Adversarial Robustness with Non-uniform Perturbations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Robustness of machine learning models is critical for security related applications, where real-world adversaries are uniquely focused on evading neural network based detectors. Prior work mainly focus on crafting adversarial examples (AEs) with small uniform norm-bounded perturbations across features to maintain the requirement of imperceptibility. However, uniform perturbations do not result in realistic AEs in domains such as malware, finance, and social networks. For these types of applications, features typically have some semantically meaningful dependencies. The key idea of our proposed approach is to enable non-uniform perturbations that can adequately represent these feature dependencies during adversarial training. We propose using characteristics of the empirical data distribution, both on correlations between the features and the importance of the features themselves. Using experimental datasets for malware classification, credit risk prediction, and spam detection, we show that our approach is more robust to real-world attacks. Finally, we present robustness certification utilizing non-uniform perturbation bounds, and show that non-uniform bounds achieve better certification.

1 Introduction

Deep neural networks (DNNs) are commonly used in a wide-variety of security-critical applications such as self-driving cars, spam detection, malware detection and medical diagnosis [1]. However, DNNs have been shown to be vulnerable to adversarial examples (AEs), which are perturbed inputs designed to fool the machine learning systems [2–4]. To mitigate this problem, a line of research has focused on adversarial robustness of DNNs as well as the certification of these methods [1, 5–10].

Adversarial training (AT) is one of the most effective empirical defenses against adversarial attacks [1, 11]. The goal during training is to minimize the loss of the DNN when perturbed samples are used. This way, the model becomes robust to real-world adversarial attacks. Though these empirical defenses do not provide theoretically provable guarantees, they have been shown to be robust against the strongest known attacks [12]. Some of the most common state-of-the-art adversarial attacks, such as projected gradient descent (PGD) [1] and fast gradient sign method (FGSM) [12], perturb training samples under a norm-ball constraint to maximize the loss of the network. The goal of certification, on the other hand, is to report whether an AE exists within an ℓ_p norm centered at a given sample with a fixed radius. Certified defense approaches introduce theoretical robustness guarantees against norm-bounded perturbations [8, 9, 13, 14].

In the computer vision domain, the adversary's goal is to generate perturbed images that cause misclassifications in a DNN. It is often assumed that limiting a uniform norm-ball constraint results in perturbations that are imperceptible to the human eye. However in other applications such as fraud detection [15], spam detection [16], credit card default prediction [17, 18] and malware detection [19–21], norm-bounded uniform perturbations may result in unrealistic transformations. Perturbed

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

samples must comply with certain constraints related to the domain, hence preventing us from borrowing these assumptions from computer vision. These constraints can be on semantically meaningful feature dependencies, expert knowledge of possible attacks, and immutable features [20, 22]. This paper proposes a methodology to generate non-uniform perturbations that takes into account the characteristics of the empirical data distribution. Our results demonstrate that these non-uniform perturbations outperform uniform norm-ball constraints in these types of applications.

1.1 Background and Motivation

AT is a min-max problem minimizing the DNN loss which is maximized by adversarial perturbations (call δ). State-of-the-art approaches for optimizing δ usually assume that all the input features require equal levels of robustness, however, this might not be the case for many applications. A toy example for AT with non-uniform perturbations is given in Appendix A.1. The intuition behind the need for non-uniform constraints is apparent across many industrial applications. A common cybersecurity application is malware detection, which identifies if an executable file is benign or malicious. Unlike images, diverse and semantically meaningful features are extracted from the executable file and are passed to a machine learning model. To maintain the functionality of an executable file during an adversarial attack, certain features may be immutable and perturbations may result in an unrealistic scenario. For example in the Android malware space, application permissions, such as permission to access a phone's location service, are required for malicious functionality and cannot be perturbed [19]. In a finance scenario where customer credit card applications are evaluated by machine learning models, a possible set of features include age, gender, income, savings, education level, number of dependents, etc. In this type of dataset there are clear dependencies between features, for example the number of dependents has a meaningful correlation with age. When detecting spammers within social networks, features are extracted from accounts and may include the length of the username, length of user description, number of following and followers as well as the ratio between them, percentage of bidirectional friends, etc. Similar to the previous finance example, there is a meaningful correlation between features such as the percentage of bidirectional friends and the ratio of followers.

In all of these scenarios, non-uniform perturbations can be used to maintain these correlations and semantically meaningful dependencies resulting in more realistic AEs. In this work, we propose adversarial training with these more realistic perturbations to increase the robustness against real-world adversarial attacks. Specifically, our contributions are: (i) Instead of considering an allowed perturbation region where all the features are treated uniformly, i.e., $\|\delta\|_p \leq \epsilon$, we consider a transformed input perturbation constraint, i.e., $\|\Omega\delta\|_p \leq \epsilon$ where Ω is a transformation matrix, which takes the available information into account, such as feature importance, feature correlations and/or domain knowledge. Hence, the transformation in the norm ball constraint results in non-uniform input perturbations over the features. (ii) For various applications such as malware detection, credit risk prediction and spam detection, we show that robustness using non-uniform perturbations outperforms the commonly-used uniform approach. (iii) To provide provable guarantees for non-uniform robustness, we modify two known certification methods, linear programming and randomized smoothing, to account for non-uniform perturbation constraints.

