# Group-based Robustness: A General Framework for Customized Robustness in the Real World

Weiran Lin
Carnegie Mellon University
weiranl@andrew.cmu.edu

Keane Lucas
Carnegie Mellon University
kjlucas@andrew.cmu.edu

Neo Eyal
Tel Aviv University
neoedan@gmail.com

Lujo Bauer
Carnegie Mellon University
lbauer@cmu.edu

Michael K. Reiter
Duke University
michael.reiter@duke.edu

Mahmood Sharif
Tel Aviv University
mahmoods@tauex.tau.ac.il

***Abstract*—Machine-learning models are known to be vulnerable to evasion attacks, which perturb model inputs to induce misclassifications. In this work, we identify real-world scenarios where the threat cannot be assessed accurately by existing attacks. Specifically, we find that conventional metrics measuring targeted and untargeted robustness do not appropriately reflect a model's ability to withstand attacks from one *set of source classes* to another *set of target classes*. To address the shortcomings of existing methods, we formally define a new metric, termed *group-based robustness*, that complements existing metrics and is better-suited for evaluating model performance in certain attack scenarios. We show empirically that group-based robustness allows us to distinguish between machine-learning models' vulnerability against specific threat models in situations where traditional robustness metrics do not apply. Moreover, to measure group-based robustness efficiently and accurately, we 1) propose two loss functions and 2) identify three new attack strategies. We show empirically that, with comparable success rates, finding evasive samples using our new loss functions saves computation by a factor as large as the number of targeted classes, and that finding evasive samples, using our new attack strategies, saves time by up to 99% compared to brute-force search methods. Finally, we propose a defense method that increases group-based robustness by up to 3.52 times.**

## I. Introduction

Machine-learning models are known to be vulnerable to evasion attacks—attacks that, by slightly perturbing the models' input, cause models to misclassify [1]. Research that evaluates the susceptibility of models to evasion attacks typically measures these models' classification accuracies with benign and evasive inputs; these metrics are commonly referred to as benign accuracy and untargeted robustness, respectively [2]–[34]. Previous work also defines models' *targeted robustness* as their ability to resist making specific (mis)classifications when faced with evasion attempts [35]–[37].

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.24084
www.ndss-symposium.org


Fig. 1. Traffic signs from GTSRB [38]. The left three columns are speed-limit and delimit signs (i.e., ones that restrict speed limit or mark the end of restrictions). The rightmost column includes three signs that signify an immediate stop: no vehicles, no entry, and stop (from top to bottom).

However, more complicated threats exist in the real world. For example, suppose adversaries want to induce traffic congestion or self-driving vehicle accidents. Such adversaries could attempt to achieve their goal by suddenly reducing the speed of certain vehicles, so that these vehicles might be hit by the vehicles behind them. To do so, they might perturb a *specific* group of traffic signs, such as speed limit and delimit signs, which restrict allowed speeds or remove such restrictions (shown in Fig. 1)). They might perturb these signs to signs that command an immediate stop, such as stop signs, no entry signs, and no vehicle signs; or they could also perturb these signs into signs that display a limit much lower than the actual limit (e.g., no more than half of the actual limit. Adversaries achieve their goal if they perturb the speed limit and delimit signs to be incorrectly classified as *any* sign that requires an immediate stop or specifies a speed limit much lower than the actual limit.

As another example, suppose a different adversary—a group of burglars, for instance—wants to illegally open a vault at a bank. The bank requires three *distinct* staff members to give their permission before the vault can be opened; none of the members can open the vault alone. The group of burglars might therefore be able to succeed in opening the vault if they are able to impersonate any three distinct individuals who work for the bank. However, they might be restricted to only a few attempts before triggering an alarm, and thus they need

attempt-efficient strategies for choosing the staff members that they would impersonate.

We observe that existing metrics—benign accuracy, untargeted robustness, and targeted robustness—do not accurately measure models' ability to resist making misclassifications in these examples and other similar scenarios. Specifically, in the traffic sign example, benign accuracy and untargeted robustness measure models' ability to resist predicting *any* inputs as *any* incorrect classes. Targeted robustness measures models' ability to resist predicting *any* inputs as a *specific* incorrect class. None of the three metrics assess models' ability to resist predicting inputs from one *set of classes* as another, mutually exclusive *set of classes*. Additionally, in the burglary example, targeted and untargeted robustness evaluate models on a per-input-instance basis, while the models' ability to resist giving authorized access cannot be measured on a per-input-instance basis: to open the vault, *multiple (>1)* burglars may impersonate authorized bank staff simultaneously. Hence, there is a need for a new metric to better evaluate the susceptibility of models to such threats. To this end, we formally define this new metric, *group-based robustness*, as a model's ability to resist attempts to cause *specific misclassifications* on data points *from certain classes*. We then empirically demonstrate that this metric gives us insight into models that previous metrics do not: models that appear similar according to existing metrics are actually very different by this new metric, and hence not equally suitable in scenarios where robustness is essential (§II).

While existing attacks can be used to estimate group-based robustness, they are inefficient at doing so. As another contribution, we designed more computationally efficient attacks, termed *group-based attacks*, to help compute group-based robustness faster while attaining a comparable or higher level of accuracy than the following naïve methods:

- One possible naïve method to perform group-based attacks is to attempt each of the specified misclassifications on each input instance. In the traffic sign example, attackers might launch three individual targeted attacks to perturb a 60 KPH speed limit sign. Each attack would try to perturb the sign into one of the three signs that require an immediate stop, a 20 KPH speed limit, or a 30 KPH speed limit. This approach tends to find the most adversarial examples, and we use it to measure group-based robustness. However, compared with running one targeted attack, this approach runs a set of targeted attacks and is more time-consuming.
- Another possible naïve approach is to randomly select a single target class from a specified set and perform standard targeted attacks. For example, attackers may launch a single targeted attack in which the target class is randomly selected from among the signs that require an immediate stop or display a limit much lower than the actual limit. While this approach costs less in time than the previous one, it tends to be significantly less successful.

To more quickly find perturbations that can cause misclassifications among a specified set of candidates, we define two new loss functions, $\ell_{\text{MDMAX}}$ and $\ell_{\text{MDMUL}}$. We empirically verify that the loss functions boost the efficiency of attacks in scenarios like the traffic sign example (§III-A4). Compared with iterating over all target classes to perform targeted attacks, attacks with our loss functions were computationally cheaper by a factor as large as the number of targeted classes while still finding similarly many successful perturbations. Compared with randomly selecting a target class for each input instance to perform targeted attacks, attacks with our loss functions were equally fast but found successful perturbations up to 15× more often (§III-A).

Next, we propose more efficient attack strategies for settings in which an attacker has a small set of inputs at their disposal and is attempting to target a specific subset of classes. In particular, we define three new attack strategies that choose which misclassification to attempt by first estimating the individual chances of success of perturbing each input instance into each target class. In the burglary scenario, this would allow burglars to make fewer impersonation attempts (or to better choose which subset of burglars attempts the break-in)—the strategies estimate each burglar's chance to impersonate each staff member and then attempt to cause impersonations only for the most promising pairs. We demonstrate that our new attack strategies boost the efficiency of attacks; e.g., in the burglary scenario, compared with randomly selecting burglars and staff to launch attacks, burglars would need up to 99% fewer attack attempts with these strategies (§III-B).

Finally, we show how formalizing the real threat allows more effective defenses against it: we demonstrate how to modify adversarial training to increase group-based robustness, without losing benign accuracy or accuracy on classes that might be impersonated (§IV). For example, in the burglary scenario, a face-recognition system with our defense obtains up to 3.52× better robustness, with similar benign accuracy for all identities and similar benign accuracy for all staff members, compared to existing defenses. We modified adversarial training to optimize these three metrics instead of conventional robustness metrics (§IV).

In summary, our contributions are the following:

- We define a new metric that better reflects many practical attack scenarios and more accurately evaluates their corresponding threat (§II).
- We propose two loss functions that help attacks find misclassifications within a given set of targeted classes markedly faster than existing methods (§III-A).
- We develop three attack strategies that when used individually or together, can produce diverse misclassifications for a given number of input instances with better time efficiency than brute-force approaches (§III-B).
- We implement a defense method that improves the robustness of machine learning models against the attacks mentioned above (§IV).

Next, in §II, we further motivate and formally define the new group-based metric. We introduce new loss functions and new attack strategies that take advantage of our metric in §III. We also propose a defense that boosts the performance of models in this metric in §IV. Finally, we position this work in an overview of related work in §V and conclude in §VI.

## II. Group-based Robustness: A New Metric

In this section, we introduce group-based robustness, a new metric to evaluate machine-learning models. We first introduce existing evasion attacks and how robustness is typically measured (§II-A). Then, we present real-world scenarios that demonstrate the importance of this new metric (§II-B). Next, we formally define the new metric, *group-based robustness*, and corresponding attacks, *group-based attacks* (§II-C); and we show that group-based attacks constitute a broader space of evasion attacks than had previously been studied (§II-D). Finally, we discuss our experiments (§II-E) and empirically demonstrate that group-based robustness offers a meaningful assessment of model susceptibility to attacks in the real world that is orthogonal to conventional metrics (§II-F).

### A. Background

Evasion attacks perturb the input of machine-learning models to induce misclassifications. There are many implementations of evasion attacks [37], [39]–[56], along with defenses against these attacks [2]–[34]. The majority of established attacks are *untargeted*, aiming to avoid the correct classification [39]–[54], while some previous works explore *targeted* adversarial attacks, aiming to cause an input to be misclassified as a member of a *single* specific, incorrect class [37], [39], [50], [51]. *Robustness* is defined as a model's ability to resist evasion attacks. One common method to assess untargeted robustness is to measure the model's accuracy on evasive examples [2], [39]. Targeted robustness can be assessed by the model's resistance to predict target classes chosen uniformly at random [37].

### B. Motivation

We suggest that untargeted and targeted robustness, as defined, are not sufficient to accurately assess risk for many real-world attack scenarios: such attack scenarios could be complicated and involve more than one misclassification, as in the traffic sign scenario discussed in §I. In this scenario, attackers might perturb speed limit and delimit signs (signs that restrict the speed limit to specific values or mark the end of previous such restrictions; see Fig. 1) into signs that require an immediate stop, including stop signs, no-entry signs, and no-vehicle signs, or display a limit much lower than the actual limit, such as no more than half of the actual limit.

As another example, suppose students in a class are trying to access materials (e.g., gradebooks) normally accessible only to TAs and professors. The students might succeed by impersonating *any* of the TAs or professors, even if they cannot impersonate a specific TA or professor. However, the students cannot succeed if they only impersonate other students. The untargeted or targeted setting is not sufficient for this scenario.

The burglary example described in §I serves as a more complicated example of an attack scenario. Access to the vault is mediated by facial recognition and is granted only if *several* of the staff are recognized as trying to open the vault together. The burglars thus succeed only if they are able to impersonate *several distinctive members* of the staff. Which burglars (from a larger group) will attempt impersonation, and which staff the impersonations will target (from among those who have access to the vault), is at the burglars' discretion.

Neither targeted nor untargeted robustness intuitively corresponds well to either of these attack scenarios, as they measure the likelihood of successfully inducing *any* misclassification or a misclassification to a *single, specific* class. Neither of those corresponds to the attackers' goals in these scenarios, and, thus, we need a different metric to better assess the risk.

### C. Definition

We propose group-based robustness as a new metric that can assess the risk in the scenarios described in §II-B. In prior work, individual evasion attacks primarily sought to optimize (mis)classification toward a specific class (targeted) or optimize (mis)classification away from a specific class (untargeted). The purpose of group-based robustness is to formalize group-based goals that encompass the results of multiple evasion attacks, where "groups" consist of sets of classes. We define group-based robustness using an experiment, inspired by experiments used to define cryptographic security properties. The experiment is parameterized by the following:

- A *classifier* is a possibly randomized algorithm that takes as input an *instance* $x$ in $\mathcal{X}$ and returns a *class* in $\mathcal{Y}$. That is, $\mathcal{X}$ denotes the set of possible inputs to be classified and $\mathcal{Y}$ denotes the set of classes. The experiment includes two classifiers:
  - The *ground truth classifier* $f$ is a deterministic classifier, i.e., a function.
  - The *targeted classifier* $\tilde{f}$ is a randomized or deterministic classifier. Intuitively, the adversary will seek to mislead this classifier, i.e., to induce classifications by $\tilde{f}$ that differ from those by $f$.
- Let $\mathcal{I} = \{I_i\}_{i=1}^{m}$ be a set of relations on $\mathcal{Y}$, i.e., each $I_i \subseteq \mathcal{Y} \times \mathcal{Y}$. Informally, the adversary's goal is to implement a misclassification defined by any $I_i \in \mathcal{I}$. That is, in order to "win," the adversary must, for some $I_i \in \mathcal{I}$, achieve *every* misclassification $(s, t) \in I_i$, in the sense that the adversary successfully transforms a given instance $x$ of class $s$ (according to $f$, $f(x) = s$) into an instance $x'$ that is classified in $t$ by $\tilde{f}$, while $f(x') = s$. It is worth pointing out that $\tilde{f}(x)$ might be different from $f(x)$, and with specific choices of $\mathcal{I}$, $x'$ might be the same as $x$.
- $\Pi$ is a predicate indicating whether $x'$ is "close enough" to $x$, e.g., according to some distance metric. For the adversary to "win," the instance $x'$ generated by modifying $x$ must also satisfy $\Pi(x, x') = 1$.
- $\mathcal{G}$ is an algorithm generating instances $X \subseteq \mathcal{X}$ by sampling them from some distribution. Conventionally, $|X| = 1$. In §II-D, we will explain that certain choices of $\mathcal{I}$ are only achievable when $|X| > 1$ while these choices correspond to realistic attack scenarios. Informally, $\mathcal{G}$ is the environment that produces instances for which the adversary can try to induce misclassifications.
- The adversary $A$ is an algorithm that takes as input a set $X \subseteq \mathcal{X}$ and produces $R \subseteq X \times \mathcal{X}$. Informally, if $(x, x') \in R$, then the adversary changes $x$ into $x'$

to satisfy the properties above. The adversary also has white-box access to $\Pi$, $f$, $\tilde{f}$, $\mathcal{G}$, and $\mathcal{I}$, which are public parameters of the experiment.

