DUMB and DUMBer: Is Adversarial Training Worth It in the Real World?

Francesco Marchiori[1][0000−0001−5282−0965], Marco Alecci[2][0000−0002−5963−4599], Luca Pajola[3][0000−0002−6749−6608], and Mauro Conti[1,3][0000−0002−3612−1934]

[1] University of Padova, Padova, Italy
[2] University Of Luxembourg, Luxembourg, Luxembourg
[3] Spritz Matter Srl, Padova, Italy
francesco.marchiori@math.unipd.it, marco.alecci@uni.lu, luca.pajola@spritzmatter.com, mauro.conti@unipd.it

Abstract. Adversarial examples are small and often imperceptible perturbations crafted to fool machine learning models. These attacks seriously threaten the reliability of deep neural networks, especially in security-sensitive domains. Evasion attacks, a form of adversarial attack where input is modified at test time to cause misclassification, are particularly insidious due to their transferability: adversarial examples crafted against one model often fool other models as well. This property, known as adversarial transferability, complicates defense strategies since it enables black-box attacks to succeed without direct access to the victim model. While adversarial training is one of the most widely adopted defense mechanisms, its effectiveness is typically evaluated on a narrow and homogeneous population of models. This limitation hinders the generalizability of empirical findings and restricts practical adoption.

In this work, we introduce **DUMBer**, an attack framework built on the foundation of the DUMB (Dataset soUrces, Model architecture, and Balance) methodology, to systematically evaluate the resilience of adversarially trained models. Our testbed spans multiple adversarial training techniques evaluated across three diverse computer vision tasks, using a heterogeneous population of uniquely trained models to reflect real-world deployment variability. Our experimental pipeline comprises over 130k evaluations spanning 13 state-of-the-art attack algorithms, allowing us to capture nuanced behaviors of adversarial training under varying threat models and dataset conditions. Our findings offer practical, actionable insights for AI practitioners, identifying which defenses are most effective based on the model, dataset, and attacker setup.

Keywords: Adversarial Attacks · Adversarial Training · Transferability.

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable performance across a wide range of tasks, particularly in computer vision, natural language processing, and autonomous systems. However, these models are known to be highly vulnerable to adversarial examples, i.e., carefully crafted inputs that cause the

model to make incorrect predictions while appearing benign to human observers. First identified in the context of image classification [10], adversarial attacks have since evolved into a rich area of research, encompassing various modalities and attack scenarios. One prominent class of these threats is *evasion attacks*, where the attacker modifies inputs at test time to evade detection or mislead the model. Real-world manifestations of such attacks have been observed in malware detection systems [17], facial recognition spoofing [21], and autonomous driving [8], raising significant concerns about the safety and reliability of AI systems deployed in adversarial environments.

A particularly troubling property of adversarial examples is their *transferability*: adversarial inputs crafted for one model often succeed in misleading other models, even if they differ in architecture, training data, or optimization details [6,25]. This phenomenon enables gray-box and black-box attacks, where the attacker has limited or no access to the target model's internals yet can still craft effective attacks using surrogate models. As a result, transferability significantly undermines the security of models in deployed settings. The DUMB framework [1] was recently proposed to study how adversarial transferability varies as a function of three key dimensions: Dataset soUrces, Model architectures, and the Balance of class distributions. It provided a standardized way to evaluate how generalizable adversarial examples are across different training conditions, laying the foundation for more robust empirical evaluations of adversarial threats.

One widely studied defense mechanism against adversarial attacks is *adversarial training*, where the model is trained on adversarial examples in addition to clean data [18,3]. This technique aims to increase the model's robustness by explicitly teaching it to resist known types of perturbations. While adversarial training has shown promise, especially in white-box settings, its effectiveness often comes with trade-offs: it can lead to significant reductions in accuracy on clean (non-adversarial) inputs [24], increase training time, and sometimes fail to generalize beyond the specific attack types used during training. Various improved techniques have been proposed [28,26], each attempting to balance robustness and standard accuracy. However, while the individual performance of adversarially trained models has been extensively evaluated, the impact of adversarial transferability on these defenses remains underexplored. Given the high computational demands of adversarial training and the practical challenges of deploying robust models in real-world settings, particularly under the threat of adversarial transferability, we are prompted to ask: *is adversarial training truly effective in real-world scenarios?* And if so, *which strategies offer the most resilience across diverse attack conditions?* This work seeks to answer these questions by systematically evaluating various adversarial training techniques under controlled yet transferable attack settings.

Contribution. This work presents **DUMBer**, an extension of the DUMB framework that evaluates the impact of adversarial training on transferability across models, datasets, and class balance conditions. While DUMB analyzed the robustness of models to transferred evasion attacks in standard training settings, DUMBer investigates whether and how commonly used adversarial training techniques improve resilience under the same conditions. We focus on the interaction between adversarial training and transferability, providing a systematic evaluation across a broad spectrum of configurations. Our aim is to address a key

limitation in the literature: adversarial training techniques are often evaluated on a narrow and homogeneous set of models, which is quite limiting given the empirical nature of these defenses. In contrast, we adopt a "**DUMB population**" where every model is unique, better capturing the variability encountered in real-world deployments. Our contributions can be summarized as follows.

- We propose **DUMBer**, an attacker model and framework that extends DUMB by incorporating adversarial training into the evaluation of adversarial transferability. Across three axes (**D**ataset so**U**rces, **M**odel architectures, and class **B**alance), our testbed evaluates the **e**vasion **r**esilience of adversarially trained models.
- We present a comprehensive empirical analysis across three computer vision tasks, 13 distinct attack types, and 10 training strategies. Our "**DUMB population**" comprises 240 uniquely trained models spanning diverse datasets and architectures, resulting in over 130,000 individual evaluations.
- Our analysis provides novel insights and best practices on how to apply adversarial training to maximize model robustness under realistic attack scenarios involving transferability.
- We release the full codebase and evaluation scripts to promote reproducibility: `https://github.com/spritz-group/DUMBer`.

Research Questions. The extensive cross-parameter evaluations conducted in this study enable us to address unique research questions that are highly valuable to AI practitioners and future research efforts.

RQ1: Which adversarial attacks are most effective across the entire DUMB population? In other words, what attack strategies perform best regardless of the adversary's level of knowledge or access?

RQ2: Which adversarial training strategies offer the highest overall robustness? What defense techniques are most effective across the diverse scenarios represented in the DUMB population, irrespective of the attacker's assumptions?

