metric named Integrated Gradient-Smoothed Gradient (IGSG). It is designed to evaluate the attributional sensitivity of each feature and the decision boundary of the classifier, two aspects that significantly influence adversarial risk, according to our theoretical analysis. Leveraging this metric, we develop an IGSG-based regularization to reduce adversarial risk by suppressing the sensitivity of categorical attributes. We conduct extensive empirical studies over categorical datasets of various application domains. The results affirm the efficacy of both IGSG and IGSG-based regularization. Notably, IGSG-based regularization surpasses the state-of-the-art robust training methods by a margin of approximately 0.4% to 12.2% on average in terms of adversarial accuracy, especially on high-dimension datasets. The code is available at https://github.com/YujunZhou/IGSG.

1. Introduction

Adversarial attacks (Goodfellow et al., 2014) pose significant concerns in safety-critical applications by exploiting vulnerabilities in deep learning models, thereby impacting their decision-making (Moosavi-Dezfooli et al., 2016; Madry et al., 2017; Carlini & Wagner, 2017). To mitigate these risks, defense strategies, known as **robustness enhancement** methods, have been developed to reduce the model's adversarial risk (Shafahi et al., 2019; Zhang et al., 2019; Wong et al., 2020). Additionally, the **evaluation of robustness** in current works is typically quantified by the success rate of various attack methods against adversarial risk (Madry et al., 2017; Carlini & Wagner, 2017; Kim et al., 2024). Despite extensive research of these aspects in the continuous domain, the discrete domain, which is prevalent in real-world safety-critical applications, has not been as thoroughly investigated (Bao et al., 2023). However, these methods for robustness evaluation and enhancement in continuous settings are not directly transferable to categorical data. Firstly, robustness evaluation through attacks is computationally demanding for categorical data. The generation of commonly used L_0-norm bounded adversarial perturbations on categorical data (Lei et al., 2019; Wang et al., 2020) poses an NP-hard problem (Lee & Leyffer, 2011). Secondly, adversarial training, a typical robustness enhancement strategy that involves iterative generation of adversarial samples for optimization, becomes computationally intensive when transferred from continuous to categorical input. Besides, due to the NP-hard nature of generating adversarial samples, adversarial training can only cover a subset of adversarial samples, leading to "robust overfitting" (Rice et al., 2020). This leads us to our research questions:

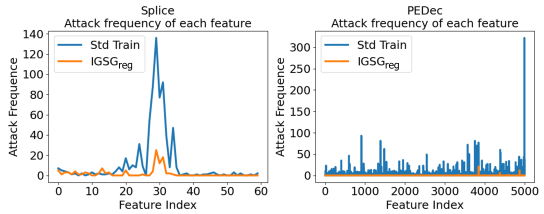
***Q1**: How can the adversarial robustness of deep learning models on categorical data be evaluated without performing attacks?*

***Q2**: How can such an attack-free robustness assessment be utilized for robustness enhancement?*

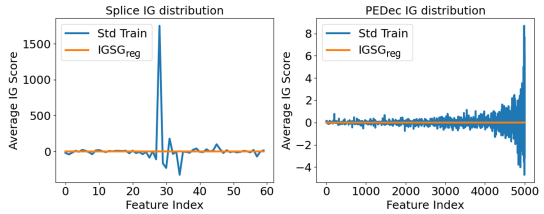
To answer these questions, we identify two primary factors affecting the adversarial robustness of one model on categorical data. The first factor concerns **the model's over-reliance on a limited subset of features**. Such dependence signifies that decisions within the model disproportionately prioritize few features, leading to increased vulnerability and a heightened risk of adversarial attacks (Grosse et al., 2016). As shown in Fig.1(a), in models trained through standard methods (*Std Train*), only a few features are subjected to attacks more frequently, e.g., the 30th to 32nd features

[*]Equal contribution [1]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, USA [2]INRIA, France [3]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. Correspondence to: Xiangliang Zhang <xzhang33@nd.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Attack frequency ($IGSG_{\text{reg}}$ vs *Std Train*)



(b) IG score distribution ($IGSG_{\text{reg}}$ vs *Std Train*)

Figure 1. Comparison of attack frequency and IG score for each feature on *Splice* and *PEDec* Datasets

in the Splice dataset, and those around the 5000th in the PEDec dataset. To quantify the impact of each feature on the model's adversarial robustness, we employ Integrated Gradient (IG), a measure grounded in theoretical principles (Sundararajan et al., 2017). As depicted in Fig. 1(b), for the *Std Train* models, features that are more frequently targeted by attacks correspondingly exhibit notably high IG scores. Given that computing IG scores is less resource-intensive compared to generating adversarial samples for categorical data, we aim to address **Q1** by exploring the feasibility of using IG as an effective measure of adversarial robustness.

Regarding the second factor, we delve into the **impact of gradient magnitude on the model's classification boundary**. The gradient magnitude, which indicates the curvature of the classification boundary, reflects an essential aspect of the model's adversarial risk (Yang et al., 2021). A larger gradient magnitude, indicating a steeper curvature, is associated with reduced robustness. In the context of discrete domains, it has been shown that gradient magnitude can reflect incremental increases in adversarial risk (Wang et al., 2020). Therefore, to address **Q1**, we further incorporate gradient magnitude into our robustness measure. Specifically, we utilize smoothed gradient (SG) magnitude (Smilkov et al., 2017) to avoid the issue of obfuscated gradients (Athalye et al., 2018), and combine it with IG to form a new robustness assessment metric, named **Integrated Gradient-Smoothed Gradient (IGSG)**. The theoretical exploration of IGSG's effectiveness in assessing adversarial robustness is detailed in Section 3.2.

Given that IGSG offers an attack-free method for assessing the adversarial robustness of a model, we explore using IGSG to address **Q2**. Specifically, we integrate IGSG as a regularization term during training to enhance a model's adversarial robustness. This novel approach, named IGSG-based regularization ($IGSG_{\text{reg}}$), significantly enhances the model's resilience to adversarial attacks. Fig.1 demonstrates this improvement: compared to *Std Train*, models regularized with IGSG show a more even distribution of IG scores across different features and experience a lower frequency of attacks on previously vulnerable features.

The contributions of our work are outlined as follows:

- As an echo to **Q1**, we employ Integrated Gradient (IG) and Smoothed Gradient (SG) as two computationally efficient metrics to evaluate the adversarial robustness of models on categorical data. Additionally, we formulate an information-theoretic upper bound that underpins the rationale for employing IGSG in robustness assessment.
- To further address **Q2**, we propose IGSG-based regularization as a novel robustness enhancement to improve the smoothness of feature contributions and decision boundaries on categorical data.
- Finally, we conduct extensive experimental evaluation involving different model architectures and diverse datasets to demonstrate the feasibility of IGSG as robustness evaluation (17-240 times faster than OMPGS) and the effectiveness of using IGSG for robustness enhancement.

2. Related Works

Robustness evaluation. Typically, robustness evaluation of models on continuous data is performed using the attack success rate under various attack methods (Goodfellow et al., 2014; Madry et al., 2017; Moosavi-Dezfooli et al., 2016; Croce & Hein, 2020; Wang et al., 2020) or by measuring certified accuracy within a specified perturbation radius in certifiable robustness research (Lee et al., 2019; Cohen et al., 2019; Zhai et al., 2020). However, evaluating a model's robustness with categorical inputs often relies on computationally intensive greedy search techniques, like FSGS (Elenberg et al., 2018). While OMPGS (Wang et al., 2020) attempts to simplify the greedy search with gradients, its attack performance degenerates, and its computational expense remains significantly higher than that of continuous domain methods like Projected Gradient Descent (PGD) (Madry et al., 2017). Moreover, certifiable robustness methods predominantly target L_p or L_∞ norm bounded perturbations and do not offer guaranteed robustness against L_0 norm bounded perturbations (Cohen et al., 2019; Salman et al., 2019). To overcome these limitations, we focus on analyzing adversarial risk with L_0 norm bounded perturbation for categorical data. We propose employing Integrated Gradient (IG) and Smoothed Gradient (SG) as two efficient metrics to evaluate the adversarial robustness.

Robustness enhancement. Adversarial training and robust regularization are two primary robustness enhancement methods. Adversarial training is the most acknowledged defense mechanism (Madry et al., 2017). It employs min-

max optimization, generating adversarial samples via Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014; Wong et al., 2020; Andriushchenko & Flammarion, 2020; Zhang et al., 2022; Huang et al., 2023) or PGD (Madry et al., 2017; Zhang et al., 2019). These adversarial samples are subsequently utilized as training data to enhance the robustness. Adversarial Feature Desensitization (AFD) (Bashivan et al., 2021) leverages a GAN-like loss to learn invariant features against adversarial perturbations. While these methods, designed for continuous data, can handle L_1-norm bounded adversaries for relaxed categorical data, their ability to deliver consistent performance remains uncertain. For categorical data, adversarial training with search-based attack methods (Lei et al., 2019; Wang et al., 2020) requires significant time to generate adversarial samples. (Xu et al., 2023) proposed adapting adversarial training from continuous to discrete domains, but the MINLP nature of such training presents challenges in generating a comprehensive range of samples for robust defense. In the language domain, FreeLB (Zhu et al., 2019; Li et al., 2021) applies multiple PGD steps to word embeddings. Nevertheless, it relies on language-specific constraints, limiting its broader applicability. Furthermore, the challenge of "robust overfitting" in adversarial training (Rice et al., 2020) is mitigated by (Chen et al., 2020; Yu et al., 2022) in the continuous domain, but our investigation reveals this overfitting issue persists in the discrete feature space, unaddressed by existing continuous domain methods. In comparison to adversarial training, our approach is less computationally demanding. Moreover, it avoids robust overfitting caused by insufficient coverage in the exploration of the categorical adversarial space.

Regularization-based methods provide an alternative for enhancing adversarial robustness by penalizing the complexity of the target classifier. Prior studies (Ross & Doshi-Velez, 2018; Finlay & Oberman, 2021) have suggested gradient magnitude regularization, while others (Jakubovitz & Giryes, 2018; Hoffman et al., 2019) have focused on penalizing the Frobenius norm of the Jacobian matrix. Some research (Chen et al., 2019; Sarkar et al., 2021) has proposed using IG to measure feature contributions and applying regularization over IG. However, these methods were not specifically tailored to enhance adversarial robustness. Our work highlights the efficacy of $IGSG_{\text{reg}}$ in the context of adversarial robust training. Crucially, we demonstrate the importance of concurrently regularizing both the gradient magnitude and the IG distribution across different features for a more comprehensive and effective defense strategy.

3. Robustness Evaluation Metric Development

Preliminary. Let's assume that a random sample $x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,p}\}$ has p categorical features and a class label y_i. Each feature $x_{i,j}$ can choose one out of m possible category values. Following the one-hot encoding scheme, we can represent x_i as a binary $\mathbb{R}^{p*m}$ matrix $b(x_i)$. Each row of $b(x_i)$ corresponds to the value chosen by feature $x_{i,j}$, i.e., $b(x_i)_{j,k^*} = 1$ when $x_{i,j}$ selects the k^*-th category value, and $b(x_i)_{j,k\neq k^*} = 0$ $(k = 1, 2, ..., m)$. An adversarial sample $\hat{x}_i = \{\hat{x}_{i,j}, {}_{j=1,\ldots,p}\}$ is generated by modifying the categorical values of a few features of x_i. The number of changed features from x_i to $\hat{x}_i$ is noted as $\text{diff}(x_i, \hat{x}_i)$. Given a classifier f and taking $b(x_i)$ as input to f, $f(b(x_i))$, simplified as $f(x_i)$, predicts its corresponding label y_i.

Definition of adversarial risk. Before establishing our robustness evaluation metric, it is essential to firstly delineate the concept of adversarial risk.

Definition 3.1. We consider a hypothesis space $\mathcal{H}$ and a non-negative loss function ℓ: $\mu_z \times \mathcal{H} \to R^+$. Following (Xu & Raginsky, 2017; Asadi et al., 2018), given a training dataset S^n composed of n i.i.d training samples $z_i \sim \mu$, we assume a randomized learning paradigm $\mathcal{A}$ mapping S^n to a hypothesis f, i.e., $f = \mathcal{A}(S^n)$, according to a conditional distribution $P_{f|S^n}$. The adversarial risk of f, noted as $\mathcal{R}_f^{adv}$, is given in Eq.1. It is defined as the expectation of the worst-case risk of f on any data point $z = (x, y) \sim \mu_z$ under the L_0-based attack budget $\text{diff}(x, \hat{x}) \leq \epsilon$. The expectation is taken over the distribution of the n training samples S^n and the classifier $f = \mathcal{A}(S^n)$.

$$\mathcal{R}_f^{adv} = \mathop{\mathbb{E}}_{S^n, P_{f|S^n}} \mathop{\mathbb{E}}_{z=(x,y)\sim\mu_z,} \sup_{\text{diff}(x,\hat{x})\leq\epsilon} \ell(f(\hat{x}), y). \quad (1)$$

As defined, $\mathcal{R}_f^{adv}$ measures the worst-case classification risk over an adversarial input $\hat{z} = (\hat{x}, y)$ where the attacker can modify at most ϵ categorical features. The empirical adversarial risk of f is given in Eq.2. It is defined as the expectation of the worst-case risk over adversarial samples $\hat{z} = (\hat{x}, y)$ over the joint distribution of S^n and $P_{f|S^n}$.

$$\hat{\mathcal{R}}_f^{adv} = \mathop{\mathbb{E}}_{S^n, P_{f|S^n}} \frac{1}{n} \sum_{z_i=(x_i,y_i)\in S^n} \sup_{\text{diff}(x_i,\hat{x}_i)\leq\epsilon} \ell(f(\hat{x}_i), y_i), \quad (2)$$

Intuitively, the empirical adversarial risk is the average loss under the strongest attack, which is exactly the attack success rate (ASR) under the strongest attack with 0-1 loss. However, considering that obtaining ASR via attacks is computationally intensive for categorical data, we aim to define the robustness evaluation without resorting to attacks.

3.1. IGSG: Robustness Evaluation without Attack

Feature-wise Integrated Gradient (IG). To measure the extent to which a model depends on specific features for making predictions, we propose to use Integrated Gradient (IG) as a metric. IG offers a theoretical guarantee for assessing the contribution of individual features to the classification output (Sundararajan et al., 2017). To adapt the computation of IG scores for categorical features, we start by defining a *baseline* input x'. This involves expanding the set of possible category values for each feature $x_{i,j}$ by introducing an

additional dummy category, labeled as $m+1$ and embedded as a zero vector. For this baseline input, each feature is set to the dummy category value, such that $b(x')_{j,m+1} = 1$, $b(x')_{j,k} = 0 (k = 1, 2, \ldots, m)$. By inputting $b(x')$ into the classifier, it transmits no useful information for classification, rendering it an ideal non-informative baseline.

With this defined baseline input x', the IG score for each categorical feature $x_{i,j}$ can be approximated as follows:

$$IG(x_i)_j = \sum_{k=1}^{m} IG(x_i)_{j,k} = \sum_{k=1}^{m} \Big[(b(x_i)_{j,k} - b(x')_{j,k}) \times \frac{1}{T} \sum_{t=1}^{T} \frac{\partial f(b(x') + \frac{t}{T} \times [b(x_i) - b(x')])}{\partial b(x_i)_{j,k}} \Big] \tag{3}$$

where T is the number of steps in the Riemman approximation of the integral. We empirically choose T=20, which provides consistently good measurement. $IG(x_i)_j$ derived along the trajectory between $b(x')$ and $b(x_i)$ hence represents the contribution of $x_{i,j}$ to the classifier's output.

Integrated Gradient (IG) of a classifier. To quantify the sensitivity of a classifier over all features, we first apply a Total-Variance (TV) function over the feature-wise IG scores: $\ell_{TV} IG(x_i) = \sum_{j=1}^{p-1} |IG(x_i)_j - IG(x_i)_{j+1}|$. This definition follows the TV loss used in time series data analysis (Chambolle, 2004), and calculates the sum of the absolute differences between the normalized IG scores of neighboring features. The IG score of a classifier is defined as:

$$IG = \mathop{\mathbb{E}}_{(x_i, y_i) \sim \mu_z} \ell_{TV} IG(x_i) \tag{4}$$

To eliminate the impact of feature ordering on the value of IG, we consider all possible permutations of the feature ordering and compute the average of the Total Variation (TV) loss across the IG scores for these permutations, denoted as IG_{avg}.