The formal definition of the experiment in which the adversary $A$ participates is as follows:

Experiment $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$
$X \leftarrow \mathcal{G}()$
$R \leftarrow A(X)$
$\tilde{I} \leftarrow \{(f(x), \tilde{f}(x')) : (x, x') \in R\}$
if $\left(\tilde{I} \in \mathcal{I} \wedge \forall (x, x') \in R : (x \in X \wedge \Pi(x, x') = 1)\right)$
return 1
else
return 0

The adversary $A$ is run on the input instances $X$ generated by $\mathcal{G}$. The relation achieved $\tilde{I}$ is computed based on the results of the adversary $A$, the ground truth classifier $f$, and the targeted classifier $\tilde{f}$. The experiment returns 1 (i.e., the adversary succeeds) if $\Pi$ indicates that the perturbation is small enough according to some distance metric *and* $\tilde{I}$ matches some $I_i$ in the desired set $\mathcal{I}$. The experiment returns 0 otherwise.

We define the imp-rel advantage of $A$ to be

$$\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A) = \mathbb{P}\left(\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A) = 1\right)$$

where the probability is taken w.r.t random choices made by $\mathcal{G}$, $\tilde{f}$, and $A$. Analogously, the group-based robustness is

$$\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A) = \mathbb{P}\left(\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A) = 0\right)$$

By requiring properties of each $I_i \in \mathcal{I}$, we can express cases of interest. For example, by requiring each $I_i$ to be a function, we require that input instances $x$ in distinct classes each be used to impersonate only one class.

### D. A Broader Attack Space

The definition of group-based attacks sheds light on a broader space of attacks than had previously been explored with attacks or defenses. As we introduced in §II-B, established attacks are either untargeted, so that $\mathcal{I} = \bigcup_{s\in\mathcal{Y}} \bigcup_{t\in\mathcal{Y}\setminus\{s\}} \{\{(s,t)\}\}$, avoiding correct classifications; or targeted so that $\mathcal{I} = \bigcup_{s\in\mathcal{Y}} \{\{(s,t_s)\}\}$ for a specific $t_s \in \mathcal{Y} \setminus \{s\}$, seeking a specific impersonation. For example, untargeted attacks against GTSRB would be represented as the former: the set of sign types would be represented by $\mathcal{Y}$, and attackers would attempt to perturb a sign $x$ so that instead of being correctly classified as belonging to class $s$ it is misclassified as any other class $t \in \mathcal{Y} \setminus \{s\}$.

To our knowledge, no existing evasion attack uses choices of $\mathcal{I}$ other than the two listed above. Consistently with that, we were unable to find defenses that are designed specifically for choices of $\mathcal{I}$ other than the two. However, as our example scenarios start to illustrate, there are many more other choices of $\mathcal{I}$ worthy of examination. In the example where students are trying to access restricted materials, $s$ could be any one of the students and $t$ could be any one of the TAs or professors. We denote the set of all student classes as $S$ and the set of TAs and professors as $T$, where $S \subseteq \mathcal{Y}$, $T \subseteq \mathcal{Y}$, and $S$ is disjoint from $T$. An attack succeeds if any student can impersonate any one of the TAs or professors, and thus we have $\mathcal{I} = \bigcup_{s\in S} \bigcup_{t\in T} \{\{(s,t)\}\}$, which is different from the $\mathcal{I}$ in traditional targeted or untargeted attacks.

In the example where attackers are perturbing traffic signs to slow down traffic, $s$ could be any of the speed limit and delimit signs. The set of speed limit and delimit signs is now $S \subseteq \mathcal{Y}$. However, for each $s \in S$, the set of classes the adversary wishes to target might be different. For example, a 20 KPH sign might be perturbed as a stop, no-entry, or no-vehicle sign, whereas a 120 KPH sign might be perturbed as a 20, 30, 50, 60 KPH, stop, no-entry, or no-vehicle sign. Thus, attackers might have *different sets of target classes $T_s$ for different choices of $s$*. Attackers succeed in slowing traffic down if they perturb any of the speed limit and delimit signs into any of the corresponding target classes, and thus we have $\mathcal{I} = \bigcup_{s\in S} \bigcup_{t\in T_s} \{\{(s,t)\}\}$.

These new choices of $\mathcal{I}$ are able to describe the goal of attackers trying to slow traffic down and students aiming to steal access-restricted materials, while traditional $\mathcal{I}$ of untargeted or targeted attacks are not able to do so. Formalizing the attackers' goals in this way reveals that current evasion attacks are not optimized for those goals.

Notice also that established attacks count success on a per-input-instance basis, using $|X| = 1$ where $X$ is the set of input instances sampled at a time, although $X$ might be resampled and $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ might be repeated many times. Here we examine cases where $|X| > 1$, enabling consideration of attacker goals for which, for example, each $I_i \in \mathcal{I}$ is a surjective function mapping classes $S$ to a target set $T$ of classes where $S \cap T = \emptyset$. In the burglary example, in which burglars impersonate several staff members of a bank to hack into a vault, $X$ is a set of images of the burglars at the time of the attack. Different burglars might impersonate different staff and hence $|X| > 1$. For each $I_i$, $S$ is a set consisting of a subset of burglars (since a subset may be enough to impersonate a sufficient number of bank staff) and $T$ is a set of several staff who together are allowed access to the vault. To achieve an $I_i$, burglars might need to use more than one $x \in X$. In this attack scenario, compared with $\mathcal{I}$ of untargeted or targeted attacks, the new choice of $\mathcal{I}$ also intuitively better depicts the burglars' goal: successfully impersonating multiple different people. We propose three attack strategies $A$ in §III-B that boost $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ for this new $\mathcal{I}$.

Our work serves as a step to search a wider space outside the crowded paradigm of existing works. New choices of $\mathcal{I}$ depict attack scenarios that often-used choices cannot.

### E. Experiment Setup

Now we turn to the experiment setups we employed to corroborate that group-based robustness complements our understanding of robustness from previously established metrics.

*1) Threat Model:* Adversaries can create more successful perturbations by acquiring more information about the architectures and weights of models [57]. The most successful attacks are white-box attacks, where the adversaries have access

to all weights of models [58], [59]. Accordingly, we use white-box evasion attacks to evaluate models to better understand the worst-case threat and to more accurately evaluate the existing defenses in the presence of the strongest adversaries.

*2) Datasets:* We used three image datasets and one text dataset to empirically measure $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ and $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ in the three scenarios described in §II-B: We used a traffic-sign dataset (GTSRB) [38] for the scenario where attackers are trying to perturb traffic signs. GTSRB consists of images of 43 traffic signs, which include but are not limited to the speed limit, speed delimit, and signs that require an immediate stop (see Fig. 1). In scenarios where students are trying to steal access-restricted materials and burglars are trying to hack a bank, one group of attackers–students or burglars, respectively–is trying to impersonate a group of victims. In both scenarios, the attackers and victims are two mutually exclusive groups of people. We used a human-face dataset (PubFig) [60], previously used in adversarial machine learning studies [50], [61], [62], for both scenarios. PubFig consists of images of 60 identities and an average of 128 images per identity. Face-recognition DNNs may need to classify more than 60 identities; some of these identities might be neither attackers nor victims of impersonation. However, the existing defense only used 10 identities [62]. With more than 60 identities, we could neither find a benchmark that has performance close to the existing defense [62] nor could we train one. As an alternative, we used an object-recognition dataset (ImageNet) [63] to mimic scenarios where there exist many identities that are neither attackers nor victims. ImageNet consists of images of 1000 objects, and we use these 1000 object classes to mimic 1000 identities. Besides image datasets, we also used one text dataset, SST-5 [64]. SST-5 has five classes, namely "very positive", "positive", "neutral", "negative" and "very negative".

*3) Benchmarks:* White-box evasion attacks have proved successful nearly $100\%$ of the time on models not specially tuned to be robust, but these attacks are less effective against models that have been tuned to be robust [40], [65]. Thus, to fairly compare the effectiveness of attacks, and to precisely compute $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ and $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ in the scenarios described in §II-B, we ran attacks against defended models. In particular, we used the following state-of-the-art defenses against white-box evasion attacks as benchmarks.

*GTSRB:* For GTSRB, we trained defenses using the free adversarial training algorithm [65]. We used two different $L_p$-norms, $L_\infty = 8/255$ and $L_2 = 0.5$, as previous works did for images with similar sizes [37], [39]. We used six different architectures including five established architectures: VGG [66], ResNet [67], SqueezeNet [68], ShuffleNet [69], and MobileNet [70]. For each combination of architecture and $L_p$-norm, we trained 100 instances for 100 iterations using the same implementation and data. The order of samples was also the same while training model instances. The only difference between instances of the same architecture was the random initialization of weights.

*PubFig:* For the PubFig dataset [60], in line with prior work, we preprocessed the face images by taking central crops and aligning faces to frontal poses via affine transformations [50], [61], [71], [72]. We split the data into 70%-20%-10% for training, testing, and validation, respectively. We adversarially trained DNNs via Wu et al.'s method [62], using their implementation. Starting from a pre-trained feature extractor based on the VGG architecture [66], Wu et al. attached a two-layer classification head and conventionally trained the DNN, minimizing cross entropy. Next, they fine-tuned their model over several epochs of adversarial training. In each iteration of adversarial training, their method located the central region of an adversarial rectangular patch via a gradient-based search. Subsequently, the rectangular patch was perturbed to induce misclassification. The resulting misclassified image with the patch and a correct label was then used to update the DNN's weights. We trained the DNN conventionally for 30 epochs, and adversarially for 5 epochs, using Wu et al.'s default choice of the optimizer (Adam [73]), step size (4), and batch size (64 for conventional training, and 32 for adversarial training). One common attack used to evaluate the robustness of facial recognition systems is the eyeglasses attack [50] which limits the perturbation to be within an eyeglasses-frame-shaped region (rather than any $L_p$ distance), simulating that attackers wear carefully painted eyeglasses to evade face recognition. The adversarially trained DNN achieved 98.25% benign test accuracy and 45.43% untargeted robustness against the eyeglass attack. By contrast, the conventionally trained model achieved 99.80% benign accuracy but only 9.14% untargeted robustness. As we described in §II-D, untargeted robustness is $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ when $\mathcal{I} = \bigcup_{s\in\mathcal{Y}} \bigcup_{t\in\mathcal{Y}\setminus\{s\}} \{\{(s,t)\}\}$ .

*ImageNet:* Face-recognition DNNs might need to classify more than 60 identities; many of these identities are neither attackers nor victims. However, we were not able to find a pre-trained face-recognition defense that uses more than 60 identities. For example, Wu et al. used a subset of VGG-Face [66] which includes ten identities [62]. We also tried to train face-recognition defense using existing methods, but the performance of our trained instances was much worse than the performance (of defenses with less than 60 identities) reported in previous works. As a mitigation, we used ImageNet instead. ImageNet has 1000 classes and we found two pre-trained state-of-the-art defenses on ImageNet by Salman et al. [74]. One of them was trained with adversarial perturbations of $L_\infty$-norm of 8/255, and the other one was trained perturbations of $L_2$-norm of 3.0. We used these two defenses to mimic face-recognition defenses that are capable of recognizing 1000 identities.

*SST-5:* We used a pre-trained model [75] on SST-5, which achieves $55.8\%$ accuracy (within top five performance at the time we conducted experiments, according to a leaderboard [76]). This model has not been specifically tuned for robustness.

*4) Measurement Process:* We implemented $\mathcal{G}$ for each test $\mathcal{I}$ on different datasets. To measure $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, we still conventionally used $|X| = 1$ where $\mathcal{G}$ always outputs $X = \{x\}$, where $x$ is one input instance, uniformly sampled from all instances associated with some $s \in S$.

On the GTSRB dataset, we tried to perturb speed limit and delimits signs to signs that would mandate (1) an immediate stop or (2) no more than half of the actual limit (shown in Fig. 1). $\mathcal{I} = \bigcup_{s\in S} \bigcup_{t\in T_s} \{\{(s,t)\}\}$. Images from different classes may not have the same set of target classes. As we introduced in §II-D, for each $s \in S$, the set of targeted classes $T_s$ might be different. A 20 KPH sign might be perturbed as a stop, no-entry, or no-vehicle sign, and a 120 KPH sign might

be perturbed as a 20 KPH, 30 KPH, 50 KPH, 60 KPH, stop, no-entry, or no-vehicle sign.

On the PubFig dataset, we have $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T}\{\{(s,t)\}\}$ as in the scenario where students are trying to steal access-restricted materials. We randomly selected two mutually exclusive sets of classes $S$ and $T$, and tried to perturb all images associated with $S$ as $T$. We used the following sizes of $S$ and $T$:

$$(|S|,|T|) \in \{(10,10),(10,20),(10,30),(10,40),(10,50),\\ (20,10),(20,20),(20,30),(20,40),(30,10),\\ 30,20),(30,30),(40,10),(40,20),(50,10)\} \quad (1)$$

These choices include all possible choices of $|S|$ or $|T|$ that are multiples of 10. For each $(|S|,|T|)$, we randomly selected 5 different pairs of subsets $S$ and $T$.

On the ImageNet dataset, we still have $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T}\{\{(s,t)\}\}$, because the attack scenario is the same as on PubFig. We first randomly selected 60 classes, and then selected $S$ and $T$ sets of sizes in Eqn. 1 from these 60 classes, as we did for PubFig.

On the SST-5 dataset we used four different goals of the adversary: 1) perturb positive instances as nonpositive (i.e. perturb instances from the "very positive" or "positive" classes as any of the rest three classes) 2) perturb negative instances as nonnegative 3) perturb each instance as a more positive class and 4) perturb each instance as a more negative class. For each of these four goals, similar to the traffic sign scenario, $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T_s}\{\{(s,t)\}\}$ with respect to different choices of $S$ and $T_s$.

