σ-ZERO: GRADIENT-BASED OPTIMIZATION OF ℓ_0-NORM ADVERSARIAL EXAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating the adversarial robustness of deep networks to gradient-based attacks is challenging. While most attacks consider ℓ_2- and ℓ_∞-norm constraints to craft input perturbations, only a few investigate sparse ℓ_1- and ℓ_0-norm attacks. In particular, ℓ_0-norm attacks remain the least studied due to the inherent complexity of optimizing over a non-convex and non-differentiable constraint. However, evaluating adversarial robustness under these attacks could reveal weaknesses otherwise left untested with more conventional ℓ_2- and ℓ_∞-norm attacks. In this work, we propose a novel ℓ_0-norm attack, called σ-zero, which leverages a differentiable approximation of the ℓ_0 norm to facilitate gradient-based optimization, and an adaptive projection operator to dynamically adjust the trade-off between loss minimization and perturbation sparsity. Extensive evaluations using MNIST, CIFAR10, and ImageNet datasets, involving robust and non-robust models, show that σ-zero finds minimum ℓ_0-norm adversarial examples without requiring any time-consuming hyperparameter tuning, and that it outperforms all competing sparse attacks in terms of success rate, perturbation size, and efficiency.

1 INTRODUCTION

Early research has revealed that machine learning models are fooled by adversarial examples, i.e., slightly-perturbed inputs optimized to cause misclassifications (Biggio et al., 2013; Szegedy et al., 2014). The discovery of this phenomenon has, in turn, demanded a more careful evaluation of the robustness of such models, especially when deployed in security-sensitive and safety-critical applications. Most of the gradient-based attacks proposed to evaluate the adversarial robustness of Deep Neural Networks (DNNs) optimize adversarial examples under different ℓ_p-norm constraints. In particular, while convex ℓ_1, ℓ_2, and ℓ_∞ norms have been widely studied (Chen et al., 2018; Croce & Hein, 2021), only a few ℓ_0-norm attacks have been considered to date. The main reason is that finding minimum ℓ_0-norm solutions is known to be an NP-hard problem (Davis et al., 1997), and thus ad-hoc approximations must be adopted to overcome issues related to the non-convexity and non-differentiability of such (pseudo) norm. Although this is a challenging task, attacks based on the ℓ_0 norm have the potential to uncover issues in DNNs that may not be evident when considering other attacks (Carlini & Wagner, 2017b; Croce & Hein, 2021). In particular, ℓ_0-norm attacks, known to perturb a minimal fraction of input values, can be used to determine the most sensitive characteristics that influence the model's decision-making process, offering a different and relevant threat model to benchmark existing defenses and a different understanding of the model's inner workings.

Unfortunately, current ℓ_0-norm attacks exhibit a largely suboptimal trade-off between their success rate and efficiency, i.e., they are either accurate but slow or fast but inaccurate. In particular, the accurate ones use complex projections and advanced initialization strategies (e.g., adversarial initialization) to find smaller input perturbations but suffer from time or memory limitations, hindering their scalability to larger networks or high-dimensional data (Brendel et al., 2019a; Césaire et al., 2021). Other attacks execute faster, but their returned solution is typically less accurate and largely suboptimal (Matyasko & Chau, 2021; Pintor et al., 2021). This results in overestimating adversarial robustness and, in turn, contributes to spreading a *false sense of security*, hindering the development of effective defense mechanisms (Carlini et al., 2019; Pintor et al., 2022). Developing a reliable, scalable, and compelling method to assess the robustness of DNN models against sparse perturbations with minimum ℓ_0 norm remains thus a relevant and challenging open problem.

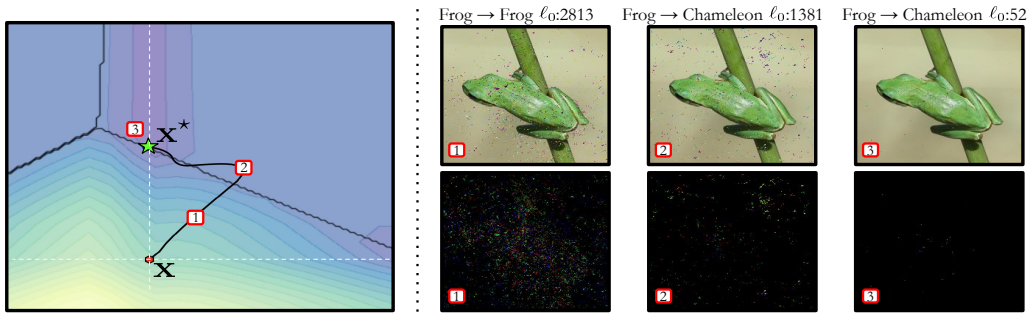

Figure 1: The leftmost plot shows the execution of σ−zero on a two-dimensional problem. The initial point $\mathbf{x}$ (*red dot*) is updated via gradient descent to find the adversarial example $\mathbf{x}^\star$ (*green star*) while minimizing the number of perturbed features (i.e., the ℓ_0 norm of the perturbation). The gray lines surrounding $\mathbf{x}$ demarcate regions where the ℓ_0 norm is minimized. The rightmost plot shows the adversarial images (*top row*) and the corresponding perturbations (*bottom row*) found by σ−zero during the three steps highlighted in the leftmost plot, along with their prediction and ℓ_0 norm.

In this work, we propose a novel ℓ_0-norm attack, named σ−zero, which iteratively promotes the sparsity of the adversarial perturbation by minimizing its ℓ_0 norm (see Figure 1 and Sect. 2). To overcome the limitations of previous approaches, our attack leverages two main technical contributions: (i) a smooth, differentiable approximation of the ℓ_0 norm to enable the minimization of the attack loss via gradient descent; and (ii) an adaptive projection operator that dynamically increases sparsity to further reduce the perturbation size while keeping the perturbed sample in the adversarial region.

Our experiments (Sect. 3) provide compelling evidence of the remarkable performance of σ−zero. We evaluate it on 3 well-known benchmark datasets (i.e., MNIST, CIFAR10, and ImageNet), using 22 different models from Robustbench (Croce et al., 2021) and the corresponding official repositories. We compare the performance of σ−zero against more than 10 competing attacks, totaling almost 450 different comparisons. Our analysis shows that σ−zero outperforms state-of-the-art attacks in terms of both attack success rate and perturbation size (lower ℓ_0 norm), while being also significantly faster (i.e., requiring fewer queries and lower runtime). Our attack also provides some additional advantages: (i) it does not require any sophisticated, time-consuming hyperparameter tuning; (ii) it does not require being initialized from an adversarial input; (iii) it is less likely to fail, i.e., it consistently achieves an attack success rate of 100% for sufficiently-large perturbation budgets, thereby enabling more reliable robustness evaluations (Carlini et al., 2019). We thus believe that σ−zero will foster significant advancements in the development of better robustness evaluation tools and more robust models against sparse attacks. We conclude the paper by discussing related work (Sect. 4), along with the main contributions and future research directions (Sect. 5).

2 σ−ZERO: MINIMUM ℓ_0-NORM ATTACKS

We present here σ−zero, a gradient-based attack that finds minimum ℓ_0-norm adversarial examples.

Threat Model. We assume that the attacker has complete access to the target model, including its architecture and trained parameters, and exploits its gradient for staging white-box untargeted attacks (Carlini & Wagner, 2017b; Biggio & Roli, 2018). This setting is useful for worst-case evaluation of the adversarial robustness of DNNs, providing an empirical assessment of the performance degradation that may be incurred under attack. Note that this is the standard setting adopted in previous work for gradient-based adversarial robustness evaluations (Carlini & Wagner, 2017b; Brendel et al., 2019b; Croce et al., 2021; Pintor et al., 2021).

Problem Formulation. In this work, we seek untargeted minimum ℓ_0-norm adversarial perturbations that steer the model's decision towards misclassification (Carlini & Wagner, 2017b). To this end, let $\mathbf{x} \in \mathcal{X} = [0,1]^d$ be a d-dimensional input sample, $y \in \mathcal{Y} = \{1, \ldots, l\}$ its associated true label, and $f : \mathcal{X} \times \Theta \mapsto \mathcal{Y}$ the target model, parameterized by $\boldsymbol{\theta} \in \Theta$. While f outputs the predicted label, we will also use f_k to denote the continuous-valued output (logit) for class $k \in \mathcal{Y}$. The goal of our attack is to find the minimum ℓ_0-norm adversarial perturbation $\boldsymbol{\delta}^\star$ such that the corresponding adversarial

example $\mathbf{x}^\star = \mathbf{x} + \boldsymbol{\delta}^\star$ is misclassified by f. This can be formalized as:

$$\boldsymbol{\delta}^\star \in \arg\min_{\boldsymbol{\delta}} \quad \|\boldsymbol{\delta}\|_0 \,, \tag{1}$$

$$\text{s.t.} \quad f(\mathbf{x} + \boldsymbol{\delta}, \boldsymbol{\theta}) \neq y \,, \tag{2}$$

$$\mathbf{x} + \boldsymbol{\delta} \in [0,1]^d \,, \tag{3}$$

where $\|\cdot\|_0$ denotes the ℓ_0 norm, which counts the number of non-zero components. The hard constraint in Eq. (2) ensures that the perturbation $\boldsymbol{\delta}$ is valid only if the target model f misclassifies the perturbed sample $\mathbf{x} + \boldsymbol{\delta}$, while the box constraint in Eq. (3) ensures that the perturbed sample lies in $[0,1]^d$.[1] Since the problem in Eqs. (1)-(3) can not be solved directly, we reformulate it as:

$$\boldsymbol{\delta}^\star \in \arg\min_{\boldsymbol{\delta}} \quad \mathcal{L}(\mathbf{x} + \boldsymbol{\delta}, y, \boldsymbol{\theta}) + \frac{1}{d}\hat{\ell}_0(\boldsymbol{\delta}) \tag{4}$$

$$\text{s.t.} \quad \mathbf{x} + \boldsymbol{\delta} \in [0,1]^d \,, \tag{5}$$

where we use a differentiable approximation $\hat{\ell}_0(\boldsymbol{\delta})$ instead of $||\boldsymbol{\delta}||_0$, and normalize it with respect to the number of features d to ensure that its value is within the interval $[0,1]$. The loss $\mathcal{L}$ is defined as:

$$\mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}) = \max\left(f_y(\mathbf{x}, \boldsymbol{\theta}) - \max_{k \neq y} f_k(\mathbf{x}, \boldsymbol{\theta}), 0\right) + \mathbb{I}(f(\mathbf{x}, \boldsymbol{\theta}) = y) \,. \tag{6}$$

The first term in $\mathcal{L}$ represents the logit difference, which is positive when the sample is correctly assigned to the true class y, and clipped to zero when it is misclassified (Carlini & Wagner, 2017b). The second term merely adds 1 to the loss if the sample is correctly classified.[2] This ensures that $\mathcal{L} = 0$ only when an adversarial example is found and $\mathcal{L} \geq 1$ otherwise. In practice, when minimizing the objective in Eq. (4), this loss term induces an *alternate* optimization process between minimizing the loss function itself (to find an adversarial example) and minimizing the ℓ_0-norm of the adversarial perturbation (when an adversarial example is found). It is also worth remarking that, conversely to the objective function proposed by Carlini & Wagner (2017b), our objective does not require tuning any trade-off hyperparameters to balance between minimizing the loss and reducing the perturbation size, thereby avoiding a computationally expensive line search for each input sample.

ℓ_0-norm Approximation. Besides the formalization of the attack objective, one of the main technical advantages of σ-zero is the smooth, differentiable approximation of the ℓ_0 norm, thereby enabling the use of gradient-based optimization. To this end, we first note that the ℓ_0-norm of a vector can be rewritten as $\|\mathbf{x}\|_0 = \sum_{i=1}^{d} \text{sign}(x_i)^2$, and then approximate the *sign* function as $\text{sign}(x_i) \approx x_i / \sqrt{x_i^2 + \sigma}$, where $\sigma > 0$ is a smoothing hyperparameter that makes the approximation sharper as $\sigma \to 0$. This, in turn, yields the following smooth approximation of the ℓ_0 norm:

$$\hat{\ell}_0(\mathbf{x}, \sigma) = \sum_{i=1}^{d} \frac{x_i^2}{x_i^2 + \sigma}, \sigma > 0, \quad \hat{\ell}_0(\mathbf{x}, \sigma) \in [0, d] \,. \tag{7}$$

Adaptive Projection Π_τ. The considered ℓ_0-norm approximation allows optimizing Eq. (4) via gradient descent. However, using such a smooth approximation tends to promote solutions that are not fully sparse, i.e., with many components that are very close to zero but not exactly equal to zero, thereby yielding inflated ℓ_0-norm values. To overcome this issue, we introduce an adaptive projection operator Π_τ that sets to zero the components with a perturbation intensity lower than a given *sparsity threshold* τ in each iteration. The sparsity threshold τ is initialized with a starting value τ_0 and then dynamically adjusted for each sample during each iteration; in particular, it is increased to find sparser perturbations when the current sample is already adversarial, while it is decreased otherwise. The updates to τ are proportional to the step size and follow its annealing strategy, as detailed below.

Solution Algorithm. Our attack, given as Algorithm 1, solves the problem in Eqs. (4)-(5) via a fast and memory-efficient gradient-based optimization. After initializing the adversarial perturbation $\boldsymbol{\delta} = \mathbf{0}$ (line 1), it computes the gradient of the objective in Eq. (4) with respect to $\boldsymbol{\delta}$ (line 3). The gradient is then normalized such that its largest components (in absolute value) equal ± 1 (line 4).