1.2 Related Work

Different levels of robustness of different features have already been studied in the literature [23–25]. Among these, [23] and [24] study the effect of robust and non-robust features in standard and adversarial training. Both works show that different types of features might either be vulnerable or robust to small input perturbations. This is a discrete interpretation of different levels of perturbation tolerance which is the core idea of our approach. Both works provide effective theoretical analysis of robust and non-robust features and their effect on clean and adversarial accuracy, while we propose a defense mechanism utilizing this phenomenon.

A closely related work to ours is [25] which considers non-uniform perturbation bounds for input features. Given a non-uniform adversarial budget ϵ for ℓ_∞ norm bounded inputs, [25] proposes a framework that maximizes the volumes of certified bounds. However, this work only proposes robustness certification of pre-trained models, and does not consider training a robust model against these non-uniform perturbations. The approach is also data-agnostic meaning that it does not take correlations or an additional knowledge on the data into account. We instead use non-uniform

perturbations in data-dependent AT to achieve robustness in a DNN model. In fact, [25] mentioned consideration of feature correlations as a potential future direction of their work.

In [26], a conditional variational autoencoder (CVAE) is trained to learn perturbation sets for image data and the generated adversarial images are then used in augmented training. The work proposes robustness against common image corruptions as well as ℓ_p perturbations. However, it mainly focuses on possible corruptions and attacks specific to image domain, and assumes access to the test data distribution. In our work, we focus on non-image data which have correlated and different scale features and we do not rely on an additional knowledge of the test set.

Recent work in the malware detection space [20] creates realistic AEs by defining a set of comprehensive and realistic constraints on how an input file can be transformed. Though this approach creates realistic AEs, collecting a representative corpus of raw input files is a challenging problem [27]. The recent release of binary feature sets, such as the EMBER dataset [28], enables model development without having access to the raw input files. Our work can be used to create more realistic AEs in situations where access to a large representative corpus of files is not possible.

2 Non-uniform Adversarial Perturbations

In adversarial training, the worst case loss for an allowed perturbation region is minimized over parameters of a function representing a DNN. The objective of the adversary can be written as the inner maximization of adversarial training:

$$\underset{\delta \in \Delta_{\epsilon,p}}{\text{maximize}} \quad \ell(f_\theta(x+\delta), y), \tag{1}$$

where $\Delta_{\epsilon,p} = \{\delta : \|\delta\|_p \leq \epsilon\}$ is an ℓ_p ball of radius ϵ which defines the feasible perturbation region. Standard PGD follows steepest descent which iteratively updates δ in the gradient direction to increase the loss:

$$\delta^{t+1} = \delta^t + \alpha \frac{\nabla_\delta \ell(f_\theta(x+\delta^t), y)}{\|\nabla_\delta \ell(f_\theta(x+\delta^t), y)\|_p} \tag{2}$$

at iteration t, and then it projects δ to the closest point onto the ℓ_p ball:

$$\mathcal{P}_{\Delta_{\epsilon,p}}(\delta) := \underset{\delta' \in \Delta_{\epsilon,p}}{\arg\min} \|\delta - \delta'\|_2^2 = \epsilon \frac{\delta}{max\{\epsilon, \|\delta\|_p\}} \tag{3}$$

where the distance between δ and δ' is the Euclidean distance, and the projection corresponds to normalizing δ to have a maximum ℓ_p norm which is equal to ϵ. Now, we introduce an adversarial constraint set that non-uniformly limits adversarial variations in different, potentially correlated dimensions, by

$$\tilde{\Delta}_{\epsilon,p} = \{\delta : \|\Omega\delta\|_p \leq \epsilon\} \tag{4}$$

where $\Omega \in \mathbb{R}^{d \times d}$. In our approach, δ is updated by equation (2) similar to the standard PGD, however, it is projected back to a non-uniform norm ball satisfying $\|\Omega\delta\|_p \leq \epsilon$. The corresponding projection operator will then be:

$$\mathcal{P}_{\tilde{\Delta}_{\epsilon,p}}(\Omega\delta) = \begin{cases} \epsilon \frac{\delta}{\|\Omega\delta\|_p} & if \quad \|\Omega\delta\|_p > \epsilon \\ \delta & \text{otherwise.} \end{cases} \tag{5}$$

The choice of Ω depends on how we model the expert knowledge or feature relationships. The following are our choices for the non-uniform perturbation sets.

2.1 Mahalanobis Distance (MD)

Euclidean distance between two points in a multi-dimensional space is a useful metric when the vectors have isotropic distribution (i.e. radially symmetric). This is because the Euclidean distance assumes each dimension has same scale (or spread) and are uncorrelated to other dimensions. However, isotropy is usually not the case for real datasets in which different features might have different scales and can be correlated. Fortunately, MD accounts for how the features are scaled and correlated to one another [29]. Hence, it is a more useful metric if the data has non-isotropic distribution.

By formal definition, MD between vectors $z, z' \in \mathbb{R}^d$ is denoted by $d_M(z, z'|M) := \sqrt{(z-z')^T M (z-z')}$, where $M \in \mathbb{R}^{d\times d}$ is a positive semi-definite matrix which can be decomposed as $M = U^T U$, for $U \in \mathbb{R}^{d\times d}$. The dissimilarity between two vectors from a distribution with covariance Σ can be measured by selecting $M = \Sigma^{-1}$. If feature vectors of a dataset are uncorrelated and have unit variances, their covariance matrix is $\Sigma = I$, which reduces their MD to Euclidean distance.

We are interested in the distance between the original and the perturbed sample. Since we assume all perturbations are additive, as common practice, the distance term we consider is $\sqrt{\delta^T M \delta}$. For a generalized MD in ℓ_p norm, selecting $\Omega = U^T$ corresponds to the perturbation set $\tilde{\Delta}_{\epsilon,p} = \{\delta : \|U^T\delta\|_p \leq \epsilon\}$ which generates AEs with feature correlations similar to the original dataset.