RQ3: How do different training strategies perform across varying evaluation scenarios `C`? In other words, what are the best- and worst-case conditions for deploying each defense strategy in practice?

RQ4: Do robust defenses against strong attacks generalize to weaker ones? Can adversarial training improve resilience uniformly across a spectrum of attack strengths, or is its benefit limited to specific threat levels?

RQ5: When and where does adversarial training fail? Specifically, how frequently does it result in a negative AMR, and are these failures uniformly distributed across scenarios `C`?

2 Related Works

We now review the state-of-the-art in adversarial training and transfearbility.

Adversarial Transferability. Adversarial examples are designed to deceive machine learning models and often transfer across different models, enabling black- and gray-box attacks through surrogate models. Recent work has investigated the factors behind adversarial transferability. Gu et al.[13] provide a comprehensive survey, categorizing methods to enhance transferability and outlining core principles and challenges. They stress the need for robust evaluation frameworks covering diverse architectures and tasks. Building on this, Yu et al.[27] show that transferability is often overestimated when evaluations are limited to similar architectures, such as CNNs, and call for broader benchmarks across different neural networks. These findings motivate the need for frameworks like DUMB and DUMBer, which systematically assess adversarial transferability across dimensions like dataset source, model architecture, and class balance.

Adversarial Training. Adversarial training is a leading defense strategy that incorporates adversarial examples into the training process to improve model robustness. Early approaches include FGSM-based training [10], which uses the Fast Gradient Sign Method, and PGD-based training [18], which applies iterative Projected Gradient Descent for stronger perturbations. Both aim to defend against both seen and unseen attacks. Curriculum adversarial training [4] extends these ideas by gradually increasing perturbation strength (ϵ) during training, promoting more stable robustness. Ensemble adversarial training [23] further diversifies defenses by introducing adversarial examples from multiple pre-trained models, improving resilience against transfer attacks. With DUMBer, we evaluate these techniques and their variants, also considering model-agnostic perturbations, to systematically measure their impact on robustness across diverse and transferable adversarial scenarios.

3 Threat Model

The original DUMB attacker model emphasizes the importance of simulating realistic conditions for adversarial transferability—conditions often neglected in the current literature [12]. Building on these foundations, our DUMBer framework explicitly addresses three critical challenges that arise during attack execution in practical settings:

- *Dataset:* Most prior works assume that both the attacker and victim have access to the same dataset, which is rarely the case in real-world scenarios. Constructing a surrogate dataset is far from trivial, as it depends heavily on corpus generation strategies that can vary significantly across domains and institutions. For example, in hate speech detection, it has been shown that existing datasets are constructed using divergent methodologies, leading to models that perform well on their training data but generalize poorly to others [11]. This undermines the assumption that transferability is guaranteed simply by using adversarial examples.
- *Ground-truth distribution:* A related yet distinct issue concerns the assumption that attacker and victim datasets share the same underlying distribution. In practice, this is rarely true. Differences may stem from distinct data collection processes or disparate preprocessing and augmentation techniques.

This mismatch becomes even more pronounced in imbalanced tasks, where techniques such as SMOTE [5] or GAN-based oversampling [9] are often employed to rebalance class distributions, further distorting the comparability of training data across parties.

– *Model architecture:* Attackers and victims typically do not rely on identical models. While prior work sometimes evaluates transferability across model families, the space of potential architectures is vast, ranging from standard CNNs to more sophisticated and customized models. For instance, in computer vision alone, one might choose among VGG variants (e.g., VGG16, VGG19) or ResNet families (e.g., ResNet18, ResNet50), each of which may respond differently to adversarial perturbations. This diversity introduces an additional layer of unpredictability in the effectiveness of transfer-based attacks.

In Table 1, we summarize eight representative attack scenarios, each capturing a possible mismatch between the attacker's surrogate model and the victim's target model. In realistic settings, the attacker typically does not know which scenario they are operating in, except in the idealized white-box case.

Table 1: Conditions for each case in the DUMB attacker model. Scenarios for each `C` are included in [1]. Subscripts a and v denote attacker and victim, respectively.

Case	$\mathbf{DU}_a$ **vs** $\mathbf{DU}_v$	$\mathbf{M}_a$ **vs** $\mathbf{M}_v$	$\mathbf{B}_a$ **vs** $\mathbf{B}_v$
C1	●	●	●
C2	●	●	○
C3	●	○	●
C4	●	○	○
C5	○	●	●
C6	○	●	○
C7	○	○	●
C8	○	○	○

● = match, ○ = mismatch.

`C1` = pure white-box, `C8` = pure black-box

Beyond these structural mismatches, an additional layer of complexity arises from the widespread adoption of adversarial training techniques. Deployed models, especially those exposed to end-users, are often hardened through such defenses to enhance robustness and safety alignment. As a result, attackers must overcome transferability challenges and contend with models explicitly trained to resist adversarial inputs. The objective of DUMBer is to systematically evaluate how these real-world conditions, ranging from mismatched assumptions to adversarial training, impact the effectiveness of transfer-based attacks.

4 Methodology

Next, we present our methodology, which is built on the foundation of the DUMB framework. While our evaluation covers the eight cases from Section 3, our main

focus is a systematic study of how adversarial training affects these scenarios. Section 4.1 defines the dimensions of transferability, Section 4.2 overviews the attacks, Section 4.3 describes training dimensions, and Section 4.4 explains our unified testing framework.[1] An overview of the training process is shown in Fig. 1.

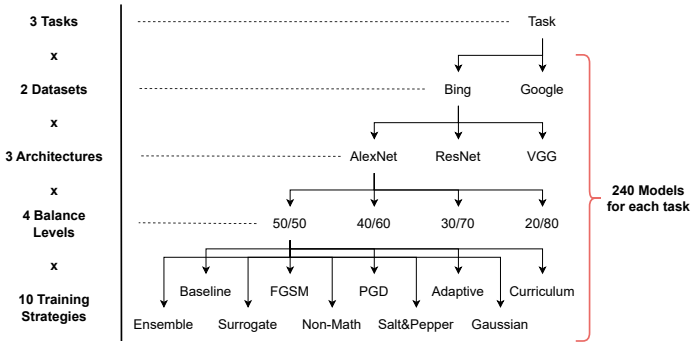


Fig. 1: Model combinations during the training phase.

