TransferBench: Benchmarking Ensemble-based Black-box Transfer Attacks

Fabio Brau[†], **Maura Pintor**[†], **Antonio Emanuele Cinà**[‡], **Raffaele Mura**[†], **Luca Scionis**[†*], **Luca Oneto**[‡], **Fabio Roli**[‡], **Battista Biggio**[†]
[†]*University of Cagliari,* [‡]*University of Genoa,* [*]*University of Sapienza, Italy.*
{fabio.brau, maura.pintor, raffaele.mura, luca.scionis, battista.biggio}@unica.it
{luca.oneto, antonio.cina, fabio.roli}@unige.it

Abstract

Ensemble-based black-box transfer attacks optimize adversarial examples on a set of surrogate models, claiming to reach high success rates by querying the (unknown) target model only a few times. In this work, we show that prior evaluations are systematically *biased*, as such methods are tested only under overly optimistic scenarios, without considering (i) how the choice of surrogate models influences transferability, (ii) how they perform against robust target models, and (iii) whether querying the target to refine the attack is really required. To address these gaps, we introduce TransferBench, a framework for evaluating ensemble-based black-box transfer attacks under more realistic and challenging scenarios than prior work. Our framework considers 17 distinct settings on CIFAR-10 and ImageNet, including diverse surrogate-target combinations, robust targets, and comparisons to baseline methods that do not use any query-based refinement mechanism. Our findings reveal that existing methods fail to generalize to more challenging scenarios, and that query-based refinement offers little to no benefit, contradicting prior claims. These results highlight that building reliable and query-efficient black-box transfer attacks remains an open challenge. We release our benchmark and evaluation code at: `https://github.com/pralab/transfer-bench`.

1 Introduction

Machine learning (ML) models are vulnerable to adversarial examples, i.e., inputs intentionally crafted to cause misclassification [6, 32]. When white-box access to the *target model* is available, one can easily find adversarial examples using gradient-based attacks [9]. This scenario is typically considered when evaluating adversarial robustness of defense mechanisms [9, 10]. However, real-world systems are typically deployed as black-box services, preventing full access to the model's architecture and parameters, and thus also to their internal gradients [5]. Under this limitation, developing effective gradient-free (black-box) attacks becomes more challenging. Two main strategies are often considered, encompassing black-box *transfer* and *query* attacks. The first approaches assume white-box access to one or more surrogate models trained to solve the same task as the (unknown) target, optimize the adversarial examples against them, and then evaluate whether the attack successfully *transfers* to the target model [13, 25]. We refer to these attacks as *query-free*, since they do not iteratively query the target model to improve the attack success rate. The second approaches, instead, are only based on querying the target model and leveraging its feedback to improve the attack, using black-box optimizers such as genetic algorithms [1], natural evolution strategies [28], and zeroth-order methods [7] to find adversarial examples. While black-box *transfer* attacks are query-free, they may suffer from low success rates when the surrogate does not closely approximate the target. Conversely, black-box *query* attacks can reach higher success rates but at the cost of many queries, given that these attacks do not leverage any knowledge/approximation of the target.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

To mitigate these issues, recent work has proposed combining these two approaches to define a stream of novel attacks, referred to as *ensemble-based black-box transfer attacks*. They are based on (i) attacking an ensemble of surrogate models to improve attack transferability against unknown targets [8], while (ii) leveraging the feedback obtained by querying the target model to refine the attack optimization [8, 16, 17, 23, 29]. We refer to these two steps as *surrogate-based attack optimization* (SBA) and *query-based attack refinement* (QBR), respectively, and present a categorization of such attacks in Sect. 2. Ensemble-based black-box transfer attacks exhibit near-perfect performance on standard benchmarks like the NeurIPS-2017 adversarial challenge [19]. In the untargeted case (i.e., when attacks do not aim for misclassification in a specific class), they often succeed without even issuing a single query to the target [36, 41].

In this work, we first show that such methods have been evaluated by considering overly optimistic, biased experimental setups. In particular, we argue that prior evaluations have considered too favorable settings in which: (i) surrogate ensembles have very similar architectures to that of the target, favoring high transfer success rates; (ii) only standard (non-robust) models have been often used as targets—making it is much easier to find successful attacks—and when robust targets are considered, only robust surrogates are included in the surrogate pool; (iii) no proper ablation studies have been conducted, making it difficult to properly assess how much *query-based attack refinement* contributes to the overall attack success rate on top of the given *surrogate-based attack optimization*. To overcome these issues, we introduce *TransferBench* (Sect. 3), a benchmark for evaluating ensemble-based black-box transfer attacks under more realistic and challenging scenarios. Our evaluation spans 17 settings on CIFAR-10 and ImageNet, incorporating (i) diverse surrogate-target combinations, (ii) robust target defenses, and (iii) transfer attack baselines that never query the target to refine the attack. This allows us to assess the contribution of the query-based refinement strategies used by several attacks over simpler, query-free transfer attack baselines. Our results (Sect. 4) show that existing methods often fail to generalize to more complex scenarios and that querying the target model provides only marginal benefits, if any, contradicting previous claims.

To summarize, our work provides the following contributions. From the *methodological* viewpoint: (i) we define an evaluation protocol for ensemble-based black-box transfer attacks under more realistic and challenging scenarios, including diverse surrogate-target combinations, and robust target and surrogate models; (ii) we include query-free naïve baselines to assess the actual improvements coming from querying the target model. (iii) we re-evaluate state-of-the-art ensemble-based black-box transfer attacks, exposing pitfalls in their original evaluations caused by overly favorable experimental conditions. From a more practical perspective, our *implementation* contributions are: (i) we introduce `TransferBench`, a plug-and-play library for fast evaluation of any p-norm black-box transfer attack on a set of default benchmark scenarios; (ii) we provide efficient (batch-wise) re-implementations of 9 ensemble-based black-box transfer attacks (in contrast to the original, inefficient sample-wise implementations); (iii) we release an online leaderboard, accessible at `https://transferbench.github.io/`, to rank and compare ensemble-based black-box transfer attacks; and (iv) we provide `trbench`, a command-line interface (CLI) to Weight&Biases that facilitates tracking experimental runs and analyzing results in detail.

We discuss related work on benchmarking black-box transfer attacks in Sect. 5, highlighting their differences with respect to `TransferBench`. We conclude by summarizing our findings in Sect. 7 and remarking that, accordingly, building reliable and query-efficient ensemble-based black-box transfer attacks remains an open and unsolved challenge, contradicting evidence from prior work.

2 Ensemble-based Black-box Transfer Attacks

We present here a novel categorization of ensemble-based black-box transfer attacks, unifying their formalization and clarifying the role of *surrogate-based attack optimization* against that of *query-based refinement*. To this end, let us denote the target model with g, and the set of m surrogate models with $\mathbf{f} = \left(f^{(1)}, \dots, f^{(m)}\right)$, assuming that they operate on a common input domain $\mathcal{X}$ and provide logit outputs in $\mathbb{R}^c$. Given an input $x_0 \in \mathcal{X}$, a target label t, and a norm parameter p, the attack aims to construct an adversarial example $x^* \in \mathcal{X}$ that fools the target model, i.e., such that $g_t(x^*) = \max_j g_j(x^*)$, and lays within a perturbation budget $\|x^* - x_0\|_p < \varepsilon$. In theory, this problem could be approached by minimizing a loss function $\mathcal{L}(g(\cdot), t)$. However, since the gradient of g is not accessible, this loss should be treated as non-differentiable with respect to the input and cannot be directly optimized using standard gradient-based methods. To circumvent this, a surrogate loss function $\mathcal{L}_{\texttt{ens}}(x, t, \mathbf{f}; z)$ is introduced. This function is differentiable with respect to x and approximates $\mathcal{L}$, becoming exact when $z = g(x)$ and each $f^{(i)}$ is a differentiable representative of g.[1]

With this notation, the attack x^* can be obtained by solving the following optimization problem:

$$x^* \in \arg\min_{x \in \mathcal{X}} \mathcal{L}_{\texttt{ens}}(x, t, \mathbf{f}; g(x)) \quad \text{s.t.} \quad \|x - x_0\|_p < \varepsilon. \tag{1}$$

This formulation unifies surrogate-based and query-based strategies, encompassing as special cases the *black-box transfer attacks* and the *black-box query attacks* (i.e., when $\mathcal{L}_{\texttt{ens}}(x, t, \mathbf{f}; \cdot)$, and $\mathcal{L}_{\texttt{ens}}(x, t, \cdot; z)$ are constant respectively).

