Your Attack Is Too DUMB: Formalizing Attacker Scenarios for Adversarial Transferability

Marco Alecci
SnT, University Of Luxembourg
Luxembourg, Luxembourg
marco.alecci@uni.lu

Mauro Conti
University of Padova
Padua, Italy
mauro.conti@unipd.it

Francesco Marchiori
University of Padova
Padua, Italy
francesco.marchiori.4@phd.unipd.it

Luca Martinelli
University of Padua
Padua, Italy
luca.martinelli.1@studenti.unipd.it

Luca Pajola*
University of Padua
Padua, Italy
luca.pajola@phd.unipd.it

ABSTRACT

Evasion attacks are a threat to machine learning models, where adversaries attempt to affect classifiers by injecting malicious samples. An alarming side-effect of evasion attacks is their ability to transfer among different models: this property is called *transferability*. Therefore, an attacker can produce adversarial samples on a custom model (surrogate) to conduct the attack on a victim's organization later. Although literature widely discusses how adversaries can transfer their attacks, their experimental settings are limited and far from reality. For instance, many experiments consider both attacker and defender sharing the same dataset, balance level (i.e., how the ground truth is distributed), and model architecture.

In this work, we propose the **DUMB** attacker model. This framework allows analyzing if evasion attacks fail to transfer when the training conditions of surrogate and victim models differ. DUMB considers the following conditions: **D**ataset so**U**rces, **M**odel architecture, and the **B**alance of the ground truth. We then propose a novel testbed to evaluate many state-of-the-art evasion attacks with DUMB; the testbed consists of three computer vision tasks with two distinct datasets each, four types of balance levels, and three model architectures. Our analysis, which generated 13K tests over 14 distinct attacks, led to numerous novel findings in the scope of transferable attacks with surrogate models. In particular, mismatches between attackers and victims in terms of dataset source, balance levels, and model architecture lead to non-negligible loss of attack performance.

CCS CONCEPTS

• **Security and privacy → Systems security**; • **Computing methodologies → Machine learning**.

KEYWORDS

Adversarial Machine Learning, Adversarial Attacks, Evasion Attacks, Transferability, Surrogate Model

ACM Reference Format:
Marco Alecci, Mauro Conti, Francesco Marchiori, Luca Martinelli, and Luca Pajola. 2023. Your Attack Is Too DUMB: Formalizing Attacker Scenarios for Adversarial Transferability. In *The 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23), October 16–18, 2023, Hong Kong, China.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3607199.3607227

1 INTRODUCTION

Evasion attacks consist in crafting a sample to produce a misclassification in a target Machine Learning (ML) model. With the integration of ML models in deployed real-life systems, the cybersecurity community increased its interest in studying how *attackers* can exploit ML vulnerabilities for some advantages. For instance, an attacker might try to make a hateful sentence look non-hateful [12] or botnets legitimate applications [3].

Although effective in theory, conducting evasion attacks in real-world scenarios is challenging since malicious actors cannot access target models' information (e.g., the gradient) [2]. Adversarial samples *transferability* is a possible solution investigated in prior works [24]: the attacker feeds the victim's model with adversarial samples computed by leveraging an own *surrogate model*.

While attempting to test the robustness of ML models of top IT companies (through their official APIs), we realized that not even transferable attacks are so simple. In particular, we found a non-negligible obstacle during our tests: *how should we train a surrogate model*? We needed a dataset to train a surrogate model, but we had no clue about the victim's dataset. Furthermore, suppose we were willing to produce a new dataset (or use an external one) for an inherently imbalanced task (e.g., a few samples of a botnet and thousands of benign samples): is the ground truth distribution matching the victim's one? And finally, what is the victim's ML architecture? Settings that differ from the victim might negatively impact the attack's success.

Contributions. Attackers, therefore, live in a state of "uncertainty" when training a surrogate model. Current literature fails to consider such scenarios, resulting in a lack of understanding of the real effect of state-of-the-art attacks. This work fills such a gap by proposing

[*] Corresponding author.

RAID '23, October 16–18, 2023, Hong Kong, China

ACM ISBN 979-8-4007-0765-0/23/10.
https://doi.org/10.1145/3607199.3607227

the **DUMB** attacker model, a framework that allows analyzing if evasion attacks fail to transfer when the training conditions of surrogate and victim models differ. In particular, DUMB faces the following conditions: **D**ataset so**U**rces, **M**odel architecture, and the **B**alance of the ground truth.

We then propose a novel testbed to analyze the evasion attacks' transferability with DUMB. The testbed consists of three distinct computer vision binary tasks, two sources that generate such datasets, four ground truth balancing levels (from balanced to highly imbalanced), and three models architecture. With this testbed, we analyzed the transferability of seven popular state-of-the-art attacks and six simple image transformations and generated 13K tests. Such extensive analyses allowed us to unveil new aspects of the transferability of evasion attacks and, furthermore, confirmed the importance of considering the three dimensions introduced with the DUMB attacker model.

Our contributions can be summarized as follows:

- We propose the **DUMB** attacker model, a novel evaluation system to measure evasion transferability.
- We propose a novel testbed to evaluate evasion transferability with the DUMB attacker model. The testbed comprises three distinct computer-vision tasks, four distinct balance levels of the classes, and three distinct state-of-the-art models.
- An extensive evaluation of state-of-the-art evasion attacks with the DUMB attacker model.

Findings. After evaluating many evasion attacks on all possible combinations of dataset source, model architecture, and class balance of the datasets, our findings can be summarized as follows:

(1) Less robust models are more susceptible to adversarial perturbations than highly performing models.
(2) Adversarial attacks in literature face difficulty transferring across architectures.
(3) Simple image obfuscation is an effective offensive strategy.
(4) Adversarial attacks struggle when transferring.
(5) Not all basic surrogate models are ideal for evading attacks.
(6) The discrepancy in class distributions between surrogate and victim datasets can greatly hinder the effectiveness of evasion attacks. Additionally, targeting the minority class seems to be easier than targeting the majority.
(7) Creating surrogate data can negatively impact the effectiveness of transferable attacks.

Our testbed and experiments are open-source and available at the following link: https://github.com/Mhackiori/DUMB.

Organization. This paper is organized as follows. Section 2 summarizes the literature on adversarial machine learning and transferable attacks. Section 3 introduces the DUMB attacker model. Section 4 describes the experimental settings. Sections 5 and 6 present the results and conclusions of our work, respectively.

2 PRELIMINARIES

Adversarial Machine Learning. Adversarial machine learning (AML) is the discipline that studies how adversaries can exploit machine learning (ML) algorithms to conduct an attack. Adversarial attacks can be classified with the following properties [4]: the *influence*, where attackers can actively affect the training procedure (causative attacks), or they simply do not alter the victims' models (exploratory attacks); the *security violation*, where attackers might attempts to alter victims' model's performance (integrity violation), to make victims' model unavailable (availability violation), or to obtain sensitive information (privacy violation); last, the *specificity* of the attack, if the attack targets a specific set of samples (targeted attack) or generic samples (untargeted attacks). The definition of an attack is further defined by the attackers' knowledge of the victims' system (e.g., training data, model architecture). In particular, we refer to *white-box* attacks when the attacker has (nearly) perfect knowledge about the victim's system, setting the worst-case scenario; on the opposite, we refer to *black-box* attacks when attackers know a little about the target.

Evasion Attacks. This work focuses on *evasion attacks*, where attackers aim to modify an input sample to produce a misclassification in the victim's model. Malicious samples x^* can be defined as $x^* = x + r$, where x is the original sample, and r is the perturbation. The perturbation r can be obtained through the following optimization process:

$$r = \arg\min_{z} f(x+z) \neq f(x). \tag{1}$$

Here, z is the variable being optimized, which represents the perturbation that is added to the original input x to create the perturbed input $x+z$. Many ML algorithms do not guarantee that the optimization is linear or convex, so we cannot always find a closed-form solution. Prior works propose different approaches to estimate such a perturbation; for instance, the Fast Gradient Signed Method (FGSM) [11]:

$$x^* = x + \varepsilon \cdot sign(\nabla_x J(\theta, x, y)), \tag{2}$$

where ε is small to ensure an "imperceptible" perturbation, J is a loss function (e.g., cross-entropy), θ the parameters of the model f, and y the ground truth for the given input x.

Transferable Attacks. A fascinating aspect of adversarial samples is their ability to potentially fool not only the model f used to find the perturbation r for a given sample x but also unknown models f'. This behavior has a strong repercussion in cyber-security: attackers can therefore leverage their own model f (named substitute or surrogate model) to produce adversarial samples for the victims model. Using a substitute model to generate an attack presents many advantages, such as white-box access. Papernot et al. [24] defined two distinct transferability scenarios by considering the surrogate and victim models. They referred to *intra-techniques transferability* when the two models share the same architecture (e.g., both logistic regression or both Deep Neural Network), or, vice-versa, to *cross-techniques transferability* when the two models have distinct architecture (e.g., one is a logistic regression and the other a Deep Neural Network).

