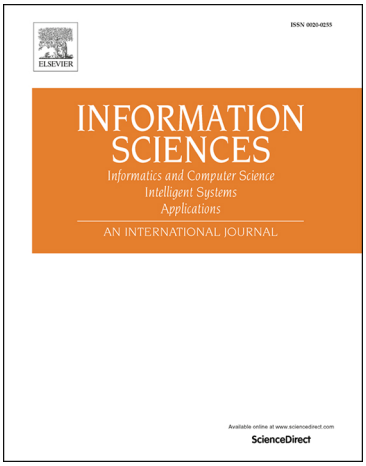

HORNET: Fast and minimal adversarial perturbations

Jiaping Wu, Antonio Emanuele Cinà, Francesco Villani, Zhaoqiang Xia, Luca Demetrio, Luca Oneto, Davide Anguita, Fabio Roli, Xiaoyi Feng

PII: S0020-0255(25)01165-X
DOI: https://doi.org/10.1016/j.ins.2025.123028
Reference: INS 123028

To appear in: *Information Sciences*

Received Date: 21 December 2024
Revised Date: 20 December 2025
Accepted Date: 22 December 2025

Please cite this article as: Wu J, Emanuele Cinà A, Villani F, Xia Z, Demetrio L, Oneto L, Anguita D, Roli F, Feng X, HORNET: Fast and minimal adversarial perturbations, *Information Sciences* (2025), doi: https://doi.org/10.1016/j.ins.2025.123028.

Graphical Abstract

HORNET: Fast and Minimal Adversarial Perturbations

Jiaping Wu, Antonio Emanuele Cinà, Francesco Villani, Zhaoqiang Xia, Luca Demetrio, Luca Oneto, Davide Anguita, Fabio Roli, Xiaoyi Feng

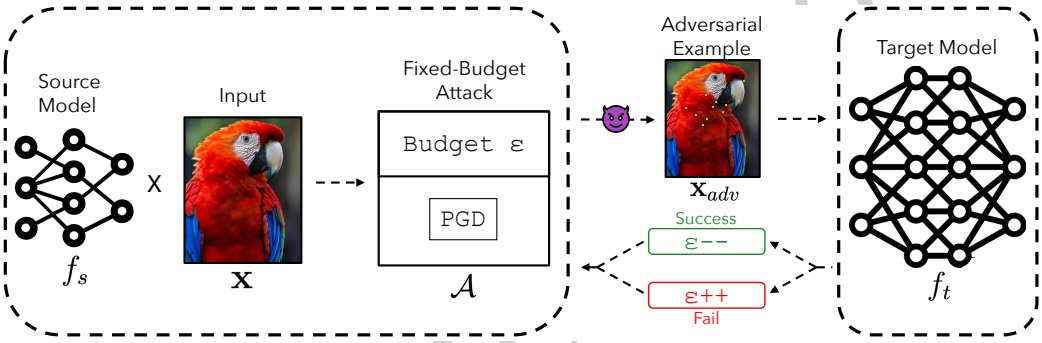

Figure 1: Visualization of HORNET. HORNET starts with a source model f_s and an input $\mathbf{x}$, used by a fixed-budget attack ($\mathcal{A}$) to create an adversarial example ($\mathbf{x}_{adv}$) with budget ε. The adversarial example x_{adv} is then tested on the unknown target model f_t. Based on whether the example successfully fools the target model, the budget ε is iteratively adjusted (decreased on success, increased on failure) and a new adversarial example is generated. This process continues until a minimal successful perturbation is found. This iterative process continues until the maximum query count for the target model is reached or the adversarial example cannot be further optimized.

Highlights

HORNET: Fast and Minimal Adversarial Perturbations

Jiaping Wu, Antonio Emanuele Cinà, Francesco Villani, Zhaoqiang Xia, Luca Demetrio, Luca Oneto, Davide Anguita, Fabio Roli, Xiaoyi Feng

- We propose HORNET, a novel framework for crafting minimum-norm transferable adversarial examples.
- HORNET integrates fixed-budget attacks with an efficient search strategy to minimize perturbation size while maintaining high transfer success rates.
- We assess HORNET across four ℓ_p-norms and leverage 41 attack implementations from widely-used adversarial libraries.
- We conduct extensive experiments comprising 1700 configurations, demonstrating that HORNET significantly outperforms state-of-the-art methods in both transferability and perturbation efficiency.

HORNET: Fast and Minimal Adversarial Perturbations

Jiaping Wu[a,b], Antonio Emanuele Cinà[b,1,*], Francesco Villani[b,1], Zhaoqiang Xia[a], Luca Demetrio[b], Luca Oneto[b], Davide Anguita[b], Fabio Roli[b], Xiaoyi Feng[a]

[a]*School of Electronics and Information, Northwestern Polytechnical University, China*
[b]*Department of Informatics Bioengineering, Robotics, and System Engineering of the University of Genoa, Italy*

Abstract

Fixed-budget attacks aim to generate adversarial examples—carefully crafted inputs designed to induce misclassifications during inference—while adhering to a predefined perturbation budget. These attacks maximize misclassification confidence and benefit from the transferability property, enabling the generated adversarial examples to remain effective even against multiple unknown models. However, to preserve their transferability, such attacks often yield perceptible perturbations, compromising the visual integrity of the adversarial examples. In this paper, we introduce HORNET, an extension of gradient-based fixed-budget attacks designed to minimize the perturbation magnitude of adversarial examples while maintaining their transferability against the target model. HORNET utilizes a distinct source model to craft the adversarial examples and employs a

*Corresponding authors
Email addresses: wujp@mail.nwpu.edu.cn (Jiaping Wu), antonio.cina@unige.it (Antonio Emanuele Cinà), francesco.villani@edu.unige.it (Francesco Villani), zxia@nwpu.edu.cn (Zhaoqiang Xia), luca.demetrio@unige.it (Luca Demetrio), luca.oneto@unige.it (Luca Oneto), davide.anguita@unige.it (Davide Anguita), fabio.roli@unige.it (Fabio Roli), fengxiao@nwpu.edu.cn (Xiaoyi Feng)
[1]Antonio Emanuele Cinà and Francesco Villani contributed equally to this work.

limited number of queries to the unknown target model to further minimize perturbation magnitude. We evaluate HORNET empirically by integrating it with 41 existing attack implementations and testing it against 9 different models, resulting in a total of 1700 unique configurations. Our results demonstrate that HORNET outperforms the state of the art in generating minimally perturbed yet highly transferable adversarial examples across all tested models. Code available at: https://github.com/louiswup/HORNET.

Keywords: Machine learning security, Adversarial examples, Adversarial Perturbation, Transferability, Deep neural networks, Machine learning

2000 MSC: 68T01, 68T45, 68T10, 68T05

1. Introduction

Adversarial examples [1–3] are intentionally crafted perturbations added to test inputs, causing models to produce incorrect outputs. These examples pose significant security threats to critical applications, such as object recognition [4], autonomous driving [5], and face recognition [6], making adversarial robustness a pressing concern in machine learning research. Adversarial attacks can be broadly classified into *minimum-norm* attacks [7, 8], which minimize the perturbation required for misclassification, and *fixed-budget* attacks [9–11], which maximize the confidence of misclassification within a pre-specified perturbation limit. *Minimum-norm* attacks prioritize imperceptibility, generating perturbations close to the original data that lie near the model's decision boundary [12], making them subtle and challenging to detect. In contrast, *fixed-budget* attacks aim to create adversarial examples far from the decision boundary to maximize confidence in misclassification [12], often at the cost of perceptibility.

An intriguing property of adversarial examples is their *transferability*, which refers to the ability of adversarial examples generated for one model (called source model) to also cause misclassification in another target model [13]. This property is typically leveraged in black-box settings, where the attacker has no information about the model architecture, hyperparameters, or even the model logits [2, 13]. In other words, it enables attackers to target closed-source or proprietary models without direct access to their internals, a realistic scenario in domains like commercial face recognition APIs [6] or autonomous driving perception stacks [5]. However, the effectiveness of transferability varies significantly between different types of attacks. *Minimum-norm* attacks tend to struggle with transferability due to their sensitivity to shifts in decision boundaries. Their crafted adversarial examples are highly optimized for the source model but often fail to generalize to unseen target models [14]. On the contrary, *fixed-budget* attacks exhibit higher transfer success rates by crafting examples that remain effective across diverse architectures [14], but this comes at the expense of larger, more noticeable perturbations, creating a trade-off between transferability and imperceptibility. Current benchmarks for evaluating adversarial transferability [15, 16] typically assess attacks under fixed perturbation budgets, providing limited insights into the relationship between transferability and perturbation magnitude. Liu et al. [17] addressed this limitation by proposing a method to find the smallest perturbation budget required for transfer success in *fixed-budget* attacks. However, their approach is suboptimal and restricted to ℓ_∞-norm attacks.

To address these challenges, we introduce HORNET, a query-efficient framework for gradient-based *fixed-budget* attacks that generates transferable adversarial examples with minimal ℓ_p-norm perturbations. HORNET iteratively crafts

examples at varying perturbation budgets and employs a binary search strategy to identify the smallest budget required for successful transfer, given a fixed number of queries. By combining the transferability advantages of *fixed-budget* attacks with efficient minimization, HORNET achieves higher transfer success rates than *minimum-norm* attacks and, compared to Liu et al. [17], significantly reduces perturbation size while accelerating convergence. We evaluate HORNET across four ℓ_p-norms (ℓ_0, ℓ_1, ℓ_2, and ℓ_∞), using 41 attack implementations from widely-used adversarial attack libraries. These attacks are tested under a broad set of transfer scenarios, covering 42 source–target model pairs on CIFAR-10 and 20 pairs on ImageNet, for a total of more than 1700 unique configurations. Our experiments establish a comprehensive benchmark for *minimum-norm* transfer attacks, leveraging the optimality metric [12] that integrates perturbation size and transfer success rate to enable fair comparisons across different attacks.

