Evaluating the Evaluators: Trust in Adversarial Robustness Tests

Antonio Emanuele Cinà[1,*,†], Maura Pintor[2], Luca Demetrio[1], Ambra Demontis[2], Battista Biggio[2] and Fabio Roli[1]

[1]DIBRIS - Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa
[2]Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice

Abstract

Despite significant progress in designing powerful adversarial evasion attacks for robustness verification, the evaluation of these methods often remains inconsistent and unreliable. Many assessments rely on mismatched models, unverified implementations, and uneven computational budgets, which can lead to biased results and a false sense of security. Consequently, robustness claims built on such flawed testing protocols may be misleading and give a false sense of security. As a concrete step toward improving evaluation reliability, we present AttackBench, a benchmark framework developed to assess the effectiveness of gradient-based attacks under standardized and reproducible conditions. AttackBench serves as an evaluation tool that ranks existing attack implementations based on a novel *optimality* metric, which enables researchers and practitioners to identify the most reliable and effective attack for use in subsequent robustness evaluations. The framework enforces consistent testing conditions and enables continuous updates, making it a reliable foundation for robustness verification.

Keywords

Adversarial Robustness, Robustness Evaluation, Adversarial Examples, Security Benchmarking, ML Security, Trustworthy ML, Machine Learning, Artificial Intelligence

1. Introduction

In recent years, the growing importance of adversarial robustness has led to the development of numerous evasion attacks [1, 2] aimed at crafting adversarial examples with increasing precision and efficiency [3, 4, 5, 6, 7, 8]. These attacks are essential tools to assess how well a model can resist against worst-case perturbations from external malicious users. As a result, they have become central to evaluating the robustness of machine learning systems, particularly in light of emerging regulatory frameworks (e.g., European AI Act [9]), which introduce strict cybersecurity and robustness requirements for high-risk AI systems. However, while evasion attack algorithms have advanced rapidly, the methods used to evaluate them have not kept pace in terms of rigor or consistency. Their evaluations often suffer from methodological flaws that undermine their reliability. Specifically, we identify three recurring and critical issues: (i) evaluations rely on inconsistent choices of target models and metrics, ranging from fixed-budget success rates [10] to median perturbation sizes [11, 6], which makes cross-paper comparisons unreliable; (ii) attack implementations in public libraries are frequently re-written without validation against the original code, leading to bugs or silent performance degradation [12, 13]; and (iii) Computational budgets are inconsistently enforced—for example, some attacks exploit internal

Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy
*Corresponding author.
†These authors contributed equally.
antonio.cina@unige.it (A. E. Cinà); maura.pintor@unica.it (M. Pintor); luca.demetrio@unige.it (L. Demetrio); ambra.demontis@unica.it (A. Demontis); antonio.cina@unige.it (B. Biggio); fabio.roli@unige.it (F. Roli)
https://cinofix.github.io (A. E. Cinà); https://maurapintor.github.io (M. Pintor); https://zangobot.github.io (L. Demetrio); https://web.unica.it/unica/page/it/ambra_demontis (A. Demontis); https://battistabiggio.github.io (B. Biggio); https://www.saiferlab.ai/people/fabioroli (F. Roli)
0000-0003-3807-6417 (A. E. Cinà); 0000-0003-3287-7352 (M. Pintor); 0000-0001-5104-1476 (L. Demetrio); 0000-0001-9318-6913 (A. Demontis); 0000-0001-7752-509X (B. Biggio); 0000-0003-4103-9190 (F. Roli)

restarts [14] or perform additional hyperparameter tuning [3, 4], which gives an unfair advantage to more resource-intensive methods.

Together, these inconsistencies introduce variance that can severely distort robustness assessments, hinder reproducibility, and create a false sense of security. This leads us to a central and urgent question:

To what extent can we trust the evaluation tests used to certify adversarial robustness?

If the tools used to evaluate ML systems are flawed or ineffective, then any robustness guarantees or certification derived from them may be invalid, potentially exposing users to real-world vulnerabilities.

As a concrete step toward addressing the unreliability of current robustness evaluations, we present AttackBench, a benchmark framework developed to systematically assess the effectiveness and efficiency of gradient-based evasion attacks. AttackBench establishes a standardized and impartial evaluation protocol that enables the identification of attack implementations most capable of revealing a model's true worst-case vulnerabilities under adversarial conditions. In this context, reliability refers to an attack's ability to consistently find adversarial perturbations require minimal distortion to successfully mislead the model while respecting a constrained query budget. To support this goal, AttackBench introduces a novel *optimality* metric, which measures how closely each attack approximates the best empirical solution across a diverse set of models and perturbation budgets. Lastly, based on this metric, AttackBench ranks attack implementations according to their effectiveness and efficiency, providing a principled comparison across different threat models. The results are published on a continuously updated online leaderboard[1], helping researchers and practitioners select the most reliable and effective attack strategy when evaluating the adversarial robustness of ML models.

2. Evasion Attacks

Evasion attacks involve manipulating input data at test time to induce misclassification. Examples include modifying malware code to evade detection (i.e., to be misclassified as legitimate) and generating adversarial examples in computer vision—images that appear unchanged to humans but deceive deep learning models [15, 16]. Formally, let $\mathbf{x} \in [0,1]^d$ be an input with true label $y \in \{1, ..., C\}$, and let $f(\mathbf{x}, \boldsymbol{\theta})$ denote the prediction of a trained model with parameters $\boldsymbol{\theta}$. These attacks typically aim to find a perturbation $\boldsymbol{\delta}$ such that the perturbed input $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$ leads to misclassification, while remaining within a bounded perturbation norm and valid input space. This objective can be formalized as the following constrained optimization problem:

$$\underset{\boldsymbol{\delta}}{\text{minimize}} \quad (L(\mathbf{x} + \boldsymbol{\delta}, y; \boldsymbol{\theta}),\ \|\boldsymbol{\delta}\|_p) \tag{1}$$

$$\text{subject to} \quad \mathbf{x} + \boldsymbol{\delta} \in [0,1]^d\,, \tag{2}$$

where L is a loss function that penalizes correct classification. Popular choices include the negative cross-entropy, the difference of logits [3], and the difference of logits ratio [17]. The perturbation size is typically constrained under ℓ_p norms (*e.g.*, ℓ_0, ℓ_1, ℓ_2, ℓ_∞), reflecting different adversarial threat models.

This bi-objective formulation reflects a trade-off between misclassification confidence and minimal perturbation. Accordingly, evasion attacks fall into two families: fixed-budget attacks aim to maximize misclassification within a given perturbation bound [18], and minimum-norm attacks seek the smallest perturbation that causes misclassification [2, 11].

2.1. Evaluation Inconsistencies of Robustness

Despite the vast number of adversarial attacks developed, each claiming improved performance over its predecessors, their evaluation has often lacked standardization across three critical dimensions: (i) the choice of models and evaluation metrics, (ii) the correctness and consistency of attack implementations, and (iii) the fairness of computational budgets. With respect to the first dimension, attacks are frequently

[1] https://attackbench.github.io

evaluated on different models and datasets using incompatible success criteria—such as the attack success rate at a fixed ℓ_p budget [10] or the median perturbation size [11, 6], which hampers meaningful comparisons. For instance, the effectiveness of attacks is commonly measured via the Attack Success Rate (ASR) under a perturbation budget ϵ, formally defined as:

$$\mathrm{ASR}_a(\varepsilon) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathbb{I}(f(\mathbf{x},\boldsymbol{\theta}) \neq y \quad \wedge \quad \|\mathbf{x}_{\mathrm{adv}} - \mathbf{x}\|_p \leq \varepsilon)\,. \tag{3}$$

This metric captures the proportion of input samples in dataset $\mathcal{D}$ for which the attack successfully induces misclassification (i.e., $f(\mathbf{x},\boldsymbol{\theta}) \neq y$) within the allowed norm constraint (i.e., $\|\mathbf{x}_{\mathrm{adv}} - \mathbf{x}\|_p \leq \varepsilon$). However, ASR is highly sensitive to the choice of ε; an attack may perform well at one value of ϵ but poorly at others, limiting the generality of the conclusions drawn. To overcome the limitations of pointwise evaluation metrics like ASR, robustness evaluation curves [19] are often used (red curve in Figure 1 (2)). These curves show the model's robust accuracy as a function of the perturbation budget ε. These curves capture the trade-off between attack strength and the model's resilience over a continuous range of perturbation magnitudes, offering a richer picture of performance than single-point estimates. A lower area under the robustness evaluation curve means the attack is more effective, as it reduces the model's accuracy more quickly. However, this metric depends on the model's starting (clean) accuracy, so it can't be fairly compared across models with different initial performance. Concerning the second dimension, many attacks are re-implemented in public libraries without proper validation against the original code, often leading to performance degradation or the introduction of subtle bugs [12].