Although the two contributions of the surrogate-ensemble and the query to the target are merged in $\mathcal{L}_{\texttt{ens}}$, from a practical perspective, Problem 1 is decoupled into the two sub-problems *surrogate-based attack optimization* (SBA) and *query-based refinement* (QBR) defined as follows:

$$x^*(w) \in \arg\min_{x \in \mathcal{X}} \mathcal{L}_{loc}(x, t, \mathbf{f}; w), \quad \text{s.t.} \quad \|x - x_0\|_p \leq \varepsilon, \tag{SBA}$$

$$w^* \in \arg\min_{w \in \mathcal{W}} \mathcal{L}(g(x^*(w)), t), \tag{QBR}$$

where the loss function $\mathcal{L}_{\texttt{loc}}$ is differentiable in x, and parameterized by $w \in \mathcal{W}$. The parameters w serve to guide $\mathcal{L}_{\texttt{loc}}$ so that its minimum aligns with that of $\mathcal{L}_{\texttt{ens}}$ at $x^*(w^*)$, and to reduce the search space of Problem QBR, which would be intractable if defined directly over $\mathcal{X}$.

Surrogate-based Attack Optimization. A solution to SBA can be estimated in various ways, depending on the optimization strategy and the choice of loss function. For example, GAA [38] assumes $\mathcal{W} = \mathbb{R}^c$ and employs the GACE loss:

$$\mathcal{L}_{\texttt{ens}}(x, t, \mathbf{f}; z) = \sum_i \left[(p_t(z) - 1)\, f_t^{(i)}(x) + \sum_{j \neq t} p_j(z)\, f_j^{(i)}(x) \right], \tag{3}$$

where $\mathcal{L}_{\texttt{loc}} = \mathcal{L}_{\texttt{ens}}$, and $p(z)$ denotes the softmax probabilities derived from z. Note that the GACE loss reduces to the standard cross-entropy loss $\mathcal{L}_{\texttt{CE}}(g(\cdot), t)$ when the surrogates are a differentiable representative of g [38]. Other methods, such as BASES [8] and DSWEA [16], adopt a *convex combination* of losses, where each surrogate $f^{(i)}$ is assigned a non-negative weight w_i with $w_i \in [0, 1]$ and $\sum_i w_i = 1$. In this formulation, the loss $\mathcal{L}_{\texttt{loc}}$ is defined as

$$\mathcal{L}_{\texttt{loc}}(x, t, \mathbf{f}; w) = \sum_i w_i \mathcal{H}_\kappa \left(f^{(i)}(x), t \right), \quad \forall x \in \mathcal{X}, \tag{4}$$

where $\mathcal{H}_\kappa$ denotes the hinge loss with margin κ [9]. This formulation enables a weighted ensemble of surrogates to guide the generation of adversarial examples, balancing their contributions based on the weight vector w within the convex set $\mathcal{W}$. In the SIMBAODS [33], SUBSPACE [15], and GFCS [23], the surrogates are randomly sampled—i.e., only one w_i at the time is not-zero—taking values in $\{\pm 1\}$, indicating the direction to be used. While both SUBSPACE and GFCS minimize the loss defined in Equation (4), the SIMBAODS method produces a sub-optimal solution by considering only (weighted) random directions without explicitly optimizing any loss. In contrast, DSA [29] interprets the weights as probability scores used for sampling the surrogate models. Finally, HYBRID [31] follows a transversal approach, by averaging the surrogates whose parameters are represented by w.

Query-based Attack Refinement. Depending on the method, the solution to Problem QBR determines how feedback from the target model is used to infer the parameters in $\mathcal{W}$. Most attacks leverage this feedback to optimize the adversarial example directly, rather than to explicitly update the weights w. This family includes methods such as SIMBAODS [33], GFCS [23], and SUBSPACE [15], where the target model is queried on two candidate adversarial examples, $x^*(w_-)$ and $x^*(w_+)$, and the one yielding the lower loss $\mathcal{L}(g(\cdot), t)$ is selected. In contrast, GAA [38] queries the target model only once to obtain the logits $z^* = g(x^*)$ for a single candidate, thus requiring only one query.

In contrast, the second family of methods leverages the target model to adapt the parameters w associated with the surrogates, thereby dynamically refining the ensemble. In BASES [8], one weight at a time is updated by adding and subtracting a constant value η, generating two solutions, $x^*(w_-)$

[1] Let $f, g \in L^1(\mathcal{X})$; f is a differentiable representative of g if f is differentiable a.e., and $\int_{\mathcal{X}} |f - g| \, d\mu = 0$.

and $x^*(w_+)$, which are fed to the target model to select the weight configuration that reduces the loss. In DSWEA [16], the victim model is queried to rank the surrogates, and the gradient magnitudes are used to update the weights w. DSA [29] queries the target model to update the score associated with each surrogate, based on the estimated likelihood of successfully attacking the victim. Finally, HYBRID [31] fine-tunes the surrogates model parameters w on the queried output scores.

2.1 Evaluation Pitfalls and Challenges

Despite recent enthusiasm around ensemble-based black-box transfer attacks, we identify critical shortcomings in how these methods have been evaluated. In particular, current protocols often assume overly favorable conditions that do not reflect realistic threat models. We expand on three major gaps in the current literature that motivate our work.

Impact of Surrogate Pool. Prior work often evaluates transfer-based attacks using a carefully curated ensemble of surrogate models. In particular, these ensembles share architectural similarities or training pipelines with the target model. Such setups correspond to an *homogeneous* surrogate setting, where the ensemble contains models that share the same architectural family as the target, e.g., attacking a ResNet-50 with other ResNet variants. However, ensembles constructed from architectures that closely mirror the target can indeed result in high success rates, not due to the attack capabilities, but rather to inherent similarities that favor transferability.

Robust Targets. A second significant limitation in current evaluations concerns defense mechanisms that may be utilized by the target. In practice, many deployed systems incorporate robustness mechanisms (e.g., adversarial training [24]) specifically to counter adversarial examples that may not be accessible to the attacker. However, several studies on ensemble-based transfer attacks target only non-robust models. Only a few studies consider robust target models, and in those cases, they are tested including robust surrogate models. As a result, it is unclear whether high transfer success rates reported in the literature carry over to robust targets attacked with non-robust surrogates.

Using Feedback from Target Models. Several recent methods attempt to increase attack effectiveness by querying the target model to refine the attack, either during the optimization process or to adaptively re-weight the surrogate ensemble based on feedback from the target's predictions. Nevertheless, these methods are rarely compared against simpler baselines such as a fixed-weight ensemble or query-free attacks that do not use queries at all. Our findings indicate that the marginal gain from query-based refinement is negligible in most settings, as a simple averaging scheme over surrogate models, or query-free baselines, achieves comparable or superior success rates, questioning the utility of complex adaptive strategies. We argue that this phenomenon is caused by the absence of a standardized evaluation methodology in this area.

3 TransferBench

We introduce `TransferBench`, a benchmark designed to assess the effectiveness of ensemble-based black-box transfer attacks in realistic and challenging *scenarios*. Each scenario includes two key factors: the set of surrogate models used to compute the attack, and the specific target model selected for the evaluation. In the easiest scenario, we assume the use of homogeneous surrogates, while more challenging scenarios involve surrogates whose architectures differ significantly from the target's. Additionally, `TransferBench` includes evaluations and comparisons against simple baselines that do not refine the attack optimization while querying the target. In the remainder of this section, we present the scenarios considered (Section 3.1), the baseline attack strategies used for reference (Section 3.2), and the implementation details of `TransferBench` (Section 3.3).

3.1 Scenarios

Transferability is strongly influenced by the architectural similarity between surrogates and the target model [18], and biased surrogate choices can therefore overestimate attack performance. To ensure fair evaluation, our benchmark includes diverse scenarios, listed in Specifically, Table 1 details the target-surrogate combinations, drawing them from open repositories such as Torchvision [26], HuggingFace [35], and PyTorch-CIFAR [11].

Table 1: Scenarios involved in the benchmark. The HeS includes only surrogates with an architecture different from the target model; the HoS includes only surrogates with the same architecture as the target; the HoS+R includes robust target models.

ImageNet		
Type	**Target**	**Surrogates**
HoS	VGG-19	Inc-v3, ConvNeXt-b, VGG-16
	ResNeXt-101	ResNet-50, ResNeXt-101, Dense-121
	ViT-B/16	Swin-B, Swin-T, ViT-B/32
HeS	VGG-19	ResNet-50, ResNeXt-101, Dense-121, Swin-{B,T}, ViT-B/32
	ResNeXt-101	Inc-v3, ConvNeXt-b, VGG-16, Swin-{B,T}, ViT-B/32
	ViT-B/16	Inc-v3, ConvNeXt-b, VGG-16, ResNet-50, ResNeXt-101, Dense-121
HoS+R	Pub-RN-50	ResNet-50, ResNeXt-101, Dense-121
	Mim-Sw-L	Swin-B, Swin-T, ViT-B/32
	Amini-Sw-L	Swin-B, Swin-T, ViT-B/32
CIFAR10		
Type	**Target**	**Surrogates**
HoS	VGG-19-bn	VGG-13-bn, ConvNeXt-t, VGG-16-bn
	ResNet-56	ResNet-44, ResNet-32, ShuffleNet-v2
	BEiT-B/16	Swin-B, Swin-T, ViT-B/16
HeS	VGG-19-bn	ResNet-{44,32}, ShuffleNet-v2, Swin-{B,T}, ViT-B/16
	ResNet-56	VGG-13-bn, VGG-16-bn, ConvNeXt-t, Swin-{B,T}, ViT-B/16
	ViT-B/16	VGG-{13,16}-bn, ConvNeXt-t, ResNet-{44,32}, ShuffleNet-v2
HoS+R	Peng-RWRN-70	ResNet-44, ResNet-32, ShuffleNet-v2
	Barto-WRN-94	ResNet-44, ResNet-32, ShuffleNet-v2

Homogeneous scenario (HoS) represents a surrogate setting where all surrogate models belong to the same family or share strong architectural similarity with the target. This is the most favorable condition for transferability, as the surrogate and target models are more likely to share decision boundaries. For instance, a transformer model is attacked using other transformers, or a CNN is attacked using CNN-based surrogates.