Adversarial Attacks in Practice. The literature primarily covers theoretical aspects of threats in machine learning systems. Little is known about attacks in practice, where challenges that occur only in real-life might not be considered in controlled environments. Therefore, real-life attacks might be utterly different from what is discussed in the literature [2, 33]. Consequently, industries might perceive as "innocuous" threats that are considered technically attractive by the research community and "serious" those that are not. For instance, consider Perspective, a toxicity detection model deployed by Google: in their recent report [18], the developers tested their model against a simple NLP attack introduced by Gröndahl et al. [12] that can be deployed by many end-users rather than more complex - and perhaps unrealistic - attacks studied in the literature.

A few noticeable works proved the feasibility of attacking deployed ML applications: "All You Need Is Love", where simple textual perturbations (e.g., typos) endangered toxicity detectors [12]; "stealthy porn", where researchers showed that social network users evaded porn detectors by applying simple image filters [33]; attacks on deployment libraries, where attackers can exploit vulnerabilities of the libraries utilized to deploy a machine learning model [32]; "camouflage attack", a threat that exploits image-scaling algorithms to produce evasion in computer-vision applications [31]; "Zero-Width Space attack", where invisible Unicode characters inserted in textual samples disrupted the textual representations of many NLP services deployed by top IT companies [23]; "Captcha attack", where researchers showed potential adversarial samples utilized by Instagram users that endanger the OCR of automatic content moderators [6].

Challenges of Transferable Attacks. Practical constraints might affect the transferability of the attacks as well. We now summarized relevant prior works that attempted to study different variables that might impact the attacks' transferability. Generally, such works are guided by a common observation: it is unrealistic that attackers have knowledge of the victims' systems (e.g., dataset, model architecture), limiting the adoption of surrogate models. For instance, training a surrogate model might be expensive (or even impossible) for an attacker since it requires possessing valid training data. We identified two types of solutions in the literature that relax the constraint of having valid data: (i) cross-domain perturbations, i.e. perturbations computed on a task (e.g, paintings, cartoons, or medical images) that transfer on models trained on a distinct task (e.g, ImageNet classes) [22]; (ii) data-free attacks, where the substitute model can be learned thanks to the cooperation between a generative model, a discriminator, and a series of queries to the victim's model [34].

Nevertheless, many works analyzed the impact of surrogates on transferable attacks. Mao et al. [20], instead, discuss the problem of transferring attacks among computer-vision Machine-Learning-as-a-Service (MLaaS) and analyze how different models' properties might impact the attack. For instance, the authors found that simple surrogates do not necessarily improve transferability and that there is no dominant architecture for surrogates. Suciu et al. [27] proposed FAIL attacker model, where the authors investigated the impact of evasion transferability under different types of knowledge of victims' systems: the feature space, the architecture of the model, the label instances, and the leverage (i.e., constraints on the type of modification at the feature space).

Compared to the previous works, with **DUMB**, we attempt to cover unique aspects of the surrogate training, and in particular **D**ataset so**U**rces, **M**odel architecture, and the **B**alance of the ground truth. In particular, while aspects like the impact of the model architecture have been covered in literature, others were not, like the source of data and the imbalance problem. Therefore, analyses combining these three aspects are, per se, novel, and they can unveil unique patterns of adversarial transferability.

3 THE DUMB ATTACKER MODEL

Suppose being in the shoes of an attacker aiming to evade a victim organization f'. What are the steps necessary to conduct a (potentially) successful attack? Current literature studies the effect of transferability on settings far from being real [13]. Consider the adversary pipeline necessary to generate an adversarial sample; it consists of: (i) finding a suitable dataset that matches the victim's, (ii) choosing a surrogate model f, and picking a methodology that produces adversarial attacks. When designing such a pipeline, we find the following challenges that *might* affect the attack execution.

The dataset choice. Prior works mainly use a dataset shared among attackers and victims. *This is unrealistic.* Building a proper surrogate dataset is all but trivial since attackers and victims might follow different corpus generation strategies. For instance, in the hate speech detection task, Gröndahl et al. [12] show that prior works tackling hate speech propose many datasets following distinct generation procedures; as a result, models trained on a specific dataset lack in terms of generalization on distinct ones. Therefore, in such cases, the transferability might be a property not fully guaranteed.

Ground truth distribution. Prior works mainly assume that attackers and victims use datasets originating from the same source and, therefore, the distributions of the ground-truth match. This is a hard constraint in real settings since such distributions might differ for many reasons. First, the two distributions might result from two distinct methodologies to produce the datasets (see *the choice of the dataset*). Second, many preprocessing techniques might be used to augment the training data. This scenario is likely especially when the task is inherently imbalanced (e.g., hate speech detection[1]). Augmentation techniques can over-sample the minority class (e.g., SMOTE [5], Generative Adversarial Networks [10]) or undersample the majority one.

Model selection. Prior works consider this scenario when analyzing the transferability of distinct adversarial attacks. Indeed, attackers and victims might use one of the many state-of-the-art models or custom ones. For instance, only in computer vision, someone might choose among several models to fine-tune, such as VGG [26] (and its many versions like VGG16 and VGG19) and ResNet [14] (e.g., ResNet18, ResNet50).

Considering such challenges, we can clearly see a need to enhance the study of adversarial transferability in many distinct scenarios and not limit empirical evaluations to a few artificial settings. Thus, experiments focusing on white-box (full access to the victims' model) and black-box (little known about the victims' model) might not be representative of the many shades that might occur in real-life. We address such a gap by proposing the **DUMB** attacker model for transferable samples that present many attack scenario cases. **DUMB** considers **D**ataset so**U**rces, **M**odel architecture, and the **B**alance of the ground truth, potential factors that might affect the

[1] In the hate-speech detection, usually datasets are strongly imbalanced toward the hateful class [12].

transferability of the attacks. In Table 1, we present eight distinct variations of attacks that can occur in a black-box attack, and in particular, potential mismatches between the source (or surrogate) and target (or victim) models. Subscript a and v stand for attacker and victim, respectively. We highlight that, in real-life conditions, attackers do not know a priori in which attack scenario they are – except for the white-box case.

4 METHODOLOGY (THE DUMB TESTBED)

To simulate the eight specific cases presented in the DUMB table (Table 1), we design a testbed that considers distinct sources of datasets, different balance levels, and different model architectures. In this section, we describe our experimental setup, starting from the data collection phase (Section 4.1), the definition of the balance levels (Section 4.2), and the choice of the models (Section 4.3). Finally, we describe the attacks that we use and their implementation (Section 4.4) and our testing methodology (Section 4.5). Our GitHub repository contains the code and datasets to reproduce our experiments.

4.1 Dataset Sources (DU-dimension)

In this work, we focused on the transferability of binary classifiers, which is a common setting in many cybersecurity applications (e.g., spam/non-spam, phishing/non-phishing, hate/non-hate speech). We focus on computer-vision tasks since most adversarial attacks literature covers this domain. We defined three distinct tasks: Bikes&Motorbikes, Cats&Dogs, and Men&Women. Given the specific requirements of our testbed, the datasets for each task have been manually collected and validated according to the following steps.

(1) *Data Collection* – We generate two distinct datasets for each binary task by manually collecting images from two popular search engines: Bing and Google. By creating our own dataset instead of using open-source ones, we can ensure their integrity and have more control over the complexity of the task and possible biases. We collect an average of 14264 images for each dataset.
(2) *Duplicate Removal* – Duplicated images in each dataset are discarded with the *difPy*[2] library. After this procedure, an average of 254 images are removed from each dataset.
(3) *Manual Check* – Through manual inspection, we ensure that the datasets do not contain erroneous samples (e.g., not coherent with the classes, paintings, sketches, or low quality). Although this procedure might reduce any bias of having different data validation strategies between attackers and victims, it allows us to reveal the true effect of having distinct sources that generate (theoretically) the same type of data. On average, we remove 1854 images from each dataset.
(4) *Image Selection* – We randomly selected, for each dataset, 10000 samples equally split among the two classes.
(5) *Image Resizing* – Using Python Imaging Library (PIL)[3], each image is resized to 300 × 300 and converted to RGB. For the resizing process, we used the `antialias` option provided by PIL to prevent aliasing artifacts.

For each class in each dataset, we split those 5000 samples into training, validation, and test sets with respective ratios of 70%, 10%, and 20% (i.e., 3500 samples for the training set, 500 samples for the validation set, and 1000 samples for the test set). The images contained in the test set will be used not only to first evaluate the models but also to generate the adversarial samples.

4.2 Ground Truth Balancing (B-dimension)

A second (potentially) critical variable is the different class balance levels between attacker and defender. We simulate different balancing levels in the training sets with the following ratios:

- *Balanced* – 50% minority class, 50% majority class.
- *Weak Imbalance* – 40% minority class, 60% majority class.
- *Medium Imbalance* – 30% minority class, 70% majority class.
- *Strong Imbalance* – 20% minority class, 80% majority class.