Our contributions can be summarized as follows:

- We propose HORNET, a novel framework for crafting minimum-norm transferable adversarial examples. HORNET integrates fixed-budget attacks with an efficient search strategy to minimize perturbation size while maintaining high transfer success rates.
- We assess Hornet across four ℓ_p-norms and leverage 41 attack implementations from widely-used adversarial libraries.
- We conduct extensive experiments comprising 1700 configurations, demonstrating that HORNET significantly outperforms state-of-the-art methods in both transferability and perturbation efficiency.

The remainder of this paper is organized as follows. Section 2 provides the nec-

essary preliminaries and background on adversarial attack formulation and taxonomy. In Section 3, we detail the HORNET framework and its algorithmic design. Section 4 presents our experimental setup and results. Section 5 discusses related works on adversarial attacks and transferability, and finally, Section 6 concludes with a summary of our findings and potential directions for future research.

2. Preliminaries

Adversarial attacks are deliberate perturbations added to input data to deceive machine learning models. These attacks exploit model vulnerabilities, where even imperceptible perturbations can lead to incorrect predictions [1, 3]. Let $\mathbf{x} \in X = [0,1]^d$ be a d-dimensional input sample, $y \in Y = \{1, \ldots, l\}$ its associated true label, and $f : X \mapsto Y$ the target model. The goal of an adversarial attack is to find a perturbation $\boldsymbol{\delta}$ such that the resulting adversarial example $\mathbf{x} + \boldsymbol{\delta}$ is both misclassified and stealthy [12]. Their stealthiness is often measured using ℓ_p norms [10], which define the size or budget of the perturbation, constraining the attack to remain imperceptible or within specific bounds. Generally speaking, the optimization process can be written as:

$$\begin{aligned} \boldsymbol{\delta}^* \in \arg\min_{\boldsymbol{\delta}} \quad & (L(\mathbf{x}+\boldsymbol{\delta}, y; f), \|\boldsymbol{\delta}\|_p), \\ \text{subject to} \quad & \mathbf{x}+\boldsymbol{\delta} \in [0,1]^d, \end{aligned} \tag{1}$$

where $L(\mathbf{x}+\boldsymbol{\delta}, y, f)$ is a loss function quantifying the confidence of the model f in classifying the adversarial example $\mathbf{x}+\boldsymbol{\delta}$ as the ground-truth label y.[2] The constraint ensures that the adversarial example remains within the data domain

[2]Minimizing it corresponds to maximizing the misclassification confidence of $\mathbf{x}$ for f.

(i.e., $[0, 1]$ for images). From this optimization problem, two main families of attacks can be identified: *minimum-norm* and *fixed-budget*.

***Minimum-norm* attacks**. *Minimum-norm* attacks focus on minimizing the perturbation size of adversarial examples while ensuring successful misclassification [7, 11, 18–21]. To achieve this, the objective in Equation 1 is modified as follows:

$$\begin{aligned} \boldsymbol{\delta}^* \in \arg\min_{\boldsymbol{\delta}} \quad & \|\boldsymbol{\delta}\|_p \,, \\ \text{subject to} \quad & f(\mathbf{x}+\boldsymbol{\delta}) \neq y \,, \\ & \mathbf{x}+\boldsymbol{\delta} \in [0,1]^d. \end{aligned} \tag{2}$$

These attacks generate examples with minimal perturbations, often producing adversarial inputs that are visually indistinguishable from the original samples, making detection highly challenging [7, 20, 22]. Popular algorithms, such as DeepFool [20], Fast Minimum Norm (FMN) [18], σ-zero [8], and the Carlini-Wagner attack [7], are prominent methods that adopt this strategy. However, the stealthiness of these attacks, achieved by crafting adversarial examples that remain close to the decision boundary of the source model, often results in reduced transferability [14]. This limits their effectiveness when tested against potentially unknown target models.

***Fixed-budget* attacks**. *Fixed-budget* attacks [2, 9, 23, 24] aim to maximize the loss of the target model while constraining the perturbation to lie within a predefined budget, denoted as ε. They can be formalized as follows:

$$\begin{aligned} \boldsymbol{\delta}^* \in \arg\min_{\boldsymbol{\delta}} \quad & L(\mathbf{x}+\boldsymbol{\delta}, y; f), \\ \text{subject to} \quad & \|\boldsymbol{\delta}\|_p \leq \varepsilon, \\ & \mathbf{x}+\boldsymbol{\delta} \in [0,1]^d \end{aligned} \tag{3}$$

where minimizing $L(\mathbf{x} + \boldsymbol{\delta}, y; f)$ corresponds to maximize the misclassification confidence for f. Popular algorithms such as Projected Gradient Descent (PGD) [10], Basic Iterative Method [23], and the APGD [25] follow this approach to iteratively solve the optimization problem. The choice of the perturbation budget ε is critical, as it directly determines the magnitude of the adversarial perturbations. Fixed-budget attacks generate perturbations that are highly effective at misleading the source model and, crucially, unseen target models. This improved transferability, which contrasts with that of *minimum-norm* attacks, stems from the use of larger perturbations. These larger perturbations are more likely to exploit shared vulnerabilities across different model architectures. However, this advantage comes at the expense of stealthiness of the attack, as the resulting adversarial examples may be more easily detectable due to their perceptible distortions [14]. Consequently, *fixed-budget* attacks present a trade-off between transferability and imperceptibility, favoring robustness against diverse models over subtlety.

3. The HORNET framework

3.1. Core Idea and Formalization

Based on the above-mentioned limitations, we propose HORNET: a query-efficient adversarial attack framework. HORNET exploits the high transferability of *fixed-budget* attacks by incorporating an adaptive search strategy to identify the smallest perturbation that still transfers with high confidence. The core idea behind HORNET is to iteratively craft adversarial examples at varying perturbation budgets ε and progressively minimize them until an adversarial example capable of deceiving the target model is obtained. We can formalize the objective function

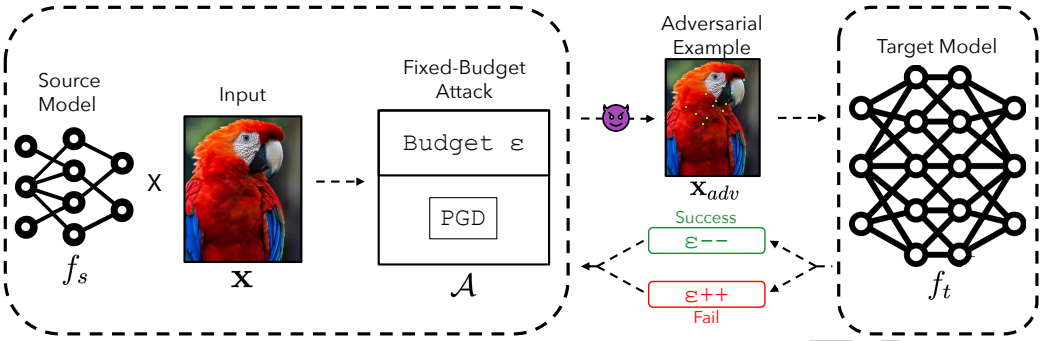

Figure 2: Visualization of HORNET. HORNET starts with a source model f_s and an input $\mathbf{x}$, used by a fixed-budget attack ($\mathcal{A}$) to create an adversarial example ($\mathbf{x}_{adv}$) with budget ε. The adversarial example x_{adv} is then tested on the unknown target model f_t. Based on whether the example successfully fools the target model, the budget ε is iteratively adjusted (decreased on success, increased on failure) and a new adversarial example is generated. This process continues until a minimal successful perturbation is found. This iterative process continues until the maximum query count for the target model is reached or the adversarial example cannot be further optimized.

for HORNET as follows:

$$\begin{aligned} \underset{\varepsilon}{\text{minimize}} \quad & \|\boldsymbol{\delta}^*\|_p, \\ \text{subject to} \quad & \boldsymbol{\delta}^* \in \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, L(\mathbf{x}+\boldsymbol{\delta}, y; f_s), \\ & f_t(\mathbf{x}+\boldsymbol{\delta}^*) \neq y, \\ & \|\boldsymbol{\delta}\|_p \leq \varepsilon,\ \mathbf{x}+\boldsymbol{\delta} \in [0,1]^d, \end{aligned} \tag{4}$$

where f_s, f_t denote source model and target model. The optimization program in Equation 4 formalizes the HORNET's core objective of creating adversarial examples that are both stealthy and transferable across different models. The outer minimization aims to reduce the perturbation size $\|\boldsymbol{\delta}^*\|_p$, similar to *minimum-norm* attacks. The first inner constraint ensures that the perturbation $\boldsymbol{\delta}^*$ is crafted to ensure high-confidence misclassification, which aligns with the principles of

fixed-budget attacks. The second inner constraint enforces that the resulting adversarial example must successfully evade the target model, thereby requiring that $f_t(\mathbf{x} + \boldsymbol{\delta}^*) \neq y$. The last constraint mirrors those seen in Equation 1 and Equation 3, ensuring that the adversarial examples remain within the domain $[0, 1]^d$. Through this formalization, HORNET effectively integrates the advantages of *minimum-norm* attacks and *fixed-budget* strategies, enabling the creation of adversarial examples that not only deceive the source model but also transfer reliably to various target models.

3.2. Overview and Pseudocode

The optimization program in Equation 4 cannot be directly optimized using gradient-based approaches, as the attacker lacks access to the architecture, weights, and logits of the target model. To address this limitation, HORNET adopts a black-box approach consisting of three distinct stages. We illustrate HORNET in Figure 2 and provide a pseudocode of it in Algorithm 1.