Heterogeneous scenario (HeS) simulates a black-box attack setting where the adversary has access to a pool of surrogate models that differ in architecture from the target model. This aims to reflect a realistic threat model where the adversary does not know the exact architecture of the target. For each target model, we select a diverse set of surrogates, such as combining convolutional and transformer-based models, to maximize architectural diversity.

Robust-Homogeneous scenario (HoS+R) evaluates transfer attacks against robustly trained models, with surrogates sharing architectures as in the Homogeneous scenario. Targets include state-of-the-art defenses—Pub-RN-50 [30], Mim-Sw-L [37], and Amini-Sw-L [2] for ImageNet; Peng-RWRN-70 [27] and Barto-WRN-94 [3] for CIFAR-10. Most models are sourced from RobustBench [12], except Pub-RN-50, which is taken from its original repository. We select models from different authors and architectures with top robust accuracy. This scenario is the most challenging, as model robustness significantly reduces attack transferability.

3.2 Baselines

To evaluate the impact of the target feedback, we provide different attack baselines in our benchmark. We include query-free methods, which do not perform any query to the target, along with two simple baselines, which instead query the target and solve the Problem SBA until success is obtained.

(Query-free) Transfer Attacks. Query-free *transfer attacks* exploit the transferability property of adversarial examples, avoiding queries to the target model. Notably, these methods may involve some weight update mechanism, but they do not query the target model to fine-tune the weights. This family includes a wide range of methods, from older static ensemble strategies with no weight update (e.g., ENS [22]) to more recent approaches, such as SASD_WS [36], which utilize a reinforcement mechanism to update the weights of the surrogate models.

Naïve Average Attacks. Ensemble-based transfer attacks performance is influenced by several factors, such as inner white-box attacks, query budget, and weight update mechanisms. Since evaluating the contribution of each component is not straightforward, we include two naïve baseline methods, `NaiveAvg10` and `NaiveAvg100`, which represent the simplest ensemble-based transfer attacks. In these methods, no weight updates are performed, i.e., w_i remains fixed at $\frac{1}{m}$ during the attack optimization. The solution of Problem SBA is estimated by considering the projected-gradient-descent, following the iterative formulation in [24], to minimize the ensemble loss function defined in Equation (4) with $\kappa = 200$. This consists in computing the following iterations,

$$x^{(k+1)} = \Pi_{B_\infty(x_0,\varepsilon)} \left(x^{(k)} - \alpha \cdot \mathrm{sgn} \nabla_x \mathcal{L}_{\texttt{loc}}(x^{(k)}, t, \mathbf{f}; w) \right), \quad \forall k < T, \tag{5}$$

where $\Pi_{B_\infty(x_0,\varepsilon)}$ is the projection on the l_∞-ball in $\mathcal{X}$, centered in x_0 having radius ε, and the step-size $\alpha = 4.8/255$. During the QBR, the black-box model is evaluated in $x^* = x^{(T)}$, to validate the attack success. If the sample x^* fails to transfer to the target model, i.e., $t \neq \max_j g_j(x^*)$, then Equation (5) is repeated by initializing $x^{(0)}$ with the previous attempt x^*. We considered two versions of the baseline, NaiveAvg10 and NaiveAvg100, that leverage 10 and 100 local iterations, respectively.

3.3 Implementation Details

`TransferBench` is a plug-and-play modular library for ensemble-based attack evaluation, written in Python, and leveraging the PyTorch framework [26]. The library supports customized attacks, models, and datasets. `TransferBench` relies on three main objects: The `AttackEval` wraps the `TransferAttack`, representing the attack to be evaluated, and runs the evaluation on the specified `Scenario`, which includes the information on the parameters, models, and datasets.

The usage of the library is kept as straightforward as possible: users can evaluate their attack on the default scenarios we selected, see an example in Listing 1, or on custom scenarios that can be easily created by instantiating a new `Scenario` object.

```python
from transferbench import AttackEval
# The user can define a custom method
def myattack(victim_model, surrogate_models, inputs, labels, targets,
    p, eps, maximum_queries) -> Tensor: ...
# Initializing and running the evaluation
evaluator = AttackEval(myattack)
evaluator.set_scenarios("omeo-imagenet-inf", "etero-cifar10-inf")
results = evaluator.run()
```

Listing 1: Standard Usage of `TransferBench` API for the evaluation of a custom attack.

Scenario. The `Scenario` object includes all the components required for evaluating a given attack, such as the target model, list of surrogates, dataset, and three attack constraints. Specifically, these constraints, stored in a non-modifiable Python `dataclass` named `HyperParams`, are shared between the `TransferAttack` and `AttackEval`. The `HyperParams` includes the query budget Q, the epsilon budget for the attack ε, and the p-norm.

Attack Protocol. The attack is performed by a `TransferAttack` function, defined as a binding to the Python `Protocol` class, i.e., a function with a fixed signature that takes only the following input arguments: `victim_model, surrogate_model, inputs, labels, targets, p, eps, maximum_queries`. The `TransferAttack` function solely performs the attack, without overriding any class methods, and returns a batch of attack samples. The computation of queries is not handled by `TransferAttack` but is externally monitored by `AttackEval`. Specifically, the target is passed as a `Callable` function that does not allow access to the model's internal parameters. To fully exploit GPU memory usage, `TransferBench` supports and recommends batched attack implementations. To properly count the number of queries, the user can leverage a mask tensor to indicate which samples require a forward pass. Attacks in `TransferBench` are collected in the `attack_zoo` module, where the original code has been enhanced by adopting a batched version of the attacks. Refer to Table 4

```
user@laptop$ trbench display --query 'victim_model=="vgg19" and status == "finished"'
>>> [INFO] 2025-05-14 22:53:01,217
| id     | status    | attack     | victim_model   | campaign   |   p |       eps |   maximum_queries | dataset   | available   |
|:------|:---------|:-----------|:---------------|:-----------|----:|---------:|------------------:|:----------|:------------|
| a3360 | finished | DSWEA      | vgg19          | omeo       | inf | 0.062745 |                50 | ImageNetT | True        |
| a290b | finished | NaiveAvg   | vgg19          | etero      | inf | 0.062745 |                50 | ImageNetT | True        |
| 9cdde | finished | SASD_WS    | vgg19          | omeo       | inf | 0.062745 |                50 | ImageNetT | True        |
| 9ce5b | finished | BASES      | vgg19          | etero      | inf | 0.062745 |                50 | ImageNetT | True        |
```

Figure 1: `trbench` usage: Each run is associated with a unique id synchronized with WB.

for a complete list. Furthermore, the two NAIVEAVG100 and NAIVEAVG10 baselines, as well as query-free transfer attacks imported from the `TransferAttack` library [14].

TRBench CLI. The `TransferBench` library includes `trbench`, a command-line interface designed to orchestrate large-scale benchmarking of black-box transfer attacks, that exposes the three commands: `run`, `display`, `report`. An example in Fig. 1. The tool integrates seamlessly with the Weights&Biases (W&B) [4] logging backend and enables programmatic inspection, filtering, and re-execution of runs based on query expressions over metadata (e.g., surrogate model, campaign, run status). It supports parallel execution by multiple users while preventing job conflicts through coordinated status tracking. This CLI tool `trbench` facilitates reproducibility by automating the management of incomplete or failed jobs and providing real-time visibility into ongoing experiments.

4 Experimental Results

This section presents the capabilities of the `TransferBench` framework and shows the pitfalls in the evaluations of the methods in the original papers, discussed earlier, each in a dedicated section. For the evaluations, we selected 13 attacks from the attack-zoo and ran them on the 17 different scenarios described in Table 1. We set the perturbation budget to $\varepsilon = 16/255$, the maximum queries allowed to $Q = 50$, and considered only targeted attacks bounded in l_∞ distance. Note that we only focused on the targeted case, since the untargeted one can be considered addressed, [22]. In this analysis, we included only black-box attacks that reached satisfactory success in some scenarios. In particular, we only included BASES, DSWEA, GAA, GFCS, SIMBAODS, HYBRID, NAIVEAVG among the query-based, and ENS, CWA, LGV, SASD_WS, SVRE, MBA, among the query-free. Experiments have been conducted using an NVIDIA RTX A6000 GPU, imposing a batch size not smaller than 20 samples for each target, with a (maximum) time limit of 30 hours per run.

