ENHANCING ADVERSARIAL ROBUSTNESS ON CATEGORICAL DATA VIA ATTRIBUTION SMOOTHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Many efforts have been contributed to alleviate the adversarial risk of deep neural networks on continuous inputs. Adversarial robustness on general categorical inputs, especially tabular categorical attributes, has received much less attention. To echo this challenge, our work aims to enhance the robustness of classification over categorical attributes against adversarial perturbations. We establish an information-theoretic upper bound on the expected adversarial risk. Based on it, we propose an adversarially robust learning method, named Integrated Gradient-Smoothed Gradient *(IGSG)*-based regularization. It is designed to smooth the attributional sensitivity of each feature and the decision boundary of the classifier to achieve lower adversarial risk, i.e., desensitizing the categorical attributes in the classifier. We conduct an extensive empirical study over categorical datasets of various application domains. The experimental results confirm the effectiveness of *IGSG*, which 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.

1 INTRODUCTION

While categorical data widely exist in real-world safety-critical applications,much less research attention has been attracted to evasion attack and defense with categorical inputs, compared to the efforts with continuous data, e.g. images. It thus becomes a must to develop *adversarially robust learning paradigms* to harden ML systems with categorical inputs.Previous research on adversarially robust learning has mainly focused on enhancing the resilience of target classifiers against L_Q and L_∞ adversarial perturbations (Goodfellow et al., 2016; Madry et al., 2017; Moosavi-Dezfooli et al., 2019; Attias et al., 2019; Yin et al., 2019; Shafahi et al., 2019; Zhang et al., 2019; Wong et al., 2020; Bashivan et al., 2021; Zhang et al., 2022). However, when dealing with categorical data, the conventional Euclidean space framework used for continuous measurements, such as pixel intensities, is not a natural fit. Categorical variables like *race* and *occupation* have non-continuous and unordered qualitative values that cannot be combined in Cartesian products or ordered numerically.Thus, L_0-norm bounded adversarial perturbations are commonly employed to assess the robustness of categorical data (Lei et al., 2019; Bao et al., 2021).

Adversarial training (Madry et al., 2017) stands out as a predominant defense strategy in the continuous domain. However, adversarial training on categorical data poses a challenging Mixed Integer Nonlinear Programming (MINLP) problem (Lee & Leyffer, 2011). It involves the iterative generation of adversarial training samples within the categorical feature space, followed by model retraining using these adversarial samples in an alternating sequence. The exponential growth of the categorical adversarial space with increasing amounts of categorical features complicates the generation of adversarial samples via heuristic search like Brand-and-Bound (Pataki et al., 2010). In Section.3.1, we identify that exploring the categorical adversarial space leads to insufficient coverage, causing a distribution gap between adversarial training and future attacks, resulting in "robust overfitting" on categorical data (Rice et al., 2020). Encoding categorical features as one-hot vectors and relaxing the adversarial training to the continuous domain, treating one-hot vectors as probabilistic representations, partially mitigates categorical data complexities. However, this approach encounters a bottleneck— the non-convex and highly non-linear nature of the relaxed adversarial training objective, stemming from the bi-level mini-max training and deep neural network architectures. Consequently, the approximated solution lacks a bounded integrality gap to the original discrete adversarial training problem, failing to guarantee optimality in the categorical feature space (Nohra et al., 2021). Thus, classifiers trained this way remain vulnerable to discrete adversarial samples in the combinatorial space. As empirically confirmed in Table.1, an MLP-based classifier tuned with the relaxed PGD-based adversarial training remains highly vulnerable to the state-of-the-art discrete adversarial attacks.

An alternative solution involves adversarial training within the embedding space of categorical variables. For instance, text classifiers can be defended using adversarial perturbations confined to the L_Q ball around the target word in its embedding space (Zhu et al., 2019; Li et al., 2021; Pan et al., 2022). While effective for text-related tasks, this approach is unsuitable for general categorical data, such as system logs in cyber intrusion detection or medical examination records, lacking a meaningful embedding space. Additionally, domain-specific constraints crucial for adversarial perturbations, like synonymous words and semantic similarity measures, may be undefined or inapplicable across various categorical domains.

Considering the limitations of the discussed solutions, we seek an alternative strategy to mitigate the adversarial risk with categorical inputs. We focus on enforcing smoothness regularization on the target classifier (Ross & Doshi-Velez, 2018a; Finlay & Oberman, 2021). Specifically, our strategy first involves *penalizing the input gradients*. According to the information-theoretic upper bound on the expected adversarial risk on categorical data detailed in Section.3.2, penalizing input gradients mitigates the excessive curvature of the classification boundary and reduces the generalization gap of the target classifier. As a result, it alleviates the classifier's over-sensitivity to input perturbation. However, our comprehensive analysis indicates that merely penalizing the input gradient is not sufficiently secured. An additional influential factor is *the excessive reliance on specific features*, where a few features contribute significantly more to the decision output than others. The adversary may choose to perturb these dominant features to significantly mislead the classifier's output. To mitigate this, we propose to perform a Total-Variation (TV) regularization (Chambolle, 2004) on the integrated gradients (IG) of one-hot encoded categorical features. This evens the attribution from different features to the classification output. While IG is widely accepted as an XAI method to interpret feature-wise attribution to the classifier's decision output, our work is the first to uncover theoretically and empirically the link between smoothing the axiomatic attribution and improving adversarial robustness of the target classifier with categorical inputs. Combining both smoothing-driven regularization techniques, we propose Integrated Gradient-Smoothed Gradient (*IGSG*)-based regularization, effectively improving the adversarial robustness of the model. As shown in Figure.1, the IGSG-trained model demonstrates approximately evenly distributed IG scores for different categorical features. In contrast, the undefended model (*Std Train*) exhibits a highly skewed distribution of IG scores across features. Connecting Figure.1 with Figure.3, we observe that highly attacked features are precisely those with high IG scores. In summary, IGSG jointly smooths the classification boundary and desensitizes categorical features. It therefore prevents adversarial attacks from exploiting the over-sensitivity of the target classifier to the adversarial inputs.

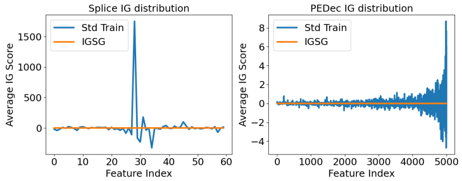


Figure 1: IG score distribution from the *IGSG* trained model and the undefended model *Std Train* on *Splice* and *PEDec* dataset.

Our technical contributions are summarized in the following perspectives:

Understanding influencing factors of adversarial risk: We've developed an information-theoretic upper bound to understand and minimize the expected adversarial risk on categorical data, providing insight into influential factors that can suppress adversarial risks effectively.

Development of a model-agnostic robust training through regularized learning for categorical features. We've reframed adversarial robustness, proposing IGSG, a method focused on minimizing our information-theoretic bound, enhancing feature contribution smoothness and decision boundary definitiveness during training. It's a universally adaptable solution for models dealing with categorical features.

Extensive experimental study. We've conducted thorough analyses comparing IGSG against the state-of-the-art adversarially robust training methods on three categorical datasets. The experimental results confirm the superior performances of models trained via IGSG.

2 Related Works

Adversarial training employs min-max optimization, generating adversarial samples via Fast Gradient Sign Method (FGSM) (Wong et al., 2020; Zhang et al., 2022) or Projected Gradient Descent (PGD) (Madry et al., 2017). TRADES (Zhang et al., 2019) optimizes a regularized surrogate loss, balancing accuracy and robustness. Adversarial Feature Desensitization (AFD) (Bashivan et al., 2021) leverages a GAN-like loss to learn invariant features against adversarial perturbations. While these methods can handle L_1-norm bounded adversaries for relaxed categorical data, ensuring consistent performance is uncertain. The challenge of ”robust overfitting” in adversarial training (Rice et al., 2020) is addressed 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. Notably, our proposed IGSG successfully mitigates this problem.

Adversarial learning for categorical data typically involves search-based methods (Lei et al., 2019; Wang et al., 2020b; Bao et al., 2021; Li et al., 2018; Jin et al., 2020). However, the substantial time cost of generating adversarial samples hinders widespread application to general categorical data tasks, as seen in cybersecurity and medical services. Xu et al. (2023) suggested extending adversarial methods from continuous to discrete domains, but the MINLP nature of adversarial training poses challenges in generating sufficient samples for comprehensive defense. In text data, Ren et al. (2019) used word saliency and classification probability for guided word replacement, while methods like FreeLB Zhu et al. (2019); Li et al. (2021) applied multiple PGD steps to word embeddings. Dong et al. (2021) modeled the attack space as a convex hull of word vectors, and Wang et al. (2020a) enhanced BERT-based model robustness using information theory, often relying on language-specific constraints, limiting their broader applicability.

