Rethinking Image-Scaling Attacks: The Interplay Between Vulnerabilities in Machine Learning Systems

Yue Gao [1] **Ilia Shumailov** [2] **Kassem Fawaz** [1]

Abstract

As real-world images come in varying sizes, the machine learning model is part of a larger system that includes an upstream image scaling algorithm. In this paper, we investigate the interplay between vulnerabilities of the image scaling procedure and machine learning models in the decision-based black-box setting. We propose a novel sampling strategy to make a black-box attack exploit vulnerabilities in scaling algorithms, scaling defenses, and the final machine learning model in an end-to-end manner. Based on this scaling-aware attack, we reveal that most existing scaling defenses are ineffective under threat from downstream models. Moreover, we empirically observe that standard black-box attacks can significantly improve their performance by exploiting the vulnerable scaling procedure. We further demonstrate this problem on a commercial Image Analysis API with decision-based black-box attacks.

1. Introduction

Recent advances in machine learning (ML) techniques have demonstrated human-level performance in many vision tasks, such as image classification (Russakovsky et al., 2015; Szegedy et al., 2016; 2017) and object detection (Ren et al., 2015; Redmon et al., 2016). As real-world images come in varying sizes, the practical ML system must include an image scaling algorithm before the downstream ML model. The scaling algorithm resizes input images to match the fixed input size of a model, which takes the input and performs vision tasks, such as classification.

The model and scaling algorithm in an ML system have become attractive targets for attackers. ML models are vulnerable to adversarial examples (Szegedy et al., 2013; Biggio et al., 2013): an adversary can add imperceptible perturbations to the input of a model and change its prediction (Madry et al., 2018; Carlini & Wagner, 2017). Recently, scaling algorithms were also found to be vulnerable to image-scaling attacks (Xiao et al., 2019; Quiring et al., 2020): an adversary can manipulate a large image such that it will change into a different image after scaling, thereby inducing an incorrect prediction in the model. However, image-scaling attacks are easily blocked by subsequent defenses (Quiring et al., 2020; Kim et al., 2021) and are not generalizable to other fields. We note that more subtle exploitation should leverage the weakness of the scaling stage *jointly* with adversarial examples against the ML model.

In this paper, we investigate the interplay between vulnerabilities of the image scaling procedure and ML models. Our investigation focuses on the more practical decision-based black-box setting, where the attacker can only query the model without a confidence score or knowledge of its internal implementation. We show that the attacker can jointly attack the scaling procedure and the ML model, posing more serious threats. From one side, black-box attacks (Brendel et al., 2018; Chen et al., 2020; Cheng et al., 2019; 2020; Li et al., 2020a; 2021; Zhang et al., 2021) can leverage the weakness of the scaling function to improve their performance significantly. On the other side, most image-scaling defenses (Quiring et al., 2020; Kim et al., 2021) are not effective in protecting the scaling function from being exploited by adversarial examples, even if they successfully prevent the image-scaling attack.

As a first step, we generalize the attack setting and re-design black-box attacks to jointly exploit the scaling function. We characterize the common approach of existing black-box attacks and identify *noise sampling* as a critical step shared by these attacks. Based on this observation, we propose to incorporate the weakness of the scaling function through a novel technique which we call Scaling-aware Noise Sampling (SNS). The overview of SNS is illustrated in Figure 1. The high-level idea is guiding traditional *low-resolution* black-box attacks to search for the adversarial examples along a direction that best exploits the scaling function. Our utilization of the noise sampling step makes SNS a plug-and-play technique applicable to different attacks. In particular,

[1]University of Wisconsin–Madison, Madison, WI, USA [2]Vector Institute, Toronto, ON, Canada. Correspondence to: Yue Gao <gy@cs.wisc.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

we integrate SNS with two representative decision-based black-box attacks: the boundary-based HSJ (Chen et al., 2020) and the optimization-based Sign-OPT (Cheng et al., 2020) attacks. We call these *high-resolution attacks*, targeting the ML pipeline as a whole and exploiting scaling.

Next, we design two novel techniques to circumvent image-scaling defenses (Quiring et al., 2020). Such defenses not only protect the scaling function but also hinder the traditional black-box attacks. To incorporate these defenses in our high-resolution attacks, we identify their root mechanism and propose novel approximations. First, we design *improved gradient estimation* for the defense that slows down the black-box attack's convergence with non-useful gradients. Second, we design *offline* expectation over transformation (Athalye et al., 2018b) to attack randomized defenses without additional queries to the black-box model. With these techniques, we circumvent 4 out of 5 state-of-the-art defenses to exploit the scaling function.

Finally, we conduct extensive experimentation to evaluate our high-resolution black-box attacks and state-of-the-art image-scaling defenses. We empirically confirm that jointly attacking the scaling function and ML model effectively improves the performance of black-box attacks. We also show that the defended scaling functions retain weaknesses that enable a stronger black-box attack. We finish with a discussion about the evaluation of trustworthy ML.

Contributions. We take the *first* step towards exploring the interplay between different vulnerabilities in black-box ML systems, with additional novel insights into circumventing pre-processing defenses in the black-box setting. Our contributions can be summarized as follows.

- **Improving black-box attacks.** We propose a novel plug-and-play technique called *scaling-aware noise sampling*, which significantly improves black-box attacks when a vulnerable scaling algorithm precedes the ML model.
- **Circumventing image-scaling defenses.** We show that 4 out of 5 state-of-the-art defenses, designed to protect the scaling stage, retain weaknesses that enable stronger black-box attacks.
- **New perspective of trustworthy ML.** We reveal that preventing attacks targeting one component in the ML system does not necessarily mitigate the vulnerability from a broader perspective. The interplay of different vulnerabilities leads to unexpected and stronger threats, such as amplifying existing attacks.

2. Related Work

Black-box attacks against ML models are drawing increasing attention due to their practical setting. In the black-box

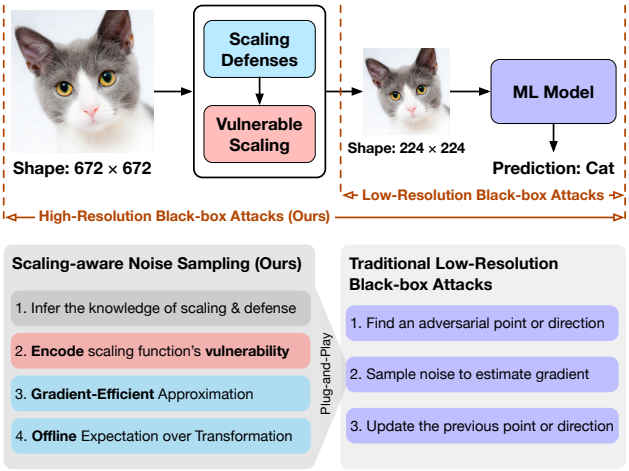


Figure 1: An overview of scaling-aware noise sampling. This plug-and-play technique guides traditional black-box attacks to exploit the weakness of the scaling function to improve their performance significantly.

setting, the attacker can only query the target model without knowledge of its internal implementation. Despite attacks leveraging the transferability of adversarial examples (Papernot et al., 2017), the more common query-based attacks fall into two categories: *score-based* and *decision-based* attacks. Score-based attacks assume access to the confidence score of the target model's prediction, which facilitates estimating the model's gradient (Li et al., 2020b; Chen et al., 2017; Ilyas et al., 2018; Tu et al., 2019). Decision-based attacks require access to the final decision without confidence scores, which is challenging but more practical (Brendel et al., 2018; Chen et al., 2020; Cheng et al., 2019; 2020; Li et al., 2020a; 2021; Zhang et al., 2021). However, all these attacks have only focused on the standalone model, omitting that practical ML models must include a scaling function to cater for input images of varying sizes.

Recently, image-scaling attacks (Xiao et al., 2019; Quiring et al., 2020) reveal that the scaling function can be exploited to hide a small image (from a different class) into a larger image, thereby fooling the model after downscaling. However, this exploitation is easily prevented (Quiring et al., 2020; Kim et al., 2021) and not generalizable to other fields. This paper shows that more subtle exploitation should leverage the weakness of scaling functions to hide the adversarial perturbation rather than an unperturbed image.

Understanding the trustworthiness of ML models is a critical objective in practice. In this work, we aim to explore if the interplay between vulnerabilities in practical ML systems could cause more harm. From one side, we explore whether the vulnerability of scaling functions could amplify black-box attacks. From the other side, we explore whether black-box attacks could still exploit defended scaling functions.

3. Background

3.1. Notation

A standard neural network f classifies low-resolution (LR) images $\boldsymbol{x} \in \mathbb{L} \coloneqq [0,1]^{p \times q}$ of height p and width q. For example, ResNet-50 (He et al., 2016) accepts 224×224 input images[1]. A scaling function g downscales the high-resolution (HR) image $X \in \mathbb{H} \coloneqq [0,1]^{m \times n}$ to the network's LR input space $\mathbb{L}$. A pre-processor h sanitizes the input X to prevent the attacker from hiding perturbation via the scaling function g. We focus on the black-box attack against the end-to-end classifier $F \coloneqq f \circ g \circ h$. For both f and F we denote their outputs as the classification label. We provide more background of scaling in Appendix A.1.

3.2. Adversarial Examples

Given an image $\boldsymbol{x} \in \mathbb{L}$ and a classifier f, the traditional adversarial example $\boldsymbol{x}'$ is visually similar to $\boldsymbol{x}$ but misclassified, i.e., $f(\boldsymbol{x}') \neq f(\boldsymbol{x})$ (Szegedy et al., 2013; Biggio et al., 2013). Traditional attacks construct the adversarial example by searching for $\boldsymbol{\delta}$ such that $f(\boldsymbol{x} + \boldsymbol{\delta}) \neq f(\boldsymbol{x})$, while minimizing $\|\boldsymbol{\delta}\|$ or maximizing the loss on $f(\boldsymbol{x} + \boldsymbol{\delta})$.

In this paper, we further consider the *high-resolution* adversarial example. Given an image $X \in \mathbb{H}$ and an end-to-end classifier F, the *high-resolution* adversarial example X' is visually similar to X but misclassified, i.e., $F(X') \neq F(X)$. Compared to the *low-resolution* $\boldsymbol{x}'$, the *high-resolution* X' additionally passes through the scaling stage $g \circ h$.

3.3. Image-scaling Attacks

Recent image-scaling attacks (Xiao et al., 2019; Quiring et al., 2020) attack the scaling stage g to hide a smaller non-adversarial image $\boldsymbol{x}^*$ from a different class into the larger image X. Conceptually, the adversarial image is visually similar to X but changes into a different image $\boldsymbol{x}^*$ after scaling, thereby inducing misclassification. The details and formulation of this attack can be found in Appendix A.2.

