SPARSE AND TRANSFERABLE UNIVERSAL SINGULAR VECTORS ATTACKS

Kseniia Kuvshinova*
Sber AI Lab,
Skolkovo Insitute of Science and Technology
Moscow, Russia
kseniia.kuvshinova@skoltech.ru

Olga Tsymboi*
Sber AI Lab,
Moscow Institute of Physics and Technology
Moscow, Russia
tsimboy.oa@phystech.edu

Ivan Oseledets
Artificial Intelligence Research Institute (AIRI),
Skolkovo Insitute of Science and Technology
Moscow, Russia
oseledets@airi.net

ABSTRACT

Mounting concerns about neural networks' safety and robustness call for a deeper understanding of models' vulnerability and research in adversarial attacks. Motivated by this, we propose a novel universal attack that is highly efficient in terms of transferability. In contrast to the existing (p, q)-singular vectors approach, we focus on finding sparse singular vectors of Jacobian matrices of the hidden layers by employing the truncated power iteration method. We discovered that using resulting vectors as adversarial perturbations can effectively attack the original model and models with entirely different architectures, highlighting the importance of sparsity constraint for attack transferability. Moreover, we achieve results comparable to dense baselines while damaging less than 1% of pixels and utilizing only 256 samples for perturbation fitting. Our algorithm also admits higher attack magnitude without affecting the human ability to solve the task, and damaging 5% of pixels attains more than a 50% fooling rate on average across models. Finally, our findings demonstrate the vulnerability of state-of-the-art models to universal sparse attacks and highlight the importance of developing robust machine learning systems.

1 INTRODUCTION

In recent years, deep learning approaches have become increasingly popular in many areas and applications, starting from computer vision Dosovitskiy et al. (2021b) and natural language processing Touvron et al. (2023); Chung et al. (2022) to robotics Roy et al. (2021) and speech recognition Baevski et al. (2020). The success and availability of pre-trained neural networks have also made it easier for researchers and developers to use these models for their applications. Despite tremendous advances, many studies discover that deep learning models are vulnerable to small imperceptible perturbations of input data called adversarial attacks that mislead models and cause incorrect predictions Szegedy et al. (2014); Goodfellow et al. (2014); Moosavi-Dezfooli et al. (2017). Adversarial attacks as a phenomenon first appeared in the field of computer vision and have raised concerns about the reliability of safety-critical machine learning applications.

Initially, adversarial examples were constructed for each input Szegedy et al. (2014), making it challenging to scale attacking methods to large datasets. In Moosavi-Dezfooli et al. (2017), the authors show the existence of universal adversarial perturbations (UAPs) that result in the model's misclassification for most of the inputs. Such attacks are crucial for adversarial machine learning research, as they are easier to deploy in real-world applications and raise questions about the safety and robustness of state-of-the-art architectures. However, the proposed optimization algorithm requires vast

*These authors contributed equally to this work.

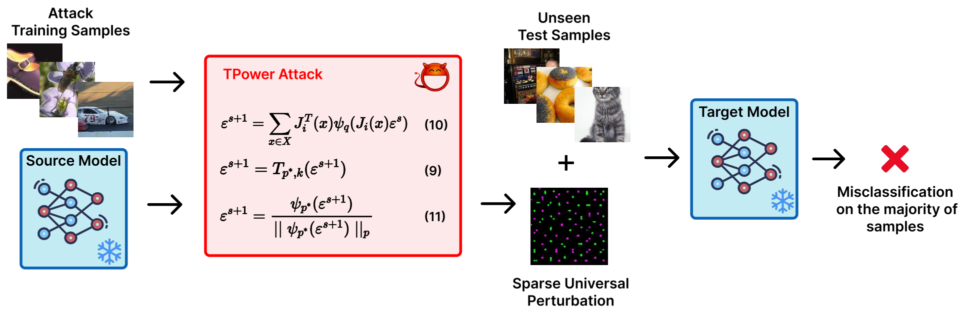


Figure 1: Demonstration of the proposed TPower Attack method. Starting with an attack training set and a source pretrained victim model, the universal perturbation is iteratively refined using the (p, q) Truncated Power Iteration algorithm. Here, $J_i(x)$ denotes i's hidden layer Jacobian matrix, and X is an attack training set. This process generates a sparse attack vector, applied to previously unseen test images, a set significantly larger than the initial attack training set. The addition of this perturbation to the test images leads to incorrect predictions for the majority of the samples.

data, making it complicated to fool real-world systems. In contrast, Khrulkov & Oseledets (2018) proposes a sample-efficient method to construct perturbation using leading (p, q)-singular vectors Boyd (1974) of the Jacobian of the hidden layers. However, computing the Jacobian is infeasible due to memory limitations. The authors address this issue using the generalized power method for the attack computation.

The abovementioned approaches formalize imperceptibility using straightforward vector norm constraints in the underlying optimization problem. However, in general, an attack can alter the image significantly, leaving human-evaluated label unchanged Song et al. (2018); Brown et al. (2018). One can step beyond the small-norm imperceptibility definition and perform a patch attack in the form of a physical sticker on an object in real-time conditions Hu & Shi (2022); Li et al. (2019); Pautov et al. (2019); Kaziakhmedov et al. (2019).

This paper focuses on l_0-bounded attacks, motivated by the following two key considerations. First, dense attacks are significantly constrained in the amplitude of adversarial perturbations; beyond a certain threshold, these perturbations render the image unrecognizable. In contrast, sparse attacks do not face this limitation, allowing for more effective adversarial perturbations. Furthermore, we find that altering a small number of pixels through sparse attacks has no impact on the human label. This crucially preserves the human ability to interpret and solve the given task accurately, a significant practical implication of our research.

There are quite a lot of methods to compute sparse adversaries Croce & Hein (2019); Modas et al. (2019); Yuan et al. (2021); Dong et al. (2020), most of them are based on adding l_0 constraints. However, the transferability of such attacks is low Papernot et al. (2016a; 2017); Liu et al. (2016). In other words, these methods may perform poorly in grey-box settings (when a surrogate model is attacked instead of the initial model). However, we should highlight that only a few works aim to incorporate sparsity constraints into universal attack setup Croce et al. (2022). It aligns differently from our approach for several reasons. It only proposes a single patch targeted attack in the universal setup, while we consider a general sparsity pattern. More than that, an auxiliary generative model is usually used to construct such transferable sparse attacks He et al. (2022); Hayes & Danezis (2018); Mopuri et al. (2018).

The main focus of this paper is to investigate computer vision models' robustness to sparse universal adversarial examples. Summing up, our main contribution is as follows:

- We propose a new approach to construct sparse UAPs on hidden layers subject to predefined sparsity patterns (see Figure 1, Algorithm 1).
- We assess our method on the ImageNet benchmark dataset Deng et al. (2009a) and evaluate it on various deep learning models. We compare it against existing universal approaches regarding the fooling rate and the transferability between models.

- Our experimental study shows that the proposed method produces highly transferable perturbations. Our approach is essential because it is efficient concerning sample size – a moderate sample size of 256 images to construct an attack on is enough for a reasonable attack fooling rate, while fooling inceptive layers is more beneficial.

2 FRAMEWORK

In this paper, we focus on the problem of untargeted universal perturbations for image classification. The problem of universal adversarial attacks can be framed as finding a perturbation ε that, when added to most input images x, causes the classifier to predict a different class than it would for the original images. Let $f : \mathcal{X} \to \mathcal{Y}$ be a classification model defined for the dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where x_i is an input and y_i is a corresponding label, then, according to Moosavi-Dezfooli et al. (2016), UAP is a perturbation ε such that

$$\mathbb{P}_{x\sim\mu}[f(x+\varepsilon) \neq f(x)] \geq 1-\delta, \quad \|\varepsilon\| \leq \xi,$$

where μ denotes a distribution of input data, $1-\delta$ is the minimal Fooling Rate (FR) and ξ is the attack magnitude. It should be highlighted that this perturbation must not change human prediction, meaning that the true class of the attacked image remains unchanged, but a small norm constraint could be omitted Song et al. (2018); Brown et al. (2018).