Stage 1: Model Zoo Construction. The first stage involves constructing a *model zoo*—a diverse collection of source models f_s accessible to the attacker. These models serve as surrogates for crafting adversarial examples and are deliberately chosen to differ from the unknown target model f_t. As shown in prior work [15, 16] and observed from our results, the diversity in the model zoo increases the likelihood of crafting transferable adversarial examples that generalize across different architectures.

Stage 2: Adversarial Example Generation. In this stage, HORNET employs a *fixed-budget* attack $\mathcal{A}$ to generate high-confidence adversarial examples for a given perturbation size ε (see Line 3 and 11 in Algorithm 1). The goal is to

find an adversarial perturbation $\boldsymbol{\delta}$ that misleads a chosen source model f_s while also evading the target model f_t (first constraint in Equation 4). At this stage, *fixed-budget* attacks ensure that the adversarial examples are crafted with a strong likelihood of transferability to the target model within the perturbation budget ε.

Stage 3: Perturbation Budget Adjustment. In this stage, the HORNET algorithm shrinks the perturbation budget ε of the adversarial examples while preserving their transferability to the target model. Specifically, HORNET uses a two-step search strategy to find the smallest effective perturbation budget ε. It combines a line search with a binary search to ensure the adversarial example, crafted using attack $\mathcal{A}$ on the source model f_s, successfully evades the target model f_t.

Initially, HORNET performs a line search to explore the upper bound of ε. During this phase (lines 2 to 7 in Algorithm 1), the algorithm iterates by doubling the initial value of ε until it finds the first perturbation budget that enables the adversarial example to successfully transfer from the source model f_s to the target model f_t. This exploration ensures that the perturbation budget is sufficiently large to evade the target model at least once.

Following the line search, HORNET enters the binary search phase (lines 9 to 17) to further minimize the perturbation size. The algorithm maintains two bounds, B_{up} and B_{lw}, which define the interval for the binary search. B_{up} is initialized to the last successfully evaded perturbation budget during the line search, representing the maximum value for binary search. B_{lw} is initialized to a lower value derived from the exploration phase, setting the lower boundary of the binary search. During the binary search phase, HORNET evaluates adversarial examples generated with decreasing perturbation budgets (Lines 12-16). If an adversarial example successfully evades the target model f_t, thus satisfying the

hard constraint $f_t(\mathbf{x} + \boldsymbol{\delta}^*) \neq y$ in Equation 4, the algorithm updates the upper bound B_{up} to decrease the current perturbation budget, thereby focusing on smaller perturbations (Line 14). If the adversarial example fails to evade the target model, the algorithm increases the lower bound B_{lw} to explore larger perturbations (Line 16). Lastly, HORNET returns the adversarial example $\mathbf{x}^*_{adv}$, which represents the smallest perturbation that still successfully transfers to the target model f_t. In other words, $\mathbf{x}^*_{adv}$ achieves minimal ℓ_p-norm while preserving transferability, effectively combining the benefits of *fixed-budget* attacks with the efficiency of minimum-norm adversarial examples.

Algorithm 1 HORNET

Input: $\mathbf{x}$: input sample; y: true class label; f_s: source model; f_t: target model; $\mathcal{A}$: attack algorithm; ε: starting perturbation budget; Q: maximum query count for target model; $p \in \{0, 1, 2, \infty\}$: perturbation norm; B_{max}: maximum perturbation budget for the attacker.

Output: $\mathbf{x}^*_{adv}$: *minimum-norm* transferable adversarial example.

$B_{lw} \leftarrow 0$; $\quad B_{up} \leftarrow \varepsilon$

/* Line search: Find first ε value that evades f_t */

while $f_t(\mathcal{A}(\mathbf{x}, y, f_s, \varepsilon)) = y$ ***and*** $Q > 0$ **do**

  $\mathbf{x}_{adv} \leftarrow \mathcal{A}(\mathbf{x}, y, f_s, \varepsilon)$ ▷ Running *fixed-budget* attack $\mathcal{A}$.

  $\varepsilon \leftarrow \min\{B_{max}, 2 \times \varepsilon\}$ ▷ Explore larger ε doubling it.

  $B_{lw} \leftarrow \varepsilon/2$; $\quad B_{up} \leftarrow \varepsilon$ ▷ Update perturbation budget bounds.

  $Q = Q - 1$ ▷ Consume queries to the target.

end

$\mathbf{x}^*_{adv} \leftarrow \mathbf{x}_{adv}$ ▷ Set best adv example so far.

/* Binary search: Find the smallest ε that evades f_t */

for $k = 1$ ***to*** Q **do**

  $\varepsilon \leftarrow \left\lfloor \frac{B_{up}+B_{lw}}{2} \right\rfloor$ ▷ Define current ε.

  $\mathbf{x}_{adv} \leftarrow \mathcal{A}(\mathbf{x}, y, f_s, \varepsilon)$ ▷ Running *fixed-budget* attack $\mathcal{A}$.

  if $f_t(\mathbf{x}_{adv}) \neq y$ ***and*** $\|\mathbf{x}_{adv} - \mathbf{x}\|_p < \|\mathbf{x}^*_{adv} - \mathbf{x}\|_p$ **then**

    $\mathbf{x}^*_{adv} \leftarrow \mathbf{x}_{adv}$ ▷ Set $\mathbf{x}^*_{adv}$ to the smallest adv example.

    $B_{up} \leftarrow \varepsilon$ ▷ Decrease upper bound.

  end

  else $B_{lw} \leftarrow \varepsilon$ ▷ Increase lower bound.

end

return $\mathbf{x}^*_{adv}$

4. Experiments

4.1. Experimental Setup

For the experimental setup, we extend the foundation provided by AttackBench [12], adapting it to address the transferability scenarios studied in this work. Specifically, we evaluate 41 fixed-budget attack implementations across four norms (ℓ_0, ℓ_1, ℓ_2, and ℓ_∞) on multiple source–target model pairs for CIFAR-10 and ImageNet. CIFAR-10 experiments cover 42 transfer settings (6 sources $\times$ 7 targets) and ImageNet 20 settings (5 sources $\times$ 4 targets). In the following paragraphs, we provide a clearer delineation of the experimental setup and the design choices underlying our experiments.

Datasets. We evaluate adversarial attack performance on two widely recognized vision datasets: CIFAR-10 [26] and ImageNet [27]. For CIFAR-10, attacks are tested on the complete test set, while for ImageNet, we randomly select 5000 samples from the validation set. We set the batch sizes to 128 samples for CIFAR-10 and 32 samples for ImageNet.

Model Zoo. To ensure a broad evaluation, we consider a diverse set of models for both CIFAR-10 and ImageNet, covering standard baselines and a wide range of defenses. For CIFAR-10, we evaluate seven models. The baseline is an undefended model from [28] (C1). The remaining models implement robust defenses based on different principles: certified robustness (C2 [29]), confidence-calibrated adversarial training (C3 [30]), activation-based defense inducing gradient obfuscation (C4 [31]), adversarial training with data augmentation (C5 [32]), randomize smoothing (C6 [33]), and adversarial purification (C7 [34]). For ImageNet, we select four models from RobustBench [28], comprising one undefended model

(I1 [35]) and three adversarially trained models: I2 [36], I3 [37], and I4 [38]. Further architectural and implementation details are provided in Appendix B.1.

Transferability Setting. We assess the transferability of adversarial examples generated by HORNET by testing each source-target model pair from our model zoo. In addition to single-source transfer, HORNET supports ensemble-based generation. We adopt a *leave-one-out* ensemble setting, denoted as E, where adversarial examples are crafted by attacking all source models except the target. The attack selects the minimally perturbed example that successfully transfers. It is important to note that source and target models are distinct. Results for equal source and target models in Table 1 serve only as a baseline, highlighting that *minimum-norm* attacks are effective primarily when the attack is crafted and evaluated on the same model.

Attack Setup. We evaluate attacks under the well-studied ℓ_p threat model with $p \in 0, 1, 2, \infty$, considering 41 untargeted attack implementations from established libraries: Adversarial Library [39], Foolbox [40], ART [41], CleverHans [42], Torchattacks [43], and DeepRobust [44]. All attacks are run with default hyperparameters. We also include the transferable attack approach by Liu et al. [17], denoted LS, which extends *fixed-budget* attacks with a suboptimal line search on ε. Following their setup, we run LS only under the ℓ_∞ norm, ranging ε from 0 to 0.3 in steps of 0.02, and retain the adversarial examples with the minimal perturbation that successfully evade the target model. For HORNET, we initialize ε to 100, 10, 1, and $1/255$ for the ℓ_0, ℓ_1, ℓ_2, and ℓ_∞ threat models, respectively. Lastly, HORNET performs the binary search with $Q = 20$ steps, querying the target model up to 20 times to adjust ε based on the attack's success.

Evaluation Metrics. We evaluate HORNET using two key metrics. First, we

report the attack success rate (ASR), defined as the percentage of adversarial examples that successfully fool the target model. Second, we adopt the optimality metric ξ from AttackBench [12], which quantifies how close an attack is to the best possible performance. Specifically, we construct security evaluation curves showing the target model's accuracy as a function of increasing perturbation size. We then follow the methodology from AttackBench to an ensemble of all tested attack results and derive an optimal security evaluation curve, representing the best achievable attack performance. Specifically, in black-box transfer scenarios (e.g., C1 $\rightarrow$ C2), this ensemble only includes attacks sourced from the source model C1, ensuring that the optimality metric reflects the empirical upper bound under the transfer constraint rather than white-box access to the target C2. The optimality of an attack is inversely correlated to the area between the attack's security curve and the optimal curve. Intuitively, $\xi = 1$ indicates that the attack is optimal (i.e., the curves overlap), while a lower positive value of ξ indicates suboptimality, with low values reflecting less optimal attacks. For further details on this metric, refer to AttackBench [12]. All experiments were conducted on a workstation with an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz and an NVIDIA GeForce RTX 4090 GPU with 24GB VRAM.