Smoothed Gradient (SG) of a classifier. We then measure the gradient magnitude of the classifier's decision boundary, since it is relevant to the model's adversarial risk (Wang et al., 2020). We compute the gradient of the classification loss with respect to the one-hot encoded representation of $b(x_i)$, represented as $\nabla_{b(x_i)} \ell(x_i, y_i; \theta) \in \mathbb{R}^{p*m}$. Each element of $\nabla_{b(x_i)} \ell(x_i, y_i; \theta)$ is defined as $\frac{\partial}{\partial b(x_i)_{j,k}} \ell(x_i, y_i; \theta)$. (Yang et al., 2021) highlighted that $\nabla_{b(x_i)} \ell(x_i, y_i; \theta)$ measures the curvature of the decision boundary around the input. A larger magnitude of this gradient indicates a more twisted decision boundary, implying a less stable decision around the input. In our work, to avoid the pitfalls of obfuscated gradients, we define Smoothed Gradient (SG) (Smilkov et al., 2017) for measuring the magnitude by:

$$SG = \mathop{\mathbb{E}}_{(x_i, y_i) \sim \mu_z} \frac{1}{R} \sum_{r=1}^{R} ||G_r||_q$$
$$\text{where } G_{r,j,k} = \frac{\partial}{\partial b(x_r)_{j,k}} \ell(x_r, y_i; \theta) - \frac{\partial}{\partial b(x_r)_{j,k^*}} \ell(x_r, y_i; \theta) \tag{5}$$

Here R is the number of sampled instances, e.g., $R = 5$.

Finally, the robustness metric IGSG is defined as:

$$IGSG = IG_{avg} + \alpha * SG \tag{6}$$

Here, α functions as a hyper-parameter, serving to balance the two terms to ensure their magnitudes are comparable.

3.2. Theoretical Explanation for IGSG

In this section, we establish an information-theoretic framework to elucidate the viability of employing IG and SG for assessing the adversarial robustness of a classifier.

Theorem 3.2. *Let $\ell(f(x_i), y_i)$ be L-Lipschitz continuous for any $z_i = (x_i, y_i)$. Let $\mathcal{D}_f$ be the diameter of the hypothesis space $\mathcal{H}$. For each x_i, the categorical features modified by the worst-case adversarial attacker and the rest untouched features are noted as ω_i and $\overline{\omega_i}$, respectively. Given an attack budget ϵ, the size of ω_i is upper bounded as $|\omega_i| \leq \epsilon$. The gap between the expected and empirical adversarial risk in Eq.1 and Eq.2 is bounded from above, as given in Eq.7.*

$$\begin{aligned}
&\mathcal{R}_f^{adv} - \hat{\mathcal{R}}_f^{adv} \leq \\
&\frac{L\,\mathcal{D}_f}{\sqrt{2}n} \times \left(\sum_{i=1}^{n} I(f; z_i) + 2 \sum_{i=1}^{n} \Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) + \sum_{i=1}^{n} \Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i}) \right)^{\frac{1}{2}}, \\
&\textit{where } \Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) = |I(x_{i,\omega_i}; f) - I(x_{i,\overline{\omega_i}}, y_i; f)|, \\
&\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i}) = \gamma |I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i, f) - I(x_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i, f)|, \\
&\gamma = \max_{z_i = (x_i, y_i) \in S^n, |\omega_i| \leq \epsilon} 1 + \sigma, \\
&\sigma = \frac{|I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i) - I(x_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i)|}{|I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i, f) - I(x_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i, f)|},
\end{aligned} \tag{7}$$

where x_{i,ω_i} and $\hat{x}_{i,\omega_i}$ are ω_i features before and after injecting adversarial modifications, and $I(X; Y)$ is the mutual information between two random variables X and Y.

The proof and the discussion of the tightness of Eq.7 can be found in App.A. Also, we demonstrate that when there are no adversaries present, meaning $\hat{z} = z$, the bound presented in Eq.7 converges to a tight characterization of generalization error for a broad range of models, resonating with the convergence unveiled in (Zhang et al., 2021; Bu et al., 2019). We link Theorem 3.2 to the randomized learning paradigm and the PAC-Bayes bound, with further details available in App.C and D.

The information-theoretical adversarial risk bound established in Eq.7 unveils two major factors influencing the adversarial risk over categorical inputs, corresponding to the consideration of SG and IG respectively. Detailed analysis of the connection between Theorem 3.2 and IGSG can be found in App.B.

Factor 1. Lower $I(f; z_i)$ for each training sample z_i indicates lower the adversarial risk f. $I(f, z_i)$ in Eq.7 represents the mutual information between the classifier

f and each training sample z_i. Pioneering works (Xu & Raginsky, 2017; Bu et al., 2019; Zhang et al., 2021) have established that lower $I(f, z_i)$ corresponds to a diminishing adversary-free generalization error. As widely acknowledged in adversarial learning and emphasized in Eq.7, a better-generalized classifier exhibits better resilience to adversarial attacks, resulting in lower adversarial risk. SG, defined in Eq.5, measures the decision boundary smoothness, thereby assessing the model's generalization capability.

Factor 2. Reducing $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ helps smooth the feature-wise impact to classification, thus reducing the adversarial risk. This corresponds to reducing the impact of excessively influential features, i.e., minimizing IG, for two reasons. First, in $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$, $I(x_{i,\omega_i}; f)$ and $I(x_{i,\overline{\omega_i}}, y_i; f)$ reflect the contribution of the feature subset ω_i and the rest features $\overline{\omega}_i$ to f. Features with higher mutual information have more substantial influence on the decision output. Minimizing $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ thus decreases the contribution gap between the attacked and untouched features. It prompts the classifier to maintain a more balanced reliance on different features, thereby making it harder for adversaries to exploit influential features. Second, $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ measures the sensitivity of features in ω_i, in terms of how adversarial perturbations to this subset of features affect both the classification output and the correlation between ω_i and $\overline{\omega}_i$. Minimizing $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ makes the classifier's output less sensitive to the perturbations over input features, which limits the negative impact of adversarial attacks. IG, defined in Eq.4, quantifies the classifier's sensitivity across all features, thereby indicating its adversarial robustness.

Beyond the two factors, minimizing the empirical adversarial risk $\hat{\mathcal{R}}_f^{adv}$ in Eq.7 may also reduce the adversarial risk. This concept is synonymous with the principles of adversarial training. Nevertheless, as analyzed in App.E, the efficacy of adversarial training is restricted.

To empirically assess this bound, we conducted experiments detailed in App.I.1. These experiments involve estimating $I(f; z_i)$, $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$, and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ as specified in the theoretical bound. Our findings suggest that models with a lower upper bound exhibit enhanced adversarial robustness against attacks.

4. *IGSG*$_{\text{reg}}$: Attack-free Robust Training

Based on the above analysis, IGSG, an attack-free robustness assessment method, can be employed for enhancing robustness. It can be incorporated during the training phase as a regularization term to minimize adversarial risk, i.e., by adding to the classification loss as follows:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x_i,y_i)\in S^n} \ell(x_i, y_i; \theta) + \beta * IGSG \tag{8}$$

where β is a hyper-parameter controlling the degree of regularization. Incorporating SG into the loss function enforces the minimization of $I(f; z_i)$, while including IG aims to minimize both $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega i})$. In App.I.1, we demonstrate through empirical evidence that *IGSG*$_{\text{reg}}$ can effectively reduce the estimated value of both mutual information-based terms. Therefore, *IGSG*$_{\text{reg}}$ results in a smoother decision boundary characterized by reduced curvature, mitigates the model's excessive dependency on specific features, and ultimately enhancing its robustness against adversarial attacks.

5. Experimental Evaluation

5.1. Experimental Settings

Summary of Used Datasets. We employ two categorical datasets and one mixed dataset with both categorical and numerical features, each from different applications and varying in the number of samples and features.

Splice-junction Gene Sequences (Splice) (Noordewier et al., 1990). The dataset includes 3190 gene sequences, each with 60 categorical features from the set {A, G, C, T, N}. Each sequence is labeled as intron/exon borders (*IE*), exon/intron borders (*EI*), or neither.

Windows PE Malware Detection (PEDec) (Bao et al., 2021). This dataset, used for PE malware detection, consists of 21,790 Windows executable samples, each represented by 5,000 binary features denoting the presence or absence of corresponding malware signatures. The samples are categorized as either benign or malicious.

Census-Income (KDD) Data (Census) (Lane & Kohavi, 2000). This dataset includes census data from surveys conducted from 1994 to 1995, encompassing 299,285 samples. Each has 41 features related to demographics and employment, with 32 categorical features and 9 numerical features. The task is to determine whether subjects fall into the low-income or high-income group.

For *Splice* and *PEDec*, we use 90% and 10% of the data samples as the training and testing set to measure the adversarial classification accuracy. For *Census*, we use the testing and the training set given by (Lane & Kohavi, 2000), i.e., 199,523 for training and 99,762 for testing.

Baseline Models. We involve one undefended model, seven state-of-the-art robust training baseline models as victim models. Specifically, we include five adversarial training baselines Adv Train (Madry et al., 2017), Fast-BAT (Zhang et al., 2022), TRADES (Zhang et al., 2019), AFD (Bashivan et al., 2021) and PAdvT (Xu et al., 2023), and two regularization-based baselines IGR (Ross & Doshi-Velez, 2018) and JR (Hoffman et al., 2019). To adapt these methods to our context, we relax the one-hot encoded representation of categorical training data except for *PAdvT*, which

is originally designed for categorical data. For the other four adversarial training baselines (*Adv Train*, *Fast-BAT*, *TRADES* and *AFD*), we adopt L_1-norm bounded adversary in the inner maximization of the adversarial training process. The details of the baselines can be found in App.H.2. The hyperparameter settings can be found in App.H.3. We employ Multi-Layer Perceptron (MLP) and Transformer (Vaswani et al., 2017) in all experiments, each conducted five times for consistency.

Robustness Evaluation Protocols. Five domain-agnostic attack methods, FSGS (Elenberg et al., 2018), OMPGS (Wang et al., 2020), PCAA (Xu et al., 2023), FEAT (Bao et al., 2023) and GradAttack (Lei et al., 2019) designed specifically for generating discrete adversarial perturbations in categorical data, are employed to evaluate adversarial robustness. Due to the discontinuous nature of categorical data, traditional attacks like PGD and FGSM cannot be directly applied. Further discussion is presented in App.G. FSGS, OMPGS, PCAA, FEAT, and GradAttack, with proven attack effectiveness across various real-world applications, are suitable for comparing the effectiveness of different robust model training methods on categorical input.

5.2. IGSG for Robustness Evaluation

Implementation Details. To validate the effectiveness of IGSG as a robustness metric, we compare it against established metrics such as Attack Success Rate (ASR) and $m_f(x)$, where $m_f(x)$ measures the difference between the highest non-true class score and the true class score (Bao et al., 2021). We assess IGSG's correlation with these metrics by averaging ASR and m_f scores for attack budgets of 1, 3, and 5, and then calculating the Pearson correlation coefficient with IGSG (Cohen et al., 2009). Additionally, we evaluate the computational efficiency of IGSG (an attack-free measure) in comparison to OMPGS (an attack-based measure). Finally, we verify whether features with the highest IG scores are indeed the most sensitive by conducting sensitivity analysis (Campbell et al., 2008), modifying one feature at a time, and measuring the average change in the classifier's output. This approach helps confirm IGSG's ability to pinpoint critical sensitivities in models, thereby highlighting areas of adversarial risk.

Robustness Evaluation Results: IGSG (attack-free) vs. other attack-based metrics. Table 1 presents the robustness evaluation results for MLP models across three datasets. These models include those trained in a standard manner without any defense (*Undefended Std Train*), models trained using five adversarial training baselines, and two based on regularization baselines. For all the models evaluated, their ASR is calculated by averaging the ASR obtained through three different attack methods: FSGS, OMPGS, and PCAA. Similarly, the metric m_f is computed by averaging its values across the same three attack methods. Although the metrics used for evaluation are all designed with the principle of "the smaller, the better," their values span different scales. Consequently, it is not practical to compare them based on absolute values. A more useful approach is to assess their agreement on the relative evaluation ranking. Hence, we examine the Pearson correlation coefficient between IGSG and m_f across all evaluated models, as well as between IGSG and ASR, to determine their consistency in evaluating model robustness. As shown in the bar plot to the right of Table 1, all correlation coefficients are greater than 0.65 (with their p-values being smaller than 0.05). This significant correlation emphasizes IGSG's effectiveness as a metric for evaluating robustness.

Robustness Evaluation Efficiency: Computational cost of IGSG (attack-free) vs. Attack-Based Methods. We provide computational complexity analysis of these evaluation methods in App.I.5. The computing time in Table 2 shows that IGSG is 17 to 240 times faster than OMPGS, demonstrating its superior efficiency.

Overlap between High-IG and High-Sensitivity Features. Feature sensitivity is a critical determinant of adversarial risk in categorical data (Bao et al., 2021). We thus compare the high-IG features with those identified as having high sensitivity on average, as reported by MLP models trained using all baseline models. Table 3 shows a significant overlap between the top-5 IG-ranked and sensitivity-ranked features. Specially, in the PEDec dataset, selecting the top-5 out of 5000 features to achieve an overlap ratio of 0.36 is notably high. The results indicate that the feature-wise IG score can serve as an effective measure of feature sensitivity, directly reflecting adversarial risk.

Table 3. Average overlap (%) of top-5 IG-ranked and sensitivity-ranked features.

Dataset	Overlap
Splice	0.82
PEDec	0.36
Census	0.58

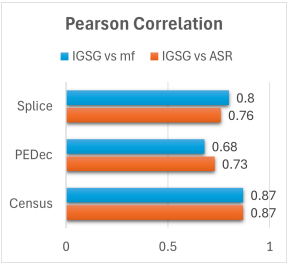
5.3. IGSG-based Robustness Enhancement

Implementation Details. To assess the enhancement in robustness provided by $IGSG_{\text{reg}}$ comparing other baselines, we utilize five attack methods to challenge all models. Due to the high computational complexity of FSGS (Bao et al., 2021), we set a fixed attack budget of 5 on all three datasets. For PCAA, FEAT, and GradAttack we also fix the attack budget to be 5. For OMPGS, we traverse varied attack budgets (the maximum number of modified features). Due to space limitations, we provide detailed attack settings in App.H.1 and App.H.4.

Performance metrics. The *adversarial accuracy* and *accuracy* are used for evaluating the performance of robustness enhancement.

Table 1. Robustness evaluation for MLP models trained in different ways. The consistency between IGSG (attack-free) and two other attack-based evaluation metrics (m_f-score and ASR) is shown by the Pearson correlation coefficients.

Dataset	Metric	Undefended	Adversarial Training baselines					Regularization baselines		**Ours**
		Std Train	Adv Train$_{L1}$	FastBAT$_{L1}$	TRADES$_{L1}$	AFD$_{L1}$	PAdvT	IGR	JR	**IGSG$_{\text{reg}}$**
Splice	$m_f \downarrow$	0.19	0.08	0.31	0.15	0.27	0.08	0.09	0.57	**-0.01**
	ASR $\downarrow$	0.53	0.47	0.6	0.51	0.59	0.51	0.47	0.77	**0.43**
	IGSG $\downarrow$	1.49	0.96	1.74	1.48	1.74	0.78	**0.31**	1.85	0.46
PEDec	$m_f \downarrow$	-0.28	-0.6	-0.69	-0.49	-0.78	-0.64	-0.52	-0.75	**-0.79**
	ASR $\downarrow$	0.32	0.14	0.09	0.21	0.07	0.12	0.18	**0.04**	**0.04**
	IGSG $\downarrow$	0.11	0.09	0.07	0.1	0.1	0.09	0.09	0.05	**0.04**
Census	$m_f \downarrow$	-0.56	**-0.71**	-0.7	0	-0.61	-0.51	-0.67	-0.62	-0.61
	ASR $\downarrow$	0.18	**0.14**	0.19	0.43	**0.14**	0.19	0.17	**0.14**	**0.14**
	IGSG $\downarrow$	0.08	0.07	0.07	0.15	0.09	0.12	0.09	**0.06**	**0.06**



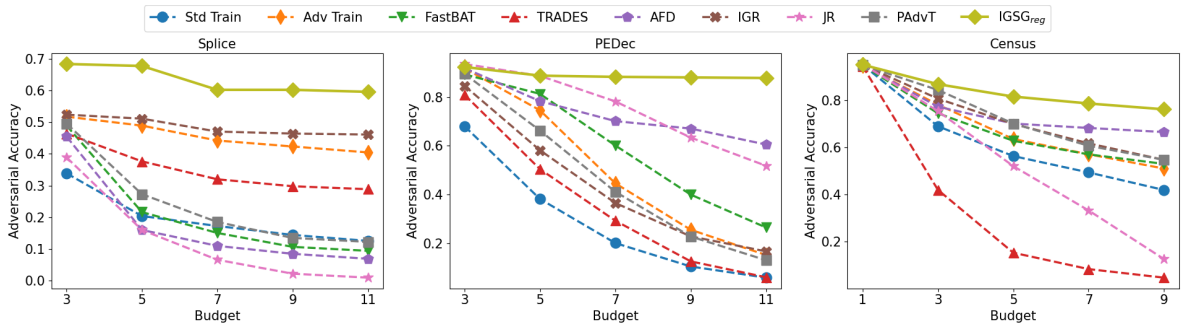


Figure 2. Adversarial accuracy of MLP models trained by *IGSG*$_{\text{reg}}$ and baselines under OMPGS attack with varied budgets.