Adversarial perturbations are often obtained via optimization problem solution. The most straightforward approach is to maximize expected cross-entropy loss $\mathcal{L}(x+\varepsilon, y)$. It was shown Khrulkov & Oseledets (2018) that instead of attacking the model output, one can attack hidden layers of a model. The error produced in this way will propagate to the last network layer, resulting in a change in model prediction. Given the i-th layer l, the optimization problem can be obtained via Taylor expansion:

$$\begin{gathered} l(x+\varepsilon) - l(x) \approx J_i(x)\varepsilon \\ \mathbb{E}_{x\sim\mu}\|l(x+\varepsilon) - l(x)\|_q^q \to \max_{\|\varepsilon\|_p \leq \xi}, \end{gathered} \tag{1}$$

where $J_i(x)$ is the i-th layer Jacobian operator and $J_i(x)\varepsilon$ is Jacobian action on ε. Finally, equation 1 is equivalent to the following problem.

$$\mathbb{E}_{x\sim\mu}\|J_i(x)\varepsilon\|_q^q \to \max_{\|\varepsilon\|_p=1}. \tag{2}$$

The solution to equation 2 can be referred to as Jacobian (p, q)-singular vector and defined up to the signed scale factor, here p, q are the hyperparameters to be tuned, and expectation is relaxed via averaging over a batch.

Our approach. In this paper, we incorporate the universal layerwise approach from above with sparsity or, formally speaking, additional non-convex l_0 constraint. Relaxing the expectation in equation 2 with an average over attack training set X, we have:

$$\begin{gathered} \sum_{x\in X}\|J_i(x)\varepsilon\|_q^q \to \max, \\ s.t.\ \|\varepsilon\|_p = 1, \quad \|\varepsilon\|_0 \leq k. \end{gathered} \tag{3}$$

When $p = q = 2$, the problem above leads to the famous problem of finding sparse eigenvalues. However, for an arbitrary pair (p, q), it is a non-convex and NP-hard problem. One way to obtain an approximate solution is to use the truncated power iteration method (TPower, Yuan & Zhang (2013)). Despite efficiency and simplicity, the major TPower drawback is the theoretical guarantee with a narrow convergency region. One way to reduce the effect of this issue is to reduce the number of nonzero entries iteratively. In this paper, we introduce an algorithm that adopts TPower for the case of arbitrary p, q and effectively solves the problem of universal perturbation finding.

Let us rewrite equation 3 using the dual norm definition, then we obtain

$$\begin{gathered} y^\top J_i\varepsilon \to \max, \\ s.t.\ \varepsilon \in \mathcal{B}_p(1), \quad y \in \mathcal{B}_{q^*}(1), \quad \|\varepsilon\|_0 \leq k, \end{gathered} \tag{4}$$

Algorithm 1: TPower Attack

Require: the number of TPower iterarions n_steps, initial ratio of damages pixels $init\_truncation \in (0, 1)$, the set of attack training images $X = \{x_j\}_{j=1}^{N}$, image size n, number of channels c, $j \in \overline{1, N}$, norm hyperparameters q and p, target cardinality top_k, $patch\_size$, number of iteration between truncation update $reduction\_steps$.

$k = init\_truncation \cdot (n//patch\_size)^2 \cdot c$.
$k = \max(k, top\_k)$.
$\varepsilon =$ random tensor of batch size.
for s from 1 to n_steps **do**
  $\varepsilon^{s+1} = T_{p^*,k}\left[\sum_{x \in X} J_i^\top(x)\psi_q(J_i(x)\varepsilon^s)\right], \quad equation\ 10$
  $\varepsilon^{s+1} = \frac{\psi_{p^*}(\varepsilon^{s+1})}{\|\psi_{p^*}(\varepsilon^{s+1})\|_p}, \quad equation\ 11$
  if $s \mod reduction\_steps = 0$ **then**
    $k_{\text{reduction}} = \text{pow}(\frac{k}{top\_k}, \frac{reduction\_steps}{n\_steps})$
    $k = \max(k/k_{\text{reduction}},\ top\_k)$
  end if
end for

Ensure: ε

where q^* and q are Hölder conjugate, i.e. $(q^*)^{-1} + q^{-1} = 1$, $\mathcal{B}_p(1) = \{x \in \mathbb{R}^n |\ \|x\|_p = 1\}$ and $J_i = [J_i(x_1))^\top, \ldots, J_i(x_N))^\top]^\top,\ x_j \in X$. The solution could be found via Alternating Maximization (AM) Method.

For any fixed perturbation vector ε, the inner problem is linear and admits a closed-form solution :

$$y = \frac{\psi_q(J_i\varepsilon)}{\|\psi_q(J_i\varepsilon)\|_{q^*}}, \quad \psi_q(y) = \text{sign}(y)|y|^{q-1}. \tag{5}$$

Changing the order of maximizations in equation 4, the subproblem for ε remains the same except l_0 constraint, which could be replaced by additional binary variable t maximization. Thus, denoting $d = J_i^\top y$ and $(\cdot)$ as a Hadamard product, we have:

$$\begin{gathered}(t \cdot d)^\top \varepsilon \to \max, \\ s.t.\ \varepsilon \in \mathcal{B}_p(1), \quad \sum_j t_j \le k, \quad t_j \in \{0, 1\},\ \forall j.\end{gathered} \tag{6}$$

For a fixed t, the problem is reduced to the previous case, and hence

$$\varepsilon = \frac{\psi_{p^*}(t \cdot d)}{\|\psi_{p^*}(t \cdot d)\|_p}, \tag{7}$$

The problem equation 6 is equivalent to the following:

$$\arg\max_t \|d \cdot t\|_{p^*}, \quad s.t. \sum_j t_j \le k, \quad t_j \in \{0, 1\},\ \forall j \tag{8}$$

and thus maximization by t is simply a selection of the greatest components of vector d in $\|\cdot\|_{p^*}$ value computed over the predefined sparsity pattern. Truncation operator could do this:

$$T_{p^*,k}(d) = \begin{cases} d_i, & i \in \text{ArgTop}_k\{\|d_i\|_{p^*}\}, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

here ArgTop_k denotes operator which returns indices of k largest elements of an input vector.

Finally, putting it together, we derive the following alternating maximization update at the step s for attack training:

$$\varepsilon^{s+1} = T_{p^*,k}\left[\sum_{x \in X} J_i^\top(x)\psi_q(J_i(x)\varepsilon^s)\right], \tag{10}$$

$$\varepsilon^{s+1} = \frac{\psi_{p^*}(\varepsilon^{s+1})}{\|\psi_{p^*}(\varepsilon^{s+1})\|_p}. \tag{11}$$

The overall algorithm is presented in Algorithm 1, where we gradually decrease the cardinality through the iterations to enhance convergency Yuan & Zhang (2013).

3 Experiments

This section presents the experiments to analyze the effectiveness of sparse UAPs described above. The experiments were implemented using PyTorch, and the code will be made publicly available on Github upon publication.

3.1 Experiments setup

Datasets. In this work, following Khrulkov & Oseledets (2018), to evaluate the performance of the proposed sparse attack, we used the validation subset of the ImageNet benchmark dataset (ILSVRC2012, Deng et al. (2009b), available for non-commercial research and educational purposes), which contains 50,000 images belonging to 1,000 object categories. We randomly sample 256 images from the ImageNet validation subset for attack training. We used 5000 images for validation but later recalculated them with a smaller validation size. For those on which FR changes, it changes slightly with the change of the optimal layer.