Regularization-based methods offer an alternative approach for enhancing adversarial robustness by penalizing the target classifier’s complexity. Previous works Smilkov et al. (2017); Ross & Doshi-Velez (2018b); Finlay & Oberman (2021) proposed gradient magnitude regularization during training. Others Gu & Rigazio (2014); Jakubovitz & Giryes (2018); Hoffman et al. (2019) focused on penalizing the Frobenius norm of the Jacobian matrix for smoother classifier behavior. Additionally, Chen et al. (2019); Sarkar et al. (2021) suggested using Integrated Gradients (IG) for feature contribution measurement and applying regularization over IG to enhance robustness. Notably, these methods did not specifically target adversarial robustness. Our work reveals the effectiveness of IG-based regularization in adversarial robust training. Importantly, we demonstrate the significance of simultaneously regularizing gradient magnitude and IG distribution across different feature dimensions for a more potent approach.

3 Understanding the Influencing Factors of Adversarial Risk

Preliminary. Let’s assume that a random sample $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,p}\}$ 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 for all other $b(x_i)_{j,k \neq k^*} = 0$ $(k = 1, 2, ..., m)$. An adversarial sample $\hat{x}_i = \{\hat{x}_{i,j,j=1,\dots,p}\}$ 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 $\textit{diff}(x_i, \hat{x}_i)$. 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.

3.1 Limitations of Adversarial Training on Categorical Data

Firstly, we 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.1. With the attack budget 5 (i.e., $\textit{diff}(x_i, \hat{x}_i) \leq 5$), both Forward Stepwise Greedy Search (FSGS) (Elenberg et al., 2018), and orthogonal matching pursuit based greedy search (OMPGS) (Wang et al., 2020b) can directly find attack samples $\hat{x}_i$. PGD attack in the 1-norm setting (PGD-1) (Madry et al., 2017) locates

Table 1: 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%	×

attack samples and subsequently discretizes them to yield feasible adversarial samples $\hat{x}_i$. Table.1 show 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.

Similar observations can be made for f when using OMPGS-based adversarial training (see Figure.4 in Appendix.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.1 and Figure.4 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 Wassernstein distance between the distributions of adversarial samples generated by PGD-1 and OMPGS on PGD/OMPGS-based adversarially trained model respectively (detailed in Appendix.F). A greater Wasserstein distance suggests a larger discrepancy between the two distributions. Two main observations are evident from Table.5. 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.1, 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.1 and Figure.4, 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 Appendix.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.

3.2 Information-theoretic bound of adversarial risk

Prior to developing our regularized learning approaches, we unveil the factors influencing adversarial risk for categorical data via the following analysis. We first define the adversarial risk.

Definition 1. *We consider a hypothesis space $\mathcal{H}$ and a non-negative loss function ℓ: $\mu_z \times \mathcal{H} \rightarrow R^+$. 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 diff$(x, \hat{x}) \leq \epsilon$. The expectation is taken over the distribution of the n training samples S^n and the classifier $f = \mathcal{A}(S^n)$.*

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

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. Similarly, we provide the empirical adversarial risk of f 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 $P_{f|S^n}$.*

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

Theorem 1. *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.3.*

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

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

The proof can be found in Appendix.A. We further discuss the tightness of Eq.3 in Appendix.A. In the adversary-free case where $\hat{z} = z$, we show in Appendix.A that the bound established in Eq.3 is reduced to a tight characterization of generalization error for a broad range of models, which was previously unveiled in (Zhang et al., 2021; Bu et al., 2019).

The information-theoretical adversarial risk bound established in Eq.3 unveils two major factors to suppress the adversarial risk over categorical inputs.

Factor 1. Reducing $I(f; z_i)$ for each training sample z_i helps suppress the adversarial risk f. $I(f, z_i)$ in Eq. 3 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 a lower value of $I(f, z_i)$ corresponds to a diminishing adversary-free generalization error. As widely acknowledged in adversarial learning research and emphasized in Eq. 3, a better generalizable classifier exhibits greater resilience to adversarial attacks, resulting in lower adversarial risk

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 contribution to classification, thus reducing the adversarial risk. We note that reducing the impact of excessively influential features can suppress adversarial risk, corresponding to minimizing the second and third terms beneath the square-root sign in Eq.3. **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, i.e. adversarially perturbing the values of these features is more likely to mislead the decision. Minimising $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ 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. In conclusion, jointly minimising $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ ensures that the classifier does not overly rely on a few highly sensitive features. It helps reduce the susceptibility of the classifier to adversarial perturbation targeting at these features, which consequently limits the adversarial risk.Beyond the two factors, **minimizing the empirical adversarial risk $\hat{\mathcal{R}}_f^{adv}$ in Eq.3 may also reduce the adversarial risk.** This concept is synonymous with the principles of adversarial training. Nevertheless, as highlighted in Section.3.1, the efficacy of adversarial training is restricted.

4 *IGSG*: Robust Training for Categorical Data

Our design of adversarially robust training is in accordance with two recommended factors to minimize the adversarial risk. However, it is challenging to derive consistent estimates of mutual information between high-dimensional variables, e.g. model parameters of deep neural networks and high-dimension feature vectors, due to the curse of dimensionality Gao et al. (2018). Directly optimizing the mutual information-based bound is thus impractical. To overcome this bottleneck, we propose the *IGSG*-based robust training paradigm. It jointly applies two smoothness-enhancing regularization techniques into the learning process of a classifier with categorical inputs, in order to mitigate the adversarial attack over categorical data.

Minimizing $I(f; z_i)$ by smoothing the curvature of the classification boundary. In previous work, Fisher information $\rho(z_i)_f$ was utilized as a quantitative measure of the information that the hypothesis f contains about the training sample z_i (Hannun et al., 2021). As shown in Wei & Stocker (2016), $\rho(z_i)_f$ is closely related to the mutual information $I(f; z_i)$, higher/lower $\rho(z_i)_f$ indicates higher/lower $I(f; z_i)$. Our work aims to minimize $\rho(z_i)_f$ to effectively penalize excessively high mutual information $I(f; z_i)$. The computation of $\rho(z_i)_f$ is detailed in Eq.16 of (Hannun et al., 2021). In this context, suppressing $\rho(z_i)_f$ (approximately suppressing $I(f; z_i)$) is equivalent to penalizing the magnitude of the gradient of the loss function with respect to each z_i. This approach, supported by findings in (Smilkov et al., 2017), uses gradient regularization to smooth the classifier's decision boundary, thereby reducing the potential risk of overfitting and enhancing adversarial resilience We calculate the gradient of the classification loss to the one-hot encoded representation of $b(x_i)$, which gives 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 formulated as $\frac{\partial}{\partial b(x_i)_{j,k}}\ell(x_i, y_i; \theta)$. According to (Yang et al., 2021), $\nabla_{b(x_i)}\ell(x_i, y_i; \theta)$ measures the curvature of the decision boundary around the input. A larger magnitude of $\nabla_{b(x_i)}\ell(x_i, y_i; \theta)$ indicates a more twisted decision boundary, thus a less stable decision around the input. Enforcing the regularization over the magnitude $\|\nabla_{b(x_i)}\ell(x_i, y_i; \theta)\|_q$ leads to a smoother decision boundary (with lower curvature) and improves the robustness of the decision output $f(x_i)$ against potential perturbation. In this work, we apply smoothed Gradient Regularization (SG) (Smilkov et al., 2017) to further boost the smoothness of the classifier.

Minimizing $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ via smoothing the distribution of feature-wise contribution to the classification output. Minimizing these terms involves evaluating the mutual information between the feature subset ω_i and the combined set of remaining features and the trained model f. Approximating this mutual information-based penalization with Fisher information is thus infeasible. The primary goal of regularizing these terms is to prevent the classifier from relying too heavily on a few influential features. To achieve this, we propose using Integrated Gradient (IG) (Sundararajan et al., 2017) to assess feature-wise contributions to the classification output. We apply Total-Variance (TV) regularization over the feature-wise Integrated Gradient to promote a smooth and balanced distribution of feature-wise attribution. In Appendix.I.1, we show empirically with toy models that performing the proposed TV regularization can reduce the estimated value of both mutual information-based terms.