4.2. Experimental Results

We present the results of HORNET evaluated across more than 1700 configurations. For CIFAR-10, we tested 42 transfer scenarios involving six source models and seven target models, under the ℓ_0, ℓ_1, ℓ_2, and ℓ_∞ threat models. On ImageNet, we examined 20 transfer scenarios with five source and four target models. Due to the higher dimensionality, we focused on the most effective attacks across the same threat models.

Table 1: Transferability comparison of minimum-norm attacks, LS-PGD, and HNT-PGD under the ℓ_∞-norm. We report the Attack Success Rate (ASR) and optimality (ξ) for each attack. The $C1 \rightarrow C1$ scenario serves as a white-box baseline, while the other columns show transfer performance. Notation "C1 $\rightarrow$ C2" indicates that adversarial examples are crafted on source model C1 and evaluated on target model C2.

Attack	C1 $\rightarrow$ C1		C1 $\rightarrow$ C2		C1 $\rightarrow$ C3		C1 $\rightarrow$ C4		C1 $\rightarrow$ C5	
	ASR	ξ	**ASR**	ξ	**ASR**	ξ	**ASR**	ξ	**ASR**	ξ
PDPGD [19]	100.00	**0.98**	0.40	0.00	0.80	0.01	5.30	0.06	0.00	0.00
FAB [11]	100.00	0.97	0.50	0.01	0.50	0.01	5.20	0.06	0.00	0.00
FMN [18]	100.00	0.97	0.50	0.00	0.70	0.01	5.30	0.05	0.00	0.00
TR [21]	100.00	0.66	0.90	0.00	5.60	0.02	5.40	0.04	0.10	0.00
DeepFool [20]	100.00	0.66	0.80	0.00	5.30	0.02	5.40	0.04	0.10	0.00
C&W [7]	100.00	0.67	0.40	0.00	1.60	0.01	5.20	0.04	0.00	0.00
LS-PGD	100.00	0.00	66.80	0.00	99.90	0.00	59.00	0.00	74.90	0.63
HNT-PGD	100.00	0.70	**98.90**	**0.69**	**100.00**	**0.72**	**100.00**	**0.68**	**93.50**	**0.97**

Transferring *Minimum-norm* Attacks. We first demonstrate that HORNET achieves superior performance compared to *minimum-norm* attacks and the line search approach proposed by Liu et al. [17]. To this end, we include in our comparison six state-of-the-art *minimum-norm* attacks and the LS-PGD attack, which applies the line search approach proposed by Liu et al. [17] on the PGD attack to minimize its perturbation budget ε. Similarly, we pair HORNET with the PGD attack (which we refer to as HNT-PGD in the tables). We provide additional results for other fixed-budget attacks in the appendix. We restrict this first analysis to the ℓ_∞-norm threat model as it is the only one supported in [17]. The results of our analysis are summarized in Table 1, where we report the ASR and the optimality metric ξ for each attack in both white-box (source equals target) and black-box (source differs from target) scenarios. These metrics measure the effectiveness of adversarial examples in deceiving a target model and evaluate the attacker's effort required to

generate them. In the white-box scenario, all attacks achieve a perfect ASR of 100%. Among them, *minimum-norm* attacks, such as PDPGD [19] and FAB [11], demonstrate superior optimality, as reflected in their higher ξ values. HNT-PGD delivers competitive optimality in this setting, whereas LS-PGD fails to match its performance. In the black-box setting, however, *minimum-norm* attacks fail to transfer effectively, as evidenced by their significantly lower ASR when tested on different target models. In contrast, HNT-PGD and LS-PGD successfully leverage a small number of black-box queries to the target model, achieving mean black-box ASRs of 98.10% and 75.15%, respectively. In summary, HNT-PGD exhibits outstanding performance in black-box scenarios, setting a new state-of-the-art approach in crafting *minimum-norm* transferable adversarial examples. Additional experiments with alternative *fixed-budget* algorithms results are provided in Appendix B.

Transferability under HORNET for Fixed-Budget Attacks. To understand which adversarial attack algorithms benefit the most from integration with HORNET, we perform a systematic comparison of 41 *fixed-budget* attack implementations. Our evaluation focuses on their effectiveness in terms of transferability and optimality when combined with HORNET. Specifically, we examine their performance across various ℓ_p-norm threat models in a single-source transfer setting, using C1 and I1 as the source models for CIFAR-10 and ImageNet, respectively. To summarize, Table 2 lists the considered attacks and their best-performing implementation libraries, along with the proportion of test cases where each attack paired with HORNET achieves top performance on CIFAR-10 and ImageNet. Our analysis shows that HNT-APGD and HNT-PGD benefit most from integration with HORNET, consistently achieving top transfer success rates and optimality

Table 2: Summary of the best-performing attacks across all evaluated scenarios when combined with HORNET. For each ℓ_p-norm threat model, we report the attack (and the best performing library) that achieves the highest average performance in terms of optimality (ξ). Results are aggregated across CIFAR-10 and ImageNet by computing the percentage of scenarios (i.e., source-target model pairs) in which each attack achieves the best performance compared to the others.

ℓ_p	Attack	Best Library	CIFAR-10 %	ImageNet %	Total %
ℓ_0	HNT-PGD$_0$	Original	100.00	100.00	100.00
ℓ_1	HNT-APGD	Original	91.67	95.00	93.34
	HNT-PGD	Foolbox	8.33	5.00	6.67
	HNT-BIM	Foolbox	0.00	0.00	0.00
	HNT-FGM	ART	0.00	0.00	0.00
ℓ_2	HNT-APGD	Original	30.56	67.50	49.03
	HNT-PGD	CleverHans	63.89	2.50	33.20
	HNT-BIM	ART	0.00	30.00	15.00
	HNT-FGM	DeepRobust	5.56	0.00	2.78
ℓ_∞	HNT-APGD	Original	33.33	85.00	59.17
	HNT-PGD	CleverHans	47.62	0.00	23.81
	HNT-FGM	ART	9.52	15.00	12.26
	HNT-BIM	ART	7.14	0.00	3.57
	HNT-FGSM	Torchattacks	2.38	0.00	1.19

across threat models. These two emerge as the most effective *fixed-budget* attacks under the HORNET framework, clearly outperforming simpler methods like HNT-BIM and HNT-FGM. Detailed results are shown in Table 3 and Table 4, reporting ASR and optimality score ξ for the top three attacks on CIFAR-10 and ImageNet, respectively. On CIFAR-10, no single attack dominates across all norms. HNT-APGD achieves the highest optimality under the ℓ_1 threat model, outperforming others in 91.67% of cases. Meanwhile, HNT-PGD leads under both ℓ_2 and ℓ_∞, showing better synergy with HORNET in these norms. Notably, in the C1 $\rightarrow$ C6 and C1 $\rightarrow$ C7 transfer tasks, where target models apply randomized smoothing and adversarial purification, HNT-PGD slightly outperforms HNT-APGD, suggest-

Table 3: Performance comparison of top *fixed-budget* attacks combined with HORNET on CIFAR-10 under different ℓ_p-norm threat models. We report the Attack Success Rate (ASR) and optimality (ξ) for the best implementation of each attack shown. Notation "C1 $\rightarrow$ C2" denotes transfer from source model C1 to target model C2.

ℓ_p	Attack	C1 $\rightarrow$ C2		C1 $\rightarrow$ C3		C1 $\rightarrow$ C4		C1 $\rightarrow$ C5		C1 $\rightarrow$ C6		C1 $\rightarrow$ C7	
		ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ
ℓ_0	HNT-PGD$_0$	95.7	**1.00**	99.4	**1.00**	100.0	**1.00**	90.6	**1.00**	74.5	**1.00**	61.0	**1.00**
ℓ_1	HNT-APGD	100.0	**0.91**	100.0	**0.98**	100.0	**0.84**	94.8	**0.96**	67.6	0.66	53.6	0.53
	HNT-BIM	96.2	0.81	99.7	0.68	98.5	0.70	94.4	0.92	70.5	0.69	56.5	0.56
	HNT-PGD	95.6	0.81	99.7	0.67	98.3	0.69	94.3	0.91	70.8	**0.70**	57.4	**0.57**
ℓ_2	HNT-APGD	100.0	**0.87**	99.0	**0.83**	100.0	**0.82**	96.0	**0.96**	71.1	0.71	53.6	0.53
	HNT-PGD	99.9	0.82	99.6	0.82	99.9	0.67	95.0	0.94	71.5	**0.71**	57.3	**0.57**
	HNT-BIM	78.3	0.70	96.9	0.74	82.7	0.55	95.7	0.92	45.3	0.40	48.9	0.45
ℓ_∞	HNT-PGD	93.5	**0.91**	99.9	0.93	97.8	**0.82**	93.5	**0.96**	69.5	0.67	53.5	0.51
	HNT-APGD	98.8	0.83	100.0	**0.94**	99.7	0.73	93.9	0.88	70.2	**0.68**	53.9	**0.52**
	HNT-BIM	90.0	0.82	99.9	0.93	96.6	0.73	94.2	0.90	65.0	0.62	52.4	0.50

ing these defenses reduce APGD's effectiveness more. On ImageNet, we observe that APGD stands out as the most effective attack in terms of generating highly transferable *minimum-norm* adversarial examples in the single-source setting. Simpler attacks such as BIM and FGSM, as expected, demonstrate lower performance across both datasets (detailed results are provided in the Appendix), reinforcing the importance of using stronger optimization-based attacks like APGD in combination with HORNET. Further experiments in Appendix B.3, reporting standard deviations across randomly sampled ImageNet subsets, confirm that our findings are statistically robust and quantitatively supported.