Table 2. Time cost (s) for robustness evaluation (IGSG vs OMPGS)

Dataset	Models	IGSG	OMPGS	Speed up (OMPGS / IGSG)
Splice	MLP	0.6	18	30
	Transformer	1.8	31	17
PEDec	MLP	15	3600	240
	Transformer	13	1140	88
Census	MLP	32	1080	34
	Transformer	90	2700	30

Robustness Enhancement Performance: *IGSG*$_{\text{reg}}$ vs. Baseline Methods. Table 4 reports the performance of *IGSG*$_{\text{reg}}$ on improving the adversarial robustness of MLP models, compared to other baselines. Under FSGS attack, we can see that the adversarial accuracy of *IGSG*$_{\text{reg}}$ significantly outperforms the baseline methods. Especially, on *PEDec*, *IGSG*$_{\text{reg}}$ can largely improve the adversarial accuracy up to 86.5%. In comparison, the best baseline of robust training, *JR* and *AFD*, only achieves an adversarial accuracy score of 74.3%. In terms of adversarial accuracy under PCAA attack, *IGSG*$_{\text{reg}}$ excels on the *Splice* dataset and has the same level of adversarial accuracy on the *PEDec* and *Census* datasets compared to the baselines. For the adversarial accuracy under FEAT and GradAttack, *IGSG*$_{\text{reg}}$ demonstrates significantly higher adversarial accuracy on the *Splice* and *PEDec* datasets. It achieves comparable adversarial accuracy on the *Census* dataset relative to the baseline methods that perform best. Besides, when no attack is applied, *IGSG*$_{\text{reg}}$ achieves comparable accuracy with other baselines on the three datasets.

Fig.2 illustrates the adversarial accuracy of all the methods tested under OMPGS attacks with varying attack budgets. Higher attack budgets indicate stronger attacks against the targeted classifier, resulting in lower adversarial accuracy overall. Similar to the undefended model, most baseline methods experience a decline in adversarial accuracy as the attack strength increases. In contrast, the proposed *IGSG*$_{\text{reg}}$ consistently achieves higher and more stable levels of adversarial accuracy across all three datasets. Specifically, on *PEDec*, *IGSG*$_{\text{reg}}$ maintains an adversarial accuracy above 88% regardless of the attack strength. On *Splice*, *IGSG*$_{\text{reg}}$ consistently outperforms other baseline methods, exhibiting a performance gain of over 10%. On *Census*, *IGSG*$_{\text{reg}}$ initially shows similar adversarial accuracy to other baselines under small attack budgets but demonstrates a significantly slower rate of decline as the attack budget increases. Notably, adversarial training methods like *Adv Train* perform poorly on *PEDec*. This is because the feature space of *PEDec* is extensive, causing adversarial training to suffer from robust overfitting on categorical data. The attack can only explore a small fraction of all possible adversarial perturbations, limiting the effectiveness of adversarial training, while *IGSG*$_{\text{reg}}$ can provide consistently robust classification regardless of the feature dimensionality. *JR* performs well on *PEDec*, while the performance on *Splice* and *Census* is constantly bad. Using regularization as well, *IGSG*$_{\text{reg}}$ has a more stable performance on different datasets. It is worth noting that *Splice* has a few particularly sensitive features. Modifying these features can result in a change in whether a sample crosses an intron/exon or exon/intron boundary, or neither physically, which causes misclassification. Thus, all

Table 4. Adversarial Accuracy (%) and Accuracy (%) of MLP models trained by different ways (baselines and $IGSG_{\text{reg}}$) under FSGS, PCAA, FEAT, and GradAttack attack methods, and without attack (*Clean*)

Dataset	Attack	Undefended	Adversarial Training baselines					Regularization baselines		**Ours**
		Std Train	Adv Train$_{L1}$	Fast-BAT$_{L1}$	TRADES$_{L1}$	AFD$_{L1}$	PAdvT	IGR	JR	$\boldsymbol{IGSG_{\text{reg}}}$
Splice	FSGS	36.7±4.8	43.6±0.7	28.7±7.4	34.1±4.6	21.1±13.0	39.1±1.7	40.9±3.0	4.3±3.7	**44.0±2.6**
	PCAA	37.2±4.0	42.6±1.9	28.7±7.4	41.3±5.2	25.8±2.4	23.2±4.0	42.5±6.0	3.5±4.0	**44.9±2.0**
	FEAT	42.1±9.5	51.1±3.8	20.6±3.9	40.9±2.3	30.9±8.6	43.3±7.3	57.8±7.9	4.2±2.7	**59.4±3.0**
	GradAttack	68.7±2.1	85.7±4.9	26.9±4.5	64.3±6.1	14.9±7.4	50.8±9.5	85.6±6.1	2.7±2.3	**89.6±1.4**
	Clean	95.2±2.5	96.2±0.4	95.6±1.0	96.3±0.3	93.4±0.7	94.9±1.3	95.2±0.6	95.2±0.9	95.9±0.7
PEDec	FSGS	14.9±0.8	53.1±1.7	62.4±2.7	31.0±2.5	74.3±3.9	46.9±2.9	31.4±0.9	74.3±0.2	**86.5±3.8**
	PCAA	94.4±0.2	94.8±0.2	95.6±0.2	**95.8±0.2**	94.7±0.2	94.9±0.1	95.6±0.2	95.1±0.2	94.7±0.3
	FEAT	50.6±0.7	75.7±2.1	84.2±0.9	67.3±1.2	86.5±3.3	81.8±0.7	72.5±1.2	87.6±0.3	**88.6±3.2**
	GradAttack	21.2±1.2	50.8±1.3	63.6±2.2	30.9±2.2	74.4±6.3	49.1±1.1	39.7±2.7	80.3±0.8	**85.6±2.7**
	Clean	96.4±0.2	96.2±0.0	96.2±0.1	96.4±0.1	96.0±0.2	96.5±0.3	96.4±0.0	95.4±0.1	95.5±0.2
Census	FSGS	46.2±1.8	54.1±2.3	63.4±3.8	49.8±1.6	60.2±1.9	61.9±5.4	45.8±1.7	48.3±3.4	**67.2±3.5**
	PCAA	92.0±0.7	**93.9±0.1**	93.1±0.7	88.8±0.8	93.2±0.1	93.4±0.4	93.6±0.0	93.4±0.1	93.8±0.0
	FEAT	76.3±2.8	88.1±2.1	49.6±1.9	81.9±4.6	77.6±0.2	76.7±2.7	**88.7±0.3**	73.2±3.6	88.4±1.9
	GradAttack	63.2±1.3	64.7±1.6	67.2±1.6	56.3±1.3	66.7±2.4	63.8±2.5	**67.4±2.1**	64.4±1.2	**67.4±1.3**
	Clean	95.4±0.1	94.5±0.3	95.0±0.1	94.8±0.3	95.2±0.2	95.2±0.1	95.3±0.1	95.4±0.1	95.5±0.2

the defense methods involved in the test do not perform well against attacks on *Splice*. In conclusion, $IGSG_{\text{reg}}$ demonstrates resistance against a wide range of previously unseen attacks, thereby achieving broad-spectrum robustness.

Robustness of models trained by $\boldsymbol{IGSG_{\text{reg}}}$ VS. baselines, measured by IGSG, ASR and m_f. In Table 1, the robustness of MLP models trained by $IGSG_{\text{reg}}$ is compared with those trained by baselines based on attack-free (IGSG), and attack-based metrics (ASR and m_f). The results show that $IGSG_{\text{reg}}$ outperforms baseline methods across all three evaluated metrics. The m_f-score quantifies the mean distance between adversarial samples and the classification boundary, where a lower m_f score indicates a greater distance, implying enhanced robustness. For *Splice* and *PEDec*, $IGSG_{\text{reg}}$ records the lowest m_f scores, aligning with its superior performance on other metrics. In the case of *Census*, although $IGSG_{\text{reg}}$ exhibits a higher m_f score compared to some baselines, it achieves the lowest ASR. Notably, $IGSG_{\text{reg}}$ significantly lowers the IGSG scores, particularly in datasets with highly sensitive features, like *Splice*.

Ablation Study. We include the following variants of the proposed $IGSG_{\text{reg}}$ method in the ablation study. $\boldsymbol{SG_{\text{reg}}}$ and $\boldsymbol{IG_{\text{reg}}}$ are designed to preserve only the smoothed gradient-based (SG_{reg}, see Eq.5) or the IG-based smoothness regularization (IG_{reg}, see Eq.4) respectively in the learning objective. We compare SG_{reg} and IG_{reg} to $IGSG_{\text{reg}}$ for demonstrating the advantage of simultaneously performing the IG and SG regularization. In $\boldsymbol{IGSG\text{-}VG_{\text{reg}}}$: we replace SG given in Eq.5 with the vanilla gradient of the one hot tensor. Another four variants to provide additional validation for the design of $IGSG_{\text{reg}}$ are presented in App.I.6.

Table 5 shows that $IGSG_{\text{reg}}$ consistently outperforms the variants in adversarial accuracy against both FSGS and OMPGS attacks, affirming the effectiveness of $IGSG_{\text{reg}}$'s design in mitigating both types of greedy search-based attacks simultaneously. SG_{reg} does not employ IG-based regularization, resulting in a classifier that may overly rely on a few highly influential features contributing most to the classification output. These sensitive features can be readily targeted by both types of greedy search-based attacks, particularly on *PEDec*. In comparison, IG_{reg} lacks the classification boundary smoothness, leading to a slight decrease in performance compared to $IGSG_{\text{reg}}$. The results with *SG* and *IG* show that the two attributional smoothness regularization terms employed by $IGSG_{\text{reg}}$ are complementary to each other in improving the adversarial robustness of the built model.

Table 5. Ablation Study. Adversarial Accuracy and Accuracy (%) for $IGSG_{\text{reg}}$ variants with an attack budget of 5.

Dataset	Attack	SG_{reg}	IG_{reg}	$IGSG\text{-}VG_{\text{reg}}$	$\boldsymbol{IGSG_{\text{reg}}}$
Splice	FSGS	43.3±3.0	40.3±5.0	39.7±2.4	**44.0±2.6**
	OMPGS	59.9±6.5	54.9±4.9	59.4±5.3	**63.8±4.2**
	Clean	95.7±0.5	94.7±1.0	95.2±1.1	95.9±0.7
PEDec	FSGS	12.7±1.8	84.2±2.9	81.6±3.8	**86.5±3.8**
	OMPGS	28.6±1.1	83.4±7.6	82.3±3.5	**88.0±4.0**
	Clean	96.4±0.1	94.8±0.3	95.2±0.2	95.5±0.2
Census	FSGS	47.9±2.1	57.8±0.8	54.1±1.6	**67.2±3.5**
	OMPGS	**71.4±7.8**	65.9±2.7	69.3±6.4	71.3±9.0
	Clean	95.1±0.3	95.5±0.1	95.4±0.0	95.5±0.2

Table 6. Performance comparison: gains of $IGSG_{\text{reg}}$ versus PGD-based adversarial training in MLP

Dataset	Attack	Adv. Acc.	Gain
Splice	PGD-1	95.6%	0.4% ∼
	OMPGS	63.8%	12.1% ↑
	FSGS	44.0%	0.4% ∼
PEDec	PGD-1	94.5%	-1.5% ∼
	OMPGS	88.0%	13.9% ↑
	FSGS	86.5%	34.0% ↑
Census	PGD-1	93.0%	-0.2% ∼
	OMPGS	71.3%	8.6% ↑
	FSGS	67.2%	13.1% ↑

$IGSG\text{-}VG_{\text{reg}}$ replaces the smoothed gradient-based regu-

larization defined in Eq.5 with a vanilla gradient. Its diminished performance shows the merit of introducing the smoothed gradient computing and the mean-field smoothing-based technique in Eq.5.

Effectiveness of Avoiding Robust Overfitting. By utilizing regularization, $IGSG_{\text{reg}}$ avoids the issue of "robust overfitting" encountered in adversarial training. This leads to improved performance, as demonstrated in Table 6, compared to the adversarial accuracy of PGD-based adversarial training. We conduct the comparison between $IGSG_{\text{reg}}$ and two works mitigating robust overfitting in the continuous domain (Chen et al., 2020; Yu et al., 2022). $IGSG_{\text{reg}}$ achieves consistently better adversarial robustness. The details are presented in App.I.4.

Additional Experimental Results. Due to space constraints, several experiments are detailed in the appendix. Classification boundary visualizations can be found in App.5. Experimental results related to Transformers are deferred to App.I.3. Time complexity analysis is available in App.I.5. The hyper-parameter sensitivity analysis is conducted in App.I.8. The adaptive attack is conducted in App.I.7.

6. Conclusion

Our work proposes an attack-free robustness evaluation method, namely IGSG, as a metric of the resilience of any arbitrary deep learning model against adversarial attacks on categorical data. The core idea is to integrate both the integrated gradient and smoothed gradient of the model to measure the adversarial risk level, yet avoiding the computationally intensive generation of adversarial samples with categorical inputs. We provide both theoretical and empirical rationality behind this metric. Furthermore, we propose to use the IGSG-based metric as an optimizable objective of robust training. We demonstrate the domain-agnostic use of $IGSG_{\text{reg}}$ across different real-world applications. In our future study, we will extend the proposed method to the text classification task and compare it with text-specific robust training methods enhanced with semantic similarity knowledge.

Impact Statement

This paper aims to advance the field of Adversarial Machine Learning by evaluating and enhancing adversarial robustness for categorical data. The research introduces the Integrated Gradient-Smoothed Gradient (IGSG) metric and an IGSG-based regularization technique, providing an efficient, attack-free method to evaluate and improve classifier robustness. While IGSG is generally effective on real-world datasets, its regularization may not be beneficial in certain extreme cases. For instance, if a dataset relies solely on a single feature with all other features being irrelevant, or if all features contribute equally across the dataset, IGSG-based regularization may not enhance model robustness for these datasets. However, these cases hardly appear in real-world applications.

For ethical concerns, this research enhances the security of AI applications by improving adversarial robustness, particularly for safety-critical systems in fields like healthcare. It aims to fortify machine learning models against attacks, enhancing reliability without introducing new vulnerabilities or biases.

Acknowledgement

We would like to thank the anonymous reviewers for their constructive feedback. This work has benefited from the Notre Dame-IBM Technology Ethics Lab, and the government grant managed by the French National Research Agency with the reference ANR-23-IAS4-0001 (CKRISP).

References

Andriushchenko, M. and Flammarion, N. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Asadi, A. R., Abbe, E., and Verdú, S. Chaining mutual information and tightening generalization bounds. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 7245–7254, Red Hook, NY, USA, 2018. Curran Associates Inc.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Bao, H., Han, Y., Zhou, Y., Shen, Y., and Zhang, X. Towards understanding the robustness against evasion attack on categorical data. In *International Conference on Learning Representations*, 2021.

Bao, H., Han, Y., Zhou, Y., Gao, X., and Zhang, X. Towards efficient and domain-agnostic evasion attack with high-dimensional categorical inputs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6753–6761, 2023.

Bashivan, P., Bayat, R., Ibrahim, A., Ahuja, K., Faramarzi, M., Laleh, T., Richards, B., and Rish, I. Adversarial feature desensitization. *Advances in Neural Information Processing Systems*, 34:10665–10677, 2021.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Brekelmans, R., Huang, S., Ghassemi, M., Steeg, G. V., Grosse, R., and Makhzani, A. Improving mutual information estimation with annealed and energy-based bounds. *arXiv preprint arXiv:2303.06992*, 2023.

Bu, Y., Zou, S., and Veeravalli, V. V. Tightening mutual information based bounds on generalization error. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 587–591, 2019. doi: 10.1109/ISIT.2019.8849590.

Campbell, J. E., Carmichael, G. R., Chai, T., Mena-Carrasco, M., Tang, Y., Blake, D., Blake, N., Vay, S. A., Collatz, G. J., Baker, I., et al. Photosynthetic control of atmospheric carbonyl sulfide during the growing season. *Science*, 322(5904):1085–1088, 2008.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Chambolle, A. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1–2):89–97, jan 2004. ISSN 0924-9907.

Chen, J., Wu, X., Rastogi, V., Liang, Y., and Jha, S. Robust attribution regularization. *Advances in Neural Information Processing Systems*, 32, 2019.

Chen, T., Zhang, Z., Liu, S., Chang, S., and Wang, Z. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., and Cohen, I. Pearson correlation coefficient. *Noise reduction in speech processing*, pp. 1–4, 2009.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley, 2005.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Elenberg, E. R., Khanna, R., Dimakis, A. G., and Negahban, S. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568, 2018.

Finlay, C. and Oberman, A. M. Scaleable input gradient regularization for adversarial robustness. *Machine Learning with Applications*, 3:100017, 2021.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Grosse, K., Papernot, N., Manoharan, P., Backes, M., and McDaniel, P. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*, 2016.

Hannun, A. Y., Guo, C., and van der Maaten, L. Measuring data leakage in machine-learning models with fisher information. In *Conference on Uncertainty in Artificial Intelligence*, 2021. URL https://api.semanticscholar.org/CorpusID:232013768.

Hoffman, J., Roberts, D. A., and Yaida, S. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

Huang, Z., Fan, Y., Liu, C., Zhang, W., Zhang, Y., Salzmann, M., Süsstrunk, S., and Wang, J. Fast adversarial training with adaptive step size. *IEEE Transactions on Image Processing*, 2023.

Jakubovitz, D. and Giryes, R. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 514–529, 2018.