We extend the computation of the IG scores in the categorical feature space by first defining a baseline input x'. We augment the set of optional category values for each feature $x_{i,j}$: *we add one dummy category* $m + 1$, with constantly all 0 values for the embedding vector in f. Each feature of x' is set to take the dummy category value, i.e., $b(x')_{j,m+1} = 1, b(x')_{j,k} = 0 (k = 1, 2, \ldots, m)$. By feeding $b(x')$ to the classifier, no useful information is conveyed for classification, making it a non-informative baseline. Given the defined baseline input x', the IG score of each categorical feature $x_{i,j}$ is approximated as:

$$IG(x_i)_j = \sum_{k=1}^{m} IG(x_i)_{j,k} = \sum_{k=1}^{m} (b(x_i)_{j,k} - b(x')_{j,k}) \times \frac{1}{T} \sum_{t=1}^{T} \frac{\partial f(b(x') + \frac{t}{T} \times [b(x_i) - b(x')])}{\partial b(x_i)_{j,k}} \tag{4}$$

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

To ensure a smooth and balanced distribution of IG scores and to mitigate excessive dependency on specific features, we propose to minimize the TV loss of the normalized IG scores, as influenced by prior work (Chambolle, 2004). Initially, we employ a softmax transformation to normalize the IG scores of each feature $x_{i,j}$, ensuring the normalized scores lie within $[0, 1]$ and collectively sum to 1. The TV regularization term is then defined as the sum of the absolute differences between neighboring features' normalized IG scores: $\ell_{TV} IG(x_i) = \sum_{j=1}^{p-1} |IG(x_i)_j - IG(x_i)_{j+1}|$, following the

TV loss used in time series data analysis (Chambolle, 2004). This minimization promotes a more balanced distribution of feature-wise contributions to the classifier's decision.

Combining Eq.18 in Appendix.E and $\ell_{TV}IG(x_i)$, the objective function of IGSG gives:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x_i,y_i)\in S^n} \ell(x_i, y_i; \theta) + \alpha \ell_{TV} IG(x_i) + \frac{\beta}{R}\sum_{r=1}^{R} ||G_r||_p$$
$$\text{where } G_{r,j,k} = \frac{\partial}{\partial b(x_r)_{j,k}}\ell(x_r, y_i; \theta) - \frac{\partial}{\partial b(x_r)_{j,k^*}}\ell(x_r, y_i; \theta) \quad (5)$$

where α and β are hyper-parameters set by cross-validation.

5 Experimental Evaluation

5.1 Experimental Setup

Summary of datasets. To evaluate the proposed IGSG algorithm, 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.
1) 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.
2) 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.
3) 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 and 9 numerical. The task is to determine whether subjects fall into the low-income (less than $50,000) 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.

Robustness evaluation protocol. Three domain-agnostic attack methods, FSGS (Elenberg et al., 2018), OMPGS (Wang et al., 2020b) and PCAA (Xu et al., 2023), 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 Appendix.G. FSGS, OMPGS and PCAA, with proven attack effectiveness across various real-world applications, are suitable for comparing the effectiveness of different robust model training methods on categorical input.

We traverse varied attack budgets (the maximum number of the modified features) for OMPGS attacks. 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, we also fix the attack budget to be 5. On each dataset, we use MLP and Transformer (Vaswani et al., 2017) as the target classifier. Due to space limitations, we provide detailed attack settings in Appendix.H.1, the experimental results on Transformer models in Appendix.I.3, and the experimental results of PCAA attack in Appendix.I.7

Baselines. We involve one undefended model and 7 state-of-the-art robust training methods as the baselines in the comparison with *IGSG*. Specifically, we include 5 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 2 regularization-based baselines IGR (Ross & Doshi-Velez, 2018b) and JR (Hoffman et al., 2019). The details of the baselines can be found in Appendix.H.2 and the details of the hyper parameter settings can be found in Appendix.H.3.

Performance metrics. We compare the *adversarial accuracy* of the target models trained using the methods above against FSGS and OMPGS attacks. We evaluate the adversarial robustness of mixed-type datasets by attacking categorical features with FSGS/OMPGS and numerical features with PGD-∞. Further details can be found in Appendix.H.4. Time complexity analysis and training time

Table 2: Adversarial Accuracy under FSGS attack and Accuracy (%) for *IGSG* and baseline models. 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, 2018b), JR (Hoffman et al., 2019)

Dataset	Attack	Undefended	Adversarial Training baselines					Regularization baselines		**Ours**
		Std Train	Adv Train	Fast-BAT	TRADES	AFD	PAdvT	IGR	JR	**IGSG**
Splice	budget=5	36.7±4.8	43.6±0.7	28.7±7.4	23.3±8.6	21.1±13.0	39.1±1.7	40.9±3.0	4.3±3.7	**44.0±2.6**
	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	budget=5	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**
	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	budget=5	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**
	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

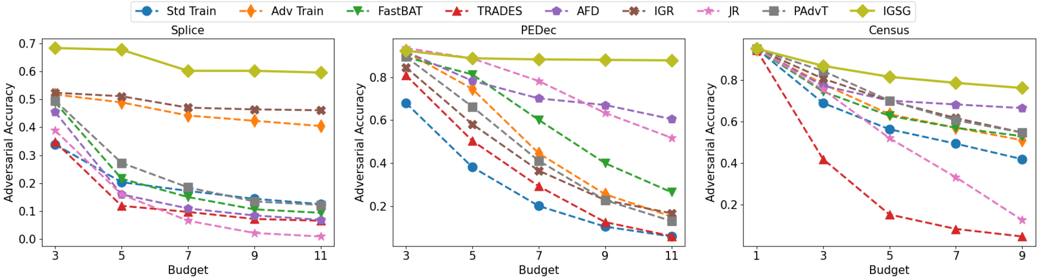


Figure 2: Adversarial accuracy for *IGSG* and baselines under OMPGS attack with varied budgets.

for different methods are provided in Appendix.I.5. The code is available at `https://github.com/fshafrh/IGSG`.

5.2 Experimental Results

Adversarial Accuracy Performance of *IGSG* Compared to Baseline Methods. Table.2 reports the accuracy and the adversarial accuracy against FSGS attacks for each robust training method. From the results, we can see that the adversarial accuracy of *IGSG* significantly outperforms the baseline methods. Especially, on *PEDec*, *IGSG* 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%. *IGSG* also achieves comparable accuracy on the three datasets.

Figure 2 illustrates the adversarial accuracy of all the methods tested under OMPGS-based 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 method, IGSG, consistently achieves higher and more stable levels of adversarial accuracy across all three datasets. Specifically, on *PEDec*, IGSG maintains an adversarial accuracy above 88% regardless of the attack strength. On *Splice*, IGSG consistently outperforms other baseline methods, exhibiting a performance gain of over 10%. On *Census*, IGSG 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* 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* 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 the defense methods involved in the test do not perform well against attacks on *Splice*.

Ablation Study. We include the following variants of the proposed *IGSG* method in the ablation study. ***SG*** and ***IG*** are designed to preserve only the smoothed gradient-based (*SG*, see Eq.18) or the IG-based smoothness regularization (*IG*, see Eq.17) respectively in the learning objective. We compare *SG* and *IG* to *IGSG* for demonstrating the advantage of simultaneously performing the IG and gradient smoothing-based regularization. ***IGSG-VG***: We replace the smoothed gradient given in Eq.18 with the vanilla gradient of the one hot tensor. Another four variants to provide additional validation for the design of *IGSG* are presented in Appendix.I.6

Table.3 shows that *IGSG* consistently outperforms the variants in adversarial accuracy against both FSGS and OMPGS attacks, affirming the effectiveness of *IGSG*'s design in mitigating both types of greedy search-based attacks simultaneously. *SG* 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* lacks the classification boundary smoothness, leading to a slight decrease in performance compared to *IGSG*. The results with *SG* and *IG* show that the two attributional smoothness regularization terms employed by *IGSG* are complementary to each other in improving the adversarial robustness of the built model.

Table 3: Ablation Study. Adversarial Accuracy and Accuracy (%) for *IGSG* variants with an attack budget of 5.

Dataset	Adversary	SG	IG	IGSG-VG	IGSG
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

IGSG-VG replaces the smoothed gradient-based regularization defined in Eq.18 and Eq.19 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.18 and Eq.19.