Robustness of an adversarially trained model is directly related to how realistic the generated AEs are during training. Now, we explore implications of selecting ℓ_2 MD to define the limits of the perturbation set. To ensure the validity of the AEs, we consider the notion of consistency of the generated sample with real samples. [18] introduced the notion of ϵ-inconsistency to quantify how likely an AE is. With slight change in their notation, we define γ-consistency as follows:

Definition 2.1. *For a consistency threshold $\gamma \in [0, 1]$, an AE is γ-consistent if $\mathbb{P}(X = x \mid y) \geq \gamma$, where $\mathbb{P}$ is a conditional Gaussian distribution with zero mean and covariance matrix Σ_y.*

Theorem 2.1. *γ-consistency of the AEs generated under MD constraint has a direct relation to ϵ such that*

$$\sqrt{2C - 2\log\gamma} \leq \epsilon. \tag{6}$$

where $C = -\log(2\pi)^{d/2}|\Sigma_y|^{1/2}$, d is the dimension of x, and $\sqrt{\delta^T \Sigma_y^{-1} \delta} \leq \epsilon$.

Theorem 2.1 implies that there is a direct relationship between limiting the MD of δ and ensuring consistent samples when the data is Gaussian. In other words, when the ℓ_2 MD of the perturbations gets smaller, AEs become more consistent. See Appendix A.4 for the proof.

2.2 Weighted Norm

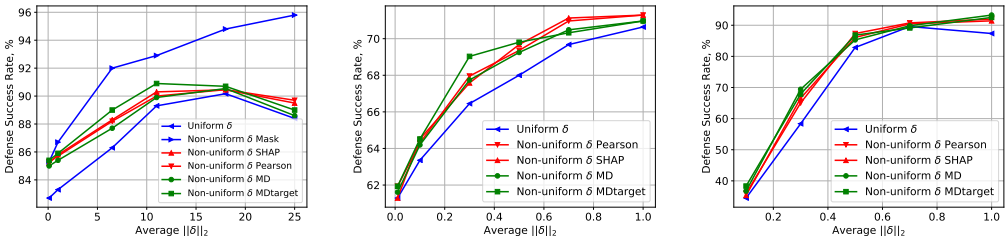
When Ω is a diagonal matrix, inner maximization constraint simply becomes the weighted norm of δ limited by ϵ, and the weights are denoted by $\{\Omega_{i,i}\}_{i=1}^d$. Projection of δ under the new constraint corresponds to projection onto an ℓ_p norm ball of radius $\frac{\epsilon}{\Omega_{i,i}}$ for i^{th} feature. These weights can be chosen exploiting domain, attack or model knowledge. For instance, more important features can be allowed to be perturbed more than the other features which have less effect on the output score of the classifier. This knowledge might come from Pearson's correlation coefficients [17] between the features of the training data and the corresponding labels, or Shapley values [30] for each feature.

Using Pearson's correlation coefficient of each feature with the corresponding target variable, i.e., $|\rho_{i,y}|$ for i^{th} feature and output y, we let larger perturbation radii for more correlated features with the output. Due to the inverse relation between $\Omega_{i,i}$ and the radius of the norm ball, for $\bar{\rho}_{i,y} = \frac{1}{|\rho_{i,y}|}$ we select $\Omega = \frac{diag(\{\bar{\rho}_{i,y}\}_{i=1}^d)}{\|\{\bar{\rho}_{i,y}\}_{i=1}^d)\|_2}$. Similarly, using Shapley values to represent feature importance, we define $\bar{s}_i = \frac{1}{|s_i|}$, where s_i is the Shapley value of feature i. Then, we choose $\Omega = \frac{diag(\{\bar{s}_i\}_{i=1}^d)}{\|\{\bar{s}_i\}_{i=1}^d)\|_2}$ by following the intuition that more important features should have larger perturbation radii.

In the malware domain, expert knowledge might help to rule out specific type of attacks crafted on immutable features due to feasibility constraints. This can be modelled by the proposed weighted norm constraint as masking the perturbations on immutable features. Hence, non-uniform perturbation approach enables various transformations on the attack space for robustness against realistic attacks.

3 Experimental Results

Here, we present experimental results to evaluate robustness of DNNs against adversarial attacks for binary classification problems on three applications: malware detection, credit risk prediction, and spam detection. We compare PGD with non-uniform perturbations during AT with PGD proposed in [1] based on uniform perturbations. For all applications, we evaluate our defense mechanisms on



(a) Malware Use-case (b) Credit Risk Use-case (c) Spam Detection Use-case

Figure 1: Defense success rate of ℓ_2-PGD AT against the problem-space attacks, where all non-uniform perturbation defense approaches outperform the uniform approach for all use-cases.

adversarial attacks proposed by other works. We use a machine with an Intel Xeon E5-2686 v4 @ 2.3 GHz CPU, and 4 Nvidia Tesla V100 GPUs. Details of the DNN architecture and pre-processing are given in A.3. Note that our goal is not to design the best possible neural network but instead compare the uniform perurbations [1] with various non-uniform perturbations during AT for a given DNN.

Adversarial training (AT): We perform AT in all use-cases by applying ℓ_2-norm PGD for uniform perturbation sets, i.e., $\Delta_{\epsilon_1,2} = \{\delta : \|\delta\|_2 \leq \epsilon_1\}$, and non-uniform perturbation sets, i.e., $\tilde{\Delta}_{\epsilon_2,2} = \{\delta : \|\Omega\delta\|_2 \leq \epsilon_2\}$. Since potential adversaries are not interested in fooling the classifiers with negative class (target class) samples, δ perturbations are only applied to the positive classes during AT as commonly used especially in malware detection [20]. Positive classes are the malicious class in malware detection, bad class in credit risk prediction, and spammer class in spam detection. Moreover, for the sake of clean accuracy within the positive class, adversarial perturbations are applied to 90% of the positive samples during training. Such hybrid approach where a weighted clean adversarial loss are optimized at once is common in literature [31].

To model the expert knowledge with diagonal Ω, we use Pearson's correlation coefficient, Shapley values, and masking to allow perturbation only in mutable features. To compute Shapley values, we use SHAP [32] which utilizes a deep learning explainer. We also consider AT under the MD constraint, and select Ω=U^T such that U^TU=Σ_y^{-1} considering two cases; Σ_y is the covariance matrix of the entire training data, i.e., y=$\{0,1\}$, and Σ_y is only for the negative (target) class y=0. We call the models after AT with non-uniform perturbations according to their Ω selection, e.g., *NU-δ-Pearson* for Pearson's coefficients, *NU-δ-SHAP* for Shapley values, *NU-δ-Mask* for masking, *NU-δ-MD* for MD using full covariance matrix and *NU-δ-MDtarget* for MD using the covariance for only y=0. The choice Ω=I corresponds to AT with uniform perturbation constraint, which we call *Uniform-δ*.

3.1 Malware Use-case

First, we consider a binary classification problem for malware detection using the EMBER dataset [28]. EMBER is a feature-based public dataset which is considered a benchmark for Windows malware detection. It contains 2381 features extracted from Windows PE files: $600K$ labeled training samples and $200K$ test samples. We refer to Appendix A.2 for more detailed description of the EMBER dataset features. Given a malware sample, an adversary's goal is to make the DNN conclude that a malicious sample is benign.

Attacks used for evaluation: In the malware domain, test-time evasion attacks can be classified as *feature-space* and *problem-space* attacks. While the former crafts AEs by modifying the features extracted from binary files, the latter directly modifies malware binaries making sure of the validity and inconspicuousness of the modified object. We evaluate the robustness of our model against evasion attacks which are crafted in problem-space, i.e., on PE files. We incorporate the most successful attacks [33] from the machine learning static evasion competition of [34]. Since the EMBER dataset only contains the extracted features of a file, a subset of malware binaries used for AE generation are obtained from VirusTotal [35] using the SHA-256 hash as identifier.

We observe that these problem-space attacks, which add various bytes to a file without modifying the core functionality, affect only the feature groups "Byte Histogram", "Byte Entropy Histogram" and

"Section Information". Experts aware of these byte padding attacks understand which features can be manipulated by an attacker. In addition to the previous AT methods, we represent this *best case expert knowledge* by $\Omega = I_{mask}$, which is an identity matrix with non-zero diagonal elements only for "Byte Histogram", "Byte Entropy Histogram" and "Section Information" features. That is, the model is trained using PGD perturbations applied only to these features, and we call it *NU-δ-Mask*.

Numeric results: To make a fair comparison between uniform and non-uniform approaches, ϵ for each method is selected such that their average distortion budgets, i.e., $\|\delta\|_2$, are approximately equal. We test the detection success of adversarially trained models with 9000 AE sets generated by the problem-space attacks described in Appendix A.6. Figure 1a illustrates the average defense success rate against various problem-space attacks and shows that non-uniform perturbation approaches outperform the uniform perturbation in all cases. Moreover, *NU-δ-MDtarget* performs closest to the best case expert knowledge *NU-δ-Mask* for all cases except when $\|\delta\|_2 = 25$. The advantage of selecting Σ from benign samples versus selecting from the entire dataset is that the direction of perturbations are led towards the target class, i.e. benign samples, for *NU-δ-MDtarget*. We also do not observe a significant performance difference between *NU-δ-Pearson* and *NU-δ-SHAP*, while *NU-δ-MD* only differs from the two for $\|\delta\|_2 = 25$. We refer to Table A.1 for detailed attack performances and to Table A.2 for defense S.R. results with clean accuracy.

3.2 Credit Risk Use-case

Our second use-case is a credit risk detection problem where the DNN's goal is to make decisions on loan applications for bank customers. For this scenario, we use the well-known German Credit dataset [36], which contains classes "good" and "bad", as well as applicant features such as age, employment status, income, savings, etc. It has 20 features and 1000 samples with 300 in the "bad" class. Similar to [17], we treat discrete features as continuous and drop non-ordinal categorical features.

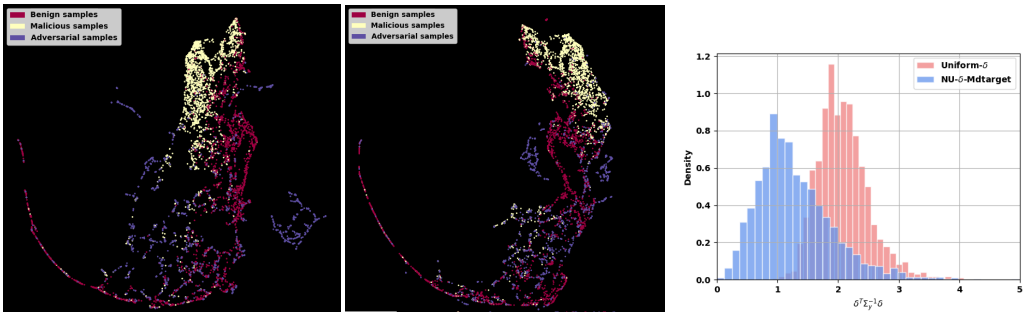
Attacks used for evaluation: The goal of an adversary in this situation is to make DNN models conclude that they are approved for a loan when they actually may not be eligible. Since modifications to tabular data can be detected by an expert eye, attackers try to fool classifiers with imperceptible attacks. We use German Credit dataset implementation of LowProFool [17] which considers attack imperceptibility and represents expert knowledge using feature correlations. We apply the attack on the "bad" class of the test set and generate 155 AEs. After dropping the non-ordinal categorical features, we treat the remaining 12 features as continuous values.

Numeric results: Similar to the malware use-case, ϵ for each method is selected such that their average $\|\delta\|_2$ are approximately equal. In Figure 1b, we report defense success rate of PGD with uniform and non-uniform perturbations in detecting 155 AEs generated by LowProFool. The figure shows that for every given $\|\delta\|_2$, non-uniform perturbations outperform uniform perturbations in PGD. Although LowProFool represents feature importance by Pearson correlation coefficients between features and the output score, surprisingly *NU-δ-Pearson* is the best approach among the other non-uniform approaches for only $\delta = \{0.7, 1\}$. We refer to Table A.3 for clean accuracy results.

3.3 Spam Detection Use-case

Finally, we evaluate robustness within the context of detecting spam within social networks. We use a dataset from Twitter, where data from legitimate users and spammers is harvested from social honeypots over seven months [37]. This dataset contains profile information and posts of both spammers and legitimate users. After pre-processing [38], we extract 31 numeric features with 14 being integers and the rest being continuous. Some examples of these features are the number of following and followers as well as the ratio between them, percentage of bidirectional friends, number of posted messages per day, etc. We treat all features as continuous values in our experiments. Moreover, we extract 41,354 samples where the training set has 17,744 "bad" and 15,339 "good" samples, and the testing set has 3885 "bad" and 4386 "good" samples. The adversary's goal is to make the DNN predict that a tweet was posted by a legitimate user when it was written by a spammer.

Attacks used for evaluation: We incorporate the evasion attack [39] from [16] for our Twitter spam detector. The attack strategy is based on minimizing the maliciousness score of an AE which is measured by a local interpretation model LASSO, while satisfying ℓ_2 norm constraint on perturbations. We generate the AEs by constraining the perturbations to $0.5 \times dist^{avg}_{pos-neg}$, where $dist^{avg}_{pos-neg}$ is defined by [16] as the average distance between the spammer samples and the closest non-spammers



(a) Uniform-δ (b) NU-δ-MDtarget (c) Histogram of MD square, $\delta^T \Sigma_{y=0}^{-1} \delta$

Figure 2: UMAP visualization of benign, malicious and adversarial samples generated by (a) Uniform-δ and (b) NU-δ-MDtarget, and (c) the density histogram of their $\delta^T \Sigma_{y=0}^{-1} \delta = 2C - 2\log\gamma$.

to these samples. We split the Twitter dataset with ratio 25% for training and testing, and generate the AEs using the spammer class of the entire test set.

Numeric results: Again, we apply perturbations only to the spammer set during AT and report the results for approximately equal average $\|\delta\|_2$ perturbations. Figure 1c illustrates defense success rate in detecting AEs of the proposed approaches against the model interpretation based attack [16] for Twitter dataset. The figure shows that non-uniform perturbations outperform uniform case in terms of defense S.R. for all given $\|\delta\|_2$. We refer to Table A.4 for clean accuracy results.

3.4 Quality of Perturbation Sets

In this section, we quantitatively and qualitatively analyze how well non-uniform perturbations capture realistic attacks using γ-consistency property defined in Section 2 and lower dimensional space visualization. Our intuition is that a successful attack evades detection since AEs appear benign to the model. That is, AEs have high likelihood according to the distribution of benign samples. Therefore, we measure a perturbed sample's quality by its γ-consistency with the benign set distribution. Definition 2.1 leverages Theorem 2.1, which shows that smaller MD for δ indicates higher γ-consistency and hence higher quality of the perturbed sample. Moreover, we expect AEs that evade the model and benign samples to be embedded closer to each other in the lower-dimensional subspace. Figure 2 illustrates UMAP visualization [40] of benign, malicious and adversarial samples for the spam detection use-case. AEs generated by NU-δ-MDtarget show better alignment with benign distribution, which shows that NU-δ-MDtarget mimics a more realistic attack. We also show the histogram of MD squares, i.e. $\delta^T \Sigma_{y=0}^{-1} \delta = 2C - 2\log\gamma$, of 1660 AEs from Uniform-δ and NU-δ-MDtarget in Figure 2c, where the average values are 2.1 and 1.28, respectively. Following Theorem 2.1 and Figure 2c, δ's from NU-δ-MDtarget have higher γ, and hence, are more realistic.

4 Certified Robustness with Non-uniform Perturbations

In this section, we present methods for certifying robustness with non-uniform perturbations. We consider two well-known certification methods; a linear programming (LP) approach [9] and randomized smoothing [41].

4.1 LP Formulation

We can provably certify the robustness of deep ReLU networks against non-uniform adversarial perturbations at the input. Our derivation follows an LP formulation of the adversary's problem with ReLU relaxations, then the dual problem of the LP and activation bound calculation. It can be viewed as an extension of [9]. Similar to [9], we consider a k layer feedforward deep ReLU network with

$$\hat{z}_{i+1} = W_i z_i + b_i, \; z_i = \max\{\hat{z}_i, 0\}, \text{ for } i = 1, \cdots, k-1 \tag{7}$$

We denote $\mathcal{Z}_{\epsilon,\Omega}(x) := \{f_\theta(x+\delta) : ||\Omega\delta||_p \leq \epsilon\}$ as the set of all attainable final-layer activations by input perturbation δ. Since this is a non-convex set for multi-layer networks which is hard to

optimize over, we consider a convex outer bound on $\mathcal{Z}_{\epsilon,\Omega}(x)$ and optimize the worst case loss over this bound to guarantee that no AEs within $\mathcal{Z}_{\epsilon,\Omega}(x)$ can evade the network. As done in [9], we relax the ReLU activations by representing $z = \max\{0, \hat{z}\}$ with their upper convex envelopes $z \geq 0, z \geq \hat{z}, -u\hat{z} + (u-l)z \leq -ul$, where l and u are the known lower and upper bounds for the pre-ReLU activations. We denote the new relaxed set of all attainable final-layer activations by $\tilde{\mathcal{Z}}_{\epsilon,\Omega}(x)$. Assuming that an adversary targets a specific class to fool the classifier, we write the LP as

$$\underset{\hat{z}_k}{\text{minimize}} \quad c^T \hat{z}_k \qquad \text{s.t.} \quad \hat{z}_k \in \tilde{\mathcal{Z}}_{\epsilon,\Omega} \tag{8}$$

where $c := e_{y^{true}} - e_{y^{target}}$ is the difference between the selection vector of true and the target class.

A positive valued objective for all classes as a solution to equation 8 indicates that there is no adversarial perturbation within $\tilde{\Delta}_{\epsilon,p}$ which can evade the classifier. To be able to solve equation 8 in a tractable way, we consider its dual whose feasible solution provides a guaranteed lower bound for the LP. It is previously shown by [9] that a feasible set of the dual problem can be formulated similar to a standard backpropagation network and solved efficiently. The dual problem of our LP with ReLU relaxation and non-uniform perturbation constraints is expressed in the following theorem.

Theorem 4.1. *The dual of the linear program 8 can be written as*

$$\begin{aligned}
&\underset{\hat{\nu},\nu}{\textit{maximize}} \quad -\sum_{i=1}^{k-1} \nu_{i+1}^T b_i + \sum_{i=2}^{k-1} \sum_{j \in \mathcal{I}_i} l_{i,j} [\hat{\nu}_{i,j}]_+ - \hat{\nu}_1^T x - \epsilon ||\Omega^{-1} \hat{\nu}_1||_q \\
&\textit{s.t.} \qquad \nu_k = -c, \;\; \hat{\nu}_i = (W_i^T \nu_{i+1}), \;\; \textit{for} \;\; i = k-1, \ldots, 1 \\
&\qquad \nu_{i,j} = \begin{cases} 0 & j \in \mathcal{I}_i^- \\ \hat{\nu}_{i,j} & j \in \mathcal{I}_i^+ \\ \frac{u_{i,j}}{u_{i,j} - l_{i,j}} [\hat{\nu}_{i,j}]_+ - \eta_{i,j} [\hat{\nu}_{i,j}]_- & j \in \mathcal{I}_i \end{cases}, \;\; \textit{for} \;\; i = k-1, \ldots, 2
\end{aligned} \tag{9}$$

where $\mathcal{I}_i^-$, $\mathcal{I}_i^+$ and $\mathcal{I}_i$ represent the activation sets in layer i for l and u are both negative, both positive and span zero, respectively.

See Appendix A.4 for the proof of Theorem 4.1. When $\eta_{i,j} = \frac{u_{i,j}}{u_{i,j} - l_{i,j}}$, Theorem 4.1 shows that the dual problem can be represented as a linear back propagation network, which provides a tractable solution for a lower bound of the primal objective. To solve equation 9, we need to calculate lower and upper bounds for each layer incrementally as explained in Appendix A.5.

For certification of robustness within a non-uniform norm ball around a test sample, we need the objective of the LP to be positive for all classes. Since the solution of the dual problem is a lower bound on the primal LP, it provides a worst case certification guarantee against the adversaries within the non-uniform norm ball. To support the approaches proposed in Section 2, we provide certification results for the robustness of *Uniform-δ* and *NU-δ-MDt* (*NU-δ-MDtarget*) for spam detection use-case in Table 1. We consider both uniform and non-uniform input constraints in certification methods, namely *Uniform-Cert* for the standard LP approach for certification with uniform perturbation constraint [9], and *NU-Cert-(.)* for the non-uniform constraint approach. We implement our non-uniform approach into the LP solution by modifying [9] with our Ω matrix, and generate various certification methods by non-uniform Ω selections, e.g. *NU-Cert-SHAP*, *NU-Cert-Pearson*, *NU-Cert-MD* and *NU-Cert-MDt*. Our purpose is not to propose the tightest certification

Table 1: Average certification margin and number of successful certified samples out of 1000 spammers for *NU-δ-MDt* and *Uniform-Cert* for Spam Detection Use-case.

Model	Defense S.R.	Cert. Method	Margin	Cert. Success
Uniform-δ	$54.87 \pm 1.1\%$	Uniform-Cert	1.07	$34.72 \pm 0.94\%$
		NU-Cert-SHAP	1.84	$72.64 \pm 0.6\%$
		NU-Cert-Pearson	2.04	$76.8 \pm 0.71\%$
		NU-Cert-MD	2.40	$80.2 \pm 0.56\%$
		NU-Cert-MDt	2.40	$80.2 \pm 0.55\%$
NU-δ-MDt	$63.4 \pm 0.74\%$	Uniform-Cert	1.11	$42.95 \pm 0.69\%$
		NU-Cert-SHAP	1.9	$74.65 \pm 0.85\%$
		NU-Cert-Pearson	2.06	$78.38 \pm 0.76\%$
		NU-Cert-MD	2.41	$81.3 \pm 0.68\%$
		NU-Cert-MDt	2.41	$81.3 \pm 0.67\%$

bounds but to show that non-uniform input constraint results in larger certification margin compared to the uniform approach.

We compare *Uniform-δ* and *NU-δ-MDt* to evaluate certification results. Dropout layers are removed from the model for LP solution, and AT is performed for $\epsilon = 0.3$. Certification is done by solving the LP for $\epsilon = 0.3$ over 1000 spammers. The objective should be positive for all classes to certify the corresponding sample. The margin between the objective and zero gives an idea about how tight the bound is [42]. Table 1 demonstrates two main results: (i) the certification success of *NU-δ-MDtarget* over *Uniform-δ* for each certification method supports our claim that non-uniform perturbations provide higher robustness than the uniform approach; and (ii) certification with non-uniform constraints provide larger certification margins and hence tighter bound.

4.2 Randomized Smoothing

Robustness certification via *randomized smoothing* [41] is an empirical alternative approach to the method discussed in Section 4.1. The idea is constructing a "smoothed" classifier g from the base classifier f. In the original formulation introduced in [41], the smoothed classifier g returns the most likely output returned by the base classifier f given input x is perturbed by isotropic Gaussian noise. Here, we provide robustness guarantee in binary case for randomized smoothing framework when non-isotropic Gaussian noise is used to allow robustness to non-uniform perturbations:

$$g(x) = \underset{y \in \mathcal{Y}}{\arg\max}\, \mathbb{P}(f(x+n) = y) \quad \text{where} \quad n \sim \mathcal{N}(0, \Sigma). \tag{10}$$

Adapting notation and Theorem 2 from [41], let p_a be the probability of the most probable class $y = a$ when the base classifier f classifies $\mathcal{N}(x, \Sigma)$. Then the below theorem holds.

Theorem 4.2. *In binary classification problem, suppose $\underline{p_a} \in (\frac{1}{2}, 1]$ satisfies $\mathbb{P}(f(x+n) = a) \geq \underline{p_a}$. Then $g(x+\delta) = a$ for all $\sqrt{\delta^T \Sigma^{-1} \delta} \leq \Phi^{-1}(\underline{p_a})$.*

See Appendix A.4 for the proof. In theorem 4.2, we show that a smoothed classifier g is robust around x within ℓ_2 Mahalanobis distance $\sqrt{\delta^T \Sigma^{-1} \delta} \leq \Phi^{-1}(p_a)$ where Φ^{-1} is inverse of the standard Gaussian CDF. The same result holds if we replace p_a with lower bound $\underline{p_a}$.

We implement our non-uniform approach into randomized smoothing by modifying [43] with our non-isotropic noise space. Table 2 shows certification S.R. of Uniform-δ and NU-δ-MDt, when they are certified by standard randomized smoothing with $\mathcal{N}(0, \sigma I)$ (UC), and our non-uniform methods with $\mathcal{N}(0, \Sigma_y)$ for corresponding Σ_y. That is, $\Sigma_{y=0}$ for NUC-MDt, $\Sigma_{y=\{0,1\}}$ NUC-MD, $\frac{1}{\hat{\rho}^2} I$ for NUC-Pearson and $\frac{1}{\hat{s}^2} I$ for NUC-SHAP are used when the average training distortion budget is $\|\delta\|_2$=5 and the average certification distortion is $\|\delta\|_2$=2.8. Table 2 shows that NU-δ-MDt is certifiably robust for more samples than Uniform-δ for all certification methods. Moreover, certification with non-uniform noise, especially with NUC-MDt, provides higher certification S.R. compared to uniform noise.

Table 2: Percentage of successfully certified samples for *NU-δ-MDt* and *Uniform-δ* with various certification approaches with randomized smoothing for Spam Detection use-case.

Model	UC	NUC-Pearson	NUC-SHAP	NUC-MD	NUC-MDt
Uniform-δ	50.96%	61.11%	64.24%	65.72%	66.45%
NU-δ-MDt	61.8%	67.14%	71.25%	85.34%	90.11%

5 Conclusion

In this work, we study adversarial robustness against evasion attacks, with a focus on applications where input features have to comply with certain domain constraints. We assume Gaussian data distribution in our consistency analysis, as well as precomputed covariance matrix and Shapley values. Under these assumptions, our results on three different applications demonstrate that non-uniform perturbation sets in AT improve adversarial robustness, and non-uniform bounds provide better robustness certification. As an unintended negative social impact, our insights might be used by malicious parties to generate AEs. However, this work provides the necessary defense mechanisms against these potential attacks.

References

[1] 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.

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, page 2, 2018.

[6] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[7] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in Neural Information Processing Systems*, 31: 10802–10813, 2018.

[8] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10877–10887, 2018.

[9] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[10] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pages 4939–4948, 2018.

[11] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

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

[13] Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Mixtrain: Scalable training of formally robust neural networks. *arXiv preprint arXiv:1811.02625*, 14, 2018.

[14] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586, 2018.

[15] Mary Frances Zeager, Aksheetha Sridhar, Nathan Fogal, Stephen Adams, Donald E Brown, and Peter A Beling. Adversarial learning in credit card fraud detection. In *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pages 112–116. IEEE, 2017.

[16] Ninghao Liu, Hongxia Yang, and Xia Hu. Adversarial detection with model interpretation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1803–1811, 2018.

[17] Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274*, 2019.

[18] Eden Levy, Yael Mathov, Ziv Katzir, Asaf Shabtai, and Yuval Elovici. Not all datasets are born equal: On heterogeneous data and adversarial examples. *arXiv preprint arXiv:2010.03180*, 2020.

[19] Deqiang Li and Qianmu Li. Adversarial deep ensemble: Evasion attacks and defenses for malware detection. *IEEE Transactions on Information Forensics and Security*, 15:3886–3900, 2020.

[20] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ml attacks in the problem space. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1332–1349. IEEE, 2020.

[21] Ishai Rosenberg, Shai Meir, Jonathan Berrebi, Ilay Gordon, Guillaume Sicard, and Eli Omid David. Generating end-to-end adversarial examples for malware classifiers using explainability. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2020.

[22] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. Lemna: Explaining deep learning based security applications. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 364–379, New York, NY, USA, 2018. Association for Computing Machinery.

[23] D. Tsipras, Shibani Santurkar, L. Engstrom, A. Turner, and A. Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *ArXiv*, abs/1805.12152, 2018.

[24] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, volume 32, pages 125–136. Curran Associates, Inc., 2019.

[25] Chen Liu, Ryota Tomioka, and Volkan Cevher. On certifying non-uniform bounds against adversarial attacks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4072–4081, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[26] Eric Wong and J. Zico Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv: 2007.08450*, 2020.

[27] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. {TESSERACT}: Eliminating experimental bias in malware classification across space and time. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 729–746, 2019.

[28] Hyrum S. Anderson and Phil Roth. EMBER: an open dataset for training static PE malware machine learning models. *arXiv preprint arXiv: 1804.04637*, 2018.

[29] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences*, 2:49–55, 1936.

[30] Lloyd S. Shapley. *Notes on the n-Person Game -; II: The Value of an n-Person Game*. RAND Corporation, Santa Monica, CA", 1951.

[31] Haotao Wang, Tianlong Chen, Shupeng Gui, Ting-Kuei Hu, Ji Liu, and Zhangyang Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. *arXiv preprint arXiv:2010.11828*, 2020.

[32] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 12 2017.

[33] William Fleshman. *Evading Machine Learning Malware Classifiers*, 2019. URL `https://towardsdatascience.com/evading-machine-learning-malware-classifiers-ce52dabdb713`.

[34] DEFCON. *Machine learning static evasion competition*, 2019. URL `https://www.elastic.co/blog/machine-learning-static-evasion-competition`.

[35] VirusTotal. URL http://www.virustotal.com/.

[36] C. J. Merz and P. Murphy. *UCI repository of machine learning databases*, 1996. URL http://www.cs.uci.edu/~mlearn/MLRepository.html.

[37] Kyumin Lee, Brian David Eoff, and James Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, Barcelona, 2011.

[38] Xiao Huang. *Twiter bot detection*, 2017. URL https://github.com/tapilab/is-xhuang1994.

[39] Ninghao Liu, Hongxia Yang, and Xia Hu. Interpretation to adversary. 2018. URL https://github.com/ninghaohello/Interpretation2Adversary.

[40] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[41] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[42] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. Benchmark for lp-relaxed robustness verification of relu-networks. 2019. URL https://github.com/Hadisalman/robust-verify-benchmark.

[43] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. 2019. URL https://github.com/locuslab/smoothing.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] As we mentioned in abstract and introduction, our contributions are firstly an empirical defense based on adversarial training with non-uniform perturbation constraints, secondly certification with non-uniform input bounds.
   (b) Did you describe the limitations of your work? [Yes] We mentioned in Definition 2.1, Theorem 2.1 and the Section 5 that we assume Gaussian data distribution. See the Section 5 for the assumption that covariance matrix and Shapley values are precomputed.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the Section 5 for the negative social impact. Although we propose a method to generate strong attacks during adversarial training, the paper already suggests a defense against these attacks in Section 2, and certification for its robustness in Section 4.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Definition 2.1 for Theorem 2.1, and Gaussian distribution assumption for both Theorem 2.1 and 4.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.4.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code, data, links to the used datasets and instructions are provided in the supplementary material. Details about the used datasets, references to the implemented attacks and their pre-processing are mentioned in Section 3 and A.6.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3 and the code in the supplementary material.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We provided standard error of the mean for critical results on Tables 1, A.2, A.3 and A.4.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Section 3.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We have used publicly shared code of the attacks in Sections 3.2 and 3.3, and modified the codes of certification methods in Section 4, and cited the creators accordingly.

(b) Did you mention the license of the assets? [Yes] For the existing codes and datasets that we used, we cited the corresponding github pages which contains their license information.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In the supplementary material, we included our code for implementation of the proposed methods, and data in the form of Shapley values and Greedy attack features obtained from EMBER dataset which is publicly available.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We mentioned in Section 3 that we used publicly shared datasets and codes, and we cited the creators accordingly.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] Datasets do not contain personally identifiable information or offensive content.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]