Models. During the empirical analyses, we restrict ourselves to the following models to be examined: DenseNet161 (Huang et al. (2017)), EffecientNetB0, EffecientNetB3 (Tan & Le (2019)), InceptionV3 (Szegedy et al. (2015)), ResNet101, ResNet152 (He et al. (2016)), Wide ResNet101 (Zagoruyko & Komodakis (2016)), DEIT base (Touvron et al. (2021)), ViT base (Dosovitskiy et al. (2021a)). For each model, we use corresponding ImageNet pre-trained checkpoints from Pytorch (Ansel et al. (2024), Apache License 2.0) for Convolutional Neural Networks (CNNs) and Transformers for Transformer models (Wolf et al. (2020), BSD-3).

Hyperparameters. In our experiments, to estimate attack performance, we vary the following hyperparameters: model, layer to be attacked, $patch\_size \in \{1, 4, 8\}$ and objective norm parameter $q \in \{1, 2, 3, 5, 7, 10\}$ while keeping p fixed, in particular, $p = \infty$, which is motivated by the previous study (Moosavi-Dezfooli et al. (2017); Khrulkov & Oseledets (2018); Naseer et al. (2020)). The number of non-zero patches top_k is also fixed in accordance with the image and patch sizes and selected so that the fraction of damaged pixels is equal to 5%, which further allows us to increase the attack magnitude up to 1 (see Table 4). We gradually went through all semantic blocks to study the performance dependence on the layer to be attacked (see Appendix A.1 for more details).

Evaluation metrics. For evaluation, we report Fooling Rate (FR) equation 12 for the best perturbation obtained on the 256 training samples. It also means that we find ourselves in an unsupervised setting and do not need access to the ground truth labels.

$$FR = \frac{1}{N} \sum_{x \in \mathcal{D}} [f(x) \neq f(x + \varepsilon)] \tag{12}$$

3.2 Main results

We train our attack on nine different models and compare it to the stochastic gradient descent (SGD) attack Shafahi et al. (2020) and the dense analog of our approach proposed by Khrulkov & Oseledets (2018), here and below, we refer the last approach as singular vectors (SV) attack. We also consider transferability setup and investigate the FR dependence on q. We also compare with SGD layer maximization attack (LMax) Co et al. (2021), essentially an unlinearized version of the SV algorithm. Attack samples are presented in Figure 7.

Following previous research, which relies on the small norm assumption, the magnitude was decreased to $10/255$ for dense baselines. Poor results in the SGD can be explained by the fact that a relatively large train set size is required to obtain an efficient attack, e.i. the number of samples should exceed the number of classes Shafahi et al. (2020), while for our proposed approach, 256 images are enough.

The grid search results are presented in Table 4, where we report optimal hyperparameters for each model with respect to validation FR. For this setting, the comparison with the baselines is provided in Table 1, where for SV and SGD attacks, we additionally perform a similar grid search on hyperparameters 6. Our TPower attack approach outperforms baselines for almost all models except EfficientNets and demonstrates diverse attack patterns.

From Table 1, one can conclude that EfficientNet is the most robust architecture. EfficientNet exhibits unique robustness properties that have garnered attention in our study and other recent studies Peng et al. (2023); Lukasik et al. (2023). Some architectural choices, like compound scaling, limit the gradient flow during backpropagation. This fact makes it more challenging for attackers to generate efficient adversarial perturbations. For more details and possible explanations of such phenomenon, see Appendix A.2.

Model	**TPower**	TPower Avg	SV	SGD	LMax
DenseNet161	**89.11**	58.97	34.25	15.9	23
EffecientNetB0	**37.09**	31.34	34.44	17.31	19.12
EffecientNetB3	**15.22**	14.62	13.49	8.4	11.21
InceptionV3	**85.04**	75.61	27.88	13.6	24.64
ResNet101	**94.57**	82.09	50.05	17.38	46.95
ResNet152	**94.84**	83.15	35.93	15.43	22.49
WideResNet101	**94.36**	84.05	36.35	15.71	28.42
DEIT base	**43.37**	26.16	31.1	29.93	23.55
ViT base	**52.5**	29.97	26.01	18.11	28.09

Table 1: Test FR for TPower, SV, SGD, and LMax adversarial perturbations. For the TPower and SV attack, we report test FR for optimal hyperparameters after the grid search. The best result is in **bold**, and the second best one is underlined. For Tpower attack, we also report the average FR of the 50% first layers while the rest of the parameters were chosen to be well-performed on average across models (See Table 4).

Additionally, for the ViT model, Figure 9 demonstrates a highly interpretable pattern. Formally speaking, during ViT preprocessing, the image is cut into fixed-size patches, which are further flattened and combined with positional encoding Wu et al. (2020). Indeed, perturbation forms a quasi-regular grid repeating the locations of the patch junctions. Moreover, attacking lower, more sensitive to preprocessing by construction layers causes the highest model vulnerability (Figure 5).

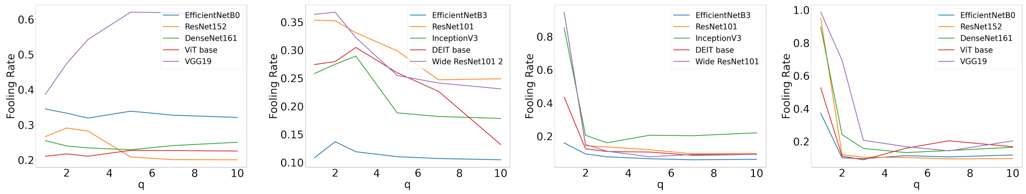
Dependence on patch size. Our empirical study shows that, in general, lower patch size values are more beneficial in terms of FR (see Table 4). One can see that pixel-wise attack mode is more efficient regarding the fooling rate. This might be related to the fact that uniform square greed is not an optimal sparsity pattern. However, for most models, the decrease in performance is not dramatic, except for the transformers one. For those models where the optimal patch size option is 4, FR does not decrease significantly compared to the single pixel patch attack, namely, only approximately 5% for ResNet101 (from 94.57% to 89.85%, 2). Finally, the small size of patches with a fixed proportion of damaged pixels allows patches to scatter more across the whole picture, resulting in more uniform perturbation of model filters' receptive fields.

Dependence on q. In Khrulkov & Oseledets (2018), on the example of VGG19, authors demonstrate that model vulnerability increases with q and saturates when $q = 5$. The last is explained by the fact that $q = 5$ is enough to smooth the approximation of $q = \infty$. On the contrary, for the majority of models, we obtained almost opposite results: higher q values in the SV attack of Khrulkov & Oseledets (2018) are less efficient in terms of FR for both methods, TPower and the SV attack (see Figure 2). However, for the sparse attack setting, the dependence of all models becomes unambiguous, even for the VGG19 model. In addition, with $q = 1$, patches are arranged more evenly across the perturbation image than for larger values of q, depicted in Figure 3.

Model / Patch size	1	4	8
DenseNet161	**89.11**	78.52	64.82
EffecientNetB0	**37.09**	29.95	22.87
EffecientNetB3	**15.22**	13.11	13.28
InceptionV3	**85.04**	22.36	77.66
ResNet101	89.85	**94.57**	83.48
ResNet152	89.53	**94.84**	76.78
WideResNet101	93.97	**94.36**	84.1
DEIT base	**43.37**	22.37	15.59
ViT base	**52.5**	15.44	14.54

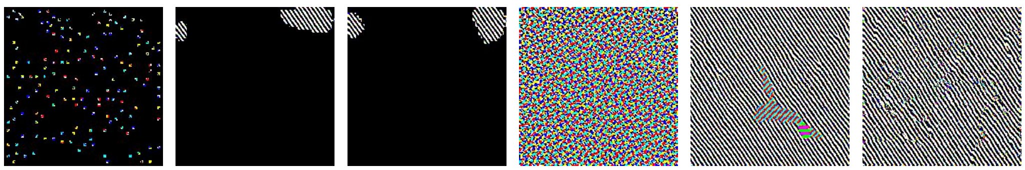
Table 2: Tpower attack FR dependence on patch size on test.