Lastly, regarding the third dimension, attacks differ significantly in their computational demands. Some rely on internal restarts [14], hyperparameter searches [3, 4], or repeated query evaluations, which can unfairly advantage them in settings without constraints on time or resources.

As a result of all these inconsistencies, researchers and practitioners may unknowingly draw conclusions from flawed comparisons, and thus deploy models with a false sense of security. For example, a model certified as robust under suboptimal evaluation attack may still be easily fooled in practice with more advanced attacks, exposing users and stakeholders to unacceptable risks.

3. The AttackBench Framework

To support the choice of a reliable attack to assess adversarial robustness, we rely on AttackBench, a benchmark framework specifically designed to test and compare the effectiveness of gradient-based attacks under consistent, fair, and reproducible conditions. Developed in prior work [20], AttackBench offers a structured and extensible platform to assess whether robustness evaluation methods themselves are reliable, i.e., whether they are close to producing the optimal (i.e., smallest) possible adversarial perturbations within a fixed query budget. AttackBench serves as a framework where attacks are evaluated against a common set of models (the *model zoo*) and datasets, using a fixed query budget that counts both forward and backward passes. At the core of AttackBench is the notion of *optimality*. Instead of measuring only whether an attack succeeds at a certain perturbation size ε, AttackBench evaluates how close each attack comes to an empirical best solution, derived by ensembling the results of all tested methods, for ε. Specifically, for each attack, AttackBench evaluate their *local optimality* score, which reflects the quality of an attack on a specific model, and the *global optimality* score, which averages this performance across a diverse set of models. Subsequently, AttackBench utilizes these scores to rank attacks, fostering the identification of those that are both reliable and efficient. Lastly, a key feature of AttackBench is its ability to support continuous updates, enabling an evolving leaderboard and encouraging ongoing contributions from the research community.

3.1. AttackBench Internals

The framework is organized into five modular stages, each designed to minimize experimental bias and promote reproducibility, depicted in Figure 1.

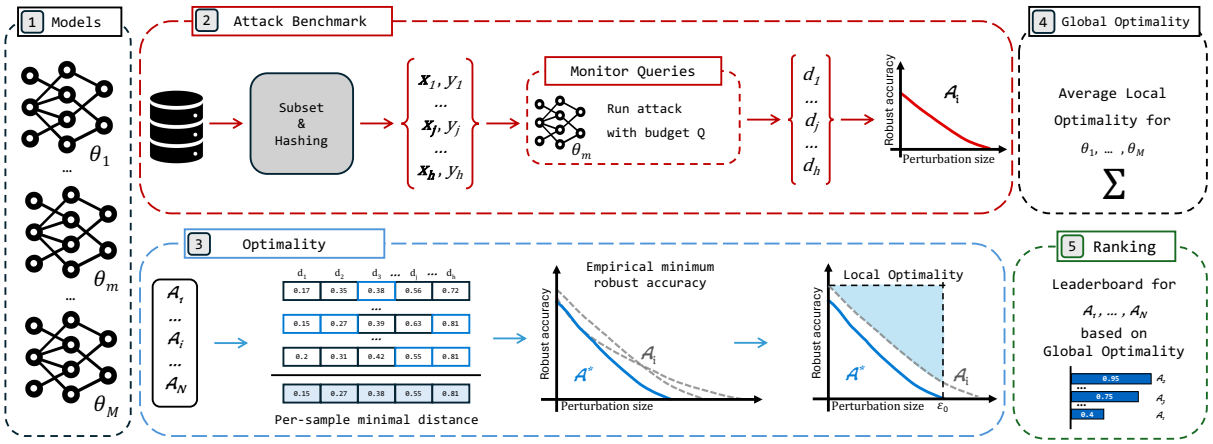

Figure 1: Overview of the five stages of AttackBench.

Stage 1 - Model Zoo. AttackBench begins by defining a diverse and extensible model zoo, which includes both robust and standard models. This ensures that attacks are tested across a range of architectures and robustness levels, preventing overfitting to specific models and enabling generalization of benchmarking results.