4.1 Transferability Dimensions

This work focuses on three distinct computer vision tasks, each representing common settings found in adversarial attack literature. These tasks are framed as binary classification problems: distinguishing between Bikes vs. Motorbikes (B&M), Cats vs. Dogs (C&D), and Men vs. Women (M&W). For each task, we systematically vary the dimensions that influence adversarial transferability.

Dataset Source To meet the specific needs of our testbed, we manually collect and validate two distinct datasets for each task from Bing and Google, ensuring control over complexity and potential biases. Starting with an average of 14,264 images per dataset, we remove duplicates using difPy, perform manual inspections to eliminate mislabeled or low-quality samples, and randomly select 10,000 balanced images per dataset. All images are then resized to 300×300 RGB using anti-aliasing to maintain quality.

Model Architecture We employ three widely-used and well-established computer vision architectures: AlexNet [15], ResNet18 [14], and VGG11 [22]. We fine-tune these models across our experimental tasks, building on architectures also explored in the DUMB framework. Training procedures align with the official PyTorch guidelines to ensure reproducibility.

[1] Underlined terms highlight key dimensions impacting the DUMB population or evaluation setup.

Ground Truth Balancing To assess the impact of class imbalance between attacker and defender, we simulate four levels of imbalance in the training sets: balanced (50/50), weak (40/60), medium (30/70), and strong (20/80), always treating the first class (Cats, Men, Bikes) as the minority across all tasks. The minority class is undersampled accordingly while keeping the majority class fixed (e.g., for strong imbalance in the Cats vs. Dogs task, 875 Cats vs. 3500 Dogs). This procedure affects only the training sets; validation and test sets remain balanced.

4.2 Attacks

We incorporate two broad categories of adversarial attacks, mathematical and non-mathematical, for a total of 13 distinct attacks across different threat models.

Mathematical Attacks Mathematical attacks are model-aware and generated through optimization methods that leverage gradient information. Our evaluation includes widely used attacks such as FGSM [10], BIM [16], PGD [18], RFGSM [23], DeepFool [20], TIFGSM [7], and Square [2], all implemented via the Torchattacks Python library. These attacks differ in complexity and transferability, offering a robust basis to test model vulnerability under white-box and black-box scenarios. It is important to note that mathematical attacks are generated at the "balance level" described in Fig. 1, without incorporating any adversarial training on the source model used for attack generation.

Non-Mathematical Attacks On the other hand, non-mathematical attacks rely on simple image transformations independent of any model, making them practical in real-world settings. Using only basic image processing via PIL, we simulate attacks like Gaussian noise, grayscale conversion, box blur, salt-and-pepper noise, random occlusion (black box), and color inversion. Though simpler, these attacks pose realistic threats to vision models and highlight the relevance of evaluating robustness beyond optimization-based techniques.

Parameter Tuning Each attack, whether mathematical or not, has at least one parameter controlling its strength (e.g., ϵ for mathematical attacks, radius or noise level for others). We tune these parameters to maximize attack success while preserving visual similarity, enforcing a minimum Structural Similarity Index Measure (SSIM) of 0.4 to prevent excessive distortion. For testing, we select optimal parameters to evaluate model robustness under pure evasion. For validation, we generate adversarial examples across multiple parameter values to support adversarial training, ensuring no information leakage between training and testing.

4.3 Training Strategies

We begin by training a baseline model on unperturbed data, reflecting nominal conditions without adversarial influence. This is a control scenario to assess the degradation caused by adversarial attacks. Subsequently, we investigate nine adversarial training strategies designed to enhance robustness against evasion

attacks. All adversarial training experiments are conducted from scratch rather than fine-tuning a nominally trained model. This choice avoids compromising between adversarial and nominal performance—an issue commonly encountered in fine-tuning approaches. For each adversarial training strategy, we construct a training dataset composed of 80% original (clean) samples and 20% adversarial samples. These adversarial examples are drawn from the validation set, following the generation procedure described in Section 4.2. While this general setup holds for all strategies, specific implementations may vary based on the training logic. We define and adapt the following training strategies to facilitate integration within our framework.

- FGSM: Includes validation-time adversarial samples generated using FGSM at a fixed $\epsilon = 0.2$, selected to balance perturbation visibility and attack strength [10].
- PGD: Mirrors the FGSM setup but employs PGD as the attack method, also with $\epsilon = 0.2$ [18].
- Ensemble: Incorporates adversarial examples from both FGSM and PGD attacks, aiming to increase robustness through attack diversity [23].
- Surrogate: Similar to Ensemble, but uses adversarial examples generated by different model architectures (trained under the same DUMB configuration) to simulate transferability-based threats.
- Curriculum: Progressively increases ϵ over training epochs while generating FGSM attacks offline [4]. This method introduces gradually harder adversarial examples during training.
- Adaptive: Also increases ϵ over time but generates adversarial examples online during training [19]. This approach adapts to the model's current weaknesses, simulating evolving attack sophistication.
- Non-Mathematical Mixture: Employs all non-mathematical attacks at fixed perturbation strengths, ensuring diversity without reliance on model gradients.
- Gaussian: Uses only Gaussian noise as the adversarial perturbation in the 20% adversarial portion of the training set.
- Salt-and-Pepper: Uses only salt-and-pepper noise to generate adversarial examples for training, enabling focused analysis of noise-based robustness.

4.4 Testing

Designing the evaluation framework presents unique challenges, as the complexity of adversarial attack generation is not aligned with the training process of the selected target model, unlike the original DUMB framework, where these components are tightly coupled.

Mathematical Attacks Gradient-based adversarial attacks must be crafted using dedicated models, referred to as *source models* M_{src}. Our transferability dimensions determine the number of these models: for each task, we consider 2 dataset sources, 3 model architectures, and 4 data balance levels, resulting in $2 \times 3 \times 4 = 24$ source models per task. Each of the 7 mathematical attacks is then evaluated against a broader set of *target models* M_{trg}, which includes the full combination

space shown in Fig. 1, amounting to 240 unique configurations. Consequently, the total number of mathematical attack evaluations amounts to: 3 tasks × 7 attacks × 24 M_{src} × 240 M_{trg} = 120,960, where 240 M_{trg} correspond to 2 dataset sources × 3 model architectures × 4 data balance levels × 10 training strategies.

Non-Mathematical Attacks Model-agnostic image perturbations, such as those not relying on gradients, do not require dedicated M_{src} and can be applied directly to the input data. Nonetheless, the dataset source remains relevant, as the images differ between the Bing and Google datasets. As a result, the total number of evaluations for non-mathematical attacks is: 3 tasks × 6 attacks × 2 datasets × 240 M_{trg} = 8,640.