Dependence on layer number. To investigate whether attack performance is layer-dependent, we introduce layer ratio: the layer number normalized to the model depth. Figure 5 shows that lower layers are more effective as victims, empirically confirming the hypothesis of perturbation propagation through the network and repeating the SV attack property. Additionally, Table 1 shows that



(a) SV Attack. (b) SV Attack. (c) TPower Attack. (d) TPower Attack.

Figure 2: Dependence of FR on q for TPower Attack. For sparse attacks, optimal parameters from gridsearch were frozen except for q (see Table 4) and reused for the dense one.

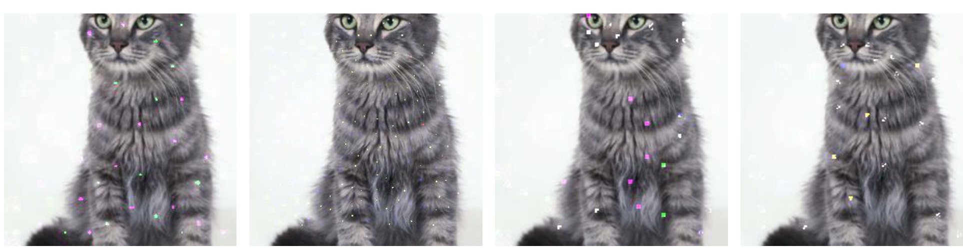


(a) $q = 1$ TPower Attack (b) $q = 5$ TPower Attack (c) $q = 10$ TPower Attack (d) $q = 1$ SV Attack (e) $q = 5$ SV Attack (f) $q = 10$ SV Attack

Figure 3: Universal adversarial perturbations constructed for the VGG19 model.

even if we do not choose a layer but take them from the first half, then on average, our attack is still better than the baseline for almost all models (see TPower Avg column).

Cardinality experiments. We analyzed one of the critical hyperparameters — the number of adversarial patches denoted top_k. This hyperparameter plays a pivotal role in determining the ratio of damaged pixels of the attack, as well as the overall performance of the attack. In the initial experiments, we selected the top_k parameter following the 5% rate of affected pixels, producing promising results on our dataset. We conducted an additional experiment to determine how many sparse adversarial patches are enough to obtain the same fooling rate as for the dense attack. We manually chose this parameter to make FR metrics the same as for the SV attack. Figure 4 illustrates the resulting images for four models. As anticipated, the choice of top_k significantly affects the attack performance. However, one can conclude that less than 1% of pixels is enough to obtain an equally efficient attack with SV one.



(a) DenseNet161, 0.5% (b) InceptionV3, 0.56% (c) ResNet101, 0.37% (d) ResNet152, 0.27%

Figure 4: Attacked images and corresponding percentage of damaged pixels obtained using TPower approach. The top_k parameter was manually selected such that sparse UAPs reach approximately the same fooling rate as SV attack (see Table 1). as SV attacks.

Transferability experiments. Following the above setup, in the transferability task, for each model we only consider the optimal perturbations in terms of FR obtained during the gridsearch. The rest of the evaluations are done on the test subset. In contrast to the direct task setting, when the adversarial perturbation is applied to the same model on which it was obtained, the attack should be adjusted to the input size of the victim model. In particular, we preprocess adversaries either centre-cropping or zero-padding them to fulfill the victim model input size restriction. Even though this affects the attack transferability performance, from Table 7, Table 3 and Table 8, one can observe higher transferability of Tpower attack across different models in the majority of cases. Winning rates were calculated as a ratio of cases where TPower outperforms SV Attack. For instance, ResNets are the most vulnerable to the TPower Attack, achieving a winning rate of 1 even in a transferability setup. Notably, EfficientNets are the most effective architectures in a gray-box setting. For more details, see Table 8, Table 9, and Table 7. TPower attack achieves an average improvement of 32% in transferability, with an average winning rate of 85%. Such high transferability is explained by superior performance in the initial setting (see Table 2)

From/To	DN	ENB0	ENB3	IncV3	RN101	RN152	WRN101	DEIT	VIT
DenseNet161 (DN)	-	15.29	8.88	55.31	83.9	68.93	92.98	16.66	17.29
EfficientNetB0 (ENB0)	78.51	-	9.41	41.58	88.44	77.42	78.67	23.03	27.78
EfficientNetB3 (ENB3)	75.82	28.2	-	41.02	76.74	66.03	71.45	24.61	29.81
InceptionV3 (IncV3)	93.94	35.9	14.65	-	98.01	95.92	96.62	31.18	58.31
ResNet101 (RN101)	90.09	18.52	9.65	60.67	-	88.08	96.48	17.42	23.12
ResNet152 (RN152)	84.39	22.05	10.21	60.41	96.98	-	94.94	21.11	29.61
Wide ResNet101 (WRN101)	86.55	16.68	8.64	56.43	90.7	79.34	-	17.45	21.12
DEIT	75.36	40.64	10.64	51.32	75.97	71.12	80.2	-	31.2
ViT	77.27	22.12	10.19	63.72	98.28	97.19	94.85	32.79	-

Table 3: Transferability results for proposed TPower attack in terms of the fooling rate. Rows refer to the model adversarial perturbation was computed on, while columns —to the victim one on which the attack was tested.

Table 3 demonstrates that the most transferable attacks in terms of the fooling rate are the ones trained on transformers and EffecientNets. Moreover, EffecientNets are the most robust among the examined models. Furthermore, the transferability of attacks among different architectures, as observed in the rest of the surrogate models, is a noteworthy finding. The attacks are transferred almost equally well, indicating a potential vulnerability across these architectures. This behavior could be explained by EfficientNet's significant differences from all other models, as this architecture was not developed manually but using the AutoML MNAS framework. Nevertheless, attacks trained on EfficientNets have a sufficient transferability fooling rate of at least 24%, preserving the attacked picture's good quality (see Figure 9) and outperforming the results for dense perturbations. It is worth mentioning that such attacks perform better in the transferability setting than in the direct one.

To sum up, the transferability of the proposed TPower approach makes it promising for a grey-box setting, where an attack is trained on one model and applied to an unknown one.

This, in turn, leads us to the universalization of adversarial attacks and their application for task-agnostic and dataset-agnostic setups.

4 Related Work

In recent years, there has been significant progress in adversarial attacks, particularly in deep neural networks (DNNs). These attacks, first introduced by Szegedy et al. in Szegedy et al. (2014), have profoundly impacted various domains, highlighting the importance of understanding model robustness through vulnerability to adversarial examples. The initial approach by Szegedy et al. utilized the L-BFGS algorithm, albeit with a high computational cost for large sample sizes. Subsequent efforts have aimed to enhance efficiency in both computational complexity and attack performance under specific constraints Goodfellow et al. (2015); Moosavi-Dezfooli et al. (2016); Carlini & Wagner (2017); Madry et al. (2017). New gradient-based attacks have since emerged, such as those employing flexible perturbation sets Wong et al. (2019); Wong & Kolter (2020), attacks relying solely on classifier output scores Guo et al. (2019b); Cheng et al. (2018); Wang et al. (2020); Guo et al. (2019a); Andriushchenko et al. (2020), and decision-based attacks with access only to predicted labels Chen et al. (2020).

While many approaches focus on perturbing all pixels, others advocate for sparsity constraints, such as using l_0 or l_1 measures Papernot et al. (2016b); Su et al. (2019); Chen et al. (2018); Modas et al. (2019). Techniques like group sparsity introduced by Xu et al. Xu et al. (2019) and generative architectures explored in works like Dong et al. (2020); He et al. (2022) have further refined imperceptibility constraints. Recent efforts, such as the smooth relaxation proposed in Zhu et al. (2021), continue to explore sparsity enhancements.

Despite advancements, many existing methods rely on sample-dependent perturbations, rendering them computationally impractical for large datasets. Universal Adversarial Perturbations (UAPs) offer a promising alternative, aiming to deceive models regardless of input specifics. Early works like Moosavi-Dezfooli et al. (2017) introduced UAPs, but scalability remains a challenge. Contrastingly, approaches like that proposed by Khrulkov et al. Khrulkov & Oseledets (2018) achieve universality with significantly fewer training samples. This concept has also been extended beyond computer vision, as seen in recent adaptations for NLP tasks Tsymboi et al. (2023).

Additionally, attacks leveraging generative models Mopuri et al. (2018); Hayes & Danezis (2018); Poursaeed et al. (2018); Chen et al. (2023) have gained traction due to their ability to capture entire perturbation distributions, offering a broader scope compared to non-generative methods. Inspired by the progress of multimodal architectures, GAN-based universal downstream-agnostic attacks Zhou et al. (2023b;a) and attacks on pretrained models Ban & Dong (2022) were proposed.

An emerging topic is the interpretability of adversarial attacks. Recent trends, such as representing universal attacks as semantic features Zhang et al. (2020) and bridging universal and non-universal settings Li et al. (2022), indicate ongoing exploration in this field.

5 LIMITATIONS

One of the restrictions of the proposed approach is the fixed predefined attack cardinality, and due to the lack of convergence, heuristic reduction to this value should be made. One way to overcome this issue is to replace the truncation operator with adaptive threshold shrinkage obtained via the Alternating Direction Method of Multipliers (ADMM, Boyd et al. (2011)), which is planned to be done in future work.

Another weakness is that for sparse attacks to be efficient, we need to use higher magnitudes while keeping the percentage of damaged pixels low. As a result, adversarial perturbation could be considered an outlier with simple defense via weights clipping at each neural network layer or median filtration. However, clipping ranges for this case must be well-estimated, as well as window size for median filtration, as shown in our experiments (see Appendix A.2). One needs to tune these parameters, e.g., estimate them based on statistics of layers outputs, which is infeasible in the grey-box setting when the perturbation is transferred between models.

Investigating the reasons behind EfficientNets' remarkable robustness is a promising direction for future research. While we present some discussion in Appendix A.2, a comprehensive exploration of this topic is beyond the scope of our paper.

Finally, it would be interesting to investigate the sparse attack transferability in more realistic settings when both model and dataset are unknown to study task-independent adversarial perturbations.

6 CONCLUSION

This paper presents a new approach for sparse universal adversarial attack generation. Following Khrulkov & Oseledets (2018), we assume that the perturbation of an intermediate layer propagates further through the network. The primary outcome of the paper is that by using only an additional truncation operator, we can construct the perturbation that will alter at most 5% of input image pixels without a decrease in fooling rate compared to the dense algorithm version (SV Attack) but with a significant increase. Moreover, our attack is still efficient regarding the sample size used for perturbation training. In particular, utilizing 256 samples is enough to achieve at least a 50% fooling rate for most of the models, with a maximum of 94%. We comprehensively study 10 architectures, revealing their vulnerability to sparse universal attacks. We also show that found attack vectors are highly transferable, revealing an extremely high vulnerability of ResNets. TPower attack achieves an average improvement of 32% in transferability, with an average winning rate of 85%, comparing to SV attack. Furthermore, our attack can be well generalized across different networks without a decrease in the fooling rate.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pp. 484–501. Springer, 2020.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 929–947, 2024.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

Yuanhao Ban and Yinpeng Dong. Pre-trained adversarial perturbations. *Advances in Neural Information Processing Systems*, 35:1196–1209, 2022.

David W Boyd. The power method for lp norms. *Linear Algebra and its Applications*, 9:95–101, 1974.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49.

Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pp. 1277–1294. IEEE, 2020.

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 AAAI conference on artificial intelligence*, volume 32, 2018.

Zhaoyu Chen, Bo Li, Shuang Wu, Kaixun Jiang, Shouhong Ding, and Wenqiang Zhang. Content-based unrestricted adversarial attack. *arXiv preprint arXiv:2305.10665*, 2023.

Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

Kenneth T Co, Luis Muñoz-González, Leslie Kanthan, Ben Glocker, and Emil C Lupu. Universal adversarial robustness of texture and shape-biased models. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 799–803. IEEE, 2021.

Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4724–4732, 2019.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6437–6445, Jun. 2022. doi: 10.1609/aaai.v36i6.20595. URL `https://ojs.aaai.org/index.php/AAAI/article/view/20595`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009a. doi: 10.1109/CVPR.2009.5206848.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009b.

Chaitanya Devaguptapu, Devansh Agarwal, Gaurav Mittal, Pulkit Gopalani, and Vineeth N Balasubramanian. On adversarial robustness: A neural architecture search perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 152–161, 2021.

Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. *Advances in Neural Information Processing Systems*, 33:11226–11236, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021a.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021b.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014. URL `https://arxiv.org/abs/1412.6572`.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *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/1412.6572`.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pp. 2484–2493. PMLR, 2019a.

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

Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 43–49. IEEE, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ziwen He, Wei Wang, Jing Dong, and Tieniu Tan. Transferable sparse adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14963–14972, 2022.

Chengyin Hu and Weiwen Shi. Adversarial color film: Effective physical-world attack to dnns. *arXiv preprint arXiv:2209.02430*, 2022.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems*, 34:5545–5559, 2021.

Edgar Kaziakhmedov, Klim Kireev, Grigorii Melnikov, Mikhail Pautov, and Aleksandr Petiushko. Real-world attack on mtcnn face detection system. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 0422–0427. IEEE, 2019.

Valentin Khrulkov and Ivan Oseledets. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International Conference on Machine Learning*, pp. 3896–3904. PMLR, 2019.

Maosen Li, Yanhua Yang, Kun Wei, Xu Yang, and Heng Huang. Learning universal adversarial perturbation by adversarial example. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1350–1358, 2022.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

Jovita Lukasik, Paul Gavrikov, Janis Keuper, and Margret Keuper. Improving native cnn robustness with filter frequency regularization. *Transactions on Machine Learning Research*, 2023.

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

Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016. doi: 10.1109/CVPR.2016.282.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 742–751, 2018.

Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016a.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016b.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Mikhail Pautov, Grigorii Melnikov, Edgar Kaziakhmedov, Klim Kireev, and Aleksandr Petiushko. On adversarial patches: real-world attack on arcface-100 face recognition system. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 0391–0396. IEEE, 2019.

Anjie Peng, Kang Deng, Hui Zeng, Kaijun Wu, and Wenxin Yu. Detecting adversarial examples via classification difference of a robust surrogate model. In *International Conference on Neural Information Processing*, pp. 558–570. Springer, 2023.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4422–4431, 2018.

Nicholas Roy, Ingmar Posner, Tim Barfoot, Philippe Beaudoin, Yoshua Bengio, Jeannette Bohg, Oliver Brock, Isabelle Depatie, Dieter Fox, Dan Koditschek, Tomas Lozano-Perez, Vikash Mansinghka, Christopher Pal, Blake Richards, Dorsa Sadigh, Stefan Schaal, Gaurav Sukhatme, Denis Therien, Marc Toussaint, and Michiel Van de Panne. From machine learning to robotics: Challenges and opportunities for embodied intelligence, 2021.

Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5636–5643, 2020.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/8cea559c47e4fbdb73b23e0223d04e79-Paper.pdf`.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL `http://arxiv.org/abs/1312.6199`.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

Olga Tsymboi, Danil Malaev, Andrei Petrovskii, and Ivan Oseledets. Layerwise universal adversarial attack on nlp models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 129–143, 2023.

Lu Wang, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Yuan Jiang. Spanning attack: Reinforce black-box attacks with unlabeled data. *Machine Learning*, 109:2349–2368, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.

Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, pp. 6808–6817. PMLR, 2019.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Do wider neural networks really help adversarial robustness? *Advances in Neural Information Processing Systems*, 34:7054–7067, 2021.

Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BkgzniCqY7`.

Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(4), 2013.

Zheng Yuan, Jie Zhang, Yunpei Jia, Chuanqi Tan, Tao Xue, and Shiguang Shan. Meta gradient adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7748–7757, 2021.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith (eds.), *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016. URL `http://www.bmva.org/bmvc/2016/papers/paper087/index.html`.

Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14521–14530, 2020.

Zeliang Zhang, Peihan Liu, Xiaosen Wang, and Chenliang Xu. Improving adversarial transferability with scheduled step size and dual example. *arXiv preprint arXiv:2301.12968*, 2023.

Ziqi Zhou, Shengshan Hu, Minghui Li, Hangtao Zhang, Yechao Zhang, and Hai Jin. Advclip: Downstream-agnostic adversarial examples in multimodal contrastive learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 6311–6320, 2023a.

Ziqi Zhou, Shengshan Hu, Ruizhi Zhao, Qian Wang, Leo Yu Zhang, Junhui Hou, and Hai Jin. Downstream-agnostic adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4345–4355, 2023b.

Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. Sparse and imperceptible adversarial attack via a homotopy algorithm. In *International Conference on Machine Learning*, pp. 12868–12877. PMLR, 2021.

Zhenyu Zhu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization). *Advances in neural information processing systems*, 35:36094–36107, 2022.

A APPENDIX

A.1 IMPLEMENTAION DETAILS

We performed the computation on four GPU's NVIDIA A100 of 80GB. The full grid search for TPower Attack took 765 GPU hours for nine models (6 values for q, 10 values for α, and 3 patch sizes). The total size of the hyperparameter set was $6 \times 10 \times 3 \times 9 = 1620$, which means that 65 GPU seconds are needed for pure attack training and approximately 2 minutes for one hyperparameter set testing, which is relatively fast.

Image preprocessing. The preprocessing stage is a crucial step in computer vision tasks that involves cleaning, standardizing, and enhancing the input data to improve model performance. In the context of these experiments, the preprocessing pipeline for the ImageNet ILSVRC2012 dataset will be discussed.

At first, we divided the ImageNet dataset between a training subset with 256 images, a validation subset of 5000 images and the rest for the test subset. The second step in the preprocessing pipeline involves resizing the input images to a fixed resolution. This is necessary to ensure that all images have the same dimensions and to reduce the computational overhead of training the models. The images will be resized to a resolution of crop sizes corresponding to the used models. After that, our actions for the subsets are different. We compute the attack tensor on the normalized train subset. We apply an attack on validation and test subsets.

Overall, the preprocessing pipeline for the ImageNet ILSVRC2012 dataset for our experiments consists of the train-validation-test split, resizing/cropping, attack applying, clipping to [0, 1] (for validation and test subset) and normalization.

Layer selection. We gradually went through all semantic blocks to study the performance dependence on the layer to be attacked:

- DenseNet161: dense layers and transition blocks,
- EffecientNetB0 and EffecientNetB3: bottleneck MBConv blocks,
- InceptionV3: max poolings and mixed blocks,
- ResNet101, ResNet152 and WideResNet101:
  residual blocks,
- Vit and DEIT: encoder layers.

A.2 EXPERIMENTS

Along with fooling rate, we focus our consideration on Attack Success Rate (ASR), namely the portion of misclassified samples after the attack performance filtered subject to the initial model's correct predictions:

$$ASR = \frac{\sum\limits_{x \in \mathcal{D}} [f(x) \neq f(x+\varepsilon)][f(x) = y]}{\sum\limits_{x,y \in \mathcal{D}} [f(x) = y]} \tag{13}$$

Grid search results for TPower attack. Optimal hyperparameters with corresponding fooling and attack success rates are presented in Table 4.

Grid search results for SV attack. Optimal hyperparameters with corresponding fooling and attack success rates are presented in Table 6. Figures 6, 7 and 8 present examples of dense adversarial perturbations. For optimal layers regarding the fooling rate, the dependence on q is ambiguous, while on average, the hypothesis that the greater q is, the better attack performance isKhrulkov & Oseledets (2018) not approved.

Grid search results for SGD attack. Optimal hyperparameters with corresponding fooling and attack success rates are presented in Table 5.

Dependence on patch size. In general, from Table 2, one can see that pixel-wise attack mode is more efficient regarding the fooling rate. This might be related to the fact that uniform square greed

Model	Top k	Patch Size	q	Attacked Layer	Test ASR	Test FR
DenseNet161	2509	1	1	features.denseblock2.denselayer6	87.61	89.11
EffecientNetB0	2509	1	1	features.2.1.block	29.66	37.09
EffecientNetB3	4500	1	1	features.1.0.block	9.66	15.22
InceptionV3	4471	1	1	maxpool2	82.83	85.04
ResNet101	157	4	1	layer2.3	93.8	94.57
ResNet152	157	4	1	layer2.3	94.24	94.84
WideResNet101	157	4	1	layer3.1	93.71	94.36
DEIT base	2509	1	1	vit.encoder.layer.0	36.12	43.37
ViT base	2509	1	1	vit.encoder.layer.0	46.76	52.5

Table 4: Metrics and hyperparameters for the best-performed sparse UAPs for each model.

Model	β	Step decay	Test FR
DenseNet161	5	1	15.9
EffecientNetB0	9	0.3	17.31
EffecientNetB3	10	0.2	8.4
InceptionV3	12	1	13.6
ResNet101	15	1	17.38
ResNet152	15	1	15.43
WideResNet101	7	1	15.71
DEIT base	6	0.6	29.93
ViT base	10	0.5	18.11

Table 5: Metrics and hyperparameters for the best-performed SGD attacks for each model. The attack magnitude was fixed at $\alpha = \frac{10}{255}$, β is clamping value.

Model	q	Attacked Layer	Test ASR	Test FR
DenseNet161	3	features.denseblock2.denselayer10	26.49	34.25
EffecientNetB0	5	features.2.1.block	26.58	34.44
EffecientNetB3	2	features.1.0.block	8.3	13.49
InceptionV3	2	Mixed_5b	19.64	27.88
ResNet101	5	layer1.0	43.54	50.05
ResNet152	5	layer1.0	28.58	35.93
WideResNet101	1	layer3.1	29.24	36.35
DEIT base	3	vit.encoder.layer.1	23.23	31.1
ViT base	5	vit.encoder.layer.0	18.75	26.01

Table 6: Metrics and hyperparameters for the best-performed SV attacks for each model. The attack magnitude was fixed at $\alpha = \frac{10}{255}$.

From/To	DN	ENB0	ENB3	IncV3	RN101	RN152	WRN101	DEIT	VIT
DenseNet161 (DN)	-	23.89	7.68	90.09	17.31	30.76	24.59	26.76	26.29
EfficientNetB0 (ENB0)	27.72	-	9.74	24.45	28.72	25.34	27.28	19.7	17.22
EfficientNetB3 (ENB3)	12.24	12.46	-	9.82	14.34	14.46	12.98	10.9	8.56
InceptionV3 (IncV3)	30.78	28.92	11.64	-	27.28	24.88	25.63	26.8	18.95
ResNet101 (RN101)	21.08	21.02	8.93	11.45	-	26.53	24.58	19.61	10.22
ResNet152 (RN152)	28.04	35.17	9.83	88.11	23.06	-	29.52	18.53	9.9
Wide ResNet101 (WRN101)	24.36	23.3	7.77	15.57	32.95	28.84	-	17.17	20.63
DEIT	17.82	17.24	7.31	9.65	25.57	18.88	20.33	-	9.8
ViT	23.87	23.13	8.5	14.28	28.29	21.06	20.96	13.64	-

Table 7: Transferability results for SV attack in terms of the fooling rate. Rows refer to the model adversarial perturbation was computed on, while columns —to the victim one on which the attack was tested.

From/To	DN	ENB0	ENB3	IncV3	RN101	RN152	WRN101	DEIT	VIT
DenseNet161 (DN)	-	-8.6	1.2	-34.78	66.59	38.17	68.39	-10.1	-9.0
EfficientNetB0 (ENB0)	50.79	-	-0.33	17.13	59.72	52.08	51.39	3.33	10.56
EfficientNetB3 (ENB3)	63.58	15.74	-	31.2	62.4	51.57	58.47	13.71	21.25
InceptionV3 (IncV3)	63.16	6.98	3.01	-	70.73	71.04	70.99	4.38	39.36
ResNet101 (RN101)	69.01	-2.5	0.72	49.22	-	61.55	71.9	-2.19	12.9
ResNet152 (RN152)	56.35	-13.12	0.38	-27.7	73.92	-	65.42	2.58	19.71
Wide ResNet101 (WRN101)	62.19	-6.62	0.87	40.86	57.75	50.5	-	0.28	0.49
DEIT	57.54	23.4	3.33	41.67	50.4	52.24	59.87	-	21.4
ViT	53.4	-1.01	1.69	49.44	69.99	76.13	73.89	19.15	-

Table 8: Difference of FR between TPower and SV attacks.

Metrics	DN	ENB0	ENB3	IncV3	RN101	RN152	WRN101	DEIT	VIT
AD on sources for target	13.98	30.58	39.74	41.21	32.58	22.19	25.79	38.73	42.835
WR averaged on sources for target	50.0	87.5	100.	100.	75.0	75.0	87.5	100.	87.5
AD on targets for source	59.50	1.78	1.36	20.88	63.94	56.66	65.04	3.89	14.58
WR averaged on targets for source	100.	37.5	87.5	75.0	100.	100.	100.	75.0	87.5

Table 9: Average differences (AD) and winning rates (WR) between TPower and SV fooling rates. WRs were calculated as a ratio of cases where TPower outperforms SV Attack.

is not an optimal sparsity pattern. However, for most models, the decrease in performance is not dramatic, except for the transformers one.

SGD with layer maximization. We also conducted additional experiments and decided to compare with SGD layer maximization attack Co et al. (2021), essentially an unlinearized version of our algorithm. Attack samples are presented in Figure 7. As we can observe, layer maximization significantly boosts classic stochastic gradient descent, but the attack still does not reach the performance of our attack or SV attack.

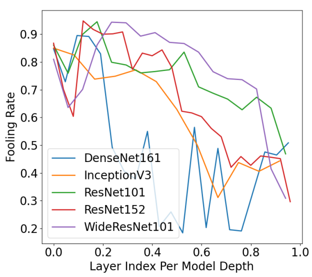
Median filtration. As mentioned above, the constructed perturbations consist of full-magnitude damaged patches scattered uniformly on the image. Due to the small patch size, one can propose median filtration of the vanilla method to mitigate such attack influence. Consequently, we have conducted experiments on the median filtration of attacked images with different window sizes. From Table 10, we observe a decrease in FR, e.g., for EfficientNetB3 and 3×3 filter, we get a $1/3$ decrease for FR from the initial one; for some models like DenseNet161, the FR decreases to only 79%. However, as a hyperparameter, the filter size should be selected for each model and balance between efficient filtration and over-blurring.

To conclude, the median filter can make the attack harder to fool the victim model but does not protect from it entirely. More reliable way to protect models is to use attack detectors or/and robust normalizations inside the models; this requires additional training for each attack type which is impractical.

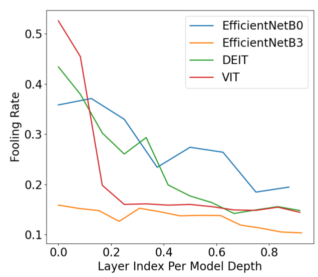
Model	3x3	5x5	7x7	11x11	15x15
DN	95.32	97.03	94.97	79.66	88.25
ENB0	17.31	29.27	40.99	66.95	82.34
ENB3	9.13	16.54	24.07	41.70	59.33

Table 10: Fooling Rate after the median filtration results for three models: EfficientNetB0 (ENB0), EfficientNetB3 (ENB3) and DenseNet161 (DN). We see that median filtration helps to eliminate attacks, but the optimal window size is not the same for all models and should be tuned. Moreover, exceeding the optimal threshold results in over-blurring and a decrease in the performance of the model, not due to the attack but because of the bad quality of the images themselves.

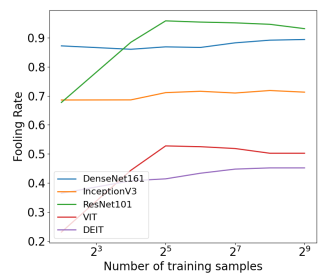
EfficientNet robustness. EfficientNet exhibits unique robustness properties that have garnered attention in recent studies. In one paper Peng et al. (2023), using EfficientNet as a surrogate in a gray-box setting demonstrated its robustness against BIM and CW attacks, using ResNet-50 as a victim. Similarly, in another study Zhang et al. (2023), EfficientNet emerged as the most robust architecture, even when considering Filter Frequency Regularization. However, in our experiments, we observed that for dense attacks, the Fooling Rate, such as in the SV attack, did not significantly differ from our attack's Fooling Rate. Additionally, recent research Devaguptapu et al. (2021) suggests that hand-crafted models may exhibit greater robustness on complex datasets like ImageNet, whereas NAS architectures like EfficientNets may fight better against weaker attacks like FGSM. Our findings present some ambiguity, as sparse attacks with such ablations have yet to be extensively studied. This warrants further investigation as an intriguing avenue for future research, although it falls outside the scope of our current paper. Nevertheless, we propose some intriguing hypotheses. One such assumption revolves around EfficientNet's compound scaling, a unique characteristic where depth, width, and resolution are scaled proportionally. Some studies suggest that unquestioningly increasing only width may degrade robustness Wu et al. (2021), while others demonstrate how depth may impact robustness differently based on initialization methods Zhu et al. (2022). Moreover, theoretical research Huang et al. (2021) suggests that increasing model depth exponentially raises the upper bound of Lipschitz constant, potentially compromising robustness. EfficientNet's unique feature lies in its balanced increase of width, depth, and resolution, maintaining proportional scaling. This approach may indirectly align with theoretical bounds on model capacity, but further investigation is needed to elucidate this relationship.



(a)

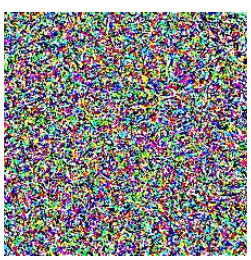


(b)

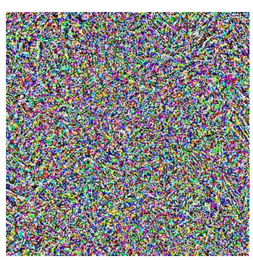


(c)

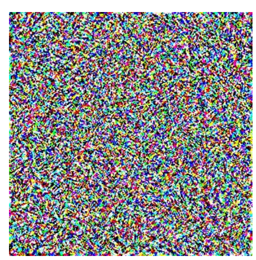
Figure 5: 5(a) and 5(b): FR dependence on layer ratio for examined models. 5(c): The example of fooling rate saturation depending on training set size for optimal hyperparameters; here, one can observe that 256 is the worst case amount among most vulnerable models.



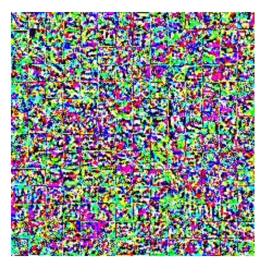
(a) DenseNet161



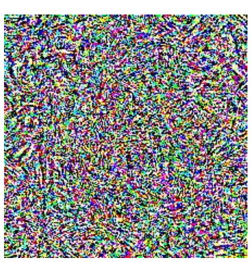
(b) EfficientNetB3



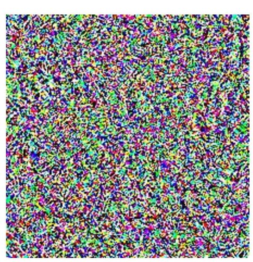
(c) InceptionV3



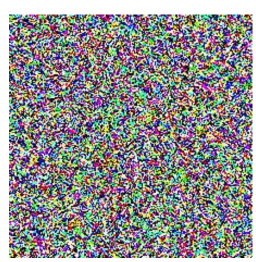
(d) ViT Base



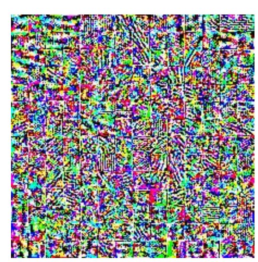
(e) EfficientNetB0



(f) ResNet101

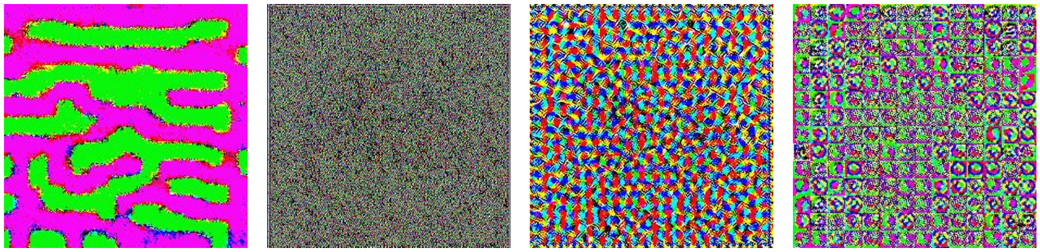


(g) ResNet152

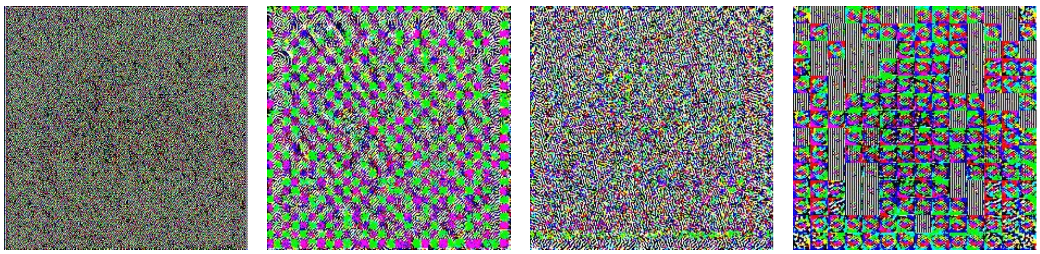


(h) DEIT Base

Figure 6: UAPs obtained using SGD attack algorithm Shafahi et al. (2020).

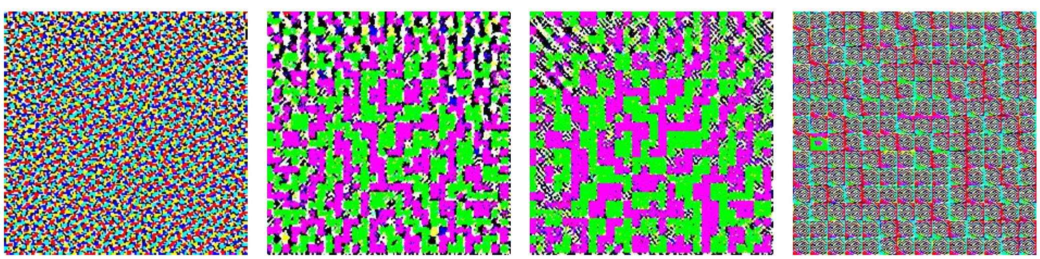


(a) DenseNet161 (b) EfficientNetB3 (c) InceptionV3 (d) ViT Base

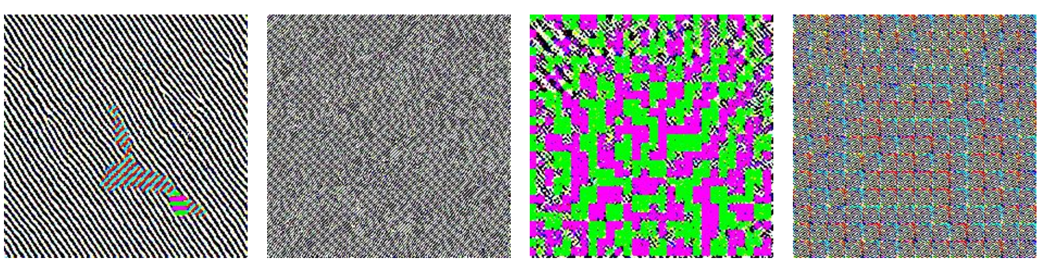


(e) EfficientNetB0 (f) ResNet101 (g) ResNet152 (h) DEIT Base

Figure 7: UAPs obtained using SGD with layer maximization attack algorithm Co et al. (2021). Selected layers are the same as optimal in TPower attack.

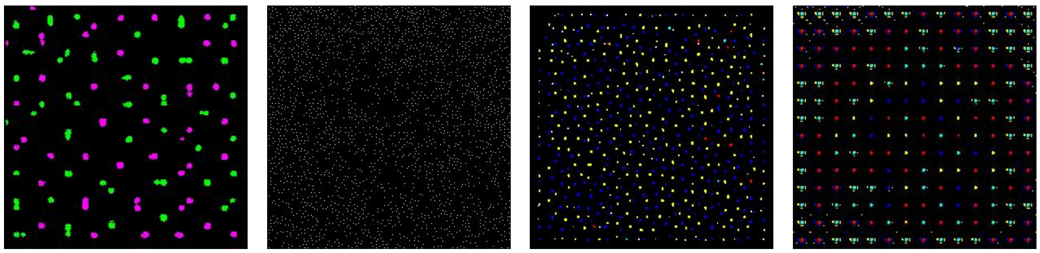


(a) VGG19, $q = 1$ (b) ResNet101, $q = 1$ (c) WideResNet101, $q = 1$ (d) DEIT Base, $q = 1$



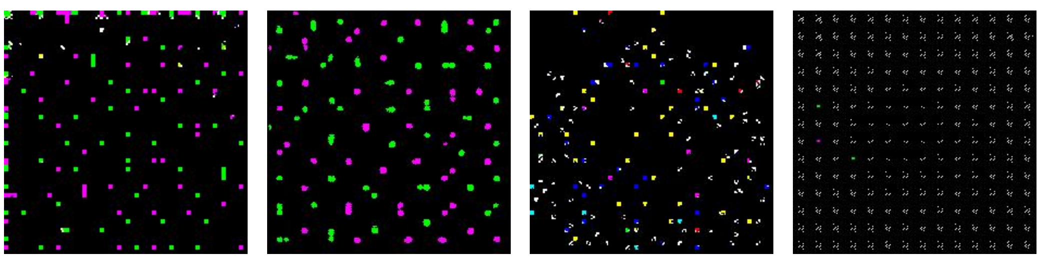
(e) VGG19, $q = 5$ (f) ResNet101, $q = 5$ (g) WideResNet101, $q = 1$ (h) DEIT Base, $q = 3$

Figure 8: UAPs using SV attack algorithm Khrulkov & Oseledets (2018). The first row refers to the fixed parameter value $q = 1$, while the second depicts best-performed perturbations.



(a) (b) (c) (d)

(e) DenseNet161 (f) EfficientNetB3 (g) InceptionV3 (h) ViT Base



(i) (j) (k) (l)

(m) Wide ResNet101 (n) ResNet101 (o) ResNet152 (p) DEIT Base

Figure 9: UAPs and corresponding attacked images obtained using our TPower approach. Perturbations were computed using the best-performed layers on gridsearch.