[1] Note that, when the source point $\mathbf{x}$ is already misclassified by f, the solution is simply $\boldsymbol{\delta}^\star = \mathbf{0}$.

[2] While a sigmoid approximation may be adopted to overcome the non-differentiability of the $\mathbb{I}$ term at the decision boundary, we simply set its gradient to zero *everywhere*, without any impact on the experimental results.

Algorithm 1 σ−zero Attack Algorithm.

Input: $\mathbf{x} \in [0,1]^d$, the input sample; y, the true class label; $\boldsymbol{\theta}$, the target model; N, the number of iterations; $\eta_0 = 1.0$, the initial step size; $\sigma = 10^{-3}$, the ℓ_0-norm smoothing hyperparameter; $\tau_0 = 0.3$, the initial sparsity threshold; $t = 0.01$, the sparsity threshold adjustment factor.

Output: $\mathbf{x}^\star$, the minimum ℓ_0-norm adversarial example.

1 $\boldsymbol{\delta} \leftarrow \mathbf{0}$; $\quad \boldsymbol{\delta}^\star \leftarrow \boldsymbol{\infty}$; $\quad \tau \leftarrow \tau_0$; $\quad \eta \leftarrow \eta_0$

2 **for** i in $1, \ldots, N$ **do**

3 $\quad \nabla \mathbf{g} \leftarrow \nabla_{\boldsymbol{\delta}}[\mathcal{L}(\mathbf{x} + \boldsymbol{\delta}, y, \boldsymbol{\theta}) + \frac{1}{d}\hat{\ell}_0(\boldsymbol{\delta}, \sigma)]$ ▷ Gradient Descent for Eq. (4).

4 $\quad \nabla \mathbf{g} \leftarrow \nabla \mathbf{g} / \|\nabla \mathbf{g}\|_\infty$ ▷ Gradient Normalization.

5 $\quad \boldsymbol{\delta} \leftarrow \texttt{clip}(\mathbf{x} + [\boldsymbol{\delta} - \eta \cdot \nabla \mathbf{g}]) - \mathbf{x}$ ▷ Box Constraints.

6 $\quad \boldsymbol{\delta} \leftarrow \Pi_\tau(\boldsymbol{\delta})$ ▷ Adaptive Projection Operator.

7 $\quad \eta = \texttt{cosine\_annealing}(\eta_0, i)$ ▷ Learning Rate Decay.

8 $\quad$ **if** $\mathcal{L}(\mathbf{x} + \boldsymbol{\delta}, y, \boldsymbol{\theta}) \leq 0$: $\tau += t \cdot \eta$, **else** $\tau -= t \cdot \eta$ ▷ Adaptive Adjustment for τ.

9 $\quad$ **if** $\mathcal{L}(\mathbf{x} + \boldsymbol{\delta}, y, \boldsymbol{\theta}) \leq 0 \ \wedge \ \|\boldsymbol{\delta}\|_0 < \|\boldsymbol{\delta}^\star\|_0$: $\boldsymbol{\delta}^\star \leftarrow \boldsymbol{\delta}$

10 **end**

11 **if** $\mathcal{L}(\mathbf{x} + \boldsymbol{\delta}^\star, y, \boldsymbol{\theta}) > 0$: $\boldsymbol{\delta}^\star \leftarrow \infty$

12 **return** $\mathbf{x}^\star \leftarrow \mathbf{x} + \boldsymbol{\delta}^\star$

This stabilizes the optimization by making the update independent from the gradient size, and also makes the selection of the step size independent from the input dimensionality (Rony et al., 2018; Pintor et al., 2021). We then update $\boldsymbol{\delta}$ to minimize the objective via gradient descent while also enforcing the box constraints in Eq. (5) through the usage of the `clip` operator (line 5). We increase sparsity in $\boldsymbol{\delta}$ by zeroing all components lower than the current sparsity threshold τ (line 6), as discussed in the previous paragraph. We then decrease the step size η via cosine annealing (line 7), as suggested by Rony et al. (2018); Pintor et al. (2021), and adjust the sparsity threshold τ accordingly (line 8). In particular, if the current sample is adversarial, we increase τ by $t \cdot \eta$ to promote sparser perturbations; otherwise, we decrease τ by the same amount to promote the minimization of $\mathcal{L}$. The above process is repeated for N iterations while keeping track of the best solution found, i.e., the adversarial perturbation $\boldsymbol{\delta}^\star$ with the lowest ℓ_0 norm (line 9). If no adversarial example is found, the algorithm sets $\boldsymbol{\delta}^\star = \infty$ (line 11). It terminates by returning $\mathbf{x}^\star = \mathbf{x} + \boldsymbol{\delta}^\star$ (line 12).

Remarks. To summarize, the main contributions behind σ−zero are: (i) the use of a smooth ℓ_0-norm approximation, along with the definition of an appropriate objective (Eq. 4), to enable optimizing ℓ_0-norm adversarial examples via gradient descent; and (ii) the introduction of an adaptive projection operator to further improve sparsity during the optimization. Our algorithm leverages also common strategies like gradient normalization and step size annealing to speed up convergence. As reported by our experiments, σ−zero provides a more effective and efficient ℓ_0-norm attack that (i) is robust to different hyperparameter choices; (ii) does not require any adversarial initialization; and (iii) enables more reliable robustness evaluations, being able to find adversarial examples also when the competing attacks may fail (Carlini et al., 2019; Pintor et al., 2022).

3 Experiments

We report here an extensive experimental evaluation comparing σ−zero against 11 state-of-the-art sparse attacks, including both ℓ_0- and ℓ_1-norm attacks. We test all attacks using different settings on 18 distinct models and 3 different datasets, yielding almost 450 different comparisons in total.

3.1 Experimental Setup

Datasets. We consider the three most popular datasets used for benchmarking adversarial robustness: MNIST (LeCun & Cortes, 2005), CIFAR-10 (Krizhevsky, 2009) and ImageNet (Krizhevsky et al., 2012). To evaluate the attack performance, we use the entire test set for MNIST and CIFAR-10 (with a batch size of 32), and a subset of 1000 test samples for ImageNet (with a batch size of 16).

Models. We use a selection of both baseline and robust models to evaluate the attacks under different conditions. We evaluate σ−zero on a vast set of models to ensure its broad effectiveness and expose vulnerabilities that may not be revealed by other attacks (Croce & Hein, 2021). For the MNIST

dataset, we consider two adversarially trained convolutional neural network (CNN) models by Rony et al. (2021), i.e., CNN-DDN and CNN-Trades. These models have been trained to be robust to both ℓ_2 and ℓ_∞ adversarial attacks. We denote them M1 and M2, respectively. For the CIFAR-10 and ImageNet datasets, we employ state-of-the-art robust models from RobustBench (Croce et al., 2021) and the paper's official repositories. For CIFAR-10, we adopt ten models, denoted as C1-C10. C1 (Carmon et al., 2019) and C2 (Augustin et al., 2020) combine training data augmentation with adversarial training to improve robustness to ℓ_∞ and ℓ_2 attacks. C3 (Croce & Hein, 2021) and C4 (Jiang et al., 2023) are ℓ_1 robust models. C5 (Croce et al., 2021) is a non-robust WideResNet-28-10 model. C6 (Gowal et al., 2021) uses generative models to artificially augment the original training set and improve adversarial robustness to generic ℓ_p-norm attacks. C7 (Engstrom et al., 2019) is an adversarially trained model that is robust against ℓ_2-norm attacks. C8 (Chen et al., 2020) is a robust ensemble model. C9 (Xu et al., 2023) is a recently proposed adversarial training defense robust to ℓ_2 attacks. C10 (Addepalli et al., 2022) enforces diversity during data augmentation and combines it with adversarial training. For ImageNet, we consider a pretrained ResNet-18 denoted with I1 (He et al., 2015), and five robust models to ℓ_∞-attacks, denoted with I2 (Engstrom et al., 2019), I3 (Hendrycks et al., 2021), I4 (Debenedetti et al., 2023), I5 (Wong et al., 2020), and I6 (Salman et al., 2020). Lastly, in the appendix, we present two ℓ_0-robust models, C11 (Zhong et al., 2024) and C12 (Zhong et al., 2024), for CIFAR-10, along with two large ℓ_∞-robust models, I7 (Peng et al., 2023) and I8 (Mo et al., 2022), for ImageNet.

Attacks. We compare σ−zero against the following state-of-the-art minimum-norm attacks, in their ℓ_0-norm variants: the Voting Folded Gaussian Attack (VFGA) attack (Césaire et al., 2021), the Primal-Dual Proximal Gradient Descent (PDPGD) attack (Matyasko & Chau, 2021), the Brendel & Bethge (BB) attack (Brendel et al., 2019a), including also its variant with adversarial initialization (BBadv),[3] and the Fast Minimum Norm (FMN) attack (Pintor et al., 2021). We also consider two state-of-the-art ℓ_1-norm attacks as additional baselines, i.e., the Elastic-Net (EAD) attack (Chen et al., 2018) and SparseFool (SF) by Modas et al. (2019). All attacks are set to manipulate the input values independently; e.g., for CIFAR-10, the number of modifiable inputs is $3 \times 32 \times 32 = 3072$.

Hyperparameters. We run our experiments using the default hyperparameters from the original implementations provided in the authors' repositories, *AdversarialLib* (Rony & Ben Ayed) and *Foolbox* (Rauber et al., 2017). We set the maximum number of iterations to $N = 1000$ to ensure that all attacks reach convergence (Pintor et al., 2022).[4] For σ−zero, we set $\eta_0 = 1$, $\tau_0 = 0.3$, $t = 0.01$, and $\sigma = 10^{-3}$, and keep the same configuration for all models and datasets.[5]

Evaluation Metrics. For each attack, we report the Attack Success Rate (ASR) at different values of k, denoted with ASR_k, i.e., the fraction of successful attacks for which $\|\boldsymbol{\delta}^\star\|_0 \leq k$, and the median value of $\|\boldsymbol{\delta}^\star\|_0$ over the test samples, denoted with $\tilde{\ell}_0$.[6] We compare the computational effort of each attack considering the mean runtime (**s**) (per sample), the mean number of queries (**q**) (i.e., the total number of forwards and backwards required to perform the attack, divided by the number of samples), and the Video Random Access Memory (VRAM) consumed by the Graphics Processing Unit (GPU). We measure the runtime on a workstation with an NVIDIA A100 Tensor Core GPU (40 GB memory) and two Intel® Xeo® Gold 6238R processors. We evaluate memory consumption as the maximum VRAM used among all batches, representing the minimum requirement to run without failure.

3.2 Experimental Results

We report the success rate and computational effort metrics of σ−zero against minimum-norm attacks in Table 1 and fixed-budget attacks in Table 3-4. In these tables, we consider the most robust models for each dataset, and we provide the remaining results in Appendix B. Finally, for ImageNet, we narrow our analysis to EAD, FMN, BBadv, and VFGA minimum-norm attacks, as they surpass competing attacks on MNIST and CIFAR-10 in terms of ASR, perturbation size, or execution time.

Effectiveness. The median values of $||\boldsymbol{\delta}^\star||_0$, denoted as $\tilde{\ell}_0$, and the ASRs are reported in Table 1 for all models and datasets. To facilitate comparison, the attacks are sorted from the least to the most effective, on average. In all dataset-model configurations, σ−zero significantly outperforms all the

[3] We utilize the Foolbox DatasetAttack (Foolbox, 2017) for adversarial initialization.

[4] Additional results using only $N = 100$ steps are reported in Appendix B.1.

[5] To show that no specific hyperparameter tuning is required, additional results are reported in Appendix A.2.

[6] If no adversarial example is found for a given $\mathbf{x}$, we set $\|\boldsymbol{\delta}^\star\|_0 = \infty$, as done by Brendel et al. (2019a).

Table 1: Minimum-norm comparison results on MNIST, CIFAR10 and ImageNet with $N = 1000$. For each attack and model (M), we report ASR at $k = 24, 50, \infty$, median perturbation size $\tilde{\ell}_0$, mean runtime s (in seconds), mean number of queries q (in thousands), and maximum VRAM usage (in GB). When VFGA exceeds the VRAM limit, we re-run it using a smaller batch size, increasing its runtime t. We denote those cases with the symbol '$\star$'. Remaining models in Appendix B, Table 6.