5 Results

In this section, we present the results of our evaluation framework and the effect that adversarial training strategies have on the different transferability cases. We first define the metrics for our evaluation in Section 5.1, followed by a baseline evaluation of our trained models in Section 5.2. We then answer the research questions detailed in Section 1.

5.1 Metrics

From a machine learning standpoint, each of our tasks is formulated as a binary classification problem, where the model learns to distinguish between two distinct classes based on the input images. Since label balancing plays a central role in our analysis, we adopt the F1 score as the primary evaluation metric to assess the classification performance of the models. The F1 score provides a balanced measure of precision and recall, making it particularly suitable for evaluating performance in scenarios where class distribution may vary. This score is defined as follows.

$$F1 = 2\frac{precision \cdot recall}{precision + recall}. \tag{1}$$

To assess both the impact of adversarial attacks and the robustness of adversarially trained models, we introduce two additional evaluation metrics: Attack Success Rate (ASR) and Attack Mitigation Rate (AMR). ASR, also used in the original DUMB framework, measures the proportion of samples classified initially correctly by a model under clean conditions but misclassified after applying the attack. This quantifies the effectiveness of the attack in degrading model performance. To enable a more fine-grained assessment, we also introduce a metric called *severity*, assigning each attack attempt to one of five levels that evenly partition the ASR range from 0% to 100%. This distinction captures attacks ranging from minimal impact (severity score 1) to significant degradation (severity score 5). AMR, instead, is the difference between the ASR before and after adversarial training, normalized by the ASR before training. Specifically, AMR is calculated as:

$$AMR = \frac{ASR_{original} - ASR_{adv}}{ASR_{original}}, \tag{2}$$

where the subscript "original" refers to the ASR measured on the baseline model, and "adv" refers to the ASR of the adversarially trained model. This metric quantifies the attack's success rate reduction following adversarial training. A higher AMR indicates that a specific training strategy has more effectively mitigated the attack's impact, improving the model's robustness. It is important to note that AMR is upper-bounded at 100% (indicating that all attacks were mitigated successfully), but it is theoretically not lower-bounded. This is because adversarial training could also deteriorate the model's performance in nominal and adversarial scenarios, resulting in a negative or zero value for AMR. To maintain interpretability, we cap AMR values at -100%, acknowledging that stronger degradations are possible, though they often occur when the original ASR was already low. We thus define +100% as perfect improvement and -100% as complete degradation.

5.2 Baseline Evaluation

Before assessing the impact of evasion attacks, it is crucial to ensure that our models perform reliably under standard, unperturbed conditions. Table 2 presents the F1 scores across all models and training strategies on the nominal dataset. As we adopt the same tasks defined in the original DUMB framework, we observe a consistent trend in the baseline evaluation, where B&M emerges as the most straightforward task, while M&W proves to be the most challenging. Overall, adversarial training does not significantly degrade performance on clean data. Most strategies maintain parity with the baseline and, in some cases, even improve it. Notably, Gaussian noise and Adaptive training slightly enhance model performance, suggesting a potential regularization effect that helps generalization. The only approach that noticeably reduces performance is the Ensemble method, which shows an average drop of 4.26% compared to the baseline. This may be due to the added complexity of combining multiple decision boundaries, which could reduce precision in non-adversarial contexts.

Table 2: Model performance (F1 score) across different training strategies. Results are averaged on the dataset source and ground-truth balance dimensions.

Task	Model	Base	FGSM	PGD	Ens.	Sur.	Cur.	Ada.	N.M.	Gau.	S.&P.
B&M	AlexNet	0.976	0.978	0.979	0.892	0.979	0.978	0.978	0.979	0.980	0.977
	ResNet	0.986	0.987	0.986	0.987	0.988	0.987	0.985	0.987	0.987	0.988
	VGG	0.985	0.985	0.986	0.987	0.988	0.986	0.985	0.987	0.985	0.986
C&D	AlexNet	0.953	0.948	0.949	0.933	0.947	0.952	0.953	0.951	0.950	0.948
	ResNet	0.978	0.978	0.979	0.938	0.978	0.978	0.978	0.978	0.979	0.978
	VGG	0.982	0.981	0.982	0.939	0.981	0.981	0.982	0.980	0.982	0.981
M&W	AlexNet	0.858	0.862	0.861	0.825	0.857	0.856	0.868	0.864	0.865	0.859
	ResNet	0.921	0.921	0.922	0.871	0.917	0.925	0.920	0.920	0.922	0.922
	VGG	0.925	0.925	0.925	0.831	0.923	0.925	0.925	0.920	0.925	0.921
Avg. Change w.r.t Base			+0.02%	+0.06%	-4.26%	-0.08%	+0.05%	+0.13%	+0.03%	+0.14%	-0.05%

Ens. = Ensemble, **Sur.** = Surrogate, **Cur.** = Curriculum, **Ada.** = Adaptive.
N.M. = Non-Math-Mix, **Gau.** = Gaussian noise, **S.&P.** = Salt-and-pepper noise.

5.3 Attack Overview

Before analyzing the performance of the DUMB population across its subgroups, we first evaluate the overall effectiveness of the attacks described in Section 4.2. An overview is presented in Fig. 2, where results are averaged across `C` scenarios and tasks to reflect general attack behavior under diverse, balanced conditions. A more detailed scenario-specific analysis was already provided in the original DUMB paper [1]. Fig. 2 shows that TIFGSM is the most effective strategy in the average DUMB scenario. This is unsurprising given its design to enhance transferability, particularly benefiting "grayer" box settings. Conversely, attacks such as BIM, Square, and RFGSM predominantly fall into severity score 1, suggesting limited effectiveness in more generalized or transfer-based contexts. Non-mathematical attacks predictably show lower severity, although techniques like RandomBlackBox and BoxBlur occasionally outperform mathematical attacks such as BIM and Square, especially when averaging across all attacker scenarios. Notably, within severity score 1, certain attacks display a large fraction of extremely low ASR cases: for example, BIM and RFGSM fall below a 5% ASR in 24% of evaluations, highlighting their limited transferability. A more detailed analysis on each task is shown in Appendix A.1.