Enhancing Transferability with Ensemble Models. We further evaluate HORNET's ability to boost transferability by leveraging multiple source models, denoted as ensemble E in Table 4 and Table 5. Across all ℓ_p-norms, attacks gener-

Table 4: Performance comparison of top *fixed-budget* attacks combined with HORNET on ImageNet under different ℓ_p-norm threat models and different sources C1 and E. We report the Attack Success Rate (ASR) and optimality (ξ) for the best implementation of each attack shown. Notation "I1 $\rightarrow$ I2" denotes transfer from source model I1 to target model I2.

ℓ_p	Attack	I1 $\rightarrow$ I2		I1 $\rightarrow$ I3		I1 $\rightarrow$ I4		E $\rightarrow$ I2		E $\rightarrow$ I3		E $\rightarrow$ I4	
		ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ
ℓ_0	HNT-PGD$_0$	67.0	**1.00**	64.4	**1.00**	41.4	**1.00**	77.6	**1.00**	77.2	**1.00**	48.9	**1.00**
ℓ_1	HNT-APGD	37.1	**0.91**	41.3	**0.93**	10.3	**0.87**	96.1	**0.99**	98.2	**1.00**	54.4	**0.78**
	HNT-BIM	10.2	0.17	8.8	0.12	5.5	0.37	75.6	0.62	79.5	0.62	53.3	0.67
	HNT-PGD	10.1	0.16	8.4	0.11	5.2	0.37	77.0	0.63	80.9	0.63	53.9	0.69
ℓ_2	HNT-APGD	89.3	**0.97**	90.5	**0.97**	74.7	**0.95**	99.7	**0.97**	100.0	**0.98**	99.2	0.90
	HNT-BIM	7.4	0.15	5.8	0.12	3.7	0.11	97.2	0.95	99.3	0.97	93.2	**0.94**
	HNT-PGD	87.6	0.93	88.4	0.93	71.1	0.89	99.4	0.96	99.9	0.97	98.3	0.85
ℓ_∞	HNT-PGD	100.0	**0.98**	100.0	**0.98**	100.0	**0.98**	100.0	0.96	100.0	0.96	100.0	0.95
	HNT-APGD	89.3	0.97	90.5	0.97	74.7	0.95	100.0	**0.98**	100.0	**0.99**	100.0	**0.99**
	HNT-BIM	100.0	0.89	100.0	0.89	100.0	0.85	100.0	0.96	100.0	0.96	100.0	0.95

Table 5: Performance comparison of top *fixed-budget* attacks combined with HORNET on CIFAR-10 under different ℓ_p-norm threat models. We report the Attack Success Rate (ASR) and optimality (ξ) for the best implementation of each attack shown. Notation "E $\rightarrow$ C2" denotes transfer from the *leave-one-out* ensemble of sources E to target model C2.

ℓ_p	Attack	E $\rightarrow$ C2		E $\rightarrow$ C3		E $\rightarrow$ C4		E $\rightarrow$ C5		E $\rightarrow$ C6		E $\rightarrow$ C7	
		ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ	ASR	ξ
ℓ_0	HNT-PGD$_0$	100.0	**1.00**	100.0	**1.00**	100.0	**1.00**	100.0	**1.00**	93.3	**1.00**	82.3	**1.00**
ℓ_1	HNT-APGD	100.0	**0.96**	100.0	**0.98**	100.0	**0.95**	100.0	**0.97**	86.0	**0.86**	78.9	**0.80**
	HNT-PGD	99.7	0.79	100.0	0.94	100.0	0.88	99.9	0.81	83.7	0.85	71.3	0.73
	HNT-BIM	99.9	0.78	100.0	0.93	100.0	0.87	99.5	0.78	81.2	0.82	67.4	0.69
ℓ_2	HNT-APGD	100.0	**0.91**	100.0	**0.88**	100.0	**0.91**	100.0	**0.92**	83.6	0.84	78.5	0.80
	HNT-PGD	100.0	0.90	100.0	0.88	100.0	0.89	100.0	0.91	90.5	**0.91**	80.6	**0.82**
	HNT-BIM	91.1	0.77	100.0	0.80	97.7	0.71	98.1	0.78	47.7	0.48	50.5	0.51
ℓ_∞	HNT-PGD	99.9	0.89	100.0	**0.98**	100.0	0.87	100.0	**0.96**	99.3	**0.94**	98.6	0.87
	HNT-BIM	100.0	**0.93**	100.0	0.97	100.0	**0.93**	100.0	0.88	99.2	0.92	98.1	0.86
	HNT-APGD	100.0	0.91	100.0	0.94	100.0	0.91	100.0	0.88	96.4	0.91	98.4	**0.91**

ated from E consistently achieve higher transfer success rates and better perturbation optimality compared to those crafted from a single model (C1). When target models are less robust, the ensemble mainly fine-tunes the results. However, the benefit becomes more pronounced against harder targets, such as C6 and C7 on CIFAR-10, highlighting the role of the ensemble in generating more generalizable perturbations. For example, under the ℓ_∞-norm on CIFAR-10, ensemble-based HNT-`APGD` reaches 96.4% and 98.4% transfer success on C6 and C7, compared to 70.2% and 53.9% using C1. A similar trend appears on ImageNet, where the ℓ_∞-norm ASR improves from 74.7% (I4) to 100% with E. Overall, HORNET combined with an ensemble of source models provides a further effective strategy for crafting minimum-norm transferable adversarial examples.

Security Evaluation Curves. To complement the effectiveness analysis, we visualize in Figure 3 the security evaluation curves obtained from the best-performing variants of HORNET against the most robust CIFAR-10 models, C6 and C7. These curves are used to compute the optimality score [12]. Each curve shows how the robust accuracy (i.e., the inverse of ASR) decreases as the allowed perturbation size grows. The black line represents the empirical optimal curve, i.e., the best performance observed across all evaluated attacks. An effective attack is expected to produce a steep curve that closely tracks the black line, indicating its ability to significantly reduce robust accuracy with minimal perturbation. In essence, the closer the curve is to the origin and the faster it drops, the higher its optimality and, consequently, the more effective the attack is at breaking the model's robustness under tight perturbation budgets.

HORNET: Query Efficiency and Transferability Trade-off. We analyze the query efficiency of HORNET by varying Q (maximum query count) and observ-

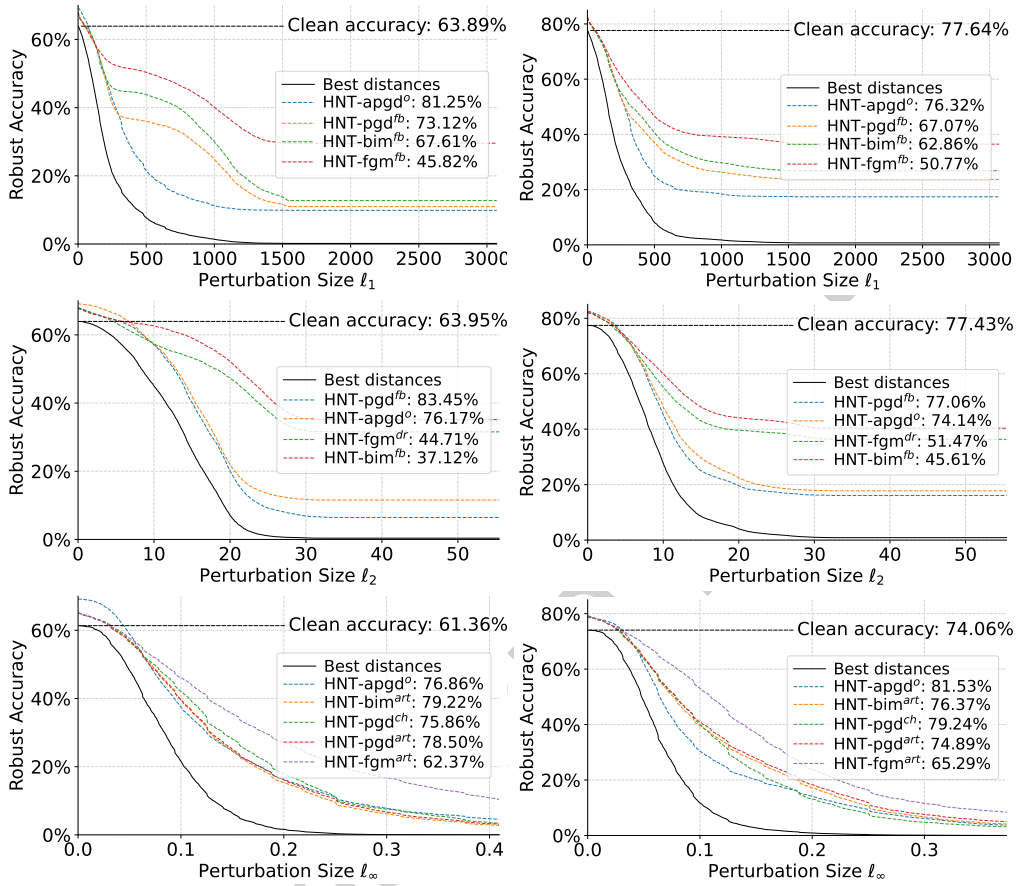

Figure 3: Security evaluation on CIFAR-10 for robust models C6 (left) and C7 (right) under ℓ_1, ℓ_2, and ℓ_∞ threat models (top to bottom).