Effectiveness of Avoiding Robust Overfitting. By utilizing regularization, IGSG avoids the issue of "robust overfitting" encountered in adversarial training. This results in improved performance, as demonstrated in Table.4, compared to the adversarial accuracy shown in Table.1. We conduct the comparison between IGSG and two works mitigating robust overfitting in continuous domain (Chen et al., 2020; Yu et al., 2022). IGSG achieves consistently better adversarial robustness. The details are presented in Appendix.I.4

Table 4: MLP with IGSG training and Performance Gain Compared to PGD-based Adversarial Training

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% ↑
Census	PGD-1	93.0%	-0.2% ∼
	OMPGS	71.3%	8.6% ↑
	FSGS	67.2%	13.1% ↑

Reduced Attack Frequency with IGSG. We compare the frequency of each feature attacked under OMPGS on *Splice* and *PEDec*. The attack frequency represents the number of times a feature appears among the altered features in all successful adversarial attack samples. As seen in Figure.3, *IGSG* results in fewer and lower peaks on *Splice* compared to the undefended model, indicating enhanced robustness. For *PEDec*, the feature with the highest attack frequency is entirely suppressed with *IGSG*. This demonstrates the effectiveness of *IGSG*, with feature desensitization being achieved post-training.

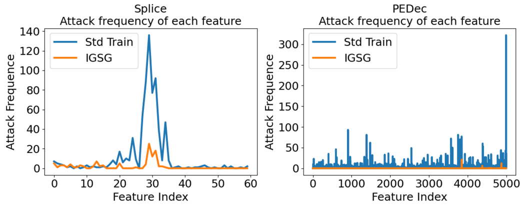


Figure 3: Attack frequency reduced by IGSG

6 Conclusion

In this work, we first unveil influencing factors of adversarial threats on categorical inputs via developping an information-theoretic upper bound of the adversarial risk. Guided by the theoretical analysis, we further propose *IGSG*-based adversarially robust model training via enforcing the two smoothness regularization techniques on categorical data, which helps mitigate adversarial attacks on categorical data. On the one hand, our method smooths the influence of different categorical features and makes different features contribute evenly to the classifier's output. On the other hand, our method smooths the decision boundary around an input discrete instance by penalizing the gradient magnitude. We demonstrate the domain-agnostic use of *IGSG* 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.

REFERENCES

Amir R. Asadi, Emmanuel Abbe, and Sergio Verdú. 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.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In Aurélien Garivier and Satyen Kale (eds.), *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pp. 162–183. PMLR, 22–24 Mar 2019. URL `https://proceedings.mlr.press/v98/attias19a.html`.

Hongyan Bao, Yufei Han, Yujun Zhou, Yun Shen, and Xiangliang Zhang. Towards understanding the robustness against evasion attack on categorical data. In *International Conference on Learning Representations*, 2021.

Pouya Bashivan, Reza Bayat, Adam Ibrahim, Kartik Ahuja, Mojtaba Faramarzi, Touraj Laleh, Blake Richards, and Irina Rish. Adversarial feature desensitization. *Advances in Neural Information Processing Systems*, 34:10665–10677, 2021.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Yuheng Bu, Shaofeng Zou, and Venugopal V. Veeravalli. 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.

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

Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust attribution regularization. *Advances in Neural Information Processing Systems*, 32, 2019.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020.

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

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. Towards robustness against natural language word substitutions. *arXiv preprint arXiv:2107.13541*, 2021.

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

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

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k -nearest neighbor information estimators. *IEEE Transactions on Information Theory*, 64(8):5629–5661, 2018. doi: 10.1109/TIT.2018.2807481.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Awni Y. Hannun, Chuan Guo, and Laurens van der Maaten. 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`.

Laurent Herault and Radu Horaud. Smooth curve extraction by mean field annealing. *Ann. Math. Artif. Intell.*, 13:281–300, 09 1995. doi: 10.1007/BF01530832.

Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8018–8025, 2020.

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

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

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alex Dimakis, Inderjit S Dhillon, and Michael J Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Proceedings of Machine Learning and Systems*, 1:146–165, 2019.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. Searching for an effective defender: Benchmarking defense against adversarial word substitution. *arXiv preprint arXiv:2108.12777*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9070–9078, 2019. doi: 10.1109/CVPR.2019.00929.

Carlos J. Nohra, Arvind U. Raghunathan, and Nikolaos Sahinidis. Spectral relaxations and branching strategies for global optimization of mixed-integer quadratic programs. *SIAM Journal on Optimization*, 31(1):142–171, 2021. doi: 10.1137/19M1271762. URL `https://doi.org/10.1137/19M1271762`.

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

Lin Pan, Chung-Wei Hang, Avirup Sil, and Saloni Potdar. Improved text classification via contrastive adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11130–11138, 2022.

Gábor Pataki, Mustafa Tural, and Erick B. Wong. Basis reduction and the complexity of branch-and-bound. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pp. 1254–1261, USA, 2010. Society for Industrial and Applied Mathematics. ISBN 9780898716986.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1085–1097, 2019.

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

Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018a. doi: 10.1609/aaai.v32i1.11504. URL https://ojs.aaai.org/index.php/AAAI/article/view/11504.

Andrew Ross and Finale Doshi-Velez. 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, 2018b.

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

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

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

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

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329*, 2020a.

Yutong Wang, Yufei Han, Hongyan Bao, Yun Shen, Fenglong Ma, Jin Li, and Xiangliang Zhang. 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, 2020b.

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

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

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

Han Xu, Pengfei He, Jie Ren, Yuxuan Wan, Zitao Liu, Hui Liu, and Jiliang Tang. Probabilistic categorical adversarial attack and adversarial training. In *International Conference on Machine Learning*, pp. 38428–38442. PMLR, 2023.

Zhuo Yang, Yufei Han, and Xiangliang Zhang. 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.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7085–7094. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yin19b.html.

Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (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`.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Jingwei Zhang, Tongliang Liu, and Dacheng Tao. 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.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pp. 26693–26712. PMLR, 2022.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*, 2019.

A Proof to Theorem.1

Definition 2. ***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:*

$$\mathcal{D}_f = \sup_{f, f' \in \mathcal{H}} d(f, f') \tag{6}$$

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

Definition 3. ***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 $\mathcal{H}$:*

$$|\ell(f(x_i), y_i) - \ell(f'(x_i), y_i))| \leq L\, d(f, f') \tag{7}$$

Proof to Eq.3: 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 $\arg\max_{\text{diff}(\hat{z}_i, z_i) \leq \epsilon} \ell(f(x_i), y_i)$ 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:

$$\mathrm{E}_f[\mathcal{R}_f^{adv}] \leq \mathrm{E}_f[\hat{\mathcal{R}}_f^{adv}] + L\,\mathcal{D}_f\, \mathbb{TV}(P_f \times \hat{\mu}_{\hat{z}}, P_{f \times \hat{z}_i}) \tag{8}$$

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.8 by letting $z = z_i$ (i=1,2,3,...,n) and using mutual information between f and $\hat{z}_i$:

$$\begin{aligned}\mathrm{E}_f[\mathcal{R}_f^{adv}] &\leq \mathrm{E}_f[\hat{\mathcal{R}}_f^{adv}] + \frac{L\,\mathcal{D}_f}{\sqrt{2}n}\sqrt{\sum_{i=1}^{n} I(f; \hat{z}_i)} \\ &\leq \mathrm{E}_f[\hat{\mathcal{R}}_f^{adv}] + \frac{L\,\mathcal{D}_f}{\sqrt{2}n}\sqrt{\sum_{i=1}^{n} I(f; z_i) + \sum_{i=1}^{n}(I(f; \hat{z}_i) - I(f; z_i))}\end{aligned} \tag{9}$$

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.

$$\begin{aligned}
& I(f; \hat{z}_i) - I(f; z_i) \\
=\; & I(f; \hat{x}_{i,\omega_i}, x_{i,\overline{\omega}_i}, y_i) - I(f; x_{i,\omega_i}, x_{i,\overline{\omega}_i}, y_i) \\
=\; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + I(\hat{x}_{i,\omega_i}; f | x_{i,\overline{\omega}_i}, y_i) - I(x_{i,\overline{\omega}_i}, y_i; f | x_{i,\omega_i}) \\
=\; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + H(\hat{x}_{i,\omega_i} | x_{i,\overline{\omega}_i}, y_i) + H(f | x_{i,\overline{\omega}_i}, y_i) - H(\hat{x}_{i,\omega_i}, f | x_{i,\overline{\omega}_i}, y_i) \\
& - H(x_{i,\overline{\omega}_i}, y_i | x_{i,\omega_i}) - H(f | x_{i,\omega_i}) + H(x_{i,\overline{\omega}_i}, y_i, f | x_{i,\omega_i}) \\
=\; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + H(\hat{x}_{i,\omega_i}) - I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) + H(f) - I(x_{i,\overline{\omega}_i}, y_i; f) \\
& - H(x_{i,\overline{\omega}_i}, y_i) + I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) - H(f) + I(x_{i,\omega_i}; f) \\
& - H(\hat{x}_{i,\omega_i}, f | x_{i,\overline{\omega}_i}, y_i) + H(x_{i,\overline{\omega}_i}, y_i, f | x_{i,\omega_i}) \\
=\; & I(x_{i,\overline{\omega}_i}, y_i; f) - I(x_{i,\omega_i}; f) + H(\hat{x}_{i,\omega_i}) - I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) + H(f) - I(x_{i,\overline{\omega}_i}, y_i; f) \\
& - H(x_{i,\overline{\omega}_i}, y_i) + I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) - H(f) + I(x_{i,\omega_i}; f) \\
& - H(f | x_{i,\overline{\omega}_i}) - H(\hat{x}_{i,\omega_i} | x_{i,\overline{\omega}_i}, f) + H(x_{i,\overline{\omega}_i}, f | x_{i,\omega_i}) \\
\leq\; & 2|I(x_{i,\omega_i}; f) - I(x_{i,\overline{\omega}_i}, y_i; f)| + |I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i, f) \\
& - I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i, f)| + |I(\hat{x}_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i) - I(x_{i,\omega_i}; x_{i,\overline{\omega}_i}, y_i)|
\end{aligned} \tag{10}$$

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 $\alpha = \underset{z_i=(x_i,y_i)\in S^n}{max} 1 + \frac{|I(\hat{x}_{i,\omega_i};x_{i,\overline{\omega_i}},y_i)-I(x_{i,\omega_i};x_{i,\overline{\omega_i}},y_i)|}{|I(\hat{x}_{i,\omega_i};x_{i,\overline{\omega_i}},y_i,f)-I(x_{i,\omega_i};x_{i,\overline{\omega_i}},y_i,f)|}$ to Eq.10, we can derive Eq.3.

We discuss about the tightness of the bound in Eq.3 from the following perspectives. First, we show this bound reduces to a 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 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 4. *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 $\mathcal{R}_f$, gives in Eq.11. The expectation is taken over the distribution of the n training samples S^n and the classifier $f = \mathcal{A}(S^n)$.*

$$\mathcal{R}_f = \underset{S^n,P_{f|S^n}}{\mathbb{E}} \underset{z=(x,y)\sim\mu_z,}{\mathbb{E}} \ell(f(x),y). \tag{11}$$

Similarly, we provide the empirical risk of f under the adversary-free scenario in Eq.12. It is taken as the expectation over the distribution of the n training samples and the classifier.

$$\hat{\mathcal{R}}_f = \underset{S^n,P_{f|S^n}}{\mathbb{E}} \frac{1}{n} \sum_{z_i=(x_i,y_i)\in S^n} \ell(f(x_i),y_i) \tag{12}$$

With the adversary-free setting, $\hat{x} = x$. This makes $\Phi(x_{i,\omega_i},\hat{x}_{i,\omega_i})$ vanish as $I(\hat{x}_{i,\omega_i};x_{i,\bar{\omega}_i},y_i,f) = I(x_{i,\omega_i};x_{i,\overline{\omega_i}},y_i,f)$. 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.3 shrinks to the following form in Eq.13:

$$\mathcal{R}_f - \hat{\mathcal{R}}_f \leq \frac{\sqrt{3}\,L\,\mathcal{D}_f}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} I(f;z_i)}. \tag{13}$$

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:

$$\mathcal{R}_f - \hat{\mathcal{R}}_f \leq \frac{1}{n}\sum_{i=1}^{n} \sqrt{2R^2 I(f;z_i)}. \tag{14}$$

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.13 and Eq.14 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.13 and Eq.14) 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.3 is bounded. The possible value of $\Phi(x_{i,\omega_i},\hat{x}_{i,\omega_i}) = |I(\hat{x}_{i,\omega_i};x_{i,\bar{\omega}_i},y_i,f) - I(x_{i,\omega_i};x_{i,\overline{\omega_i}},y_i,f)|$ 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:

$$\begin{aligned}\Phi(x_{i,\omega_i},\hat{x}_{i,\omega_i}) &\leq \log(q\epsilon)\\ \Psi(x_{i,\omega_i},x_{i,\overline{\omega_i}}) &\leq I(z_i;f)\end{aligned} \tag{15}$$

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.3 is bounded from above as:

$$\begin{aligned}\mathcal{R}_f^{adv} - \hat{\mathcal{R}}_f^{adv} \leq & \frac{L\,\mathcal{D}_f}{\sqrt{2n}} \sqrt{\sum_{i=1}^{n} I(f;z_i) + 2\sum_{i=1}^{n} \Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}}) + \sum_{i=1}^{n} \Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})} \\ \leq & \frac{L\,\mathcal{D}_f}{\sqrt{2n}} \sqrt{\sum_{i=1}^{n} 3I(f;z_i) + n\log(q\epsilon)}\end{aligned} \tag{16}$$

In Eq.16, 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.13. 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.16 gives a bounded but rough estimate of the adversarial risk, as not all of the features are useful for attack. Only the perturbation over influential features may cause effectively the rise of adversarial risk. In this sense, Eq.3 provides more accurate estimate to the actual adversarial risk than Eq.16.

B Connection between the theoretical analysis and the design of IGSG

Our design of adversarially robust training is in accordance with two recommended factors to minimize the adversarial risk. However, deriving consistent and differentiable estimates of mutual information between high-dimensional variables, such as the parameters of deep neural networks and input categorical feature vectors, remains an open and challenging problem due to the curse of dimensionality Gao et al. (2018). This makes direct optimization of the mutual information-based bound impractical. To reach this goal, we propose the *IGSG*-based robust training paradigm. It jointly applies two smoothness-enhancing regularization techniques into the learning process of a classifier with categorical inputs, in order to mitigate the adversarial attack over categorical data. We discuss the design of IGSG in the followings. To further confirm the effectiveness of *IGSG* in minimizing the mutual-information-based adversarial risk bound, we provide approximated computation of the mutual-information based bound with the toy model in Appendix.I.1. We derive the estimated bound value derived with and without applying our proposed robust training mechanism. The empirical observations show that enforcing the two regularization terms indeed decreases the estimated value of the bound, which echoes the rise of adversarial accuracy.

Minimizing $I(f;z_i)$ by smoothing the curvature of the classification boundary. In previous work, Fisher information $\rho(z_i)_f$ was utilized as a quantitative measure of the information that the hypothesis f contains about the training sample z_i (Hannun et al., 2021). As shown in Wei & Stocker (2016), $\rho(z_i)_f$ is closely related to the mutual information $I(f;z_i)$, higher/lower $\rho(z_i)_f$ indicates higher/lower $I(f;z_i)$. Our work aims to minimize $\rho(z_i)_f$ to effectively penalize excessively high mutual information $I(f;z_i)$. The computation of $\rho(z_i)_f$ is detailed in Eq.16 of (Hannun et al., 2021). In this context, suppressing $\rho(z_i)_f$ (approximately suppressing $I(f;z_i)$) is equivalent to penalizing the magnitude of the gradient of the loss function with respect to each z_i. This approach, supported by findings in (Smilkov et al., 2017), uses gradient regularization to smooth the classifier's decision boundary, thereby reducing the potential risk of overfitting and enhancing adversarial resilience

Minimizing $\Psi(x_{i,\omega_i}, x_{i,\overline{\omega_i}})$ and $\Phi(x_{i,\omega_i}, \hat{x}_{i,\omega_i})$ via smoothing the distribution of feature-wise contribution to the classification output. Minimizing these terms involves evaluating the mutual information between the feature subset ω_i and the combined set of remaining features and the trained model f. Approximating this mutual information-based penalization with Fisher information is thus infeasible. The primary goal of regularizing these terms is to prevent the classifier from relying too heavily on a few influential features. To achieve this, we propose using Integrated Gradient (IG) (Sundararajan et al., 2017) to assess feature-wise contributions to the classification output. We apply Total-Variance (TV) regularization over the feature-wise Integrated Gradient to promote a smooth and balanced distribution of feature-wise attribution. In Appendix.I.1, we show empirically with toy models that performing the proposed TV regularization can reduce the estimated value of both mutual information-based terms.

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 does 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 roots 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.3 thus determine 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 Detailed Design of Integrated Gradient and Smoothed Gradient Regularization

Definition 5. *(**Total Variation of IG-based Regularization**). The objective function of the classifier f with TV loss is defined as,*

$$\min_{\theta} \mathop{\mathbb{E}}_{(x_i,y_i)\sim\mu_z} \{\ell(x_i, y_i; \theta) + \alpha \ell_{TV} IG(x_i)\} \tag{17}$$

where α is a hyper-parameter tuning the weight of the TV regularization term, and θ is the parameters of f. $\ell(x_i, y_i; \theta)$ is the learning loss of f, e.g. the cross entropy loss function. $\ell_{TV}(\cdot)$ denotes the TV loss of the IG scores of x_i. We follow the implementation of the TV loss over time series data, i.e. $\ell_{TV} IG(x_i) = \sum_{j=1}^{p-1} |IG(x_i)_j - IG(x_i)_{j+1}|$.

In $\ell_{TV}(\cdot)$, we normalize the IG scores of each feature $x_{i,j}$ with softmax transformation. Therefore, the normalized IG score of each feature is valued within $[0, 1]$ and sums up to 1. By minimizing the regularization term based on TV loss, the distribution of the IG scores is driven to be as uniform as possible.

Definition 6. ***(Smoothed Gradient Regularization).*** *With R randomly sampled data points $x_1, x_2, \ldots, x_R$ around the input instance x_i, the gradient smoothing-oriented regularization term defined on x_i is given as follows:*

$$\min_{\theta} \underset{(x_i,y_i)\sim\mu_z}{\mathbb{E}} \ell(x_i, y_i; \theta) + \frac{\beta}{R}\sum_{r=1}^{R} ||G_r||_q \tag{18}$$

where β is a hyper-parameter, and $G_r \in \mathbb{R}^{p*m}$ is a gradient matrix with:

$$G_{r,j,k} = g_{r,j,k} - g_{r,j,k^*} \tag{19}$$

where $g_{r,j,k} = \frac{\partial}{\partial b(x_r)_{j,k}}\ell(x_r, y_i; \theta)$. We use L_Q norm to calculate the norm of the gradient. Specifically, we choose q=2 for all the experiments following (Ross & Doshi-Velez, 2018b).

We next elaborate on the details of the calculation in Eq.18 for categorical data, since it is different from the vanilla smoothed gradient computing with continuous input (Smilkov et al., 2017). **First**, we choose the categorical instances $\{x_r, r = 1...R\}$ by randomly changing a few features of x_i, such that $|diff(x_i, x_r)|$ equals to the attack budget ϵ. By taking the gradients associated with $\{x_r, r = 1...R\}$ that are similar categorical vectors to x_i, we aim to obtain a more accurate measurement of the smoothness of the decision boundary around x. We average the magnitudes of the gradient vectors of $\{x_r, r = 1...R\}$ for each categorical instance x_i. Empirically, we choose $R = 5$, which brings consistently good results without very high time complexity. **Second**, instead of using the vanilla gradient, we inherit the idea of mean field smoothing as defined in Eq.10 and 13 of (Herault & Horaud, 1995) over the gradient values associated with each categorical feature of x_r. As shown in Eq.19, for each feature of x_r (noted as $x_{r,j}$), we minimize the norm of the difference between $g_{r,j,k}$ and g_{r,j,k^*}, where k^* denotes the category value carried by $x_{r,j}$. It is formulated as minimizing the L_Q norm of the difference of gradients G_r in Eq.18. By optimizing with the regularization term, our aims are two-fold: a) We suppress the magnitude of the gradient with respect to each categorical feature $x_{r,j}$ to reduce the adversarial risk. b) We smooth the distribution of the gradient values $g_{r,j,k}$ associated with the optional category values of each categorical feature $x_{r,i}$. Domain-agnostic discrete attacks, e.g., Orthogonal Matching Pursuit Greedy Search (OMPGS) (Wang et al., 2020b), rank the gradient values associated with the one-hot encoded vector $x_{r,i}$. The top-ranked category values other than $x_{r,i} = k^*$ are selected by OMPGS as the candidates of feasible adversarial perturbation to replace $x_{r,i} = k^*$. Minimizing the difference between$g_{r,j,k}$ and g_{r,j,k^*} produces a set of uniformly distributed gradient values $g_{r,j,k}$. It prevents gradient-guided attack methods like OMPGS from identifying promising candidates for generating effective adversarial perturbation.

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 distribution gap is conducted by comparing the following 4 groups of Wassernstein distance scores.

Wassernstein distance between in-distribution samples ($\mathbf{WD_{in}}$): We first measure the Wassernstein 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 Wassernstein distance between the probe and gallery set. This process is repeated for 20 times. We record all the Wassernstein distance scores to measure the distribution gap between in-distribution adversarial samples within each set. $\mathrm{WD_{in}}$ is considered as a baseline. We expect the Wassernstein distance scores between adversarial samples from different distributions (Out-Of-Distribution) to be significantly larger than the distance scores in $\mathrm{WD_{in}}$.

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

Wassernstein distance between the training and testing adversarial samples produced by the OMPGS-based method ($\mathbf{WD^{O}_{out}}$): For O_{tr} and O_{te}, we randomly sample 90% of the samples from each set and compute the Wassernstein 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 5: Average and standard (AVG) deviation (STD) of the Wassernstein distance scores

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

Wassernstein distance between the training and testing adversarial samples produced by the PGD-based and OMPGS-based attack methods ($\mathbf{WD^{PO}_{out}}$): 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 Wassernstein 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 Wassernstein distance scores to assess the distribution difference between training and testing adversarial samples generated using different attack methods.

In Table.5, the averaged Wassernstein scores of $\mathrm{WD_{in}}$ and $\mathrm{WD^{P}_{out}}$ are the smallest among the four groups of distance values. Conversely, $\mathrm{WD^{PO}_{out}}$ and $\mathrm{WD^{O}_{out}}$ 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 $\mathrm{WD_{in}}$ and $\mathrm{WD^{P}_{out}}$. 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.1. Second, we conduct Mann-Whitney U tests between $\mathrm{WD_{in}}$ and $\mathrm{WD^{O}_{out}}$, as well as between $\mathrm{WD_{in}}$ and $\mathrm{WD^{PO}_{out}}$. The hypothesis tests reveal that $\mathrm{WD^{O}_{out}}$ and $\mathrm{WD^{PO}_{out}}$ are significantly higher than $\mathrm{WD_{in}}$, with p-values of 0.02 and 0.01, respectively. This 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 Figure.4.

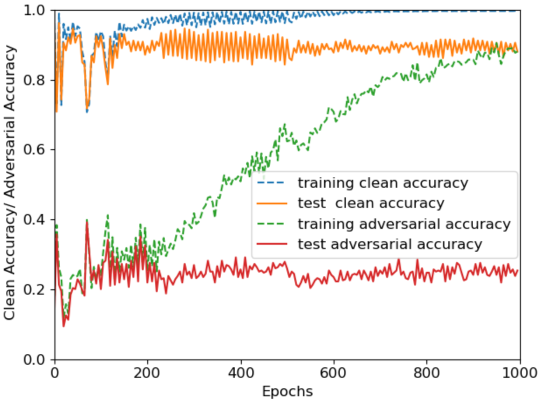


Figure 4: The "robust overfitting" of adversarially trained MLP on *Splice*.

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., 2020b). 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 volume 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 relying on the relaxed solution to the discrete attack 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, 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 employe the FSGS attack and OMPGS attack, shown in Algorithm.1 and 2. The definition of the notations can be found in Appendix.H.4. It's also worth noting

Algorithm 1 FSGS for general categorical data

```
**Input:** The candidate set $H = \{1, 2, ...p\}$ of all categorical features, categorical attack budget $\epsilon$
$S \leftarrow \emptyset$
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
        $m_f(x(j)) = \underset{s \subset S, |s| < \epsilon}{max} m_f(\hat{x}(j, s))$
    end for
    $m_f(x, S) = \underset{j \in H/S}{max} m_f(x(j))$
    $j^* = \underset{j \in H/S}{\arg\max} m_f(x(j))$
    $S \leftarrow S \cup \{j^*\}$
    if $m_f(x, S) \geq 0$ then attack successfully
    if $Time \geq \Gamma$ then timeout
end for
```

Algorithm 2 OMPGS for general categorical data

```
**Input:** The candidate set $H = \{1, 2, ...p\}$ of all categorical features, categorical attack budget $\epsilon$
$S \leftarrow \emptyset$
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
```

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*, 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 Appendix.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.
7. Input Gradient Regularization (*IGR*) (Ross & Doshi-Velez, 2018b) 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. 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 Appendix.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."