Stage 2 - Attack Benchmarking. Attacks are executed against each model in the zoo under strict constraints, producing, for each model-attack configuration, the corresponding robustness evaluation curve (red curve in Figure 1 (2)). AttackBench wraps each model in a query-tracking interface that counts both forward and backward passes, ensuring all attacks are evaluated within the same computational budget. Importantly, it records the best adversarial perturbation found within this budget rather than returning the result from the last iteration—an improvement over many existing libraries.

Stage 3 – Local Optimality. To enable meaningful comparisons between different adversarial attacks, AttackBench introduces the *local optimality* metric—a model-agnostic measure of attack effectiveness. Rather than focusing solely on individual scalar values such as attack success rate at a fixed perturbation size ε, this metric evaluates how close an attack comes to the best-known lower bound on robustness, as estimated by aggregating the results of multiple attacks (blue curve in the Figure 1). Specifically, local optimality is computed from robustness evaluation curves obtained during Stage 2 of each attack. Specifically, AttackBench ensembles all attacks run against a given model and constructs an empirical lower envelope curve representing the best-known attack performance at each perturbation size. The local optimality score for a specific attack is then calculated as the normalized area under the curve between the attack's robustness curve and the lower envelope. Formally, the smaller the area between these two curves, the closer the attack is to the best-known bound, and the higher its optimality score. This value is normalized to lie within $[0, 1]$, where a score of 1 indicates that the attack achieves performance indistinguishable from the ensemble lower bound across the full perturbation range.

Stage 4 - Global Optimality. Since local optimality depends on the specific target model, AttackBench aggregates local scores across all models in the zoo to compute a *global optimality* score. This reflects the average effectiveness of an attack across diverse scenarios, penalizing methods that perform well only on specific architectures. The global score enables ranking attacks in a model-agnostic way.

Stage 5 - Ranking and Leaderboard. Attacks are ranked by their global optimality score and grouped according to the ℓ_p threat model they assume. A key advantage of AttackBench is its incremental update capability: when a new attack is evaluated, only the ensemble statistics and rankings are updated—previous attacks do not need to be re-run. This enables continuous integration and real-time leaderboard updates.

3.2. Main Take-Home Messages

We now summarize the main take-home messages derived from AttackBench [20]. Our benchmarking campaign spans 102 adversarial attacks, evaluated across 2 datasets (CIFAR-10 and ImageNet) and 9 deep neural networks. Lastly, AttackBench offers a comprehensive perspective on attack performance, efficiency, and implementation fidelity across multiple ℓ_p threat models.

Overall Attack Performance. Our large-scale evaluation using AttackBench yields several critical insights into the reliability and practical utility of gradient-based adversarial attacks. First and foremost, our results confirm that a small subset of attacks, i.e., σ-zero, DDN, PDPGD, and APGD, consistently outperform others across both CIFAR-10 and ImageNet benchmarks. These attacks exhibit high optimality scores and produce robustness evaluation curves that closely track the empirical best attack.

Effectiveness-Efficiency Tradeoffs. Another central observation concerns the effectiveness-efficiency tradeoffs. While high optimality scores are desirable, they do not always imply computational efficiency. For instance, although APGD demonstrates strong optimality, it incurs higher computational costs compared to PDPGD, especially on high-dimensional datasets like ImageNet. Conversely, attacks such as VFGA deliver remarkable speed due to early stopping but suffer a notable drop in attack success rate and optimality when scaled to more complex models.

Implementation Variability. Equally important are the discrepancies observed across different implementations of the same attack. Our benchmark reveals significant variations in performance depending on the source library. For example, the APGD attack implemented in the AdvLib library or its original repository achieves optimal or near-optimal results, whereas the same attack in the ART library shows a drastic performance degradation. Specifically, the optimality drops from 90.9% with the AdvLib implementation to 26% with the ART library on CIFAR-10. We highlight that these inconsistencies are often due to subtle but impactful implementation details, such as the number of restarts or the choice of loss function. These findings underscore the necessity for practitioners to carefully audit attack implementations before using them for model evaluation, as seemingly minor differences can dramatically alter the perceived robustness of a model.

Implementation Pitfalls. Finally, our benchmark identifies several recurring pitfalls in existing libraries. Some attacks crash under specific conditions (e.g., initialization issues, label index bugs), while others fail to support crucial features such as per-sample ε evaluations, compromising the usability of attack tools in practice.