Dataset. For these experiments, we considered the NeurIPS-17 challenge[2], containing a subset of 1000 images taken from ImageNet [21] associated with both a ground-truth and a target label. Furthermore, we included a subset of 1000 images of CIFAR-10 [20], where the targeted label t_i has been determined by considering the label l_{i+1} of the next sample, or $l_i + 1\%10$ in case of two consecutive samples with the same ground-truth.

Metrics. The Attack Success Rate (ASR) is defined as the proportion of adversarial samples that are successfully classified as the target label by the victim model. We also consider the *average queries per success*, $\bar{q}$, which measures the average number of queries required to achieve a successful attack.

4.1 Impact of Biased Surrogate Selection

Considering the aggregated results in Figure 2 and Table 2, it becomes clear that the choice of the surrogates has a huge impact on the attack success rate. The query-free attack SASD_WS and the simple baselines are capable of achieving a satisfactory success rate under the homogeneous scenarios, i.e., where the architecture of the surrogates and the targets corresponds to the same family. A sudden drop in the success rate is visible when attacks are compared under the heterogeneous scenarios. This experiment suggests that a biased choice of surrogates may easily lead to the wrong conclusion, albeit needed for a rich evaluation of the attacks.

Attacking the ViT models. An interesting finding emerges from the analysis of the ViT model, which consistently exhibits a low attack success rate (ASR) across all scenarios. As noted in [34], this is likely due to artifacts introduced by the image tokenizer into the gradient—a phenomenon specific

[2] www.kaggle.com/competitions/nips-2017-targeted-adversarial-attack/data

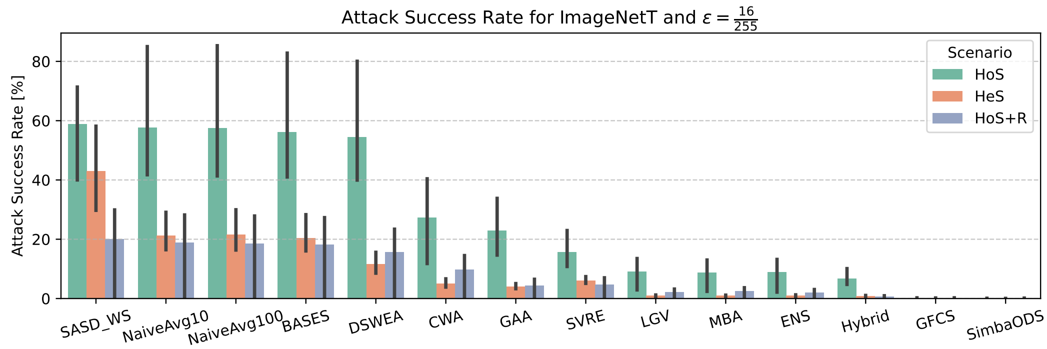


Figure 2: ASR on ImageNet. The bars represent the mean among different targets, while the error bars show the inter-quartile range. On average, the SASD_WS baseline has higher success rates among all the attacks, including those that leverage multiple queries.

Table 2: ASR and averaged queries-per-success for the ImageNet dataset.

	`ResNeXt-101`				`VGG-19`				`ViT-B/16`				`Pub-RN-50`		`Amini-Sw-L`		`Mim-Sw-L`	
	`HoS`		`HeS`		`HoS`		`HeS`		`HoS`		`HeS`		`HoS+R`		`HoS+R`		`HoS+R`	
Attack	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$
BASES	86.3	7.4	29.0	11.9	79.3	8.0	27.6	12.7	2.7	15.3	4.5	15.3	54.6	11.1	0.0	-	0.0	-
DSWEA	83.5	6.2	17.7	12.1	76.6	5.9	13.5	10.0	3.3	10.4	3.6	14.4	46.8	10.1	0.0	-	0.0	-
GAA	28.1	7.3	5.5	7.2	39.5	7.1	4.6	6.9	1.2	7.5	2.0	7.8	13.0	7.5	0.0	-	0.0	-
GFCS	0.0	-	0.1	17.0	0.2	26.0	0.3	10.7	0.3	10.3	0.1	39.0	0.5	11.4	0.0	-	0.0	-
SimbaODS	0.0	-	0.0	-	0.1	38.0	0.1	6.0	0.1	1.0	0.0	-	0.3	29.7	0.0	-	0.0	-
Hybrid	9.5	25.3	0.9	28.0	10.7	25.1	1.2	27.0	0.0	-	0.0	-	1.9	28.0	0.0	-	0.0	-
NaiveAvg10	89.2	6.4	30.8	11.8	80.8	7.2	27.4	11.5	2.7	15.9	5.5	14.8	56.4	10.4	0.0	-	0.0	-
NaiveAvg100	89.6	3.4	31.8	9.1	81.0	4.2	28.1	9.8	1.7	18.6	4.6	14.2	55.7	8.7	0.0	-	0.0	-
ENS	22.1	0.0	0.7	0.0	4.3	0.0	1.8	0.0	0.0	-	0.3	0.0	6.1	0.0	0.0	-	0.0	-
CWA	58.3	0.0	6.0	0.0	22.5	0.0	7.3	0.0	1.1	0.0	1.7	0.0	29.0	0.0	0.0	-	0.0	-
LGV	21.5	0.0	0.8	0.0	5.5	0.0	1.6	0.0	0.3	0.0	0.3	0.0	6.4	0.0	0.0	-	0.0	-
MBA	21.5	0.0	0.8	0.0	4.5	0.0	1.5	0.0	0.3	0.0	0.3	0.0	7.3	0.0	0.0	-	0.0	-
SASD_WS	96.4	0.0	69.2	0.0	46.3	0.0	47.1	0.0	33.7	0.0	12.4	0.0	59.8	0.0	0.0	-	0.0	-
SVRE	25.4	0.0	7.7	0.0	20.5	0.0	7.1	0.0	1.1	0.0	3.1	0.0	14.0	0.0	0.0	-	0.0	-

to the ViT architecture and closely tied to the number of input patches. This highlights a limitation of gradient-based attack strategies and presents a challenge for future ensemble-based attack methods.

4.2 Impact of Defense Mechanism

As detailed in Table 2, all the attacks fail to effectively transfer adversarial examples from non-robust models to the two robust models, `Mim-Sw-L` and `Amini-Sw-L`, which exhibit robust accuracy above 70% against perturbations of magnitude up to $8/255$—half the magnitude considered in our scenarios. This trend is consistent on the CIFAR-10 dataset, as shown in Table 3. An interesting exception is the `Pub-RN-50` model, which, likely due to gradient obfuscation strategies, appears robust to targeted white-box attacks but remains vulnerable to transfer attacks. It is worth noting, however, that the perturbation budget of $16/255$ used in our evaluation exceeds the one claimed in the original paper.

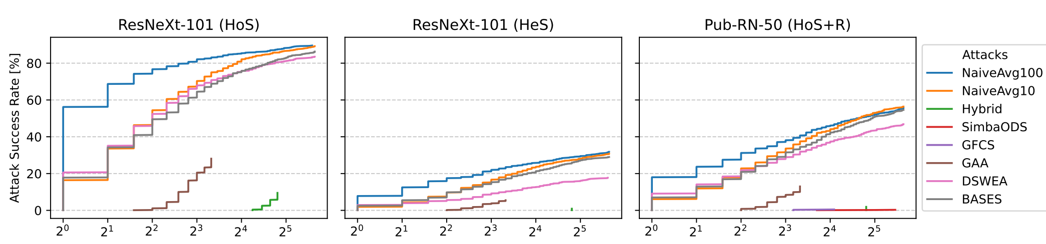


Figure 3: ASR-vs-Query curves for `ResNeXt-101` and `Pub-RN-50` tested on the ImageNet dataset.

Table 3: Results for CIFAR-10.