ing the impact on ASR and optimality. In Figure 4 we report the results on CIFAR-10 under ℓ_1, ℓ_2, and ℓ_∞ norms using the best-performing fixed-budget attack variant, i.e., HNT-APGD. Across all norms, the ASR increases steadily with more queries, while the median perturbation norm consistently decreases. This trend highlights a clear trade-off: more queries lead to higher-quality, lower-norm adversarial examples that are more transferable—but at a higher computational cost. The computational complexity of HORNET indeed scales linearly with the num-

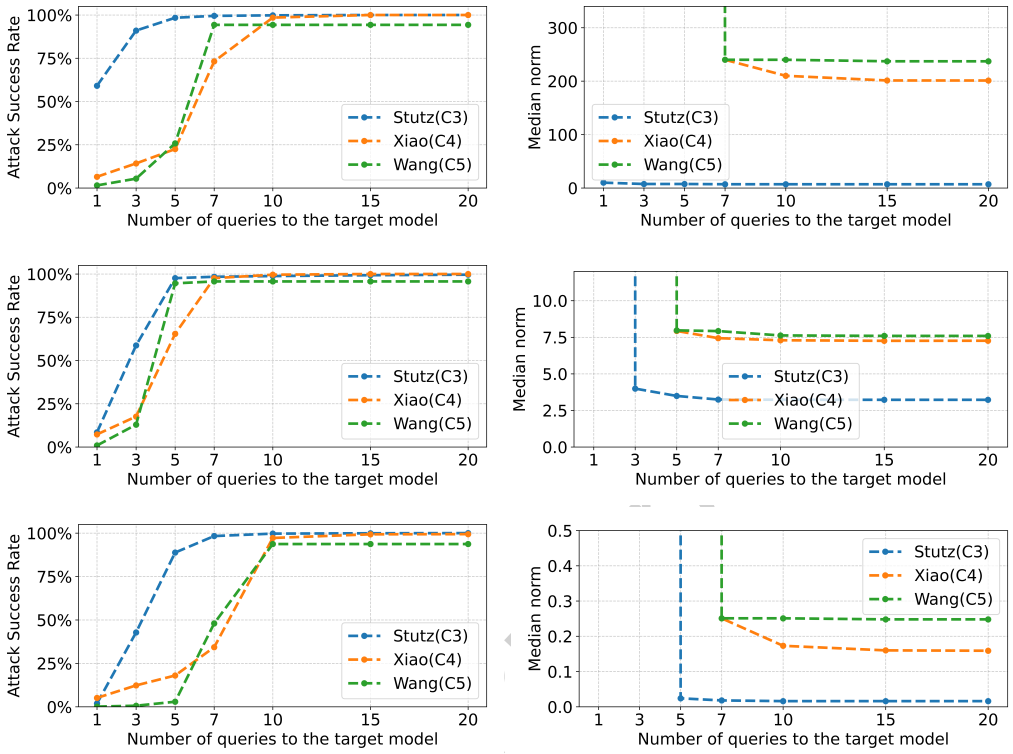

Figure 4: Query efficiency of HNT-APGD on CIFAR-10 for target models C3, C4, and C5. Rows correspond to ℓ_1 (top), ℓ_2 (middle), and ℓ_∞ (bottom) threat models. We report ASR (left) and median perturbation norm (right) versus the number of queries.

ber of queries, as it re-executes the underlying fixed-budget attack at each step. However, this process pays off quickly: already with a budget of 5 or 7 queries, HORNET consistently achieves near-optimal ASR and norm values across multiple targets, as shown in Figure 4. Beyond this point, performance gains saturate, indicating that HORNET is highly efficient in practice. This demonstrates the practical appeal of HORNET, which enables attackers to craft minimum-norm transferable examples with just a few queries, while allowing them to adjust the performance-cost trade-off based on query constraints.

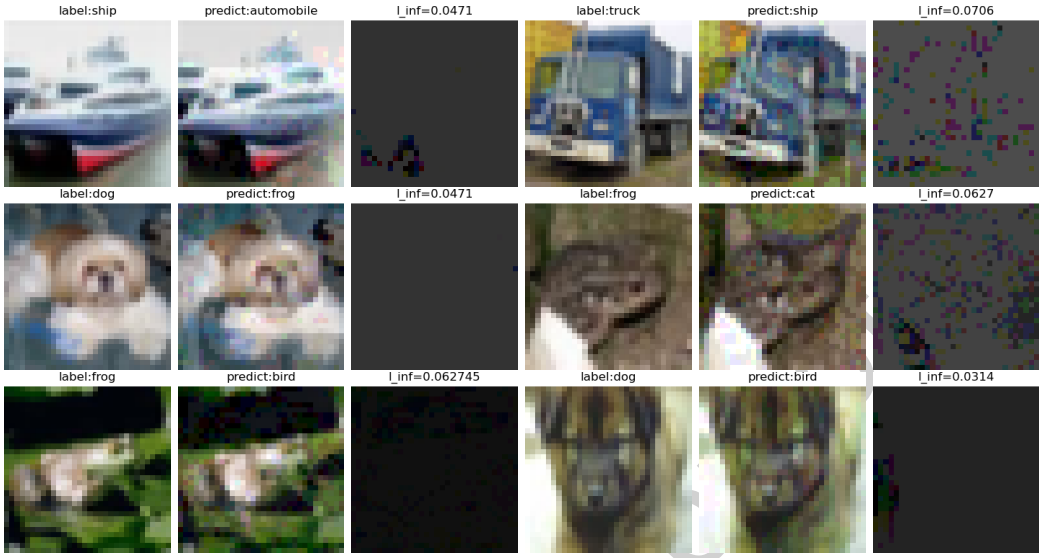

Figure 5: Adversarial examples generated by HNT-PGD on CIFAR-10. Each row shows two samples, for each of which we present the clean image (left), the corresponding adversarial image (middle), and the amplified adversarial perturbation (right). We then report the true labels (left), the adversarial predictions (middle), and the final ℓ_∞ norm of the perturbation (right).

Visual Comparison. Figure 5 shows adversarial examples generated by HNT-PGD on CIFAR-10. The left column contains clean images with ground-truth labels, while the next middle column displays adversarial examples generated ensemble model E and transferred to the target model C1. The right column shows the perturbation of the attack examples. We observe that the adversarial examples generated by HORNET successfully fool the target model while remaining visually indistinguishable from the original images. This confirms the effectiveness of HORNET in producing transferable attacks with minimal perturbation norms. The perturbation maps are visually amplified to make the underlying noise pattern visible; in practice, such perturbations are imperceptible, highlighting the stealthiness of the attack.

Perceptual Comparison. We lastly present in Figure 6 a perceptual analysis of the adversarial examples generated by HORNET and LS-PGD. Specifically, we employ the SSIM (Structural Similarity Index) and LPIPS (Learned Perceptual Image Patch Similarity) metrics to quantify the perceptual distance between adversarial examples and their corresponding clean images. The SSIM metric is computed using an 11×11 sliding window, while LPIPS is evaluated with the AlexNet backbone. As shown in the figure, HORNET achieves higher perceptual similarity to the original samples than LS-PGD, indicating that the perturbations generated by HORNET are less perceptible and better preserve the visual quality of the input images.

Figure 6: Visual comparison of adversarial examples generated by LS-PGD and HNT-PGD on CIFAR-10. The top row shows the original clean images. The middle row displays the adversarial examples from LS-PGD, and the bottom row shows those from HNT-PGD. The SSIM and LPIPS perceptual metrics are reported for each generated image.

5. Related work

There has been significant research into the factors influencing adversarial transferability. Liang et al. [45] proposed two knowledge transfer-based metrics that represent complementary perspectives of transferability. Demontis et al. [14] identified two key factors contributing to attack transferability: the intrinsic adversarial vulnerability of the target model and the complexity of the source model

used to optimize the attack. Zhang et al. [46] argued that transferability is jointly influenced by model smoothness and gradient similarity.

Unlike the aforementioned works, which investigate factors affecting adversarial transferability, our research focuses on developing a novel attack strategy to craft *minimum-norm* transferable adversarial examples.

Most of the literature in this field has concentrated on developing strategies to improve the transferability of *fixed-budget* attacks. Liu et al. [47] proposed ensemble-based approaches to generate transferable adversarial examples by leveraging multiple source models. Xie et al. [48] improved transferability by applying random transformations to input images during the generation process. Zhu et al. [49] introduced the Intrinsic Adversarial Attack, which aligns the adversarial perturbation with the gradient of the joint data distribution to enhance transferability. Wang et al. [50] discovered a negative correlation between transferability and the interactions within adversarial perturbations and proposed a novel loss function to penalize these interactions, improving transferability.

While these works focus solely on maximizing the transferability success rate, they often disregard the stealthiness of the perturbations. In contrast, our work aims to develop attacks that achieve high transferability while maintaining a minimal perturbation budget to ensure stealthiness. Liu et al. [17] were the first to explore this direction, proposing a geometry-aware framework to generate *minimum-norm* transferable adversarial perturbations. However, their method relied on a simple discrete search approach to find the optimal budget and was limited to ℓ_∞ FGSM-based attacks.

In this work, we address these limitations by employing a more efficient search method and conducting extensive experiments on a wide range of white-box *fixed-*

budget attacks. Our approach demonstrates that it is possible to achieve both high transferability and minimal perturbation magnitude, advancing the state-of-the-art in crafting *minimum-norm* transferable adversarial examples.

6. Conclusion and Future Work

We introduced HORNET, a new framework for generating transferable adversarial examples with minimal ℓ_p-norm perturbations. Unlike standard *minimum-norm* attacks, which often fail to transfer reliably in black-box settings, HORNET is designed to enhance transferability while keeping perturbations small. It combines any gradient-based *fixed-budget* attack with an adaptive, query-efficient search strategy that adjusts the perturbation budget on the fly. Through extensive experiments across four ℓ_p threat models, using 41 attack variants from major adversarial libraries, we demonstrated that HORNET consistently improves both ASR and perturbation optimality. Moreover, its query-efficient design allows it to reach near-optimal results with just a few queries to the target model.