Kim, H., Park, J., Choi, Y., and Lee, J. Fantastic robustness measures: The secrets of robust generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

Lane, T. and Kohavi, R. UCI census-income (kdd) data set, 2000. URL https://archive.ics.uci.edu/ml/datasets/Census-Income+(KDD).

Lee, G.-H., Yuan, Y., Chang, S., and Jaakkola, T. Tight certificates of adversarial robustness for randomly smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Lee, J. and Leyffer, S. *Mixed integer nonlinear programming*, volume 154. Springer Science & Business Media, 2011.

Lei, Q., Wu, L., Chen, P.-Y., Dimakis, A., Dhillon, I. S., and Witbrock, M. J. Discrete adversarial attacks and submodular optimization with applications to text classification. *Proceedings of Machine Learning and Systems*, 1:146–165, 2019.

Li, Z., Xu, J., Zeng, J., Li, L., Zheng, X., Zhang, Q., Chang, K.-W., and Hsieh, C.-J. Searching for an effective defender: Benchmarking defense against adversarial word substitution. *arXiv preprint arXiv:2108.12777*, 2021.

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

McAllester, D. Some pac-bayesian theorems. *Machine Learning*, 37:355–363, 1999.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Noordewier, M., Towell, G., and Shavlik, J. Training knowledge-based neural networks to recognize genes in dna sequences. *Advances in neural information processing systems*, 3, 1990.

Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Ross, A. and Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Sarkar, A., Sarkar, A., and Balasubramanian, V. N. Enhanced regularizers for attributional robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2532–2540, 2021.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

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

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Y., Han, Y., Bao, H., Shen, Y., Ma, F., Li, J., and Zhang, X. Attackability characterization of adversarial evasion attack on discrete data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1415–1425, 2020.

Wei, X.-X. and Stocker, A. A. Mutual information, fisher information, and efficient coding. *Neural computation*, 28(2):305–326, 2016.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Xu, A. and Raginsky, M. Information-theoretic analysis of generalization capability of learning algorithms. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/ad71c82b22f4f65b9398f76d8be4c615-Paper.pdf.

Xu, H., He, P., Ren, J., Wan, Y., Liu, Z., Liu, H., and Tang, J. Probabilistic categorical adversarial attack and adversarial training. In *International Conference on Machine Learning*, pp. 38428–38442. PMLR, 2023.

Yang, Z., Han, Y., and Zhang, X. Attack transferability characterization for adversarially robust multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 397–413. Springer, 2021.

Yu, C., Han, B., Shen, L., Yu, J., Gong, C., Gong, M., and Liu, T. Understanding robust overfitting of adversarial training and beyond. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25595–25610. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/yu22b.html.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. Macer: Attack-free and

scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Zhang, J., Liu, T., and Tao, D. An optimal transport analysis on generalization in deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2021. doi: 10.1109/TNNLS.2021.3109942.

Zhang, Y., Zhang, G., Khanduri, P., Hong, M., Chang, S., and Liu, S. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pp. 26693–26712. PMLR, 2022.

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., and Liu, J. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*, 2019.

A. Proof to Theorem 3.2

Definition A.1. Diameter of f: Assuming that the hypothesis space $\mathcal{H}$ is a bounded Banach space, the diameter of $f \in \mathcal{H}$ is defined as:

$$\mathcal{D}_f = \sup_{f,f' \in \mathcal{H}} d(f, f') \tag{9}$$

where d is the distance metric of $\mathcal{H}$.

Definition A.2. Lipschitz continuousity of ℓ: Assuming that $\ell(f(x_i), y_i)$ is L-Lipschitz for any $z_i = (x_i, y_i)$, the following inequality holds for any f and f' in $\mathcal{H}$:

$$|\ell(f(x_i), y_i) - \ell(f'(x_i), y_i))| \leq L\, d(f, f') \tag{10}$$

Proof to Eq.7: Given μ_z and a classifier f trained using S^n, we assume the distribution of the worst-case adversarial samples of f as $\hat{\mu}_z$, determined by μ_z and f jointly. Any worst-case adversarial sample $\hat{z}_i$ derived by solving the loss maximization problem $\underset{\text{diff}(\hat{z}_i, z_i) \leq \epsilon}{\arg} \max \ell(f(x_i), y_i)$ can be thus considered as a sample from $\hat{\mu}_{\hat{z}}$. We can then extend the Total Variation (TV) distance-based generalization bound of f, which is established by Theorem 2 in (Zhang et al., 2021) as below:

$$\mathrm{E}_f[\mathcal{R}_f^{adv}] \leq \mathrm{E}_f[\hat{\mathcal{R}}_f^{adv}] + L\, \mathcal{D}_f\, \mathbb{TV}(P_f \times \hat{\mu}_{\hat{z}}, P_{f \times \hat{z}_i}) \tag{11}$$

where $\mathbb{TV}(\cdot, \cdot)$ denotes the Total Variation distance between two probabilistic distribution. P_f and $\hat{\mu}_{\hat{z}}$ are the marginal distribution of f and the worst-case adversarial sample $\hat{z}_i$. $P_{f \times \hat{z}_i}$ denotes the joint distribution of f and $\hat{z}_i$.

Pinsker's inequality in information theory (Cover & Thomas, 2005) gives further the upper bound of the Total-Variation distance: $\mathbb{TV}(P_f \times \hat{\mu}_{\hat{z}}, P_{f \times \hat{z}_i}) \leq \sqrt{\frac{D_{KL}(P_{f,\hat{z}_i}, P_f \times P_{\hat{z}_i})}{2}} = \sqrt{\frac{I(f, \hat{z}_i)}{2}}$, where D_{KL} is the KL divergence between the two probabilistic distributions. Based on this, we can further formulate Eq.11 by letting $z = z_i$ (i=1,2,3,...,n) and using mutual information between f and $\hat{z}_i$:

$$\begin{aligned} \mathrm{E}_f[\mathcal{R}_f^{adv}] &\leq \mathrm{E}_f[\hat{\mathcal{R}}_f^{adv}] + \frac{L\, \mathcal{D}_f}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} I(f; \hat{z}_i)} \\ &\leq \mathrm{E}_f[\hat{\mathcal{R}}_f^{adv}] + \frac{L\, \mathcal{D}_f}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} I(f; z_i) + \sum_{i=1}^{n} (I(f; \hat{z}_i) - I(f; z_i))} \end{aligned} \tag{12}$$

where $\{z_i = (x_i, y_i)\} \in S^n$ are statistically independent training samples and $\hat{z}_i$ the corresponding worst-case adversarial sample. We can extend $I(f; \hat{z}_i) - I(f; z_i)$ as below. In this study, we only consider feature perturbation and exclude label-flipping attacks from the proposed attack scenario. We first split $\hat{z}_i = (\hat{x}_i, y_i)$ and $z_i = (x_i, y_i)$ into $\hat{z}_i = (\hat{x}_{i,\omega_i}, x_{i,\overline{\omega}_i}, y_i)$ and $\hat{z}_i = (\hat{x}_{i,\omega_i}, x_{i,\overline{\omega}_i}, y_i)$ respectively. Since features in $\overline{\omega}_i$ remain untouched in the attack, we use the same notation of these unmodified features in $\hat{z}_i$ and z_i.

$$
\begin{aligned}
& I(f;\hat{z}_i) - I(f;z_i) \\
= \; & I(f;\hat{x}_{i,\omega_i}, x_{i,\overline{\omega}_i}, y_i) - I(f;x_{i,\omega_i}, x_{i,\overline{\omega}_i}, y_i) \\
= \; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + I(\hat{x}_{i,\omega_i}; f|x_{i,\overline{\omega}_i}, y_i) - I(x_{i,\overline{\omega}_i}, y_i; f|x_{i,\omega_i}) \\
= \; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + H(\hat{x}_{i,\omega_i}|x_{i,\overline{\omega}_i}, y_i) + H(f|x_{i,\overline{\omega}_i}, y_i) - H(\hat{x}_{i,\omega_i}, f|x_{i,\overline{\omega}_i}, y_i) \\
& - H(x_{i,\overline{\omega}_i}, y_i|x_{i,\omega_i}) - H(f|x_{i,\omega_i}) + H(x_{i,\overline{\omega}_i}, y_i, f|x_{i,\omega_i}) \\
= \; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + H(\hat{x}_{i,\omega_i}) - I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) + H(f) - I(x_{i,\overline{\omega}_i}, y_i; f) \\
& - H(x_{i,\overline{\omega}_i}, y_i) + I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) - H(f) + I(x_{i,\omega_i}; f) \\
& - H(\hat{x}_{i,\omega_i}, f|x_{i,\overline{\omega}_i}, y_i) + H(x_{i,\overline{\omega}_i}, y_i, f|x_{i,\omega_i}) \\
= \; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + H(\hat{x}_{i,\omega_i}) - I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) + H(f) - I(x_{i,\overline{\omega}_i}, y_i; f) \\
& - H(x_{i,\overline{\omega}_i}, y_i) + I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) - H(f) + I(x_{i,\omega_i}; f) \\
& - H(f|x_{i,\overline{\omega}_i}) - H(\hat{x}_{i,\omega_i}|x_{i,\overline{\omega}_i}, f) + H(x_{i,\overline{\omega}_i}, f|x_{i,\omega_i}) \\
\leq \; & 2|I(x_{i,\omega_i}; f) - I(x_{i,\overline{\omega}_i}, y_i; f)| + |I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i, f) \\
& - I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i, f)| + |I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) - I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i)|
\end{aligned}
\tag{13}
$$

where $H(X|Y)$ and $I(X;Y|Z)$ denotes the conditional entropy of a random variable X given the other random variable Y and the conditional mutual information between X and Y given another random variable Z. By introducing $\alpha = \max_{z_i=(x_i,y_i)\in S^n} 1 + \frac{|I(\hat{x}_{i,\omega_i};x_{i,\overline{\omega_i}},y_i)-I(x_{i,\omega_i};x_{i,\overline{\omega_i}},y_i)|}{|I(\hat{x}_{i,\omega_i};x_{i,\overline{\omega_i}},y_i,f)-I(x_{i,\omega_i};x_{i,\overline{\omega_i}},y_i,f)|}$ to Eq.13, we can derive Eq.7.

We discuss about the tightness of the bound in Eq.7 from the following perspectives. First, we show this bound reduces to an individual sample-based upper bound of the generalization error of f in the adversary-free case. It converges to zero when $n \to \infty$ with the same speed as that established in Proposition.1 of (Bu et al., 2019). This bound enjoys a close level of tightness *in the adversary-free scenario* as that proposed in (Bu et al., 2019).

We first give the definition of the expected and empirical risk under the adversary-free setting, following Definition.1.

Definition A.3. Following (Xu & Raginsky, 2017; Asadi et al., 2018), given a training dataset S^n composed of n i.i.d training samples $z_i \sim \mu$, we assume a randomized learning paradigm $\mathcal{A}$ mapping S^n to a hypothesis f, i.e., $f = \mathcal{A}(S^n)$, according to a conditional distribution $P_{f|S^n}$. The expected classification risk of f under the adversary-free scenario, noted as $\mathcal{R}_f$, given in Eq.14. The expectation is taken over the distribution of the n training samples S^n and the classifier $f = \mathcal{A}(S^n)$.

$$
\mathcal{R}_f = \mathop{\mathbb{E}}_{S^n, P_{f|S^n}} \mathop{\mathbb{E}}_{z=(x,y)\sim\mu_z,} \ell(f(x), y). \tag{14}
$$

Similarly, we provide the empirical risk of f under the adversary-free scenario in Eq.15. It is taken as the expectation over the distribution of the n training samples and the classifier.

$$
\hat{\mathcal{R}}_f = \mathop{\mathbb{E}}_{S^n, P_{f|S^n}} \frac{1}{n} \sum_{z_i=(x_i,y_i)\in S^n} \ell(f(x_i), y_i) \tag{15}
$$

With the adversary-free setting, $\hat{x} = x$. This makes $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ vanish as $I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i, f) = I(x_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i, f)$. Similarly, $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) = |I(x_{i,\omega_i}; f) - I(x_{i,\overline{\omega_i}}, y_i; f)|$ is reduced to $I(z_i; f)$, since $\omega_i = \emptyset$ for each training sample z_i. As a result, the bound given in Eq.7 shrinks to the following form in Eq.16:

$$
\mathcal{R}_f - \hat{\mathcal{R}}_f \leq \frac{\sqrt{3}\, L\, \mathcal{D}_f}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} I(f; z_i)}. \tag{16}
$$

where $\mathcal{R}_f$ and $\hat{\mathcal{R}}_f$ are expected and empirical risk under the adversary-free setting. In comparison, Proposition.1 (Eq.19 and 20) in (Bu et al., 2019) provides the upper bound of the generalization error of f in a similar form:

$$
\mathcal{R}_f - \hat{\mathcal{R}}_f \leq \frac{1}{n} \sum_{i=1}^{n} \sqrt{2R^2 I(f; z_i)}. \tag{17}
$$

with the condition that the loss function $\ell(f, z)$ is R-sub-Gaussian under $z \sim \mu_z$ for all $f \in \mathcal{H}$. We can find that the two adversary-free bounds in Eq.16 and Eq.17 only differ in the scaling constant. When n (the number of training samples) goes to infinity, both bounds vanish with the same convergence speed. Compared to the training set mutual information $I(f; S^n)$ based bound proposed Theorem.1 of (Xu & Raginsky, 2017), the individual sample mutual information-based bound (Eq.16 and Eq.17) poses a tighter bound over the generalization error according to the theoretical and empirical analysis conducted in (Bu et al., 2019). In (Xu & Raginsky, 2017), the information-theoretic bound is built by assuming that the loss function $\ell(f, z)$ has a bounded cumulative generating function with $z \sim \mu_z$ and $f \in \mathcal{H}$. Nevertheless, this assumption does not necessarily hold. Our study thus avoids this shortcoming and adopts the individual sample mutual information to develop the adversarial risk analysis. In conclusion, we develop theoretical analysis under a more general condition about the cumulative generating function of the loss function compared to (Xu & Raginsky, 2017), which makes our work applicable to a broad range of problems.

Second, The value of Eq.7 is bounded. The possible value of $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i}) = |I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i, f) - I(x_{i,\omega_i}; x_{i,\overline{\omega_i}}, y_i, f)|$ and $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) = |I(x_{i,\omega_i}; f) - I(x_{i,\overline{\omega_i}}, y_i; f)|$ follow the constraint that:

$$\begin{aligned} \Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i}) &\leq \log(q\epsilon) \\ \Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) &\leq I(z_i; f) \end{aligned} \tag{18}$$

where the maximum cardinality of any single feature in the feature subset ω_i is denoted as q. ϵ is the maximum number of features that the attacker may perturb, a.k.a the attack budget. the number of the features in ω_i, noted as $|\omega_i|$ is no more than ϵ. With this constraint, the value of Eq.7 is bounded from above as:

$$\begin{aligned} \mathcal{R}_f^{adv} - \hat{\mathcal{R}}_f^{adv} &\leq \frac{L\,\mathcal{D}_f}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} I(f; z_i) + 2\sum_{i=1}^{n} \Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) + \sum_{i=1}^{n} \Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})} \\ &\leq \frac{L\,\mathcal{D}_f}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} 3I(f; z_i) + n\log(q\epsilon)} \end{aligned} \tag{19}$$

In Eq.19, the first term under the squared root symbol is $\sum_{i=1}^{n} 3I(f; z_i)$. It measures the generalization error under the adversary-free setting according to Eq.16. The second term $\log(q\epsilon)$ measures the strength of the attack by considering the cardinality of the feature subset ω_i. A higher cardinality $\log(q\epsilon)$ implies a larger combinatorial set of possible categorical feature values available to the attacker (more features that the attacker may perturb and/or more category values per feature that the attacker may choose to replace the original feature value). The attacker selects one set of categorical values in this combinatorial set to replace the original feature values within the feature subset ω_i, in order to deliver the adversarial attack. Consequently, a higher cardinality indicates greater flexibility to organize feature manipulation over ω_i, which signifies a stronger attack and thereby elevates the adversarial risk. Eq.19 gives a bounded but rough estimate of the adversarial risk, as not all of the features are useful for the attack. Only the perturbation over influential features may effectively cause the rise of adversarial risk. In this sense, Eq.7 provides a more accurate estimate of the actual adversarial risk than Eq.19.

B. The Connection between Theorem 3.2 and the Practical Design of the IGSG-based Robust Training

With the proposed Theorem 3.2, we aim to find an algorithm that can minimize the upper bound defined in Eq.7. However, directly optimizing the mutual information terms proves to be non-trivial given high-dimensional inputs (Brekelmans et al., 2023). We thus pursue an approximation method to reach this objective in the algorithmic design. We find that SG regularization is derived from minimizing $I(f, z_i)$ (one mutual information term in Eq.7 of Theorem 3.2.), and the SG and IG regularization suppresses $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$, two other terms in Eq.7. We next discuss in detail the questions regarding the connection between them and the rationale behind the design.

B.1. How Mutual Information Mathematically Connects with IG and SG?