For all our tasks, we choose the first class to be the minority class (i.e., Cats, Men, and Bikes), and this choice will be uniform for all balance levels. The number of class samples for each level of ground truth balancing is shown in Table 2. To achieve this, we fix the number of samples for the majority class and randomly undersample the minority class accordingly. For instance, to obtain a *strong imbalance* for the Cats&Dogs task, we keep all the 3500 images of Dogs and randomly select only 875 images of Cats. The validation set and the test set are unaffected by this procedure and contain an equal number of samples between the two classes.

4.3 Model Architectures (M-dimension)

We utilize three state-of-the-art computer vision models for fine-tuning tasks: AlexNet [16], ResNet [14] (ResNet18 version), and VGG [26] (VGG11-bn version). The training procedure follows what is described in the official PyTorch documentation.[4] We train a total of 3 tasks × 2 sources × 4 class distribution levels × 3 architectures = 72 models. A graphical overview of the training combinations is shown in Figure 1.

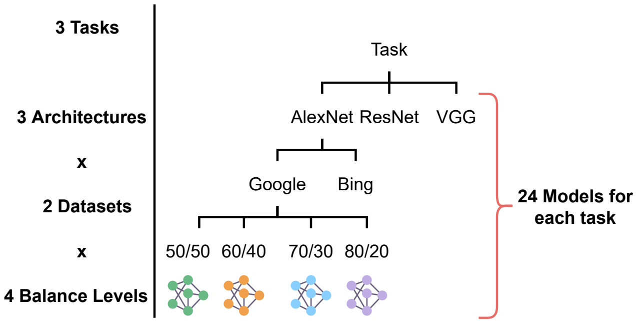


Figure 1: Model combinations during the training phase.

After training the models on each dataset, we evaluate their baseline performance on the test set. As a metric of evaluation, we will use the *F1 score*, which is defined as the harmonic mean of the precision and recall. This metric provides a balanced measure

[2] https://github.com/elisemercury/Duplicate-Image-Finder
[3] https://pillow.readthedocs.io/en/stable/
[4] https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html

Table 1: DUMB attacker model of adversarial transferable samples. DU **= Dataset soUrce,** M **= Model architecture,** B **= balance level.**

Case	Condition	Attack Scenario
C1	$DU_a = DU_v$ $M_a = M_v$ $B_a = B_v$	The ideal case for an attacker. We identified two potential attack scenarios. (i) Attackers legally or illegally gain information about the victims' system. (ii) Attackers and victims use the state-of-the-art.
C2	$DU_a = DU_v$ $M_a = M_v$ $B_a \neq B_v$	Attackers and victims use state-of-the-art datasets and model architecture. However, victims modify the class balance to boost the model's performance. This scenario can occur especially with imbalanced datasets.
C3	$DU_a = DU_v$ $M_a \neq M_v$ $B_a = B_v$	Attackers and victims use standard datasets to train their models. However, there is a mismatch in the model architecture. This scenario might occur when state-of-the-art presents many comparable models. Or similarly, the victims choose a specific model based on computational constraints.
C4	$DU_a = DU_v$ $M_a \neq M_v$ $B_a \neq B_v$	Attackers and victims use standard datasets to train their models, while models' architectures differ. Furthermore, victims adopt data augmentation or preprocessing techniques that alter the ground truth distribution (balancing). This scenario can occur especially with imbalanced datasets.
C5	$DU_a \neq DU_v$ $M_a = M_v$ $B_a = B_v$	Attackers and victims use different datasets to accomplish the same classification task. The ground truth distribution can be equal, especially in inherently balanced tasks. Similarly, models can be equal if they both adopt the state-of-the-art.
C6	$DU_a \neq DU_v$ $M_a = M_v$ $B_a \neq B_v$	Attackers and victims use different datasets to accomplish the same classification task. Datasets have different balancing because they are inherently generated in different ways (e.g., see hate speech datasets example) or because the attackers or victims augmented them. Attackers and victims use the same state-of-the-art architecture.
C7	$DU_a \neq DU_v$ $M_a \neq M_v$ $B_a = B_v$	Attackers and victims use different datasets to accomplish the same classification task. Datasets ground truth distribution matches. Attackers and victims use different models' architecture.
C8	$DU_a \neq DU_v$ $M_a \neq M_v$ $B_a \neq B_v$	The worst-case scenario for an attacker. Attackers do not match the victims' dataset, balancing, and model architecture.

For simplicity, C1 corresponds to the white-box setting, where attackers can access the victims' model, including gradients.

Table 2: Number of samples in different levels of imbalance of the training dataset.

Balance Level	Minority Class	Majority Class
Balanced	3500	3500
Weak Imbalance	2334	3500
Medium Imbalance	1500	3500
Strong Imbalance	875	3500

that considers both aspects of model performance, which is relevant in scenarios with possibly unbalanced dataset distributions. In particular, the F1 score is expressed as follows:

$$F1 = 2\frac{precision \cdot recall}{precision + recall}. \tag{3}$$

In Table 3a and 3b, we show the average performance of our models at the varying of task, architecture, and class balance levels for models trained on Bing and Google, respectively. All models are able to achieve good results on all balancing levels, but some differences can be noticed between the different tasks. Indeed, Men&Women appear to be the most complex task for any model, while Bikes&Motorbikes seem to be the easiest among the three.

Table 3: Baseline evaluation of the models. A = AlexNet, R = ResNet, V = VGG.

(a) Models trained on Bing.

	Bikes&Motorbikes			Cats&Dogs			Men&Women		
	A	R	V	A	R	V	A	R	V
20/80	0.99	0.99	0.99	0.93	0.97	0.97	0.85	0.92	0.92
30/70	0.98	0.99	0.99	0.94	0.97	0.97	0.87	0.93	0.93
40/60	0.99	0.99	0.99	0.95	0.98	0.98	0.89	0.94	0.94
50/50	0.99	0.99	0.99	0.95	0.98	0.98	0.90	0.95	0.95

(b) Models trained on Google.

	Bikes&Motorbikes			Cats&Dogs			Men&Women		
	A	R	V	A	R	V	A	R	V
20/80	0.96	0.96	0.97	0.95	0.97	0.98	0.82	0.89	0.88
30/70	0.97	0.98	0.98	0.96	0.98	0.98	0.86	0.91	0.92
40/60	0.97	0.98	0.98	0.96	0.98	0.99	0.86	0.92	0.93
50/50	0.97	0.99	0.98	0.96	0.98	0.99	0.86	0.93	0.93

4.4 Attacks

We consider two distinct attack families: *mathematical*, if the result of an optimization process (e.g., FGSM), and *non-mathematical*, if the result of a transformation that does not take into account any machine learning model (e.g., blurring).

Mathematical Attacks. For the mathematical attacks, we use the following popular attacks.

- *BIM* – Basic Iterative Method adversarial attack, as proposed by Kurakin et al. in their paper [17], is a method for generating adversarial examples for image classifiers. The attack works by iteratively perturbing the input image and using gradient descent to optimize the perturbation such that it causes the image classifier to produce the wrong output. One of the key features of the BIM attack is that it can be used to generate adversarial examples that are robust to various types of transformations, such as scaling and rotation.
- *DeepFool* – Moosavi-Dezfooli et al. proposed an algorithm to compute a minimal norm adversarial perturbation for a given image in an iterative manner [21]. At each iteration, the algorithm adds some perturbation that is computed to take the image to the edge of the region confined by the decision boundaries of the classifier; after that, the perturbations are accumulated to compute the final perturbation, which it is shown to be smaller than the one computed by FGSM in terms of their norm.
- *FGSM* – Fast Gradient Sign Method is one first and simplest adversarial attacks, first proposed by Goodfellow in a paper from 2014 [11]. It works by computing the gradient of the loss of the prediction made by a model based on the true class label of an image and using its sign to construct the adversarial image.
- *PGD* – Madry et al. proposed the Projected Gradient Descent [19]: an adversarial attack in which an attacker perturbs the input to a machine learning model in such a way as to cause the model to produce the wrong output. The attack works by iteratively calculating the gradient of the loss function with respect to the input and then using this gradient to update the input in the direction that will most likely cause the model to produce the wrong output.
- *RFGSM* – Tramèr et al. proposed an upgraded version of the FGSM attack called Random Fast Gradient Sign Method [28]. The most significant difference is that the FGSM attack generates the perturbation simultaneously, while the RFGSM attack generates the perturbation in a series of "random" steps. This makes the RFGSM attack more computationally efficient, as it can often find an adversarial example faster than the FGSM attack.
- *Square* – Andriushchenko et al. [1] proposed a new black-box attack called Square attack that does not rely on local gradient. It is a score-based attack, meaning that, while not having access to the target model, it can query the probability distribution over the classes predicted by the classifier.
- *TIFGSM* – The paper by Dong et al. [9] proposed a new method for generating adversarial examples, the Translation-Invariant Fast Gradient Sign Method (TI-FGSM), which aims to evade defenses that are based on input transformations by adding a translation-invariant constraint to the iterative FGSM algorithm. The key aspect of the paper is that it achieves high transferability of adversarial examples across different models by making the adversarial perturbations translation-invariant.