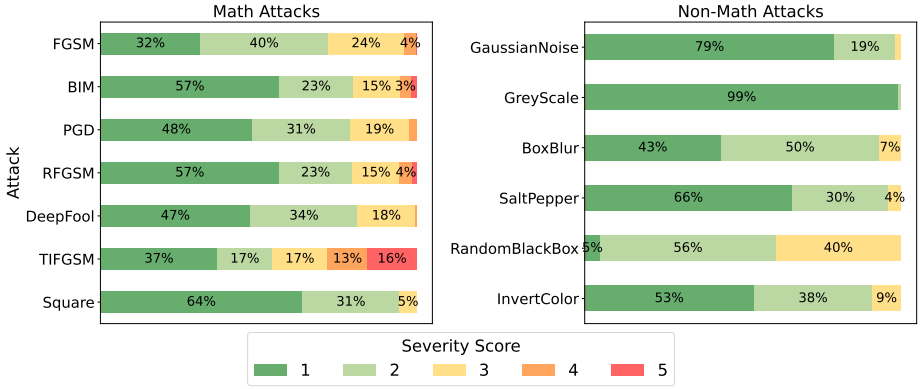


Fig. 2: Severity score distribution of each attack.

5.4 Adversarial Training Overview

Following the approach in Section 5.3, we now assess the AMR of each training strategy across the full DUMB population. The goal is to identify which method performs best when the defender has no knowledge of the attacker's capabilities. This evaluation considers each `C` scenario to reflect how a "naive" defender might deploy protection without tailored assumptions. Results are shown in Fig. 3. Severity 5 is missing for B&M, as no attacks reached that level. Severity 1 is excluded across all tasks, as such attacks have negligible ASR and would distort

AMR interpretation by exaggerating insignificant changes. Our focus on higher severities aligns with the paper's aim to support resilience against impactful threats. *Adaptive* training stands out, achieving up to 96.69% AMR on B&M. *Curriculum* and *surrogate* also perform well, while non-mathematical approaches offer modest but more consistent improvements. Some negative AMR values appear, especially on the M&W task. This stems from the task's lower baseline F1 score (see Table 2), making it more sensitive to training disruptions. Still, since AMR remains strongly positive at higher severities, practitioners should weigh these occasional drops—often linked to low-impact attacks—against the broader gains in robustness.

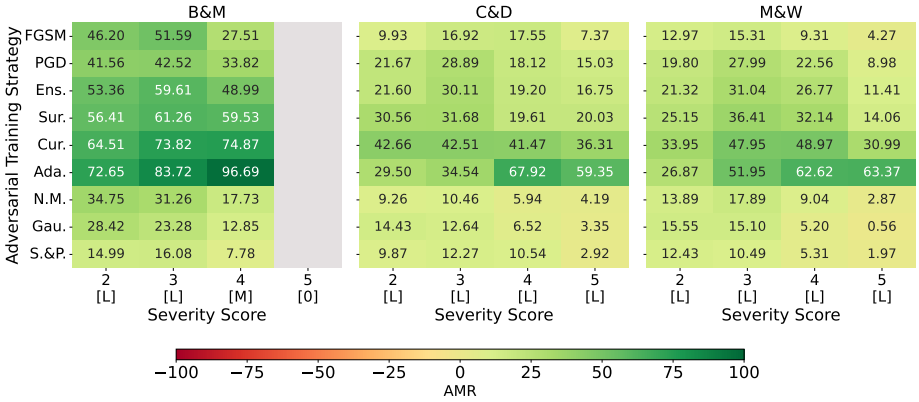


Fig. 3: Training strategies' AMR at different attack severity scores. Labels below scores indicate sample sizes for each cell: `[S]` (1–10), `[M]` (10–50), and `[L]` (50+).

5.5 Adversarial Training on DUMB

In Sections 5.3 and 5.4, we provided general overviews of attacks and training strategies by averaging across all DUMB scenarios. We now focus on a scenario-specific analysis, examining how each strategy performs under different `C` settings. This detailed view is intended for practitioners familiar with their threat model and the critical attack scenarios they face, as discussed in the original DUMB paper. To ensure relevance, we limit our analysis to attacks with severity scores of 3 or higher. This filters out cases where negative AMR values reflect negligible performance changes and emphasizes scenarios where adversarial training meaningfully impacts robustness. These results are summarized in Fig. 4. We observe that adversarial training is most effective when the source and target models share the same architecture (e.g., `C1`, `C2`, `C5`, `C8`), likely due to higher ASR in those cases, as noted in the original DUMB study. Consistent with Section 5.4, *adaptive* training can yield negative AMR when model architectures differ, possibly due to lower initial ASR. A noteworthy trend in the M&W task is that AMR drops significantly when the dataset source differs, a pattern less evident

in other tasks. This aligns with prior findings identifying M&W as particularly sensitive to dataset mismatch, resulting in larger ASR discrepancies. Finally, some task-scenario combinations are under-represented due to our severity $\geq$3 filter, especially where data is already sparse (e.g., `C1-4-5-7` in B&M, `C3-4-7-8` in C&D). As such, outliers like the *adaptive* `C3` case in C&D should not be over-interpreted; broader evaluation is needed to confirm such trends.

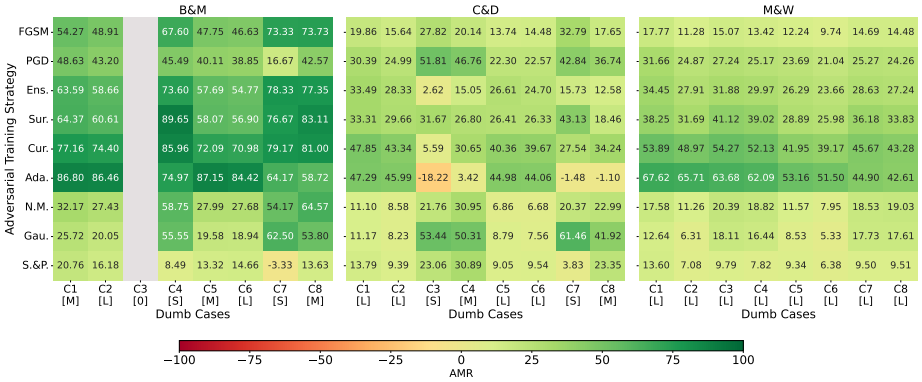
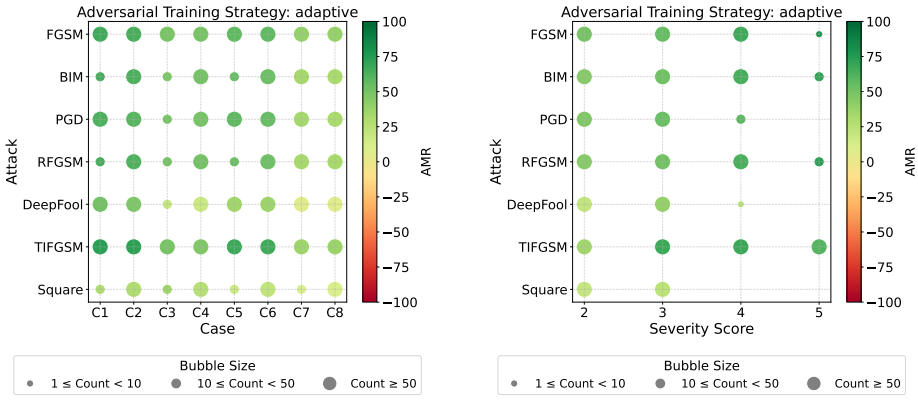