4. Conclusion

In summary, AttackBench provides a robust and actionable foundation for evaluating the trustworthiness of adversarial attacks. Our findings stress the importance of algorithmic design, implementation rigor, and careful tuning when benchmarking model robustness. They also caution against naive reliance on off-the-shelf attack implementations without thorough validation, especially in safety-critical or regulatory contexts.

Acknowledgments

This work has been partially supported by project FISA-2023-00128 funded by the MUR program “Fondo italiano per le scienze applicate"; the EU—NGEU National Sustainable Mobility Center (CN00000023), Italian Ministry of University and Research Decree n. 1033—17/06/2022 (Spoke 10); the project Sec4AI4Sec, under the EU’s Horizon Europe Research and Innovation Programme (grant agreement no. 101120393); the project ELSA, under the EU’s Horizon Europe Research and Innovation Programme (grant agreement no. 101070617); and projects SERICS (PE00000014) and FAIR (PE0000013) under the MUR NRRP funded by the EU—NGEU.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly for spelling check. All substantive research content, methodology, analyses, and conclusions were conceived and developed entirely by the authors. The authors take full responsibility for the publication's content.

References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Machine Learning and Knowledge Discovery in Databases (ECML PKDD), volume 8190, 2013.

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

[3] N. Carlini, D. A. Wagner, Towards evaluating the robustness of neural networks, in: IEEE Symposium on Security and Privacy, IEEE Computer Society, 2017, pp. 39–57.

[4] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, C.-J. Hsieh, Ead: elastic-net attacks to deep neural networks via adversarial examples, in: Thirty-second AAAI conference on artificial intelligence, 2018.

[5] J. Rony, L. G. Hafemann, L. S. Oliveira, I. Ben Ayed, R. Sabourin, E. Granger, Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4322–4330.

[6] M. Pintor, F. Roli, W. Brendel, B. Biggio, Fast minimum-norm adversarial attacks through adaptive norm constraints, in: Thirty-fifth Conference on Neural Information Processing Systems, 2021.

[7] A. E. Cinà, F. Villani, M. Pintor, L. Schönherr, B. Biggio, M. Pelillo, σ-zero: Gradient-based optimization of ℓ_0-norm adversarial examples, in: International Conference on Learning Representations, 2025.

[8] Y. Zheng, L. Demetrio, A. E. Cinà, X. Feng, Z. Xia, X. Jiang, A. Demontis, B. Biggio, F. Roli, Hardening rgb-d object recognition systems against adversarial patch attacks, Information Sciences 651 (2023) 119701.

[9] S. Nativi, S. De Nigris, AI Standardisation Landscape: state of play and link to the EC proposal for an AI regulatory framework, ICT Standardisation Observatory and Support Facility in Europe, 2021.

[10] F. Croce, M. Hein, Sparse and imperceivable adversarial attacks, 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 4723–4731.

[11] W. Brendel, J. Rauber, M. Kümmerer, I. Ustyuzhaninov, M. Bethge, Accurate, reliable and fast robustness evaluation, in: Thirty-third Conference on Neural Information Processing Systems, 2020, pp. 12817–12827.

[12] N. Carlini, A critique of the deepsec platform for security analysis of deep learning models, 2019. arXiv:1905.07112.

[13] M. Pintor, L. Demetrio, A. Sotgiu, A. Demontis, N. Carlini, B. Biggio, F. Roli, Indicators of attack failure: Debugging and improving optimization of adversarial examples, in: Advances in Neural Information Processing Systems, 2022.

[14] F. Croce, M. Hein, Minimally distorted adversarial examples with a fast adaptive boundary attack, in: International Conference on Machine Learning, PMLR, 2020, pp. 2196–2205.

[15] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Machine Learning and Knowledge Discovery in Databases - European Conference, 2013.

[16] N. Carlini, D. A. Wagner, Towards evaluating the robustness of neural networks, in: IEEE Symposium on Security and Privacy, SP, IEEE Computer Society, 2017, pp. 39–57.

[17] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: ICML, 2020.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: ICLR, 2018.
[19] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognition 84 (2018) 317–331.
[20] A. E. Cinà, J. Rony, M. Pintor, L. Demetrio, A. Demontis, B. Biggio, I. B. Ayed, F. Roli, Attackbench: Evaluating gradient-based attacks for adversarial examples, Proceedings of the AAAI Conference on Artificial Intelligence (2025) 2600–2608. doi:10.1609/aaai.v39i3.32263.