	`ResNet-56`				`VGG-19-bn`				`ViT-B/16`		`BEiT-B/16`		`Peng-RWRN-70`		`Barto-WRN-94`	
	`HoS`		`HeS`		`HoS`		`HeS`		`HeS`		`HoS`		`HoS+R`		`HoS+R`	
Attack	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$	ASR	$\bar{q}$
BASES	99.9	1.2	99.6	1.9	99.4	2.1	98.6	1.9	73.0	5.6	99.9	1.8	1.9	1.7	1.8	2.1
DSWEA	100.0	1.2	99.7	1.9	99.8	1.8	97.3	2.1	76.0	3.6	99.8	1.5	2.4	2.1	2.5	1.1
GAA	89.4	5.1	69.2	5.9	65.2	5.7	77.2	5.6	55.7	6.1	76.9	5.7	2.0	2.8	2.3	3.5
GFCS	25.3	10.2	26.9	11.3	19.8	9.3	18.2	8.9	9.3	14.2	22.7	12.3	1.9	3.5	2.2	2.9
SimbaODS	27.3	50.0	25.5	11.7	18.3	7.2	18.3	9.7	9.3	14.5	19.1	15.9	1.9	3.9	2.3	3.2
Hybrid	86.2	4.8	62.1	11.1	59.8	11.4	66.4	10.0	30.7	19.5	67.0	9.8	1.6	28.0	1.5	28.0
NaiveAvg10	99.8	1.2	99.5	1.6	99.7	1.8	98.3	1.9	73.9	5.4	99.9	1.6	1.9	1.0	1.9	1.8
NaiveAvg100	99.9	1.1	99.5	1.1	99.6	1.1	98.7	1.8	73.7	4.9	99.6	1.2	1.8	4.6	1.8	5.7
ENS	97.2	0.0	97.4	0.0	98.3	0.0	86.7	0.0	40.2	0.0	93.6	0.0	1.3	0.0	1.8	0.0
CWA	98.5	0.0	97.9	0.0	97.8	0.0	93.5	0.0	60.2	0.0	99.0	0.0	1.6	0.0	1.7	0.0
LGV	97.4	0.0	97.9	0.0	98.5	0.0	84.7	0.0	36.0	0.0	93.4	0.0	1.7	0.0	1.6	0.0
MBA	96.9	0.0	97.4	0.0	98.8	0.0	84.3	0.0	35.5	0.0	93.0	0.0	1.4	0.0	1.5	0.0
SASD_WS	99.8	0.0	98.8	0.0	99.2	0.0	97.5	0.0	71.8	0.0	99.0	0.0	1.8	0.0	1.9	0.0
SVRE	95.2	0.0	98.5	0.0	99.0	0.0	92.7	0.0	56.3	0.0	98.3	0.0	2.1	0.0	1.4	0.0

4.3 Impact of Using the Feedback from the Target Models

For a finer analysis, we disentangle the analysis of the contribution of the target's feedback to both the ASR (the higher, the better) and the average number of queries $\bar{q}$ (the lower, the better). As shown in Table 2 and Figure 2, feedback from the target generally leads to higher ASR, allowing query-based methods to (though not consistently) outperform query-free ones. Nevertheless, among the query-based attacks, our baselines NaiveAvg10 and NaiveAvg100—which do not perform any weight updates—outperform other methods that actively leverage target feedback to refine the attack optimization. This indicates that the favorable conditions under which those methods were previously evaluated may have led to an overestimation of the contribution of the refinement mechanisms.

Concerning the average number of queries per successful attack, Figure 3 shows that feedback from the target becomes more crucial when only a small number of internal iterations are performed during the local attack stage. Indeed, although NaiveAvg10 achieves a comparable ASR to NaiveAvg100, it requires a significantly higher number of queries. In conclusion, the query-based feedback used by existing methods in the literature appears to function primarily as a mechanism for reinitializing the attack—either from a previously failed attempt or from a new random starting point—rather than being effectively exploited by the weight update or refinement process.

5 Related Work

We discuss here related work on benchmarking black-box transfer attacks: `TransferAttack` [14], `TransferAttackEval` [39], and `BlackBoxBench` [40].

The library `TransferAttack` [14] and the benchmark `TransferAttackEval` [39] implement various black-box attacks, including ensemble-based transfer methods, but support only query-free approaches. Moreover, only surrogate pools of the original paper, without exploring alternative configurations, have been used. In contrast, our benchmark, `TransferBench`, is designed for systematic comparison across diverse scenarios, reflecting realistic and challenging settings. Moreover, `TransferBench` also enables evaluation with attacks wrapped from `TransferAttackEval`.

`BlackBoxBench` [40] includes only query-free transfer attacks and evaluates them using a fixed surrogate set—up to five non-robust residual and convolutional models—shared across all target models. `TransferBench` expands it by including a broader range of surrogate configurations (`HoS`, `HeS`, and `HoS+R`), enabling a deeper investigation into how surrogate diversity impacts transferability.

6 Ethical Considerations and Broader Impacts

Being the `TransferBench` tools usable for plug-and-play attacks evaluation, a malicious actor could potentially exploit them for secondary aims. This section aims to provide more insights for a responsible use of the benchmark, as well as mitigation strategies for model providers.

Responsible Use of the Benchmark. While tools for generating and evaluating adversarial examples can advance our understanding of model vulnerabilities, they also present inherent dual-use risks. To address this, we clearly define the intended use of our tool as a resource for defenders, i.e., researchers and practitioners focused on strengthening model robustness and advancing adversarial defense strategies. Acceptable uses are strictly limited to research aimed at improving model robustness, advancing secure AI system design, and enhancing the evaluation of adversarial defenses, facilitating proactive red-teaming strategies. We explicitly prohibit any offensive applications, including but not limited to unauthorized security testing of live systems, surveillance, privacy violations, or the deployment of adversarial attacks for malicious purposes. As highlighted in our recommended practices, the evaluation of adversarial attack performance is intended solely to inform and improve safety measures, not to exploit model vulnerabilities.

Safeguards and Mitigation Strategies. To reduce the risk of transfer-based adversarial attacks that exploit query access to target models, we recommend several mitigation strategies. First, model providers should implement strict access controls, such as rate limiting, authentication, and anomaly detection, to monitor and restrict potentially abusive querying behavior. Defensive techniques like input filtering, adversarial training, or certified defenses can also reduce the effectiveness of surrogate-based attacks.

Additional Insights from the Benchmark. Although the primary aim of our benchmark is to support the evaluation and development of defenses, it also reveals important findings. Notably, our results expose a significant gap between the claimed robustness of some defenses and practical resilience in more advanced transfer-based settings. This highlights the need for more comprehensive and rigorous benchmarks that reflect the complexities of available adversarial tools, and encourages the development of defenses that generalize beyond narrow evaluation protocols.

7 Conclusion and Future Work

We introduced `TransferBench`, a plug-and-play benchmarking tool for evaluating ensemble-based black-box transfer attacks under realistic and challenging conditions. Unlike prior benchmarks, which operate under overly optimistic assumptions, `TransferBench` accounts for surrogate model diversity, robust target defenses, and the role of target feedback. Across 17 settings on CIFAR-10 and ImageNet for each attack, our evaluation revealed key insights: (i) attack success is highly sensitive to surrogate choice and diversity; (ii) many state-of-the-art methods fail against robust targets; and (iii) query-based refinement often provides little to no gain over simple transfer baselines. These findings challenge common assumptions and highlight the need for more principled, robust attack strategies.

While our work has limitations, future research will focus on improving the surrogate pools by incorporating additional criteria (e.g., number of parameters). On the implementation side, we plan to extend `TransferBench` with new evaluation metrics, such as surrogate forward/backward counts and memory usage. Since the evaluations in this paper represent only a subset of the scenarios `TransferBench` supports, we aim to broaden the experimental coverage with more p-norms, ε budgets, and datasets. We believe `TransferBench` will foster progress toward more reliable and query-efficient black-box attack algorithms.

Acknowledgments and Disclosure of Funding

This work has been partly supported by the EU-funded Horizon Europe projects ELSA (GA no. 101070617), Sec4AI4Sec (GA no.101120393), and CoEvolution (GA no. 101168560); and by the projects SERICS (PE00000014) and FAIR (PE00000013) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU. It has also been partially supported by the EU—NGEU National Sustainable Mobility Center (CN00000023), Italian Ministry of University and Research Decree n. 1033—17/06/2022 (Spoke 10). This work was conducted while Luca Scionis was enrolled in the Italian National Doctorate on AI run by Sapienza University of Rome in collaboration with the University of Cagliari.

References

[1] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B. Srivastava. Genattack: practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*, pages 1111–1119. ACM, 2019.

[2] Sajjad Amini, Mohammadreza Teymoorianfard, Shiqing Ma, and Amir Houmansadr. Meansparse: Post-training robustness enhancement through mean-centered feature sparsification. *arXiv preprint arXiv:2406.05927*, 2024.

[3] Brian R. Bartoldson, James Diffenderfer, Konstantinos Parasyris, and Bhavya Kailkhura. Adversarial robustness limits via scaling-law and human-alignment studies. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[4] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

[5] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[6] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, 2013.

[7] HanQin Cai, Yuchen Lou, Daniel McKenzie, and Wotao Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139, pages 1193–1203. PMLR, 2021.

[8] Zikui Cai, Chengyu Song, Srikanth Krishnamurthy, Amit Roy-Chowdhury, and Salman Asif. Blackbox attacks via surrogate ensemble search. *Advances in Neural Information Processing Systems*, 35:5348–5362, 2022.