Fig. 4: Training strategies' AMR at different DUMB cases. Labels below scores indicate sample sizes for each cell: `[S]` (1–10), `[M]` (10–50), and `[L]` (50+).

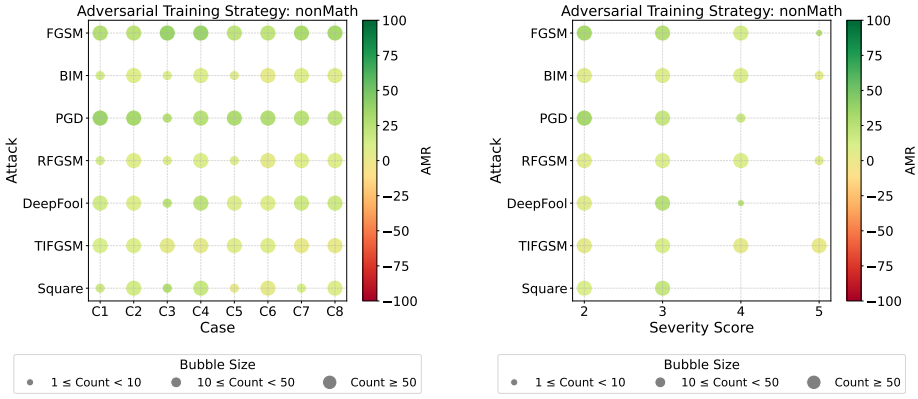
5.6 Attacks vs. Adversarial Training

An ideal defense would maintain strong protection regardless of an attack's severity. However, adversarial training often involves trade-offs, where optimizing for one threat type can weaken resilience to others. To explore this, we analyze two representative training strategies across varying attacks: *adaptive*, a strong defense particularly effective against high-severity attacks (Fig. 5), and *Non-Math*, a simpler, model-agnostic approach (Fig. 6). A detailed overview of the specific AMR values is provided in Appendix A.2. Examining their behavior across DUMB scenarios and attack severities reveals clear patterns. *Adaptive* training offers strong resilience against high-severity attacks, often achieving AMR values above 60% for severities 4 and 5. However, its performance drops against lower severities, where most negative AMR values occur. Conversely, *Non-Math* achieves more consistent but modest positive AMR across lower severities, though it struggles against stronger attacks. Neither method achieves uniform robustness: *adaptive* favors severe threats, while *Non-Math* provides broader but shallower protection. This highlights a key trade-off for practitioners: optimizing for strong attacks may expose vulnerabilities to weaker but more frequent perturbations.



(a) DUMB cases analysis. (b) Attack severity score analysis.

Fig. 5: *Adaptive* training strategy AMR under different attacks.



(a) DUMB cases analysis. (b) Attack severity score analysis.

Fig. 6: *Non-Math* training strategy AMR under different attacks.

5.7 Adversarial Training Failiures

Understanding when adversarial training harms rather than helps is critical for practitioners considering its adoption. A negative AMR signals that the model has become more vulnerable to specific attacks than a baseline trained only on clean data. Our evaluation shows that adversarial training is not universally beneficial: 20.53% of all assessments resulted in a negative AMR, highlighting a notable risk of performance degradation. When failures occur, the average negative AMR is $-35.89\% \pm 34.06$, indicating that the loss in robustness can be substantial. As shown in Figure 7, while most negative outcomes are moderate, practitioners must be aware of a long tail of severe degradations.

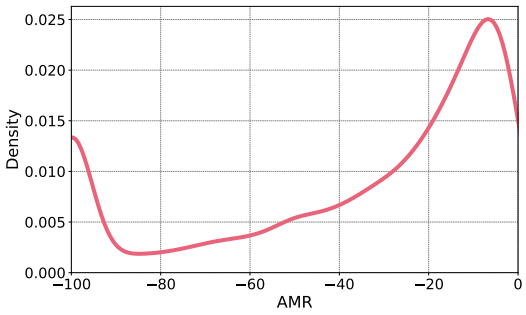


Fig. 7: Negative AMR distribution

To better understand where adversarial training most often fails, we analyze the distribution of negative AMR across different experimental dimensions (Table 3). Failures are heavily concentrated in DUMB scenarios involving a significant mismatch between attacker and victim models (Table 3a). Notably, `C4` and `C8` (model and data mismatches) showed the highest rates of negative AMR, at 32.00% and 28.70%, respectively, while white-box settings (`C1`) exhibited minimal failure (1.61%). Different training strategies also varied in robustness (Table 3b): simple perturbation methods like *Salt & Pepper* and *FGSM* led to frequent but less severe degradations, while the *adaptive* strategy, despite strong overall performance, suffered the most severe failures when they did occur (-52.56% on average). Attack type also plays a key role (Table 3c): *DeepFool* and *Square* attacks triggered failures most frequently, with *DeepFool* causing the steepest average drop in robustness (-53.73%). Finally, failure rates were highly skewed toward weaker attacks (Table 3d), with 72.53% of negative AMR instances occurring against attacks initially classified as Severity 1; stronger attacks rarely caused adversarial training to backfire. These results highlight that adversarial training, while powerful in some settings, is highly sensitive to mismatches, training strategies, attack types, and the initial strength of adversarial perturbations.

6 Conclusions

This paper examined the effectiveness of various adversarial training strategies across the scenarios defined by the DUMB framework. Based on 130k evaluations across all folds of our updated and adapted attacker model, we provide the following answers to our research questions.

- **A1:** Attack strategies designed for transferability (e.g., TIFGSM) pose a greater threat to the overall DUMB population, though non-mathematical, model-agnostic methods also demonstrate significant ASRs.
- **A2:** When the threat model is unknown, strategies such as *adaptive* and *curriculum* generally yield the highest AMRs. However, adversarial training can slightly degrade performance on lower-impact attacks in sub-optimal tasks.

Table 3: Values and distributions of negative AMR across experimental dimensions. "`% Neg.`" indicates, for each dimension, the percentage of negative samples relative to the total number of negative samples across all dimensions., "`Avg.`" shows the mean AMR values and standard deviation, "`Med.`" indicates the median AMR.

(a) By DUMB case.

Case	% Neg.	Avg.	Med.
C1	1.61%	−35.45±35.85	−20.00
C2	7.45%	−32.23±36.34	−13.89
C3	10.84%	−44.12±35.19	−33.33
C4	32.00%	−42.75±35.18	−31.25
C5	2.17%	−32.88±35.10	−15.79
C6	7.44%	−28.40±33.32	−12.20
C7	9.79%	−29.97±30.01	−18.18
C8	28.70%	−30.28±30.87	−17.24

(b) By adversarial training strategy.

Train.	% Neg.	Avg.	Med.
FGSM	14.70%	−34.40±33.41	−20.69
PGD	9.11%	−35.92±31.74	−25.00
Ada.	12.61%	−52.56±36.15	−45.83
Cur.	7.41%	−43.09±34.35	−32.58
Ens.	10.01%	−42.70±34.77	−31.40
Gau.	11.81%	−27.55±31.87	−13.04
N.M.	12.23%	−30.43±33.06	−15.29
S&P	14.01%	−21.80±25.82	−11.54
Sur.	8.11%	−42.31±34.69	−30.71

(c) By attack.

Attack	% Neg.	Avg.	Med.
BIM	13.19%	−30.23±30.65	−17.24
DeepFool	24.30%	−53.73±38.02	−47.37
FGSM	5.96%	−22.38±24.88	−12.31
PGD	9.41%	−30.66±29.86	−19.36
RFGSM	13.48%	−30.97±31.39	−17.65
Square	22.39%	−40.41±32.41	−30.36
TIFGSM	11.27%	−12.44±16.69	−7.25

(d) By attack severity score.

Sev.	% Neg.	Avg.	Med.
1	72.53%	−43.13±35.13	−30.77
2	19.77%	−21.30±23.59	−12.05
3	4.77%	−5.87±5.91	−4.00
4	1.50%	−4.93±4.78	−3.29
5	1.43%	−2.72±3.08	−1.29

A3: Adversarial training proves most effective when the source and target models align. This also holds for mismatched datasets, but only when the original task is sub-optimal in baseline evaluation. Performance can degrade in scenarios with multiple mismatches, particularly when attacks have lower ASRs.

A4: The most effective adversarial training strategies generalize across different target attacks, though their success varies depending on the attacker's knowledge. However, specific attacks (e.g., *DeepFool* and *Square*) can negatively impact adversarially trained models, highlighting the need for focused attention on these cases.

A5: Adversarial training is less effective against lower severity attacks, indicating that it should be employed primarily when facing significantly impactful threats.

Future Works. Future research could develop advanced adversarial training strategies to overcome weaknesses against low-severity attacks. Exploring hybrid defenses that combine model-agnostic and transfer-optimized methods may further enhance robustness across diverse scenarios. Expanding evaluations to broader threat models and real-world datasets would improve understanding of defense generalizability. Finally, a deeper study of the trade-offs between attack severity and training strategies could guide the design of defenses tailored to specific applications.

References

1. Alecci, M., Conti, M., Marchiori, F., Martinelli, L., Pajola, L.: Your attack is too dumb: Formalizing attacker scenarios for adversarial transferability. In:

Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses. p. 315–329. RAID '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3607199.3607227, https://doi.org/10.1145/3607199.3607227
2. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European conference on computer vision. pp. 484–501. Springer (2020)
3. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. Advances in Neural Information Processing Systems **33**, 16048–16059 (2020)
4. Cai, Q.Z., Du, M., Liu, C., Song, D.: Curriculum adversarial training. arXiv preprint arXiv:1805.04807 (2018)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
6. Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., Roli, F.: Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: 28th USENIX security symposium (USENIX security 19). pp. 321–338 (2019)
7. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4312–4321 (2019)
8. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1625–1634 (2018)
9. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Synthetic data augmentation using gan for improved liver lesion classification. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). pp. 289–293. IEEE (2018)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
11. Gröndahl, T., Pajola, L., Juuti, M., Conti, M., Asokan, N.: All you need is" love" evading hate speech detection. In: Proceedings of the 11th ACM workshop on artificial intelligence and security. pp. 2–12 (2018)
12. Grosse, K., Bieringer, L., Besold, T.R., Biggio, B., Krombholz, K.: Machine learning security in industry: A quantitative survey. IEEE Transactions on Information Forensics and Security **18**, 1749–1762 (2023)
13. Gu, J., Jia, X., de Jorge, P., Yu, W., Liu, X., Ma, A., Xun, Y., Hu, A., Khakzar, A., Li, Z., et al.: A survey on transferability of adversarial examples across deep neural networks. arXiv preprint arXiv:2310.17626 (2023)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Krizhevsky, A.: One weird trick for parallelizing convolutional neural networks. arXiv preprint arXiv:1404.5997 (2014)
16. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018)
17. Ling, X., Wu, L., Zhang, J., Qu, Z., Deng, W., Chen, X., Qian, Y., Wu, C., Ji, S., Luo, T., et al.: Adversarial attacks against windows pe malware detection: A survey of the state-of-the-art. Computers & Security **128**, 103134 (2023)
18. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
19. Marchiori, F., Conti, M.: Canederli: On the impact of adversarial training and transferability on can intrusion detection systems. In: Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning. pp. 8–13 (2024)

20. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
21. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 acm sigsac conference on computer and communications security. pp. 1528–1540 (2016)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
23. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
24. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152 (2018)
25. Wang, X., He, X., Wang, J., He, K.: Admix: Enhancing the transferability of adversarial attacks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16158–16167 (2021)
26. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: International conference on learning representations (2019)
27. Yu, W., Gu, J., Li, Z., Torr, P.: Reliable evaluation of adversarial transferability. arXiv preprint arXiv:2306.08565 (2023)
28. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International conference on machine learning. pp. 7472–7482. PMLR (2019)