All mathematical attacks are implemented with *Torchattacks* [15], a popular python library used in the community [29, 30].

Non-mathematical Attacks. The other type of attacks we consider is *non-mathematical* attacks. These kinds of attacks do not require any gradient computation and are independent of the model or the task considered. Indeed, non-mathematical attacks have been shown to be effective in real-life ML applications [33]. We implemented these attacks using the PIL library since only simple image processing is required. More in detail, we implemented the following transformations:

- *Box Blur* – By applying this filter, it is possible to blur the image by setting each pixel to the average value of the pixels in a square box extending radius pixels in each direction. It is possible to specify a radius of arbitrary size.
- *Gaussian Noise* – A statistical noise having a probability density function equal to normal distribution. It is possible to specify a σ value.
- *Grayscale Filter* – To get a grayscale image, the color information from each RGB channel is removed, leaving only the luminance values. Grayscale images contain only shades of gray and no color because maximum luminance is white and zero luminance is black, so everything in between is a shade of gray.
- *Invert Color* – An image negative is produced by subtracting each pixel from the maximum intensity value, so for color images, colors are replaced by their complementary colors.
- *Random Black Box* – We draw a black square in a random position inside the central portion of the image to cover some crucial information. It is possible to define a size for the black square.
- *Salt and Pepper* – An image can be altered by modifying a certain amount of the pixels in the image either black or

white. The effect is similar to sprinkling white and black dots (salt and pepper) in the image. It is possible to specify the proportion of salt and pepper noise.

Parameters tuning. All the considered attacks need parameters that regulate the intensity of the perturbations. For instance, all the mathematical attacks have the parameter ϵ, except for DeepFool, which is regulated by the "overshoot" parameter. Similarly, some non-mathematical attacks have a parameter as well: radius for Box Blur, σ for the Gaussian Noise, the size of a black square for the Random Black Box, and the proportion of salt and pepper noise for Salt and Pepper. In general, we identified optimal parameters γ through the following optimization procedure:

$$\begin{aligned} \gamma &= \arg\max_{s} \frac{1}{n} \sum_{i=1}^{n} f(x_i) \neq f(x_i^*), \\ \text{subject to } & \frac{1}{n} \sum_{i=1}^{n} SSIM(x_i, x_i^*) \geq \alpha. \end{aligned} \tag{4}$$

In the notation, f is the model owned by the attacker and used during the optimization process, x^* is the adversarial samples derived by $\mathcal{A}(f, x; s)$, and $\mathcal{A}$ is the adversarial procedure with parameter s. The reader might notice that the first part of the equation is nothing more than the Attack Success Rate (ASR), where the higher, the more samples evaded. The optimization is constrained by the SSIM (Structural Similarity Index Measure), a measure that, given two images, computes their similarity. α is the minimum degradation threshold we accept by the perturbations. In our experiments, we set $\alpha = 0.4$ for all types of attacks. For all mathematical attacks except for DeepFool (i.e., attacks using ϵ as a parameter), we test ϵ values in the range $[0.01, 0.3]$ with a step of 0.01, while for DeepFool overshoot was tested in the range $[10, 100]$ with a step of 1. For non-mathematical attacks with a parameter, ranges and steps were determined individually and based on performance and perturbation. More details on the ranges for the attack parameters can be found in the attack generation script in our repository.

4.5 Testing Methodology

In Section 4.4, we presented a total of 13 attacks, comprising 7 mathematical and 6 non-mathematical attacks. After the searching phase for the optimal attacks' configuration (see Equation 4), we generate sets of adversarial samples containing 300 instances equally distributed among the classes. The images are randomly selected from the corresponding test set, which, however, is filtered in order to consider only images that the model correctly classified. In this way, we ensure that any misclassified adversarial sample can count in the Attack Success Rate. In the remaining part of the section, we discuss our testing methodology for the adversarial samples against our models separately for mathematical and non-mathematical attacks, as these attacks rely on different approaches.

Mathematical Attacks. Generating adversarial samples for mathematical attacks such as the FGSM requires an input model to compute and generate a perturbed image. Figure 2 shows an overview of the process. For each of the seven attacks we want to test, we need to evaluate all possible combinations of the following pairs (M_{src}, M_{trg}):

- M_{src} – The model used to generate the adversarial sample. The source model is the surrogate in the transferability setting.
- M_{trg} – The model against which the adversarial sample was tested. The target model is the victim's model in the transferability setting.

As explained in Section 4.3, we trained 24 models for each task and used each of them as the M_{src} to generate a set of adversarial samples. We tested each set against 24 different M_{trg}, resulting in $24^2 \times 3$ tasks = 1728 observations for each attack. Since we need to perform this evaluation for each of the seven mathematical attacks, we obtain a total of 1728 × 7 = 12096 observations.

Non-mathematical Attacks. Non-mathematical attacks, instead, are generated differently since they are transformations applied to the test set of a dataset and do not rely on any model. Thus, for each non-mathematical attack, we generate a total of 2 sets of samples (i.e., the datasets), and we test them on each model, obtaining a total of 2 × 24 × 3 tasks = 144. This is valid for each of the 6 non-mathematical attacks we consider, obtaining 144 × 6 = 864.

Therefore, the total number of observations performed in our study is 12096 + 864 = 12960.

5 RESULTS

In this section, we will discuss the evaluation results carried out with our experimental setup. Given the number of variables that potentially affect our results, we first evaluate the performance of state-of-the-art evasion attacks in the scenarios detailed by the DUMB attacker model (Section 5.1). We then evaluate individually the impact of the model (Section 5.2), class distribution (Section 5.3), and dataset source (Section 5.4). All the raw files from which our results are obtained can be found in the `results` folder in our repository.

5.1 DUMB Evaluation

In this section, we assess how adversarial attacks perform in the eight distinct cases of our proposed DUMB attacker model. We start by analyzing the results of the mathematical attacks, shown in Figure 3. In that Figure, we can observe the effect of two main variables: the task and the attacks. Note that all the performances are averaged among the three DUMB dimensions.

Task. The first outcome of the analysis highlights how transferability highly varies at the varying of tasks. For instance, attacks poorly transfer in Bikes&Motorbikes, while they are effective in the Men&Women task. A possible explanation can be linked with models' performance (reported in Table 3), where the attack poorly transfers when models greatly solve the task: in the Bikes&Motorbikes, indeed, models almost perfectly distinguish the two classes, while, on the opposite, on Men&Women they struggle. The outcome suggests malicious actors might easily transfer attacks on models with performances that are far from perfect. This finding is concerning if we consider that many real-life tasks are challenging, and state-of-the-art performance is even much below 0.90 of the F1-score.

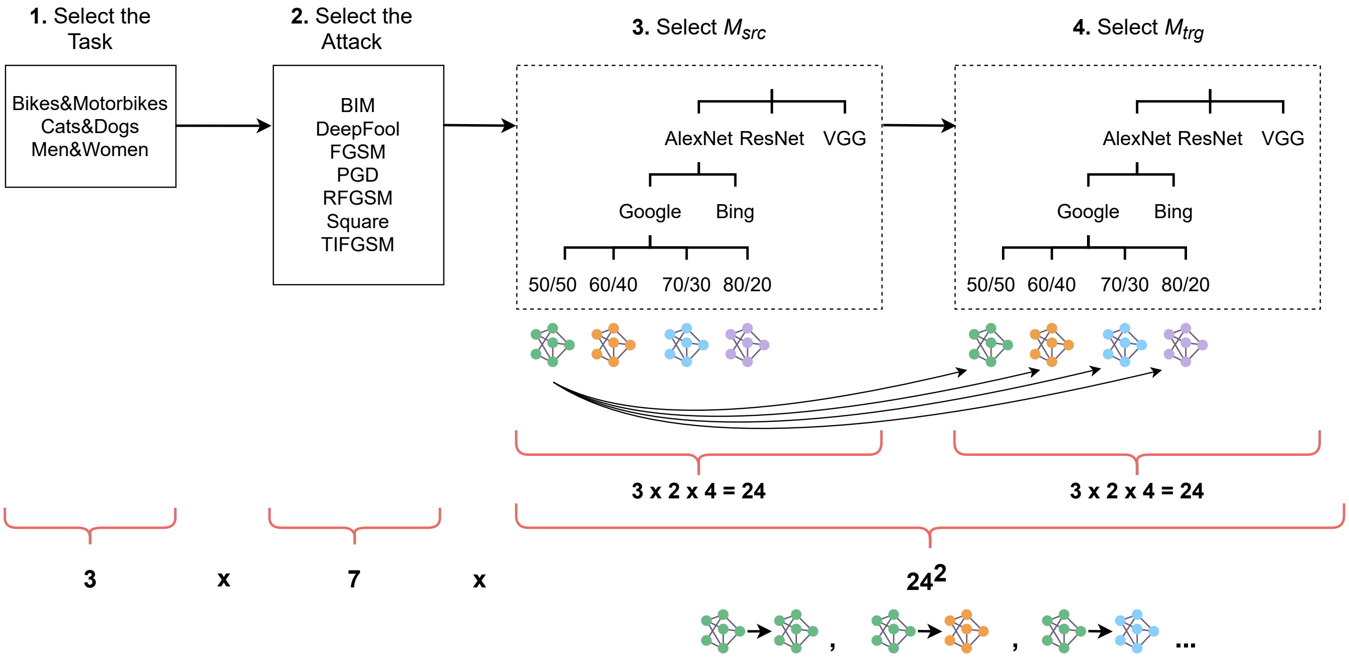


Figure 2: Pipeline of our testing methodology regarding mathematical attacks.