3.4. Image-scaling Defenses

The image-scaling attacks are prevented by several defenses, which fall into *pre-processing* and *detection* defenses. We briefly introduce five state-of-the-art defenses below and provide more details in Appendix A.3.

Pre-processing Defenses. Quiring et al. (2020) propose to sanitize the input image with *median* or *randomized* filtering operations before scaling; they reconstruct pixels by a median or randomly picked pixel within the sliding window. These defenses instantiate the preprocessor h in Section 3.1.

Detection Defenses. Kim et al. (2021) propose three detection defenses using spatial and frequency transformations: unscaling, minimum-filtering, and centered spectrum. These transformations result in discernible differences when applied to benign and perturbed images. For example, the unscaling invokes downscale and upscale operations sequentially to reveal the hidden image.

[1]We omit the channel dimension for simplicity.

3.5. Threat Models

We focus on the decision-based black-box setting, where we do not know the implementation of f and F; we only know the final predicted label without scores. This is the same threat model as considered by image-scaling attacks and defenses (Quiring et al., 2020; Kim et al., 2021). However, we can still *infer* the knowledge of the scaling stage $g \circ h$ with the brute-forcing method from Xiao et al. (2019). This knowledge can be reused for subsequent attacks, as it is typically fixed for a deployed ML model. We explain more details of this method in Appendix A.3.3.

The objective of our attack is to leverage the weakness of the scaling function to amplify existing black-box attacks in terms of fewer queries, less perturbation, and higher optimization efficiency. Since our attack exploits the weakness of the scaling function, we also incorporate defenses that are supposed to protect it. *As a result, attacking the full pipeline F (with scaling) yields significantly better performance than attacking the standalone model f (without scaling).*

4. Attack Methodology

Existing black-box attacks have only focused on the model f rather than the end-to-end pipeline F. Even if these attacks are deployed to attack the full pipeline F, they are not designed to exploit the scaling stage $g \circ h$ to hide the adversarial perturbation. Therefore, the main challenge is *how to make black-box attacks aware of the weakness of the scaling function in a generalizable manner*.

To address this challenge, we propose a novel technique which we call Scaling-aware Noise Sampling (SNS). As illustrated in Figure 1, SNS is a plug-and-play technique that guides existing black-box attacks to leverage the weakness of the scaling stage explicitly. We also propose two novel designs to let SNS incorporate pre-processors that are supposed to protect the scaling function.

4.1. Scaling-aware Noise Sampling (SNS)

We start by characterizing the common property of existing black-box attacks. Most decision-based black-box attacks walk near the decision boundary (Brendel et al., 2018; Chen et al., 2020; Li et al., 2020a; 2021; Zhang et al., 2021) or optimize for a particular objective function (Cheng et al., 2019; 2020). These attacks have varying algorithms, but all share a similar iterative approach: (1) find an adversarial

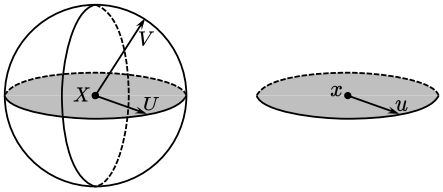


Figure 2: Illustration of our proposed SNS. In the HR space (left), randomly sampled noise V is unlikely to find the exploitable space (grey). SNS overcomes this problem by first sampling noise $\boldsymbol{u}$ in the LR space (right) and then projecting it back to the HR U, which lies in the LR space.

point or direction through linear search; (2) sample noise to estimate the gradient of a particular objective function; and (3) use the gradient to update the previous point or direction.

Based on the above characterization, we propose to incorporate the weakness of the scaling stage into the noise sampling by Scaling-aware Noise Sampling (SNS). The high-level idea of SNS is illustrated in Figure 2. Instead of sampling in the HR space $\mathbb{H}$ (as is done by naively applying attacks to the full pipeline), we sample noise in the post-scaling space $\mathbb{L}$ and project it back to $\mathbb{H}$. This novel sampling strategy guides black-box attacks to search adversarial examples along the direction that hides the most perturbation through the scaling function. As a result, attacking the full pipeline F yields significantly better performance than attacking the standalone model f. However, projecting noise from $\mathbb{L}$ to $\mathbb{H}$ requires reversing the projection defined by $g \circ h$. This operation is not straightforward, especially when incorporating the pre-processor h.

4.1.1. Straightforward SNS

We first notice that reversing the projection of $g \circ h$ is an instance of the image-scaling attack (Quiring et al., 2020), as we also need to search for a point $U \in \mathbb{H}$ that projects to the sampled $\boldsymbol{u} \in \mathbb{L}$. As such, a straightforward solution is to solve the following image-scaling attack problem:

$$U^* \coloneqq \underset{U \in \mathbb{H}}{\arg\min} \, \|(g \circ h)(X+U) - ((g \circ h)(X) + \boldsymbol{u})\|_2^2, \quad (1)$$

where $U^* \in \mathbb{H}$ is the HR noise that lies on the LR space.

This approach is computationally prohibitive because we need to solve an optimization problem for every sampled noise. We overcome this problem with efficient SNS.

4.1.2. Efficient SNS

To improve the efficiency of SNS, we note that the final objective of SNS is to sample an HR *noise* that lies in the LR space – it is not necessary to find the *exact* projection of a noise sample, as it is already random. Inspired by this observation, we find that an *imprecise* projection suffices to guide the gradient estimation in black-box attacks. One imprecise yet efficient projection we find is the gradient of Equation (1), written as

$$\tilde{U} \coloneqq \nabla_U \|(g \circ h)(X+U) - ((g \circ h)(X) + \boldsymbol{u})\|_2^2. \quad (2)$$

We use this more efficient solution to construct SNS, as summarized in Algorithm 1. It directs black-box attacks to efficiently search adversarial examples along the direction that hides the most perturbation through the scaling function.

Algorithm 1 Scaling-aware Noise Sampling (SNS)

Require: Scaling procedure $g \circ h$, initial point $X \in \mathbb{H}$.
Ensure: A noise $U \in \mathbb{H}$ that lies on the space $\mathbb{L}$.
Sample random noise $\boldsymbol{u} \in \mathbb{L}$ (i.e., input space of f).
Compute $\tilde{U} \in \mathbb{H}$ using Equation (2).
Output $U \leftarrow \tilde{U}$.

4.2. Incorporating Median Filtering Defenses

Although SNS is compatible with any projection defined by the scaling stage $g \circ h$, the defense h may have a side effect of hindering black-box attacks. Our empirical evaluation in Section 5.3.1 reveals that *median filtering defense can slow down the convergence of black-box attacks*.

We identify the root cause of this problem as the median function's robustness to outliers. When black-box attacks conduct line search along a fixed direction, the median filtering operation may not change its output in most of the searching steps. As a result, the attack converges slower and returns suboptimal results. Without loss of generality, we illustrate how the HSJ attack (Chen et al., 2020) conducts line search under the median function. Consider a starting point $\boldsymbol{x} = [1, 2, 3]$ and gradient $\boldsymbol{g} = [0, 1, 0]$. In this case, the line search procedure simply attempts $\{\boldsymbol{x} + \boldsymbol{g}, \boldsymbol{x} + 2\boldsymbol{g}, ...\}$ until reaching the decision boundary. This procedure, however, only increases the perturbation without changing the output of the median function after reaching $\boldsymbol{x} + 2\boldsymbol{g}$.

We overcome this problem by providing an estimate of the gradient that is more amenable to the line search. Since our attack estimates the gradient using noise sampled from Equation (2), we improve the gradient estimation by using a *trimmed and weighted average* function in its backward pass. The formulation of our improved median function and the overall filtering defense can be found in Appendix C.

Our evaluation in Section 5.3.1 verifies that our improved estimation not only reduces the query number of searching adversarial examples but also exploits the scaling function as much as possible. This is different from other differentiable approximation approaches like BPDA (Athalye et al., 2018a); e.g., using the identity function for approximation will not be able to exploit the scaling function.

4.3. Incorporating Randomized Defenses

In this section, we explain how to modify SNS to incorporate randomized pre-processing defenses that protect the scaling function. Although such defenses can be easily circumvented in the white-box setting with expectation over transformation (EOT) (Athalye et al., 2018a;b), we note that *directly applying EOT in black-box attacks is query-inefficient due to a large number of sampling operations*.

We overcome this challenge by computing the EOT *offline* without querying the black-box model. That is, instead of attacking the *expectation over the full pipeline*

$$\mathbb{E}_{h\sim\mathcal{H}}\, F(X) = \mathbb{E}_{h\sim\mathcal{H}}(f\circ g\circ h)(X), \tag{3}$$

we attack the *expectation over pre-processors*

$$(f\circ \mathbb{E}_{h\sim\mathcal{H}}(g\circ h))(X), \tag{4}$$

where $\mathcal{H}$ is the space that draws a randomized defense h. It is also possible to attack the expectation over the defense $\mathbb{E}_{h\sim\mathcal{H}}\, h(X)$, but we found this expectation hard to compute, as image-scaling defenses h typically work jointly with the scaling function g to be effective.

The above strategy effectively reduces the number of samples in EOT to zero; this is possible because we are able to infer the knowledge of the scaling stage (see Appendix A.3.3). As such, we only need to change the original scaling stage $g\circ h$ in Equation (2) to $\mathbb{E}_{h\sim\mathcal{H}}(g\circ h)$. In fact, we can derive the closed-form expectation for a particular defense, as we will discuss later in Section 5.3.2.

4.4. Plug-and-Play Integration with Black-box Attacks

Since noise sampling is a critical step for most black-box attacks, our proposed SNS is directly applicable to all of these attacks, as illustrated in Figure 1. We demonstrate its generalizability with integrations of two representative black-box attacks: the boundary-based HSJ attack (Chen et al., 2020) and the optimization-based Sign-OPT attack (Cheng et al., 2020). We refer to attacks on the model f as LR attacks, and our improved attacks on the full pipeline F as HR attacks.

High-Resolution HSJ Attack. HSJ (Chen et al., 2020) extends the boundary attack (Brendel et al., 2018) by walking near the decision boundary while adopting noise sampling to improve the gradient estimation. We apply SNS to guide its gradient estimation to hide perturbation. The detailed algorithm of our HR HSJ attack can be found in Appendix B.1.