[9] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[10] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *ArXiv e-prints*, 1902.06705, 2019.

[11] Yaofo Chen. pytorch-cifar-models: Pretrained cifar10/cifar100 models in pytorch. https://github.com/chenyaofo/pytorch-cifar-models, 2020. GitHub repository, accessed on 2025-05-09.

[12] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

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

[14] Trustworthy AI Group. Devling into adversarial transferability on image classification: A review, benchmark and evaluation. https://github.com/Trustworthy-AI-Group/TransferAttack, 2025.

[15] Yiwen Guo, Ziang Yan, and Changshui Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] Cong Hu, Zhichao He, and Xiaojun Wu. Query-efficient black-box ensemble attack via dynamic surrogate weighting. *Pattern Recognition*, 161:111263, 2025.

[17] Zhichao Huang and Tong Zhang. Black-box adversarial attack with transferable model-based embedding. *arXiv preprint arXiv:1911.07140*, 2019.

[18] Jaehui Hwang, Dongyoon Han, Byeongho Heo, Song Park, Sanghyuk Chun, and Jong-Seok Lee. Similarity of neural architectures using adversarial attack transferability. In *European Conference on Computer Vision*, pages 106–126. Springer, 2024.

[19] Kaggle. Nips 2017: Targeted adversarial attack. https://www.kaggle.com/competitions/nips-2017-targeted-adversarial-attack/data, 2017.

[20] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *Dataset*, 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

[21] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231. Springer, 2018.

[22] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017.

[23] Nicholas A Lord, Romain Mueller, and Luca Bertinetto. Attacking deep networks with surrogate-based adversarial black-box methods is easy. *arXiv preprint arXiv:2203.08725*, 2022.

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

[25] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *ArXiv e-prints*, abs/1605.07277, 2016.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *White Paper*, 2017.

[27] ShengYun Peng, Weilin Xu, Cory Cornelius, Matthew Hull, Kevin Li, Rahul Duggal, Mansi Phute, Jason Martin, and Duen Horng Chau. Robust principles: Architectural design principles for adversarially robust cnns. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023. URL https://papers.bmvc2023.org/0739.pdf.

[28] Hao Qiu, Leonardo Lucio Custode, and Giovanni Iacca. Black-box adversarial attacks using evolution strategies. In Krzysztof Krawiec, editor, *GECCO '21: Genetic and Evolutionary Computation Conference, Companion Volume, Lille, France, July 10-14, 2021*, pages 1827–1833. ACM, 2021.

[29] Meng Shen, Changyue Li, Qi Li, Hao Lu, Liehuang Zhu, and Ke Xu. Transferability of white-box perturbations: Query-Efficient adversarial attacks against commercial DNN services. In *33rd USENIX Security Symposium (USENIX Security 24)*, 2024.

[30] Chawin Sitawarin, Jaewon Chang, David Huang, Wesson Altoyan, and David Wagner. Pubdef: Defending against transfer attacks from public models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Tvwf4Vsi5F.

[31] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *29th USENIX security symposium (USENIX Security 20)*, pages 1327–1344, 2020.

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[33] Yusuke Tashiro, Yang Song, and Stefano Ermon. Diversity can be transferred: Output diversification for white-and black-box attacks. *Advances in neural information processing systems*, 33: 4536–4548, 2020.

[34] Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, and Yu-Gang Jiang. Towards transferable adversarial attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2668–2676, 2022.

[35] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Scott Gray, Muskan Kumar, Teven Le Scao, Patrick von Platen, Anisha Joshi, Joshua P. D. McGibbon, Fabien Barret, Maarten Bos, and Warner orr. Hugging face transformers: State-of-the-art natural language processing. https://github.com/huggingface/transformers, 2019. Accessed: 2025-05-09.

[36] Han Wu, Guanyan Ou, Weibin Wu, and Zibin Zheng. Improving transferable targeted adversarial attacks with model self-enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24615–24624, 2024.

[37] Xiaoyun Xu, Shujian Yu, Zhuoran Liu, and Stjepan Picek. Mimir: Masked image modeling for mutual information-based adversarial robustness. *arXiv preprint arXiv:2312.04960*, 2023.

[38] Xiangyuan Yang, Jie Lin, Hanlin Zhang, and Peng Zhao. Improving query efficiency of black-box attacks via the preference of deep learning models. *Information Sciences*, page 121013, 2024.

[39] Zhengyu Zhao, Hanwei Zhang, Renjue Li, Ronan Sicre, Laurent Amsaleg, Michael Backes, Qi Li, and Chao Shen. Revisiting transferable adversarial image examples: Attack categorization, evaluation guidelines, and new insights. *arXiv preprint arXiv:2310.11850*, 2023.

[40] Meixi Zheng, Xuanchen Yan, Zihao Zhu, Hongrui Chen, and Baoyuan Wu. Blackboxbench: A comprehensive benchmark of black-box adversarial attacks. *arXiv preprint arXiv:2312.16979*, 2023.

[41] Yao Zhu, Jiacheng Sun, and Zhenguo Li. Rethinking adversarial transferability from a data distribution perspective. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gVRhIEajG1k.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Pitfalls and countermeasures exposed in the abstract are shown by dedicated evaluations using the proposed benchmark.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations and future improvements have been discussed together with the conclusions of the work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setups are detailed in the dedicated section, and results can be reproduced applying the benchmark API;

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is publicly available in the dedicated repository, with detailed documentations that are also summarized in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Instructions for for tests reproducibility are detailed in the the experimental setup subsection and in the description of the protocols involved in the proposed benchmark.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars have been exploited in the bar-plots for the sake of a fair visualization of aggregated results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information on the computer resources have been included in the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper fully complies with the NeurIPS Code of Ethics. It uses only public, licensed datasets without personal data, avoids deprecated resources, and clearly reports dataset limitations. No human subjects or harmful applications are involved. Potential misuse is acknowledged with suggested safeguards. All code and data for reproducibility will be released under appropriate licenses.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussion on broader impact of this work has been discussed in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: A description of the safeguards to avoid misuse has been included in Section 6.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the authors of external repositories, machine learning libraries, and dataset have been referenced in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Proposed benchamark is furnished with detailed instructions, and example of usage shown on Python Notebooks.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Supplementary materials of "TransferBench: Benchmarking Ensemble-based Black-box Transfer Attacks"

A Methods involved in the benchmark

In this benchmark, we considered the works described in Table 4. The DSA and SubSpace methods have not been directly compared, as they exhibit near-zero performance in the targeted scenarios with such a constrained amount of queries. All the attacks have been tested on the scenarios of the original papers, achieving the same performance. Original scenarios can be found in the `transferbench/config/scenarios` path of the benchmark.

Table 4: ASR and average-queries-per-success claimed in the original papers.

Attack	**Venue**	m	HeS	HoS+R	Targeted	p	ε	ASR [%]	$\bar{q}$
SubSpace [15]	NeurIPS 2019	3	✓	✗	✗	∞	$13/255$	98.9%	462
SimbaODS [33]	NeurIPS 2020	4	✗	✗	✓	∞	$13/255$	92.0%	985
Hybrid [31]	Usenix 2020	3	✗	✗	✓	∞	$13/255$	100%	14.3
GFCS [23]	ICLR 2022	4	✗	✗	✓	2	$\sqrt{0.001d}$[1]	60.0%	20
BASES [8]	NeurIPS 2022	20	✗	✗	✓	∞	$16/255$	99.7%	1.8
GAA [38]	PR 2024	4	✗	✗	✓	∞	$16/255$	46.0%	3.9
DSA [29]	Usenix 2024	3	✓	✓	✗	∞	$16/255$	96.9%	136
DSWEA [16]	PR 2025	10	✗	✗	✓	∞	$16/255$	96.6%	2.7

[1] Images included in the experiments have $d = 3 \cdot 299 \cdot 299$ pixels, from which $\varepsilon \approx 16.37$

B Instructions

The `TransferBench` codebase is accompanied by three main instructional resources:

- The primary `Readme.md` provides installation guidance and a quick-start tutorial for using the API with minimal setup.
- A companion example notebook offers in-depth, hands-on instructions, demonstrating how to use the framework with varying levels of customization.
- The `attacks_zoo/Readme.md` explains the implementation of the `TransferAttack` protocol within the `attacks_zoo` module.
- Instructions for setting up and using the `trbench` CLI command are detailed in the dedicated `benchmark_tools/Readme.md`.

Further details and the complete codebase are available on the official GitHub repository: `https://github.com/pralab/transfer-bench`.

C Licenses of external assets

The benchmark involved external assets for the models and query-free attacks.

Robust models The robust models `Mim-Sw-L` [37], `Amini-Sw-L` [2], `Peng-RWRN-70` [27], `Barto-WRN-94` [3] have been imported from RobustBench [12] released under MIT license, except for `Pub-RN-50` [30], which has been taken from its original repository, released under Apache 2.0 license.