The link of IG and SG with the mutual information is derived from the fact that the magnitude of the gradient of the classifier f with respect to the input feature x_{i,ω_i}, i.e. $\frac{\partial \log(f_{y_i}(x_i))}{\partial x_{i,\omega_i}}$, is proportional to the value of the mutual information between the

feature x_{i,ω_i} and the model's classification output $f_{y_i}(x_i)$ for a specific class y_i, noted as $I(x_{i,\omega_i}; f)$. Our analysis stems from (Wei & Stocker, 2016), which shows that the Fisher information I_{fisher} serves as a close approximation to mutual information.

Specifically in our case, we have the upper bound of the mutual information $I(x_{i,\omega_i}; f)$ based on the gradient term $\frac{\partial \log(f_{y_i}(x_i))}{\partial x_{i,\omega_i}}$ (Hannun et al., 2021) : $I(x_{i,\omega_i}; f) \approx I_{\text{fisher}} = \int \left(\frac{\partial \log(p(f_{y_i}(x_i)|x_{i,\omega_i}))}{\partial x_{i,\omega_i}}\right)^2 p(f_{y_i}(x_i)|x_{i,\omega_i}) df_{y_i}(x_i) \leq \sup_{f_{y_i}(x_i)} (\frac{\partial \log(f_{y_i}(x_i))}{\partial x_{i,\omega_i}})^2$

B.1.1. Enforcing the SG Regularization Suppresses $I(f, z_i)$, thus Reducing the Adversarial Risk Bound

Performing the SG regularization (as in Eq.5) penalizes the magnitude of the gradient term $\frac{\partial \log(f_{y_i})}{\partial x_i}$. With this revealed relation between the mutual information term and the gradient, the SG regularization thus effectively suppresses the mutual information $I(f, z_i = (x_i, y_i))$. Consequently, with a diminishing mutual information value of $I(f, z_i)$, the adversarial risk bound is reduced. As reported by previous works, reducing $I(f, z_i)$ improves the generalization capability of the classifier in the adversary-free scenario and boosts its resilience to adversarial attacks.

B.1.2. Enforcing the SG and IG Regularization Suppresses $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega}_i})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$, thus Reducing the Adversarial Risk Bound

Building upon the insight above, we introduce Integrated Gradient (IG), defined based on the gradient term $\frac{\partial \log(f_{y_i}(x_i))}{\partial x_{i,\omega_i}}$, to measure the contribution of the feature x_{i,ω_i} to the model's classification output $f_{y_i}(x_i)$. In the discrete feature case, the Integrated Gradient (IG) can be expressed as:

$IG(\omega_i) = \int_{b_{\omega_i}=0}^{1} \frac{\partial \log(f_{y_i})}{\partial b_{\omega_i}} db_{\omega_i}$

Here, b_{ω_i} represents the binary indicator corresponding to the categorical feature x_{i,ω_i}, as introduced in the preliminary of Section 3. Performing the TV loss regularization over the IG of different categorical features (as in Eq.4) is to align the gradient with respect to the feature ω_i (noted as $\frac{\partial \log(f_{y_i})}{\partial b_{\omega_i}}$) to that with respect to another feature $\overline{\omega}_i$ other than ω_i (noted as $\frac{\partial \log(f_{y_i})}{\partial b_{\overline{\omega}_i}}$). Therefore, enforcing the TV loss regularization reduces the gap between the mutual information $I(x_{i,\omega_i}; f)$ and that of the other features $I(x_{i,\overline{\omega}_i}; f)$, which in turn reduces the value of $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega}_i})$.

Similarly, performing the SG-based regularization (as in Eq.5) penalizes the magnitude of the gradient term $\frac{\partial \log(f_{y_i})}{\partial b_{\omega_i}}$. This regularization thus effectively suppresses the value of $I(x_{i,\omega_i}; f)$. Consequently, with a reduced mutual information value of $I(x_{i,\omega_i}; f)$, injecting adversarial perturbations to the feature ω_i (modifying x_i to $\hat{x}_i$) then does not trigger large difference between $I(x_{i,\omega_i}; f)$ and $I(\hat{x}_{i,\omega_i}; f)$. This in turn reduces the value of $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$.

In summary, the proposed IGSG method combines the TV loss and SG-based loss, as seen in Eq.6, to minimize $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega}_i})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$, which suppresses the mutual information-based adversarial risk bound in Theorem 3.2.

C. Difference between PAC-Bayes Bounds and Our Study

Following (Xu & Raginsky, 2017; Bu et al., 2019), we don't impose any prior distribution assumption over $P_{f|S^n}$. This characterizes the major difference between our study and PAC-Bayes generalization bounds (McAllester, 1999). Though PAC-Bayesian bounds also connect information-theoretic quantities to generalization and are similar to the mutual information approach, these bounds are usually output-dependent. That is, they give a generalization bound for a particular output hypothesis or hypothesis distribution, rather than uniformly bounding the expected error of the algorithm as in the mutual-information-based bound in our study. We adopt the mutual-information-based technique to exploit the fact that the generalization error depends strongly not only on the underlying true data-generating distribution but also on the correlation between the collection of empirical risks of the available hypotheses and the final output of the learning algorithm.

D. Discussion about the Randomized Learning Mechanism

It is worth noting that our information-theoretic analysis is rooted in the research of mutual-information-based generalization

error analysis in (Xu & Raginsky, 2017; Bu et al., 2019). This line of inquiry adopts an information-theoretic perspective to enhance the generalization capabilities of machine learning algorithms. Within this theoretical framework, a model training algorithm is conceptualized as a randomized mapping or an information-transmitting channel, employing the language of information theory. This mapping or channel takes a training dataset as input and yields a hypothesis as output. The randomness inherent in this mapping/channel manifests in two dimensions. First, the training dataset provided to the channel is a sample selected from all possible combinations of n training data points. Second, the resulting hypothesis from this channel is one sample chosen from the set of possible hypotheses within the hypothesis space. The mutual information-based bound in Eq.7 thus determines the expected adversarial risk over all possible hypothesis functions in the hypothesis space. In other words, we offer an averaged estimate of the potential adversarial risk, irrespective of the hypothesis chosen as the output by the learning algorithm. In this sense, for a classifier used in a concrete learning task, whether the parameters/decision outputs of this classifier are deterministic or randomized, our mutual-information-based bound is applicable.

E. Limitations of Adversarial Training on Categorical Data

To evaluate the limitations of adversarial training on categorical data. We implement f as a Multilayer Perceptron (MLP) and conduct PGD-based adversarial training on it across three datasets. Subsequently, the resistance of f to three evasion attacks is outlined in Table 7. With the attack budget 5 (i.e., $\mathit{diff}(x_i, \hat{x}_i) \leq 5$), both Forward Stepwise Greedy Search (FSGS) (Elenberg et al., 2018), and orthogonal matching pursuit based greedy search (OMPGS) (Wang et al., 2020) can directly find attack samples $\hat{x}_i$. PGD attack in the 1-norm setting (PGD-1) (Madry et al., 2017) locates attack samples and subsequently discretizes them to yield feasible adversarial samples $\hat{x}_i$. Table 7 shows that the adversarially trained f is only resilient against the PGD-1-based attack (high adversarial accuracy), remaining vulnerable facing the other two attacks (significantly lower adversarial accuracy). This suggests that the PGD-based adversarial training may not account for all possible adversarial samples, causing the model to overfit to the samples discovered by the PGD method.

Table 7. MLP with PGD-based adversarial training

Dataset	Attack	Adv. Acc.	Defend
Splice	PGD-1	95.2%	✓
	OMPGS	51.7%	×
	FSGS	43.6%	×
PEDec	PGD-1	96.0%	✓
	OMPGS	74.1%	×
	FSGS	52.5%	×
Census	PGD-1	93.2%	✓
	OMPGS	62.7%	×
	FSGS	54.1%	×

Similar observations can be made for f when using OMPGS-based adversarial training (see Fig.3 in App.F). For the first 200 epochs, the adversarial accuracy and clean accuracy on the test set mirrored those on the training set. However, with further adversarial training, there is a notable increase in the adversarial accuracy and clean accuracy on the training set, while those on the test set remain unchanged, which indicates robust overfitting. The findings in Table 7 and Fig.3 show that the adversarial examples encountered during training do not generalize well to the test set. It suggests the presence of a distribution gap between discrete adversarial samples generated by different attack methods, as well as a distribution gap between adversarial samples generated during training and those encountered in the test set using the same attack method.

To provide further evidence of this distribution gap, we calculate the Wasserstein distance between the distributions of adversarial samples generated by PGD-1 and OMPGS on PGD/OMPGS-based adversarially trained model respectively (detailed in App.F). A greater Wasserstein distance suggests a larger discrepancy between the two distributions. Two main observations are evident from Table 8. First, while PGD-based methods yield discrete adversarial samples with consistent distributions during both training and testing phases, these samples present significantly disparate distributions compared to those produced by OMPGS-based methods. This consistency in distribution with PGD-based methods is coherent with the results in Table 7, revealing substantial accuracy against PGD-based attacks but a lack of substantial defense against OMPGS-based attacks. Second, the adversarial samples derived via OMPGS exhibit a prominent distribution gap pre and post adversarial training. This distinction is indicative of the declining adversarial accuracy of the retrained classifier, as noted in Table 7 and Fig.3, through the course of the adversarial training.

Robust overfitting with categorical vs. continuous data. While robust overfitting in adversarial training with continuous data has been extensively researched (Yu et al., 2022), the root causes differ when dealing with categorical data. Methods based on adversarial training typically employ heuristic search techniques like PGD or OMPGS to discover discrete adversarial samples for training. Due to the NP-hard nature of combinatorial search, these techniques can only explore a subset of adversarial samples, leaving samples outside this range to be perceived as Out-of-Distribution (OOD) by the classifier. This situation poses significant challenges for the model to generalize its robustness to unseen adversarial samples during testing. Attempted solutions such as thresholding out small-loss adversarial samples (Yu et al., 2022) have

proven inadequate on categorical data in App.I.4. Therefore, we opt for regularized learning-based paradigms for enhanced robustness in training with categorical data, avoiding the necessity to generate discrete adversarial samples.

F. Empirical Study of the Robust Overfitting Issue

Let P_{tr} and P_{te} (O_{tr} and O_{te}) denote the adversarial samples produced by the PGD-based attack P (OMPGS-based attack O), which are used respectively for adversarial training (tr) and testing (te). The empirical evaluation of the distribution gap is conducted by comparing the following 4 groups of Wasserstein distance scores.

Wasserstein distance between in-distribution samples (WD_{in}): We first measure the Wasserstein distance between samples within each of P_{tr}, P_{te}, O_{tr} and O_{te}. For each set, we randomly shuffle twice the adversarial samples and select 90% of the samples from the set as the probe and gallery set. We then compute the Wasserstein distance between the probe and gallery set. This process is repeated 20 times. We record all the Wasserstein distance scores to measure the distribution gap between in-distribution adversarial samples within each set. WD_{in} is considered as a baseline. We expect the Wasserstein distance scores between adversarial samples from different distributions (Out-Of-Distribution) to be significantly larger than the distance scores in WD_{in}.

Wasserstein distance between the training and testing adversarial samples produced by the PGD-based method ($\text{WD}^{P}_{\text{out}}$): For P_{tr} and P_{te}, we randomly sample 90% of the adversarial samples from each set and compute the Wasserstein distance between the selected subset from the training and testing set. We repeat this process for 20 times and obtain the Wasserstein distance scores to measure the distribution gap between the training and testing adversarial samples generated by the PGD-based method.

Wasserstein distance between the training and testing adversarial samples produced by the OMPGS-based method ($\text{WD}^{O}_{\text{out}}$): For O_{tr} and O_{te}, we randomly sample 90% of the samples from each set and compute the Wasserstein distance between the two selected subsets. This process is repeated 20 times to obtain all the Wasserstein distance scores, assessing the distribution difference between training and testing adversarial samples generated by the OMPGS-based attack method.

Table 8. Average and standard (AVG) deviation (STD) of the Wasserstein distance scores

Group of Wasserstein distance	AVG	STD
WD_{in}	0.06	0.003
$\text{WD}^{P}_{\text{out}}$	0.05	0.001
$\text{WD}^{O}_{\text{out}}$	0.12	0.002
$\text{WD}^{PO}_{\text{out}}$	0.18	0.002

Wasserstein distance between the training and testing adversarial samples produced by the PGD-based and OMPGS-based attack methods ($\text{WD}^{PO}_{\text{out}}$): We conduct a cross-check in this part. We randomly sample 90% of the samples from P_{tr} and O_{te} respectively and compute the Wasserstein distance between the selected subset of adversarial samples from the two sets. The same distance computing operation is also conducted on the subsets from O_{tr} and P_{te}. This process is repeated for 20 times and obtain the Wasserstein distance scores to assess the distribution difference between training and testing adversarial samples generated using different attack methods.

In Table 8, the averaged Wasserstein scores of WD_{in} and $\text{WD}^{P}_{\text{out}}$ are the smallest among the four groups of distance values. Conversely, $\text{WD}^{PO}_{\text{out}}$ and $\text{WD}^{O}_{\text{out}}$ rank as the largest and second largest, respectively. Our findings can be summarized from two perspectives. First, we conduct a Mann-Whitney U test on the distance scores of WD_{in} and $\text{WD}^{P}_{\text{out}}$. The test results indicate no significant difference between the distance scores in these two groups, yielding a p-value of 0.20. This suggests that the PGD-based method generates discrete adversarial samples with similar distributions for both training and testing. Consequently, the PGD-based adversarial training achieves high adversarial accuracy, as observed in Table 7. Second, we conduct Mann-Whitney U tests between WD_{in} and $\text{WD}^{O}_{\text{out}}$, as well as between WD_{in} and $\text{WD}^{PO}_{\text{out}}$. The hypothesis tests reveal that $\text{WD}^{O}_{\text{out}}$ and $\text{WD}^{PO}_{\text{out}}$ are significantly higher than WD_{in}, with p-values of 0.02 and 0.01, respectively. This

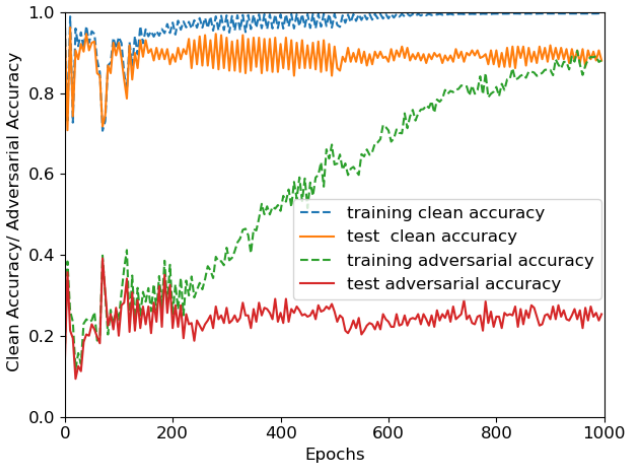


Figure 3. The "robust overfitting" of adversarially trained MLP on *Splice*.

indicates that 1) the training and testing adversarial samples generated by the OMPGS-based adversarial training method have different distributions and 2) the training adversarial samples generated by one method (either PGD-based or OMPGS-based) have a different distribution from the testing adversarial samples generated by the other method. These results align with the low adversarial accuracy of the PGD-based adversarial training method when facing the OMPGS-based attack, and vice versa. Additionally, the observations confirm the occurrence of robust overfitting in the OMPGS-based adversarial training method, as illustrated in Fig.3.

G. Distinctive Factors in Robustness with Categorical Data

G.1. Distinctive Factors in Assessing Robustness with Categorical Data

We emphasize three critical distinctions in characterizing and evaluating the adversarial robustness of categorical data compared to continuous data. Firstly, categorical data exists in discrete space, where each feature represents a unique category. Adversarial manipulation of categorical features involves switching from one feasible category to another, rendering traditional L_Q distance metrics inapplicable. Consequently, samples generated through PGD and FGSM attacks are considered infeasible to use over discrete data directly (Lei et al., 2019; Bao et al., 2021; Wang et al., 2020). However, PGD adversarial training and TRADES are both applicable to relaxed categorical data. Adversarial samples are generated by relaxing $b(x_i)$ into continuous data, yielding float categorical values in $\mathbb{R}^{p*m}$. While these samples are inappropriate for directly evaluating model robustness in the discrete domain, they are effective for adversarial training, fostering improved robustness, as discussed in the global response.

Secondly, attacking discrete data entails a complex NP-hard mixed-integer nonlinear programming challenge (Lee & Leyffer, 2011). Moreover, the size of the adversarial space expands exponentially with the feature dimension. Although transitioning the discrete problem to the continuous domain yields approximate solutions, the intricate combinatorial nature impedes complete coverage of feasible discrete adversarial samples. Adversarial training, which depends on these relaxed solutions, risks overfitting to these approximations. Our study confirms this limitation, where adversarial training struggles to significantly bolster the robustness of discrete data—especially in high-dimensional settings with substantial attack budgets.

Finally, it is essential to recognize that certifiable adversarial robustness and adversarial risk bounds established for the image domain do not hold for discrete data. These bounds are based on L_Q distance ($q \geq 1$) and do not adequately explain the true factors influencing the adversarial risk of discrete data, as demonstrated in Theorem 1 of (Bao et al., 2021). Therefore, applying these bounds to discrete data would yield inaccurate and unreliable results.