High-Resolution Sign-OPT Attack. Sign-OPT (Cheng et al., 2020) optimizes a direction for minimal distance to the decision boundary. We apply SNS to guide its gradient to search a direction that meanwhile causes minimal perturbation before scaling. The detailed algorithm of our HR Sign-OPT attack can be found in Appendix B.2.

5. Evaluation

Finally, we perform an empirical evaluation of our improved HR black-box attacks and five state-of-the-art defenses designed to protect the scaling procedure. Our evaluation is designed to answer the following questions.

Q1: Can we improve black-box attacks by exploiting the scaling function to hide adversarial perturbation?

We observe a significant improvement when LR black-box attacks leverage our proposed SNS to attack the entire ML pipeline. With the same query budget, our HR attacks generate adversarial examples with less perturbation. We demonstrate this problem in a real-world Image Analysis API with decision-based and transfer-based black-box attacks.

Q2: Can we still improve black-box attacks when the scaling function is protected by defenses?

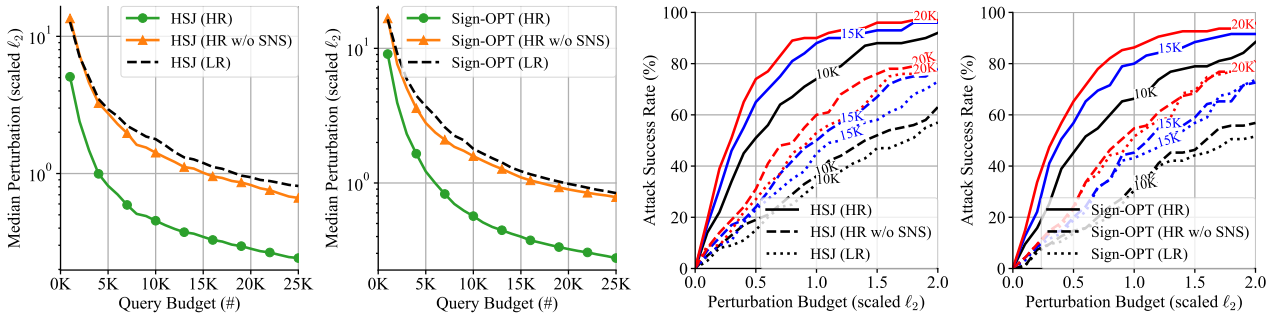
Our HR black-box attacks can still outperform their LR primitives under four out of five state-of-the-art defenses. These defenses include median filtering and all three detection defenses. We also analyze why these defenses fail to protect the scaling function despite their success in preventing standard image-scaling attacks.

5.1. Evaluation Setup

Dataset and Models. We use ImageNet (Russakovsky et al., 2015) and CelebA (Liu et al., 2015) datasets. For ImageNet, we randomly choose 1,000 images whose scaling ratio is at least 3 and downscale them to 672×672; the target model is a pre-trained ResNet-50 model (He et al., 2016) that accepts input images of size 224×224. For CelebA, we randomly choose 1,000 images and rescale their faces to 672×672; the target model is a pre-trained ResNet-34 model that accepts facial images of size 224×224 and predicts the `Mouth_Slightly_Open` attribute. We provide more details of these datasets and models in Appendix D.1. We also include decision-based and transfer-based attacks on the Tencent Image Analysis API, whose details and settings can be found in Appendix D.2.

Attacks and Setup. We implement HR black-box attacks based on the HSJ and Sign-OPT attacks as described in Section 4.4. We use OpenCV's linear scaling algorithm to represent the vulnerable scaling algorithm (Quiring et al., 2020). We also provide evaluations of our HR attacks and the above defenses in the white-box setting in Appendix G. Our code is available at `https://github.com/wi-pi/rethinking-image-scaling-attacks`.

Evaluation Metrics. We use standard metrics: (1) scaled ℓ_2-norm quantifies the adversarial perturbation divided by the scaling ratio to compare perturbation across different resolutions; (2) attack success rate (ASR) at various scaled ℓ_2-norm thresholds under a particular query budget.



(a) HSJ Attack (b) Sign-OPT Attack (c) HSJ Attack (d) Sign-OPT Attack

Figure 3: Comparison of our HR HSJ and Sign-OPT attacks with their LR primitives *under no defenses*. (a) and (b) compare the adversarial perturbation under different query budgets. (c) and (d) compare the attack success rate under different perturbation and query budgets. We include an ablation study to evaluate the effectiveness of our proposed SNS technique.

5.2. Evaluation of Q1: Undefended Image Scaling

In this experiment, we examine if black-box attacks can exploit the scaling function to improve their performance. For each query budget $q \in \{1000, 2000, ..., 25000\}$, we use the LR and HR attacks to generate a set of LR and HR adversarial examples, respectively. After that, we compare the perturbation between these two sets of adversarial examples.

Evaluation of the Perturbation. Figures 3a and 3b show the *median perturbation* generated by attacks when given different query budgets. When there is no defense to protect the scaling function, our HR attacks can reduce the perturbation significantly.

Evaluation of the Attack Success Rate. Figures 3c and 3d show the *attack success rate* of LR and HR attacks when given different perturbation budgets. When there is no defense to protect the scaling function, our HR attacks boost the success rate by a large margin. The solid lines of HR attacks are way above the dotted lines of LR attacks.

Ablation Study (SNS). We have included an ablation study in Figure 3 by disabling our proposed SNS. HR attacks without SNS reduce to similar performance as LR attacks. It shows that simply attacking the entire pipeline cannot exploit the scaling function to gain benefits. We highlight some of the comparisons in Table 3.

We also provide an ablation study that compares the performance between the straightforward SNS and efficient SNS discussed in Section 4.1, where we implement the straightforward SNS by solving Equation (1) using gradient descent with the Adam (Kingma & Ba, 2015) optimizer (1000 steps, 0.01 learning rate), which decreases the objective function to around 0.1. Due to the prohibitive computational cost, we only compare the attack effectiveness between precise and imprecise projections over 50 images. Table 1 shows that Equation (2) loses little attack effectiveness while avoiding the cost of Equation (1). It confirms our insight that improving the precision of a noise is not necessary — as long as the noise lies in the desired subspace, estimating gradients using such noise suffices to incorporate the vulnerability.

Table 1: Comparison of HR attacks on CelebA with 10K queries using the straightforward and efficient SNS.

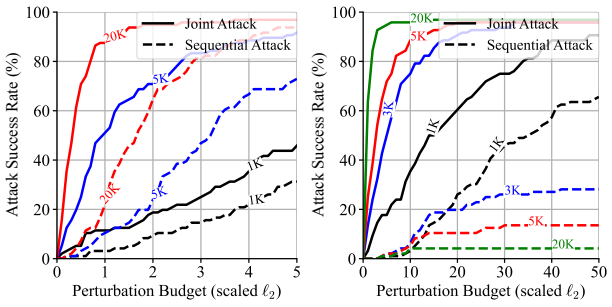
Attacks	ℓ_2	ASR under different ℓ_2 budgets				
		1.0	2.0	3.0	4.0	5.0
HSJ (HR) + Eq. (1)	**1.68**	**22.9%**	**63.7%**	**85.7%**	**97.1%**	**100.0%**
HSJ (HR) + Eq. (2)	1.72	20.0%	60.0%	82.9%	**97.1%**	**100.0%**

Table 2: Comparison of HR and LR attacks on CelebA with certain query budgets and image sizes. ASR is the attack success rate under perturbation budget $\ell_2 = 2.0$.

Attacks	Query = 10K		Query = 15K		Query = 20K	
	ℓ_2	ASR	ℓ_2	ASR	ℓ_2	ASR
HSJ (LR, 224×224)	5.26	10.8%	4.16	14.0%	3.58	18.3%
HSJ (HR, 672×672)	1.73	59.6%	1.37	74.5%	1.17	86.2%
HSJ (HR, 1120×1120)	**1.03**	**90.4%**	**0.82**	**98.9%**	**0.70**	**100.0%**

Ablation Study (Scaling Ratio). We provide an ablation study in Table 2 to examine the effectiveness of our chosen scaling ratio. Here, the attacker chooses a scaling ratio of 3 and 5. The results show that the attacker is free to choose a larger scaling ratio to yield even better results (under the assumption that the target model permits such large images).

Ablation Study (Attack Strategy). We evaluate a *sequential* attack (detailed in Appendix E) against the scaling function and ML model in Figure 4; our joint attack outperforms the sequential attack by a large margin. Interestingly, the sequential attack in Figure 4b becomes worse when the target adversarial example was obtained with more queries, as opposed to the joint attack. It suggests that naively hiding a pre-generated adversarial example under the defense is harder than directly targeting the whole pipeline.



(a) No Defense (b) Median Filtering Defense

Figure 4: Comparison of attack strategies. Our joint attack is significantly better than the naive sequential combination of image-scaling attacks and adversarial examples.

5.3. Evaluation of Q2: Defended Image Scaling

In this experiment, we examine if black-box attacks can still leverage the scaling function to improve their performance when there are defenses to protect the scaling function. We will discuss the details of each defense and analyze why they can (or cannot) prevent our improvement even if they have successfully blocked the image-scaling attack.

5.3.1. Median Filtering

Defense Details. The median filtering defense sanitizes the input image by applying the median filter $\boldsymbol{k}_{\text{med}}$. To evade this defense, an adaptive attacker has to perturb pixels in each window $\boldsymbol{w}$, such that the filtering output $w_m = \boldsymbol{k}_{\text{med}}(\boldsymbol{w})$ changes to the desired value w_t. This defense was regarded as robust to an adaptive attacker that changes the filter's output by setting pixels within the range $R := [w_m, w_t]$ to w_t. However, given that $|R| \leq |\boldsymbol{w}|/2$, the attacker needs only to modify at most half of the pixels to change the filtering output into a target value.

Discussion. The above observation implies that median filtering's effectiveness relies on the (large) value of $|R|$. In its original evaluation, $|R|$ is always large as they only hide a non-adversarial image. When considering adversarial perturbation, the target pixel w_t will be close to the original output w_m, implying a small range $R = [w_m, w_t]$ that decreases the effectiveness. Thus, although the median filtering defense can effectively prevent image-scaling attacks, it does not completely close the scaling function's vulnerability of stealthily hiding perturbation.

Evaluation. We evaluate the median filtering defense using our HR attacks and the improved gradient estimation described in Section 4.2. The comparison of HR and LR attacks are shown in Figure 5. As we can observe, our HR attacks still converge faster to a better solution than their LR primitives. As for attack success rate, HR attacks are able to outperform LR attacks even with 5K fewer queries.