Black-box attacks Query-free black box attacks involved for comparison have been imported form `TransferAttack` [14] under the MIT license.

D Analysis on the Important Factors

The composition of the surrogate pools is determined based on considerations of the models' architectures, as discussed in the paper. However, determining the optimal pool composition from the available models is inherently challenging. Indeed, an attack-driven selection of surrogates would require an exhaustive search over all possible model combinations, resulting in a combinatorial explosion of experiments on the order of $\binom{N}{K}$, where N is the number of available models and K is the maximum size of each surrogate pool. Beyond its computational infeasibility, such a strategy would also require fixing one or more attack algorithms in advance, thereby introducing a methodological bias in the pool selection.

To remain faithful to the objectives of this benchmark—particularly, to expose and mitigate suboptimal evaluation practices commonly adopted in transfer-based attack studies—we deliberately opted for a fixed subset of target and surrogate models. To validate the robustness of this design choice, we conduct an *a posteriori* factor analysis aimed at quantifying the actual impact of the adopted configurations on the benchmark results. Specifically, to examine the influence of key factors on the attack success rate (ASR), we represent each experimental configuration—defined by the target model, surrogate pool, and scenario type (`HeS`, `HoS`, `HoS+R`)—as a structured feature vector comprising:

1. A one-hot vector encoding the architectural characteristics of the target model, where the first, second, and third entries are set to 1 if the model lacks skip connections, includes residual connections, or employs attention layers, respectively;

2. A one(s)-hot vector encoding the aggregated architectural properties of the surrogate pool, following the same principle—that is, each entry indicates whether the pool contains models without skip connections, or with residuals, or attention layers;

3. A one-hot vector representing the scenario typology, derived from the categorical variable "Scenario Type".

The value of ε has been included as well. This setup avoids cherry-picking and enables a controlled study of the factors influencing ASR. In Table 5, we analyze the impact of the factors using a correlation analysis derived from fitting a linear model to predict the ASR from the features described above.

Table 5: Linear model coefficients predicting ASR from configuration features. `Res_T`, `CNN_T`, and `Att_T` denote the presence of residual, convolutional, and attention components in the target model; `Res_S`, `CNN_S`, and `Att_S` refer to the corresponding properties of the surrogate pool.

Res_T	CNN_T	Att_T	Res_S	CNN_S	Att_S	HeS	HoS	HoS+R	ε
0.16	0.13	-0.09	0.02	0.01	0.035	0.05	0.25	-0.28	0

From the analysis, we can deduce that Transformer-based targets correlate negatively with ASR, CNN and ResNet targets positively, and while individual surrogate architectures have a limited effect, their joint configuration strongly influences transferability—robust pools reduce ASR, homogeneous scenarios enhance it, confirming that transferability depends more on target–surrogate architectural relationships than on specific model types.

E Additional Results

We include in this section further plots not displayed in the main paper. Figure 4 involves the success vs average-queries-per-success curves for the ImageNet dataset, while the same curves relative to the CIFAR-10 dataset are visualized in Figure 5. Figure 6 shows aggregated success rates of the various attacks for the CIFAR-10 dataset. The empty plots are due to the fact that when the attack reaches zero success rate, the average-queries-per-success metric is not defined, and curves can not be displayed.

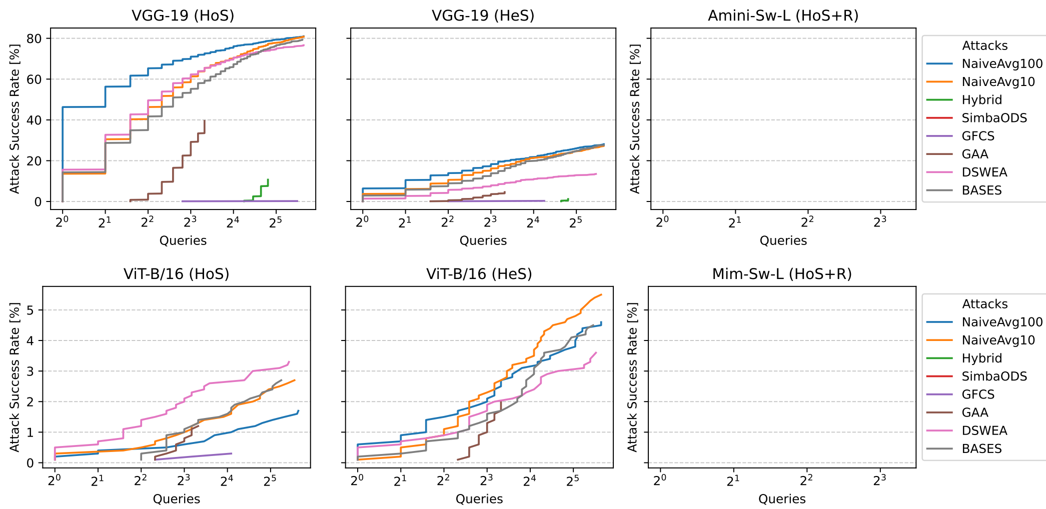


Figure 4: ASR-vs-Query curves on the ImageNet dataset.

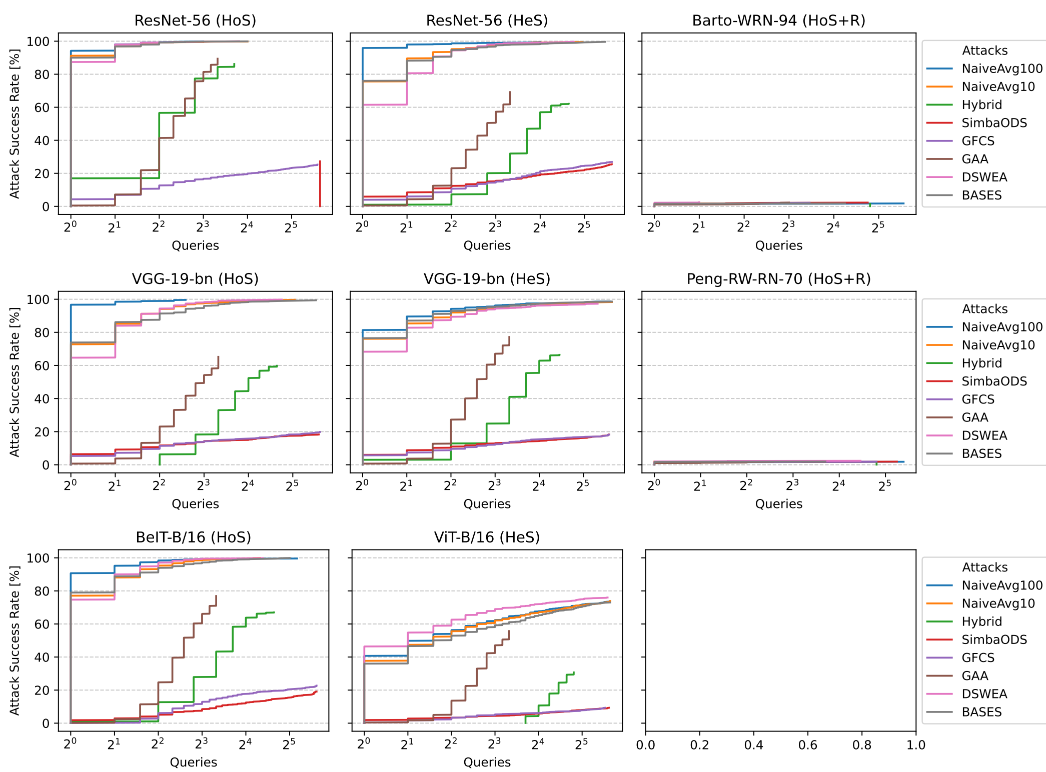


Figure 5: ASR-vs-Query curves on the ImageNet dataset for different victims.

F Statistical Significance of the Results

Since attacks are evaluated on subsets containing 1000 samples, this section aims at discussing whether such an amount of data is sufficient for the main claims of the work. In particular for the ASR, since the success can be modeled as a Bernoulli random variable, the variance of the sample mean p (i.e., the ASR) is known in closed form, $\mathrm{var}(\bar{X}) = \frac{p(1-p)}{n}$ (where p is the probability of attack success, i.e., the ASR). Therefore, the worst case is for ASR $\approx 50\%$, where the standard deviation

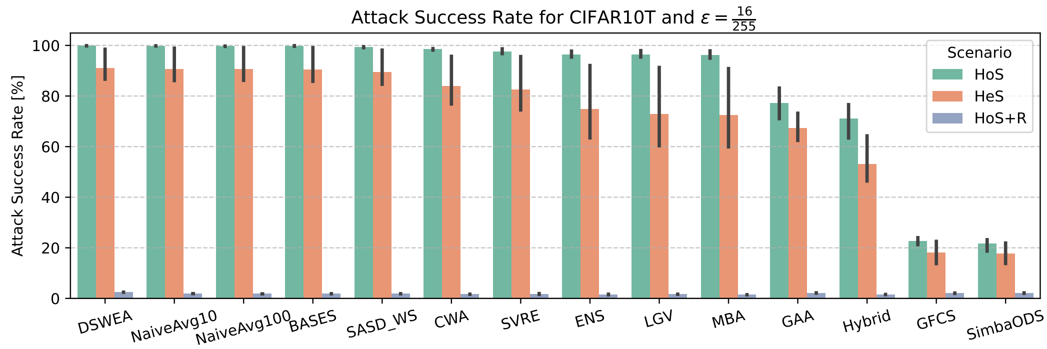


Figure 6: Aggregated attack success rate on the CIFAR10 dataset. Several attacks have an almost perfect success rate.