When tuning the hyper-parameters α and β of the proposed *IGSG* method in Eq.5, we analyze their sensitivity by testing different parameter values ranging from 0.01 to 100. We employ 10-fold cross-validation and evaluate the robustness using the OMPGS attack. In detail, we randomly and evenly divide the training set into 10 parts. Each time, we use one part as test set and others as training set. We train an MLP classifier with varied α and β. We do the whole process for 10 times and each part is regarded as the test set for once. After that, we calculate the average of the adversarial accuracy under OMPGS attack for each setting of the hyper parameters. Figure 5 illustrates the adversarial robustness of the MLP model for *Splice* and *PEDec* datasets. For *Splice*, we consistently obtain excellent results, as different combinations of α and β have small impact on the adversarial accuracy. However, for *PEDec*, we observe that the left side of the box consistently performed well. When using a small α value, good results are achieved regardless of the choice of β. Hence, when applying the *IGSG* method, it is unnecessary to exhaustively explore all combinations of α and β. Balancing the three parts of the loss function typically leads to satisfactory performance. Confidence intervals are calculated to gauge the reliability of the adversarial accuracy obtained through cross-validation. For *PEDec*, the length of the confidence interval ranges from 0.1 to 0.15 at a 95% confidence level. Conversely, for *Splice*, the interval is approximately 0.1 at a 95% confidence level. In the case of the MLP model, we choose α values of 10, 0.01, and 1 for the three datasets, respectively. As for β, we use values of 100, 0.1, and 3 for the respective datasets. For the Transformer model, α is set to 100 for all three datasets, while β takes values of 1, 0.1, and 1 for the respective datasets.

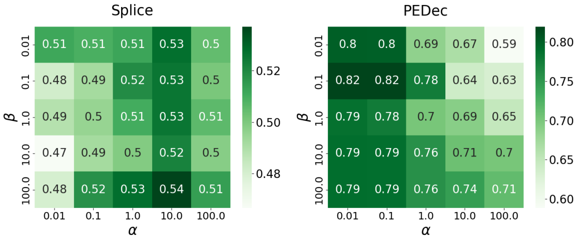


Figure 5: Adversarial Accuracy of *IGSG* under different α and β of the MLP model

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.5. 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 the all the 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 Figure.2 and Figure.8.

H.4 Special settings for mixed-type dataset

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

Algorithm 3 FSGS + PGD for mixed-type data

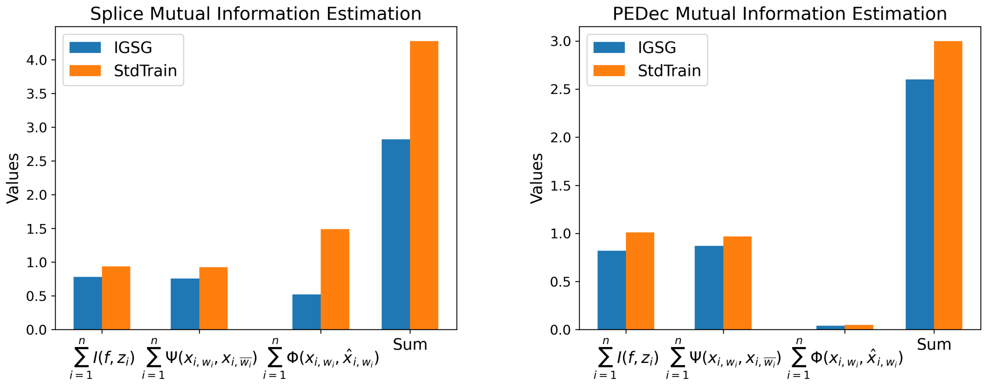
```
Input: The candidate set $H = \{1, 2, ...p_{cat}\}$ of all categorical features, PGD attack budget $\epsilon_n$ for
    numerical data, categorical attack budget $\epsilon_c$
$S \leftarrow \emptyset$
for $iter = 0, 1, 2, \dots$ do
    for each $j \in H/S$ do
        for each $s \subset S$, if $|s| < \epsilon_c$ 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
        $m_f(x(j) + \delta(j, S)) = \underset{s \subset S, |s| < \epsilon_c}{max}\ m_f(\hat{x}(j, s))$
    end for
    $m_f(x + \delta, S) = \underset{j \in H/S}{max}\ m_f(x(j) + \delta(j, S))$
    $j^* = \underset{j \in H/S}{\arg\max}\ m_f(x(j) + \delta(j, S))$
    $S \leftarrow S \cup \{j^*\}$
    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 $\epsilon_n$ for
    numerical data, categorical attack budget $\epsilon_c$
$S \leftarrow \emptyset$
for $iter = 0, 1, 2, \dots$ 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
        $s_j = \underset{s_j \subset S, |s_j| < \epsilon_c}{\arg\max}\ |r_s[j]|,\ \hat{x}_j = B(x, \{j\} \cup s_j)$
    end for
    $j^* \leftarrow \underset{j \in H/S}{\arg\max}\ m_f(\hat{x}(j) + \text{PGD}_\infty(\hat{x}(j), \epsilon_n))$
    $S \leftarrow S \cup \{j^*\}$
    if $Time \geq \Gamma$ then timeout
end for
```

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.



(a) Splice

(b) PEDec

Figure 6: Mutual Information Estimation for terms in Eq.3 for *Splice* and *PEDec* Datasets

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.3), 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.3 .

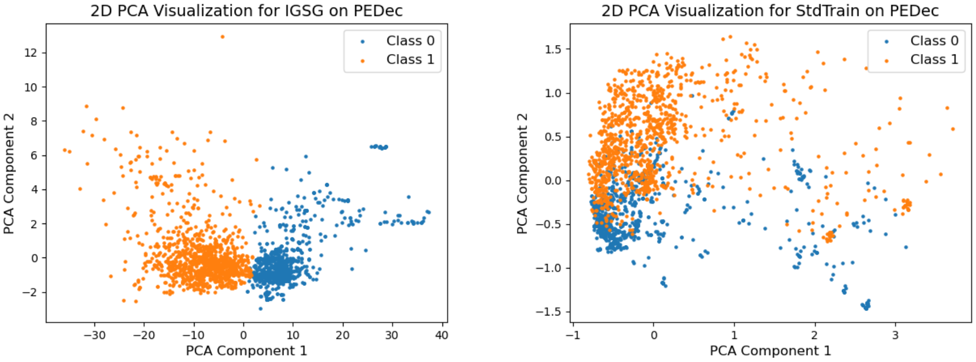
For training, we randomly selected 200 and 500 samples, 20 times each, from the training sets of *Splice* and *PEDec* datasets, respectively. These samples undergo training using Std Train and IGSG approaches, with a learning rate of 0.001, over 200 and 1000 epochs, respectively. This process yields an approximate accuracy of 0.9 for both datasets. Subsequently, we 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.3, 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 Figure 6, 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 reuslt is shown in Table 6.

This experiment aims to demonstrate two key aspects. Firstly, IGSG-trained networks exhibit a reduction in the mutual information terms in Eq.3, 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 Figure 6 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.3. We can refer to Table 6 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.3 and that this reduction positively correlates with improved adversarial accuracy.

Table 6: Average Adversarial Accuracy on 20 logistic regression models for *PEDec* and *Splice* datasets.

Dataset	Attack	IGSG	Std Train
Splice	FSGS	0.019	0.010
	OMPGS	0.139	0.122
PEDec	FSGS	0.709	0.648
	OMPGS	0.748	0.668



(a) IGSG

(b) Std Train

Figure 7: 2D PCA Boundary Visualization on *PEDec* Dataset

Focusing on the first three terms in Figure 6, 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 Figure 3. 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.3 (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.3. Furthermore, we consider adversarial accuracy as a measure of actual 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 Figure.7, 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.