Table 3: Comparison of HR and LR attacks with certain query budgets. ASR is the attack success rate under perturbation budget $\ell_2 = 1.0$. (*our SNS or improved gradient estimation is disabled, †under median filtering defense)

Attacks	Query = 10K ℓ_2	Query = 10K ASR	Query = 15K ℓ_2	Query = 15K ASR	Query = 20K ℓ_2	Query = 20K ASR
HSJ (LR)	1.78	33.0%	1.19	45.0%	0.94	53.0%
HSJ (HR*)	1.42	36.0%	1.01	50.0%	0.82	60.0%
HSJ (HR)	**0.45**	**74.0%**	**0.35**	**88.0%**	**0.28**	**90.0%**
HSJ (HR†*)	6.01	19.0%	3.29	30.0%	2.31	34.0%
HSJ (HR†)	**1.24**	**39.2%**	**0.85**	**57.7%**	**0.68**	**67.0%**
Sign-OPT (LR)	1.79	30.5%	1.22	43.2%	0.99	51.6%
Sign-OPT (HR*)	1.58	32.6%	1.10	45.3%	0.90	54.7%
Sign-OPT (HR)	**0.57**	**66.3%**	**0.40**	**80.0%**	**0.32**	**86.3%**
Sign-OPT (HR†*)	5.07	20.8%	3.30	29.2%	2.33	34.4%
Sign-OPT (HR†)	**1.44**	**34.4%**	**0.95**	**51.0%**	**0.74**	**59.4%**

Ablation Study. We have included an ablation study in Figure 5 by disabling our improved gradient estimation of the median filtering defense. When our improved estimation is diabled, HR attacks are impractical and converge significantly slower. We have also highlighted some of the comparisons in Table 3.

5.3.2. Randomized Filtering

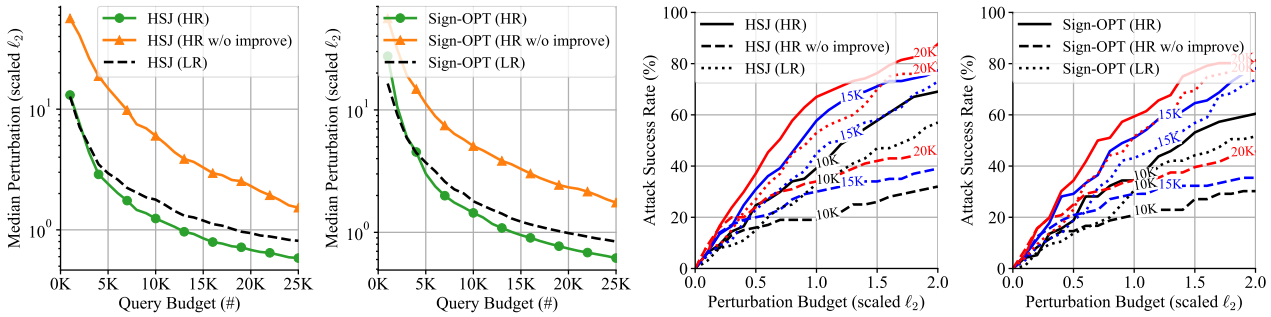
Defense Details. The randomized filtering defense (Quiring et al., 2020) sanitizes the input image by applying the random filter $\boldsymbol{k}_{\text{rnd}}$, which randomly picks a pixel from $\boldsymbol{w}$. To evade this defense, an adaptive attacker has to set every pixel in a window to the desired value. This defense was regarded as robust but lacked a rigorous argument.

Discussion. We adopt the circumventing strategy in Section 4.3 to analyze this defense. Specifically, we find that the expectation of randomized filtering scaling can be formulated as a uniform scaling procedure:

$$\mathbb{E}_{h\sim\mathcal{H}}(g \circ h)(X) = X \star \boldsymbol{k}_{\text{u}}, \tag{5}$$

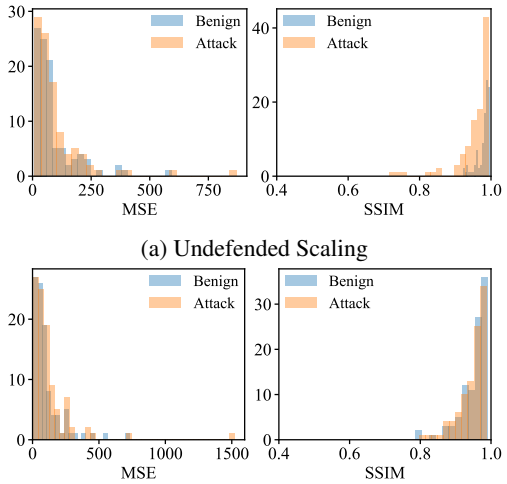
where $\star$ denotes 2D convolution and $\boldsymbol{k}_{\text{u}}$ is the uniform scaling kernel (detailed in Appendix F). This observation shows that the randomized filter can process the input such that any followed scaling function will be made uniform in expectation, thereby leaving no space to hide perturbation stealthily. We discuss effective defenses and robust scaling algorithms with more details in Appendices A.3.4 and A.3.5.

Evaluation. We are not able to circumvent the randomized filtering defense to improve attacks. This result is verified in the white-box setting where we use PGD (Madry et al., 2018) to attack the entire pipeline (details in Appendix G.2). It shows that the randomized filtering defense can properly address the actual weakness of scaling functions.



(a) HSJ Attack (b) Sign-OPT Attack (c) HSJ Attack (d) Sign-OPT Attack

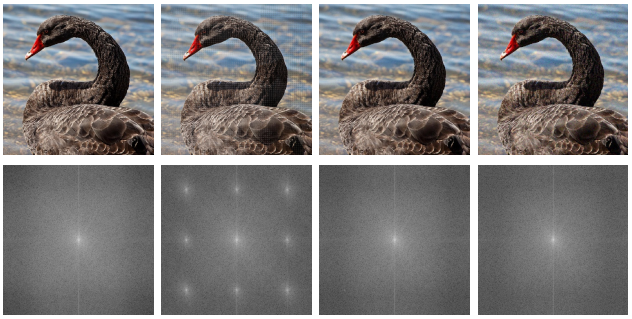
Figure 5: Comparison of our HR HSJ and Sign-OPT attacks with their LR primitives *under the median filtering defense*. (a) and (b) compare the perturbation under different query budgets. (c) and (d) compare the attack success rate under different perturbation and query budgets. We include an ablation study that disables our improved gradient estimation for the median.



(a) Undefended Scaling

(b) Median-defended Scaling

Figure 6: The histogram of distortions as measured by the unscaling defense. Benign images and adversarial examples produced by our HR black-box attacks are indistinguishable.



(a) Benign (b) Image-Scaling Attack (c) HR-HSJ (undefended) (d) HR-HSJ (median)

Figure 7: The centered spectrum of benign and adversarial examples. Our HR attacks do not exhibit artifacts like (b).

5.3.3. Spatial Domain Detection

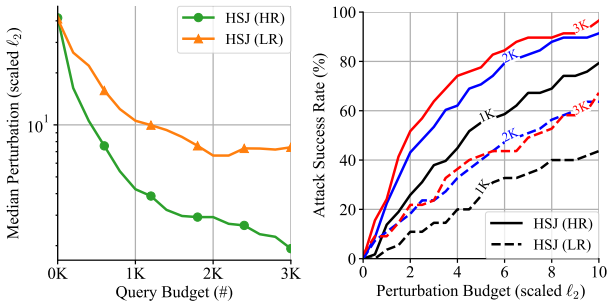
In the spatial domain, Kim et al. (2021) leverage *unscaling* and *min-filtering* to reveal the injected perturbation through processing the input image X with some function T. For example, the unscaling defense considers T as the composition of downscaling and upscaling, which explicitly reveals the hidden perturbation. After that, they quantify the resulting distortion with perceptual metrics like MSE and SSIM. One could model the distortion score as $t(X) := \text{MSE}(X - T(X))$ and employ a threshold-based detector to determine whether the input image X is benign or perturbed. Our evaluation in Figure 6 shows that these defenses cannot detect the hidden perturbation when they are sufficiently small. Results from the minimum-filtering defense are not shown as they show similar observations.

5.3.4. Frequency Domain Detection

In the frequency domain, Kim et al. (2021) examine the number of peaks in the spectrum image, as it assumes injected perturbations manifest as high-frequency and high-energy noise. It applies a low-pass filter (with a predefined threshold) on the input's spectrum image to reveal such peaks. However, our HR attacks in Figure 7 do not have such artifacts. A sophisticated defender could employ learning-based detectors to detect the hidden perturbation; we leave its black-box circumvention to future work.

5.4. Attacking Cloud API

Finally, we conduct black-box attacks on the Tencent Cloud API. This experiment demonstrates that the attacker can also exploit the scaling function in online APIs to improve their attack. For decision-based online attacks, we test 100 ImageNet images, each with 3K queries ($1.18 USD per image). The results are shown in Figure 8 and highlighted in Table 4, where the HR attack significantly outperforms its LR counterpart.



(a) Perturbation (scaled ℓ_2) (b) Attack Success Rate (%)

Figure 8: Comparison of the decision-based LR and HR HSJ attacks on the Tencent Cloud API.

Table 4: Highlighted comparison of the decision-based LR and HR HSJ attacks on Tencent Cloud API. ASR is the attack success rate under perturbation budget $\ell_2 = 2.0$.

Attacks	Query = 1K		Query = 2K		Query = 3K	
	ℓ_2	ASR	ℓ_2	ASR	ℓ_2	ASR
HSJ (LR)	10.57	10.9%	6.64	18.2%	7.42	21.8%
HSJ (HR)	**4.24**	**25.9%**	**2.93**	**43.1%**	**1.92**	**51.7%**

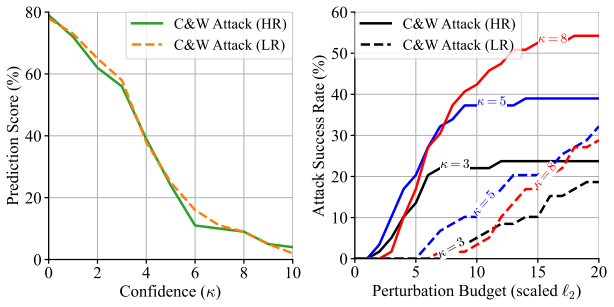
The transfer-based attacks in Figure 9 confirm that C&W attack (Carlini et al., 2019) can achieve significantly higher transferability and success rates than their LR counterparts. In particular, Figure 9a show that HR attacks do not misuse the confidence parameter to overclaim improvements.