Attack	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM
MNIST																
SF	*M1*	6.66	6.76	96.98	469	1.07	0.18	0.06	*M2*	1.03	1.21	91.68	463	2.87	0.86	0.07
EAD		3.83	53.66	100.0	49	0.47	6.28	0.05		2.13	55.57	100.0	48	0.50	6.73	0.05
PDPGD		26.77	74.08	100.0	38	0.23	2.00	0.04		16.91	66.30	100.0	42	0.23	2.00	0.04
VFGA		43.58	82.42	99.98	27	0.05	0.77	0.21		5.00	39.33	99.95	57	0.05	1.33	0.21
FMN		35.90	93.74	100.0	29	0.21	2.00	0.04		50.74	91.84	99.41	24	0.22	2.00	0.04
BB		71.23	97.86	100.0	18	0.90	2.99	0.05		56.53	91.62	100.0	18	0.74	3.71	0.05
BBadv		67.06	91.23	100.0	19	0.77	2.01	0.07		29.17	40.88	100.0	89	0.71	2.01	0.07
σ-zero		**83.79**	**99.98**	100.0	**16**	0.31	2.00	0.04		**98.03**	**100.0**	100.0	**9**	0.31	2.00	0.04
CIFAR-10																
SF	*C1*	18.71	18.77	56.39	3072	11.31	1.40	1.62	*C3*	20.46	24.36	58.29	3072	1.63	0.48	0.66
EAD		16.32	30.38	100.0	90	1.92	5.70	1.47		13.01	13.23	100.0	800	0.94	4.89	0.65
PDPGD		26.84	42.50	100.0	63	0.64	2.00	1.32		22.30	35.13	100.0	75	0.41	2.00	0.59
VFGA		51.06	75.37	99.92	24	0.59	0.78	11.71		28.47	49.98	99.72	51	0.32	1.25	4.44
FMN		48.89	74.70	100.0	26	0.59	2.00	1.31		27.45	48.87	100.0	52	0.24	2.00	0.60
BB		13.27	14.24	14.70	∞	0.63	2.05	1.47		16.88	22.91	27.64	∞	1.04	2.25	0.65
BBadv		65.96	90.57	100.0	16	4.68	2.01	1.64		36.47	72.43	100.0	34	5.28	2.01	0.64
σ-zero		**76.53**	**95.38**	100.0	**11**	0.73	2.00	1.53		**38.60**	**73.02**	**100.0**	**32**	0.43	2.00	0.71
SF	*C2*	19.66	21.22	98.74	3070	3.62	0.46	1.90	*C4*	31.76	43.07	91.14	69	4.32	1.49	0.66
EAD		9.73	11.42	100.0	360	2.53	5.62	1.89		24.21	24.78	100.0	768	1.04	4.99	0.65
PDPGD		28.02	45.15	100.0	55	1.12	2.00	1.8		26.89	42.38	100.0	66	0.40	2.00	0.60
VFGA		39.58	66.50	99.62	34	0.48	0.94	16.53		46.71	69.47	99.83	28	0.25	0.82	4.22
FMN		39.30	71.70	100.0	33	1.08	2.00	1.8		43.06	62.96	100.0	34	0.35	2.00	0.59
BB		38.73	56.78	58.64	33	2.31	2.89	1.89		25.95	27.98	29.50	∞	0.54	2.09	0.65
BBadv		70.07	96.31	100.0	17	3.92	2.01	1.99		53.17	82.46	100.0	22	3.03	2.01	0.65
σ-zero		**74.63**	**97.55**	100.0	**15**	1.41	2.00	1.92		**55.42**	**82.92**	100.0	**20**	0.42	2.00	0.72
ImageNet																
EAD	*I1*	35.4	36.3	100.0	460	4.13	2.69	0.46	*I3*	27.0	28.4	100.0	981	19.25	5.49	1.41
VFGA		57.9	72.5	99.9	14	1.22*	1.08	> 40		46.7	59.5	97.9	31	6.93*	1.98	> 40
FMN		62.6	81.0	100.0	12	0.73	2.00	0.66		49.1	67.7	100.0	25	1.98	2.00	2.30
BBadv		77.5	93.2	100.0	7	231.67	2.01	0.72		64.7	85.5	100.0	14	205.11	2.01	2.41
σ-zero		**82.6**	**95.9**	100.0	**5**	1.18	2.00	0.84		**66.7**	**86.9**	100.0	**13**	2.76	2.00	2.52
EAD	*I2*	46.8	51.0	100.0	42	18.10	5.45	1.42	*I4*	32.8	33.5	100.0	572	11.43	5.34	1.68
VFGA		54.7	63.4	96.7	12	8.21*	2.35	> 40		40.0	46.5	95.5	66	33.88*	3.97	> 40
FMN		57.8	67.0	100.0	9	1.97	2.00	2.30		40.3	47.2	100.0	58	4.28	2.00	2.97
BBadv		71.0	82.3	100	4	182.65	2.01	2.40		46.8	59..8	100.0	31	178.06	2.01	3.07
σ-zero		**76.9**	**87.4**	100.0	**3**	2.75	2.00	2.52		**50.7**	**65.1**	100.0	**23**	5.72	2.00	3.20

considered attacks. Taking the best-performing attack among the fastest competitors as a reference (i.e., FMN), σ-zero is able to find smaller perturbations and higher ASRs in all configurations. In particular, on CIFAR-10, σ-zero reduces the median number of manipulated features from 52 to 32 against the most robust model (C3), with an average reduction of 49% across all models. On ImageNet, this improvement is even more pronounced, with a reduction of up to 58%. In the best case (I4), the median $\|\boldsymbol{\delta}^\star\|_0$ is reduced from 58 to 23, and in the worst case (I2), from 9 to 3. Alternatively, the most competitive attack in finding small perturbations is BBadv, which is significantly slower and requires starting from an already-adversarial input. The ASR_∞ of BB (i.e., without adversarial initialization) indeed decreases with increasing input dimensionality (e.g., CIFAR-10). This occurs because BB often stops unexpectedly before reaching the specified number of steps due to initialization failures; in particular, Table 1 shows that the median perturbation size found by BB is sometimes ∞, as its ASR_∞ is lower than 50%. Although BBadv does not suffer from the same issue, as it leverages adversarial initialization, it is still outperformed by σ-zero. Specifically, σ-zero reduces the ℓ_0 norm of the adversarial examples from 16 to 11 in the best case (C1), while achieving an average improvement of 24% across all dataset-model configurations.

Efficiency. We evaluate the computational effort required to run each attack by reporting in Table 1 the mean runtime s (in seconds), the mean number of queries q issued to the model (in thousands), and the maximum VRAM used. Note that, while the runtime s and the consumed VRAM may

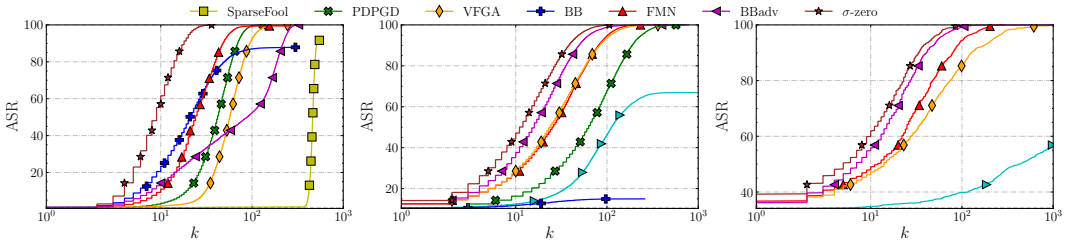

Figure 2: Robustness evaluation curves (ASR vs. perturbation budget k) for M2 on MNIST (*left*), C1 on CIFAR-10 (*middle*), and I1 on ImageNet (*right*).

depend on the attack implementation, the number of queries q counts the total number of forward and backward passes performed by the attack, thus providing a fairer evaluation of the attack complexity. In fact, some attacks perform more than 2000 queries even if $N = 1000$, i.e., they perform more than one forward and one backward pass per iteration (see, e.g., EAD and BB). Other attacks, instead, might use less than 2000 queries as they implement early stopping strategies. The results indicate that σ−zero exhibits similar runtime performance when compared to the fastest algorithms FMN, PDPGD, and VFGA, while preserving higher effectiveness. In contrast, when compared against the BBadv attack, which competes in terms of $\tilde{\ell}_0$, our attack is much faster across all the dataset-model configurations, especially for Imagenet. For example, σ−zero is 10 times faster than BBadv on C4 and 100 times faster on I3 on ImageNet. This confirms that σ−zero establishes a better effectiveness-efficiency trade-off than that provided by state-of-the-art ℓ_0-norm attacks.

Reliability. Complementary to Table 1, we present the robustness evaluation curves in Figure 2 for each attack on M2, C1, and I1. In Appendix B.3, we include similar curves for all other configurations. These curves go beyond the only median statistic and ASR_k, providing further evidence that σ−zero achieves higher ASRs with smaller ℓ_0-norm perturbations compared to the competing attacks. More importantly, the ASR of σ−zero reaches almost always 100% as the perturbation budget grows, meaning that its optimization only rarely fails to find an adversarial example. In Appendix B.1, we further demonstrate that even when the number of iterations is reduced to $N = 100$, σ−zero consistently achieves an ASR_∞ of 100% across all models. This is not observed with other attacks, which often fail when using fewer iterations, thereby increasing the risk of overestimating adversarial robustness. These results reinforce our previous findings, confirming that σ−zero can help mitigate the issue of overestimating adversarial robustness – a crucial aspect to foster scientific progress in defense developments and evaluations (Carlini et al., 2019; Pintor et al., 2022).

Ablation Study. In Table 2 we present an ablation study to evaluate the relevance of σ−zero's components. Our findings indicate that all the non-trivial components in σ−zero are essential for ensuring the effectiveness of the attack. Specifically, we observe that the ℓ_0-norm approximation $\hat{\ell}_0$ (Eq. 7, line 3) leads the optimization algorithm to perturb all input features, albeit with small contributions. The projection operator (line 6) plays a crucial role by significantly decreasing the number of perturbed features, effectively removing the least significant contributions. Furthermore, gradient normalization (line 4) accelerates convergence, enhancing efficiency. Lastly, the adaptive projection operator (line 8) fine-tunes the results, reduces the number of perturbed features, and mitigates the dependency on hyperparameter choices. These results underline the importance of each component in σ−zero, highlighting their contributions to the overall performance of the attack.

Comparison with Fixed-budget Attacks. We complement our analysis by comparing σ−zero with three fixed-budget ℓ_0-norm attacks, i.e., the ℓ_0-norm Projected Gradient Descent (PGD-ℓ_0) attack (Croce & Hein, 2019), the Sparse Random Search (Sparse-RS) attack (Croce et al., 2022),[7] and the Sparse-PGD attack (Zhong et al., 2024). For Sparse-PGD, we consider the implementation with sparse (sPGD_p) and with unprojected (sPGD_u) gradient. In contrast to minimum-norm attacks, fixed-budget attacks optimize adversarial examples within a given maximum perturbation budget k. For a fairer comparison,as done in fixed-budget approaches, we early stop the σ−zero optimization

[7]Sparse-RS is a gradient-free (black-box) attack, which only requires query access to the target model. We consider it as an additional baseline in our experiments, but it should not be considered a direct competitor of gradient-based attacks, as it works under much stricter assumptions (i.e., no access to input gradients).

Table 2: Ablation study on the σ−zero components integrated in Algorithm 1. Columns describe respectively: Gradient normalization factor (line 4); dynamic projection adjustment line 8; projection operator Π_τ (line 6); and the ℓ_0 norm approximation $\hat{\ell}_0$ (line 3).

Model	Normalization	Adaptive τ	Projection	$\hat{\ell}_0$	ASR_{10}	ASR_{50}	ASR	$\|\|\delta\|\|_0$
C10	✓	✓	✓	✓	21.68	**73.02**	100.0	**32**
	✓		✓	✓	**21.89**	71.66	100.0	**32**
		✓	✓	✓	16.81	39.76	100.0	65
			✓	✓	12.95	13.23	100.0	505
				✓	12.95	12.95	100.0	3004
	✓			✓	12.95	12.95	100.0	3070
C5	✓	✓	✓	✓	**37.27**	**82.92**	100.0	**20**
	✓		✓	✓	37.01	79.83	100.0	21
		✓	✓	✓	29.56	52.83	100.0	46
			✓	✓	25.46	32.84	100.0	144
				✓	23.78	23.78	100.0	3064
	✓			✓	23.78	23.78	100.0	3068

Table 3: Fixed-budget comparison results with $N = 1000$ ($N = 2000$ for Sparse-RS) on MNIST and CIFAR-10 at budgets $k = 24, 50, 100$. Columns q_{24} and s_{24} show the average number of queries (in thousands) and the average execution time per sample (in seconds) at $k = 24$.

Attack	**M**	$\textbf{ASR}_{24}$	$\textbf{ASR}_{50}$	$\textbf{ASR}_{100}$	$\textbf{q}_{24}$	$\textbf{s}_{24}$	**VRAM**	**M**	$\textbf{ASR}_{24}$	$\textbf{ASR}_{50}$	$\textbf{ASR}_{100}$	$\textbf{q}_{24}$	$\textbf{s}_{24}$	**VRAM**
MNIST														
PGD-ℓ_0		73.99	99.90	100.0	2.00	0.09	0.04		61.87	94.15	98.50	2.00	0.09	0.04
Sparse-RS		79.54	96.35	99.79	0.83	0.21	0.04		**98.92**	99.96	100.0	0.24	0.07	0.04
sPGD_p	M1	65.55	97.97	99.99	0.46	0.09	0.05	M2	67.92	98.57	99.97	0.92	0.08	0.05
sPGD_u		82.79	99.65	100.0	0.09	0.08	0.05		62.25	98.11	99.99	1.00	0.09	0.05
σ−zero		**83.71**	**99.98**	100.0	0.43	0.02	0.06		98.11	**100.0**	100.0	0.14	0.01	0.06
CIFAR-10														
PGD-ℓ_0		38.18	59.67	87.19	2.00	0.78	1.90		22.99	36.20	67.54	2.00	0.35	0.69
Sparse-RS		72.51	86.59	94.28	0.77	0.36	1.95		30.87	45.65	63.26	1.47	0.28	0.68
sPGD_p	C1	66.37	89.21	99.36	0.74	0.41	2.06	C3	31.82	58.62	93.19	1.39	0.17	0.73
sPGD_u		66.33	91.07	99.75	0.72	0.41	2.06		36.16	70.06	98.07	1.30	0.16	0.73
σ−zero		**77.08**	**95.33**	**99.95**	0.65	0.29	2.07		**38.67**	**73.00**	**98.53**	1.33	0.15	0.75
PGD-ℓ_0		32.41	59.19	89.22	2.00	0.57	2.46		34.35	44.99	68.61	2.00	0.35	0.70
Sparse-RS		59.24	79.81	92.43	1.04	0.35	2.46		49.35	63.01	76.51	1.11	0.37	0.68
sPGD_p	C2	58.91	88.15	99.42	0.89	0.39	2.57	C4	50.41	75.86	97.52	1.02	0.18	0.73
sPGD_u		64.8	93.15	99.92	0.76	0.48	2.56		**55.89**	**84.64**	**99.56**	0.91	0.19	0.73
σ−zero		**75.09**	**97.67**	**100.0**	0.65	0.17	2.68		55.69	82.72	99.07	0.94	0.11	0.75

process as soon as an adversarial example with an ℓ_0-norm perturbation smaller than k is found. In these evaluations, we set $N = 1000$ for σ−zero, PGD-ℓ_0, sPGD_p, and sPGD_u, while using $N = 2000$ for Sparse-RS. Therefore, when using $N = 1000$ steps for σ−zero (which amounts to performing 1000 forward and 1000 backward calls), we set $N = 2000$ steps for Sparse-RS (which amounts to performing 2000 forward calls).[8] Furthermore, to compute the ASR at different k (ASR_k), we separately execute fixed-budget attacks for $k = 24, 50, 100$ features on MNIST and CIFAR-10, and with $k = 150$ features on ImageNet (excluding PGD-ℓ_0 due to computational demands), reporting only the maximum number of queries and execution time across all distinct runs. We report the average query usage at k ($\mathbf{q}_k$) and the average execution time per sample at k ($\mathbf{s}_k$). We report the execution time of $\mathbf{s}_k$ for the smaller k, as it requires, on average, more iterations due to the more challenging problem. The results, shown in Tables 3-4, confirm that σ−zero outperforms competing approaches in 17 out of 18 configurations (see Appendix B.2 for additional results). Only against C4 the fixed-budget attack sPGD_u slightly increases the ASR. The advantages of σ−zero become even

[8] $N = 2000$ is suggested as a lower bound number of iterations to ensure the convergence of Sparse-RS by Croce et al. (2022). Additional results with $N = 5000/10000$ for Sparse-RS can be found in Appendix B.2.

Table 4: Fixed-budget comparison results with $N = 1000$ ($N = 2000$ for Sparse-RS) on ImageNet at budgets $k = 100, 150$. See the caption of Table 3 for further details.

Attack	M	ASR_{100}	ASR_{150}	$\mathbf{q}_{100}$	$\mathbf{s}_{100}$	VRAM	M	ASR_{100}	ASR_{150}	$\mathbf{q}_{100}$	$\mathbf{s}_{100}$	VRAM
ImageNet												
Sparse-RS	I1	89.3	91.5	0.39	0.32	1.29	I2	81.1	84.1	0.53	0.5	4.39
sPGD_p		95.4	98.5	0.31	0.16	1.40		85.6	91.2	0.33	0.64	4.48
sPGD_u		93.6	97.8	0.33	0.12	1.40		82.6	88.7	0.37	0.39	4.49
σ−zero		**99.7**	**100.0**	0.19	0.06	1.79		**94.7**	**97.1**	0.15	0.17	4.90
Sparse-RS	I3	69.1	72.2	0.81	0.62	4.39	I4	45.9	47.4	1.17	1.12	5.72
sPGD_p		85.4	93.4	0.32	0.55	4.49		66.3	74.9	0.73	1.39	5.84
sPGD_u		83.9	92.1	0.35	0.39	4.49		66.0	76.0	0.72	1.01	5.84
σ−zero		**97.7**	**99.6**	0.34	0.37	4.90		**78.8**	**85.8**	0.49	0.70	6.29

more evident when looking at the results on ImageNet, where, on average, it improves the ASR_{100} of 9.6% across all models in Table 4. The results also indicate that early stopping enables σ−zero to save a significant number of queries and runtime while preserving a high ASR. In Appendix B.2, we also report additional comparisons with $N = 2500$ and $N = 5000$, i.e. a more favorable scenario for the competing attacks, confirming that σ−zero remains competitive even at higher budgets.

Summary. Our experiments show that σ−zero: (i) outperforms minimum-norm attacks by improving the success rate and decreasing the ℓ_0 norm of the generated adversarial examples (see Table 1 and Appendix B.1); (ii) is significantly faster and scales easily to large datasets (see Table 1 and Appendix B.1); (iii) is robust to hyperparameter selection, not requiring sophisticated and time-consuming tuning (see Appendix A.2); (iv) does not require any adversarial initialization (see Table 1); (v) provides more reliable adversarial robustness evaluations, consistently achieving 100% ASRs (see Table 1, Figure 2, Appendix B.3); and (vi) remains competitive against fixed-budget attacks even when given the same query budget (Table 3-4).

4 Related Work

Optimizing ℓ_0-norm adversarial examples with gradient-based algorithms is challenging due to non-convex and non-differentiable constraints. We categorize them into two main groups: (i) multiple-norm attacks extended to ℓ_0, and (ii) attacks specifically designed to optimize the ℓ_0 norm.

Multiple-norm Attacks Extended to ℓ_0. These attacks have been developed to work with multiple ℓ_p norms, including extensions for the ℓ_0 norm. While they can find sparse perturbations, they often rely heavily on heuristics in this setting. Brendel et al. (2019a) initialize the attack from an adversarial example far away from the clean sample and optimizes the perturbation by following the decision boundary to get closer to the source sample. In general, the algorithm can be used for any ℓ_p norm, including ℓ_0, but the individual optimization steps are very costly. Pintor et al. (2021) propose the FMN attack that does not require an initialization step and converges efficiently with lightweight gradient-descent steps. However, their approach was developed to generalize over ℓ_p norms, but does not make special adaptations to minimize the ℓ_0 norm specifically. Matyasko & Chau (2021) use relaxations of the ℓ_0 norm (e.g., $\ell_{1/2}$) to promote sparsity. However, this scheme does not strictly minimize the ℓ_0 norm, as the relaxation does not set the lowest components exactly to zero.

ℓ_0-specific Attacks. Croce et al. (2022) introduced Sparse-RS, a random search-based attack that, unlike minimum-norm attacks, aims to find adversarial examples that are misclassified with high confidence within a fixed perturbation budget. On the same track we find Sparse-PGD (Zhong et al., 2024) and PGD-ℓ_0 (Croce & Hein, 2019), white-box fixed-budget alternatives to Sparse-RS. Lastly, Césaire et al. (2021) induces folded Gaussian noise to selected input components, iteratively finding the set that achieves misclassification with minimal perturbation. However, it requires considerable memory to explore possible combinations and find an optimal solution, limiting its scalability.

Overall, current implementations of ℓ_0-norm attacks present a crucial suboptimal trade-off between their success rate and efficiency, i.e., they are either accurate but slow (e.g., BB) or fast but inaccurate (e.g., FMN). This is also confirmed by a recent work that has benchmarked more than 100 gradient-

based attacks (Cinà et al., 2024) on 9 additional robust models. In that open-source benchmark, σ−zero consistently and significantly outperformed all the existing implementations of competing ℓ_0-norm attacks, establishing a performance very close to that of the empirical *oracle* (obtained by ensembling all the attacks tested). In summary, our attack combines the benefits of the two families of attack detailed above, i.e., effectiveness and efficiency, providing the state-of-the-art solution for adversarial robustness evaluations of DNNs when considering ℓ_0-norm attacks.

5 Conclusions and Future Work

In this work, we propose σ−zero, a novel attack aimed to find minimum ℓ_0-norm adversarial examples, based on the following main technical contributions: (i) a differentiable approximation of the ℓ_0 norm to define a novel, smooth objective that can be minimized via gradient descent; and (ii) an adaptive projection operator to enforce sparsity in the adversarial perturbation, by zeroing out the least relevant features in each iteration. σ−zero also leverages specific optimization tricks to stabilize and speed up the optimization. Our extensive experiments demonstrate that σ−zero consistently discovers more effective and reliable ℓ_0-norm adversarial perturbations across all models and datasets while maintaining computational efficiency and robustness to hyperparameters choice. In conclusion, σ−zero emerges as a highly promising candidate to evaluate robustness against ℓ_0-norm perturbations and promote the development of novel robust models against sparse attacks.

Ethics Statement. Based on our comprehensive analysis, we assert that there are no identifiable ethical considerations or foreseeable negative societal consequences that warrant specific attention within the limits of this study. This study will rather help improve the understanding of adversarial robustness of DNNs and identify potential ways to improve it.

Reproducibility. To ensure the reproducibility of our work, we have detailed the experimental setup in Section 3.1, where we describe the datasets, models, and attacks used, along with their respective sources. Additionally, we have provided our source code as part of the supplementary material, which will be made publicly available as open source upon acceptance.

References

Sravanti Addepalli, Samyak Jain, and Venkatesh Babu R. Efficient and effective augmentation strategy for adversarial training. In *NeurIPS*, 2022.

Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in- and out-distribution improves explainability. In *Computer Vision - ECCV 2020 - 16th European Conference*, volume 12371 of *Lecture Notes in Computer Science*, pp. 228–245. Springer, 2020.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, 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*, volume 8190 of *Lecture Notes in Computer Science*, pp. 387–402. Springer, 2013.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019a.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In *Conference on Neural Information Processing Systems (NeurIPS))*, 2019b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (S&P)*, 2017a.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy SP*, pp. 39–57. IEEE Computer Society, 2017b.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian J. Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *CoRR*, abs/1902.06705, 2019.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Conference on Neural Information Processing Systems (NeurIPS))*, 2019.

Manon Césaire, Lucas Schott, Hatem Hajri, Sylvain Lamprier, and Patrick Gallinari. Stochastic sparse adversarial attacks. In *33rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pp. 1247–1254. IEEE, 2021.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pp. 10–17. AAAI Press, 2018.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 696–705. Computer Vision Foundation / IEEE, 2020.

Antonio Emanuele Cinà, Jérôme Rony, Maura Pintor, Luca Demetrio, Ambra Demontis, Battista Biggio, Ismail Ben Ayed, and Fabio Roli. Attackbench: Evaluating gradient-based attacks for adversarial examples. *arXiv preprint arXiv:2404.19460*, 2024.

Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4723–4731, 2019.

Francesco Croce and Matthias Hein. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2201–2211. PMLR, 2021.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks*, 2021.

Francesco Croce, Maksym Andriushchenko, Naman D. Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: A versatile framework for query-efficient sparse black-box adversarial attacks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pp. 6437–6445. AAAI Press, 2022.

Geoff Davis, Stephane Mallat, and Marco Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13:57–98, 1997.

Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2023. URL `https://openreview.net/forum?id=IztT98ky0cKs`.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Foolbox. Datasetattack, 2017. URL `https://foolbox.readthedocs.io/en/stable/modules/attacks.html#foolbox.attacks.DatasetAttack`.

Justin Gilmer, Ryan P. Adams, Ian J. Goodfellow, David Andersen, and George E. Dahl. Motivating the rules of the game for adversarial example research. *CoRR*, abs/1807.06732, 2018.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A. Mann. Improving robustness using generated data. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, pp. 4218–4233, 2021.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV*, pp. 8320–8329. IEEE, 2021.

Yulun Jiang, Chen Liu, Zhichao Huang, Mathieu Salzmann, and Sabine Süsstrunk. Towards stable and efficient adversarial training against l1 bounded adversarial attacks. In *International Conference on Machine Learning*, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005.

Alexander Matyasko and Lap-Pui Chau. PDPGD: primal-dual proximal gradient descent adversarial attack. *CoRR*, abs/2106.01538, 2021. URL https://arxiv.org/abs/2106.01538.

Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. *Advances in Neural Information Processing Systems*, 35:18599–18611, 2022.

Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Conference on computer vision and pattern recognition (CVPR)*, 2019.

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*, pp. 739–740. BMVA Press, 2023. URL http://proceedings.bmvc2023.org/739/.

Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS*, pp. 20052–20062, 2021.

Maura Pintor, Luca Demetrio, Angelo Sotgiu, Ambra Demontis, Nicholas Carlini, Battista Biggio, and Fabio Roli. Indicators of attack failure: Debugging and improving optimization of adversarial examples. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23063–23076. Curran Associates, Inc., 2022.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models, 2017. URL https://github.com/bethgelab/foolbox.

Jérôme Rony, Luiz G. Hafemann, Luiz Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4317–4325, 2018.

Jérôme Rony, Eric Granger, Marco Pedersoli, and Ismail Ben Ayed. Augmented lagrangian adversarial attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV*, pp. 7718–7727. IEEE, 2021.

Jérôme Rony and Ismail Ben Ayed. Adversarial Library. URL https://github.com/jeromerony/adversarial-library.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020.

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 (ICLR)*, 2014.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR*. OpenReview.net, 2020.

Yuancheng Xu, Yanchao Sun, Micah Goldblum, Tom Goldstein, and Furong Huang. Exploring and exploiting decision boundary dynamics for adversarial robustness. In *International Conference on Learning Representations (ICLR)*, 2023.

Xuyang Zhong, Yixiao Huang, and Chen Liu. Towards efficient training and evaluation of robust models against l_0 bounded adversarial perturbations. In *International Conference on Machine Learning ICML*. PMLR, 2024.

A APPENDIX

A.1 ROBUST MODELS

The experimental setup described in this paper (Section 3.1) utilizes pre-trained baseline and robust models obtained from RobustBench (Croce et al., 2021). The goal of RobustBench is to track the progress in adversarial robustness for ℓ_∞- and ℓ_2-norm attacks since these are the most studied settings in the literature. We summarize in Table 5 the models we employed for testing the performance of σ−zero. Each entry in the table includes the label reference from RobustBench, the short name we assigned to the model, and the corresponding clean and robust accuracy under the specific threat model. The robustness of these models is evaluated against an ensemble of white-box and black-box attacks, specifically AutoAttack. We also include in our experiments models trained to be robust against ℓ_1 sparse attacks, i.e., C3 (Croce & Hein, 2021) and C4 (Jiang et al., 2023). Our experimental setup is designed to encompass a wide range of model architectures and defensive techniques, ensuring a comprehensive and thorough performance evaluation of the considered attacks.

Table 5: Summary of Robustbench models used in our experiments. For each model, we report its reference label in Robustbench, its threat model, and the corresponding clean and robust accuracy.

Dataset	Reference	Model	Threat model	Clean accuracy %	Robust accuracy %
CIFAR-10	Carmon2019Unlabeled	C1 (Carmon et al., 2019)	ℓ_∞	89.69	59.53
	Augustin2020Adversarial	C2 (Augustin et al., 2020)	ℓ_2	91.08	72.91
	Standard	C5 (Croce et al., 2021)	-	94.78	0
	Gowal2020Uncovering	C6 (Gowal et al., 2021)	ℓ_2	90.90	74.50
	Engstrom2019Robustness	C7 (Engstrom et al., 2019)	ℓ_∞- ℓ_2	87.03 - 90.83	49.25 - 69.24
	Chen2020Adversarial	C8 (Chen et al., 2020)	ℓ_∞	86.04	51.56
	Xu2023Exploring_WRN-28-10	C9 (Xu et al., 2023)	ℓ_∞	93.69	63.89
	Addepalli2022Efficient_RN18	C10 (Addepalli et al., 2022)	ℓ_∞	85.71	52.48
ImageNet	Standard_R18	I1 (He et al., 2015)	-	76.52	0
	Engstrom2019Robustness	I2 (Engstrom et al., 2019)	ℓ_∞	62.56	29.22
	Hendrycks2020Many	I3 (Hendrycks et al., 2021)	ℓ_∞	76.86	52.90
	Debenedetti2022Light_XCiT-S12	I4 (Debenedetti et al., 2023)	ℓ_∞	72.34	41.78
	Wong2020Fast	I5 (Wong et al., 2020)	ℓ_∞	55.62	26.24
	Salman2020Do_R18	I6 (Salman et al., 2020)	ℓ_∞	64.02	34.96
	Peng2023Robust	I7 (Peng et al., 2023)	ℓ_∞	73.44	48.94
	Mo2022When_Swin-B	I8 (Mo et al., 2022)	ℓ_∞	74.66	38.30

A.2 σ−ZERO: HYPERPARAMETER ROBUSTNESS

To assess the strength and potential limitations of our proposed attack, we conducted an ablation study on its key hyperparameters, τ_0, σ, and t.

The parameter τ_0 governs the initial tolerance threshold in Algorithm 1, which induces sparsity within the adversarial perturbation. The parameter σ defines the approximation quality of $\hat{\ell}_0$ in Eq. (7) compared to the actual ℓ_0 function. Our ablation study, depicted in Figure 3, involved two distinct models: C10 (top row) and I1 (bottom row). We executed the attack on 1000 randomly selected samples from each dataset and recorded the ASR at different perturbation budgets k and the median ℓ_0 norm of the resulting adversarial perturbations. We observe a significant robustness of σ−zero with respect to these two hyperparameters; in particular: (i) the choice of the initial value of τ_0 exerts negligible influence on the ultimate outcome, given that the parameter dynamically adapts throughout the optimization process; and (ii) the selection of σ is not particularly challenging, especially when incorporating the sparsity projection operator.

We also conducted an ablation study on the sparsity threshold adjustment factor t used to adaptively update τ. In the following we keep the default values for $\tau_0 = 0.3$ and $\sigma = 10^{-3}$. We executed the attack on 1000 randomly selected samples against C3 and C4 models and recorded the ASR_{50} and the median l_0 norm of the resulting adversarial perturbations. In Figure 4, we once against observe the robustness of σ−zero to this parameter, yielding similar and effective results when $t \leq 10^{-1}$.

Overall, the ablation study revealed consistent trends across the distinct models and datasets. In all cases, we identified a broad parameter configuration range where our attack maintained robustness to the hyperparameter selection, making hyperparameter optimization for the attacker a swift task. This robustness is further evidenced by the results presented in the experimental comparisons, where our

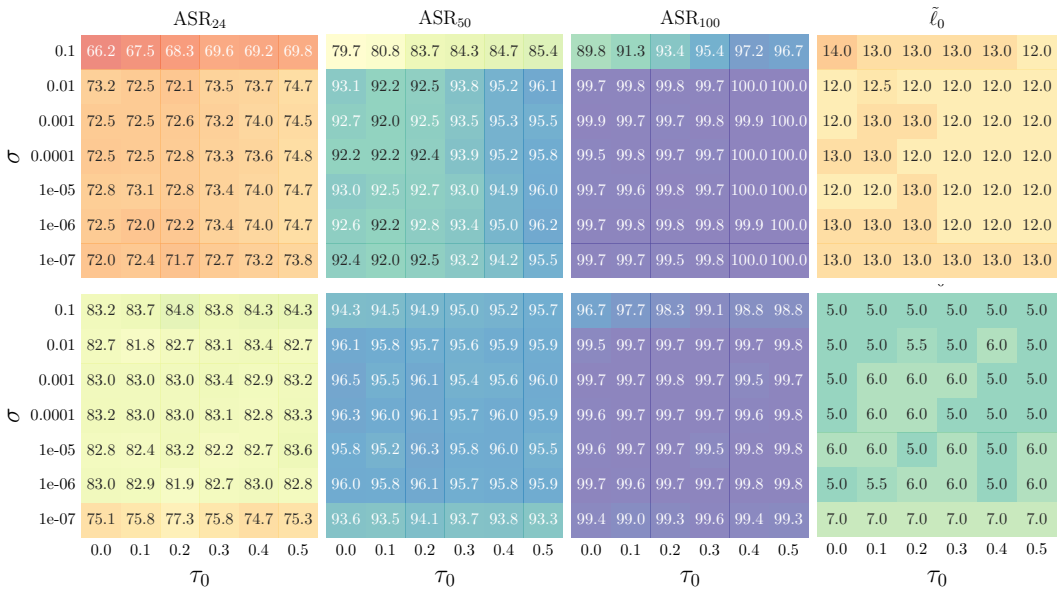

Figure 3: Ablation study on σ (y-axis) and τ_0 (x-axis) for CIFAR-10 C10 (top-row), ImageNet I1, (bottom-row). For each combination, we report the attack success rate at different k and the median ℓ_0 perturbation value.

attack consistently outperforms competing attacks even with a shared hyperparameter configuration across all models.

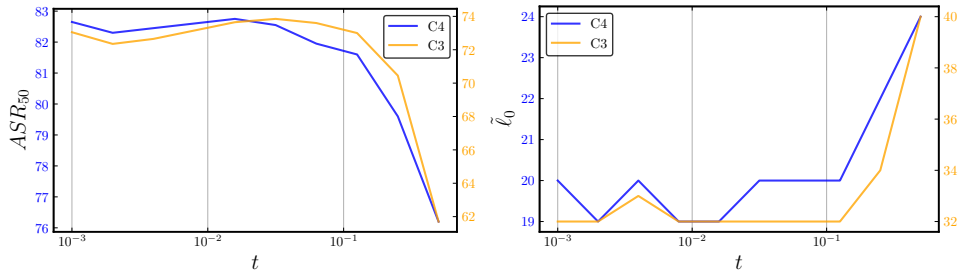

Figure 4: Ablation study on t for CIFAR-10 C3 and C4. For each, we report the attack success rate at the 50 feature budget (*left*) and the median ℓ_0 norm of the adversarial perturbation (*right*).

B Additional Experimental Comparisons

B.1 Comparisons with Minimum-norm Attack

In our experimental setup, we also consider a reduced number of queries, to test whether the attack can also run faster while remaining effective. We thus replicate our experimental comparison involving σ−zero and state-of-the-art sparse attacks while restricting the number of steps to $N = 100$. The results are summarized in Tables 7-9. Compared to the results with $N = 1000$ steps reported in Tables 1 and 6, the ASR of most competitive attacks decreases, while σ−zero remains effective by consistently reaching an ASR of 100%. This shows that σ−zero remains an effective, reliable and fast approach to crafting minimum-norm attacks even with reduced query budgets.

B.2 Comparisons with Fixed-budget Attacks

Fixed-budget attacks, i.e., Sparse-RS (Croce et al., 2022), PGD-ℓ_0 (Croce & Hein, 2019), and Sparse-PGD (Zhong et al., 2024), have been designed to generate sparse adversarial perturbations given a

fixed-budget k, therefore, drawing comparisons with minimum-norm attacks is not a straightforward task. Specifically, in their threat model, the attacker imposes a maximum limit on the number of perturbed features, and the attack then outputs the adversarial example that minimizes the model's confidence in predicting the true label of the sample. However, since the fixed-budget threat model differs from the minimum-norm scenario we consider in this paper, which does not assume a maximum norm value k, we evaluate σ−zero in a fixed-budget fashion by discarding all adversarial perturbations that exceed k. Furthermore, as for fixed-budget attack, we let σ−zero to leverage the input parameter k to early stop the optimization procedure and reduce the number of consumed queries to the target model. Throughout this evaluation, the number of steps taken by Sparse-RS is always doubled compared to the other two white-box attacks, as it does not utilize the backward pass employed by the others. The main paper reports to this end an evaluation of σ−zero in a fixed-budget approach (cf. Tables 3-4). The remaining experiments, involving additional models for CIFAR-10 and ImageNet, are reported in Tables 10-11. Furthermore, to explore the effects of increased iterations on convergence and success rate, we increased the number of iterations up to $N = 10000$ (Tables 12-15), while always doubling the iterations for Sparse-RS. These additional experiments cover the three datasets MNIST, CIFAR-10, and ImageNet, 18 distinct models, and various feature budgets. The results again affirm that, σ−zero consistently outperforms competing approaches or synergizes well with them for a comprehensive robustness assessment.

B.3 Robustness Evaluation Curves

We provide robustness evaluation curves for fixed-budget attacks on a CIFAR-10 model (C3), running each attack multiple times across various perturbation budgets k. The number of iterations is set to $N = 1000$ and $N = 5000$, with Sparse-RS allocating twice the iterations due to its reliance solely on forward passes. The results, depicted in Figure 5, demonstrates that σ−zero consistently outperforms fixed-budget attacks across all perturbation budgets k. Additionally, we present in Figs. 6-7 the robustness evaluation curves depicting the performance of minimum-norm ℓ_0-attacks against all the models analyzed in our paper. These findings reinforce our experimental analysis, explicitly demonstrating that the σ−zero attack consistently achieves higher values of ASR while employing smaller ℓ_0-norm perturbations compared to alternative attacks.

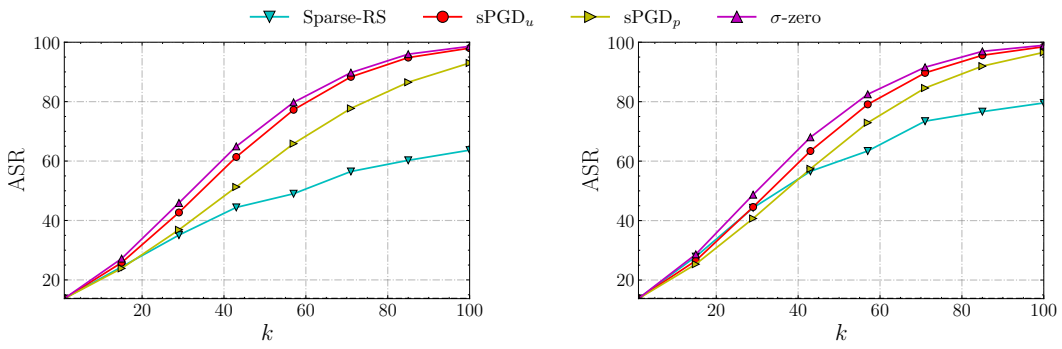

Figure 5: Robustness evaluation curves for fixed-budget attacks on C3. For each budget level k, each attack has been run with 1000 iterations (left-most plot) and 5000 iterations (right-most plot). Sparse-RS has been run with double the iterations as it relies solely on forward calls.

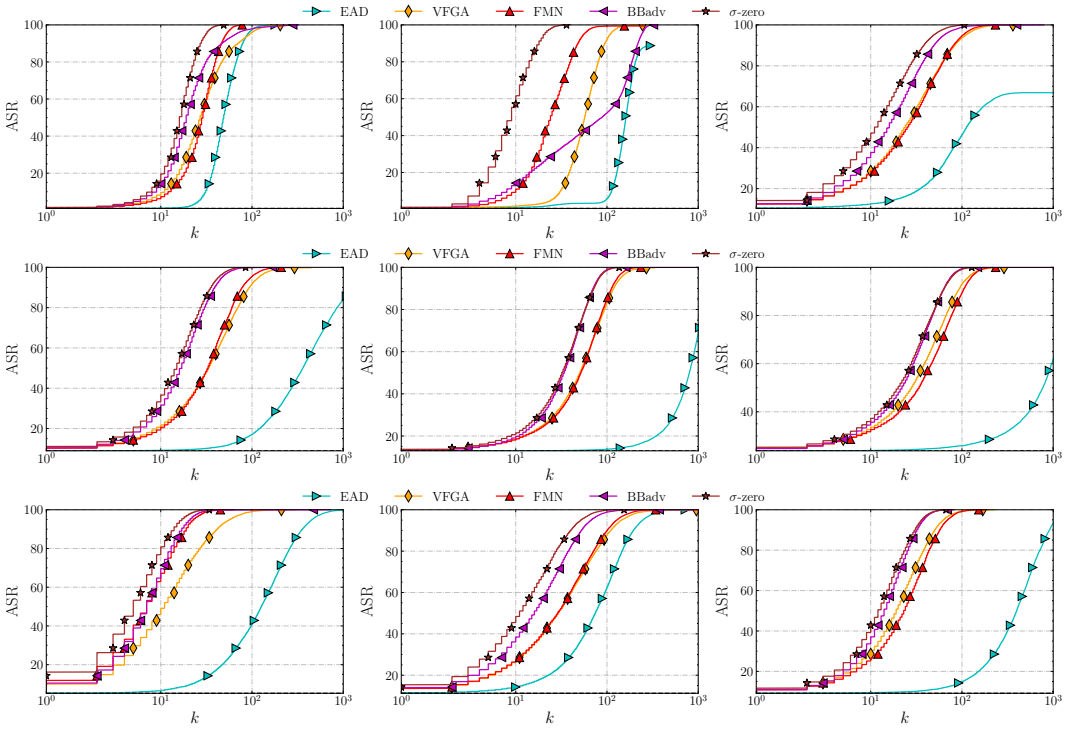

Figure 6: From the leftmost to the rightmost we report the robustness evaluation curves for M1, M2, C1 (top-row), C2, C3, C4 (middle-row) and C5, C6, C7 (bottom-row).

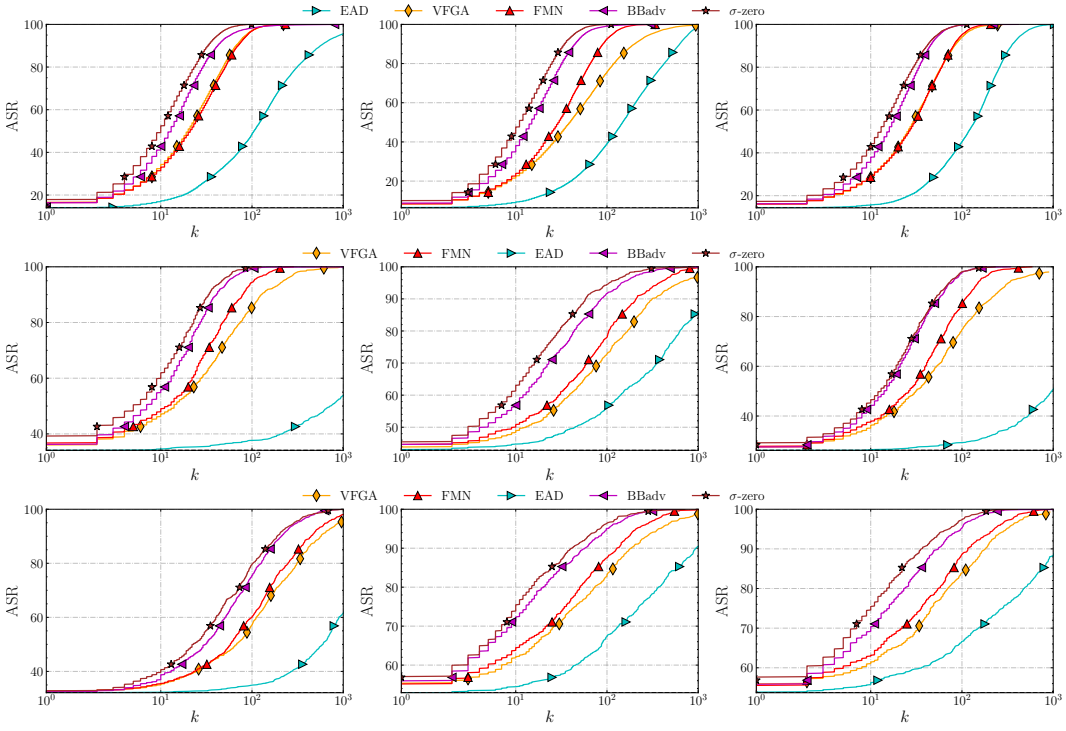

Figure 7: From the leftmost to the rightmost we report the robustness evaluation curves for C8, C9, C10 (top-row), I1, I2, I3 (middle-row) and I4, I5, I6 (bottom-row)

Table 6: Minimum-norm comparison results for CIFAR-10 and ImageNet with $N = 1000$ on remaining models. For each attack and model (M), we report ASR at $k = 10, 50, \infty$, median perturbation size $\tilde{\ell}_0$, mean runtime s (in seconds), mean number of queries q ($\div$ 1000), and maximum VRAM usage (in GB). When VFGA exceeds the VRAM limit, we re-run it using a smaller batch size, increasing its runtime t. We denote those cases with the symbol '$\star$'. Lastly we indicate with σ−zero * the case where we use $\sigma = 1$ and $\tau_0 = 0.1$.

Attack	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM
CIFAR-10																
SF	C5	11.19	11.19	56.56	3072	1.42	0.37	1.57	C8	23.67	24.85	62.41	3072	9.86	0.20	5.50
EAD		10.91	21.33	100.0	126	2.32	6.90	1.47		23.23	33.68	100.0	105	8.33	5.37	5.39
PDPGD		41.70	78.97	100.0	27	0.64	2.00	1.31		33.38	48.96	99.82	51	2.15	2.00	5.12
VFGA		77.25	93.41	99.99	11	0.17	0.32	11.96		56.76	81.79	99.89	20	4.30*	0.62	> 40
FMN		95.99	99.97	100.0	7	0.60	2.00	1.3		54.74	79.70	100.0	21	2.05	2.00	5.12
BB		97.43	99.79	100.0	7	5.81	2.76	1.47		59.82	78.76	83.58	16	12.49	3.14	5.39
BBadv		97.50	99.86	100.0	7	4.57	2.01	1.63		74.51	93.42	100.0	13	6.99	2.01	5.51
σ−zero		**99.20**	**100.0**	100.0	5	0.74	2.00	1.51		**81.23**	**97.33**	100.0	**10**	2.75	2.00	5.90
SF	C6	16.78	16.79	35.38	∞	19.74	0.62	10.00	C9	12.12	12.14	70.77	3072	3.28	0.22	2.25
EAD		20.75	35.90	100.0	74	10.76	5.55	9.92		14.51	23.62	100.0	148	2.23	5.80	2.15
PDPGD		23.84	40.89	100.0	69	3.96	2.00	8.86		25.31	38.41	100.0	69	0.76	2.00	2.0
VFGA		45.28	67.51	99.88	29	4.91*	1.02	> 40		38.42	56.72	99.81	39	3.15*	1.84	> 40
FMN		45.73	68.38	100.0	29	3.91	2.00	8.86		44.38	70.24	100.0	30	0.73	2.00	2.0
BB		15.26	17.14	17.94	∞	3.46	2.08	9.93		70.11	93.24	100.0	15	6.49	2.87	2.16
BBadv		64.47	88.92	100.0	16	8.85	2.01	10.03		69.45	92.91	100.0	15	6.02	2.01	2.22
σ−zero		**75.63**	**94.47**	100.0	**11**	4.41	2.00	10.43		**79.59**	**96.93**	100.0	**11**	0.89	2.00	2.65
SF	*C7*	29.51	40.86	93.82	3039	9.3	1.56	1.90	*C10*	25.88	26.54	51.80	3072	0.58	0.33	0.51
EAD		9.92	11.14	100.0	398	2.57	5.66	1.89		19.44	29.23	100.0	118	1.01	5.32	0.41
PDPGD		32.60	49.19	100.0	51	1.16	2.00	1.8		29.98	41.00	100.0	66	0.44	2.00	0.36
VFGA		61.19	90.04	99.88	19	0.28	0.52	16.53		48.63	74.15	99.54	25	0.17	0.77	3.07
FMN		52.14	85.60	100.0	23	1.09	2.00	1.8		47.89	73.71	100.0	26	0.41	2.00	0.36
BB		21.44	31.03	31.36	∞	3.01	2.37	1.89		68.37	91.83	100.0	15	10.90	2.93	0.41
BBadv		77.88	99.11	100.0	14	4.51	2.01	1.99		67.35	93.04	100.0	16	4.60	2.01	0.54
σ−zero		**81.38**	**99.15**	100.0	**12**	1.39	2.00	1.91		**73.96**	**94.21**	100.0	**13**	0.63	2.00	0.51
FMN	C11	39.57	74.75	100.0	32	0.11	2.00	0.59	C12	48.3	78.16	100.0	26	0.11	2.00	0.59
BBadv		14.07	18.57	100.0	183	2.52	2.01	0.65		18.33	19.75	100.0	290	2.57	2.01	0.65
σ−zero		12.38	15.91	100.0	144	0.24	2.00	1.03		18.52	21.31	100.0	187	0.33	2.00	1.03
σ−zero *		**44.78**	**85.05**	100.0	**27**	0.23	2.00	1.03		**54.62**	**90.12**	100.0	**22**	0.23	2.00	1.03
ImageNet																
Attack	**M**	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM
EAD	I5	56.6	60.2	100.0	0	21.38	5.50	1.41	I6	59.0	61.4	100.0	0	7.89	5.29	0.48
VFGA		69.0	76.2	98.8	0	6.11*	1.43	> 40		66.8	76.6	99.3	0	1.74*	1.21	> 40
FMN		71.0	79.5	100.0	0	1.97	2.00	2.30		70.9	78.7	100.0	0	0.72	2.00	0.67
BBadv		82.3	89.0	100.0	0	185.34	2.01	2.41		80.3	89.6	100	0	199.47	2.01	0.73
σ−zero		**85.1**	**91.4**	100.0	0	2.76	2.00	2.52		**86.2**	**92.8**	100.0	0	1.13	2.00	0.84
FMN	I7	39.2	48.5	100	54.5	5.84	2	17.44	I8	38.1	46.8	100	67	5.17	2	7.91
BBadv		49.7	62	100	25.5	128.2	2	17.86		46.5	58.6	100	29.5	113.87	2	8.30
σ−zero		**55.4**	**68.2**	100	**16**	8.62	2	19.03		**50.1**	**64.4**	100	**24**	6.6	2	9.47

Table 7: Minimum-norm comparison results for MNIST with $N = 100$. See the caption of Table 6 for further details.

Attack	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM
MNIST																
SF	*M1*	5.11	6.76	96.98	469	1.07	0.18	0.07	*M2*	0.98	1.21	91.68	463	2.87	0.86	0.07
EAD		3.73	46.65	100.0	52	0.06	1.14	0.07		3.51	35.57	100.0	61	0.06	0.99	0.07
PDPGD		0.98	0.98	100.0	359	0.01	0.20	0.07		0.52	0.52	95.02	254	0.01	0.20	0.07
VFGA		4.82	82.68	100.0	27	0.07	0.76	0.23		4.82	38.99	99,98	57	0.07	1.34	0.24
BB		**68.52**	98.00	100.0	**20**	0.13	1.19	0.08		62.98	83.00	87.87	18	0.13	1.69	0.08
FMN		33.03	83.09	88.92	30	0.01	0.20	0.07		10.05	14.03	14.81	∞	0.01	0.20	0.07
BBadv		62.29	90.88	100.0	21	0.09	0.21	0.08		41.19	58.80	100.0	34	0.07	0.21	0.08
σ−zero		61.12	**98.45**	100.0	22	0.01	0.20	0.08		**87.20**	**99.82**	100.0	**13**	0.01	0.20	0.08

Table 8: Minimum-norm comparison results for CIFAR-10 with $N = 100$. See the caption of Table 6 for further details.

Attack	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM
CIFAR-10																
SF		17.67	17.76	47.26	∞	3.17	0.35	1.62		16.75	16.79	35.36	∞	19.74	0.62	10.00
EAD		16.74	28.74	100.0	100.0	0.27	0.80	1.53		19.79	32.94	100.0	83	1.58	0.82	10.04
PDPGD		10.31	10.31	99.39	2421	0.05	0.20	1.43		11.26	11.26	99.75	2814	0.32	0.2	8.97
VFGA	*C1*	50.73	75.34	93.69	24	0.23	0.72	11.83	*C6*	45.33	67.05	87.75	29	3.75	0.86	> 40
FMN		46.90	69.36	80.68	27	0.05	0.20	1.43		42.67	61.49	72.34	33	0.31	0.2	8.98
BB		12.98	14.29	14.97	∞	0.44	1.95	1.59		14.99	16,88	17.91	∞	2.67	1.95	10.04
BBadv		60.52	86.63	100.0	18	0.41	0.21	1.59		59.61	84.59	100.0	18	0.76	0.21	10.04
σ-zero		**63.60**	**88.27**	100.0	**16**	0.08	0.20	1.84		**63.44**	**87.2**	100.0	**16**	0.44	0.2	10.29
SF		17.86	20.59	94.26	3071	2.44	0.26	1.91		21.07	38.76	82.71	3062	4.30	9.67	1.90
EAD		9.50	10.67	100.0	451	0.30	0.71	2.01		9.68	10.56	100.0	434	0.48	0.90	2.00
PDPGD		8.92	8.92	75.31	3052	0.09	0.20	1.91		9.17	9.17	99.90	2709	0.12	0.20	1.91
VFGA	*C2*	39.31	66.46	91.64	33	0.34	0.87	16.64	*C7*	60.94	90.04	99.16	19	0.29	0.52	16.64
FMN		37.13	62.41	71.3	36	0.08	0.20	1.92		50.70	79.48	87.20	24	0.08	0.20	1.91
BB		38.18	53.53	57.05	40	0.59	1.90	2.00		26.39	32.41	32.83	∞	0.50	1.93	2.00
BBadv		**63.56**	**92.74**	100.0	**19**	0.40	0.21	2.00		**74.91**	**98.37**	100.0	**15**	0.41	0.21	2.00
σ-zero		56.94	88.60	100.0	21	0.11	0.20	2.25		68.14	94.90	100.0	16	0.11	0.20	2.25
SF		20.89	24.36	58.29	3072	1.63	0.48	0.66		23.87	24.85	62.42	3072	9.86	0.2	5.50
EAD		13.03	13.18	100.0	835	0.11	0.65	0.64		21.71	29.59	100.0	128	0.67	0.66	5.51
PDPGD		12.95	12.98	99.47	2566	0.04	0.20	0.59		13.96	13.96	54.16	3072	0.21	0.2	5.23
VFGA	*C3*	28.63	49.73	82.94	51	0.13	1.13	4.44	*C8*	56.81	82.04	97.08	20	4.32	0.61	> 40
FMN		26.76	37.90	43.90	∞	0.03	0.20	0.59		53.42	76.59	87.1	22.0	0.21	0.2	5.24
BB		16.40	22.91	27.64	∞	1.04	2.25	0.65		60.74	78.14	84.46	17	1.36	1.67	5.55
BBadv		**33.68**	**66.79**	100.0	**37**	0.40	0.21	0.65		**70.31**	91.58	100.0	14	0.55	0.21	5.51
σ-zero		30.56	57.71	100.0	43	0.04	0.20	0.89		69.49	**91.87**	100.0	**14**	0.25	0.2	6.76
SF		31.85	42.97	84.45	70	1.54	0.47	0.66		12.03	12.14	70.77	3072	3.28	0.22	2.25
EAD		24.1	24.4	100.0	844	0.12	0.66	0.65		13.61	21.61	100.0	162	0.31	0.8	2.27
PDPGD		23.78	23.78	66.62	3072	0.04	0.2	0.59		6.31	6.31	96.2	2773	0.06	0.2	2.11
VFGA	*C4*	46.7	69.52	93.05	28	0.14	0.77	4.22	*C9*	38.22	56.56	75.79	39.5	1.45	1.06	> 40
FMN		42.69	58.78	65.83	35	0.03	0.2	0.59		40.27	59.69	68.88	35	0.06	0.2	2.19
BB		25.91	27.98	29.51	∞	0.54	2.09	0.65		66.02	90.74	100.0	16	0.65	1.07	2.27
BBadv		**52.25**	**80.64**	100.0	**23**	0.36	0.21	0.65		64.41	89.7	100.0	17	0.42	0.21	2.27
σ-zero		49.74	73.75	100.0	25	0.04	0.2	0.89		**65.96**	**90.95**	100.0	**16**	0.09	0.2	2.52
SF		11.19	11.19	56.56	3072	1.42	0.37	1.56		24.28	26.54	51.90	3072	0.58	0.33	0.52
EAD		10.42	19.09	100.0	146	0.26	0.77	1.58		18.82	26.17	100.0	144	0.11	0.79	0.52
PDPGD		5.23	5.23	100.0	3057	0.05	0.20	1.43		14.29	14.29	90.95	3057	0.03	0.2	0.47
VFGA	*C5*	77.22	93.44	98.99	11	0.17	0.38	12.08	*C10*	48.49	74.14	94.16	26	0.12	0.73	3.18
FMN		89.83	97.72	98.86	**8**	0.05	0.20	1.43		46.75	69.77	80.68	27	0.03	0.2	0.48
BB		84.42	97.55	100.0	10	0.62	0.95	1.59		**63.70**	89.39	100.0	**17**	0.43	1.13	0.53
BBadv		83.81	97.35	100.0	10	0.45	0.21	1.59		63.29	**90.08**	100.0	**17**	0.35	0.21	0.53
σ-zero		**91.54**	**99.84**	100.0	9	0.08	0.20	1.83		60.79	86.02	100.0	18	0.04	0.2	0.77

Table 9: Minimum-norm comparison results for ImageNet with $N = 100$. See the caption of Table 6 for further details.

Attack	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM	M	ASR_{24}	ASR_{50}	ASR_∞	$\tilde{\ell}_0$	s	q	VRAM
ImageNet																
EAD		34.7	35.9	100.0	484	1.02	0.67	0.46		32.4	33.0	100.0	808	5.15	0.7	1.68
VFGA		58.3	72.2	85.3	14	1.06	0.70	> 40		40.0	46.8	56.9	66.5	9.23	1.2	> 40
FMN	*I1*	55.4	64.5	68.1	14	0.08	0.20	0.66	*I4*	39,9	46.2	47.5	∞	0.44	0.2	2.97
BBadv		67.6	83.3	100.0	10	23.02	0.21	0.72		**46.4**	**58.0**	99.9	32	21.23	0.21	3.07
σ-zero		**69.2**	**86.9**	100.0	**10**	0.13	0.20	0.84		43.7	55.2	100.0	**32**	0.61	0.2	3.20
EAD		47.1	50.1	100.0	48	2.32	0.68	1.42		56.2	60.2	100.0	0	2.46	0.72	1.41
VFGA		54.3	63.2	96.7	13	2.88	0.72	> 40		68.9	76.0	83.0	0	2.33	0.59	> 40
FMN	*I2*	55.9	60.0	62.4	10	0.20	0.20	2.30	*I5*	67.8	72.0	74.3	0	0.20	0.20	2.30
BBadv		70.1	80.1	100.0	5	20.49	0.21	2.40		80.8	87.8	100.0	0	18.60	0.21	2.41
σ-zero		**71.0**	**82.7**	100.0	**4**	0.29	0.20	2.52		**81.8**	**89.3**	100.0	0	0.29	0.20	2.52
EAD		26.9	27.7	100.0	1108	0.58	0.61	1.41		57.4	60.0	100.0	0	1.03	0.72	0.48
VFGA		47.0	58.7	74.0	31	3.07	0.96	> 40		66.8	75.2	83.9	0	0.91	0.59	> 40
FMN	*I3*	44.4	50.6	53.2	47	0.16	0.2	2.30	*I6*	69.2	74.9	77.2	0	0.07	0.2	0.67
BBadv		**53.6**	**74.7**	100.0	**20**	23.86	0.21	2.41		80.1	89.1	100.0	0	19.68	0.21	0.73
σ-zero		52.9	74.4	100.0	21	0.23	0.2	2.52		**82.2**	**90.5**	100.0	0	0.12	0.2	0.84

Table 10: Fixed-budget comparison results with $N = 1000$ on CIFAR10 remaining models. Sparse-RS was executed with double the steps, $2N$, to ensure fair comparison as it lacks backward passes. For each attack, we report the corresponding ASR with different feature budget levels (24,50,100). We report the execution time $\mathbf{s}_{24}$ and query usage $\mathbf{q}_{24}$ for the smaller $k = 24$, as it requires, on average, more iterations due to the more challenging problem.

Attack	M	ASR_{24}	ASR_{50}	ASR_{100}	$\mathbf{q}_{24}$	$\mathbf{s}_{24}$	VRAM	M	ASR_{24}	ASR_{50}	ASR_{100}	$\mathbf{q}_{24}$	$\mathbf{s}_{24}$	VRAM
CIFAR-10														
PGD-ℓ_0		68.60	88.89	98.14	2.00	1.95	1.89		42.81	66.19	90.49	2.00	3.24	7.36
Sparse-RS		99.71	100.0	100.0	0.08	0.10	1.91		72.54	86.72	94.84	0.78	1.10	7.35
sPGD$_p$	C5	**99.82**	100.0	100.0	0.02	0.16	2.06	C8	68.47	90.47	99.56	0.70	1.48	7.62
sPGD$_u$		97.84	99.98	100.0	0.09	0.37	2.06		73.55	94.55	99.97	0.60	1.65	7.62
σ-zero		99.20	100.0	100.0	0.22	0.11	2.07		**81.23**	**97.33**	**99.97**	0.52	0.54	7.76
PGD-ℓ_0		32.80	50.53	77.06	2.00	4.88	12.79		31.45	52.79	80.27	2.00	2.01	2.91
Sparse-RS		**76.61**	89.88	96.22	0.67	1.98	12.74		68.77	82.06	89.81	0.85	0.56	2.89
sPGD$_p$	C6	63.66	87.07	98.67	0.80	2.83	13.77	C9	61.0	83.49	96.76	0.87	0.88	3.03
sPGD$_u$		64.28	88.25	99.09	0.77	2.77	13.77		63.48	87.59	98.49	0.81	0.85	3.03
σ-zero		75.63	**94.47**	**99.78**	0.66	1.75	13.82		**79.59**	**96.93**	**99.91**	0.57	2.04	2.91
PGD-ℓ_0		37.91	68.90	95.31	2.00	1.96	2.47		38.33	61.88	89.50	2.00	1.12	0.51
Sparse-RS		63.75	84.49	95.74	0.97	0.61	2.46		64.80	81.46	91.13	0.91	0.45	0.50
sPGD$_p$	C7	72.82	96.74	99.98	0.61	0.94	2.57	C10	59.94	84.87	98.82	0.87	0.44	0.55
sPGD$_u$		**81.64**	99.09	100.0	0.42	0.69	2.57		65.07	90.78	**99.83**	0.75	0.41	0.55
σ-zero		81.38	**99.15**	100.0	0.46	0.21	2.68		**73.96**	**94.21**	99.80	0.67	0.14	0.57
Sparse-RS		28.08	41.89	58.45	0.38	1.53	0.59		51.64	71.27	86.57	0.32	1.14	0.59
sPGD$_u$		15.87	21.43	32.67	0.26	1.71	0.65		22.78	26.56	34.09	0.31	1.58	0.64
sPGD$_p$	C11	13.61	17.07	30.11	0.25	1.74	0.65	C12	24.52	34.39	59.89	0.30	1.54	0.65
σ-zero		12.38	15.91	30.43	0.20	1.77	1.03		18.52	21.31	28.81	0.27	1.65	1.03
σ-zero*		**44.78**	**85.05**	**99.76**	0.15	1.33	1.03		**54.62**	**90.12**	**99.94**	0.13	1.15	1.03

Table 11: Fixed-budget comparison results for ImageNet with $N = 1000$ on remaining models. Sparse-RS was executed with double the steps, $2N$, to ensure fair comparison as it lacks backward passes. For each attack, we report the corresponding ASR with budget level $k = 150$. We report the execution time $\mathbf{s}_{100}$ and query usage $\mathbf{q}_{100}$ for the smaller $k = 100$, as it requires, on average, more iterations due to the more challenging problem.

Attack	M	ASR_{100}	ASR_{150}	$\mathbf{q}_{100}$	$\mathbf{s}_{100}$	VRAM	M	ASR_{150}	ASR_{150}	$\mathbf{q}_{100}$	$\mathbf{s}_{100}$	VRAM
ImageNet												
Sparse-RS		83.6	87.5	0.44	2.85	4.39		85.4	89.2	0.41	3.46	1.29
sPGD$_p$	I5	89.8	94.5	0.24	1.64	4.48	I6	90.4	95.2	0.22	0.98	1.33
sPGD$_u$		86.5	92.6	0.29	1.55	4.48		89.1	94.0	0.24	1.15	1.33
σ-zero		**95.9**	**98.2**	0.12	0.16	4.90		**98.1**	**98.8**	0.10	0.08	1.79
Sparse-RS		58.20	60.60	0.95	5.21	17.43		49.20	52.10	1.13	3.28	7.89
sPGD$_p$	I7	67.50	75.50	0.70	4.85	17.80	I8	65.10	75.20	0.75	3.56	8.24
sPGD$_u$		65.70	75.10	0.73	5.37	17.82		65.10	75.20	0.73	5.68	8.23
σ-zero		**82.10**	**87.00**	0.43	1.87	19.03		**78.00**	**86.20**	0.50	1.67	9.46

Table 12: Fixed-budget comparison results with $N = 5000$ on MNIST. See the caption of Table 10 for further details.

Attack	M	ASR_{24}	ASR_{50}	ASR_{100}	$\mathbf{q}_{24}$	$\mathbf{s}_{24}$	VRAM	M	ASR_{24}	ASR_{50}	ASR_{100}	$\mathbf{q}_{24}$	$\mathbf{s}_{24}$	VRAM
MNIST														
Sparse-RS		88.13	99.26	99.99	2.45	1.86	0.04		**99.88**	99.97	100.0	0.31	0.17	0.04
sPGD$_p$	M1	81.22	99.30	100.0	2.83	1.50	0.05	M2	83.88	99.88	99.97	2.33	0.9	0.05
sPGD$_u$		87.30	99.85	100.0	1.60	1.44	0.05		74.38	99.46	99.99	3.47	0.96	0.05
σ-zero		**88.63**	**100.0**	100.0	1.38	0.20	0.08		99.67	**100.0**	100.0	0.24	0.02	0.08

Table 13: Fixed-budget comparison results with $N = 5000$ on CIFAR10. See the caption of Table 10 for further details.