Table 7: Adversarial Accuracy under FSGS attack and Accuracy (%) for *IGSG* 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, 2018b), JR (Hoffman et al., 2019)

Dataset	Attack	Undefended	Adversarial Training baselines					Regularization baselines		**Ours**
		Std Train	Adv Train	Fast-BAT	TRADES	AFD	PAdvT	IGR	JR	**IGSG**
Splice	budget=5	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**
	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	budget=5	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**
	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	budget=5	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**
	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

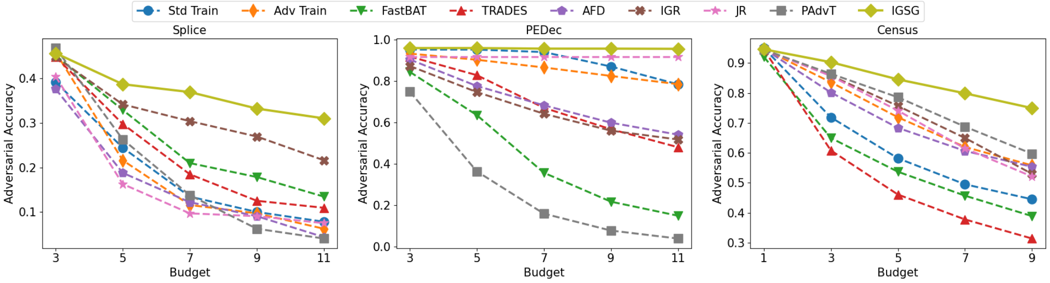


Figure 8: Adversarial accuracy for *IGSG* and baselines under OMPGS attack with varied attack budgets for the Transformer model.

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

I.3 Experimental results on Transformer models

In addition to implementing *IGSG* on the MLP model to demonstrate its effectiveness, we also conducted experiments on a Transformer model. Table 7 presents the accuracy and adversarial accuracy against FSGS attack for each robust training method used with the Transformer model. For the *Splice* dataset, we observed that none of the methods provided effective defense for the Transformer model. This could be attributed to the presence of particularly sensitive features in the *Splice* dataset, as mentioned in Section.5.2. The Transformer model amplifies this effect by focusing more attention on these features, resulting in lower adversarial accuracy. However, *IGSG* achieved comparatively higher adversarial accuracy. Regarding the *PEDec* dataset, *IGSG* demonstrated slight improvement compared to other methods, and the differences in adversarial robustness among the different robust training methods were 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 did not exhibit substantial improvement over the baseline model. However, *IGSG* showed a significant improvement of 10.2% compared to the undefended model.

In Figure 8, we present the adversarial accuracy of all the methods when subjected to OMPGS attacks with varying budgets for the Transformer model. As discussed in Section.5.2, 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* outperforms the baseline models in terms of adversarial accuracy under OMPGS attacks. Specifically, for the *Splice* dataset, *IGSG* 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* 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

Table 8: Adversarial Accuracy and Accuracy over clean samples (%) for *IGSG* and other methods alleviating robust overfitting.

Dataset	Adversary	Adv Train	KD+SWA	MLCAT$_{\text{LS}}$	MLCAT$_{\text{WP}}$	IGSG
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

to *IGSG*. 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* 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* 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 implementation, denoted as MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ 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.8. 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 categorical domain.

I.5 Time complexity analysis

In this section, we give the time complexity of *IGSG* and compare the training time of *IGSG* with other baseline methods. Suppose T in Eq.4 is the number of steps in the Riemman approximation of the integral in Integrated Gradient, R in Eq.18 is the number of randomly sampled neighbors for each data point and N is the number of samples in the training set. The time complexity for each iteration is thus $O(N * (T + R + 1))$. In comparison, OMPGS-based adversarial training has a complexity of $O(N * (2^{\kappa} + p * \kappa))$ for each iteration, where κ represents the number of iterations within each attack and p is the number of features.

Table 9: Time cost (min) for the training process for *IGSG* 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
IGSG	39	117	82	124	71	89

We also measure the runtime cost of *IGSG* with the other baselines in Table.9, based on our implementation using the Python library PyTorch and conducting all the experiments on Linux server with a single GPU (NVIDIA V100). On *Splice*, *IGSG* requires significantly less training time compared to some adversarial training methods like *Adv Train*, *AFD* and *TRADES*. On *PEDec*, *IGSG* requires similar run-time, compared

Table 10: Additional Ablation Study. Adversarial Accuracy and Accuracy over clean testing samples (%) for *IGSG* variants for the MLP model.

Dataset	Adversary	IGSG-VSG	SGSG	IGIG	L_2-IGSG	**IGSG**
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

to *Adv Train*, *AFD* and *TRADES*. On *Census*, *Fast-BAT*, *JR* and *IGR* need less time than *IGSG*, but there is a large gap between the time cost of *IGSG* and that of those methods.

I.6 Detailed ablation study

Here, we introduce another three variants of *IGSG*.

SGSG: We replace the TV loss of the IG scores with the TV loss defined over the smoothed gradient given in Eq.19.
IGIG: 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* and *IGIG* to verify the validity of the two robustness-enhancing regularization terms.
IGSG-VSG: We replace the difference of gradient computing given in Eq.5 with the standard smoothed gradient (Smilkov et al., 2017). We introduce *IGSG-VG* and *IGSG-VSG* to demonstrate the necessity of introducing the mean field smoothing-driven smoothed gradient (given by Eq.19) into the gradient smoothing-based regularization term.

L_2***-IGSG***: 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*.

In Table 10, we provide the adversarial accuracy of the four variants—*IGSG-VSG*, *IGIG*, *SGSG* and L_2-*IGSG* —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*.

SGSG replaces the total variation (TV) loss of *IGSG* with the TV loss of the smoothed gradient. It exhibits slightly inferior performance compared to *IGSG* 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 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*. 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-IGSG replaces the TV loss in the regularization of integrated gradient with an L_2 norm. Compared to *SG*, L_2-IGSG 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 11, 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 7 and Figure 8, *SG* achieves slightly better adversarial robustness due to the smoothing. The only exception is the adversarial accuracy under OMPGS

Table 11: Ablation Study. Adversarial Accuracy and Accuracy over clean testing samples (%) for *IGSG* variants for the Transformer model.

Dataset	Adversary	SG	IG	IGSG-VG	IGSG-VSG	SGSG	IGIG	**IGSG**
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

Table 12: Adversarial accuracy of IGSG and baseline models on MLP and Transformer model structures under PCAA attack for the three datasets.

Model	Dataset	Undefended	Adversarial Training baselines					Regularization baselines		Ours
		Std Train	Adv Train	Fast-BAT	TRADES	AFD	PAdvT	IGR	JR	IGSG
MLP	Splice	37.2±4.0	42.6±1.9	28.7±7.4	27.3±2.4	25.8±2.4	23.2±4.0	42.5±6.0	3.5±4.0	**44.9±2.0**
	PEDec	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
	Census	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
Transformer	Splice	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
	PEDec	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
	Census	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

attack for *PEDec*, where *SG* achieves much better robustness. This may be a result of the smoothness of gradients among neighboring samples. Notably, most variants of *IGSG* achieve very high adversarial accuracy under OMPGS attack, suggesting that both *IG* and *SG* training can defend against OMPGS attack on *PEDec*. Regarding *IG*, *IGSG-VG*, and *IGSG-VSG*, their performance varies across datasets, indicating instability. On the other hand, *SGSG* and *IGIG* 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 Robustness Evaluation under Probabilistic Categorical Adversarial Attack (PCAA)

In this section, we assess the robustness of our proposed IGSG and baseline methods against the PCAA attack (Xu et al., 2023) on three datasets using MLP and Transformer models. The evaluation maintains a consistent setting from previous experiments, with a budget limit of 5 for each dataset.

Table 12 presents the outcomes of this evaluation. It is evident that PCAA does not ensure uniform effectiveness across different datasets. When compared with the results in Table 2 and Table 7, PCAA demonstrates comparable effectiveness to FSGS in attacking the *Splice* dataset with the MLP model and slightly lesser efficacy with the Transformer model. However, its performance on the *PEDec* and *Census* datasets is markedly weaker. The adversarial accuracy for undefended models remains above 87% for both model architectures on these datasets. This could be due to *PEDec*'s high-dimensional feature space and the diverse and extensive categorical dimensions in *Census*, suggesting that PCAA is not an effective measure for assessing robustness in these contexts.

In terms of adversarial accuracy under PCAA attack, IGSG excels on the *Splice* dataset with the MLP model and attains second-best performance with the Transformer model, closely trailing IGR. Although IGSG does not show high adversarial accuracy on the *PEDec* and *Census* datasets compared to the baselines under PCAA attack, this attack strategy is not a reliable measure for these datasets due to its limited effectiveness. Nonetheless, the results from the *Splice* dataset indicate that IGSG notably enhances model robustness.