On all the benchmarks trained with $L_p$-norm (benchmarks on GTSRB and ImageNet), we ran Auto-PGD [39] attacks using the same $L_p$ distance. That is, in $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, if attacking a model $\tilde{f}$ trained with $L_\infty = 8/255$, for example, then $\Pi(x,x') = 1$ if and only if $L_\infty(x,x') \le 8/255$. To the best of our knowledge, Auto-PGD attacks are the strongest currently available $L_p$-norm attacks that do not require model-specific tuning. On the DOA defenses, we ran eyeglasses attacks [50] implemented by the authors of DOA [62]. On the SST-5 dataset, we ran T-PGD attacks [77], a state-of-the-art text domain attack that empirically achieves human imperceptibility. In our experiments, we exclusively modified the loss function of any attacks we used. Using their default settings, we ran Auto-PGD for attacks for 100 iterations, eyeglasses attacks for 300 iterations, and T-PGD attacks for 100 iterations.

### F. Results

§II-D showed how our new metric, group-based robustness, conceptually accommodates real-world attack scenarios that existing metrics cannot. To empirically show this, we measured group-based robustness along with three metrics: accuracy, untargeted robustness, and targeted robustness. As we described in §II-D, untargeted robustness is $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ when $\mathcal{I} = \bigcup_{s\in\mathcal{Y}}\bigcup_{t\in\mathcal{Y}\setminus\{s\}}\{\{(s,t)\}\}$, and targeted robustness is $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ when $\mathcal{I} = \bigcup_{s\in\mathcal{Y}}\{\{(s,t_s)\}\}$ for a specific $t_s \in \mathcal{Y}\setminus\{s\}$. When measuring group-based robustness, we used choices of $\mathcal{I}$ motivated by attack scenarios described in §II-B.

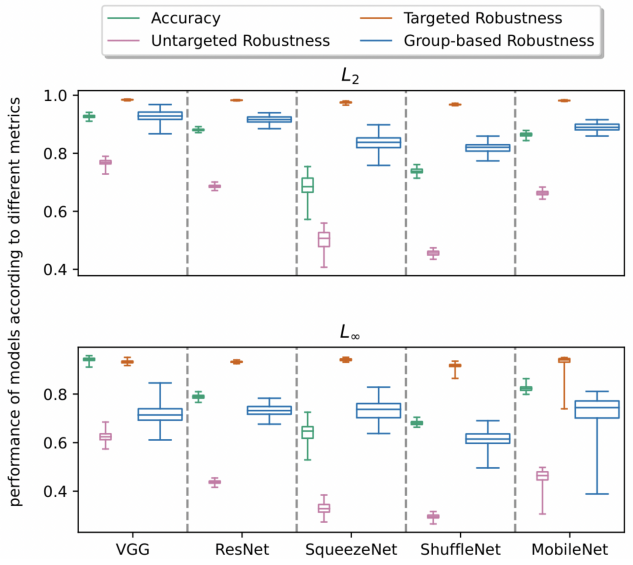


Fig. 2. Performance of models on GTSRB measured by four metrics: accuracy, untargeted robustness, targeted robustness, and group-based robustness. With each combination of $L_p$-norm and architecture, the distribution of group-based robustness, depicted as the wider boxes, is different from those of the other three metrics. With each combination, the performance of models varies only due to different randomly initialized weights, using seeds 0 – 99.

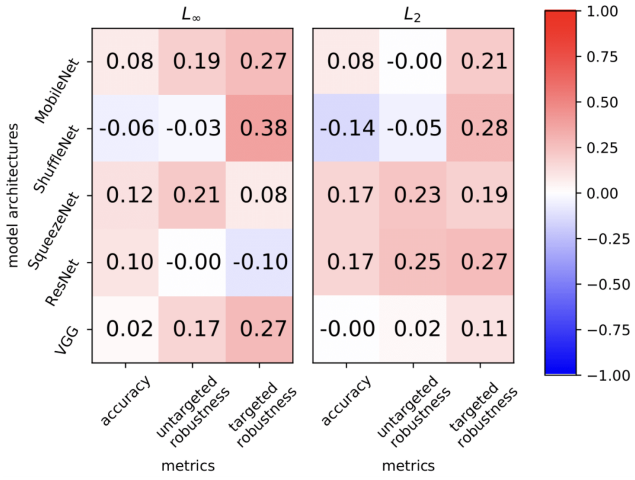


Fig. 3. Pearson correlation coefficients between group-based robustness and three existing metrics: accuracy, untargeted robustness, and targeted robustness on GTSRB. Across most of the combinations of model architecture and $L_p$-norm, the correlations are negligible or weak as the coefficients have a magnitude smaller than 0.4 [78].

On the benchmarks we trained on GTSRB, as shown in Fig. 2, group-based robustness has different distribution (mean and range) from the other three metrics. For some combinations of architecture and $L_p$-norms, the range of group-based robustness, the distribution barely overlaps with those of other metrics. Meanwhile, the variance for group-based robustness is higher than the variance of other metrics at most of the combinations of architecture and $L_p$-norm. Models with close performance by other metrics can have very different performance by group-based robustness.

We verified the difference statistically: as shown in Fig. 3, at each combination of architecture and $L_p$-norm, the Pearson

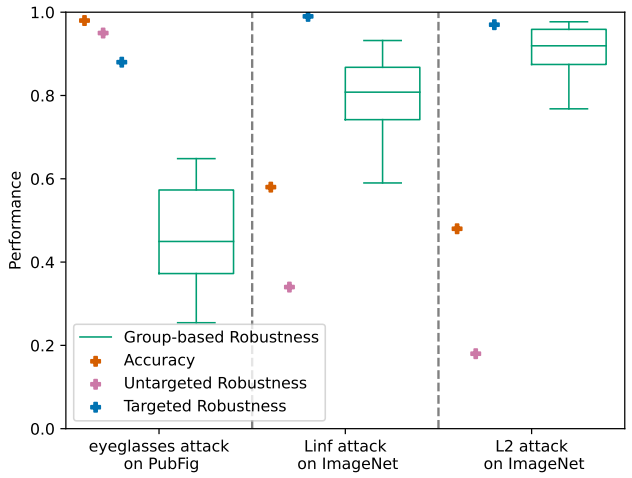


Fig. 4. Performance of models measured by accuracy, untargeted robustness (UR), targeted robustness (TR), and group-based robustness (GBR). These models were adversarially trained with PubFig and ImageNet. On each model, GBR has a wide range due to different choices of $T$ and $S$, whereas the other metrics report only a single value that is sometimes out of the GBR range.

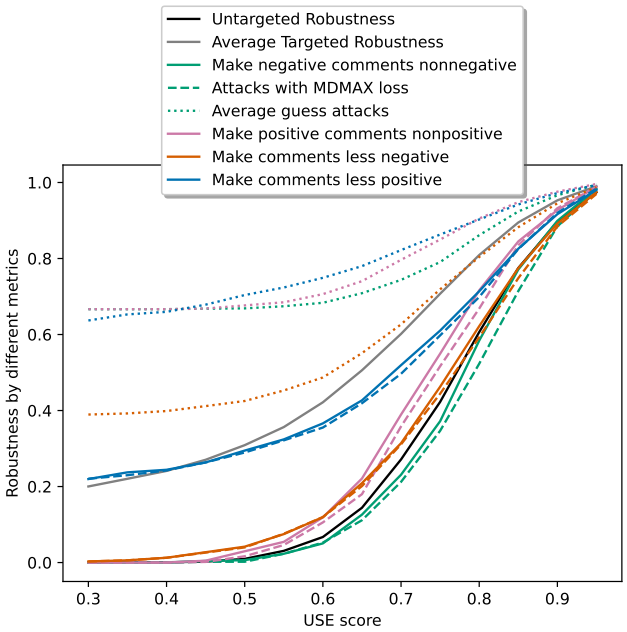


Fig. 5. Robustness as defined by different metrics on the SST-5 dataset. The USE score, as used by T-PGD, denotes the imperceptibility of the perturbation: the higher the USE score is, the more imperceptible the perturbation is to humans. Group-based robustness has a wide range, whereas untargeted robustness and targeted robustness are sometimes out of the range.

correlation coefficients between group-based robustness and each of the three metrics are always between $-0.4$ to $0.4$. According to Schober et al., correlations are weak or negligible of the Pearson coefficient is between $-0.4$ and $0.4$ [78]. Group-based robustness is always negligibly or weakly correlated with each of the three metrics.

We also measured the performance of benchmarks we have on the PubFig and ImageNet datasets with these four metrics. As shown in Fig. 4, group-based robustness reports a wide range on each of the three models, whereas each of the other three metrics reports a single number, some laying outside the range. Group-based robustness reports different numbers due to different choices of $S$ and $T$, which correspond to different attack scenarios, such as adversaries slowing traffic down or burglars hacking into a vault at a bank. We also measured group-based robustness, along with targeted and untargeted robustness as shown in Fig. 5. The benign accuracy is a constant number, and similar to what we saw on other datasets, group-based robustness reports a range, and targeted or untargeted robustness is sometimes outside the range.

**Takeaways (Metric)**

Group-based robustness $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ measures the robustness of models using different choices of $\mathcal{I}$ in accordance with the attack scenarios. Conventional metrics cannot measure the true threat in these sophisticated scenarios as accurately as group-based robustness does. Thus, we conclude that group-based robustness offers a new meaningful assessment of model susceptibility to attacks in the real world compared to conventional metrics.

## III. More Efficient Attacks

In this section, we introduce several algorithms $A$ that either increase the advantage of attacks $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, or boost the speed to achieve a close advantage of attacks, helping attacks become more computationally efficient than existing naïve attacks. We start by describing two loss functions to help adversaries perturb one input instance so that the input instance is misclassified as any of a specific set of target classes (§III-A). Then, we introduce three attack strategies to help adversaries perturb several input instances so that they are misclassified as different target classes among a specific set (§III-B).

Later, we also leverage these new attacks to build defenses against group-based attacks (see §IV).

### A. Attack Loss Functions

In certain scenarios, adversaries may have a limited amount of time or attempts to attack systems. In the bank robbery example, burglars might only have a brief time window to access the face recognition system, and they might trigger an alarm if they consecutively make many failed attempts to impersonate bank staff. Meanwhile, it might be impractically costly both for the attackers to try impersonating each of the employees as a brute-force approach, and for the bank to assess the group-based robustness in such a manner, as the bank might have many staff (e.g., Dresdner Bank, a major European bank, has 50,659 employees [79]). Thus, we need group-based attacks that are more time-efficient than the brute-force approach. We designed two new loss functions that formalize the attackers' goal in such scenarios.

We came up with these two loss functions by modifying a state-of-the-art loss function for targeted attacks, the Minimal Difference (MD) loss [37], which outperforms well-known loss functions such as the Carlini-Wagner (CW) loss [51] and the

Difference of Logits Ratio (DLR) loss [39]. It aims to assign the highest logit to the target class:

$$\ell_{MD} = \sum_i ReLU(Z_i + \delta - Z_t) \tag{2}$$

where $i$ iterates over all classes, $t$ is the target class, $Z$ is the logit, $\delta$ is a minimal value set to 1e-15, and *ReLU* is the rectified linear unit function. Said differently, attacks $A$ with the MD loss have higher $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than attacks with other previously proposed loss functions, for targeted attacks with $\mathcal{I} = \bigcup_{s\in\mathcal{Y}\setminus\{t\}} \{\{(s,t)\}\}$ for a specific target class $t \in \mathcal{Y}$, perturbing more input instances $x \in \mathcal{X}$ to be misclassified as the target class $t$. The attack succeeds *if and only if* the MD loss is zero. When the MD loss is zero, $Z_t$ is larger than any other $Z_i$ by at least $\delta$, and thus the attack succeeds. On the other hand, if the attack succeeds, $Z_t$ is the largest logit, larger than any other $Z_i$ by at least the minimal value $\delta$; thus the MD loss is zero.

As we explained in §II, an adversary could be interested in some set of target classes $T$, rather than a single target class $t$. In this case, an attack is considered successful if the adversary can cause an input from a class in $S$ to be misclassified as *any* class within $T$. More formally, as we described in §II-D, in the example where students are trying to steal access-restricted materials, $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T}\{\{(s,t)\}\}$, where $S$ is the set of all students and $T$ is the set of all TAs and professors. If a student impersonates *any* of the TAs or professors, the attack succeeds. In the example where attackers perturb signs to slow traffic down, $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T_s}\{\{(s,t)\}\}$, $s$ is one of the speed limit and delimit signs, and $T_s$ is the corresponding set of signs that can mislead traffic to be slower than intended. The attackers succeed if they can perturb a sign from the class $s$ to be classified as any sign type in $T_s$. Notice that $T_s$ depends on $s$, e.g., the attack succeeds if a 20 KPH sign is perturbed into a stop, no-entry, or no-vehicle sign; or if a 120 KPH sign is perturbed into a 20, 30, 50, 60 KPH, stop, no-entry, or no-vehicle sign.

A naïve exhaustive approach to carry out such misclassification would be to launch $|T|$ targeted attacks, one for each $t \in T$. That is, to maximize $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, $A$ invokes subroutines $A_t$ to find an impersonation for each $t \in T$ independently. $A$ succeeds when *any* of the $|T|$ targeted attacks succeed. When $\mathcal{I} = \bigcup_{s\in\mathcal{Y}}\bigcup_{t\in\mathcal{Y}\setminus\{s\}}\{\{(s,t)\}\}$ and so $T = \mathcal{Y}\setminus\{s\}$, this naïve approach finds more adversarial examples, obtaining higher $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than any other untargeted attacks $A$ that aim to directly maximize $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ [39]. However, this naïve exhaustive approach requires running targeted attacks $|T|$ times and thus is $|T|$ times more costly to run than untargeted attacks. Another naïve approach that does not suffer from the same overhead is to randomly pick a class $t \in T$ and launch a targeted attack $A_t$ targeting $t$. However, we found this approach finds significantly fewer adversarial examples, obtaining much smaller $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ (more details can be found in §III-A4).

To address the shortcomings of the naïve approaches, we propose two new loss functions for when $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T}\{\{(s,t)\}\}$ or $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T_s}\{\{(s,t)\}\}$. These loss functions help attackers obtain larger $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the non-exhaustive naïve approach, and obtain a close $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ while consuming much less computation time than the exhaustive naïve approach.