Observation 1: *Compared with high-performant models, models with performance far from perfect appear more vulnerable to adversarial perturbations.*

Attacks. Another noticeable outcome is the superiority of TIFGSM, which outperforms all the other attacks in most cases. We recall that this is the only attack among the considered set explicitly designed for transferability purposes. The attacks produce a strong transferability in the Men&Women task, with an evasion rate close to 1 (perfection) in four out of eight cases.

Last, the "rectangular" shape of TIFGSM. By cross-looking with the DUMB attacker model, we can see that TIFGSM, and more in general, all the considered attacks, provide better performance on attacks where attackers and defenders use the same model architecture (i.e., C1, C2, C5, and C6). Conversely, much lower performance (almost unsuccessful) occurs when attackers and defenders do not share the same model architecture.

Observation 2: *Literature proposes adversarial attacks that struggle to transfer among different architecture.*

Non-Mathematical Attacks. A different pattern can instead be seen in the non-mathematical attacks, shown to be effective in the past by [33]. For simplicity, we report in the paper only the case of Men&Women, while more details about the other tasks are available in our GitHub repository. Figure 4 shows the results. Generally, it appears that simple obfuscations are not effective on our complex models (i.e., AlexNet, ResNet, and VGG). The most effective attack is the RandomBlackBox, which, in contrast, results in the most "altered" images.

While the TIFGSM generally outperforms non-mathematical attacks, this is not always true for the rest of the considered mathematical attacks. Therefore, we count how many times non-mathematical outperforms mathematical attacks for each case of the DUMB attacker model and for each task. We applied 42 comparisons (7 mathematical × 6 non-mathematical) for each case, for a total of 336 tests (42×8 cases). Overall, non-mathematical attacks outperform mathematical 79, 81, and 101 times out of 336 cases for Bikes&Superbikes, Cats&Dogs, and Men&Women, respectively. Furthermore, we analyzed if such successes are uniformly distributed or centered in some of the DUMB cases. The result is shown in Figure 5. The reader can observe that the higher values are found in C3, C4, C7, and C8, highlighting the fragility of mathematical attacks in cases where surrogate and victims do not share the same model architecture.

Observation 3: *Simple obfuscations are solid offensive black-box techniques.*

5.2 Models Impact (M-dimension)

Demontis et al. [8] observed that adversarial transferability depends on the complexity of the surrogate and victim's model. In particular, low-complexity surrogates produce stronger evasion attacks. Similarly, low-complexity victims' models are more resilient to evasions. Low-complexity models should be preferred by both attackers and defenders since, for the former, models tend to produce stable gradients that better align with victims' ones. For the latter, models tend to produce smaller gradients size.

Therefore, we now investigate if we observe similar behavior in our testbed. Due to its effectiveness, we focus on the TIFGSM attack. Figure 6 presents the analysis results by averaging the ASR among

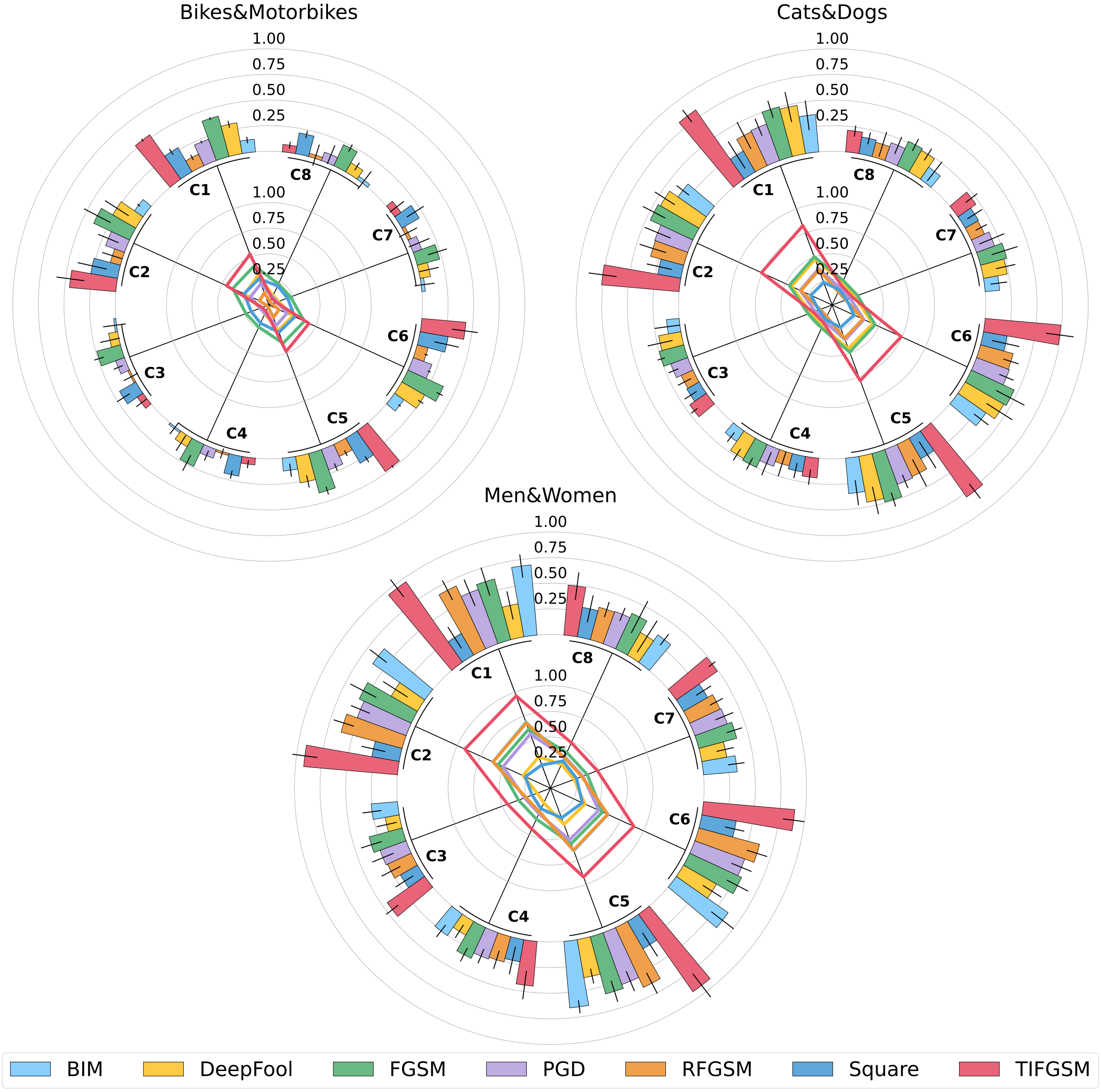


Figure 3: ASR for mathematical attacks. Each subfigure corresponds to a different task and contains two different graphs. The external one shows the performance of our attacks in each of the scenarios of our DUMB attacker model through bar charts. The internal one overviews their overall ASR through a spider chart. While with the former the individual ASR of each attack is more clear, the latter shows their overall performance and trends across the different scenarios. The definitions of the scenarios have been clarified in Table 1.

the three different datasets. We can first observe that, as expected, the highest ASR corresponds to those cases where the source model M_{src} and target model M_{trg} share the same architecture. Second, VGG is the weakest victim model for both AlexNet and ResNet. This is shown by the fact that when VGG is the target model, the ASR is the second highest for all other source models (after the case in which $M_{src} = M_{trg}$). Third, ResNet seems to be the most effective surrogate model. Indeed, when using it as a source model, we see that ASR values are relatively low. At the same time, it is particularly effective as a target model when attacking vulnerable architectures such as VGG. We find such results not aligned with what was discussed by Demontis et al. [8], and in agreement with Mao et al. [20]. Consider the complexity of our models, measured in the number of parameters: 61M for AlexNet, 11M for ResNet, and 132M for VGG. Therefore, ResNet and VGG are the lower and higher complexity models, respectively.