G.2. Distinctive Factors in L_0 Robustness

(Tsipras et al., 2018) demonstrated that a model relying on multiple weakly correlated features with the label can make high-confidence (low entropy) predictions, which appears to conflict with our proposed method for smoothing the impact of different features. However, (Tsipras et al., 2018) primarily focused on the L_Q attack scenario, where experiments involve L_2 and L_∞ attacks. However, our focus is on enhancing the adversarial robustness of categorical data, When perturbing categorical features, the concept of "modification magnitude" loses relevance. Instead, each feature undergoes a transformation by switching between distinct category values (switching from its original category value to another one). In this context, evaluating robustness using L_∞ attacks is infeasible, as mentioned in our earlier responses. Therefore, adversarial attacks on categorical data are framed within the L_0 attack framework, rather than the L_∞ attack scenario. It's important to underline that distinct attack scenarios can yield varying conclusions regarding adversarial robustness. However, the fundamental concept driving adversarial robustness remains consistent for both L_0 and L_Q attacks — mitigating overfitting on the training data is paramount.

For instance, in the context of L_∞ attacks, overfitting often occurs with respect to the background. As every pixel can be perturbed to some extent, classifiers that overfit to background elements become susceptible to adversarial attacks. This concurs with the findings of (Tsipras et al., 2018). Standard models that utilize all features tend to be vulnerable, while adversarially trained models tend to focus on influential features. This vulnerability arises from the classifier's overfitting to background features. This leads us to the insight that due to the permissible perturbation of any feature within certain bounds, changing influential features to alternative patterns is notably more challenging than altering background features, thus rendering background overfitting a significant adversarial vulnerability.

Nonetheless, in the context of an L_0 norm bounded attack, the scenario differs. When weakly correlated features are perturbed, highly influential features still remain untouched within the confines of the L_0 norm constraint. Consequently,

Algorithm 1 FSGS for general categorical data

Input: The candidate set $H = \{1, 2, ...p\}$ of all categorical features, categorical attack budget ϵ

$S \leftarrow \emptyset$
for $iter = 0, 1, 2, \ldots$ **do**
  for each $j \in H/S$ **do**
    for each $s \subset S$, if $|s| < \epsilon$ **do**
      $\hat{x}(j, s) = B(x, \{j\} \cup s)$
    end for
    $m_f(x(j)) = \max_{s \subset S, |s| < \epsilon} m_f(\hat{x}(j, s))$
  end for
  $m_f(x, S) = \max_{j \in H/S} m_f(x(j))$
  $j^* = \arg\max_{j \in H/S} m_f(x(j))$
  $S \leftarrow S \cup \{j^*\}$
  if $m_f(x, S) \geq 0$ **then attack successfully**
  if $Time \geq \Gamma$ **then timeout**
end for

targeting the most influential features becomes a pathway to a successful attack, which is a contrast to the L_∞ attack situation. As an echo, our defense thus aims to smooth the feature-wise contribution to the classifier, making the adversary difficult to identify influential features. This fundamental discrepancy is at the root of the disparities between our findings and those presented in (Tsipras et al., 2018).

H. Detailed Experimental Settings

H.1. The Settings of FSGS and OMPGS

To evaluate adversarial robustness, we employ the FSGS attack and OMPGS attack, shown in Algorithm.1 and 2. The definition of the notations can be found in App.H.4. It's also worth noting that, in terms of attack methods for discrete data, while FSGS is a black-box attack and OMPGS is white-box, FSGS, with an extensive search, often encompasses the search space of OMPGS under the same attack budget, yielding higher success rates, as demonstrated in (Bao et al., 2021). For both methods, we impose a time constraint on each dataset. Specifically, we allocate 1s, 150s, and 2s for FSGS, and 1s, 5s, and 1.2s for OMPGS, corresponding to *Splice*, *PEDec*, and *Census* datasets, respectively. Adversarial accuracy, which measures the prediction accuracy on adversarial samples generated by FSGS or OMPGS, is used as the metric for assessing robustness. These settings are consistently applied to all methods, including $IGSG_{\text{reg}}$, the baseline methods, and the ablation methods. In the case of mixed-type datasets like *Census*, we devise variations of FSGS and OMPGS to enhance the effectiveness of the attack. Further details can be found in App.H.4.

H.2. Details of the Baseline Methods

1. Standard Training (*Std Train*) is the model trained with adversary-free data by cross-entropy.
2. PGD Adversarial Training (*Adv Train*) is the vanilla adversarial training (Madry et al., 2017).
3. *Fast-BAT* (Zhang et al., 2022) advances vanilla adversarial training from the perspective of bi-level optimization. It achieves a better accuracy-robustness balance than *Adv Train*.
4. *TRADES* (Zhang et al., 2019) optimizes a regularized surrogate loss composed of empirical risk minimization and a robustness regularization term.
5. Adversarial Feature Desensitization (*AFD*) (Bashivan et al., 2021) improves robustness by learning a feature space where the adversary-free and adversarial instances share the same distribution.
6. Probabilistic Adversarial Training (PAdvT) (Xu et al., 2023) first use Probabilistic Categorical Adversarial Attack (PCAA) proposed in the same paper to generate adversarial samples in discrete space and then uses these adversarial samples for adversarial training.

Algorithm 2 OMPGS for general categorical data

Input: The candidate set $H = \{1, 2, ...p\}$ of all categorical features, categorical attack budget ϵ

$S \leftarrow \emptyset$
for $iter = 0, 1, 2, \ldots$ **do**
  for $s \subset S$, if $|s| \leq \epsilon$ **do**
    $r_s \leftarrow \nabla f_y(B(x, s))$
    if $m_f(B(x, s)) \geq 0$
    then attack successfully
  end for
  for $j \in H/S$ **do**
    $s_j = \underset{s_j \subset S, |s_j| < \epsilon}{\arg\max} |r_s[j]|, \ \hat{x}_j = B(x, \{j\} \cup s_j)$
  end for
  $j^* \leftarrow \underset{j \in H/S}{\arg\max} \ m_f(\hat{x}(j))$
  $S \leftarrow S \cup \{j^*\}$
  if $Time \geq \Gamma$ **then timeout**
end for

7. Input Gradient Regularization (*IGR*) (Ross & Doshi-Velez, 2018) penalizes the magnitude of the vanilla gradient of the classification loss with respect to the training data.
8. Jacobian Regularization (*JR*) (Hoffman et al., 2019) proposes to penalize the approximation of the Frobenius norm of the Jacobian matrix.

The last seven baselines except the sixth baseline are all originally designed for continuous input. We relax the one-hot encoded representation of categorical training data when adapting these baselines to our test except for *PAdvT*, which is originally designed for categorical data. For four adversarial training baselines (*Adv Train*, *Fast-BAT*, *TRADES* and *AFD*), we adopt L_1-norm bounded adversary in the inner maximization of the adversarial training process. When a mixture of categorical and numerical features presents (e.g., in *Census* dataset), the PGD-1 attack is applied for the categorical features, and the PGD-∞ attack is used for numerical features. For two regularization-based baselines (*IGR* and *JR*), we compute the gradient of the classifier's output (*JR*) / the classification loss (*IGR*) with respect to the continuous relaxation of the categorical data. The details about the hyper-parameters during training can be found in App.H.3.

H.3. The Settings of the Hyper-parameters in the Training Phase

First, we talk about the learning rate. We experiment with different learning rates for the MLP model. Specifically, we set the learning rates to 0.07, 0.2, and 0.008 for *Splice*, *PEDec*, and *Census* datasets, respectively. All methods utilizing IG regularization achieve the best performance using the same learning rate. For other methods, unless otherwise specified, we use learning rates of 0.07, 0.00001, and 0.008 to achieve optimal performance for the MLP model. In the case of *PEDec* using the IG-based training paradigm, we use a larger learning rate to achieve optimal solutions of the smoothness of IG scores for each feature. It is important to note that large learning rates would decrease both robustness and accuracy in other situations. For the Transformer model, we adopt learning rates of 0.003, 0.002, and 0.02 for *Splice*, *PEDec*, and *Census*, respectively.

As for the hyper-parameter α in Eq.6, we select a value to make IG and SG in the same magnitude without tuning them. Empirically, we choose 10, 0.01, and 0.1 for MLP and 0.001, 10, 0.001 for Transformer on *Splice*, *PEDec* and *Cencus* respectively. For the hyper-parameters β of the proposed $IGSG_{\text{reg}}$ method in Eq.8, we empirically choose 0.01, 1, 10 for MLP on the three datasets respectively and 100 for Transformer on all datasets to balance the two loss terms and ensure they are of the same magnitude. Furthermore, we conduct a hyper-parameter sensitivity analysis in App.I.8.

In the case of the PGD-1 attack in the *Adv Train*, *AFD*, and *TRADES* methods, we set ϵ to be 5 for the three datasets. The attack consists of 20 iterations, with the attack step size set to $\epsilon/10$. Regarding Fast-BAT (Zhang et al., 2022), we also set ϵ to be 5 for the three datasets. The attack step size is determined as $\epsilon/4$.

In the *IGR* method, the parameter that weighs the importance of the norm of the input gradient is set to the same value as β in Eq.8. For the MLP model, we use the values of 100, 0.1, and 3 for the three datasets, respectively. As for the Transformer

model, the values are set as 1, 0.1, and 1 for the respective datasets.

In the *JR* method, the hyper-parameter that weighs the importance of the Frobenius norm of the Jacobian matrix is tuned to achieve optimal robustness. For the MLP model, we set the values of 0.5, 1, and 0.02 for the three datasets, respectively. As for the Transformer model, the values are set as 1, 0.05, and 0.1 for the respective datasets.

In the *AFD* method, Algorithm 1 in (Bashivan et al., 2021) includes three learning rates. For the MLP model, we set the values of α to be 0.01, 0.00001, and 0.008, β to be 0.001, 0.0005, and 0.0001, and γ to be 0.001, 0.00005, and 0.0001 for the three datasets, respectively. As for the Transformer model, we set α to be 0.001, 0.002, and 0.0001, β to be 0.001, 0.0001, and 0.001, and γ to be 0.001, 0.0001, and 0.0001 for the three datasets, respectively.

In the *TRADES* method described in (Zhang et al., 2019), we set the parameter λ to balance accuracy and robustness. Specifically, for the MLP model, we set $\lambda = 1$ for the *Splice* and *Census* datasets, and $\lambda = 0.2$ for the *PEDec* dataset. As for the Transformer model, we set $\lambda = 1$ for all three datasets.

In Eq.13 of the *Fast-BAT* method (Zhang et al., 2022), we set the values of the parameters as follows: $\alpha_1 = \epsilon/4$, $\lambda = 1/\alpha_1$, $\alpha_2 = 1$ for the *Splice* dataset, and $\alpha_2 = 0.1$ for the *PEDec* and *Census* datasets.

For the training epochs, we execute 3000, 180, and 100 epochs on *Splice*, *PEDec*, and *Census* respectively. We perform 5 runs of all the methods and computed the average score and standard deviation. When evaluating the adversarial accuracy under OMPGS attack of different methods on different attack budgets, we pick the best one among the 5 runs for each method to draw Fig.2 and Fig.6.

H.4. Special Settings for Mixed-type Datasets

For mixed-type datasets that contain both categorical and numerical features, direct application of FSGS, OMPGS, or PGD attacks is not suitable for evaluating the robustness of the classifier. This is because categorical data requires an L_0 attack, while numerical data typically necessitates an L_2 or L_∞ attack.

To address this challenge and evaluate the adversarial robustness of a mixed-type classifier, an iterative approach is employed. This approach involves running FSGS or OMPGS along with PGD attacks iteratively to obtain a more effective adversary. This combination allows for a comprehensive evaluation of the robustness of the mixed-type classifier.

Before talking about the details, we note that there are p_{cat} categorical features and p_{num} numerical features. Each categorical feature has m candidate values. For a sample x, the value of feature j is k^*. After perturbation, the value is $\hat{k}$. The ground truth label of x is y^*. During the attack, we maintain a greedy set S, showing the alterable features. Each feature not in S cannot be changed, i.e. for $j \notin S$, $\hat{k} = k^*$. For the features in S, it is possible to choose any of the m candidate values, and it is also acceptable to remain unchanged. Here we introduce the notation in (Bao et al., 2021). Given a greedy set S,

$$m_f(x) = \max_{y \neq y^*} \{f_y(x)\} - f_{y^*}(x)$$
$$m_f(x, S) = \max_{\textit{diff}(x,\hat{x}) \subset S} m_f(\hat{x})$$

where we denote *diff*$(x, \hat{x})$ as the set of feature indices where $\hat{k} \neq k^*$. The function $m_f(x)$ indicates whether the sample x is misclassified. If $m_f(x) < 0$, it means that x is classified correctly, while $m_f(x) \geq 0$ indicates misclassification. The function $m_f(x, S)$ checks whether the attack is successful under the constraints of the feature set S. The notation $B(x, s)$ represents the adversarial sample $\hat{x}$ obtained by modifying the features of x as indicated by the binary vector s. Algorithm.3 outlines the attack process using FSGS+PGD for mixed-type data, while Algorithm.4 describes the attack process using OMPGS+PGD for mixed-type data. For general categorical data where there are no numerical features, the "PGD" step in the algorithms can be ignored or ϵ_n can be set to 0.

During the experiment, each feature is normalized before applying the PGD attack. For PGD-∞ attack, we set $\epsilon_n = 0.2$ for the *Census* dataset, with a total of 20 attack steps. The attack step size is set to 0.02. During the training process of *Adv Train*, *AFD*, *TRADES*, and *Fast-BAT*, we use a combination of PGD-1 attack for categorical features and PGD-∞ attack for numerical features to generate adversarial samples for mixed-type data. The same attack settings are applied during the training of *Adv Train*, *AFD*, and *TRADES*. For *Fast-BAT*, we also set $\epsilon_n = 0.2$, but the attack step size is adjusted to 0.05.

Algorithm 3 FSGS + PGD for mixed-type data

Input: The candidate set $H = \{1, 2, ...p_{cat}\}$ of all categorical features, PGD attack budget ϵ_n for numerical data, categorical attack budget ϵ_c

$S \leftarrow \emptyset$
for $iter = 0, 1, 2, \ldots$ **do**
  for each $j \in H/S$ **do**
    for each $s \subset S$, if $|s| < \epsilon_c$ **do**
      $\hat{x}(j, s) = B(x, \{j\} \cup s)$
      $\delta(j, s) = \text{PGD}_\infty(\hat{x}(j, s), \epsilon_n)$
      $\hat{x}(j, s) = \hat{x}(j, s) + \delta(j, s)$
    end for
    $m_f(x(j) + \delta(j, S)) = \max_{s \subset S, |s| < \epsilon_c} m_f(\hat{x}(j, s))$
  end for
  $m_f(x + \delta, S) = \max_{j \in H/S} m_f(x(j) + \delta(j, S))$
  $j^* = \arg\max_{j \in H/S} m_f(x(j) + \delta(j, S))$
  $S \leftarrow S \cup \{j^*\}$
  if $m_f(x + \delta, S) \geq 0$ **then attack successfully**
  if $Time \geq \Gamma$ **then timeout**
end for

Algorithm 4 OMPGS + PGD for mixed-type data

Input: The candidate set $H = \{1, 2, ...p_{cat}\}$ of all categorical features, PGD attack budget ϵ_n for numerical data, categorical attack budget ϵ_c

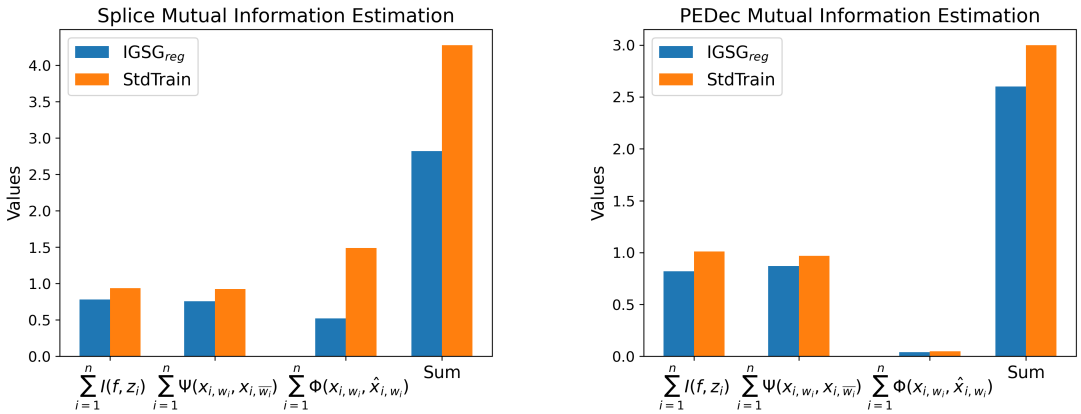
$S \leftarrow \emptyset$
for $iter = 0, 1, 2, \ldots$ **do**
  for $s \subset S$, if $|s| \leq \epsilon_c$ **do**
    $r_s \leftarrow \nabla f_y(B(x, s))$
    if $m_f(B(x, s) + \text{PGD}_\infty(B(x, s), \epsilon_n)) \geq 0$
    then attack successfully
  end for
  for $j \in H/S$ **do**
    $s_j = \arg\max_{s_j \subset S, |s_j| < \epsilon_c} |r_s[j]|, \ \hat{x}_j = B(x, \{j\} \cup s_j)$
  end for
  $j^* \leftarrow \arg\max_{j \in H/S} m_f(\hat{x}(j) + \text{PGD}_\infty(\hat{x}(j), \epsilon_n))$
  $S \leftarrow S \cup \{j^*\}$
  if $Time \geq \Gamma$ **then timeout**
end for

I. Additional Experimental Results

I.1. Approximation to the Mutual Information-based Adversarial Risk Bound

In this section, we evaluate the mutual information as delineated in the adversarial risk bound (Eq.7), comparing models trained via Std Train and IGSG methods. Given the intricacies and potential inaccuracies in assessing an entire neural network, we focus on a simplified model comprising a single fully connected layer, with softmax activation for multi-class classification and sigmoid activation for binary classification. We utilize the Mutual Information Neural Estimation (MINE) technique (Belghazi et al., 2018) to assess the terms and their weighted sum in Eq.7.

For training, we randomly selected 200 and 500 samples, 20 times each, from the training sets of *Splice* and *PEDec* datasets, respectively. These samples undergo training using Std Train and IGSG approaches, with a learning rate of 0.001, over 200 and 1000 epochs, respectively. This process yields an approximate accuracy of 0.9 for both datasets. Subsequently, we



(a) Splice (b) PEDec

Figure 4. Mutual Information Estimation for terms in Eq.7 for *Splice* and *PEDec* Datasets

evaluate the adversarial robustness of 20 models each from Std Train and IGSG, employing FSGS and OMPGS attacks. Regarding the most sensitive features ω_i in Eq.7, we predetermine them based on the top 5 features exhibiting the highest attack frequency in Std Train models on MLP under OMPGS attacks. These features were fixed across all samples. For *Splice*, ω_i are [28, 29, 30, 31, 32], and for *PEDec*, [3592, 3755, 3808, 4390, 4918]. Using these predetermined ω_i, we calculate the four mutual information terms, as illustrated in Fig.4, based on the 20 sampled datasets and corresponding model parameters, utilizing the MINE methodology. We also calculate the average adversarial accuracy on FSGS and OMPGS, the result is shown in Table 9.

This experiment aims to demonstrate two key aspects. Firstly, IGSG-trained networks exhibit a reduction in the mutual information terms in Eq.7, suggesting a lower adversarial risk bound. Secondly, beyond just a lower adversarial risk bound, IGSG-trained networks also empirically manifest enhanced adversarial accuracy.

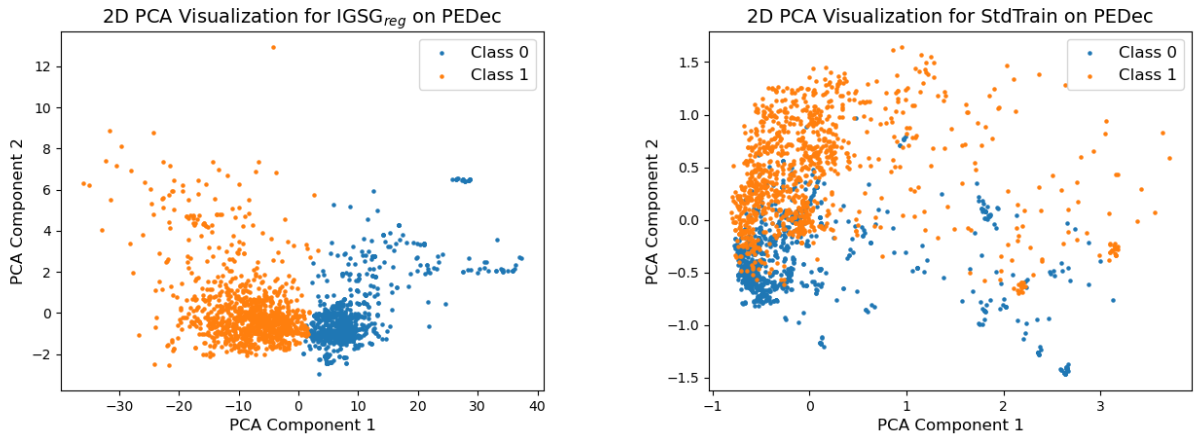
The results displayed in Fig.4 encompass four mutual information terms related to the adversarial risk bound. We first examine "Sum". "Sum" is defined as $\sum_{i=1}^{n} I(f;z_i)+2\sum_{i=1}^{n}\Psi(x_{i,\omega_i},x_{i,\overline{\omega_i}})+\sum_{i=1}^{n}\Phi(x_{i,\omega_i},\hat{x}_{i,\omega_i})$, representing the adversarial risk bound in Eq.7. We can refer to Table 9 for the average adversarial accuracy across 20 models trained on randomly sampled data under FSGS and OMPGS attacks. For both *Splice* and *PEDec* datasets, the IGSG method typically yields lower "Sum" values and higher adversarial accuracy, corroborating that IGSG effectively reduces the adversarial risk bound in Eq.7 and that this reduction positively correlates with improved adversarial accuracy.

Table 9. Average Adversarial Accuracy on 20 logistic regression models for *PEDec* and *Splice* datasets.

Dataset	Attack	*IGSG*$_{\text{reg}}$	Std Train
Splice	FSGS	0.019	0.010
	OMPGS	0.139	0.122
PEDec	FSGS	0.709	0.648
	OMPGS	0.748	0.668

Focusing on the first three terms in Fig.4, we observe that $\sum_{i=1}^{n} I(f,z_i)$, indicative of adversary-free generalization error, is lower after using SG regularization compared to Std Train, signifying a more generalized classifier. The term $\sum_{i=1}^{n}\Psi(x_{i,\omega_i},x_{i,\overline{\omega_i}})$ quantifies the differential contribution of highly vulnerable features ω_i and other features $\overline{\omega_i}$. Here, classifiers trained with IGSG typically exhibit lower values, suggesting a more balanced reliance on diverse features. For $\sum_{i=1}^{n}\Phi(x_{i,\omega_i},\hat{x}_{i,\omega_i})$, which measures the sensitivity of the most vulnerable features ω_i to adversarial perturbations, IGSG-trained classifiers generally show lower values, particularly in the *Splice* dataset. This trend is attributed to the high vulnerability of certain features in ω_i for *Splice*, as evident in Fig.1(a). Perturbations in a single feature often lead to significant drops in prediction scores, resulting in larger values for Std Train, while IGSG effectively reduces this effect. For *PEDec*, successful attacks are usually driven by a combinatorial search. The combination of features with high attack frequency does not necessarily lead to successful attacks, hence the lower values for both Std Train and IGSG in this term.

In summary, we observe that the classifier trained with IGSG exhibits lower values for all the four mutual information terms in the proposed upper bound in Eq.7 (thus a globally lower bound value) and higher adversarial accuracy across the two datasets. This finding firstly indicates that enforcing IGSG regularization can reduce the mutual information-based upper bound of the adversarial risk proposed in Eq.7. Furthermore, we consider adversarial accuracy as a measure of actual



(a) $IGSG_{\text{reg}}$ (b) Std Train

Figure 5. 2D PCA Boundary Visualization on *PEDec* Dataset

adversarial risk. Higher adversarial accuracy indicates lower adversarial risk and vice versa. This quantitative evaluation demonstrates the correlation between the upper bound and actual adversarial risk. Lower values of the mutual information bound signify higher adversarial accuracy, thus indicating a reduced level of adversarial risk.

I.2. Visualization of the Classification Boundaries

In this section, we present a visualization of classification boundaries for classifiers trained using IGSG and Std Train methods, specifically for the *PEDec* dataset. We employ Multi-Layer Perceptron (MLP) classifiers trained via both IGSG and Std Train approaches. The visualization focuses on the features preceding the final fully connected layer within the test set. These features are compressed into a 2-dimensional space using Principal Component Analysis (PCA) for clearer representation.

Each sample in this visualization is labeled according to its predicted class by each respective classifier, offering an intuitive depiction of the classification boundaries. The results, as illustrated in Fig.5, reveal distinct differences between the two training methodologies. The IGSG-trained classifier exhibits an almost linear and distinct boundary between the two classes in the PCA visualization. In contrast, the Std Train-trained classifier's visualization does not present a clear demarcation. There is considerable overlap between the two classes in the PCA visualization of features from the last layer, indicating a twisted classification boundary.

This observation underscores that, compared to Std Train, $IGSG_{\text{reg}}$ facilitates a smoother and more discernible classification boundary. Such a visualization not only highlights the distinctiveness of the $IGSG_{\text{reg}}$ method but also demonstrates its efficacy in achieving clearer class separations.

I.3. Experimental Results on Transformer Models

In addition to implementing $IGSG_{\text{reg}}$ on the MLP model to demonstrate its effectiveness, we also conducted experiments on a Transformer model. Table 10 presents the accuracy and adversarial accuracy against FSGS attack for each robust training method used with the Transformer model. For the *Splice* dataset, we observe that none of the methods provide an effective defense for the Transformer model. This could be attributed to the presence of particularly sensitive features in the *Splice* dataset, as mentioned in Sec.5.3. The Transformer model amplifies this effect by focusing more attention on these features, resulting in lower adversarial accuracy. However, $IGSG_{\text{reg}}$ achieves comparatively higher adversarial accuracy. Regarding the *PEDec* dataset, $IGSG_{\text{reg}}$ demonstrates slight improvement compared to other methods, and the differences in adversarial robustness among the different robust training methods are not significant. This may be due to the self-attention layer in the Transformer model, which makes the relationships between different features less flexible compared to the MLP model. For the *Census* dataset, most of the baseline methods do not exhibit substantial improvement over the baseline model. However, $IGSG_{\text{reg}}$ shows a significant improvement of 10.2% compared to the undefended model.

Table 10. Adversarial Accuracy under FSGS attack, PCAA attack and Accuracy (%) for $IGSG_{\text{reg}}$ and baseline models for the Transformer model. Adv Train (Madry et al., 2017), Fast-BAT (Zhang et al., 2022), TRADES (Zhang et al., 2019), AFD (Bashivan et al., 2021), PAdvT (Xu et al., 2023), IGR (Ross & Doshi-Velez, 2018), JR (Hoffman et al., 2019)

Dataset	Attack	Undefended	Adversarial Training baselines					Regularization baselines		**Ours**
		Std Train	Adv Train$_{L1}$	Fast-BAT$_{L1}$	TRADES$_{L1}$	AFD$_{L1}$	PAdvT	IGR	JR	$\boldsymbol{IGSG_{\text{reg}}}$
Splice	FSGS	0.9±0.9	0.4±0.5	1.0±1.1	0.0±0.0	0.2±0.4	0.2±0.4	0.4±0.3	0.1±0.1	**2.3±1.4**
	PCAA	8.6±3.9	2.8±0.9	10.5±3.2	7.3±1.2	2.4±1.7	6.5±1.8	**11.3±3.5**	7.8±3.4	11.1±2.6
	Clean	96.9±0.4	96.7±0.8	96.4±0.5	96.2±0.6	93.7±1.5	95.6±0.6	96.4±0.2	92.9±1.7	96.7±0.7
PEDec	FSGS	41.1±4.1	60.6±0.7	49.9±3.8	59.0±3.9	48.1±9.8	22.6±1.3	59.5±5.0	62.2±1.9	**63.5±3.7**
	PCAA	87.1±2.4	75.5±1.2	87.8±0.9	**90.8±0.6**	86.7±3.3	87.4±1.2	89.7±2.3	90.6±1.0	89.2±1.3
	Clean	96.2±0.5	96.0±0.2	96.1±0.1	96.7±0.1	96.1±0.4	96.2±0.1	95.5±0.1	93.1±1.8	95.7±0.3
Census	FSGS	27.6±4.3	34.1±2.7	33.1±6.1	32.2±8.0	32.2±1.0	30.4±3.4	25.1±5.3	32.7±0.4	**37.8±4.3**
	PCAA	92.3±1.0	94.5±0.3	91.8±1.2	91.5±1.9	92.9±1.2	**94.3±0.2**	93.3±0.3	93.8±0.7	93.7±0.2
	Clean	95.2±0.1	95.2±0.1	93.4±1.1	94.4±0.1	95.1±0.0	95.1±0.0	95.1±0.1	94.9±0.2	94.8±0.1

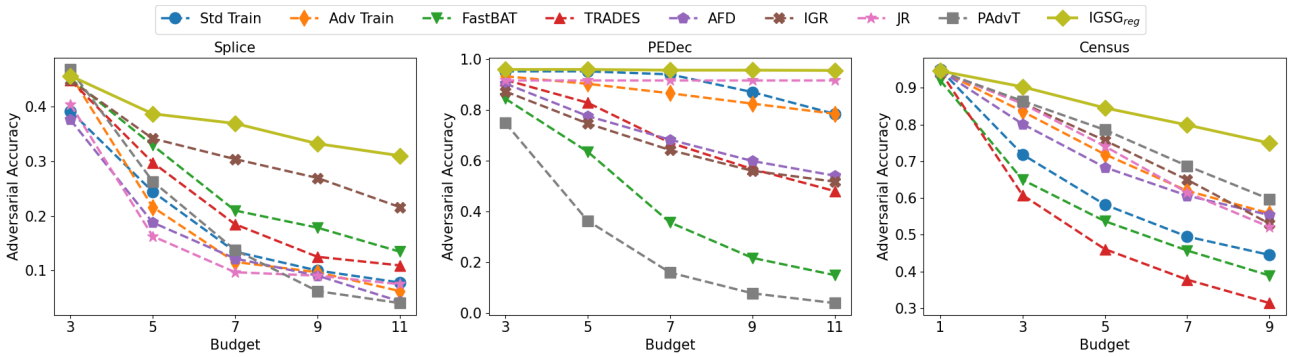


Figure 6. Adversarial accuracy for $IGSG_{\text{reg}}$ and baselines under OMPGS attack with varied attack budgets for the Transformer model.

In Fig.6, we present the adversarial accuracy of all the methods when subjected to OMPGS attacks with varying budgets for the Transformer model. As discussed in Sec.5.3, higher adversarial accuracy and a lower decrease rate of adversarial accuracy with increasing attack budgets indicate better model robustness. Similar to the results obtained with the MLP model, we observe that $IGSG_{\text{reg}}$ outperforms the baseline models in terms of adversarial accuracy under OMPGS attacks. Specifically, for the *Splice* dataset, $IGSG_{\text{reg}}$ exhibits a noticeably lower decrease rate of adversarial accuracy, although its adversarial accuracy is similar to some baseline methods when the attack budget is small. For *PEDEC*, most methods demonstrate very high adversarial accuracy compared to the MLP model. This may be because the multi-head paradigm in the self-attention layer makes the gradient less informative compared to the MLP model. In this scenario, $IGSG_{\text{reg}}$ achieves the highest adversarial accuracy, with almost no samples successfully attacked as the attack budget increases. The *JR* method also maintains a constant adversarial accuracy as the attack budgets increase, but it is susceptible to attacks on a few samples when the budget is small and its accuracy is inferior to $IGSG_{\text{reg}}$. Regarding the *Census* dataset, we observe that nearly all methods achieve an adversarial accuracy above 0.9 when modifying a single feature. As the attack budget increases, $IGSG_{\text{reg}}$ exhibits a significantly lower decrease rate compared to other methods.

I.4. Comparison to Methods Targeting at Robust Overfitting

In this section, we give a comparison of the adversarial robustness between $IGSG_{\text{reg}}$ and proposed methods aiming to address robust overfitting. We consider two works in this comparison. (Yu et al., 2022) found that small-loss adversarial samples are the cause of robust overfitting. MLCAT was proposed to constrain the minimum loss. Loss scaling and weight perturbation are used for two implementations, denoted as MLCAT_{LS} and MLCAT_{WP} respectively. (Chen et al., 2020) used learned smoothing to mitigate robust overfitting. It introduced knowledge distillation to smooth the logits, and performed stochastic weight averaging to smooth the weights (denoted as KD+SWA). We implement these two works on the original PGD adversarial training (Adv Train (Madry et al., 2017)). The results are shown in Table 11. We can observe that IGSG consistently outperforms both of the two methods when alleviating the robust overfitting issue on categorical data. Also, KD+SWA has better performance than Adv Train on *PEDec* and *Census* datasets, but is inferior on *Splice* dataset. However, MLCAT is inferior to Adv Train under both LS and WP implementations. This may demonstrate that the statement that small-loss data cause robust overfitting may not be correct in the categorical domain.

Table 11. Adversarial Accuracy and Accuracy over clean samples (%) for $IGSG_{\text{reg}}$ and other methods alleviating robust overfitting.