*1) The MDMUL Loss:* We propose a new loss function following the intuition that attackers only need one class in the targeted set $T$ to have a higher logit than any classes not in the set. In particular, we formalize the attackers' goal—to assign a higher logit to some $t \in T$ than to any $i \notin T$—as $\sum_{i\notin T} ReLU(Z_i + \delta - Z_t)$. This term is always non-negative, and it is zero if and only if the corresponding $t$ has higher logit than any $i \notin T$. To capture that an adversary only needs one $t \in T$ to have higher logit than any $i \notin T$, we can write $\prod_{t\in T}\sum_{i\notin T} ReLU(Z_i + \delta - Z_t)$, which, again, evaluates to zero if and only if the attack succeeds. Due to the finite arithmetic of Python, in which we implement these loss functions, this product can be $\infty$ and yield undefined gradients. As a remedy, we compute the natural logarithm of the product instead and come up with the Minimal Difference Multiplied (MDMUL) loss:

$$\ell_{\text{MDMUL}} = \sum_{t\in T} ln(\sum_{i\notin T} ReLU(Z_i + \delta - Z_t)) \tag{3}$$

where $t$ iterates over all classes in $T$ and $i$ iterates over all classes not in $T$. We acknowledge that mathematically $ln(0)$ is undefined, while Python computes $ln(0)$ as $-\infty$. Each natural logarithm result is $-\infty$ if and only if a successful attack has been found: $Z_t$ is larger than all $Z_i$, and the whole equation is $-\infty$ if and only if at least one of the natural logarithm results is $-\infty$. Thus $\ell_{\text{MDMUL}}$ is $-\infty$ if and only if a successful attack has been found.

*2) The MDMAX Loss:* Attackers only need one class in the targeted set $T$ to have a higher logit than any classes not in the set. One strategy to achieve this is to greedily keep trying to increase the current maximum logit from among the targeted classes, perturbing inputs toward the target class that is most likely to succeed. We formalize this approach as the Minimal Difference Maximum (MDMAX) loss:

$$\ell_{\text{MDMAX}} = \sum_{i\notin T} ReLU(Z_i + \delta - \max_{t\in T} Z_t) \tag{4}$$

where $i$ iterates over all classes $\notin T$ and the largest $Z_t$ among all classes $t \in T$ is used. $\ell_{\text{MDMAX}}$ is also non-negative and zero if and only if a successful attack has been found: if $\ell_{\text{MDMAX}} > 0$, there is at least one class $i \notin T$ that has higher logits than all classes $t \in T$; otherwise, if $\ell_{\text{MDMAX}} = 0$, there is at least one class $t \in T$ that has higher logits than all classes $i \notin T$.

*3) Experiment Setup:* To illustrate the new loss functions' improvements for adversaries $A$, either in terms of advantage or speed, we used the same threat model, datasets, and benchmarks as we did in §II-E. We also implemented $\mathcal{G}$ for each test $\mathcal{I}$ on different datasets as we did in §II-E: we used conventionally $|X| = 1$ where $\mathcal{G}$ always outputs $X = \{x\}$, and $x$ is one input instance, and uniformly sampled from all instances associated with *some* $s \in S$. In addition, we used the baselines and measurement process below.

*Baselines:* We created three baseline attacks to compare with attacks using loss functions proposed in §III-A. As we elaborated in §II, adversaries could face scenarios where they succeed by forcing any misclassifications within a set $T$ of target classes. More formally, they seek to maximize

$\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, when $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T}\{\{(s,t)\}\}$ or $\mathcal{I} = \bigcup_{s\in S}\bigcup_{t\in T_s}\{\{(s,t)\}\}$ (introduced in §II-D). They could exhaustively iterate over all classes in $T_s$ or randomly pick a $t \in T_s$, and launch targeted attacks. We depict these naïve methods as follows:

- The best guess: with every $x \in X$ to be perturbed, $A$ could iterate over all $t \in T_s$, and launch $|T_s|$ targeted attacks. Each targeted attack $A_t$ aims to produce pairs $(x, x')$ such that $\tilde{f}(x') \in t$. $A$ succeeds on this $x$ if *any* of the $|T_s|$ targeted attacks succeed. This approach tends to be the strongest naïve approach, obtaining the highest $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$. However, as it searches exhaustively, $A$ is $|T_s|$ times as expensive to run as each $A_t$.
- The average guess: on each $x \in X$, $A$ randomly picks a $t$ from $|T_s|$ and outputs pairs $(x, x')$ such that $\tilde{f}(x') = t$. $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of the average guess attacks is the mean of all $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}_t}(A)$ whose $\mathcal{I}_t = \bigcup_{s\in S}\{\{(s,t)\}\}, t \in T$. The average running time of $A$ is the average running time of all $A_t$.

We run the two naïve methods with $\ell_{MD}$, the state-of-the-art loss function for targeted attacks (introduced in §III-A).

*Measurement Process:* We computed $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of four attacks, using two new loss functions, the MDMAX loss and the MDMUL loss, and the two baseline naïve attacks, on the three image datasets. We also ran attacks with the MDMAX loss along with the baselines on the SST-5 dataset. The performance of attacks is computed on a per-image basis: in experiments related to the new loss functions, $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ always uses a $\mathcal{G}$ that samples $X$ of size $|X| = 1$ uniformly.

*4) Results:* On the GTSRB dataset, as illustrated in Fig. 1, $|T|$ ranges from 3 (for a 20 KPH sign) to 7 (for a 120 KPH sign). Compared with the best guess attacks, attacks with the MDMAX loss or the MDMUL loss take intuitively and empirically one-third to one-seventh of the time on GTSRB (more details later). As shown in Fig. 6, the $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of attacks with new loss functions is 0.62–1.04× that of the best guess attacks. Attacks with the MDMAX loss or the MDMUL loss take much less time than the best guess attacks, but obtain close $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$. The average guess attacks take the same amount of time as attacks with the MDMAX loss or the MDMUL loss on GTSRB. The advantages of attacks with the new loss functions are 1.04–2.56× that of the average guess attacks, i.e., always larger than the advantages of average guess attacks. On the PubFig and ImageNet datasets, $|T|$ ranges from 10 to 50. On the PubFig dataset, when $|T|$ increases, the $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of attacks with the MDMAX loss, attacks with the MDMUL loss, and the best guess attacks also increase, whereas the $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of the average guess attacks stay about the same. We observe the same phenomenon in these two datasets (shown in Fig. 7–Fig. 9). Since $|T|$ ranges from 10 to 50 on PubFig and ImageNet, the best guess attacks are 10 to 50 times slower than attacks with the MDMAX loss or the MDMUL loss. The $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of new loss functions is 0.30–1.21× as large as that of the best guess attacks on PubFig, and 0.62–1.15× on ImageNet. The $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of new loss functions is 1.19–6.22× as large as that of the average guess attacks on PubFig, and 7.37–41.53× on ImageNet. On the SST-5 dataset, attacks with the MDMAX loss obtain $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ that is not only larger than the average guess attacks but also unintuitively larger than the best guess attacks, as shown in Fig. 5. T-PGD uses a weighted sum of two parts as the loss function: the first part aims to induce misclassification, which we replace with the MDMAX loss, and the second part aims to increase the USE score. Using the MDMAX loss, T-PGD finds successful attacks earlier and thus spends more iterations to increase the

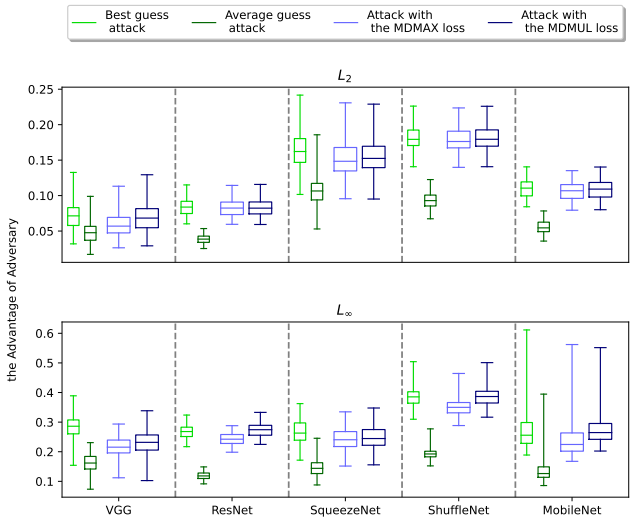


Fig. 6. $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of four attacks, two naïve methods and two with new loss functions, on the GTSRB dataset. Although the ranges of advantages of different attacks overlap, attacks with the MDMAX loss or the MDMUL loss, depicted as the wider boxes, usually have slightly lower $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the best guess attack but always have higher advantages than the average guess attack. With each combination of $L_p$-norm and architecture, the performance of models varies only due to different randomly initialized weights, using seeds 0 – 99.

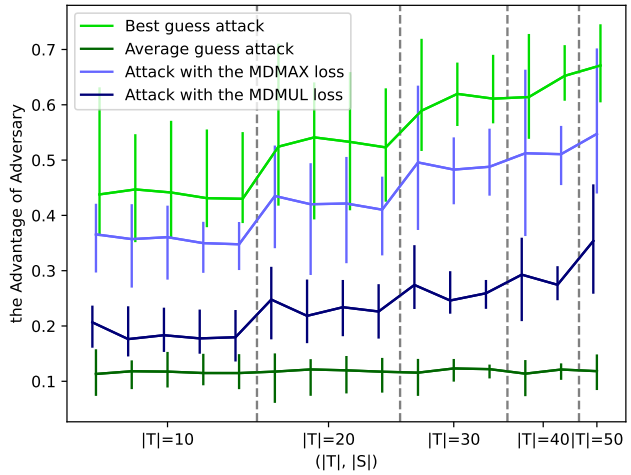


Fig. 7. Advantages of four attacks, two naïve methods and two with new loss functions, on the PubFig dataset. Ranges are due to choices of $S$ and $T$. Although the ranges of advantages of different attacks overlap, for each specific $(S, T)$, attacks with the MDMAX loss or the MDMUL loss always have lower $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the best guess attack but have higher advantages than the average guess attack.

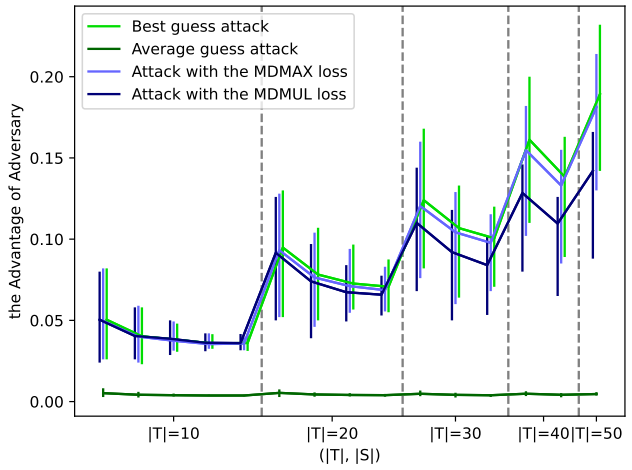


Fig. 8. Advantages of four attacks, two naïve methods and two with new loss functions, on the ImageNet dataset with $L_2$-norm. Ranges are due to choices of $S$ and $T$. Although the ranges of advantages of different attacks overlap, for each specific $(S, T)$, attacks with the MDMAX loss or the MDMUL loss usually have slightly lower $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the best guess attack but always have higher advantages than the average guess attack.

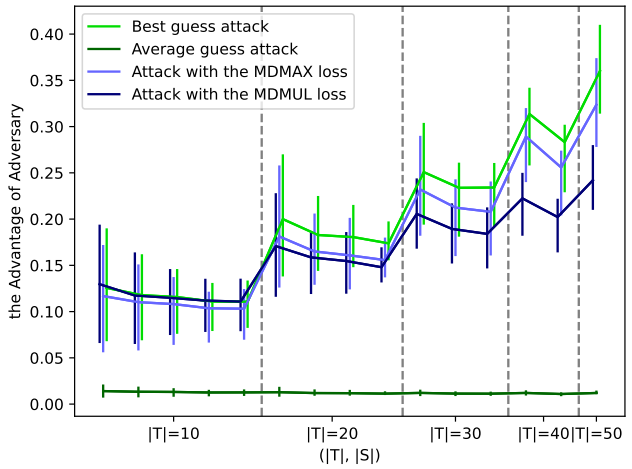


Fig. 9. Advantages of four attacks, two naïve methods and two with new loss functions, on the ImageNet dataset with $L_\infty$-norm. Ranges are due to choices of $S$ and $T$. Although the ranges of advantages of different attacks overlap, for each specific $(S, T)$, attacks with the MDMAX loss or the MDMUL loss usually have slightly lower $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the best guess attack but always have higher advantages than the average guess attack.

USE score. In summary, attacks with the MDMAX loss or the MDMUL loss always obtain larger advantages than the average guess attacks do.

To empirically verify that the two new loss functions reduce the computation time of attacks, we measured the average running time to perturb one batch of images on an RTX 3090 GPU. We used 10 images from ImageNet, 64 images from PubFig, or 512 images from GTSRB, as one batch of images. We used the largest $|T|$ for each dataset: 50 for ImageNet and PubFig, and 7 for GTSRB. We compared four attacks: the best guess attacks, the average guess attacks, attacks with the MDMUL loss, and attacks with the MDMAX loss. The detailed results are shown in Fig. 11. The results confirmed our hypothesis: attacks with the MDMUL loss or the MDMAX

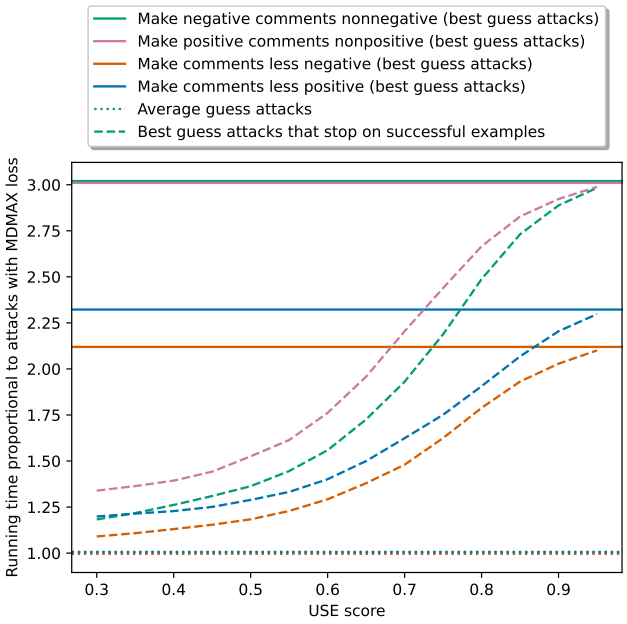


Fig. 10. Running time of two baselines and attacks with the MDMAX loss on SST-5. Attacks with the MDMAX loss take about the same time as the average guess attacks. As the USE score increases, finding successful attacks becomes harder, and best guess attacks take much more time than attacks with the MDMAX loss.