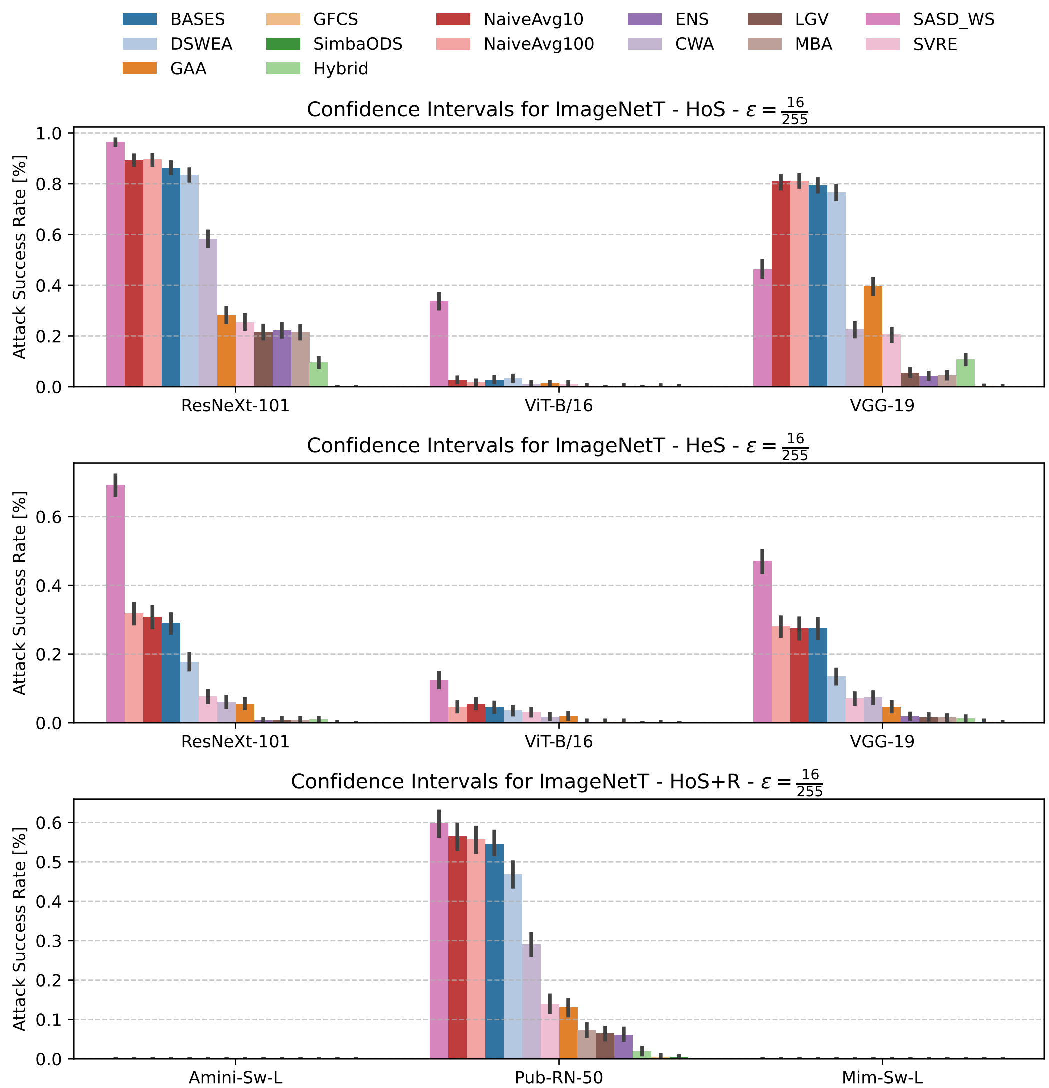
would be $\sigma_{50} = \sqrt{\frac{0.25}{1000}} \approx 1.58e-2$. This is an upper-bound, which means that the 95%-level confidence intervals I_p, for an ASR of p, would be always included in $[\bar{X} - 2\sigma_{50}, \bar{X} + 2\sigma_{50}]$, i.e., the attack-success rates a_1, a_2 of two methods that differ more than 6.32% can be considered statistically different, thereby proving that our claims are statistically sound. For a more detailed analysis, we collected the ASR of the methods on the bar plots shown in Figure 7, which provide a model-wise comparison while also highlighting the confidence intervals at the 0.95 level.

G Comparisons with other Perturbation Budgets

This section aims to evaluate the performance of the attacks with different perturbation budgets. Specifically, since success in homogeneous scenarios is easily achievable, we considered a lower perturbation budget of $8/255$. Results are reported in Figure 8, where barplots are used to compare the accuracy among different target models included in this scenario, and also confidence intervals are shown. The take-out messages are aligned with the $16/255$ perturbations budget, even though, as expected, a slightly lower ASR is achieved.

Furthermore, Figure 9 compares ASR for the robust models with a higher perturbation budget of $32/255$, showing that, surprisingly, both `Amini-Sw-L` and `Mim-Sw-L` models are still robust against such a larger perturbation.

Figure 7: Comparison of the ASR among different target models with confidence interval. Non-overlapping intervals indicate that the difference is statistically significant.



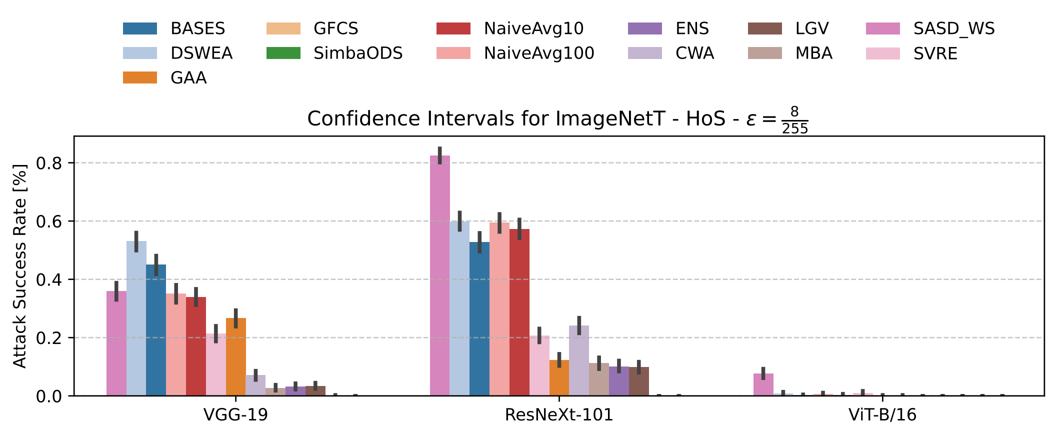


Figure 8: Homogeneous scenario with a smaller perturbation budget.

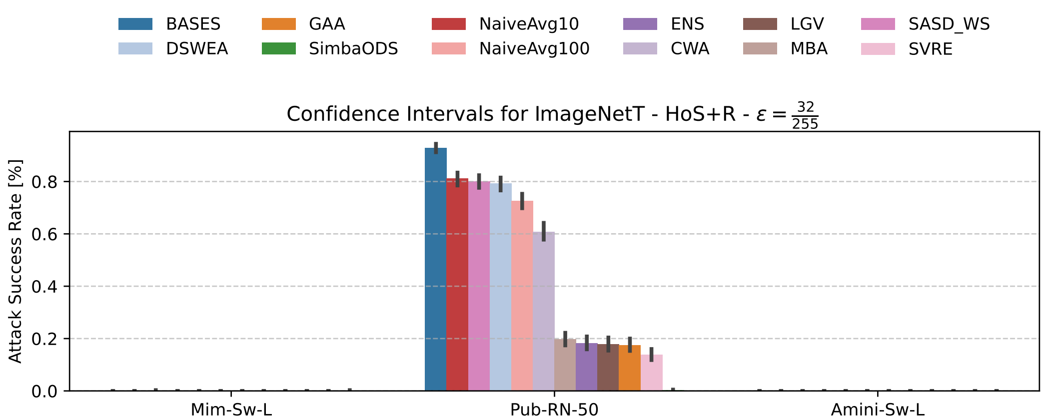


Figure 9: Attacking robust models is still challenging with a larger perturbation budget.