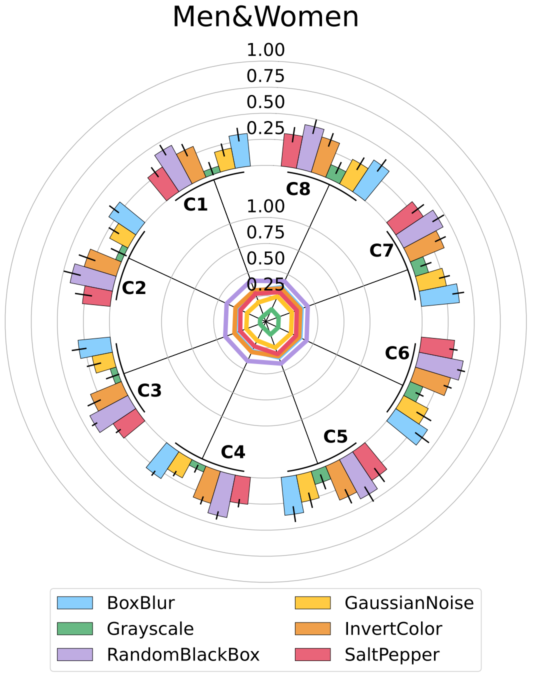


Figure 4: ASR for non-mathematical attacks for the Men&Women task. Information is conveyed in the same way as Figure 3.

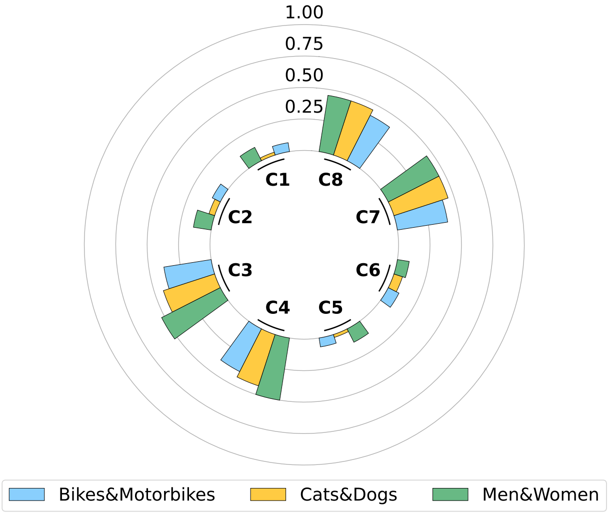


Figure 5: Ratio of non-mathematical outperforming mathematical attacks, grouped by DUMB attacker model cases and tasks.

> **Observation 4:** *Simple surrogate models are not always optimal to transfer evasion attacks.*

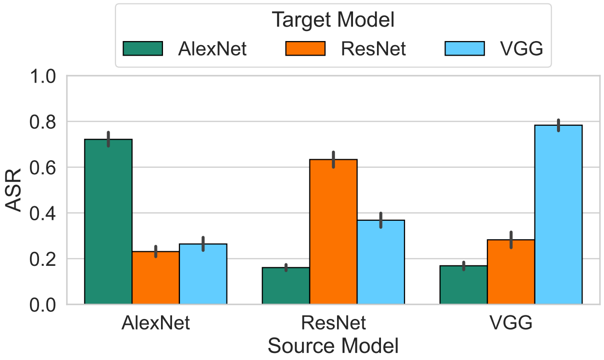


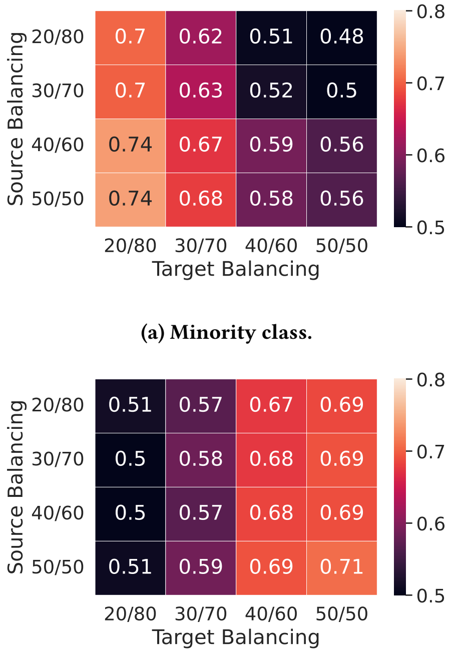
Figure 6: ASR at the varying of source model M_{src} and target model M_{trg}. Here, the "source model" refers to the model that has been used for adversarial attack generation. In contrast, the "target model" refers to the model on which we test the adversarial samples.

5.3 Class Distribution Impact (B-dimension)

One of the hypotheses of our work is that attackers and defenders might have different ground-truth distributions. Therefore, we investigate how class balance levels impact the success of a transferable attack. For simplicity, we show the performance of TIFGSM for the Men&Women dataset. Figure 7 shows the results. The reader can observe an opposite behavior in the transferability between minority and majority classes. In particular, attacking a minority class under a 20/80 ratio is always effective in every source condition (first column of Figure 7a). The attack increases its complexity as we reach a balancing equilibrium. Conversely, it appears to be extremely complex to camouflage a majority sample as a minority one (fist column of Figure 7b). This observed behavior might be extremely relevant, especially in the context of cybersecurity, where ML classifiers are applied in extremely imbalanced contexts, like malware [25] and hate speech detection [7], making such applications weak to transferable attacks.

Another important aspect to consider when the source model is trained in an imbalanced dataset is the choice of the perturbation size. As we introduced in Equation 4, we computed the global ASR for both classes for each task. However, as shown in Figure 8, the majority class tends to require more perturbation to be effective, while the minority requires a little. Therefore, attackers that aim to produce optimal attacks while preserving as much as possible the quality of the samples, should create separate hyperparameter tuning processes for each class.

> **Observation 5:** *The mismatch between the surrogate and victim datasets' class distributions might severely penalize the transferability of the evasion attacks. Furthermore, attacking the minority class appears to be easier compared to the majority.*



(a) Minority class.

(b) Majority class.

Figure 7: ASR for the minority and majority classes. Here, the "source balancing" refers to the balance level of the model that has been used for the adversarial attack generation. In contrast, the "target balancing" refers to the balance level of the model on which we test the adversarial samples.

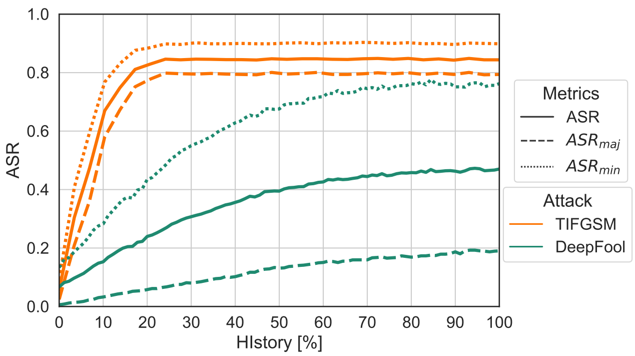


Figure 8: Attacks parameter tuning for Cats&Dogs dataset, in the 20/80 balance level setting. Since TIFGSM and DeepFool use two different types of parameters with different ranges, we use "history" to characterize their level of perturbation.

5.4 Sources Impact (DU-dimension)

Last, we investigate whether the choice of the dataset impacts the attack transferability. The data source has a non-negligible impact if we find at least one case where the choice of the source produces a varied effect on the attack outcome. For example, we can examine the strong imbalance setting (20/80) for the Cats&Dogs and Men&Women tasks. This scenario is particularly interesting to study since models typically perform well on the former task but struggle with the latter, often achieving an F1-score lower than 90, as previously shown in Table 3. We analyze the ASR obtained using the mathematical attacks by considering data sources mismatch, i.e., a surrogate trained on the Bing dataset used to attack models previously trained on the Google dataset, and vice versa. This corresponds to C5, C6, C7 and C8 of our DUMB attacker model (Table 1).

Figure 9 shows the observed distributions. We can notice that for Cats&Dogs, the two curves almost overlap, while there is a partial mismatch in Men&Women. Specifically, regarding the Men&Women task, it appears that attacks directed toward models trained on the Google dataset (and thus generated from a model trained on the Bing dataset) yield better results. This behavior also reflects the baseline evaluation for the two datasets in Table 3, where on the same tasks, models trained on Google had lower F1 scores with respect to the ones trained on Bing. We statistically confirmed what was observed with the Kolmogorov-Smirnov test (two-sided, the null hypothesis is that the two distributions are equal). We reject the null hypothesis in the Cats&Dogs case with a $p_{val} = 0.01$. We, therefore, conclude that the choice of the dataset impacts the attack transferability.

Observation 6: *Generating surrogate data might produce a drop in the performance of transferable attacks.*

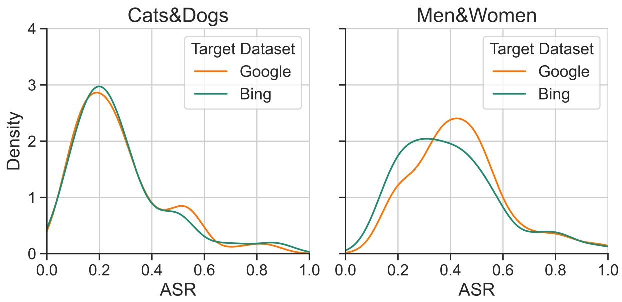


Figure 9: Probability Density Function of ASR for mismatch sources, over 20/80 balance level. The range of possible ASR is reported on the x-axis, while the y-axis shows the probability density of each ASR. The curve represents the shape of the PDF, where the peak corresponds to the most likely success rate and the width indicates the range of success rates that are probable.

6 CONCLUSION

Transferring evasion attacks among different machine learning models is challenging in a real-world scenario. While the use of surrogate models has been widely studied in the field of adversarial transferability, many more variables must be considered to depict the full picture of its effectiveness.

In this work, we fill such a gap by proposing the **DUMB** attacker model. This framework allows analyzing if evasion attacks fail to transfer when the training conditions of surrogate and victim models differ. This framework considers three distinct conditions: **D**ataset so**U**rce, **M**odel architecture, class **B**alance of the dataset. Therefore, surrogate and victim models might vary based on the combinations of these conditions, e.g., surrogate and victim models are trained on the same dataset and ground-truth distribution, but they use different architectures.

We evaluated the DUMB attacker model on our novel **DUMB** testbed, consisting of 3 distinct binary computer-vision tasks, with two dataset versions each – collected with Bing and Google as sources –, 4 type of imbalance conditions (from balanced to highly imbalanced), and 3 state-of-the-art model architectures. By analyzing 7 well-known evasion attacks and 6 simple image transformations, we explored a total of 13K attacks.

Considerations. Our extensive evaluation unveiled aspects that were ignored in the literature or not extensively investigated, with the following repercussions:

(1) The complexity of the task might have a direct impact on the success of evasion attacks' transferability. As shown in Section 5.1, models showing lower performance on the task appear less robust to adversarial attacks. Future works should better investigate the interplay between performance and robustness.
(2) The above point has a direct impact on real-life machine-learning applications. In particular, often, such tools show performance far from being perfect. This results in tools that are more prone to fail in the presence of adversaries. Therefore, the cybersecurity community should utilize both toy-sh and real-world tasks, where with the former, researchers can gain insights about attacks, and with the latter, adapt such insights to complex scenarios.
(3) In general, it appears that evasion attacks fail to transfer when the training conditions of surrogate and victim models differ. Future researchers might benefit from both the **DUMB** attacker model and the **DUMB testbed** to analyze the transferability of novel proposed attacks.
(4) While the literature extensively covers the effect of model architecture, little is known about the impact of dataset source and class balancing. For the former, the data generation and labeling process might introduce biases that might impact the transferability. For the latter, many tasks are inherently imbalanced (e.g., spam/non-spam, malware/non-malware), and due to data generation processes or undersampling/oversampling strategies, it is likely that attacker and victim datasets present different ground truth distributions.
(5) Targeting different classes might lead to different transferable performances. Little attention has been given to the properties of the target class we aim to attack. For instance, when considering the MNIST dataset, the choice seems arbitrary. On the opposite, in cybersecurity tasks, the usual class is the malicious one (e.g., spam, malware). An important property to consider we observed is its numerosity: minority classes of highly imbalanced datasets appear to require limited perturbations to fool (see Figure 8). Future researchers should include such a consideration since many real-life tasks are inherently highly imbalanced, especially those covered by the cybersecurity community.
(6) We did not observe a model architecture superior in acting as a surrogate model. Future researchers should better investigate the interplay between complex model architectures and their ability to generate transferable attacks.

Limitation and Future Work. In this study, we aim to provide a systematic view of factors affecting transferability related to the training of a surrogate model. Therefore, some conclusions remain not fully answered and require further studies. For instance, our proposed testbed is defined by binary tasks, and our conclusions might not be extended to multiclass tasks. Furthermore, our experiment included our novel testbed containing somehow toy-sh tasks, and therefore, far from real conditions. However, our testbed allowed us to clarify different aspects of transferable attacks. Therefore, we believe the proposed testbed might be a precious resource for future researchers conducting analyses in adversarial machine learning. In particular, we believe that both **DUMB** attacker model and testbed can be utilized to extend the analyses of attacks, for instance, from evasion to poisoning. Moreover, we believe that our work can inspire the definition of novel testbeds, considering, for instance, cybersecurity tasks such as spam and malware detection and network intrusion detection systems.

REFERENCES

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*. Springer, 484–501.

[2] Giovanni Apruzzese, Hyrum S Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin A Roundy. 2022. " Real Attackers Don't Compute Gradients": Bridging the Gap Between Adversarial ML Research and Practice. *arXiv preprint arXiv:2212.14315* (2022).

[3] Giovanni Apruzzese and Michele Colajanni. 2018. Evading botnet detectors based on flows and random forest with adversarial samples. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–8.

[4] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. 2006. Can Machine Learning Be Secure?. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security* (Taipei, Taiwan) *(ASIACCS '06)*. Association for Computing Machinery, New York, NY, USA, 16–25.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[6] Mauro Conti, Luca Pajola, and Pier Paolo Tricomi. 2022. Captcha Attack: Turning Captchas Against Humanity. *arXiv preprint arXiv:2201.04014* (2022).

[7] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, Vol. 11. 512–515.

[8] Ambra Demontis, Marco Melis, Maura Pintor, Jagielski Matthew, Battista Biggio, Oprea Alina, Nita-Rotaru Cristina, Fabio Roli, et al. 2019. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX security symposium*. USENIX Association, 321–338.

[9] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. 4307–4316. https://doi.org/10.1109/CVPR.2019.00444

[10] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. 2018. Synthetic data augmentation using GAN for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 289–293.

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

[12] Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is" love" evading hate speech detection. In *Proceedings of the 11th ACM workshop on artificial intelligence and security*. 2–12.

[13] Kathrin Grosse, Lukas Bieringer, Tarek Richard Besold, Battista Biggio, and Katharina Krombholz. 2022. " Why do so?"–A Practical Perspective on Machine Learning Security. *arXiv preprint arXiv:2207.05164* (2022).

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[15] Hoki Kim. 2020. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950* (2020).

[16] Alex Krizhevsky. 2014. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997* (2014).

[17] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[18] Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. *arXiv preprint arXiv:2202.11176* (2022).

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

[20] Yuhao Mao, Chong Fu, Saizhuo Wang, Shouling Ji, Xuhong Zhang, Zhenguang Liu, Jun Zhou, Alex X Liu, Raheem Beyah, and Ting Wang. 2022. Transfer Attacks Revisited: A Large-Scale Empirical Study in Real Computer Vision Settings. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1423–1439.

[21] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.

[22] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. 2019. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems* 32 (2019).

[23] Luca Pajola and Mauro Conti. 2021. Fall of Giants: How popular text-based MLaaS fall against a simple evasion attack. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 198–211.

[24] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[25] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, Lorenzo Cavallaro, et al. 2019. TESSERACT: Eliminating experimental bias in malware classification across space and time. In *Proceedings of the 28th USENIX Security Symposium*. USENIX Association, 729–746.

[26] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[27] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. 2018. When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, 1299–1316.

[28] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* (2017).

[29] Xueping Wang, Shasha Li, Min Liu, Yaonan Wang, and Amit K Roy-Chowdhury. 2021. Multi-expert adversarial attack detection in person re-identification using context inconsistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15097–15107.

[30] Ruoyu Wu, Taegyu Kim, Dave Jing Tian, Antonio Bianchi, and Dongyan Xu. 2022. {DnD}: A {Cross-Architecture} Deep Neural Network Decompiler. In *31st USENIX Security Symposium (USENIX Security 22)*. 2135–2152.

[31] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. 2019. Seeing is Not Believing: Camouflage Attacks on Image Scaling Algorithms.. In *USENIX Security Symposium*. 443–460.

[32] Qixue Xiao, Kang Li, Deyue Zhang, and Weilin Xu. 2018. Security risks in deep learning implementations. In *2018 IEEE Security and privacy workshops (SPW)*. IEEE, 123–128.

[33] Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. 2019. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 952–966.

[34] Jie Zhang, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, Lei Zhang, and Chao Wu. 2022. Towards efficient data free black-box adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15115–15125.

A ADDITIONAL EXPERIMENT DETAILS

A.1 Attack Examples

In this paper, we consider many types of attacks to evaluate the different models we train on several datasets and balance levels of the classes. To generate those attacks, as stated in Section 4.4, we must first tune their parameters in order to produce images that maximize the probability of fooling a model while perturbating them as little as possible.

For most of the considered attacks, the level of perturbation added to an image is related to a parameter that can be tuned accordingly. However, by increasing the parameter, the image becomes gradually more degraded until it is unrecognizable even to a human being. For this reason, in Equation 4 we included a constraint on the similarity of the image; namely, the SSIM of the perturbed image must be more than α. In Figure 10, we show the increasing degradation of a sample while increasing the attack parameter. More in particular, we show the effect of ϵ for the FGSM attack on the AlexNet model trained on the Cats&Dogs task with a strong unbalance level.

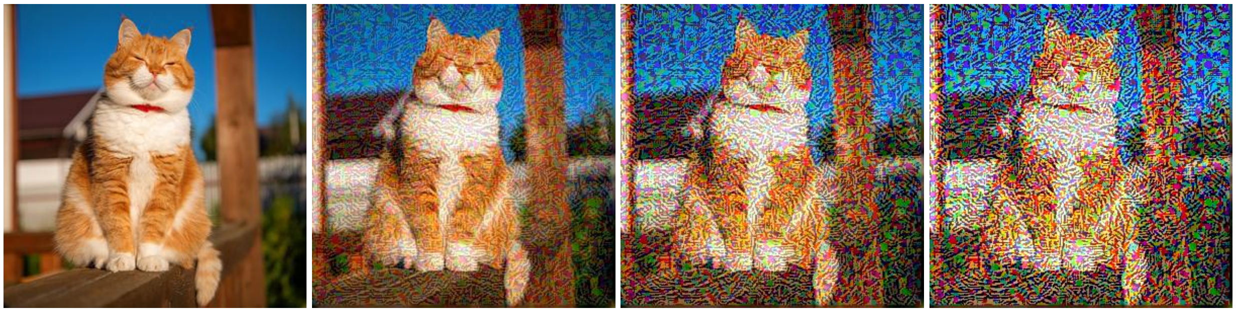
In Figure 11 instead, we extend the visualization to all other mathematical attacks we consider in our testbed. In particular, each sample has been perturbed with its optimal value for ϵ. The samples have been generated to target the AlexNet model trained on the Cats&Dogs task with a strong unbalance level.

Finally, in Figure 12, we show the same effect for non-mathematical attacks. Since some of those image transformations also include a parameter, we also include its optimal value.

A.2 Hardware and Software Configuration

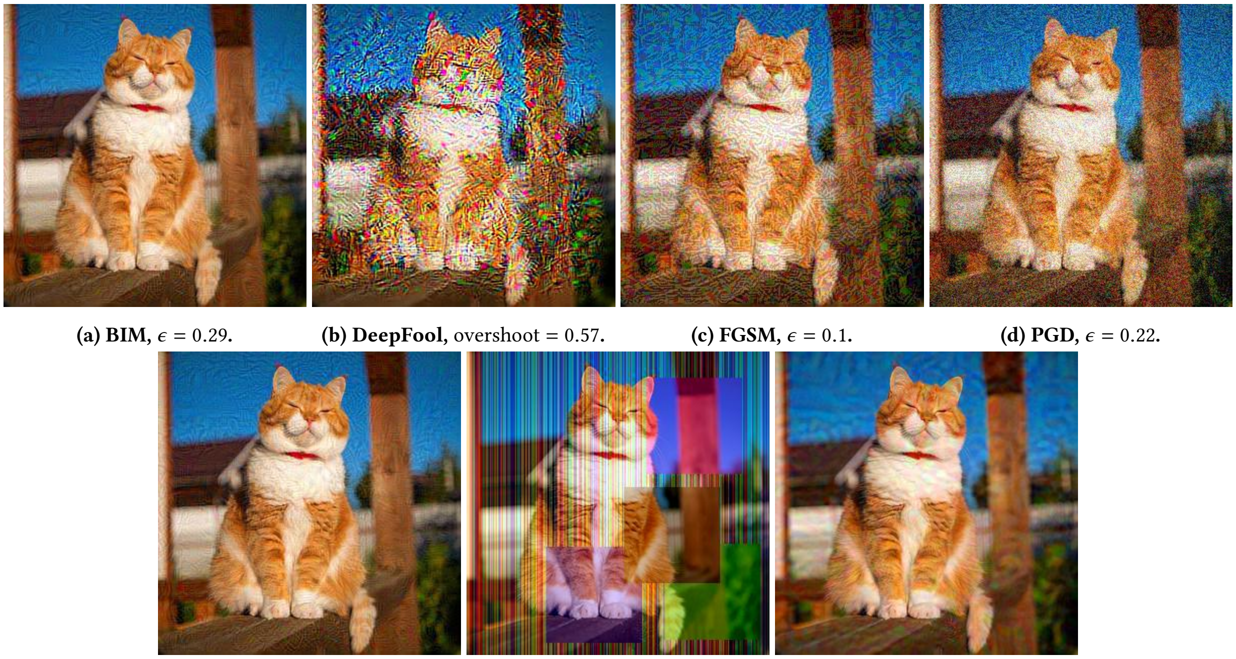
All experiments have been conducted on a workstation with the following configurations:

- **Operating System**: Ubuntu 20.04.4 LTS.
- **CPU**: AMD Ryzen 5 3600X.
- **GPU**: NVIDIA RTX 3090.
- **Software**: Python 3.8.10, Pytorch 1.7.1.



(a) Original. **(b)** ϵ = 0.1. **(c)** ϵ = 0.2. **(d)** ϵ = 0.3.

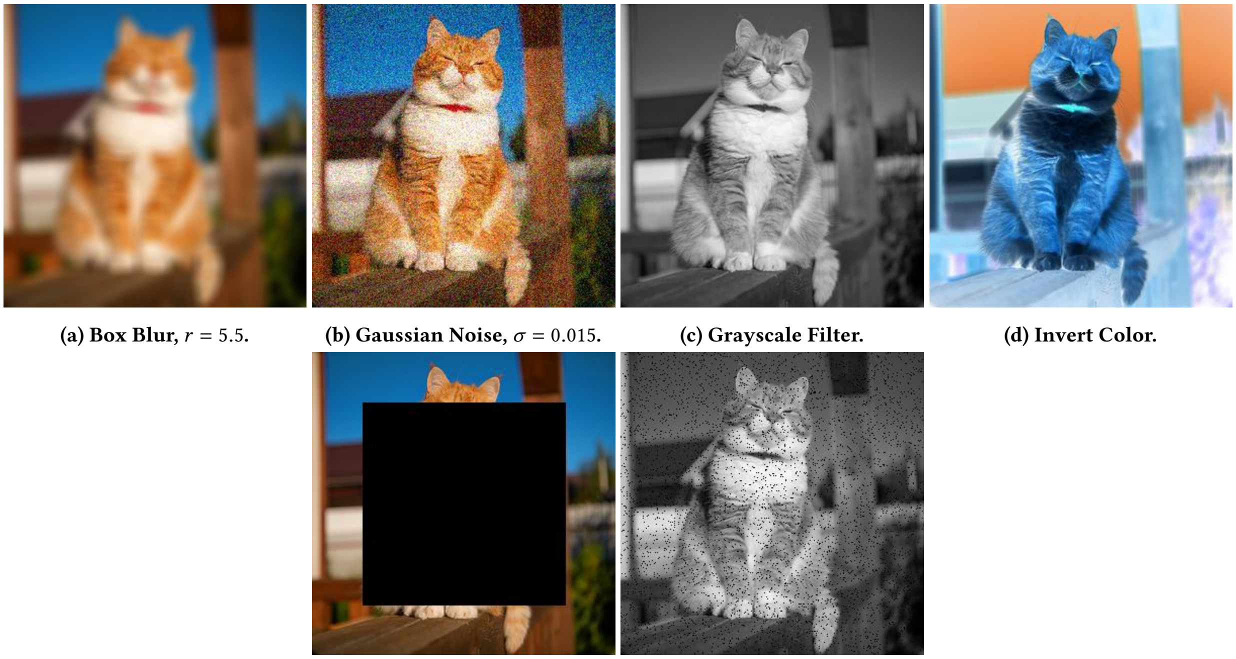
Figure 10: Effect of the attack parameter on the degradation of a sample.



(a) BIM, $\epsilon = 0.29$**.** **(b) DeepFool,** overshoot = 0.57**.** **(c) FGSM,** $\epsilon = 0.1$**.** **(d) PGD,** $\epsilon = 0.22$**.**

(e) RFGSM, $\epsilon = 0.27$**.** **(f) Square,** $\epsilon = 0.15$**.** **(g) TIFGSM,** $\epsilon = 0.28$**.**

Figure 11: Examples of mathematical attacks perturbed with optimal parameter values.



(a) Box Blur, $r = 5.5$**.** **(b) Gaussian Noise,** $\sigma = 0.015$**.** **(c) Grayscale Filter.** **(d) Invert Color.**

(e) Random Black Box, size = 200**.** **(f) Salt and Pepper,** amount = 0.05**.**

Figure 12: Examples of non-mathematical attacks perturbed with optimal parameter values when possible.