We acknowledge several limitations that motivate future work: (i) the transferability of HORNET may decrease when using narrow or misaligned model zoos as source models f_s; (ii) although query-efficient, the overall cost still scales linearly with the maximum query count Q, as the inner attack $\mathcal{A}$ is rerun at each search step; and lastly (iii) HORNET is currently tailored to vision and ℓ_p-based distances. Extending it to other domains requires two adaptations: first, replace the attack A with domain-specific attack algorithms that respect discrete, functional, or temporal constraints (e.g., tokens for NLP, functionality for malware, or temporal consistency for time series); second, replace ℓ_p norms with perceptually or task-appropriate distance measures (e.g., inaudibility metrics for audio, or the

number of system calls for malware).

Acknowledgments

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

References

[1] 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.

[2] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: Y. Bengio, Y. LeCun (Eds.), International Conference on Learning Representations, ICLR, 2015.

[3] 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: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2013, pp. 387–402.

[4] 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.

[5] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634.

[6] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, J. Zhu, Efficient decision-based black-box adversarial attacks on face recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7714–7722.

[7] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: IEEE symposium on security and privacy (sp), 2017, pp. 39–57.

[8] 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, pp. 49108–49120.

[9] F. Croce, M. Hein, Sparse and imperceivable adversarial attacks, in: IEEE/CVF international conference on computer vision, 2019, pp. 4724–4732.

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

[11] 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.

[12] 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, Vol. 39, 2025, pp. 2600–2608. doi:10.1609/aaai.v39i3.32263.

[13] N. Papernot, P. McDaniel, I. Goodfellow, Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, arXiv preprint arXiv:1605.07277 (2016).

[14] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, F. Roli, Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks, in: USENIX security symposium, 2019, pp. 321–338.

[15] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, J. Zhu, Benchmarking adversarial robustness on image classification, in: IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 321–331.

[16] Z. Zhao, H. Zhang, R. Li, R. Sicre, L. Amsaleg, M. Backes, Towards good practices in evaluating transfer adversarial attacks, arXiv preprint arXiv:2211.09565 (2022).

[17] F. Liu, C. Zhang, H. Zhang, Towards transferable adversarial perturbations with minimum norm, in: ICML 2021 Workshop on Adversarial Machine Learning, 2021.

[18] M. Pintor, F. Roli, W. Brendel, B. Biggio, Fast minimum-norm adversarial attacks through adaptive norm constraints, Advances in Neural Information Processing Systems 34 (2021) 20052–20062.

[19] A. Matyasko, L.-P. Chau, Pdpgd: Primal-dual proximal gradient descent adversarial attack, arXiv preprint arXiv:2106.01538 (2021).

[20] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.

[21] Z. Yao, A. Gholami, P. Xu, K. Keutzer, M. W. Mahoney, Trust region based adversarial attack on neural networks, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11350–11359.

[22] F. Villani, I. Maljkovic, D. Lazzaro, A. Sotgiu, A. E. Cinà, F. Roli, Robust image classification with multi-modal large language models, Pattern Recognition Letters (2025). doi:10.1016/J.PATREC.2025.04.022.

[23] A. Kurakin, I. J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial intelligence safety and security, 2018, pp. 99–112.

[24] F. Croce, M. Hein, Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers, in: International Conference on Machine Learning, PMLR, 2021, pp. 2201–2211.

[25] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: International conference on machine learning, PMLR, 2020, pp. 2206–2216.

[26] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255.

[28] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, Robustbench: a standardized adversarial robustness benchmark, arXiv preprint arXiv:2010.09670 (2020).

[29] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, C.-J. Hsieh, Towards stable and efficient training of verifiably robust neural networks, in: International Conference on Learning Representations, 2019.

[30] D. Stutz, M. Hein, B. Schiele, Confidence-calibrated adversarial training: Generalizing to unseen attacks, in: International Conference on Machine Learning, 2020, pp. 9155–9166.

[31] C. Xiao, P. Zhong, C. Zheng, Enhancing adversarial defense by k-winners-take-all, in: International Conference on Learning Representations, 2020.

[32] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, S. Yan, Better diffusion models further improve adversarial training, in: International Conference on Machine Learning, 2023, pp. 36246–36263.

[33] J. Cohen, E. Rosenfeld, Z. Kolter, Certified adversarial robustness via randomized smoothing, in: international conference on machine learning, PMLR, 2019, pp. 1310–1320.

[34] Y. Yang, G. Zhang, D. Katabi, Z. Xu, ME-Net: Towards effective adversarial robustness with matrix estimation, in: International Conference on Machine Learning, 2019.

[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 770–778.

[36] E. Wong, L. Rice, J. Z. Kolter, Fast is better than free: Revisiting adversarial training, in: International Conference on Learning Representations, 2019.

[37] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, A. Madry, Do adversarially robust imagenet models transfer better?, Advances in Neural Information Processing Systems 33 (2020) 3533–3545.

[38] E. Debenedetti, V. Sehwag, P. Mittal, A light recipe to train robust vision transformers, in: IEEE Conference on Secure and Trustworthy Machine Learning, 2023, pp. 225–253.

[39] J. Rony, I. Ben Ayed, Adversarial Library. doi:10.5281/zenodo.5815063.

[40] J. Rauber, W. Brendel, M. Bethge, Foolbox: A python toolbox to benchmark the robustness of machine learning models, arXiv preprint arXiv:1707.04131 (2017).

[41] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, B. Edwards, Adversarial robustness toolbox v1.2.0, CoRR 1807.01069 (2018).

[42] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, R. Long, Technical report on the cleverhans v2.1.0 adversarial examples library, arXiv preprint arXiv:1610.00768 (2018).

[43] H. Kim, Torchattacks: A pytorch repository for adversarial attacks, arXiv preprint arXiv:2010.01950 (2020).

[44] Y. Li, W. Jin, H. Xu, J. Tang, Deeprobust: A pytorch library for adversarial attacks and defenses, arXiv preprint arXiv:2005.06149 (2020).

[45] K. Liang, J. Y. Zhang, B. Wang, Z. Yang, S. Koyejo, B. Li, Uncovering the connections between adversarial transferability and knowledge transferability, in: International Conference on Machine Learning, PMLR, 2021, pp. 6577–6587.

[46] Y. Zhang, S. Hu, L. Y. Zhang, J. Shi, M. Li, X. Liu, H. Jin, Why does little robustness help? a further step towards understanding adversarial transferability, in: IEEE Symposium on Security and Privacy, Vol. 2, 2024.

[47] Y. Liu, X. Chen, C. Liu, D. Song, Delving into transferable adversarial examples and black-box attacks, in: International Conference on Learning Representations, 2016.

[48] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, A. L. Yuille, Improving transferability of adversarial examples with input diversity, in: IEEE/CVF

conference on computer vision and pattern recognition, 2019, pp. 2730–2739.

[49] Y. Zhu, J. Sun, Z. Li, Rethinking adversarial transferability from a data distribution perspective, in: International Conference on Learning Representations, 2021.

[50] X. Wang, J. Ren, S. Lin, X. Zhu, Y. Wang, Q. Zhang, A unified approach to interpreting and boosting adversarial transferability, in: International Conference on Learning Representations, 2020.

Appendix A. *Fixed-budget* and *Minimum-norm* Attacks

In this paper, we consider several well-known *minimum-norm* and *fixed-budget* attacks and different implementations of them (see Table B.6).

Minimum-norm Attacks. For *minimum-norm* attacks, we consider C&W, DeepFool, PDPGD, FAB, FMN, and TR. DeepFool [20], a seminal *minimum-norm* method, generates adversarial examples by iteratively linearizing the decision boundary, achieving minimal perturbations under ℓ_2-norm constraints. The C&W [7] attack is also popular for its ability to minimize perturbations, though it is computationally intensive as it requires internal optimization of hyperparameters. PDPGD [19] iteratively minimizes perturbations under the ℓ_∞-norm, while FAB [11], with its intuitive geometric foundation, quickly yields high-quality results and minimizes the size of perturbations. Trust Region (TR) methods [21] provide an alternative approach by leveraging trust regions to ensure stable convergence, although these methods can introduce higher computational costs. FMN [18] offers a fast and scalable alternative that balances perturbation size with computational efficiency.

Fixed-budget Attacks. As for *fixed-budget* attack we consider FGSM, BIM, PGD, PGD-ℓ_0, and APGD. FGSM [2] is a pioneering gradient-based method that computes adversarial perturbations in a single iteration. BIM [23] and PGD [10] extend FGSM to multi-step iterations, making PGD one of the most widely used and robust gradient-based attack techniques. PGD-ℓ_0 [9] further adapts PGD to the ℓ_0 norm, enabling sparse perturbations. APGD [25] refines the PGD attack by addressing issues related to suboptimal step size selection and shortcomings of the loss function. Additionally, APGD-ℓ_1 [24] extends APGD to operate under the ℓ_1 norm, broadening its applicability to different types of perturbations.