6. Discussion

The evaluation of the trustworthiness of ML systems must consider the *interplay* of different vulnerabilities. When weaknesses coexist in the different stages of the ML pipeline, defenders should carefully analyze if these weaknesses could amplify attacks designed to exploit any of them. This work shows that black-box attacks can be made stronger given the knowledge of a vulnerable preprocessing stage, scaling in our case. Further, it shows that defenses, which are effective against the standalone image-scaling attack, can be exploited to make the black-box attack stronger.

To prevent a false sense of security, the defender should also address the *weakness* exploited by the attack rather than the attack itself. Although existing image-scaling defenses can successfully block the image-scaling attack, most of them fail to mitigate the underlying vulnerability of scaling algorithms properly. This leaves an open space for an attacker to jointly exploit other weaknesses in ML, like the adversarial example, as is done in this work.

Limitations. In this paper, we mainly focus on the black-box attacks that estimate gradients and optimize for ℓ_2 norm. There are other attacks that do not directly estimate gradients and optimize for the ℓ_∞ norm, such as RayS (Chen & Gu, 2020). In such cases, it is possible to integrate them by projecting their search direction to the exploitable subspace; for instance, adapting our Equation (2) to optimize the direction instead of the noise. Despite, the benefits would still manifest in ℓ_2 norm; the vulnerability of scaling algorithms mainly improve perceptual quality, so ℓ_∞ norm is inapplicable. Benefits in terms of ℓ_∞ norm would require future work to explore other vulnerabilities that discard a large amount of information in the magnitude of pixel values.



(a) Prediction Score (%) (b) Attack Success Rate (%)

Figure 9: Comparison of the transfer-based LR and HR C&W attacks on the Tencent Cloud API. The reported prediction score is returned by the API and indicates how confident it predicts the input as its ground-truth label.

7. Conclusion

This paper explores the interplay between vulnerabilities of image scaling and ML models in the black-box setting. We propose a novel sampling strategy to make black-box attacks exploit the weakness of scaling functions. With our novel circumvention strategy, we show that 4 out of 5 state-of-the-art defenses, designed to protect the scaling stage, retain weaknesses that enable stronger black-box attacks. The purpose of this work is to raise the concern of threats that jointly exploit different vulnerabilities, whereas current efforts focusing on defending against each vulnerability separately. Further work is necessary to identify and mitigate other threats that jointly target different ML components.

Acknowledgement

We thank all anonymous reviewers for their insightful comments and feedback. This work is partially supported by the DARPA GARD program under agreement number 885000 and the NSF through awards: CNS-1838733, CNS1942014, and CNS-2003129.

References

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 274–283, 2018a. URL http://proceedings.mlr.press/v80/athalye18a.html.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 284–293, 2018b. URL http://proceedings.mlr.press/v80/athalye18b.html.

Ben-Israel, A. and Greville, T. N. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SyZI0GWCZ.

Carlini, N. and Farid, H. Evading deepfake-image detectors with white- and black-box attacks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pp. 2804–2813, 2020. doi: 10.1109/CVPRW50498.2020.00337. URL https://doi.org/10.1109/CVPRW50498.2020.00337.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49. URL https://doi.org/10.1109/SP.2017.49.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. On evaluating adversarial robustness, 2019.

Chen, J. and Gu, Q. Rays: A ray searching method for hard-label adversarial attack. In Gupta, R., Liu, Y., Tang, J., and Prakash, B. A. (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 1739–1747. ACM, 2020. doi: 10.1145/3394486.3403225. URL https://doi.org/10.1145/3394486.3403225.

Chen, J., Jordan, M. I., and Wainwright, M. J. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pp. 1277–1294, 2020. doi: 10.1109/SP40000.2020.00045. URL https://doi.org/10.1109/SP40000.2020.00045.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26, 2017.

Cheng, M., Le, T., Chen, P., Zhang, H., Yi, J., and Hsieh, C. Query-efficient hard-label black-box attack: An optimization-based approach. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJlk6iRqKX.

Cheng, M., Singh, S., Chen, P. H., Chen, P., Liu, S., and Hsieh, C. Sign-opt: A query-efficient hard-label adversarial attack. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SklTQCNtvS.

Dumoulin, V. and Visin, F. A guide to convolution arithmetic for deep learning. *CoRR*, abs/1603.07285: 6–8, 2016a. URL http://arxiv.org/abs/1603.07285.

Dumoulin, V. and Visin, F. A guide to convolution arithmetic for deep learning. *CoRR*, abs/1603.07285: 6–8, 2016b. URL http://arxiv.org/abs/1603.07285.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., and Holz, T. Leveraging frequency analysis for deep fake image recognition. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pp. 3247–3258, 2020. URL http://proceedings.mlr.press/v119/frank20a.html.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2142–2151. PMLR, 2018. URL http://proceedings.mlr.press/v80/ilyas18a.html.

Kim, B., Abuadbba, A., Gao, Y., Zheng, Y., Ahmed, M. E., Nepal, S., and Kim, H. Decamouflage: A framework to detect image-scaling attacks on CNN. In *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2021, Taipei, Taiwan, June 21-24, 2021*, pp. 63–74. IEEE, 2021. doi: 10.1109/DSN48987.2021.00023. URL https://doi.org/10.1109/DSN48987.2021.00023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Li, H., Xu, X., Zhang, X., Yang, S., and Li, B. QEBA: query-efficient boundary-based blackbox attack. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 1218–1227, 2020a. doi: 10.1109/CVPR42600.2020.00130. URL https://doi.org/10.1109/CVPR42600.2020.00130.

Li, H., Li, L., Xu, X., Zhang, X., Yang, S., and Li, B. Nonlinear projection based gradient estimation for query efficient blackbox attacks. In Banerjee, A. and Fukumizu, K. (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3142–3150. PMLR, 2021. URL http://proceedings.mlr.press/v130/li21f.html.

Li, J., Ji, R., Liu, H., Liu, J., Zhong, B., Deng, C., and Tian, Q. Projection & probability-driven black-box attack. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 359–368. Computer Vision Foundation / IEEE, 2020b. doi: 10.1109/CVPR42600.2020.00044. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Li_Projection__Probability-Driven_Black-Box_Attack_CVPR_2020_paper.html.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I., and Edwards, B. Adversarial robustness toolbox v1.6.0. *CoRR*, 1807.01069, 2018. URL https://arxiv.org/pdf/1807.01069.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In Karri, R., Sinanoglu, O., Sadeghi, A., and Yi, X. (eds.), *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pp. 506–519. ACM, 2017. doi: 10.1145/3052973.3053009. URL https://doi.org/10.1145/3052973.3053009.

Quiring, E., Klein, D., Arp, D., Johns, M., and Rieck, K. Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pp. 1363–1380, 2020. URL https://www.usenix.org/conference/usenixsecurity20/presentation/quiring.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 779–788, 2016. doi: 10.1109/CVPR.2016.91. URL https://doi.org/10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 91–99, 2015. URL https://proceedings.

neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Li, F. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 4278–4284, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.

Tu, C., Ting, P., Chen, P., Liu, S., Zhang, H., Yi, J., Hsieh, C., and Cheng, S. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 742–749. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.3301742. URL https://doi.org/10.1609/aaai.v33i01.3301742.

Xiao, Q., Chen, Y., Shen, C., Chen, Y., and Li, K. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pp. 443–460, 2019. URL https://www.usenix.org/conference/usenixsecurity19/presentation/xiao.

Zhang, J., Li, L., Li, H., Zhang, X., Yang, S., and Li, B. Progressive-scale boundary blackbox attack via projective gradient estimation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12479–12490. PMLR, 2021. URL http://proceedings.mlr.press/v139/zhang21l.html.

Supplementary Materials: The Interplay Between Vulnerabilities in Machine Learning Systems

A. Background of Image Scaling

In this section, we provide additional details of the image-scaling attack and the formulation of scaling algorithms.

A.1. Image Scaling

The scaling procedure $g(\cdot)$ resizes a high-resolution (HR) source image $X \in \mathbb{H} := [0,1]^{m \times n}$ to the low-resolution (LR) output image $\boldsymbol{x} \in \mathbb{L} := [0,1]^{p \times q}$. The overall scaling ratio is defined as $\beta = \min\{\beta_h, \beta_v\}$, where $\beta_h = n/q$ and $\beta_v = m/p$ are the scaling ratios in two directions. In this paper, we only consider downscaling where $\beta > 1$.

The scaling function can be implemented in different ways; we review two formulations that facilitate our analysis. Both formulations indicate that the standard scaling function is a linear operation, thus the post-scaling space $\mathbb{L}$ can be viewed as a subspace of the pre-scaling space $\mathbb{H}$.

Matrix Multiplication. Xiao et al. (2019) conduct an empirical analysis of common scaling functions. They represent image scaling as matrix multiplications:

$$\boldsymbol{x} = g(X) = L \times X \times R, \tag{6}$$

where $L \in \mathbb{R}^{p \times m}$ and $R \in \mathbb{R}^{n \times q}$ are the two constant-coefficient matrices determined by the applied scaling function. They also provide an efficient strategy to deduce approximations of these matrices from given implementations.

Convolution. Quiring et al. (2020) interpret scaling as a convolution[2] between the source image X and a fixed linear kernel $\boldsymbol{k}$ determined by the scaling algorithm:

$$\boldsymbol{x} = g(X) = X \star \boldsymbol{k}, \tag{7}$$

where $\star$ denotes the 2D convolution with proper padding and stride size to match the desired output shape.

A.2. Image-Scaling Attacks

The scaling attacks (Xiao et al., 2019; Quiring et al., 2020) target *only* the scaling procedure in an ML system pipeline. They demonstrate that an attacker can exploit the scaling procedure to compromise an arbitrary downstream ML model.

[2]More precisely, this should be cross-correlation (Dumoulin & Visin, 2016a). But we will use the term convolution for consistency.

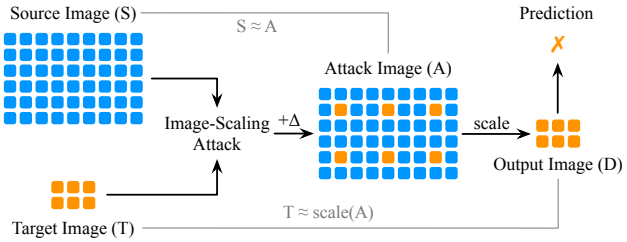


Figure 10: Illustration of the image-scaling attack (Quiring et al., 2020). The HR adversarial example A looks similar to the clean image S, but changes into a target image T after scaling. The color indicates pixels from different images.