Fig. 11. Average time (seconds) to perturb one batch of images. Intuitively attacks with $\ell_{\text{MDMAX}}$ or $\ell_{\text{MDMUL}}$ are faster by a factor of $|T|$ than the best guess attacks. We measured the average running time on a RTX 3090 GPU, with the largest $|T|$ for each dataset: 50 for ImageNet and PubFig, and 7 for GTSRB.

| *Dataset* | $L_p$ *norm* | *architecture* | *Best guess attack* | *Average guess attack* | *Attack with* $\ell_{\text{MDMAX}}$ | *Attack with* $\ell_{\text{MDMUL}}$ |
|---|---|---|---|---|---|---|
| ImageNet | $L_2$ | – | 279.44 | 5.59 | 5.07 | 5.67 |
| | $L_\infty$ | – | 280.89 | 5.61 | 5.06 | 5.51 |
| PubFig | – | – | 845.93 | 16.92 | 15.33 | 16.48 |
| GTSRB | $L_2$ | VGG | 18.54 | 2.65 | 2.48 | 2.59 |
| | | ResNet | 18.01 | 2.57 | 2.39 | 2.51 |
| | | SqueezeNet | 11.27 | 1.61 | 1.57 | 1.59 |
| | | ShuffleNet | 17.57 | 2.51 | 2.37 | 2.51 |
| | | MobileNet | 19.88 | 2.84 | 2.44 | 2.53 |
| | $L_\infty$ | VGG | 17.18 | 2.45 | 2.28 | 2.32 |
| | | ResNet | 17.04 | 2.43 | 2.24 | 2.35 |
| | | SqueezeNet | 11.16 | 1.58 | 1.47 | 1.64 |
| | | ShuffleNet | 16.84 | 2.41 | 2.25 | 2.48 |
| | | MobileNet | 18.06 | 2.58 | 2.37 | 2.57 |

loss are faster than the best guess attacks by a factor of $|T|$. We also noticed that if adversaries choose to perturb images by instances, instead of conventional batches, they might save time to run the best guess attacks by stopping perturbing an image that they have already succeeded in perturbing. While empirically such a variant of the best guess attacks save time for the adversaries, the time needed by this variant is still much larger than attacks with the MDMUL loss or the MDMAX loss. The detailed results are shown in Fig. 11. We observe the similar results as shown in Fig. 10: attacks with the MDMAX loss take about the same amount of time as average guess attacks, and they take much less time especially when the USE

Fig. 12. Average time (seconds) to perturb one image. Intuitively attacks with $\ell_{\text{MDMAX}}$ or $\ell_{\text{MDMUL}}$ are faster by a factor of $|T|$ than the best guess attacks. We measured the average running time on a RTX 3090 GPU, with the largest $|T|$ for each dataset: 50 for ImageNet and PubFig, and 7 for GTSRB.

| *Dataset* | $L_p$ *norm* | *architecture* | *Best guess attack* | *Best guess attack (until success)* | *Attack with* $\ell_{\text{MDMAX}}$ | *Attack with* $\ell_{\text{MDMUL}}$ |
|---|---|---|---|---|---|---|
| ImageNet | $L_2$ | – | 205.04 | 177.40 | 4.09 | 4.19 |
| | $L_\infty$ | – | 189.06 | 148.15 | 3.87 | 3.92 |
| PubFig | – | – | 135.96 | 58.21 | 2.58 | 7.77 |
| GTSRB | $L_2$ | VGG | 8.73 | 8.19 | 1.27 | 1.49 |
| | | ResNet | 10.02 | 9.28 | 1.44 | 1.64 |
| | | SqueezeNet | 10.06 | 7.47 | 1.49 | 1.54 |
| | | ShuffleNet | 16.54 | 13.70 | 2.30 | 2.45 |
| | | MobileNet | 15.55 | 13.90 | 2.25 | 2.43 |
| | $L_\infty$ | VGG | 8.18 | 5.94 | 1.24 | 1.54 |
| | | ResNet | 9.92 | 7.35 | 1.41 | 1.61 |
| | | SqueezeNet | 10.61 | 8.29 | 1.45 | 1.61 |
| | | ShuffleNet | 15.78 | 13.52 | 2.16 | 2.31 |
| | | MobileNet | 14.97 | 13.75 | 2.27 | 2.40 |

score is high.

**Takeaways (Loss Functions)**

Attacks with the MDMAX loss or the MDMUL loss achieve comparable or slightly lower $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the best guess attacks, consume markedly less time and are markedly more efficient, finding more attacks per time unit. Attacks with the MDMAX loss or the MDMUL loss consume the same amount of time as the average guess attacks and have much higher $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$. The MDMUL loss and the MDMAX loss boost the efficiency of attacks in the attack scenarios we tried.

### B. Attack Strategies

The previous section (§III-A) introduced loss functions that increased the efficiency of evasion attacks that perturb a single input toward a set containing multiple target classes. In contrast, this section introduces strategies that can be used to increase efficiency when there are also multiple inputs to be perturbed.

Previous works evaluate attacks by either the success rate $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ (e.g., [37], [39]–[50]) or $L_p$-norm distance $\Pi$ (e.g., [51]–[54]). These metrics measure the per-input-sample performance of attacks. That is, $|X| = 1$ for $\mathcal{G}$. However, as described in §II, adversaries could face scenarios where they need to cause each of several images that belong to different classes to be incorrectly classified into several other different classes. In scenarios such as the burglary example, more than one burglar may impersonate the bank staff, and more than one staff member needs to be impersonated since no staff member can grant access individually. Attacks $A$ need to maximize $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ when $|X| > 1$ and each $I_i \in \mathcal{I}$ is a surjective function mapping classes $S$ to a target set $T$ of classes where $S \cap T = \emptyset$. A major obstacle to computing $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ when $|X| > 1$ is that whether an attack can successfully perturb an input instance $x \in X$ as a target class $t \in T$ remains unknown until an exhaustive attack is fully performed. Suppose $x$ is a specific instance in $X$ generated by $\mathcal{G}$, $t$ is a specific target class $t \in T$, $s = f(x)$, $\tilde{\mathcal{I}} = \{\{(s,t)\}\}$ and $\tilde{\mathcal{G}}$ is an algorithm producing instance $\tilde{X} = \{x\}$ ($|X| > 1$ for $\mathcal{G}$ and $|\tilde{X}| = 1$ for $\tilde{\mathcal{G}}$). If the adversary $A$ can estimate the pairwise success rate $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$ to perturb $x$ to $t$, it may choose the $x$ and $t$ pairs accordingly to focus its attack, obtaining a higher probability to have $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ return 1 and thus maximizing $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ by definition. We will now introduce three strategies to estimate $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$, the pairwise success rate to perturb each $x \in X$ as each $t \in T$.

*1) Estimate by Computing a Prior from Validation Set:* $A$ can launch targeted attacks using a validation set, perturbing input instances associated with each $s \in S$ as each $t \in T$. In the bank burglary example, the burglars might collect images of staff ahead of time to construct the validation set. $A$ can compute a prior probability of perturbing input instances associated with each $s \in S$ as each $t \in T$. More formally, $\tilde{A}$ is trying to transform a random instance of $s$ to $t$, as specified by $\tilde{\mathcal{I}}$. $A$ can compute $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\tilde{\mathcal{I}}}(\tilde{A})$, and use it as an estimate of $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$. This approach does not require $x$, the actual instance to be perturbed, to estimate. In the burglary example, using this approach, the burglars $A$ do not need to maintain the same poses before the facial recognition camera when they try to impersonate different staff members. When $A$ executes, $A$ always tries the $(s,t)$ class pair with highest prior $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\tilde{\mathcal{I}}}(\tilde{A})$ first. It iterates over input instances associated with class $s$ and performs targeted attacks towards class $t$ until a successful perturbation has been found or all images have been tried. Then it repeats the above process by picking the $(s,t)$ class pair with the next highest prior, and skipping such a pair if a successful perturbation targeting $t$ has already been found.

*2) Estimate by MD Loss Without Perturbation:* As discussed in §III-A, the smaller the MD loss, the closer the attack is to succeeding. $A$ can perform one forward propagation for every input instance $x \in X$ to be perturbed to get the logits. Then $A$ can compute the MD loss towards each target class $t \in T$. The smaller the MD loss is, the larger the estimated $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$ is. When $A$ is carried out, $A$ repeatedly selects $(x,t)$ pairs with the next smallest MD loss, and it skips pairs where a successful perturbation has been found targeting class $t$.

*3) Estimate by MD Loss After One Attack Iteration:* For every input instance $x \in X$ and each target class $t \in T$, $A$ perform one iteration of the attack $\tilde{A}$ [41], [42] before computing the MD loss. The smaller the MD loss is, the larger the estimated $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$ is. When $A$ is carried out, $A$ repetitively selects $(x,t)$ pairs with the next smallest MD loss, and it skips pairs where a successful perturbation has been found targeting class $t$.

$A$ do not have to use the above three strategies independently. Although the MD loss is not a probability estimate, we expect it to be inversely correlated with the probability estimate of pairwise attack success $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$. Given an input instance $x \in X$ from class $f(x) = s \in S$, and $t \in T$, $A$ can compute the prior, $(s,t)$, $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\tilde{\mathcal{I}}}(\tilde{A})$, and MD loss of $(x,t)$, with or without adding perturbations, as above-mentioned. One approach to combine these two values is to

construct a product as $(1-prior)*\ell_{MD}$ for each $(x,t)$ pair. The smaller the product, the larger the estimated $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$ is. When $A$ is carried out, $A$ repeatedly selects $(x,t)$ pairs with the next smallest product and skips pairs where a successful perturbation has been found targeting class $t$. Other approaches to combine strategies might also work.

*4) Experiment Setup:* To illustrate the new loss strategies' improvements of $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, we used the same threat model, datasets and benchmarks as we did in §II-E. In addition, we used the baselines and measurement process below.

*Baselines:* We created a baseline strategy to compare with attack strategies proposed in §III-B. As we introduced in §II, when $|X|>1$, $A$ may consider goals for which, e.g., each $I_i \in \mathcal{I}$ is a surjective function mapping classes $S$ to a target set $T$ of classes where $S \cap T = \emptyset$. For all $x \in X$ and $t \in T$, the baseline strategy randomly selects $(x,t)$ pairs without replacement and skips $(x,t)$ pairs where a successful misclassification targeting a $t$ has been made.

*Measurement Process:* We evaluated the attack strategies on PubFig and ImageNet using the bank burglary scenario. We implemented $\mathcal{G}$ for each test $\mathcal{I}$ on different datasets. However, different from §II-E and §III-A3, we used $|X|>1$ as we introduced in §II-D, and $I_i \in \mathcal{I}$ is a surjective function. The size of the codomain of the surjective function is $K$, $K>1$. We compared the baseline strategy with five strategies: three strategies listed in §III-B, and two strategies that are combinations of the previous three (also mentioned in §III-B).

The strategy that estimates pairwise success rates of attacks, $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$, by computing a prior, $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\tilde{\mathcal{I}}}(\tilde{A})$ , requires a validation set besides the data samples we used for evaluation. PubFig has its own validation set. However, ImageNet only has a testing set, so we randomly split it by a 2:1 ratio, as PubFig has.

We sampled images from the testing sets to evaluate the attack strategies. For each of the $(S,T)$ pairs described above, we first randomly selected one $x \in X$ from each $s \in S$, for a total of $|S|$ images. Then we tried to perturb these images as any $K$ diverse classes. In the bank burglary scenario, $K$ is the least number of staff that need to agree to grant access to the treasury.

According to the estimated pairwise success rates, $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$, we select the next most likely vulnerable $(x,t)$ pairs to launch targeted attacks. We launch Auto-PGD with MD loss for each $(x,t)$ pair and counted it as one attempt. The less attempts needed to find $K$ diverse misclassifications $\in T$, the more often $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ returns 1, and the larger $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ is. As defined in §II-C, $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ is the probability that $\mathbf{Expt}^{\text{imp-rel}}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ returns 1.

Assuming at least $K$ diverse misclassifications could always be found, we compared attack strategies by the number of attempts needed to find $K$ diverse misclassifications. We ensured the assumption always holds when selecting 1000 sets of $|S|$ images for each $(S,T)$ pair. We used $K \in \{5,10,15,20,25,30,35,40,45\}$ on PubFig and $K \in \{2,3,4,5,6,7,8\}$ on ImageNet. We eliminated choices of $K$ if $K > |T|$ or the probability that the assumptions holds is smaller than 1%. For choices of $K$ that are larger than the ones we listed above, the probability that the assumptions hold was always smaller than 1%.

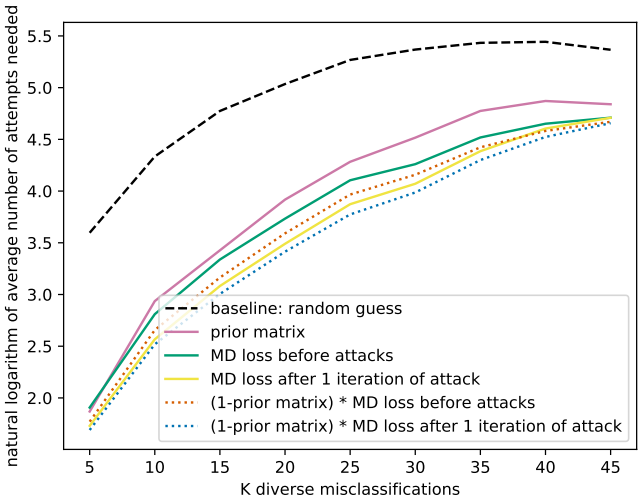


Fig. 13. The natural logarithm of the average number of attempts needed by attack strategies to find $K$ diverse misclassifications, using eyeglasses attacks on the PubFig dataset. Average is taken over images and choices of $(S,T)$.

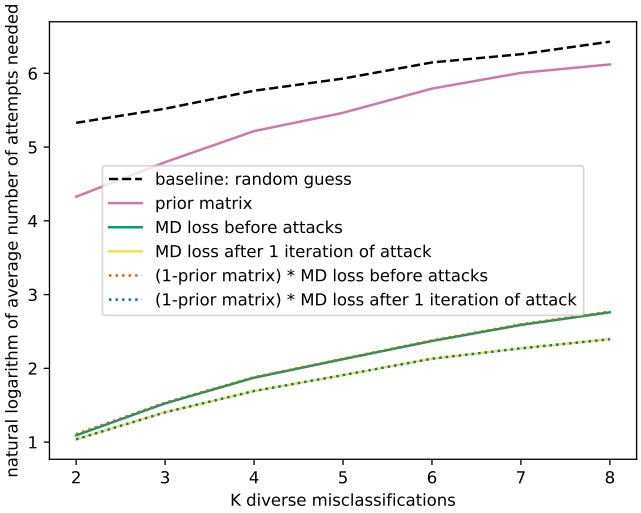


Fig. 14. The natural logarithm of the average number of attempts needed by attack strategies to find K diverse misclassifications, using $L_2$ attacks on the ImageNet dataset. Average is taken over images and choices of $(S,T)$.

*5) Results:* Now we turn to empirically demonstrate that the new attack strategies help to increase $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, where $|X|>1$ and $I_i \in \mathcal{I}$ is a surjective function. The detailed results are shown in Figs. 13–15. On all datasets and benchmarks, estimating the success rate only by the prior matrix is more efficient than random guess but less efficient than all other methods. This strategy needs only 13.48–60.25% of the number of attempts needed by the baseline on PubFig, 23.84–92.63% on ImageNet with $L_2$-norm, and 10.92–89.00% on ImageNet with $L_\infty$-norm. This strategy has larger $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than random guess.

Estimating the pairwise success rate $\mathbf{Adv}_{\Pi,f,\tilde{f},\tilde{\mathcal{G}},\tilde{\mathcal{I}}}(\tilde{A})$ with MD loss after 1 iteration of the attack is less efficient than without any perturbation only when using small values of

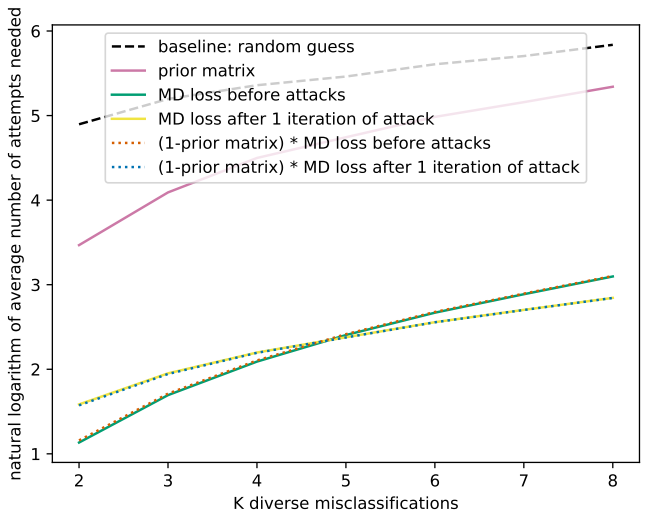


Fig. 15. The natural logarithm of the average number of attempts needed by attack strategies to find $K$ diverse misclassifications, using $L_\infty$ attacks on the ImageNet dataset. Average is taken over images and choices of $(S, T)$.

K and $L_\infty$ norm on the ImageNet dataset. Combining the prior with other strategies on the PubFig dataset creates more efficient strategies. We verified that these conclusions have statistical significance, using the same Wilcoxon signed-rank test as Lin et al. used to compare success rates of attacks [37]. Combining the prior with other strategies on the ImageNet dataset has minimal effects on the efficiency. The most efficient strategy needs only 12.41–49.19% of the number of attempts needed by the baseline on PubFig, 0.79–4.53% on ImageNet with $L_2$-norm, and 1.26–9.40% on ImageNet with $L_\infty$-norm.

We additionally evaluated the strategies on ImageNet with two variants of the current setup, each with an additional constraint on $A$. In the first variant, we randomly selected a five-class subset of $S$ as managers, allowing the vault to be opened only if 1) at least $K$ staff members agree and 2) at least one of these $K$ staff members is a manager. We modified $A$ so it would still try to impersonate any of the staff members until $K-1$ success, and then only try to impersonate any of the managers if no manager has been impersonated. The relationship between attack strategies remains the same: previously more efficient strategies are still more efficient in this setting. The most efficient strategy needs only 0.89–3.41% of the number of attempts needed by the baseline with $L_2$-norm, and 2.81–9.40% with $L_\infty$-norm.

In the second variant, we required that the face-recognition system recognizes all $K$ staff members simultaneously and hence each burglar might impersonate once at most. This is in contrast with the previous two settings, where the face-recognition system takes one face at a time, and the same burglar may impersonate more than one staff members. The relationship between attack strategies still remains the same. The most efficient strategy needs only 0.95–3.55% of the number of attempts needed by the baseline with $L_2$-norm, and 2.99–6.74% with $L_\infty$-norm.

**Takeaways (Attack Strategies)**

Overall, across multiple setups, we observed that by applying the attack strategies we propose, adversaries can find more attacks per time unit and obtain larger $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$.

## IV. Enhancing Group-based Robustness

We previously showed that our new metric can reveal new insights about models' susceptibility to realistic threats (§II), as well as that formalizing these threats makes it possible to design faster or more successful attacks (§III). In this section we introduce defenses that focus on defending against group-based attacks and examine their performance relative to previous defenses. We summarize the desired properties of such defenses in §IV-A. We propose an approach to systematically build such defenses in §IV-B. We empirically verify that our defense achieves the desired properties in §IV-D, with the setup described in §IV-C.

### A. Defense Objectives

We desire the new defenses to have high benign accuracy on all inputs, and high $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, preventing the attacker to achieve any $I_i \in \mathcal{I}$. We also noticed that naïvely preventing the attacker from achieving their goal could also render the ML model useless in benign cases, when no attacker is present. For example, by never classifying *any* traffic sign as a stop sign, a classifier might achieve better group-based robustness because it might prevent unexpected vehicle stops caused by attacks. Such a classifier might also maintain high average accuracy because there are many traffic signs other than stop signs that could still be classified correctly. However, such a classifier might not be considered useful in practice.

Hence, we desire the new defenses to perform better than existing defenses on the following three metrics simultaneously: 1) group-based robustness $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, 2) average accuracy, and 3) accuracy on unperturbed inputs associated with classes that might be impersonated. When such classes are underrepresented, (2) might not naturally imply (3): (2) can still hold if there are many classes, of which only a few might be impersonated, and the classifier never emits them.

### B. Defense Approach

The state-of-the-art defense against evasion attacks is adversarial training, which involves rapidly and adaptively generating adversarial examples. The more time-efficient group-based attacks that we have developed (§III-A) make it possible to generate adversarial examples fast enough for adversarial training specifically against group-based attacks.

Existing adversarial training defenses train for better untargeted robustness, emphasizing that the model should always predict the *correct class* [40], [65]. However, to have high group-based robustness, a model does not have to be always correct on perturbed (i.e., adversarial) inputs. Hence, we propose focusing on training models to maintain accuracy on all inputs when no attacker is present, while also allowing them to misclassify inputs supplied by the attacker, as long as those

misclassifications do not further the attacker's objectives. For example, if an attacker, Eve, attempts to impersonate a member of the bank staff, there is no harm to the bank if the classifier is fooled into thinking that the attacker is *another* attacker, Mallory; the only harm is in predicting that Eve is a member of the bank staff.

We believe this approach will enable the achievement of higher group-based robustness while maintaining high benign accuracy. We modified established adversarial training algorithms to build stronger defenses against group-based attacks. Besides generating adversarial examples with group-based attacks, we also designed a new data-fetching mechanism and new loss function, detailed below.

*1) Loss Function:* Many existing adversarial training defenses use the same loss function for training as when no adversaries are present (e.g., [40], [65]). Intuitively, to have high group-based robustness, we only need to train the model to avoid predicting any $t \in T$ on adversarial examples, and thus we have

$$\ell_{\text{MDTRAIN}} = \kappa * \sum_{t \in T} ReLU(Z_t + \delta - \max_{i \notin T}(Z_i)) \quad (5)$$

where $t$ is iterated over all classes in $T$ and the largest $Z_i$ among all classes $i$ not in $T$ is used. $\kappa$ is a non-negative weighting factor.

Instinctively, a larger $\kappa$ implies higher group-based robustness and lower benign accuracy, and vice versa. We realized that the choice of $\kappa$ could depend on the specific needs of implementation scenarios. For example, if we expect the model to maintain very high benign accuracy while gaining some group-based robustness, a large $\kappa$ might be preferred. In our implementation, we ran linear search and chose a $\kappa$ value for each dataset such that our defense outperforms existing ones on all three metrics listed in §IV-A on a validation set.

$\ell_{\text{MDTRAIN}}$ is non-negative, and equals zero if and only if an attack has been prevented. If $\ell_{\text{MDTRAIN}} > 0$, there is at least one class $t \in T$ that has higher logits than all classes $i \notin T$; otherwise, if $\ell_{\text{MDTRAIN}} = 0$, there is at least one class $i \notin T$ that has higher logits than all classes $t \in T$.

*2) Data Fetching:* Existing adversarial training defenses use adversarial examples generated from inputs associated with all classes. To have high group-based robustness, we only need to train the model against evasion attacks that use input instances associated with some $s$ in $S$. As described in §IV-A, we also expect the model to maintain high benign accuracy. Thus, we modify the data-fetching process so every fetched batch consists of two data partitions: the first partition consists of inputs associated with all classes and the second partition consists of inputs associated with some $s$ in $S$. In our implementation, the ratio of sizes of partitions is close to the ratio of the inputs of these two types in the training set. The first partition is never perturbed, and we generate adversarial examples on the second partition using $\ell_{\text{MDMUL}}$ (§III-A). We train the model to correctly classify inputs in the first partition, by minimizing the cross-entropy loss which is commonly used for benign training. We train the model to prevent group-based attacks on the second partition by minimizing $\ell_{\text{MDTRAIN}}$.

### C. Experiment Setup

To verify that our defense boosts $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ at the same level of benign accuracy, we used the same threat model, datasets and benchmarks as we did in §II-E. In addition, we used the baselines and measurement process below.

*1) Baselines:* To illustrate its improvements to $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ at the same level of benign accuracy, we examined the defense on the GTSRB dataset at $L_\infty = 8/255$ with ResNet architecture, and also on the PubFig dataset. We reused the 100 adversarial training instances as baselines on GTSRB, and the VGG model trained by Wu et al.'s method [62] for five epochs as a baseline on PubFig (they were used in §II-E).

*2) Measurement Process:* As we did in §II-E, §III-A3, and §III-B4, we implemented $\mathcal{G}$ for each test $\mathcal{I}$ on different datasets. On the GTSRB dataset, we used the same setup as we did for attack loss functions, perturbing speed limit and delimits signs to (1) one of the signs that would lead to an immediate stop or (2) no more than half of the actual limit. $\mathcal{I} = \bigcup_{s \in S} \bigcup_{t \in T_s} \{\{(s,t)\}\}$. We trained an instance with free adversarial training for 50 epochs, and then trained it with the method we described in §IV-B for another 50 iterations. Other choices of numbers of iterations might also work. We compared the robustness of this instance with the robustness of the 100 baselines, on adversarial examples generated by the best guess method.

Again we used the PubFig dataset to simulate the burglary scenario where burglars are trying to attack a bank. Due to limited computation resources, we slightly modified the setup such that we still randomly selected two mutually exclusive sets of classes $S$ and $T$, but each of $S$ and $T$ consists of five classes. We randomly selected four different $S$ and $T$ pairs. $T$ is the set of staff members who have legal access, and $S$ is the set of burglars. As described in §I, the bank requires three *distinct* staff members to agree before a vault can be opened. The burglars need to impersonate three *distinct* staff members and each $I_i \in \mathcal{I}$ is a surjective function. We started with an instance that has been adversarially trained for five epochs using Wu et al.'s method [62], and we trained with our method (described in §IV-B) for one epoch. We also trained the VGG model using Wu et al.'s method for one more epoch (six epochs in total) to fairly compare with our defense. As described in §IV, our defenses are specific to $\mathcal{I}$, corresponding to choices of $S$ and $T$. We trained four instances according to the four choices. To measure $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, we exhaustively searched through all possible choices of $X$: we fixed $|X| = 5$, and $\mathcal{G}$ sampled one $x$ from each $s \in S$ uniformly. With respect to the $X$ and $T$, we exhaustively search through all possible choices of $I_i$ and verify if there is such an $I_i$ that the burglars can achieve, as we described in §II-C. It is worth noticing that our defense aims to prevent attacks from happening rather than make attacks slower (but only against the attack strategies mentioned in §III-B).

As we explained in §IV-A, we expect the model to also maintain high accuracy on classes that might be impersonated. We measured the benign accuracy on inputs associated with all possible $t$ in $T_s$. On the GTSRB dataset, we measured the accuracy on any inputs $x$ that are stop signs, no entry signs, no vehicle signs, or speed limits no higher than 60 KPH. On the

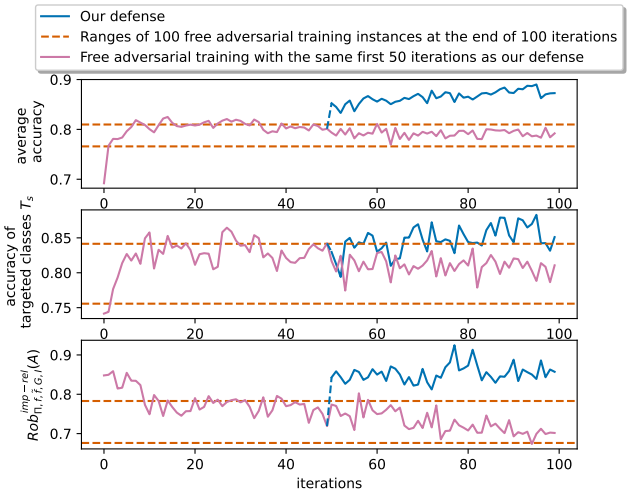


Fig. 16. Average accuracy, accuracy on targeted classes $T_s$, and $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of our defense and free adversarial training. As soon as we switched to our defense, average accuracy and $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ boosted while accuracy on $T_s$ remained about the same. As we trained for more iterations, the accuracy on $T_s$ and average accuracy kept increasing, while $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ remained at the same level.

Fig. 17. Peformance of our defense compared with its two baselines on the PubFig dataset. Each instance of our defense corresponds to a specific choice of $S$ and $T$. Our defenses achieve similar average accuracy and accuracy on $T$ to the baselines. Specifically, with $\#1$ choice of $T$, although our defense is up to 5% less accurate on inputs associated with $T$, we are only two data points worse because there are only 38 input instances associated with $T$ in the test set. Meanwhile, the $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ of our defense is much higher than the baselines. We measured $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ with two choices of $\mathcal{I}$ as we did in §III-B5. One choice of $\mathcal{I}$ allows the reuse of attackers, a.k.a. the same burglar *may* impersonate more than one staff members, whereas the other choice of $\mathcal{I}$ does not.

| *Metric* | *Choice of T* | *Wu et al.'s* | *Wu et al.'s +1 epoch* | *Our Defense #1* | *Our Defense #2* | *Our Defense #3* | *Our Defense #4* |
|---|---|---|---|---|---|---|---|
| Average Accuracy | - | 0.98 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
| Accuracy on $T$ | $\#1$ | 0.95 | 0.97 | 0.92 | - | - | - |
| | $\#2$ | 0.99 | 0.99 | - | 0.99 | - | - |
| | $\#3$ | 1.00 | 0.98 | - | - | 1.00 | - |
| | $\#4$ | 0.95 | 1.00 | - | - | - | 0.97 |
| Robustness *With* Reuse of Attackers | $\#1$ | 0.25 | 0.31 | 0.89 | - | - | - |
| | $\#2$ | 0.27 | 0.38 | - | 0.65 | - | - |
| | $\#3$ | 0.84 | 0.55 | - | - | 0.97 | - |
| | $\#4$ | 0.73 | 0.68 | - | - | - | 0.84 |
| Robustness *Without* Reuse of Attackers | $\#1$ | 0.38 | 0.37 | 0.99 | - | - | - |
| | $\#2$ | 0.72 | 0.73 | - | 0.91 | - | - |
| | $\#3$ | 0.91 | 0.88 | - | - | 0.99 | - |
| | $\#4$ | 0.95 | 0.98 | - | - | - | 1.00 |

PubFig dataset, we measured the accuracy on inputs associated with any $t$ in $T$ with respect to the specific choices of $T$.

### D. Results

Fig. 16 demonstrates the results on the GTSRB dataset. With one iteration of training, our defense achieved higher average accuracy, higher $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ and about the same accuracy on $T_s$ compared to the baselines. As the training process progressed, average accuracy and accuracy on $T_s$ increased, while $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ were always higher than that of the baselines. On the PubFig dataset, as shown in Fig. 17, with one iteration of training, our defenses achieve similar average accuracy and accuracy on $T$ to the baselines. Meanwhile, our defenses boost $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ by up to 3.52 times, relatively. On both datasets, our defense has no worse average accuracy and accuracy on $T$ than the baselines while obtaining higher $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$. We acknowledge that our tuning of parameters might not be the best: other tunings might achieve higher performance by some or all of the three metrics mentioned above. However, our experiments successfully demonstrated that we can systematically generate defenses to meet the goals described in §IV-A.

**Takeaways (Defense)**

By modifying existing adversarial training algorithms, we were able to generate defenses that outperform existing ones on all three metrics mentioned in §IV-A: 1) group-based robustness $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, 2) average accuracy, and 3) accuracy on unperturbed inputs associated with classes that might be impersonated.

## V. Related Work

Several instances of prior work were discussed in §II and §III, which motivated the group-based metric and attacks. Additionally, §II-E leverages prior work as benchmarks and baselines. This section complements the prior work already addressed by positioning group-based attacks within multiple domains, including role-based access control, natural perturbations, privacy, and fairness. We discuss each of these in turn.

*Role-based Access Controls:* Ferraiolo and Kuhn defined Role-based Access Controls as a mechanism to group users by roles and grant access accordingly [80]. In a hospital setting, for example, doctors have read and write access to prescriptions but pharmacists only have read access. Doctors and pharmacists are two groups of users, and doctors have more access. Schaad et al. described a real-world implementation of Role-based Access Controls in Dresdner Bank, a major European bank where 50,659 employees are grouped into about 1,300 roles [79]. In the setup of our experiments, we also have two groups of users: potential attackers and potential targets, e.g., students and instructors in the class materials theft scenario, and burglars and bank staff in the burglary scenario. In both scenarios, users from groups with less access attempt to impersonate members from the other groups to gain more access. The new metric we propose, group-based robustness, measures the true threat of such disguise more accurately than existing metrics. The new loss functions (§III-A) and strategies (§III-B) we propose help estimate this threat more efficiently than than naïve methods.

*Natural Perturbations:* Machine-learning models have been found to perform similarly [81] or better [82] than humans on image-classification tasks. However, when there is noise in the images, these models tend to perform much worse than humans [81]. For example, Mu et al. and Hendrycks et al. found that natural, non-adversarial, perturbations, such as blurring, can significantly harm the functionality of models [83],

[84]. Similarly, machine learning models' vulnerability to group-based misclassifications demonstrated in this work could still exist even without the presence of an adversary.

*Evasion Attacks as a Defense for Privacy:* The prevalence of machine learning raises privacy concerns, spurring efforts to protect private information and resist surveillance. Wenger et al. identified various cases where facial recognition is deployed [85] along with many anti-facial recognition tools, including evasion attacks (e.g., [50], [86]–[92]). Abdullah et al. summarized real-world use cases of voice recognition systems [93], as well as corresponding evasion attacks against these systems (e.g., [94]–[102]). As an introduction to a more general form of evasion based on groups (§II), our paper also provides a framework for evasion attacks that could similarly be used by individuals as countermeasures against unwanted surveillance or data collection (i.e., an individual from an oppressed or vulnerable minority group may want to prevent themselves from being automatically identified via facial recognition).

*Fairness in Machine Learning:* In our experiments, the group-based robustness of various models was significantly affected by different choices of $T$ and $S$ even when we used the same $|T|$ and $|S|$. We also observed that the success rate of perturbing images from $s \in S$ as $t \in T$ differed significantly based on choices of $s$ and $t$, which led us to suggest computing a prior probability of success for each $(s, t)$ to identify vulnerable $(x, t)$ pairs, with $x \in X$ (§III-B). Researchers have noticed a similar problem with accuracy and untargeted robustness: the performance of models varies with different class distribution [103], [104]. Previous work argues for fairness in machine learning, advocating that accuracy and untargeted robustness should be independent of classes. However, whether a counterpart of fair machine learning is feasible for targeted robustness or group-based robustness remains unknown.

## VI. Conclusion

In this paper, we identified a limitation in the previous evaluation process of defenses against evasion attacks: in some real-world attack scenarios, the performance of these models cannot be accurately measured by existing metrics. We formally defined a new metric, group-based robustness, to measure the true threat in these attack scenarios, and statistically verified that group-based robustness is negligibly or weakly correlated with every existing metric. We also proposed approaches that, while maintaining a close $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, boost the speed of attacks in these attack scenarios: two new loss functions, the MDMAX loss and the MDMUL loss, and three new attack strategies. We additionally innovated a defense that elevates group-based robustness $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ while maintaining high benign accuracy. We validated the improvement with experiments across datasets, defenses, distance metrics, and attack scenarios. Overall, we explored a new attack space where some real-world attacks reside but existing research works have not addressed.

## Acknowledgments

The work described in this paper was supported in part by NSF grants 1801391, 2112562, and 2113345; by DARPA under contract HR00112020006; by the National Security Agency under award H9823018D0008; by Len Blavatnik and the Blavatnik Family Foundation; by a Maof prize for outstanding young scientists; and by the Neubauer Family Foundation.

## References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *European Machine Learning and Data Mining Conference*, 2013.

[2] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "Robustbench: a standardized adversarial robustness benchmark," arXiv preprint 2010.09670, Jun. 2021.

[3] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "Mma training: Direct input space margin maximization through adversarial training," in *International Conference on Learning Representations*, 2020.

[4] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020.

[5] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *International Conference on Machine Learning*, 2019.

[6] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *International Conference on Learning Representations*, 2020.

[7] V. Sehwag, S. Wang, P. Mittal, and S. Jana, "Hydra: Pruning adversarially robust neural networks," in *Conference on Neural Information Processing Systems*, 2020.

[8] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi, "Unlabeled data improves adversarial robustness," in *Conference on Neural Information Processing Systems*, 2019.

[9] D. Wu, S. tao Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," in *Conference on Neural Information Processing Systems*, 2020.

[10] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust imagenet models transfer better?" in *Conference on Neural Information Processing Systems*, 2020.

[11] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[12] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*, 2019.

[13] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*, 2018.

[14] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, pp. 1–30, 2019.

[15] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[16] B. Li, S. Wang, S. Jana, and L. Carin, "Towards understanding fast adversarial training," *ArXiv*, vol. abs/2006.03089, 2020.

[17] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations*, 2018.

[18] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *International Conference on Learning Representations*, 2018.

[19] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. A. Mann, "Improving robustness using generated data," in *Conference on Neural Information Processing Systems*, 2021.

[20] Q. Kang, Y. Song, Q. Ding, and W. P. Tay, "Stable neural ode with lyapunov-stable equilibrium points for defending against adversarial attacks," in *Conference on Neural Information Processing Systems*, 2021.

[21] T. Pang, M. Lin, X. Yang, J. Zhu, and S. Yan, "Robustness and accuracy could be reconcilable by (proper) definition," in *International Conference on Machine Learning*, 2022.

[22] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal, "Robust learning meets generative models: Can proxy distributions improve adversarial robustness?" in *International Conference on Learning Representations*, 2022.

[23] H. Huang, Y. Wang, S. M. Erfani, Q. Gu, J. Bailey, and X. Ma, "Exploring architectural ingredients of adversarially robust deep neural networks," in *Conference on Neural Information Processing Systems*, 2021.

[24] K. Sridhar, O. Sokolsky, I. Lee, and J. Weimer, "Improving neural network robustness via persistency of excitation," in *American Control Conference*, 2022.

[25] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware instance-reweighted adversarial training," in *International Conference on Learning Representations*, 2021.

[26] M. Andriushchenko and N. Flammarion, "Understanding and improving fast adversarial training," in *Conference on Neural Information Processing Systems*, 2020.

[27] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "Understanding and improving fast adversarial training," in *Conference on Neural Information Processing Systems*, 2019.

[28] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang, "Adversarial robustness: From self-supervised pre-training to fine-tuning," in *Conference on Computer Vision and Pattern Recognition*, 2020.

[29] J. Cui, S. Liu, L. Wang, and J. Jia, "Learnable boundary guided adversarial training," in *International Conference on Computer Vision*, 2021.

[30] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*, 2019.

[31] L. Huang, C. Zhang, and H. Zhang, "Self-adaptive training: beyond empirical risk minimization," in *Conference on Neural Information Processing Systems*, 2020.

[32] L. Rice, E. Wong, and J. Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning*, 2020.

[33] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *International Conference on Machine Learning*, 2020.

[34] M. Augustin, A. Meinke, and M. Hein, "Adversarial robustness on in- and out-distribution improves explainability," in *European Conference on Computer Vision*, 2020.

[35] T.-W. Weng, H. Zhang, P.-Y. Chen, A. Lozano, C.-J. Hsieh, and L. Daniel, "On extensions of clever: A neural network robustness evaluation algorithm," in *IEEE Global Conference on Signal and Information Processing*, 2018.

[36] D. Gopinath, G. Katz, C. S. Pasareanu, and C. Barrett, "Deepsafe: A data-driven approach for assessing robustness of neural networks," in *Automated Technology for Verification and Analysis*, 2018.

[37] W. Lin, K. Lucas, L. Bauer, M. K. Reiter, and M. Sharif, "Constrained gradient descent: A powerful and principled evasion attack against neural networks," in *International Conference on Machine Learning*, 2022.

[38] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks*, 2013.

[39] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*, 2020.

[40] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[41] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[42] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations*, 2017.

[43] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[44] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2016.

[45] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2018.

[46] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy*, 2016.

[47] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, "On the limitation of convolutional neural networks in recognizing negative images," in *IEEE International Conference on Machine Learning and Applications*, 2017.

[48] F. Tramèr and D. Boneh, "Adversarial training and robustness for multiple perturbations," in *International Conference on Neural Information Processing Systems*, 2019.

[49] J. Uesato, B. O'Donoghue, A. van den Oord, and P. Kohli, "Adversarial risk and the dangers of evaluating against weak attacks," in *International Conference on Machine Learning*, 2018.

[50] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *ACM Conference on Computer and Communications Security*, 2016.

[51] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, vol. 1, 2017, pp. 39–57.

[52] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," in *International Conference on Learning Representations*, 2016.

[53] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *IEEE Symposium on Security and Privacy*, 2020.

[54] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018.

[55] H. Kwon, H. Yoon, and D. Choi, "Priority adversarial example in evasion attack on multiple deep neural networks," in *International Conference on Artificial Intelligence in Information and Communication*, 2019.

[56] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, "Multi-targeted adversarial example in evasion attack on deep neural network," in *IEEE Access*, 2018.

[57] S.-M. Moosavi-Dezfooli, A. Shrivastava, and O. Tuzel, "Divide, denoise, and defend against adversarial attacks," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[58] Y. Tashiro, Y. Song, and S. Ermon, "Diversity can be transferred: Output diversification for white-and black-box attacks," in *Conference on Neural Information Processing Systems*, 2020.

[59] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *USENIX Conference on Security Symposium*, 2022.

[60] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *International Conference on Computer Vision*, 2009.

[61] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "A general framework for adversarial examples with objectives," *ACM Trans. Priv. Secur.*, vol. 22, no. 3, pp. 16:1–16:30, Jun. 2019.

[62] T. Wu, L. Tong, and Y. Vorobeychik, "Defending against physically realizable attacks on image classification," in *International Conference on Learning Representations*, 2020.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE*

*conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[64] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

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

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2016.

[68] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv:1602.07360*, 2016.

[69] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *European Conference on Computer Vision*, 2018, pp. 116–131.

[70] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2018.

[71] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," in *IEEE Signal Processing Letters*, 2016.

[72] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2015.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[74] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust imagenet models transfer better?" in *Conference on Neural Information Processing Systems*, 2020.

[75] E. S. Jo, "Unso/roberta-large-finetuned-sst5," https://huggingface.co/Unso/roberta-large-finetuned-sst5, Accessed: 2023-09-21.

[76] "Sentiment analysis on sst-5 fine-grained classification," https://paperswithcode.com/sota/sentiment-analysis-on-sst-5-fine-grained, Accessed: 2023-09-21.

[77] L. Yuan, Y. Zhang, Y. Chen, and W. Wei, "Bridge the gap between cv and nlp! a gradient-based textual adversarial attack framework," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

[78] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: Appropriate use and interpretation," *Anesthesia & Analgesia*, vol. 126, no. 5, pp. 1763–1768, 2018.

[79] A. Schaad, J. Moffett, and J. Jacob, "The role-based access control system of a european bank: A case study and discussion," in *ACM symposium on Access control models and technologies*, 1992.

[80] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," in *National Computer Security Conference*, 1992.

[81] S. F. Dodge and L. Karam, "A study and comparison of human and deep learning recognition performance under visual distortions," in *International Conference on Computer Communication and Networks*, 2017.

[82] Y. C. Goh, X. Q. Cai, W. Theseira, G. Ko, and K. A. Khor, "Evaluating human versus machine learning performance in classifying research abstracts," in *Scientometrics*, vol. 125, 2020, p. pages1197–1212.

[83] D. H. andThomas Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations*, 2019.

[84] N. Mu and J. Gilmer, "Mnist-c: A robustness benchmark for computer vision," arXiv preprint 1906.02337, Jun. 2019.

[85] E. Wenger, S. Shan, H. Zheng, and B. Y. Zhao, "Sok: Anti-facial recognition technology," in *IEEE Symposium on Security and Privacy*, 2023.

[86] T. Cilloni, W. Wang, C. Walter, and C. Fleming, "Ulixes: Facial recognition privacy with adversarial machine learning," in *Privacy Enhancing Technologies Symposium*, 2022.

[87] V. Chandrasekaran, C. Gao, B. Tang, K. Fawaz, S. Jha, and S. Banerjee, "Face-off: Adversarial face obfuscation," in *Privacy Enhancing Technologies Symposium*, 2021.

[88] D.-L. Nguyen, S. S. Arora, Y. Wu, and H. Yang, "Adversarial light projection attacks on face recognition systems: A feasibility study," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2020.

[89] S. Komkov and A. Petiushko, "Advhat: Real-world adversarial attack on arcface face id system," in *International Conference on Pattern Recognition*, 2021.

[90] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu, "Efficient decision-based black-box adversarial attacks on face recognition," in *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2019.

[91] X. Yang, Y. Dong, T. Pang, H. Su, J. Zhu, Y. Chen, and H. Xue, "Towards face encryption by generating adversarial identity masks," in *International Conference on Computer Vision*, 2020.

[92] Z.-A. Zhu, Y.-Z. Lu, and C.-K. Chiang, "Generating adversarial examples by makeup attacks on face recognition," in *IEEE International Conference on Image Processing*, 2019.

[93] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems," in *IEEE Symposium on Security and Privacy*, 2021.

[94] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning*, 2019.

[95] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Network and Distributed Systems Security Symposium*, 2019.

[96] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," in *International Joint Conference on Artificial Intelligence*, 2019.

[97] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, "Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices," in *USENIX Conference on Security Symposium*, 2020.

[98] T. Sugawara, B. Cyra, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: Laser-based audio injection attacks on voice-controllable systems," in *USENIX Conference on Security Symposium*, 2020.

[99] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *USENIX Conference on Security Symposium*, 2016.

[100] H. Abdullah, M. S. Rahman, W. Garcia, L. Blue, K. Warren, A. S. Yadav, T. Shrimpton, and P. Traynor, "Hear "no evil", see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems," in *IEEE Symposium on Security and Privacy*, 2021.

[101] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *USENIX Conference on Security Symposium*, 2018.

[102] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *Network and Distributed Systems Security Symposium*, 2019.

[103] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *International Conference on Machine Learning*, 2021.

[104] V. Nanda, S. Dooley, S. Singla, S. Feizi, and J. P. Dickerson, "Fairness through robustness: Investigating robustness disparity in deep learning," in *ACM Conference on Fairness, Accountability, and Transparency*, 2021.

## VII. Artifact Appendix

On behalf of the authors, we are happy to publish the source code of our paper as an artifact. However, our experiments took several months on the GPUs, while we are told by the artifact chairs that artifact reviewers are not assumed to have GPUs and artifacts to be reviewed are expected to take no longer than 24 hours. Thus we 1) decide to apply for the "Functional" and "Available" badges but not the "Reproduced" badge and 2) provide a "hello-world" style mini-experiment that takes less than two hours on laptop CPUs but still verifies our claims.

### A. Description & Requirements

This section lists all the information necessary to recreate the experimental setup.

*1) How to access:* Our implementation is stored in this public GitHub repository: https://github.com/linweiran/GBR. The experiments we specifically designed for artifact evaluation can be found in the GTSRB repository, a.k.a. https://github.com/linweiran/GBR/tree/main/GTSRB. We have created a Zenodo version at https://zenodo.org/records/10104298, with the DOI: 10.5281/zenodo.10104297.

*2) Hardware dependencies:* The experiments we propose in the artifact would take less than two hours on laptop CPUs, so there is no other hardware dependency.

*3) Software dependencies:* We recommend using python3 of version 3.10.9 to run our scripts. Additionally, we require a list of python packages to run our code. The specified list of python packages can be found at https://github.com/linweiran/GBR/blob/main/GTSRB/requirements-cpu.txt. To install all packages, you may easily run "pip3 install -r requirements-cpu.txt" within the GTSRB folder.

*4) Benchmarks:* The dataset we used in the artifact is the GTSRB dataset. Please download the dataset from the downloads section of https://benchmark.ini.rub.de/gtsrb_dataset.html#Downloads where a link can be found. Specifically, please download files named "GTSRB_Final_Training_Images.zip", "GTSRB_Final_Test_Images.zip", and "GTSRB_Final_Test_GT.zip". After extracting these zip files, please move the directories named "Final_Test" and "Final_Training", along with the file "GT-final_test.csv" to the same directory. The path of this directory will be used as the only parameter (for both scripts mentioned below (§VII-B and §VII-C)

### B. Artifact Installation & Configuration

We preprocessed GTSRB images as the first step. You may switch to the GTSRB directory in our repo and run "python3 preprocess.py –data_path DATA_PATH" where "DATA_PATH" is the path to the directory where the extracted files are stored (described in §VII-A4). An example could be "python3 preprocess.py –data_path \data\GTSRB". The results will be printed out to the console.

### C. Experiment Workflow

We wrapped up all the mini-experiments as a single script named "hello_world.py" under the GTSRB directory. To reproduce experiments, run "python3 hello_world.py –data_path DATA_PATH" where "DATA_PATH" is the path to the directory where the extracted files are stored (described in §VII-A4 and §VII-B). An example could be "python3 hello_world.py –data_path \data\GTSRB".

### D. Major Claims

The major claims we made in the paper include:

- (C1): Group-based robustness $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ measures the robustness of models using different choices of $\mathcal{I}$ in accordance with the attack scenarios. Conventional metrics cannot measure the true threat in these sophisticated scenarios as accurately as group-based robustness does. Thus, we conclude that group-based robustness offers another meaningful assessment of model susceptibility to attacks in the real world compared to conventional metrics (§2).
- (C2): Attacks with the MDMAX loss or the MDMUL loss achieve comparable or slightly lower $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ than the best-guess attacks, consume markedly less time and are markedly more efficient, finding more attacks per time unit. Attacks with the MDMAX loss or the MDMUL loss consume the same amount of time as the average-guess attacks and have much higher $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$. The MDMUL loss and the MDMAX loss boost the efficiency of attacks in the attack scenarios we tried (§3.A).
- (C3): By applying the attack strategies we propose, adversaries can find more attacks per time unit and obtain larger $\mathbf{Adv}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$ (§3.B).
- (C4): By modifying existing adversarial training algorithms, we were able to generate defenses that outperform existing ones on all three metrics: 1) group-based robustness $\mathbf{Rob}_{\Pi,f,\tilde{f},\mathcal{G},\mathcal{I}}(A)$, 2) average accuracy, and 3) accuracy on unperturbed inputs associated with classes that might be impersonated (§4).

### E. Evaluation

This section documents the details of our experiments. As we documented at the beginning of this appendix, we understand that artifact reviewers have much more limited computation resources than we have. Thus we are not applying for the "Reproduced" badge and designed this scale-down set of mini-experiments. Specifically, we performed the following modifications to reduce experiments:

- We ran mini-experiments only on a specific combination of settings. The full-scale experiments used multiple datasets, distance limits, model architectures, and random seeds. The min-experiments used only the GTSRB dataset, $L_\infty$ distance at $8/255$, SqueezeNet architecture, and 0 as the random seed. We used such a combination as it is one of the least time-costly combinations.

- Mini-experiments ran all attacks by only one iteration. The full-scale experiments ran attacks with their default settings, ranging from 100 to 300 iterations.
- Mini-experiments trained models by as few iterations as possible (such that the models still converge). Full-scale experiments trained models by 100 to 300 iterations. while the mini-experiments only trained up to eight iterations.

We acknowledge that compared to full-scale experiments, the mini-experiments used weaker attacks and worse-performing models (e.g. less robust defenses). However, the major conclusions we made (§VII-D) still hold on the mini-experiments. It is worth noticing that the mini-experiments and full-scale experiments call exactly the same functions: only the parameters (specified above) sent to these functions differ. The overall time of our experiments shall take less than two hours on laptop CPUs.

*1) Experiment (E1): [Setup]* As described in the main paper, attackers are perturbing speed limit and delimit signs into signs that:

- require an immediate stop, including stop signs, no-entry signs, and no-vehicle signs; or
- display a limit much lower than the actual limit, such as no more than half of the actual limit.

*[How to]* We will document the steps required to prepare and configure the environment for this experiment, the steps to run this experiment, and the steps required to collect and interpret the results for this experiment in the following three blocks correspondingly.

*[Preparation]* First, install Python packages as we described in §VII-A3. We highly recommend the use of a virtual environment. Then download the dataset as we described in §VII-A4. Ultimately, preprocess the downloaded data as we described in §VII-B

*[Execution]* As we described in §VII-C, we wrapped up all the mini-experiments as a single script named "hello_world.py". You may run "python3 hello_world.py –data_path DATA_PATH".

*[Results]* Four numbers will be reported: benign accuracy, untargeted robustness, targeted robustness, and group-based robustness. Group-based robustness is different from all the other three. This supports C1.

*2) Experiment (E2): [Setup]* Same as the setup of E1.

*[How to]* Same as the [How to] of E1.

*[Preparation]* Same as the [Preparation] of E1.

*[Execution]* Same as the[Execution] of E1.

*[Results]* Five numbers will be reported, which correspond to the success rates of five attacks: attacks with the MDMAX loss, attacks with the MDMUL loss, the best guess attacks, the worst guess attacks, and the average guess attacks. The success rate of attacks with the MDMAX loss or attacks with the MDMUL loss is higher than that of the worst guess attacks or the average guess attacks. This supports C2.

*3) Experiment (E3): [Setup]* We did not evaluate strategies on the GTSRB dataset in the main paper. Here the setup is that attackers are perturbing speed limits no less than 70 (five classes) as speed limits no higher than 60 (four classes). Attackers sample one image from each of the five higher-speed classes, in total five images as a set. For each set of five images, attackers can only claim success if they can manipulate these images as all of the lower-speed four classes. They may manipulate the same image as different signs. We use the worst-performing strategy that we proposed (Estimate by Computing a Prior from a Validation Set).

*[How to]* Same as the [How to] of E1.

*[Preparation]* Same as the [Preparation] of E1.

*[Execution]* Same as the[Execution] of E1.

*[Results]* Two numbers will be reported, which are the number of attempts needed by attackers to find the same number of successful attacks, with or without using our strategies. Attackers using our strategies need fewer attempts. This supports C3.

*4) Experiment (E4): [Setup]* Same as the setup of E1 and E2.

*[How to]* Same as the [How to] of E1.

*[Preparation]* Same as the [Preparation] of E1.

*[Execution]* Same as the[Execution] of E1.

*[Results]* Three sets of numbers are reported. Each set consists of three numbers: benign accuracy, benign accuracy on the targeted set, and group-based robustness. The three sets correspond to an existing defense, adversarial training (7 training iterations), adversarial training with one more iteration of our defense training (8 iterations in total), and adversarial training with one more iteration (8 iterations of adversarial training). The model with our defense outperforms the other two models in all three metrics. This supports C4.