A Additional Results

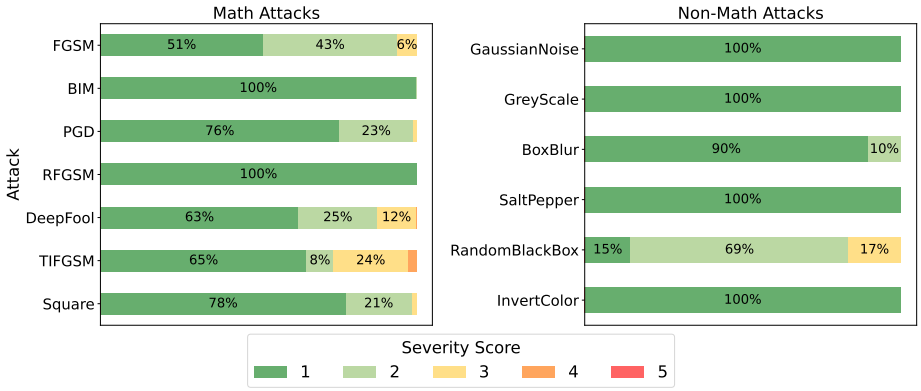
We now provide more details on the results shown in Section A.

A.1 RQ1

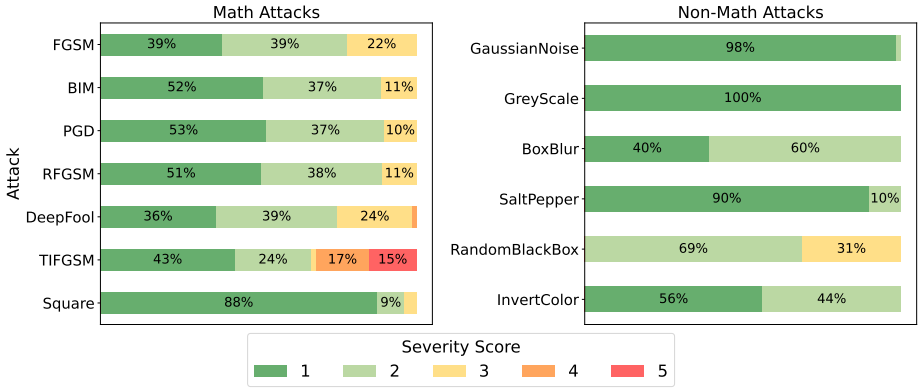
While Section 5.3 provides a cross-task overview of attack severity scores, Fig. 8 breaks them down by individual task. As noted in the original DUMB paper, attacks tend to become more effective as tasks grow more challenging. Specifically, the more straightforward task (B&M) shows no high-severity mathematical attacks, whereas the most complex task (M&W) features many. A similar trend appears for non-mathematical attacks, although their overall ASR remains generally lower.

A.2 RQ4

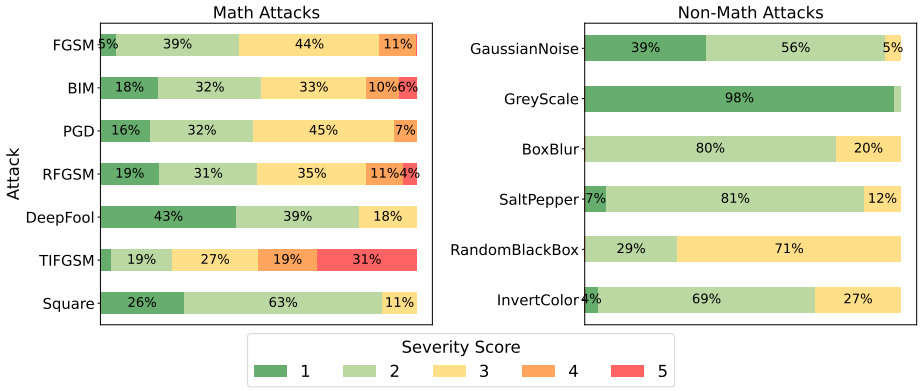
In Fig. 9 and Fig. 10, we extend the analysis from Section 5.6 by including the exact AMR values for each case.



(a) B&M.

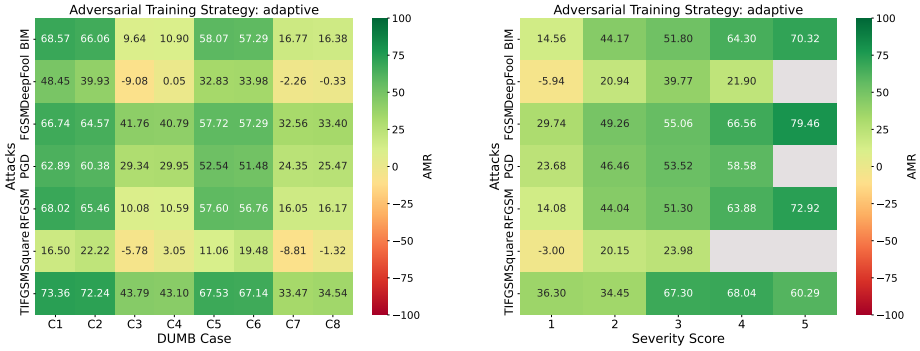


(b) C&D.



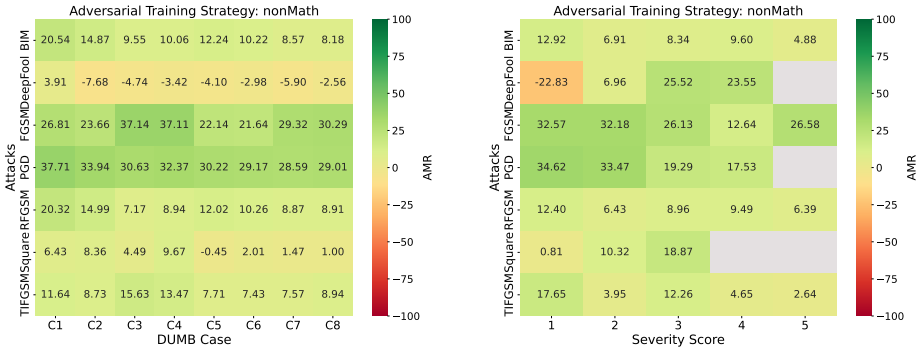
(c) M&W.

Fig. 8: Severity score distribution of each attack on each task.



(a) DUMB cases analysis. (b) Attack severity score analysis.

Fig. 9: *Adaptive* training strategy AMR under different attacks.



(a) DUMB cases analysis. (b) Attack severity score analysis.

Fig. 10: *Non-Math* training strategy AMR under different attacks.