Dataset	Adversary	Adv Train	KD+SWA	MLCAT_{LS}	MLCAT_{WP}	$\boldsymbol{IGSG_{\text{reg}}}$
Splice	FSGS	43.6±0.7	36.8±1.9	25.4±2.8	24.6±1.7	**44.0±2.6**
	OMPGS	51.7±1.4	41.2±2.3	30.3±2.7	29.9±2.0	**63.8±4.2**
	Clean	96.2±0.4	94.0±1.5	94.4±0.8	94.6±1.2	95.9±0.7
PEDec	FSGS	53.1±1.7	62.5±3.5	45.8±3.2	52.8±4.5	**86.5±3.8**
	OMPGS	74.1±2.1	80.2±2.0	67.9±2.4	68.8±4.7	**88.0±4.0**
	Clean	96.2±0.0	96.6±0.1	96.8±0.1	95.3±0.2	95.5±0.2
Census	FSGS	54.1±2.3	65.4±4.4	53.2±3.7	52.6±2.9	**67.2±3.5**
	OMPGS	62.7±3.3	66.5±5.6	67.5±1.9	66.5±3.5	**71.3±9.0**
	Clean	94.5±0.3	95.3±0.1	94.6±0.0	94.8±0.2	95.5±0.2

I.5. Time Complexity Analysis

In this section, we first outline the time complexity of IGSG and compare its computational efficiency with the OMPGS robustness evaluation method (Wang et al., 2020). OMPGS leverages gradients to streamline the greedy search process used in FSGS (Elenberg et al., 2018), resulting in faster attack speeds and similar performance. The time complexity of IGSG is denoted as $O(N * (T + R + 1))$, indicating its reliance on the number of samples N, the steps T in the Riemann approximation for Integrated Gradient, and the R randomly sampled neighbors for smoothing. Contrastingly, the complexity of OMPGS is $O(N * (2^{\kappa} + p * \kappa))$, with κ indicating the attack iterations and p the feature count, showcasing its higher computational demand.

Runtime comparisons between IGSG and OMPGS on MLP and Transformer models across three datasets are conducted in Table 2. Here we choose $\kappa = 5$ for OMPGS, $T = 20$, and $R = 5$ for IGSG. It reveals that IGSG is significantly faster, ranging from 17 to 240 times quicker than OMPGS.

Additionally, we examine the training time complexity of *IGSG*reg in relation to other baseline methods. The per-iteration complexity for *IGSG*reg mirrors that of IGSG, at $O(N * (T + R + 1))$. Conversely, the per-iteration complexity for OMPGS-based adversarial training matches OMPGS's own complexity: $O(N * (2^{\kappa} + p * \kappa))$. This comparison underlines the superior efficiency of IGSG-based methodologies.

We also measure the runtime cost of $IGSG_{\text{reg}}$ with the other baselines in Table 12. On *Splice*, $IGSG_{\text{reg}}$ requires significantly less training time compared to some adversarial training methods like *Adv Train*, *AFD* and *TRADES*. On *PEDec*, $IGSG_{\text{reg}}$ requires similar run-time, compared to *Adv Train*, *AFD* and *TRADES*. On *Census*, *Fast-BAT*, *JR* and *IGR* need less time than $IGSG_{\text{reg}}$, but there is a large gap between the time cost of $IGSG_{\text{reg}}$ and that of those methods.

Table 12. Time cost (min) for the training process for $IGSG_{\text{reg}}$ and baseline methods.

Model	MLP			Transformer		
Dataset	Splice	PEDec	Census	Splice	PEDec	Census
Std Train	6	8	12	17	9	7
Adv Train	78	112	84	223	74	130
Fast-BAT	27	40	37	91	29	67
TRADES	114	108	210	307	81	197
AFD	276	126	316	285	101	231
IGR	9	11	19	25	13	10
JR	13	47	23	39	14	31
$\boldsymbol{IGSG_{\text{reg}}}$	39	117	82	124	71	89

All our implementations are conducted in the Python library PyTorch on a Linux server with a single GPU (NVIDIA V100).

I.6. Detailed Ablation Study

Here, we introduce another three variants of $IGSG_{\text{reg}}$.

$\boldsymbol{SGSG_{\text{reg}}}$: We replace the TV loss of the IG scores with the TV loss defined over the smoothed gradient given in Eq.5.
$\boldsymbol{IGIG_{\text{reg}}}$: Instead of penalizing the l_p norm of the smoothed gradient, we choose to penalize the norm of the IG score vector of each instance x. We use $SGSG_{\text{reg}}$ and $IGIG_{\text{reg}}$ to verify the validity of the two robustness-enhancing regularization terms.
$\boldsymbol{IGSG\text{-}VSG_{\text{reg}}}$: We replace the difference of gradient computing given in Eq.8 with the standard smoothed gradient (Smilkov et al., 2017). We introduce $IGSG\text{-}VG_{\text{reg}}$ and $IGSG\text{-}VSG_{\text{reg}}$ to demonstrate the necessity of introducing the mean field

Table 13. Additional Ablation Study. Adversarial Accuracy and Accuracy over clean testing samples (%) for $IGSG_{\text{reg}}$ variants for the MLP model.

Dataset	Adversary	$IGSG\text{-}VSG_{\text{reg}}$	$SGSG_{\text{reg}}$	$IGIG_{\text{reg}}$	$L_2\text{-}IGSG_{\text{reg}}$	$\mathbf{IGSG_{\text{reg}}}$
Splice	FSGS	40.4±3.5	41.5±4.1	15.6±8.2	40.2±1.1	**44.0±2.6**
	OMPGS	56.3±5.9	59.2±8.6	45.9±3.5	57.9±0.9	**63.8±4.2**
	Clean	95.7±1.4	94.1±0.4	90.7±7.9	96.0±0.4	95.9±0.7
PEDec	FSGS	85.7±2.2	11.9±2.5	86.4±2.2	81.7±2.6	**86.5±3.8**
	OMPGS	84.5±3.1	30.6±2.1	85.7±4.6	83.0±1.4	**88.0±4.0**
	Clean	95.3±0.3	96.3±0.1	95.3±0.4	95.4±0.2	95.5±0.2
Census	FSGS	56.8±3.6	66.5±2.1	50.2±2.3	62.5±1.1	**67.2±3.5**
	OMPGS	68.6±4.6	**71.6±6.8**	62.3±4.2	70.6±2.4	71.3±9.0
	Clean	95.3±0.3	95.1±0.3	95.5±0.1	95.3±0.0	95.5±0.2

Table 14. Ablation Study. Adversarial Accuracy and Accuracy over clean testing samples (%) for $IGSG_{\text{reg}}$ variants for the Transformer model.

Dataset	Adversary	SG_{reg}	IG_{reg}	$IGSG\text{-}VG_{\text{reg}}$	$IGSG\text{-}VSG_{\text{reg}}$	$SGSG_{\text{reg}}$	$IGIG_{\text{reg}}$	$\mathbf{IGSG_{\text{reg}}}$
Splice	FSGS	0.3±0.2	2.2±1.6	1.5±1.4	0.7±0.7	1.0±1.0	1.3±1.3	**2.3±1.4**
	OMPGS	33.3±3.7	34.9±1.3	36.1±4.1	34.5±5.0	35.9±5.5	33.2±3.1	**36.8±4.3**
	Clean	96.1±0.6	96.5±0.5	96.7±0.4	96.7±0.3	96.4±0.6	96.7±0.5	96.7±0.7
PEDec	FSGS	60.4±4.4	57.1±6.0	53.9±3.6	60.6±4.3	59.9±5.4	57.2±6.8	**63.5±3.7**
	OMPGS	**95.7±0.2**	95.6±0.1	95.2±0.3	95.2±0.1	92.4±2.8	91.8±6.6	95.6±0.2
	Clean	95.8±0.3	95.7±0.1	95.3±0.4	95.5±0.2	95.1±0.4	95.6±0.1	95.7±0.3
Census	FSGS	28.6±0.7	31.1±1.1	36.6±4.5	33.6±2.5	28.9±0.9	26.2±2.2	**37.8±4.3**
	OMPGS	56.9±1.1	68.7±6.1	70.1±6.5	73.4±7.2	58.3±1.5	63.3±2.5	**76.9±4.8**
	Clean	95.0±0.0	94.9±0.1	95.0±0.3	93.6±0.0	95.0±0.0	95.2±0.0	94.8±0.1

smoothing-driven smoothed gradient (given by Eq.5) into the gradient smoothing-based regularization term.

L_2-$\boldsymbol{IGSG_{reg}}$: To achieve attribution smoothing, L_2 norm regularization is also simple and widely used. We replace the TV loss with an L_2 norm of the IG score. We introduce it to further confirm the effectiveness of the TV loss design in $IGSG_{\text{reg}}$.

In Table 13, we provide the adversarial accuracy of the four variants—$IGSG\text{-}VSG_{\text{reg}}$, $IGIG_{\text{reg}}$, $SGSG_{\text{reg}}$ and $L_2\text{-}IGSG_{\text{reg}}$—under FSGS attack and OMPGS attack with a budget of 5 for the three datasets on an MLP model. We also compare their performance with that of $IGSG_{\text{reg}}$.

$SGSG_{\text{reg}}$ replaces the total variation (TV) loss of $IGSG_{\text{reg}}$ with the TV loss of the smoothed gradient. It exhibits slightly inferior performance compared to $IGSG_{\text{reg}}$ on the *Splice* and *Census* datasets but performs poorly on the *PEDec* dataset. This can be attributed to the fact that regularizing the TV loss of the smoothed gradient evenly distributes the sensitivity of each feature. However, the gradient information only reflects local sensitivity and does not provide a comprehensive understanding of feature contribution.

$IGIG_{\text{reg}}$ replaces the regularization of the smoothed gradient with the L_Q norm of the IG score. Without the use of smoothed sampling, the smoothness of the classifier is inferior to that of $IGSG_{\text{reg}}$. Additionally, IG captures global information about feature contributions but is not as explicit as the gradient in guiding the direction of attack for each category. Therefore, minimizing the magnitude of IG is not as beneficial for the *Splice* and *Census* datasets.

$L_2\text{-}IGSG_{\text{reg}}$ replaces the TV loss in the regularization of the integrated gradient with an L_2 norm. Compared to SG_{reg}, $L_2\text{-}IGSG_{\text{reg}}$ generally has better adversarial accuracy. However, the L_2 norm-regulated IG term consistently yields a little lower adversarial accuracy when subjected to FSGS and OMPGS attacks, showing the effectiveness of the TV loss.

In Table 14, we present the accuracy and adversarial accuracy under FSGS attack and OMPGS attack for the Transformer model. The results are similar to those of the MLP model. Compared to the performance of *IGR* shown in Table 10 and Fig.6, SG_{reg} achieves slightly better adversarial robustness due to the smoothing. The only exception is the adversarial accuracy under OMPGS attack for *PEDec*, where SG_{reg} achieves much better robustness. This may be a result of the smoothness of gradients among neighboring samples. Notably, most variants of $IGSG_{\text{reg}}$ achieve very high adversarial accuracy under

Table 16. Adversarial Accuracy for the sensitivity analysis of α in $IGSG_{\text{reg}}$

Dataset	Attack	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$
Splice	FSGS	**42.5±1.5**	41.3±1.4	40.6±2.6	39.4±1.6
	OMPGS	**61.3±5.7**	59.9±5.4	59.1±3.7	58.3±6.7
PEDec	FSGS	**91.6±3.2**	89.0±2.8	89.3±3.5	**91.6±2.9**
	OMPGS	89.9±3.0	87.0±2.5	88.0±2.1	**93.8±2.3**
Census	FSGS	55.4±0.8	**67.5±3.6**	/	/
	OMPGS	63.5±2.5	**75.5±5.6**	/	/

Table 17. Adversarial Accuracy for the sensitivity analysis of β in $IGSG_{\text{reg}}$

Dataset	Attack	$\beta = 0.01$	$\beta = 0.1$	$\beta = 1$	$\beta = 10$
Splice	FSGS	**45.8±1.1**	44.3±2.1	42.9±1.8	44.0±2.6
	OMPGS	64.4±3.1	62.7±1.4	**66.3±0.7**	63.8±4.2
PEDec	FSGS	86.5±3.8	81.4±3.1	**88.4±2.6**	80.4±2.9
	OMPGS	88.0±4.0	**91.6±1.3**	90.1±2.0	87.6±2.7
Census	FSGS	62.7±1.4	59.8±2.4	67.2±3.5	**68.2±2.8**
	OMPGS	67.4±3.8	72.1±2.6	71.3±9.0	**74.3±3.7**

OMPGS attack, suggesting that both IG_{reg} and SG_{reg} training can defend against OMPGS attack on *PEDec*. Regarding IG_{reg}, $IGSG\text{-}VG_{\text{reg}}$, and $IGSG\text{-}VSG_{\text{reg}}$, their performance varies across datasets, indicating instability. On the other hand, $SGSG_{\text{reg}}$ and $IGIG_{\text{reg}}$ do not perform well on any dataset, suggesting that the roles of IG and SG cannot be effectively altered by each other in the loss function.

I.7. Adaptive Attack

Leveraging $IGSG_{\text{reg}}$, our strategy aims to diminish the impact of highly sensitive features to achieve a more balanced feature contribution across the board. Since $IGSG_{\text{reg}}$ focuses on reducing the influence of these sensitive features, it naturally allows for other features to have an increased role in classification to preserve accuracy. To adaptively tailor an attack method for $IGSG_{\text{reg}}$-enhanced models, we suggest focusing on less sensitive features, leaving the desensitized ones unchanged. Specifically, we adjust the FSGS attack to exempt the top 5 features with the highest IG scores identified in the Std Train model from being perturbed. The outcomes, as detailed in Table 15, reveal that this adaptive attack strategy underperforms compared to standard FSGS shown in Table 4 and 10. This indicates that $IGSG_{\text{reg}}$ effectively reduces the sensitivity of targeted features without increasing the vulnerability of others, thereby genuinely enhancing model robustness against adversarial threats in categorical data.

Table 15. Adversarial Accuracy of $IGSG_{\text{reg}}$ under FSGS based adaptive attack

Dataset	Architecture	Adv. Acc.
Splice	MLP	73.0±1.3
	Transformer	43.1±1.9
PEDec	MLP	91.2±3.3
	Transformer	68.4±1.6
Census	MLP	84.0±3.2
	Transformer	90.8±0.1

I.8. Hyper-parameter Sensitivity Analysis

In this section, we investigate the sensitivity of the hyper-parameters α and β in $IGSG_{\text{reg}}$ as defined in Eq.8, the number of steps T in the Riemann approximation of the integral for IG score calculation as specified in Eq.3, and the number of sampled instances R as defined in Eq.5. Our analysis focuses on the impact of these parameters on MLP models across three datasets. We employ FSGS and OMPGS as the attack methods for this evaluation. For both FSGS and OMPGS attacks, we maintain a consistent attack budget of 5 for each dataset and assess adversarial robustness for each setting. To evaluate the sensitivity of the four parameters, we explore α and β values of 0.01, 0.1, 1, and 10, T values of 5, 10, 20, 50, 100, and R values of 2, 5, 10, 20, 50.

The sensitivity analysis for α is presented in Table 16. Generally, tuning α has minimal impact on the adversarial robustness for the *Splice* and *PEDec* datasets. For the *Census* dataset, an α value of 0.1 significantly improves performance. However, for α values of 1 or 10, the model fails to converge, so adversarial accuracy is not reported for these settings.

Results for the sensitivity analysis of β are shown in Table 17. The analysis indicates that while β does influence adversarial accuracy, the effect is not substantial. Generally, achieving a balance between α and β tends to yield satisfactory performance.

Table 18. Adversarial Accuracy for the sensitivity analysis of T in Eq.3

Dataset	Attack	T=5	T=10	T=20	T=50	T=100
Splice	FSGS	41.8±1.7	44.0±2.0	44.0±2.6	42.1±1.9	**44.1±4.3**
	OMPGS	65.8±5.3	**66.8±0.8**	63.8±4.2	62.4±2.8	65.6±6.5
PEDec	FSGS	**87.0±6.0**	86.3±6.2	86.5±3.8	86.8±4.4	86.3±2.9
	OMPGS	**90.2±5.1**	87.9±4.5	88.0±4.0	90.0±4.6	86.9±2.7
Census	FSGS	68.3±2.3	**69.2±5.8**	67.8±4.7	65.7±3.6	67.8±3.9
	OMPGS	78.1±4.3	**78.2±7.3**	75.7±4.7	75.9±2.6	75.6±5.3

Table 19. Adversarial Accuracy for the sensitivity analysis of R in Eq.5

Dataset	Attack	R=2	R=5	R=10	R=20	R=50
Splice	FSGS	42.8±2.8	**44.0±2.6**	43.0±1.2	42.1±1.3	43.7±3.2
	OMPGS	**67.9±2.9**	63.8±4.2	65.3±4.9	63.1±0.8	65.4±2.5
PEDec	FSGS	**87.8±4.5**	86.5±3.8	86.1±2.7	87.1±2.1	87.2±4.3
	OMPGS	88.1±6.3	88.0±4.0	**89.1±3.4**	88.7±4.1	88.3±2.9
Census	FSGS	**68.4±3.1**	67.8±4.7	67.4±3.8	67.1±3.4	66.8±3.8
	OMPGS	**77.1±2.9**	75.7±4.7	74.9±4.2	75.6±2.8	76.3±3.1

Sensitivity analyses for T and R are displayed in Table 18 and Table 19, respectively. The results indicate that varying T and R does not significantly affect adversarial robustness. Therefore, selecting smaller values for T and R is recommended for time efficiency.