Attack	M	ASR_{24}	ASR_{50}	ASR_{100}	q_{24}	s_{24}	VRAM	M	ASR_{24}	ASR_{50}	ASR_{100}	q_{24}	s_{24}	VRAM
CIFAR-10														
Sparse-RS	C1	**82.94**	94.77	98.68	2.55	1.81	1.92	C5	**99.93**	99.98	99.99	0.15	0.14	1.91
$sPGD_p$		71.88	93.17	99.75	3.21	2.78	2.06		99.93	100.0	100.0	0.17	0.18	2.05
$sPGD_u$		69.73	92.86	99.79	3.34	2.98	2.06		99.73	100.0	100.0	0.26	1.31	2.05
σ-zero		80.91	**96.81**	**99.98**	3.23	1.35	2.09		99.81	100.0	100.0	0.61	0.18	2.05
Sparse-RS	C2	71.21	90.28	97.69	3.89	2.49	2.46	C7	77.06	94.33	99.25	3.38	2.15	2.46
$sPGD_p$		64.88	92.03	99.85	3.97	3.69	2.57		78.41	98.36	100.0	2.59	5.87	2.57
$sPGD_u$		68.61	94.99	99.96	3.49	3.46	2.57		**84.17**	99.41	100.0	1.85	6.26	2.57
σ-zero		**78.14**	**98.39**	**100.0**	3.16	1.02	2.70		83.99	**99.49**	100.0	1.82	0.69	2.70
Sparse-RS	C3	38.43	58.27	79.59	6.77	2.61	0.69	C9	79.69	89.98	95.3	2.81	1.95	2.89
$sPGD_p$		34.62	65.54	96.55	6.70	2.52	0.73		66.39	87.65	98.19	3.73	2.75	3.03
$sPGD_u$		37.29	72.03	98.49	6.48	2.96	0.73		68.03	90.7	99.23	3.56	2.64	3.04
σ-zero		**40.99**	**76.00**	**98.98**	6.32	1.42	0.77		**83.92**	**98.39**	**99.99**	2.19	0.99	3.09
Sparse-RS	C4	54.85	71.95	86.25	5.03	2.37	0.69	C10	76.62	91.5	97.89	3.20	1.46	0.50
$sPGD_p$		53.36	80.97	99.13	4.94	2.24	0.73		65.92	90.38	99.72	3.86	1.63	0.55
$sPGD_u$		**57.31**	**86.11**	**99.72**	4.46	2.36	0.73		68.59	92.93	99.91	3.71	1.62	0.55
σ-zero		57.11	84.54	99.34	4.47	1.04	0.77		**77.74**	**95.86**	**99.92**	2.75	0.57	0.59

Table 14: Fixed-budget comparison results with $N = 5000$ on ImageNet. See the caption of Table 11 for further details.

Attack	M	ASR_{100}	ASR_{150}	q_{100}	s_{100}	VRAM	M	ASR_{100}	ASR_{150}	q_{100}	s_{100}	VRAM
ImageNet												
Sparse-RS	I1	94.2	95.1	1.39	7.73	1.29	I4	48.8	51.7	5.68	13.51	5.73
$sPGD_p$		97.3	99.6	0.45	1.95	1.41		66.2	78.4	3.68	21.65	5.84
$sPGD_u$		93.6	98.5	0.71	2.48	1.40		64.1	78.5	3.78	20.71	5.84
σ-zero		**100.0**	**100.0**	0.72	0.31	1.83		**77.3**	**87.8**	2.42	4.17	6.33
Sparse-RS	I2	85.1	86.8	2.06	11.0	4.39	I5	89.6	91.3	1.54	3.35	4.39
$sPGD_p$		85.9	92.8	1.63	8.33	4.49		92.0	95.9	0.94	3.35	4.39
$sPGD_u$		81.3	90.5	2.00	6.88	4.49		88.0	93.1	1.28	5.89	4.48
σ-zero		**94.6**	**97.3**	0.63	0.61	4.94		**96.9**	**98.4**	0.41	0.34	4.94
Sparse-RS	I3	74.8	76.6	3.54	6.38	4.39	I6	87.5	92.7	1.36	2.05	1.29
$sPGD_p$		87.6	95.4	1.61	6.29	4.49		96.5	97.0	0.83	1.39	1.33
$sPGD_u$		81.4	93.7	2.04	6.54	4.49		90.4	94.7	2.04	2.35	1.33
σ-zero		**98.2**	**99.7**	1.68	1.44	4.94		**97.2**	**99.1**	0.35	0.12	1.83

Table 15: Fixed-budget comparison results with $N = 10000$ on CIFAR-10 and ImageNet. See the caption of Tables 10-11 for further details.

Attack	M	ASR_{24}	ASR_{50}	ASR_{100}	q_{24}	s_{24}	VRAM	M	ASR_{100}	ASR_{150}	q_{100}	s_{100}	VRAM
Sparse-RS	C3	41.12	63	83.99	3.44	12.82	0.60	I2	87.2	88.3	9.66	3.67	0.14
$sPGD_u$		37.95	72.6	98.59	1.99	12.56	0.65		86.9	91.6	7.84	1.95	0.15
$sPGD_p$		35.89	67.92	97.42	2.12	13.2	0.65		81.8	88.9	8.4	3.85	0.15
σ-zero		**41.67**	**76.38**	**99.01**	0.33	3.07	2.41		**95.1**	**97.2**	1.31	0.23	0.21

C VISUAL COMPARISON

In Figures 8-10, we show adversarial examples generated with competing ℓ_0-attacks, and our σ−zero. First, we can see that ℓ_0 adversarial perturbations are clearly visually distinguishable Carlini & Wagner (2017a); Brendel et al. (2019a); Pintor et al. (2021). Their goal, indeed, is not to be indistinguishable to the human eye – a common misconception related to adversarial examples (Biggio & Roli, 2018; Gilmer et al., 2018) – but rather to show whether and to what extent models can be fooled by just changing a few input values.

A second observation derived from Figures 8-10 is that the various attacks presented in the state of the art can identify distinct regions of vulnerability. For example, note how FMN and VFGA find similar perturbations, as they mostly target overlapping regions of interest. Conversely, EAD finds sparse perturbations scattered throughout the image but with a lower magnitude. This divergence is attributed to EAD's reliance on an ℓ_1 regularizer, which promotes sparsity, thus diminishing perturbation magnitude without necessarily reducing the number of perturbed features. Conversely, our attack does not focus on specific areas or patterns within the images but identifies diverse critical features, whose manipulation is sufficient to mislead the target models. Given the diverse solutions offered by the attacks, we argue that their combined usage may still improve adversarial robustness evaluation to sparse attacks.

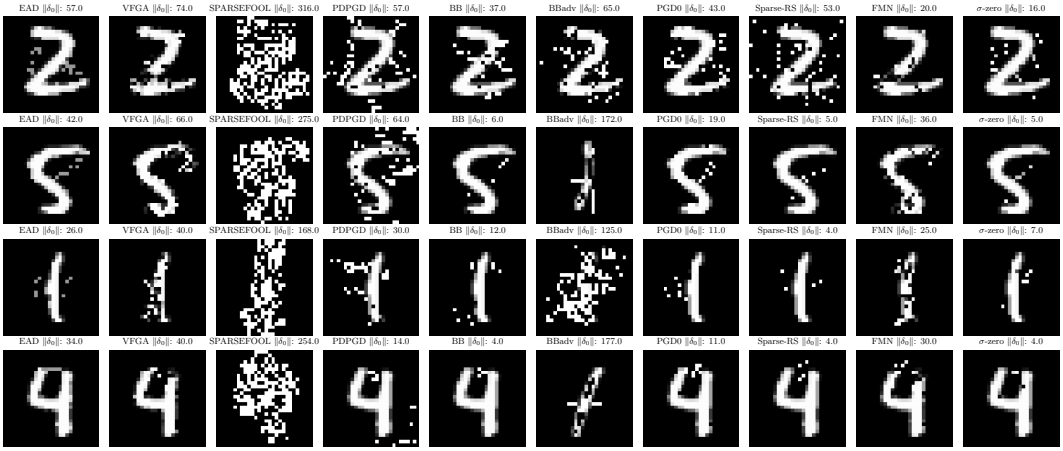

Figure 8: Randomly chosen adversarial examples from MNIST M2.

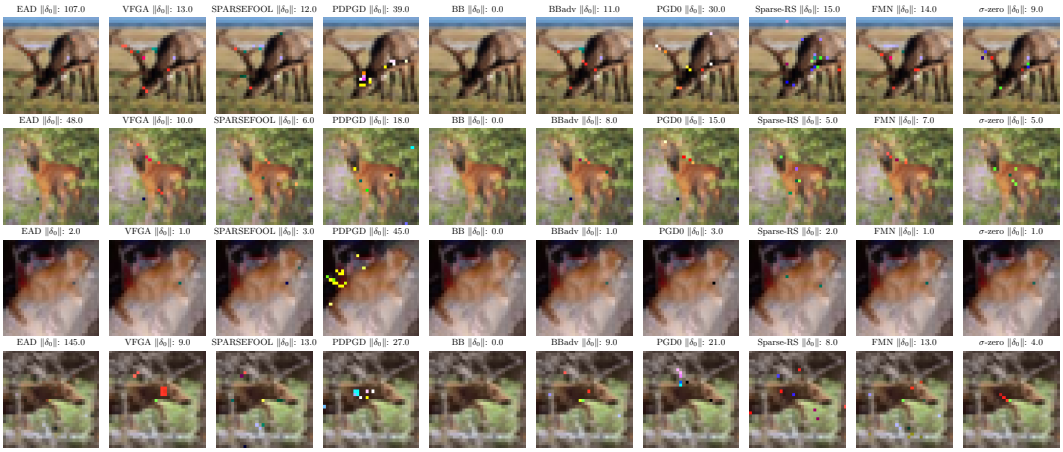

Figure 9: Randomly chosen adversarial examples from CIFAR-10 C1.

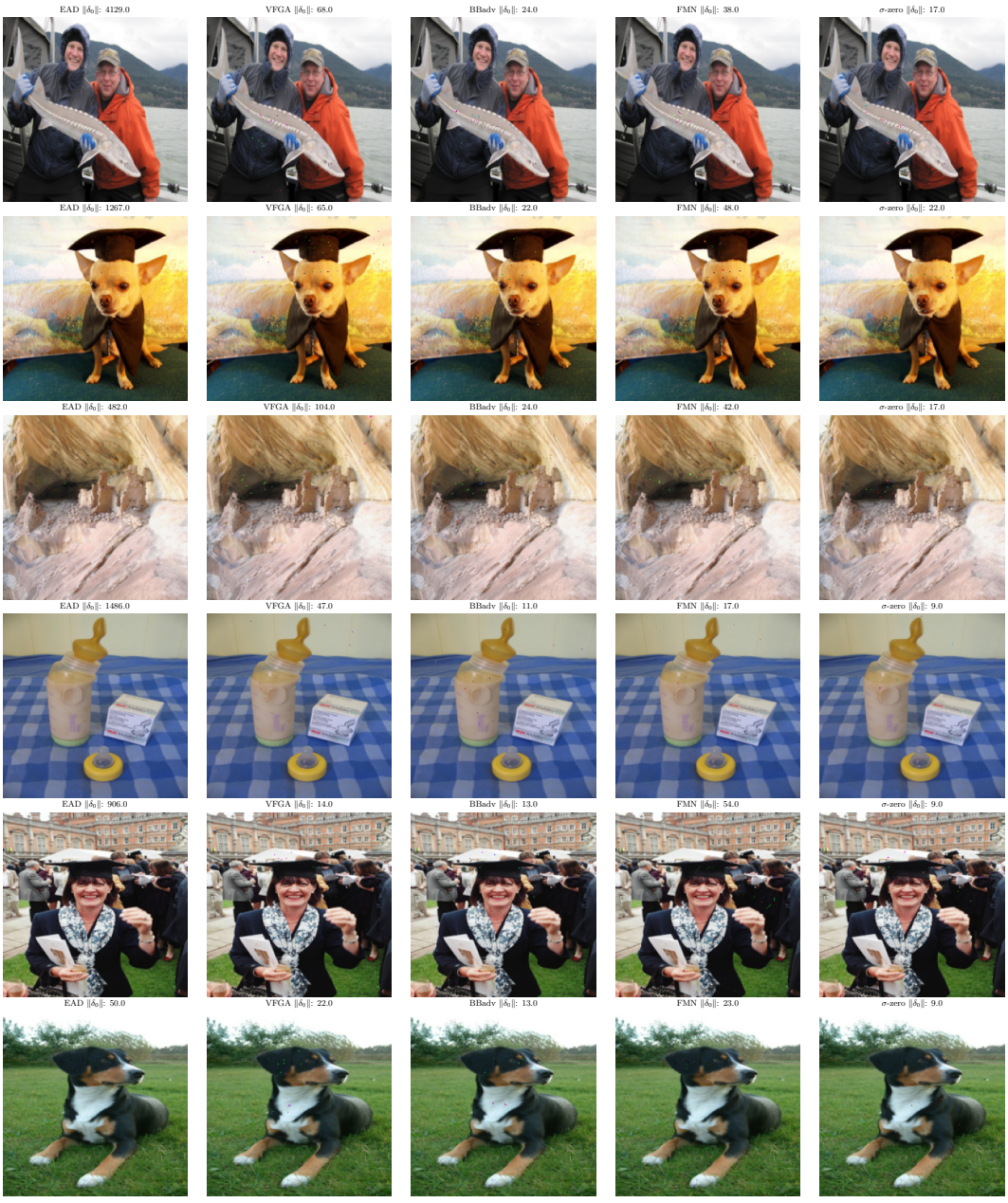

Figure 10: Randomly chosen adversarial examples from ImageNet I1.