Figure 10 illustrates the pipeline of this attack. An adversary computes the attack image A by adding imperceptible perturbations Δ to the source image S, such that it becomes similar to a target image T (with a different label from S) after scaling, thereby fooling the downstream classifier. They formulate the attack as a quadratic optimization problem:

$$\min \|\Delta\|_2^2 \quad \text{s.t. } \|g(S+\Delta) - T\|_\infty \leq \epsilon, \tag{8}$$

where the attack image $A := S + \Delta$ satisfies the box constraint $A \in [0,1]^{m \times n}$.

Besides empirical attacks, Quiring et al. (2020) conduct an in-depth analysis of common scaling algorithms and corresponding convolution kernels used in Equation (7). They identify the use of non-uniform kernels of a fixed width as the root cause for scaling attacks. Such kernels assign higher weights to a small set of *vulnerable pixels* in the source image. For example, in Figure 10, the attacker only needs to modify a few vulnerable (orange) pixels in the source image to change the scaling output completely.

The scaling attack works under the black-box setting. It only needs hundreds of decision-only queries to deduce the fixed scaling algorithm in an ML system (Xiao et al., 2019).

A.3. Image-Scaling Defenses

Researchers proposed several add-on defenses against the scaling attack; these defenses fall into two categories: prevention and detection defenses. We review five state-of-the-art defenses as summarized in Table 5.

Defense	Type	Technique
Median (Quiring et al., 2020)	Pre-processing	Apply median filtering in each window to remove injected perturbation.
Randomized (Quiring et al., 2020)	Pre-processing	Randomly sample pixels in each window to remove injected perturbation.
Unscaling (Kim et al., 2021)	Detection	Down-scale and then up-scale an image to reveal and detect perturbation.
Min-filtering (Kim et al., 2021)	Detection	Apply minimum filtering to reveal injected perturbation.
Spectrum (Kim et al., 2021)	Detection	Identify more-than-one peaks in the centered spectrum.

Table 5: Techniques used by recent scaling defenses.

A.3.1. Prevention Defenses

Quiring et al. (2020) propose the only two prevention defenses, i.e., median and randomized filtering. Both defenses apply filtering operations to sanitize the input image before scaling. Specifically, they reconstruct each vulnerable pixel by a median or a randomly picked pixel within a sliding window. As a result, the attacker has to perturb a significantly larger number of pixels within a window to evade these defenses. Finally, they claim that the median filtering defense is more practical; the randomized filtering defense could hurt the downstream classifier's performance. We analyzed these defenses in more detail in Section 5.3.

A.3.2. Detection Defenses

Kim et al. (2021) propose three detection defenses using spatial and frequency transformations: unscaling, minimum-filtering, and centered spectrum. These transformations result in discernible differences when applied to benign and attack images. The unscaling invokes downscale and upscale operations sequentially to reveal the injected image. If the input image is benign, this procedure should reveal a similar image to the original input one. In case of an attack image, this sequence would reveal a different image. Similarly, minimum filtering reveals such differences using the minimum filter operation. By measuring this difference, they construct a threshold-based detector using mean squared error (MSE) and structural similarity index (SSIM). They also notice that the attack perturbation manifests as high-energy and high-frequency noise, which is detectable by examining the spectrum image. We analyzed these defenses in more detail in Section 5.3.

A.3.3. Infer the Knowledge of Defenses

One can easily infer the knowledge of deployed scaling function and defense even in the black-box setting. Existing image-scaling attacks (Xiao et al., 2019) brute-force the scaling function g and the network's input space $\mathbb{L}$ with hundreds of black-box queries. They run image-scaling attacks with all combinations of standard scaling functions and input sizes until succeed. Note that this inferred knowledge can be reused in the following attacks, as this setting is typically fixed for a deployed ML model. Finally, one can easily extend this method to infer the knowledge of the defense h, as the number of defense parameters is also limited.

A.3.4. Robust Scaling Algorithms

Besides the above add-on defenses, Quiring et al. (2020) identify several scaling algorithms that are naturally robust to the scaling attack. These algorithms are robust because they use either uniform kernels or dynamic kernel widths. For instance, the area (i.e., uniform) scaling algorithm convolves the input image (of scaling ratio β) with a uniform kernel $\boldsymbol{k}_{\mathrm{u}}$ of size $\beta \times \beta$, where each entry of $\boldsymbol{k}_{\mathrm{u}}$ is set to $1/\beta^2$ – the kernel considers each pixel in the window equally. Thus, the attacker cannot find vulnerable pixels to inject another image stealthily.

Ideally, such algorithms should be part of the ML pipeline, but the default scaling algorithm in common ML frameworks is not robust (Quiring et al., 2020). Thus, switching to a different (robust) algorithm faces compatibility issues, such as changing dependent libraries, performance degradation, and even model retraining. As such, deployed ML systems would prefer add-on defenses that can easily fit as plugin modules (Kim et al., 2021). However, our discussion and evaluation in Section 5.3 show that ML systems need to avoid such add-on defenses; they should deploy scaling algorithms that are robust by design.

A.3.5. Effective Image-Scaling Defenses

We discuss what it means for a scaling defense to be effective (or "robust" under the terminology from Quiring et al. (2020)). Prevention defenses are effective if images do not change their appearance after the defended scaling, such that the attacker cannot hide a large perturbation stealthily. Since a defended scaling procedure $g \circ h$ can be viewed as a new scaling function, we note that it must also satisfy the argument from (Quiring et al., 2020) about robust scaling algorithms (refer Appendix A.3.4). This simple observation indicates that **an effective prevention defense should process the input, such that the followed scaling function can weight all pixels uniformly**. As for detection defenses, they are effective if they can detect the attack with acceptable false acceptance and rejection rates.

B. Integration with Black-box Attacks

In this section, we provide the detailed algorithm of our improved HR black-box attacks.

B.1. High-resolution HSJ Attack

Algorithm 2 High-Resolution HSJ Attack (Simplified)

Require: Scaling function g, classifier F, an image $X \in \mathbb{H}$, iterations T, other parameters for HSJ attack.
Ensure: Perturbed image $X_t \in \mathbb{H}$.
Initialize $\tilde{X}_0$ such that $F(\tilde{X}_0) \neq F(X)$.
for t in $1, 2, ..., T$ **do**
▷ Binary Search
Find X_t near the boundary between X and $\tilde{X}_{t-1}$.
▷ **Scaling-aware Noise Sampling (Section 4.1)**
Sample unit vectors $\{U_1, U_2, ...\}$ using Algorithm 1.
▷ Gradient-direction Estimation
Estimate gradient direction $\boldsymbol{g}$ with $\{U_1, U_2, ...\}$.
▷ Update Perturbed Image
Search the step size ξ.
Set $\tilde{X}_t \leftarrow X_t + \xi \cdot \boldsymbol{g}$.
end for
Find X_t near the boundary between X and $\tilde{X}_{t-1}$.
Output X_t.

B.2. High-resolution Sign-OPT Attack

Algorithm 3 High-Resolution SignOPT Attack (Simplified)

Require: Scaling function g, classifier F, an image $X \in \mathbb{H}$, iterations T, other parameters for Sign-OPT attack.
Ensure: Adversarial direction $\boldsymbol{\theta}_t$.
Initialize adversarial direction $\boldsymbol{\theta}_0$.
for t in $1, 2, ..., T$ **do**
▷ **Scaling-aware Noise Sampling (Section 4.1)**
Sample unit vectors $\{U_1, U_2, ...\}$ using Algorithm 1.
▷ Gradient-direction Estimation
Estimate a better gradient $\hat{\boldsymbol{g}}$ with $\{U_1, U_2, ...\}$.
▷ Update adversarial direction
Set $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta \cdot \hat{\boldsymbol{g}}$.
Search a point near the boundary along $\boldsymbol{\theta}_t$.
end for
Output $\boldsymbol{\theta}_t$.

C. Circumventing Median Filtering Defense

In this section, we provide more details of our circumvention of the median filtering defense, including its precise formulation, technical implementation, and additional empirical evaluations.

C.1. Improve Gradient Estimation for Median

For any input sequence $\boldsymbol{z} \in [0, 1]^n$, the improved median function can be written as

$$\text{improved-median}(\boldsymbol{z}) \coloneqq \frac{\sum_{i=1}^{n} z_i \cdot \omega_i}{\sum_{i=1}^{n} \omega_i}, \tag{9}$$

where $\boldsymbol{\omega} \in \mathbb{R}^n$ is the weighting vector.

A useful weighting vector should satisfy two important properties: (1) it proportionally extends the gradient to non-median values; (2) it limits the number of changed values to mitigate the perturbation. We satisfy these two properties through quantile bounding and the absolute deviation to median, which define the weight as

$$\omega_i \coloneqq (1 - |z_i - \text{median}(\boldsymbol{z})|) \cdot \mathbb{1}\{z_{(a)} \leq z_i \leq z_{(b)}\}, \tag{10}$$

where $z_{(a)}, z_{(b)}$ are the a-th and b-th quantile of scalar values in $\boldsymbol{z}$. We set (a, b) to $(0.2, 0.8)$ based on an empirical evaluation in Appendix C.2. Intuitively, values that deviate more from the median are assigned smaller gradients, and the total number of changed values is limited if all values are close to the median.

We note that, however, the above approximation of median function may not be optimal; we leave better-optimized approximations as the future work.

C.2. Approximation of Median's Gradient

In Appendix C.1, we approximate the median function by "trimmed and weighted average" to provide a useful gradient for black-box attacks. To this end, we introduce the weight with quantile bounding, as defined in Equation (10). Here, we provide empirical evaluations of different choices of the quantile position a and b using our HR HSJ attack.

Figure 11 shows that constrained bounds can result in suboptimal performance, such as $(0.4, 0.6)$ and $(0.3, 0.7)$. In contrast, relaxed bounds could obtain better performance, such as $(0.2, 0.8)$ and $(0.1, 0.9)$. We finally choose $(0.2, 0.8)$ as it obtains better performance when given lower query budgets (5K) and higher budgets (25K).

C.3. Formalizing the Median Filtering Defense

Our circumvention strategy in Section 4.2 describes the high-level idea of improving the gradient estimation of the median *function*. However, formulating the median filtering *defense* is not straightforward because it is originally proposed as a *selective* operation that only modifies vulnerable pixels, to which the scaling function gives high weights.

We will first show how to formulate the selective filtering defense as a *masked pooling layer* with a boolean mask that represents vulnerable pixels. After that, we show a simple strategy to identify these vulnerable pixels.

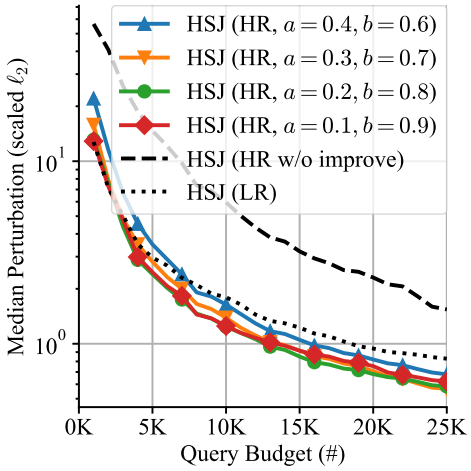


Figure 11: The performance of our HR HSJ (with improved median) under the median filtering defense and the LR HSJ. We show different quantile bounds in Equation (10).

Masked Pooling Layer. We describe the pooling layer as a convolution, as the pooling layer works like a discrete convolution but replaces the linear kernel with some other function (Dumoulin & Visin, 2016b). This allows us to represent the defense as

$$h(X) \coloneqq p(X) \cdot \texttt{mask} + X \cdot (1 - \texttt{mask}), \qquad (11)$$

where p is the pooling function, `mask` is a boolean mask with 1 denoting vulnerable pixels. The pooling function is given as $p(X) \coloneqq X \star \boldsymbol{k}$, where $\star$ denotes 2D convolution with reflect padding to keep the same shape, and $\boldsymbol{k}$ denotes the filter function determined by the defense. In the case of the median filtering defense, we set $\boldsymbol{k}$ to the improved median function in Equation (9) in the backward pass, and switch to the standard median function in the forward pass.

Identifying Vulnerable Pixels. We then explain how to determine the boolean mask in Equation (11). For any fixed scaling algorithm, we can write the scaling function as a matrix multiplication like Equation (6). By setting all entries in $\boldsymbol{x} \in \mathbb{L}$ to one and solving for $X \in \mathbb{H}$, we have

$$X^* = L^+ \times D \times R^+ \in \mathbb{H}, \qquad (12)$$

where L^+ and R^+ are the pseudo-inverse (Ben-Israel & Greville, 2003) of L and R. Conceptually, this recovers the element-wise weight of each pixel in the source image during scaling. That is, every non-zero entry in X^* indicates a vulnerable pixel in the source image; we thus determine the boolean mask as

$$\texttt{mask} = \mathbb{1}\{X^* \neq 0\}, \qquad (13)$$

where the indicator function $\mathbb{1}$ and operator $\neq$ are all computed in element-wise.

In summary, we have formalized the median filtering defense as a masked pooling layer in Equation (11). This formulation allows us to apply SNS in Section 4.1 with a precise definition of $(h \circ g)$.

D. Experimental Details

In this section, we provide more details of our evaluation setup. We run all experiments on 8 Nvidia RTX 2080 Ti GPUs, each with 11 GB memory.

D.1. Datasets and Models

We use two datasets in our evaluation: ImageNet (Russakovsky et al., 2015) and CelebA (Liu et al., 2015).

ImageNet. We randomly choose 1,000 images larger than 672×672 and downscale them to 672×672. The target model is a ResNet-50 (He et al., 2016) model pre-trained by TorchVision[3], which attains 76.13% Top-1 accuracy and 92.86% Top-5 accuracy on ImageNet. We discard source images that are mis-classified before running the attack.

In the white-box setting, we use a ResNet-50 model adversarially trained by Engstrom et al. (2019), which attains 57.90% Top-1 accuracy on benign inputs and 35.09% Top-1 accuracy under PGD attack with 100 steps and an ℓ_2-norm budget of 3. We only choose correctly classified images to avoid the artifacts of its slightly lower benign accuracy.

CelebA. We randomly choose 1,000 images and rescale their faces to 672×672. For simplicity, we pre-cropped the facial images and directly evaluate LR and HR attacks on such images, but it is straightforward to adopt the complete pre-processing pipeline that includes facial extraction, as this procedure is not randomized. The target model is a pre-trained ResNet-34 model that accepts facial images of size 224×224 and predicts the `Mouth_Slightly_Open` attribute with 92.4% accuracy.

D.2. Image Analysis API

The Tencent Image Analysis API accepts a variety of images and returns Top-5 labels (with probability scores) that best describe the image. This API uses OpenCV's linear scaling as inferred by Xiao et al. (2019).

We define the ground-truth label as the benign input's Top-1 label; we only consider benign inputs whose Top-1 score is above 50%. A successful attack should decrease the true label's score to below 10%. Our attacks did not leverage these scores; they only have access to the final decision. When running transfer-based attacks on this API, we adopt the robust model in Appendix D.1 as the surrogate model;

[3] https://pytorch.org/vision/stable/models.html

attacks on a non-robust model cannot transfer to this API.

D.3. Attacks

We use HSJ (Chen et al., 2020), C&W(Carlini et al., 2019), and PGD (Madry et al., 2018) attacks implemented by Adversarial Robustness Toolbox[4] (Nicolae et al., 2018). For the Sign-OPT attack (Cheng et al., 2020), we use its official implementation[5]. Particularly, we did not change the default parameters used in black-box attacks; all optimization parameters are fixed to the official recommendation.

For the C&W attack, we set the binary search step to 20 with a maximum of 1,000 iterations. The confidence parameter κ is set to $\{0, 1, ..., 10\}$. For the PGD attack, we set the number of steps to 100 with ℓ_2-norm budget $\epsilon = \{1, 2, \dots 20\}$ and step size $0.1 \times \epsilon$.

E. Exploiting Vulnerabilities Sequentially

In Section 5.2, we have shown how to *jointly* exploit vulnerabilities in upstream scaling and downstream classifier. However, we note that it is also possible to *sequentially* exploit these two vulnerabilities. For example, the attacker can deploy the conventional black-box attacks on the downstream model, and leverage the image-scaling attack to hide the adversarial example within its original clean image. This requires solving Equation (8), where T is the standard adversarial example generated by a black-box attack on the downstream model.

However, we note that the sequential attack is suboptimal. This attack only finds an HR image whose downscaled version is close enough to the given LR adversarial example; it cannot guarantee that the obtained HR image is still adversarial after downscaling. As black-box adversarial examples are typically near the decision boundary, the final solution may still lie in the correct label's decision area even if it is close enough to the given adversarial example. This problem is more severe when it is hard to precisely invert the median filtering defense.

Figure 4 compares the performance of our joint attack and the alternative sequential attack (all based on HSJ). The sequential attack injects adversarial examples from LR HSJ to their HR source images. Figure 4a shows that jointly attacking the pipeline uses the query budget more efficiently; the 5K joint attack even beats the 20K sequential attack. Figure 4b shows that the sequential attack is suboptimal under the median defense; it becomes worse when the target adversarial example was obtained with more queries (thus, more sensitive to the imprecise results from image-scaling attacks). We thus focus on the more practical joint attack, which directly optimizes to attack the entire ML system.

Finally, we conjecture that the sequential attack only works when the given target image can induce a misclassification with high probability, such as an image from another class (like the standard image-scaling attack) or an adversarial example generated by the C&W attack with high confidence.

[4] https://github.com/Trusted-AI/adversarial-robustness-toolbox

[5] https://github.com/cmhcbb/attackbox

F. Analyzing Randomized Filtering Defense

In the following arguments, we show that the randomized filtering defense, when viewed jointly with the scaling function, can be regarded as a uniform scaling procedure.

Without loss of generality, we study randomized filtering over a 3×3 window $\boldsymbol{w}$ in the source image S and an arbitrary convolution kernel $\boldsymbol{k}$. We also pad the input properly so the window $\boldsymbol{w}$ is always surrounded by other pixels. Since both the scaling and filtering functions can be written as a convolution, we restate the defended scaling $D = (g \circ h)(S)$ over a window $\boldsymbol{w}$ with output pixel $d_{2,2}$ as

$$
\begin{aligned}
d_{2,2} &= \boldsymbol{w} \star \boldsymbol{k}_{\text{rnd}} \star \boldsymbol{k} \\
&= \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \end{bmatrix} \star f_{\text{rnd}} \star \begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix},
\end{aligned}
$$

where the randomized filter $\boldsymbol{k}_{\text{rnd}}$ randomly picks a pixel from the 3×3 window w with probability $1/9$.

We study the central pixel $w_{2,2}$ and its weight $w'_{2,2}$ during this defended scaling. First, the randomized filtering slides a 3×3 window around each pixel $w_{i,j}$ and randomly changes its value to $w_{2,2}$ with a probability $\Pr[w_{i,j} \leftarrow w_{2,2}] = 1/9$. Second, the scaling algorithm gives the weight $k_{i,j}$ to each pixel $w_{i,j}$. Since the pixel $w_{i,j}$ could hold the value of $w_{2,2}$, the overall weight of $w_{2,2}$ can be described as $\Pr[w'_{2,2} \leftarrow k_{i,j}] = 1/9$. Thus, we can write the expected value of the weight $w'_{2,2}$ as

$$
\mathbb{E}_{\boldsymbol{k}\sim\mathcal{K}}[w'_{2,2}] = \sum_{1\le i,j\le 3} \frac{1}{9} \cdot k_{i,j} = \frac{1}{9}, \tag{14}
$$

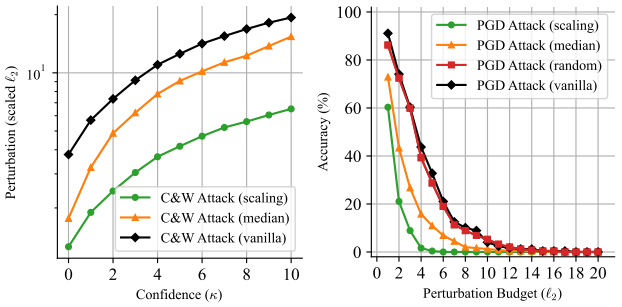
where $\mathcal{K}$ is the filter space determined by $\boldsymbol{k}_{\text{rnd}}$ and we have assumed a normalized scaling kernel $\boldsymbol{k}$. This shows that the pixel $w_{2,2}$ is given a uniform weight in expectation. Extending to other pixels, we have:

$$
\mathbb{E}_{h\sim\mathcal{H}}[(g \circ h)(S)] = S \star \boldsymbol{k}_{\text{u}}, \tag{15}
$$

where $\mathcal{H}$ is the space of defense functions chosen by the randomized filtering defense and $\boldsymbol{k}_{\text{u}}$ is the uniform area scaling kernel defined in Appendix A.3.4.

(a) C&W Vanilla (scaled $\ell_2 = 17.44$) (b) C&W Scaling (scaled $\ell_2 = 5.84$) (c) C&W Medain (scaled $\ell_2 = 14.4$)

Figure 12: Adversarial examples from (a) C&W, (b) HR C&W, and (c) HR C&W under the median filtering defense. The confidence is set to $\kappa = 2$. The shape is 224×224 for (a), and 672×672 for (b) and (c). HR C&W attack produces less perturbation even under the median defense.



(a) Perturbation (scaled ℓ_2) (b) Accuracy (%)

Figure 13: Comparison of HR white-box attacks under different constraints and prevention scaling defenses. Only the randomized filtering defense is robust.

G. High-Resolution White-box Attacks

In this section, we provide details of extending white-box attacks (Carlini et al., 2019; Madry et al., 2018) to the full ML pipeline. This is useful for transfer-based black-box attacks as well as evaluating the worst-case robustness of ML pipeline and image-scaling defenses. Since white-box attacks can easily circumvent defenses using BPDA (Athalye et al., 2018a) and EOT (Athalye et al., 2018b), this evaluation is only for the completeness of this paper and does not claim any technical novelty.

G.1. Formulating White-box Attacks

Formulating white-box attacks is straightforward by viewing the entire ML pipeline as a sequential model. For example, the objective function of C&W attack becomes

$$\begin{aligned} &\min \|\Delta\|_2 + c \cdot (f' \circ g \circ h)(X + \Delta) \\ &\text{s.t. } X + \Delta \in \mathbb{H}, \end{aligned} \tag{16}$$

where f' is the loss function quantifying the confidence of the model f's ground-truth prediction, X is the HR source image, and Δ is the HR adversarial perturbation. Similarly, the objective function of PGD attack becomes

$$\begin{aligned} &\max J\big((f \circ g \circ h)(X + \Delta), y^*\big) \\ &\text{s.t. } \|\Delta\|_2 \leq \epsilon, \end{aligned} \tag{17}$$

where $J(\cdot)$ is the cross entropy loss, y^* is the ground truth label of X, and ϵ is the specified perturbation budget.

G.2. Evaluating Image-scaling Defenses

G.2.1. Pre-processing Defenses

From the perspective of a whole ML pipeline, pre-processing defenses for the scaling function can be circumvented by BPDA (Athalye et al., 2018a) in white-box attacks. However, as we find the gradient of our masked pooling layer formulation in Appendix C.3 to be useful, we will keep the median filtering formulated as Equation (11) for consistency. Adversarial examples produced by HR C&W attacks are shown in Figure 12.

We use HR C&W and PGD attacks in Appendix G.1 to evaluate the robustness under the confidence and perturbation constraint, respectively. Similar to the setting in Section 5.3, we generate two sets of adversarial examples using the LR and HR attacks, respectively.

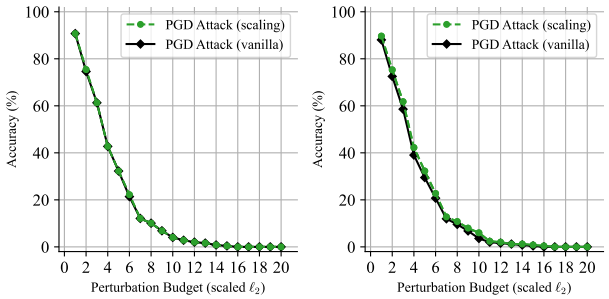
Figure 13 shows the performance of HR white-box attacks when attacking the entire ML system pipeline. Overall, HR attacks are able to gain incentives under no scaling defense or the median filtering defense. In Figure 13a, the HR C&W attack was able to achieve the same confidence with lower perturbation. In Figure 13b, the HR PGD attack was able to decrease more accuracy with the same perturbation budget.

The only exception is the randomized filtering defense in Figure 13b, which successfully protects the ML system's robustness against threats from the scaling procedure. We show that scaling algorithms adopting uniform kernels or dynamic kernel widths (Quiring et al., 2020) are robust as well in Appendix G.2.3.

G.2.2. Detection Defenses

Although we empirically observe that HR white-box attacks evade all existing detection defenses out of the box, we note that a learning-based detector may still work. In that case, one could adaptively evaluate the detector's effectiveness using the following approach. Given any detection function d, we first choose a loss function L so that $L(X + \Delta)$ is minimized when the detection $d(X + \Delta)$ is incorrect. We then add the loss function L as a regularizer to our objective functions. For instance, Equation (17) would become:

$$\begin{aligned} &\max J\big((f \circ g \circ h)(X + \Delta), y\big) - \gamma \cdot L(X + \Delta) \\ &\text{s.t. } \|\Delta\|_2 \leq \epsilon, \end{aligned}$$



(a) CV Area (b) PIL Linear

Figure 14: Compare the performance of vanilla and HR PGD attacks on scaling algorithms that are robust by design. This verifies the robustness of such scaling algorithms.

where γ is the hyper-parameter that controls the weight of the added regularizer. This approach is similar to the attack from Carlini & Farid (2020) against a learning-based deep-fake detector (Frank et al., 2020), which also detects the artifacts of deep-fakes in the spectrum domain. We leave the investigation of such attacks to future work.

G.2.3. Robust Scaling Algorithms

Recall that Quiring et al. (2020) have identified a few scaling algorithms that are robust against the scaling attack (see Appendix A.3.4). We also evaluate their robustness against the HR image-scaling attack. In Figure 14, we report the evaluation of scaling algorithms that adopt uniform kernels (CV Area) or dynamic kernel widths (PIL Linear). As evident from the plots, HR PGD attacks cannot exploit these scaling algorithms to improve the vanilla one. This verifies the robustness of known-robust scaling algorithms in a setting where the attacker jointly targets the whole ML system pipeline.

H. Black-box Adversarial Examples by HSJ

Figure 15 shows more black-box adversarial examples from our HR HSJ attack. HR HSJ attack is able to produce less perturbation than the LR HSJ attack for a given query budget. The same observation holds for median-defended scaling after 200 model queries.

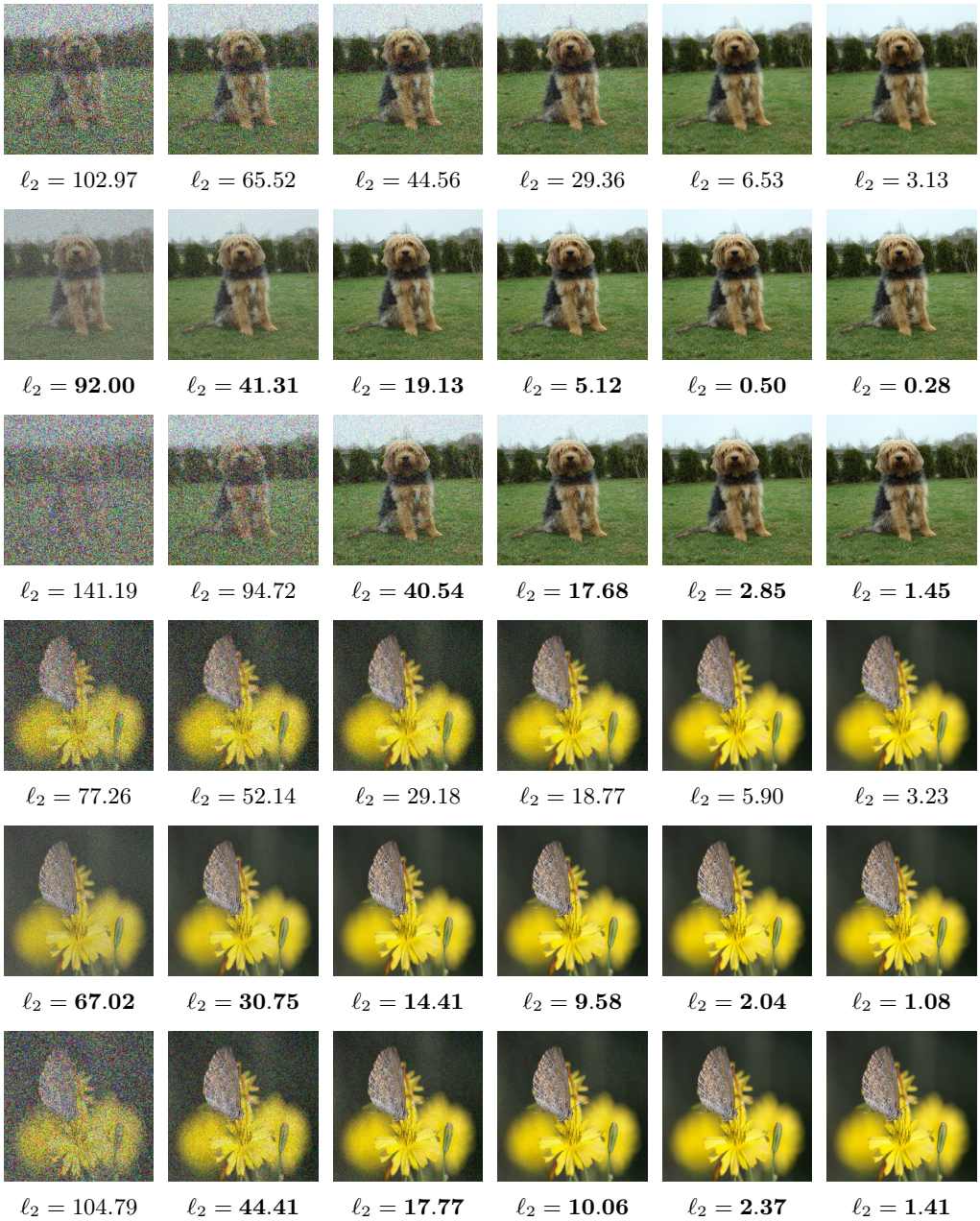


Figure 15: Adversarial examples in the black-box setting. 1st–3rd (4th–6th) rows: examples generated by HSJ, HR HSJ, and HR HSJ under the median filtering defense. 1st–6th columns: examples at 100, 200, 500, 1K, 5K, 10K model queries. Perturbations: the scaled ℓ_2-norm distance to the original image, numbers in bold font denote obtaining less perturbation than the LR HSJ attack. The shape of above images from LR and HR HSJ attacks is 224×224 and 672×672, respectively.