Appendix B. Experiments

Appendix B.1. Model Zoo

To perform a thorough evaluation, we benchmark the adversarial attacks on both standard and robust models. For CIFAR-10, we test seven distinct models, including an undefended WideResNet-28-10 [28] (C1) as baseline. Regarding the robust models, we test a variety of defenses: Zhang et al. [29] (C2), which proposes a certified defense; we select a lightweight CNN model from their publicly available checkpoints. Stutz et al. [30] (C3) offer a confidence calibration approach combined with adversarial training on a ResNet-20. Xiao et al. [31] (C4) introduce a new activation function to improve robustness; we source their pre-trained Sparse-ResNet-18 model. Wang et al. [32] (C5) combine data augmentation with adversarial training, for which we use their WideResNet-28 of width ten equipped with a Swish activation function. It is worth noting that model C4 has been specifically designed to cause gradient obfuscation; we deliberately include this model in the evaluation to assess the efficacy of the attacks under masked gradient conditions. Lastly, we include a robust certified defense via randomized smoothing from Cohen et al. [33] (C6) and an adversarial purification defense from Yang et al. [34] (C7). For the ImageNet dataset, we source pre-trained models from the Robustbench [28] library. We select an undefended pre-trained ResNet-50 model [35](I1) as a baseline; three adversarially trained models, including a ResNet-50 from Wong et al. [36] (I2), a WideResNet-50-2 from Salman et al. [37] (I3), and an XCiT-S12 model from Debenedetti et al. [38] (I4).

Appendix B.2. Transferability of Minimum-norm Adversarial Examples

In the main paper, our minimum-norm experiments (Table 1) focused on PGD to provide a direct methodological comparison between HORNET and a standard Line Search (LS) baseline. To provide a more comprehensive perspective, Table B.7 expands this evaluation. This table includes the additional fixed-budget attacks considered in our study (e.g., APGD) and reports the median L_∞ perturbation norm along with the optimality metric. The median norm illustrates HORNET's advantage in black-box transfer scenarios. For standard minimum-norm attacks like LS-PGD, the median norm in transfer settings (e.g., C1→C3) is often ∞. This occurs because the attack is ineffective and fails to achieve at least a 50% attack success rate (ASR), meaning a median successful perturbation is never found. In contrast, LS-paired and HORNET-paired attacks consistently find adversarial examples, resulting in finite median norms. Furthermore, we can see how HORNET-paired attacks consistently find lower medians and are overall more optimal than the baseline LS. We omit the C2 model from this table. Its low clean accuracy resulted in misleading median norms of 0.0 for most attacks, as they trivially succeeded on samples that were already misclassified. Finally, this expanded table shows that HNT-APGD is highly effective, achieving white-box performance on par with top-performing attacks like FMN, while maintaining superior black-box transferability.

Appendix B.3. Statistical Reliability of HORNET

HORNET does not involve any randomization that could affect outcomes: applying existing attacks or HORNET itself across the full dataset produces identical outputs. The only potential source of variability comes from the specific subsets of data used during evaluation on ImageNet. To quantify this, Table B.8 reports

Table B.6: List of attack implementations considered in HORNET.

ℓ_p	Attacks	Library						
		AdvLib	ART	Cleverhans	DeepRobust	FoolBox	TorchAttacks	Original
ℓ_0	PGD-ℓ_0 [9]							✓
ℓ_1	FGSM [2]		✓	✓		✓		
	BIM [23]		✓			✓		
	PGD [10]		✓			✓		
	APGD-ℓ_1 [24]	✓						✓
ℓ_2	FGSM [2]		✓	✓	✓	✓		
	BIM [23]		✓			✓		
	PGD [10]		✓	✓	✓	✓		
	APGD [25]	✓						✓
ℓ_∞	FGSM [2]		✓	✓	✓	✓	✓	
	BIM [23]		✓			✓		
	PGD [10]		✓	✓	✓	✓	✓	
	APGD [25]	✓						✓

the ASR of HORNET when evaluated on different randomly sampled subsets of ImageNet. The observed deviations are consistently small, confirming that our findings are robust and statistically reliable.

Appendix B.4. Hyperparameters

For the HORNET attacks, such as HNT-`PGD`, we set the maximum perturbation (i.e., the initial upper bound for the HORNET binary search—to the theoretical maximum attack budget permitted by the input data). Here, the maximum refers to the largest possible ℓ_p norm that a perturbation can assume given the dimensions and value range of the input image. Based on these calculations, the resulting maximum perturbations for CIFAR-10 were $\ell_0 = 3072$, $\ell_1 = 3072$, $\ell_2 = 55.426$, and $\ell_\infty = 1.0$, while for ImageNet they were $\ell_0 = 150528$, $\ell_1 = 150528$, $\ell_2 = 387.979$, and $\ell_\infty = 1.0$.

For all attacks considered in our experiments, including minimum-norm and

Table B.7: Transferability comparison of minimum-norm attacks and fixed-budged attacks paired with a line-search strategy(LS) and HORNET (HNT) under the ℓ_∞-norm. We report the Attack Success Rate (ASR), optimality (ξ) and median perturbation ℓ_∞-norm for each attack. The $C1 \rightarrow C1$ scenario serves as a white-box baseline, while the other columns show transfer performance. Notation "C1 $\rightarrow$ C2" indicates that adversarial examples are crafted on source model C1 and evaluated on target model C2.

Attack	C1 $\rightarrow$ C1			C1 $\rightarrow$ C3			C1 $\rightarrow$ C4			C1 $\rightarrow$ C5		
	ASR	ξ	$\tilde{\ell}_\infty$	ASR	ξ	$\tilde{\ell}_\infty$	ASR	ξ	$\tilde{\ell}_\infty$	ASR	ξ	$\tilde{\ell}_\infty$
PDPGD [19]	100,0	0,980	0,003	0,8	0,008	∞	5,3	0,060	∞	0,0	0,000	∞
FAB [11]	100,0	0,972	0,003	0,5	0,006	∞	5,2	0,059	∞	0,0	0,000	∞
FMN [18]	100,0	0,967	0,003	0,7	0,007	∞	5,3	0,058	∞	0,0	0,000	∞
TR [21]	100,0	0,662	0,005	5,5	0,018	∞	5,4	0,046	∞	0,1	0,001	∞
DeepFool [20]	100,0	0,663	0,005	5,3	0,018	∞	5,4	0,046	∞	0,1	0,001	∞
C&W [7]	100,0	0,668	0,005	1,6	0,012	∞	5,2	0,044	∞	0,0	0,000	∞
LS-FGSM	97,2	0,554	0,006	98,0	0,217	0,014	42,9	0,149	0,100	70,8	0,636	0,180
LS-APGD	100,0	0,000	0,020	99,9	0,000	0,020	59,1	0,000	0,120	79,4	0,661	0,220
LS-PGD	100,0	0,000	0,020	99,9	0,000	0,020	59,0	0,000	0,100	74,9	0,634	0,240
LS-BIM	100,0	0,000	0,020	99,7	0,000	0,040	45,5	0,000	0,140	50,9	0,424	0,300
HNT-APGD	100,0	0,988	0,003	100,0	1,000	0,003	100,0	1,000	0,002	93,9	0,976	0,003
HNT-PGD	100,0	0,702	0,004	100,0	0,724	0,004	100,0	0,731	0,003	93,5	0,967	0,005
HNT-BIM	100,0	0,698	0,004	99,9	0,720	0,004	96,6	0,701	0,003	94,2	0,975	0,005
HNT-FGSM	99,8	0,559	0,006	99,8	0,553	0,006	94,8	0,516	0,003	94,7	0,956	0,006

Table B.8: ASR comparison of HORNET-paired attacks on ImageNet under different ℓ_p-norms. Results show the mean and standard deviation of the Attack Success Rate (ASR) for the best implementation of each attack. Both single-source ($I1 \rightarrow I2$) and ensemble-source ($E \rightarrow I2$) transfer scenarios are reported.

ℓ_p	Attack	I1 → I2	I1 → I3	I1 → I4	E → I2	E → I3	E → I4
ℓ_0	HNT-PGD$_0$	**67.0 ± 0.22**	**64.4 ± 0.26**	**41.4 ± 0.24**	**77.6 ± 0.17**	**77.2 ± 0.26**	**48.9 ± 0.24**
	HNT-APGD	**37.1 ± 0.15**	**41.3 ± 0.27**	**10.3 ± 0.24**	**96.1 ± 0.15**	**98.2 ± 0.21**	**54.4 ± 0.13**
ℓ_1	HNT-BIM	10.2 ± 0.17	8.8 ± 0.26	5.5 ± 0.24	75.6 ± 0.21	79.5 ± 0.26	53.3 ± 0.24
	HNT-PGD	10.1 ± 0.15	8.4 ± 0.26	5.2 ± 0.24	77.0 ± 0.37	80.9 ± 0.26	53.9 ± 0.24
	HNT-APGD	**89.3 ± 0.25**	**90.5 ± 0.26**	**74.7 ± 0.24**	**99.7 ± 0.12**	**100.0 ± 0.26**	**99.2 ± 0.24**
ℓ_2	HNT-BIM	7.4 ± 0.14	5.8 ± 0.21	3.7 ± 0.08	97.2 ± 0.35	99.3 ± 0.26	93.2 ± 0.25
	HNT-PGD	87.6 ± 0.27	88.4 ± 0.26	71.1 ± 0.24	99.4 ± 0.35	99.9 ± 0.26	98.3 ± 0.24
	HNT-PGD	**100.0 ± 0.35**	**100.0 ± 0.21**	**100.0 ± 0.24**	100.0 ± 0.35	100.0 ± 0.16	100.0 ± 0.24
ℓ_∞	HNT-APGD	89.3 ± 0.25	90.5 ± 0.26	74.7 ± 0.14	100.0 ± 0.30	100.0 ± 0.22	100.0 ± 0.32
	HNT-BIM	100.0 ± 0.43	100.0 ± 0.26	100.0 ± 0.24	100.0 ± 0.37	100.0 ± 0.16	100.0 ± 0.28

fixed-budget ones, we followed the default hyperparameters provided in the respective official implementations (e.g., 'adv_lib' defaults of 40 iterations and step size 0.01/0.3 for PGD). These parameters are applied at each evaluated ϵ during HORNET's binary-search procedure. When multiple implementations were available, we verified that the total number of iterations and the overall computational budget were equivalent across methods to ensure a fair and consistent comparison.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: