Transferable Structural Sparse Adversarial Attack Via Exact Group Sparsity Training

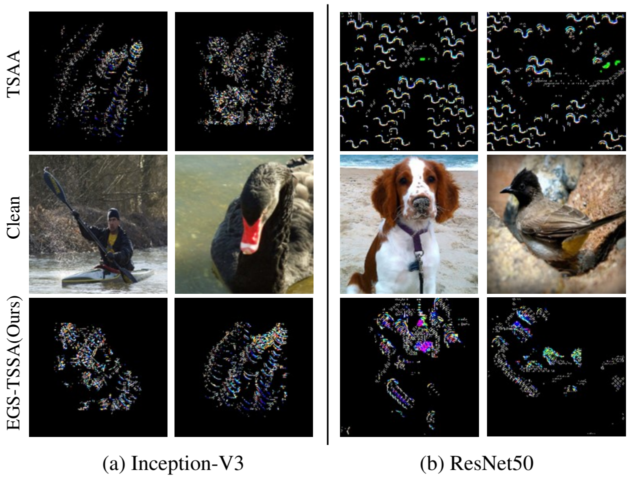
Di Ming, Peng Ren, Yunlong Wang, Xin Feng*
School of Computer Science and Engineering, Chongqing University of Technology
Chongqing, China
diming@cqut.edu.cn, misterr_2019@163.com, ylwang@cqut.edu.cn, xfeng@cqut.edu.cn

Abstract

Deep neural networks (DNNs) are vulnerable to highly transferable adversarial attacks. Especially, many studies have shown that sparse attacks pose a significant threat to DNNs on account of their exceptional imperceptibility. Current sparse attack methods mostly limit only the magnitude and number of perturbations while generally overlooking the location of the perturbations, resulting in decreased performances on attack transferability. A subset of studies indicates that perturbations existing in the significant regions with rich classification-relevant features are more effective. Leveraging this insight, we introduce the structural sparsity constraint in the framework of generative models to limit the perturbation positions. To ensure that the perturbations are generated towards classification-relevant regions, we propose an exact group sparsity training method to learn pixel-level and group-level sparsity. For purpose of improving the effectiveness of sparse training, we further put forward masked quantization network and multi-stage optimization algorithm in the training process. Utilizing CNNs as surrogate models, extensive experiments demonstrate that our method has higher transferability in image classification attack compared to state-of-the-art methods at approximately same sparsity levels. In cross-model ViT, object detection, and semantic segmentation attack tasks, we also achieve a better attack success rate. Code is available at https://github.com/MisterRpeng/EGS-TSSA.

1. Introduction

Deep neural networks (DNNs) have demonstrated remarkable performance in various computer vision tasks, including image classification [12, 17, 35, 38], object detection [13, 19, 21, 32, 39] and semantic segmentation [2, 16, 34]. These high-precision DNNs are crucial to systems requiring robust security, such as autonomous vehicle [20] and facial recognition [44], where errors in classification can lead to significant consequences. However, introducing carefully designed imperceptible perturbations to benign images, known as adversarial examples (AEs), can easily induce prediction errors in these systems [9, 37]. These AEs are especially threatening due to their transferability, meaning adversarial perturbations can deceive not only surrogate models (white-box attack) but can also affect target models never encountered during the attack (black-box attack), revealing a fundamental vulnerability in DNNs [42].



(a) Inception-V3 (b) ResNet50

Figure 1. Distribution of sparse perturbations for various methods. This figure highlights our method's structural sparse perturbations. Unlike the dispersed distribution seen in TSAA [14], our method EGS-TSSA effectively concentrates sparse perturbations crafted by different threat models into classification-relevant regions.

Most current adversarial attack methods employ the ℓ_p norm paradigm to constrain the perturbation generation. For instance, studies [7, 9, 18, 22, 23, 26, 27, 37, 46, 47] commonly utilize ℓ_∞ or ℓ_2 constraints, resulting in dense perturbations. In contrast, sparse adversarial attacks [3, 5, 8, 14, 24, 29, 45, 49] target a limited number of pixels and yet achieve high success rates. However, a key limitation of many sparse attack techniques is their low transferabil-

[*] Corresponding Author: Xin Feng

ity. Transferable sparse adversarial attack (TSAA) [14] enhances the transferability of sparse attacks using generators, but neglects structural constraints on perturbation positions. As shown in the 1st row of Fig. 1, the patterns of perturbations generated by different surrogate models vary greatly, which can reduce the transferability. The pixel-level sparse constraint does not enable the model to learn structural semantic information.

To address this challenging problem, we construct the structural sparsity constraint to help the model learn semantic information, which can not only generate more structured perturbations but also unify the perturbation patterns. [6] have shown that perturbations in classification-relevant regions, which typically have a richer texture, are less detectable and more effective. Inspired by this, to further improve the transferability of sparse perturbations, we make the perturbations exist as much as possible in the overlapping regions where the classification features of different models are important, as shown in the 3rd row of Fig. 1.

To implement structural sparsity constraints, we define the group feature importance, making it easier for the model to find the most vulnerable regions. The masked quantization module is further introduced to ensure that structural sparsity can be trained properly. From the experiments, it is found that overly strong structural sparse constraints lead to the disappearance of perturbations, so we propose an effective multi-stage optimization for sparse training. To validate the effectiveness of our approach, we introduce a novel analytical approach to examine the distribution of perturbations crafted by different threat models using feature importance.

In summary, our paper's contributions include:

- An exact group sparsity training method is proposed to generate transferable structural sparse perturbations that result in consistent perturbation patterns and blend more naturally into classification-relevant features.
- To guarantee that structural sparsity can be learned effectively, we construct a masked quantization network to help discrete perturbation positions to be able to be optimized during sparse training. In Addition, we introduce a multi-stage optimization algorithm to ensure that perturbation positions and sparsity are limited by our constraints while preventing perturbations from vanishing.
- Comprehensive experiments show that our method outperforms state-of-the-art sparse attack methods in terms of transferability. We further validate the effectiveness of our method in attacking ViT and computer vision tasks.

2. Related Work

Magnitude-constrained Adversarial Attacks. The perturbation amplitude is limited by ℓ_∞ or ℓ_2 norm to meet the attack requirements. Szegedy *et al.* [37] pioneer the field with an adversarial attack that uses the L-BFGS method. However, this technique lacks scalability and operates slowly. FGSM [9] constructs faster attacks by calculating the sign of the gradient. I-FGSM [18] and MI-FGSM [7] utilize iteration and momentum to enhance attack performances even further. PGD [23] provides a more powerful attack but does not completely resolve the convergence issues. Universal adversarial perturbation (UAP) [26] is introduced to improve the efficiency of above image-specific attacks. Zhang *et al.* [46] train data-dependent UAP that has dominant features. Contrarily, GD-UAP [27] maximizes convolution activations to generate data-free UAPs. Cosine-UAP [47] and TRM-UAP [22] further boost the transferability via cosine similarity and truncated ratio maximization.

Sparsity-constrained Adversarial Attacks. Limiting the number of perturbations using ℓ_0 or ℓ_1 norm allows the generated perturbations to be sparse enough. JSMA [29] utilizes saliency metrics to identify the most impactful pixel for sparse perturbation. PGD$_0$ [3] crafts sparse perturbation via the projection onto ℓ_0 ball. SparseFool [24] extends the approach of DeepFool [25] to non-targeted sparse attacks. StrAttack [45] optimizes the perturbation magnitude and sparsity alternatively via ADMM, leading to improved outcomes. SAPF [8] factorizes sparse perturbation into continuous magnitudes and binary selection factors to solve a mixed integer programming problem. Homotopy attack [49] leverages accelerated proximal gradient to jointly tackle sparsity and perturbation bound. Nonetheless, these approaches tend to be resource-intensive.

Mask-guided Adversarial Attacks. The generation of perturbations can be enhanced through the guidance of masks typically derived from semantic information. Dong *et al.* [6] crafts superpixel-level perturbations in the most prominent areas of the image's attention map. FIA [41] decreases and increases important features of the perturbation corresponding to positive and negative values in the aggregate gradient mask respectively. Zhang *et al.* [48] adopts the CAM mask to define weighted cosine similarity to craft transferable perturbation across different domains. Wei *et al.* [43] drop out model-specific patches via learnable masks to reduce overfitting in the perturbation training. However, these methods employ masks to generate dense or local perturbations, not the sparse perturbation.

Generator-based Adversarial Attacks. GAP [30] utilizes the generator structure to learn the mapping relationship from original image to adversarial example. Mopuri *et al.* [28] suggest a generative model that leverages class-specific features to craft generic adversarial perturbations. GreedyFool [5] employs GAN-based framework for distortion map generation to craft sparse perturbations. Similarly, TSAA [14] utilizes the generator to create sparse perturbations, with a particular focus on enhancing the transferability of sparse attacks. A common limitation of these methods is the lack of structured constraints on perturbation positions, potentially leading to decreased transferability.

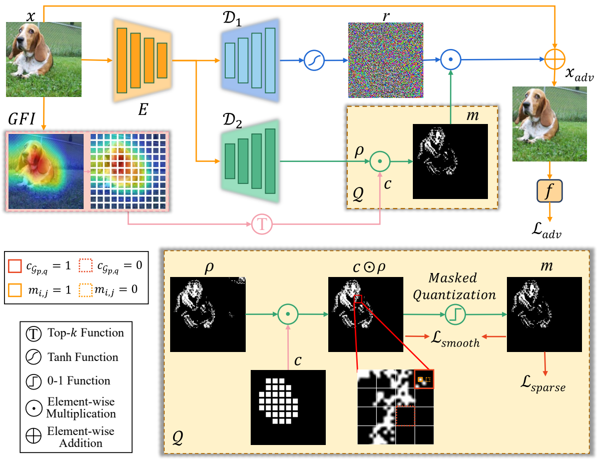


Figure 2. The overall framework of our proposed transferable structural sparse attack method (top: generative network G; bottom: masked quantization network Q).

3. Methodology

3.1. Preliminary

Notations. $\boldsymbol{x} \in [0, 255]^{W\times H\times C}$ and $y \in \{1, \cdots, K\}$ denote the original image (W: width, H: height, C: channel) and its corresponding ground-truth class label (K: total number of classes). $f(\boldsymbol{x})_k$ represents the output logit value of threat model f for class k. Thus, the original benign image should satisfy $\arg\max_k f(\boldsymbol{x})_k = y$. By adding tiny perturbations $\boldsymbol{\delta}$ deliberately to $\boldsymbol{x}$, the crafted adversarial example $\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{\delta}$ satisfies $\arg\max_k f(\boldsymbol{x}_{adv})_k \neq y$.

To guarantee that the perturbation is sparse and imperceptible, sparse adversarial attack methods try to minimize the ℓ_0 loss[1] of the perturbation $\boldsymbol{\delta}$:

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & \|\boldsymbol{\delta}\|_0 \\ \text{s.t.} \quad & \arg\max_k \; f(\boldsymbol{x}+\boldsymbol{\delta})_k \neq y \\ & \|\boldsymbol{\delta}\|_\infty < \epsilon \end{aligned} \tag{1}$$

where ϵ is a constant to constrain the perturbation $\boldsymbol{\delta}$ in the ℓ_∞-norm bound. Compared to non-targeted attack, the inequality constraint is replaced by $\arg\max_k f(\boldsymbol{x}+\boldsymbol{\delta})_k = y_t$ for targeted attack, where y_t is a target label. Due to the high dependence on threat models, sparse adversarial examples have low transferability on target models.

Recently, generative models have attracted considerable attention for boosting the transferability of sparse adversarial examples. He *et al.* [14] propose transferable sparse adversarial attack (TSAA) to reformulate the perturbation as $\boldsymbol{\delta} = \boldsymbol{r} \odot \boldsymbol{m}$ in generative network G, where $\delta_{ijk} = r_{ijk} \cdot m_{ij}$, $\boldsymbol{r} = D_1(E(\boldsymbol{x})) \in [0, 255]^{W\times H\times C}$ and $\boldsymbol{m} = D_2(E(\boldsymbol{x})) \in \{0, 1\}^{W\times H}$ represent the magnitude and position of sparse perturbations, and G contains encoder E and decoders D_1, D_2. Thus, TSAA can be mathematically formulated as

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{(\boldsymbol{x},y)\in\mathbb{D}} \mathcal{L}_{adv}(\boldsymbol{x}+\boldsymbol{r}\odot\boldsymbol{m}, y) + \lambda\|\boldsymbol{m}\|_1 \\ \text{s.t.} \quad & \|\boldsymbol{r}\|_\infty < \epsilon, \; \boldsymbol{r} = D_1(E(\boldsymbol{x})), \; \boldsymbol{m} = D_2(E(\boldsymbol{x})) \end{aligned} \tag{2}$$

where $\mathcal{L}_{adv}(\boldsymbol{x}, y) = \max\left(f(\boldsymbol{x})_y - \max_{k\neq y}\{f(\boldsymbol{x})_k\}, -\kappa\right)$ denotes the C&W [1] adversarial loss (a surrogate loss of the inequality constraint in Eq. 1 for untargeted attack, and $\mathcal{L}_{adv}(\boldsymbol{x}, y_t) = \max_{k\neq y_t}\{f(\boldsymbol{x})_k\} - f(\boldsymbol{x})_{y_t}$ for targeted attack), $\mathbb{D}$ denotes training dataset, and $\boldsymbol{\theta}$ denotes all the parameters in generative network G. In practical attack, sparse adversarial examples can be quickly crafted via pre-trained generator, *i.e.*, $\boldsymbol{x}_{adv} = \boldsymbol{x} + G(\boldsymbol{x})$ for any given input $\boldsymbol{x}$.

3.2. Transferable Structural Sparse Attack

Compared to prior works, TSAA [14] improves the transferability of sparse attack on target models. However, TSAA only relies on the ℓ_1-norm based pixel-level sparsity to train generative network G, without taking consideration of the structured semantic information. Towards further enhancing the transferability across different target models, we are interested in the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{(\boldsymbol{x},y)\in\mathbb{D}} \mathcal{L}_{adv}(\boldsymbol{x}+\boldsymbol{r}\odot\boldsymbol{m}, y) + \mathcal{L}_{sparse}(\boldsymbol{m}) \\ \text{s.t.} \quad & \|\boldsymbol{r}\|_\infty < \epsilon, \; \boldsymbol{r} = D_1(E(\boldsymbol{x})), \; \boldsymbol{m} = D_2(E(\boldsymbol{x})) \end{aligned} \tag{3}$$

and $\mathcal{L}_{sparse}$ is the structural sparsity-inducing penalty term for constraining the perturbation position $\boldsymbol{m}$ defined as:

$$\mathcal{L}_{sparse}(\boldsymbol{m}) = \lambda_1||\boldsymbol{m}||_1 + \lambda_2||\boldsymbol{m}||_{21}^{\mathcal{G}} \tag{4}$$

where ℓ_1-norm controls fine-grained sparsity at pixel level and group ℓ_{21}-norm controls coarse-grained sparsity at group level. We partition $\boldsymbol{m} \in \{0, 1\}^{W\times H}$ into $P \times Q$ equal-sized and non-overlapped groups with a predefined stride S. Thus, group ℓ_{21}-norm can be rewritten as $||\boldsymbol{m}||_{21}^{\mathcal{G}} = \sum_{p=1}^{P}\sum_{q=1}^{Q} ||\boldsymbol{m}_{\mathcal{G}_{p,q}}||_2$, where $\boldsymbol{m}_{\mathcal{G}_{p,q}}$ represents the vector containing the elements of $\boldsymbol{m}$ in (p, q)-th group.

Relaxed Formulation. To decouple the optimization between two sparsity-inducing penalties, an auxiliary variable $\boldsymbol{\rho}$ is introduced. On the other hand, group ℓ_{21}-norm is replaced with group ℓ_{20}-norm to achieve exact k sparsity, which corresponds to a certain λ_2 and can effectively decrease the computation cost of finetuning sparsity hyperparameters. Thus, the original problem (3) becomes to:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{(\boldsymbol{x},y)\in\mathbb{D}} \mathcal{L}_{adv}(\boldsymbol{x}+\boldsymbol{r}\odot\boldsymbol{m}, y, \boldsymbol{\rho}) + \lambda_1||\boldsymbol{m}||_1 \\ \text{s.t.} \quad & \|\boldsymbol{r}\|_\infty < \epsilon, \; \boldsymbol{r} = D_1(E(\boldsymbol{x})), \; \boldsymbol{\rho} = D_2(E(\boldsymbol{x})), \\ & \boldsymbol{m} = Q(\boldsymbol{\rho}), \; ||\boldsymbol{\rho}||_{20}^{\mathcal{G}} = k \end{aligned} \tag{5}$$

where structural sparsity of $\boldsymbol{m}$ is generated via two concatenated networks G and Q (see Fig. 2), Q is the quantization network connecting $\boldsymbol{m}$ and $\boldsymbol{\rho}$ (a smoothness loss

[1] For notational simplicity, $||\boldsymbol{x}||_p$ denotes the ℓ_p-norm of the vectorized $\boldsymbol{x}$, *i.e.*, $||\text{vec}(\boldsymbol{x})||_p$, where $\text{vec}(\cdot)$ transforms any tensor into a vector.

introduced later in Sec. 3.4 to constrain this connection), $||\boldsymbol{\rho}||_{20}^{\mathcal{G}} = \sum_{p=1}^{P}\sum_{q=1}^{Q} I(||\boldsymbol{\rho}_{\mathcal{G}_{p,q}}||_2)$ where $\boldsymbol{\rho}_{\mathcal{G}_{p,q}}$ represents the vector containing the elements of $\boldsymbol{\rho}$ in (p,q)-th group, $I(x) = 1$ if $x \neq 0$ and $I(x) = 0$ if $x = 0$, and the sparse penalty $\mathcal{L}_{sparse}$ in Eqs. 3-4 is reduced to $\lambda_1 ||\boldsymbol{m}||_1$ only.

3.3. Exact Group Sparsity Training

For purpose of efficiently solving (5), we introduce the exact group sparsity training method which combines sparse training [11, 31] with classification-relevant semantics to learn structural sparsity in both pixel-level and group-level.

Exact *k* Group Sparsity. To satisfy group ℓ_{20}-norm based equality constraint, problem (5) is rewritten as:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{(\boldsymbol{x},y)\in\mathbb{D}} \mathcal{L}_{adv}(\boldsymbol{x} + \boldsymbol{r} \odot \boldsymbol{m}, y, \boldsymbol{\rho}, \boldsymbol{c}) + \lambda_1 ||\boldsymbol{m}||_1 \\ \text{s.t.} \quad & \|\boldsymbol{r}\|_\infty < \epsilon, \ \boldsymbol{r} = D_1(E(\boldsymbol{x})), \ \boldsymbol{\rho} = D_2(E(\boldsymbol{x})), \\ & \boldsymbol{m} = Q(\boldsymbol{c} \odot \boldsymbol{\rho}), \sum_{p,q} ||\boldsymbol{c}_{\mathcal{G}_{p,q}}||_1 = kS^2 \end{aligned} \tag{6}$$

where the group mask $\boldsymbol{c}_{\mathcal{G}_{p,q}} = \{\mathbf{0}_{S^2}, \mathbf{1}_{S^2}\}$ indicates whether S^2 elements in $\boldsymbol{\rho}_{\mathcal{G}_{p,q}}$ are selected or not, $\mathbf{0}_{S^2}$ and $\mathbf{1}_{S^2}$ represent the S^2-by-1 vector with all zeros and all ones. Based on this, exact k group sparsity can be achieved through sparse training, which endeavors to identify k most important groups out of total $P \times Q$ groups and prune all the remaining groups for satisfying $||\hat{\boldsymbol{\rho}}||_{20}^{\mathcal{G}} = k$, $\hat{\boldsymbol{\rho}} = \boldsymbol{c} \odot \boldsymbol{\rho}$.

Group Feature Importance. To incorporate classification-relevant semantics into sparse training, we further propose the group feature importance based on class activation map (*e.g.*, Grad-CAM [33]) to generate exact k group sparsity.

With respect to a selected intermediate layer of the surrogate model, feature importance $\boldsymbol{FI} \in \mathbb{R}^{W \times H}$ can be computed as $\boldsymbol{FI} = \mathcal{P}_{\mathcal{O}}\big(\sum_d (\sum_i \sum_j Grad_{i,j}^{(d)}) \cdot \boldsymbol{Feat}^{(d)}\big)$, where $\boldsymbol{Grad}^{(d)}$ and $\boldsymbol{Feat}^{(d)}$ represent gradient and feature matrices in d-th channel, and $\mathcal{P}_{\mathcal{O}}(\cdot)$ upsamples any variable $\boldsymbol{x}$ to the original space $\mathcal{O}$, *i.e.*, $\mathcal{P}_{\mathcal{O}}(\boldsymbol{x}) \in \mathbb{R}^{W \times H}$. Building upon this, we define the group feature importance $\boldsymbol{GFI} \in \mathbb{R}^{P \times Q}$ w.r.t. (p,q)-th group as:

$$GFI_{p,q} = ||\boldsymbol{FI}_{\mathcal{G}_{p,q}}||_2 \tag{7}$$

where $\boldsymbol{FI}_{\mathcal{G}_{p,q}}$ denotes the vector containing the elements of $\boldsymbol{FI}$ in (p,q)-th group. This group ℓ_{21}-norm based value represents the aggregated classification-relevant importance of each group $\mathcal{G}_{p,q}$. Thus, $\boldsymbol{GFI}$ can be ranked to obtain the index set of top-k largest values via $\mathcal{K} = top(\boldsymbol{GFI}, k)$. If $(p,q) \in \mathcal{K}$, $\boldsymbol{c}_{\mathcal{G}_{p,q}} = \mathbf{1}_{S^2}$. Otherwise, $\boldsymbol{c}_{\mathcal{G}_{p,q}} = \mathbf{0}_{S^2}$.

It's worth noting that the role of $\boldsymbol{GFI}$ differs significantly in training and testing phases. In the training phase, $\boldsymbol{GFI}$ guides the generator to learn structured perturbation positions for better transferability. For a fair comparison, in the testing phase, $\boldsymbol{GFI}$ limits the perturbation to sparsity levels consistent with other methods. Appendix A provides further details regarding the perturbation generation.

3.4. Masked Quantization Network

As standard quantization network can only generate pixel-level sparsity for the perturbation, in the following, we describe how to integrate aforementioned exact k group sparsity into a novel masked quantization network Q to generate structural sparse perturbation and stabilize the training.

Masked Quantization. Based on the Bernoulli distribution $B(p)$ and the group mask $\boldsymbol{c}$, structural sparse position $\boldsymbol{m}$ is generated through the network Q. Given the output $\boldsymbol{\rho}$ of the decoder D_2, each element $\rho_{i,j}$ is randomly sampled to be quantized using $X \sim B(p)$:

$$m_{i,j} = \begin{cases} \rho_{i,j} & X = 1 \\ q(\rho_{i,j}) & X = 0 \end{cases} \tag{8}$$

where $q(\cdot)$ is the masked quantization function defined as

$$q(\rho_{i,j}) = \begin{cases} 0 & \rho_{i,j} \cdot c_{i,j} \le \tau \\ 1 & \rho_{i,j} \cdot c_{i,j} > \tau \end{cases} \tag{9}$$

and τ is the threshold. Thus, in each epoch, about $(100 \cdot p)\%$ of elements in $\boldsymbol{\rho}$ will be updated through back-propagation.

Masked Smoothness Loss. As can be seen, some elements in $\boldsymbol{m}$ have been quantized to 0 or 1, which could be treated as pseudo labels to guide the learning of $\boldsymbol{\rho}$. Towards generating structural sparsity effectively, we further introduce the smoothness loss to enforce variables $\boldsymbol{\rho}$ and $\boldsymbol{m}$ to be close:

$$\mathcal{L}_{smooth} = \begin{cases} ||\boldsymbol{c} \odot \boldsymbol{\rho} - \boldsymbol{c} \odot \boldsymbol{m}||_2^2 & \text{(hard)} \\ ||\boldsymbol{\rho} - \boldsymbol{c} \odot \boldsymbol{m}||_2^2 & \text{(soft)}, \end{cases} \tag{10}$$

where the hard loss $\mathcal{L}_{smooth}^{(hard)}$ can learn structural sparse positions within the group mask $\boldsymbol{c}$, and the soft loss $\mathcal{L}_{smooth}^{(soft)}$ can learn structural sparse positions beyond the group mask $\boldsymbol{c}$. More details of their difference are provided in Appendix B. On the other hand, this smoothness loss also can guarantee the convergence of optimization, which works as the regularization term to minimize the difference between original variable $\boldsymbol{m}$ and auxiliary variable $\boldsymbol{\rho}$ defined in Eq. 5.

3.5. Multi-Stage Optimization Algorithm

Overall Loss. To combine masked quantization network with exact group sparsity training, we define the overall loss of the proposed transferable structural sparse attack as:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_1 \mathcal{L}_{sparse} + \lambda_2 \mathcal{L}_{smooth} \tag{11}$$

where $\mathcal{L}_{smooth}$ can be hard or soft masked smoothness loss, and λ_1, λ_2 are hyperparameters balancing relative importance between different losses and adjusting sparsity levels.

Multi-Stage Optimization. In the preliminary results, we found out that structural sparsity is difficult to minimize within generative models. If setting λ_1 as a larger value, the ℓ_1 loss of perturbation position will quickly become exact

Algorithm 1 Multi-Stage Optimization Algorithm

```
Input: Dataset $\mathbb{D}$, benign image $\boldsymbol{x}$, ground truth label $y$, surrogate model $f$, maximum perturbation bound $\epsilon$, maximum iteration number $T$, learning rate $\alpha$, partition stride $S$, group sparsity level $k$, quantization threshold $\tau$, Bernoulli distribution $B(p)$, hyperparameters $\lambda_1$, $\lambda_2$
Output: All the parameters $\boldsymbol{\theta}$ in generative network $G$
Randomly initialize the generative network $G$.
for $\lambda_1, \lambda_2, T$ in $[\underbrace{(\lambda_1^{(1)}, \lambda_2^{(1)}, T^{(1)})}_{\text{“1st stage”}}, \underbrace{(\lambda_1^{(2)}, \lambda_2^{(2)}, T^{(2)})}_{\text{“2nd stage”}}]$ do
    for $t = 0, 1, \cdots, T$ do
        for $(\boldsymbol{x}, y) \sim \mathbb{D}$ do
            Generate magnitude $\boldsymbol{r} = D_1(E(\boldsymbol{x}))$
            Calculate $\boldsymbol{GFI}$ via Eq. (7)
            Generate group mask $\boldsymbol{c}$ via $top(\boldsymbol{GFI}, k)$
            Generate auxiliary variable $\boldsymbol{\rho} = D_2(E(\boldsymbol{x})))$
            Quantize posistion $\boldsymbol{m}$ via Eqs. (8-9)
            Generate adversarial example $\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{r} \odot \boldsymbol{m}$
            Calculate overall loss $\mathcal{L} = \mathcal{L}_{adv}(\boldsymbol{x}_{adv}, y)$
                $+ \lambda_1 \mathcal{L}_{sparse}(\boldsymbol{m}) + \lambda_2 \mathcal{L}_{smooth}(\boldsymbol{\rho}, \boldsymbol{m}, \boldsymbol{c})$
            Update parameters $\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}$
        end for
    end for
end for
Return $\boldsymbol{\theta}$
```

zero after several training iterations, due to the imbalance between different loss values (see Appendix C.1). However, a smaller λ_1 cannot generate sparse perturbation and attain the desired sparsity level consistent with other methods. To resolve this unstable training problem, we propose a multi-stage optimization algorithm. Specifically, in the first stage of training, we encourage the model to search the most vulnerable positions only under the mild guidance of group mask, without any sparsity constraint on the perturbation position, *i.e.*, $\lambda_1^{(1)} = 0$ and $\lambda_2^{(1)} > 0$. In the second stage of training, the ℓ_1 sparsity constraint is used to generate a more sparse adversarial perturbation under the stronger guidance of group mask, *i.e.*, $\lambda_1^{(2)} > 0$ and $\lambda_2^{(2)} > 0$ ($\lambda_2^{(2)} \gg \lambda_2^{(1)}$). The entire procedure of our proposed transferable structural sparse attack is summarized in Algorithm 1.

4. Experiment

4.1. Experimental Setting

Setups. For generator training, we employed Inception-V3 (IncV3) [38] and Resnet50 (Res50) [12] as surrogate models. Utilizing the mapping relations learned by these generators, we could quickly generate corresponding adversarial examples from original images. The IncV3 model required cropping the input image to 299×299, while Res50 required 224×224. Additionally, we employed VGG16 [35] and Densenet161 (Dense161) [17] as target models.

Benchmark Datasets. In this study, we utilized the widely used Imagenet dataset [4] as our training set. To ensure a fair comparison with other sparse attacks, we utilized the same test set as TSAA [14], which is 5000 images randomly selected from the test set in the Imagenet dataset.

Evaluation Metrics. Sparsity was determined by the average perturbation rate of all adversarial examples in the test set. To evaluate the transferability of adversarial perturbations, we used the average attack success rate (ASR) across all models. All experimental results are expressed as (%).

Comparative Methods. We compare four standard sparse attack methods: PGD$_0$ [3], SparseFool [24], GreedyFool [5], TSAA [14]. We use the official implementation to finetune it to meet our sparsity requirements.

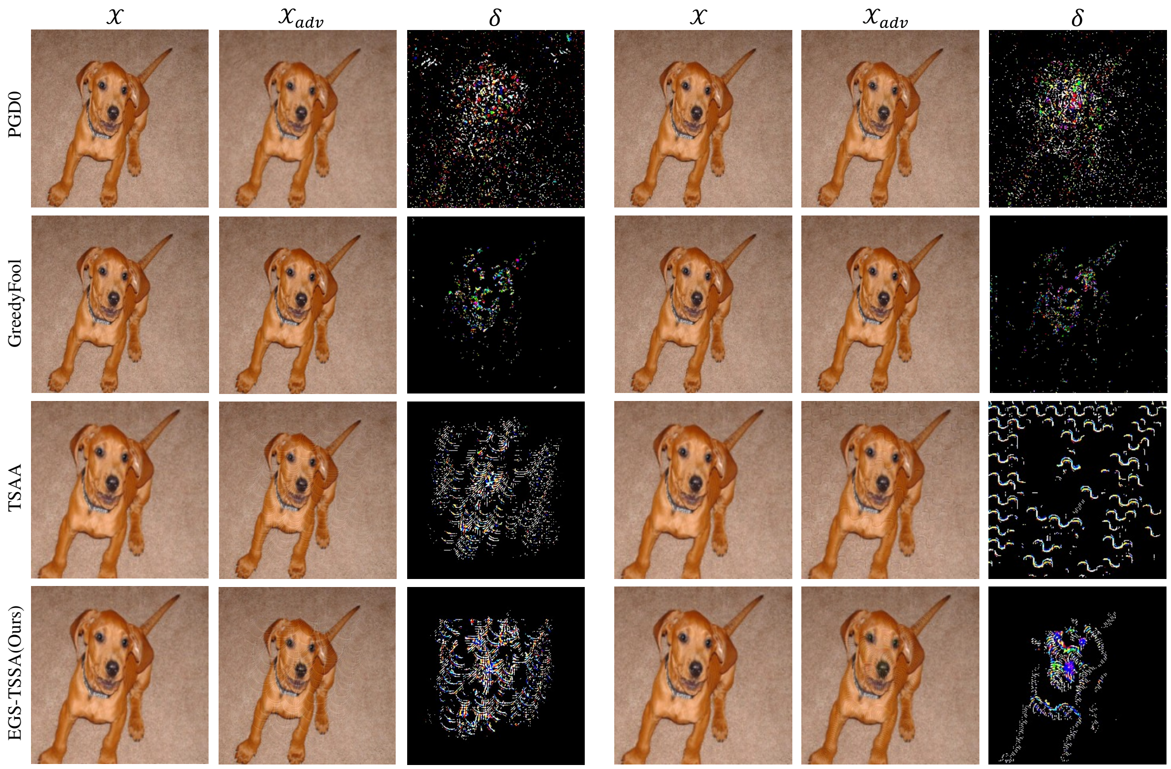
Implementation Details. All experiments were conducted using PyTorch on an NVIDIA V100 Tensor Core GPU. For the C&W loss, we set $\kappa = 0$ and a binarization threshold of $\tau = 0.5$. The granularity (*i.e.*, stride S) of group sparsity was optimized to 13×13 for IncV3 and 8×8 for Res50. To address training convergence issues, we adjusted the Bernoulli distribution probability p based on perturbation sizes. In selecting the top-k value, we determined the most effective k to be 0.6 through preliminary exploratory experiments. Furthermore, sparsity hyperparameters λ_1 and λ_2 were adjusted to accommodate different sparsity levels.

4.2. Comparison with State-of-the-Art Methods

In this section, we assessed the perturbation's transferability under varying ℓ_∞ constraints. Results of white-box and black-box attacks were shown in different sparsity levels.

Experimental results on the Imagenet dataset with $\ell_\infty = 10$ are presented in Table 1. When using IncV3 as surrogate model, ASR was observed to be superior to TSAA for both soft and hard constraints. Notably, since soft constraints are easier to optimize, ASR was higher under these conditions in our experiments. The improvement in transferability with Res50 as surrogate model was quite apparent. However, for black-box attack on Dense161, the performance unexpectedly didn't perform as well as TSAA. This discrepancy is attributed to the alteration in the original perturbation generation pattern of Res50, making it more suited to VGG16 than Dense161. With $\ell_\infty = 255$, as shown in Table 2, our method's transferability still exceeded that of TSAA. We compared the results of our approach with TSAA in Fig. 3. It was observed that our method's perturbations were more structured and more focused on classification targets.

In terms of inference speed comparison on IncV3, Res50 models, from low to high: GreedyFool (*73.581s, 17.619s*), PGD0 (*55.184s, 16.041s*), SparseFool (*14.78s, 6.354s*), EGS-TSSA (*0.0096s, 0.0115s*), TSAA (*0.0057s, 0.0056s*). Our EGS-TSSA closely matches the TSAA method.



(a) Inception-V3

(b) ResNet50

Figure 3. Comparison of perturbation pattern across different sparse adversarial attack methods. Our EGS-TSSA method produces perturbations that are noticeably more concentrated and more structured as compared to other methods.

Surrogate	Method	Sparsity	IncV3	Res50	VGG16	Dense161	Average	Average$_{bb}$
IncV3	PGD0	14.54	97.89*	9.70	12.73	8.16	32.12	10.20
	SparseFool	1.65	99.98*	4.94	9.10	4.08	29.53	6.04
	SparseFool(λ=10)	12.56	100.00*	7.99	12.63	11.40	33.01	10.67
	GreedyFool	0.55	100.00*	0.94	0.58	2.08	25.90	1.20
	GreedyFool(κ=40)	18.19	100.00*	10.67	11.24	6.67	32.15	9.53
	TSAA	14.47	87.72*	45.32	50.38	28.98	53.10	41.56
	EGS-TSSA$_{hard}$(Ours)	14.43	92.14*	45.42	54.56	35.70	56.96	45.23
	EGS-TSSA$_{soft}$(Ours)	14.14	91.66*	**47.42**	**57.16**	**39.26**	**58.88**	**47.95**
Res50	PGD0	9.96	11.38	99.54*	21.42	20.74	38.27	17.85
	SparseFool	1.27	2.92	99.96*	2.94	2.02	26.96	2.63
	SparseFool(λ=15)	9.72	11.87	100.00*	13.39	14.23	34.87	13.16
	GreedyFool	0.59	3.20	100.00*	2.76	1.42	26.85	2.46
	GreedyFool(κ=30)	12.64	12.35	100.00*	17.09	20.89	37.58	16.78
	TSAA	10.52	9.20	72.90*	39.48	**51.18**	43.19	33.29
	EGS-TSSA$_{hard}$(Ours)	10.48	14.70	84.52*	57.54	39.90	49.17	37.38
	EGS-TSSA$_{soft}$(Ours)	10.52	**14.78**	91.26*	**57.66**	44.86	**52.14**	**39.10**

Table 1. Comparison of non-targeted attack transferability on the Imagenet dataset under $\ell_\infty = 10$ perturbation magnitude constraints. "*" denotes white-box setting, "Average" and "Average$_{bb}$" denote the average ASR of all models and all black-box models respectively.

4.3. Practical Structural Sparse Attack

In all following experiments, the best-trained weights were used for testing on a variety of attack tasks. Additional comparison results are provided in Appendix C.5 and C.6.

Attack on Target Label. Following the same setting used in TSAA with $\ell_\infty = 255$, we selected the target category "bubble" (ID: 971) for a targeted attack comparison. For IncV3 and Res50 as surrogate models, as shown in Table 3, we found that the transferability of our targeted attack was the best, improving the average ASR by a large margin.

Attack on ViT. We further evaluated the performance of

Surrogate	Method	Sparsity	IncV3	Res50	VGG16	Dense161	Average	Average$_{bb}$
IncV3	PGD0	0.56	56.50*	21.95	23.60	9.69	27.94	18.41
	SparseFool	0.26	99.90*	7.34	14.24	5.04	31.63	8.87
	SparseFool(λ=10)	0.52	100.00*	11.76	24.50	6.96	35.81	14.41
	GreedyFool	0.11	100.00*	2.16	5.38	1.38	27.23	2.97
	GreedyFool(κ=15)	0.67	100.00*	15.09	26.37	11.94	38.35	17.80
	TSAA	0.46	61.24*	63.76	85.94	46.22	64.29	65.31
	EGS-TSSA$_{hard}$(Ours)	0.46	70.32*	64.94	86.56	44.66	66.62	65.39
	EGS-TSSA$_{soft}$(Ours)	0.45	72.02*	**67.86**	**86.68**	**47.06**	**68.41**	**67.20**
Res50	PGD0	0.60	20.54	75.74*	43.50	16.72	39.13	26.92
	SparseFool	0.41	21.56	98.74*	25.34	9.90	38.89	18.93
	SparseFool(λ=10)	0.66	27.18	100.00*	35.40	13.56	44.04	25.38
	GreedyFool	0.22	2.52	100.00*	8.88	1.80	28.30	4.40
	GreedyFool(κ=15)	0.75	29.12	100.00*	43.88	30.09	50.77	34.36
	TSAA	0.59	25.90	79.04*	**85.96**	60.18	62.77	57.35
	EGS-TSSA$_{hard}$(Ours)	0.59	41.00	83.82*	81.98	61.78	67.15	61.59
	EGS-TSSA$_{soft}$(Ours)	0.59	**41.62**	84.08*	82.76	**61.78**	**67.56**	**62.05**

Table 2. Comparison of non-targeted attack transferability on the Imagenet dataset under $\ell_\infty = 255$ perturbation magnitude constraints. "*" denotes white-box setting, "Average" and "Average$_{bb}$" denote the average ASR of all models and all black-box models respectively.

Surrogate	Method	Sparsity	IncV3	Res50	VGG16	Dense161	Average	Average$_{bb}$
IncV3	PGD0	0.56	0.00*	2.25	6.50	0.38	2.28	3.04
	GreedyFool	0.42	99.90*	0.10	0.16	0.06	25.06	0.11
	TSAA	0.55	35.38*	10.38	9.08	3.66	14.63	7.71
	EGS-TSSA(Ours)	0.54	54.34*	**34.68**	**53.72**	**24.54**	**41.82**	**37.65**
Res50	PGD0	0.82	0.40	1.52*	1.74	0.88	1.14	1.01
	GreedyFool	0.75	0.90	95.82*	0.94	7.22	25.06	3.02
	TSAA	0.64	0.42	12.64*	10.90	8.10	8.02	6.47
	EGS-TSSA(Ours)	0.64	**2.22**	63.18*	**30.48**	**19.06**	**28.74**	**17.25**

Table 3. Comparison of targeted attack transferability on the Imagenet dataset under $\ell_\infty = 255$ perturbation magnitude constraints. Target category is "Bubble" (ID: 971). "*" denotes white-box setting, "Average" and "Average$_{bb}$" denote the average ASR of all models and all black-box models respectively.

Surrogate	Method	ViT-S/32[36]	DeiT-T/16[40]	PiT-T[10]	LeViT-128[15]	Average
Incv3	GreedyFool	1.08	1.50	2.10	1.06	1.44
	TSAA	25.88	13.04	15.98	10.70	16.40
	EGS-TSSA(Ours)	**32.18**	**13.20**	**16.82**	**11.54**	**18.44**
Res50	GreedyFool	1.14	1.94	1.66	0.78	1.38
	TSAA	23.54	9.04	8.00	5.14	11.43
	EGS-TSSA(Ours)	**28.72**	**9.60**	**9.22**	**8.76**	**14.08**

Table 4. Transferability of various attacks on ViT models. The evaluation metric for all ViT experiments was the ASR.

Surrogate	Method	FCN[34]	DeepLabV3[2]	LR-ASPP[16]	Average
	w/o attack	64.01	69.11	60.77	64.63
incv3	TSAA	27.34	46.10	**40.26**	37.90
	EGS-TSSA(Ours)	**24.52**	**43.99**	41.02	**36.51**
res50	TSAA	45.73	58.77	54.60	53.03
	EGS-TSSA(Ours)	**34.04**	**49.18**	**47.26**	**43.50**

Table 5. Performance of various attacks in segmentation task. The evaluation metric for all segmentation experiments was the mIoU.

adversarial examples generated on white-box CNNs to attack black-box ViTs [10, 15, 36, 40], the results of which are presented in Table 4. All the ViT results are obtained directly using training weights for untargeted attack with $\ell_\infty = 255$, so the sparsity level remains the same as in Table 2. The image size needs to be adjusted to 224×224 when the surrogate model is incv3 to meet the input requirements of ViT. We found that the transferability was significantly reduced in both TSAA and our method. Our analysis suggests perturbations with high transferability generated using CNN models have a relatively strong structure and the perturbations are continuous. ViTs divided the image into many patches, which might have destroyed this structure.

Attack on Semantic Segmentation. We also attacked the semantic segmentation task using pretrained generator under non-targeted and $\ell_\infty = 255$ settings. The validation set of VOC2012 dataset was used as the test data. As evidenced by the mean intersection over union (mIoU) metric in Table 5, exploiting structural sparse attack similarly led to significant degradation in model segmentation performance.

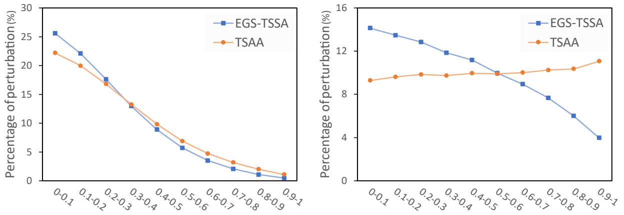
Attack on Object Detection. Moreover, we further attack the object detection model using generators trained by non-targeted attack with $\ell_\infty = 255$, where the validation set of COCO dataset is used as test data. The results in Table 6 indicate our structural sparse perturbations cause significant degradation in model detection performance, which is evaluated by average precision (AP) and average recall (AR) using predefined setting of averaged over IoU $= [0.5 : 0.95]$.

Surrogate	Method	MaskRCNN[13]		FasterRCNN[32]		SSD300[21]		RetinaNet[19]		FCOS[39]		Average	
		AP	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR
w/o attack		37.85	52.00	36.95	50.85	25.11	36.47	36.36	53.97	39.19	56.23	35.09	49.90
incv3	TSAA	25.70	39.51	23.81	37.46	19.39	29.71	22.48	40.05	26.42	43.73	23.56	38.09
	EGS-TSSA(Ours)	**24.81**	**38.24**	**22.95**	**36.26**	**18.55**	**29.00**	**21.70**	**39.08**	**25.89**	**43.24**	**22.78**	**37.16**
res50	TSAA	24.68	37.49	22.97	35.50	21.59	32.22	20.97	37.47	26.21	42.65	23.29	37.07
	EGS-TSSA(Ours)	**23.89**	**36.99**	**21.47**	**34.16**	**18.38**	**28.16**	**17.80**	**34.48**	**22.41**	**38.75**	**20.79**	**34.51**

Table 6. Performance of various attacks in object detection task. Compared to TSAA, our EGS-TSSA results in a more significant reduction in object detection performance. AP and AR denote average precision and average recall respectively, with all experimental results averaged over IoU $= [0.5 : 0.95]$.

4.4. Analyze the Property of Sparse Perturbation

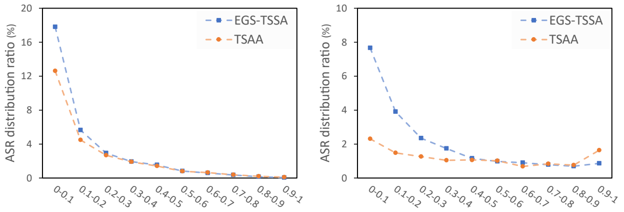
Distribution of Sparse Perturbation. Our enhanced top-k module divides the predefined $P \times Q$ groups into 10 regions with respect to $\boldsymbol{GFI}$, see Appendix C.8. As seen in Fig. 4 (a), after applying the $\boldsymbol{GFI}$ bootstrap constraints, the IncV3 model's perturbations concentrated more on regions of classification importance. Fig. 4 (b) highlights the significant difference in perturbation distribution before and after applying these constraints. With the $\boldsymbol{GFI}$ constraints, the perturbation distribution for the Res50 model almost reversed compared to its earlier pattern, concentrating more on regions where classification features are important. This led to a certain degree of unification in the perturbation distribution between the IncV3 and Res50 models.



(a) Inception-V3

(b) ResNet50

Figure 4. Comparative analysis of perturbation distributions. Using the selected 10 regions, we count the perturbations crafted by EGS-TSSA and TSAA. It is apparent that our approach harmonizes the perturbation distributions of two surrogate models.

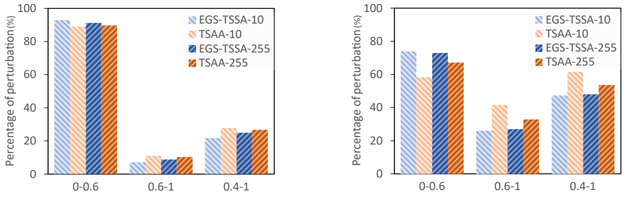


(a) Inception-V3

(b) ResNet50

Figure 5. ASR for each regional perturbation. The ASR is calculated for the perturbations of each region in Fig.4, and the results show that our method produces perturbations with higher ASR in feature-important regions .

Effectiveness of Sparse Perturbation. We further analyzed the attack success rate of different regional perturbations, as shown in Fig. 5. The results indicated that perturbations constrained by $\boldsymbol{GFI}$, when closer to feature-important regions, contributed more significantly to the overall ASR. From Fig. 6, we also can see that our EGS-TSSA method concentrates the perturbations of different models more on classification-relevant important regions.



(a) Inception-V3

(b) ResNet50

Figure 6. Perturbation distributions generated by different methods. "0-0.6" indicates that the most important 60% of the region of perturbation is selected based on $\boldsymbol{GFI}$, "0.6-1" is the remaining 40% of the region, and "0.4-1" indicates the selection of the less important 60%. The vertical coordinates display the percentage of perturbations in the corresponding regions.

5. Conclusion

In this paper, we propose a novel approach for generating structural sparse perturbations, which not only enhances their transferability but also makes them more imperceptible. Unlike traditional sparse attacks, our strategy places greater emphasis on the perturbation position to improve the structure of sparse perturbations. We realize structural sparsity constraint via exact group sparsity training, and also introduce masked quantization network as well as multi-stage optimization algorithm to improve the effectiveness of sparse training. Extensive experiments demonstrate the superior transferability of our structural sparse attack.

Acknowledgement. This work is supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN202301142), the Scientific Research Foundation of Chongqing University of Technology (Grant No. 2022ZDZ026), the Natural Science Foundation of Chongqing, China (Grant No. CSTB2022NSCQ-MSX0493), the Key Project of Chongqing Technology Innovation and Application Development (Grant No. cstc2021jscx-dxwtBX0018).

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 3

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 2017. 1, 7

[3] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4724–4732, 2019. 1, 2, 5

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[5] Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 5

[6] Xiaoyi Dong, Jiangfan Han, Dongdong Chen, Jiayang Liu, Huanyu Bian, Zehua Ma, Hongsheng Li, Xiaogang Wang, Weiming Zhang, and Nenghai Yu. Robust superpixel-guided attentional adversarial attack. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 12892–12901, 2020. 2

[7] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, and Jianguo Hu, Xiaolin Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 1, 2

[8] Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In *European Conference on Computer Vision*, pages 35–50, 2020. 1, 2

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2

[10] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: A vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021. 7

[11] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016. 4

[12] 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*, pages 770–778, 2016. 1, 5

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 1, 8

[14] 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*, pages 14963–14972, 2022. 1, 2, 3, 5

[15] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021. 7

[16] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 1, 7

[17] 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*, pages 4700–4708, 2017. 1, 5

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshops*, 2017. 1, 2

[19] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2999–3007, 2017. 1, 8

[20] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1028–1035, 2019. 1

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, 2016. 1, 8

[22] Yiran Liu, Xin Feng, Yunlong Wang, Wu Yang, and Di Ming. Trm-uap: Enhancing the transferability of data-free universal adversarial perturbation via truncated ratio maximization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4739–4748, 2023. 1, 2

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2

[24] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9087–9096, 2019. 1, 2, 5

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 2

[26] 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*, pages 1765–1773, 2017. 1, 2

[27] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2452–2465, 2018. 1, 2

[28] Konda Reddy Mopuri, Phani Krishna Uppala, and R. Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *European Conference on Computer Vision*, pages 20–35, 2018. 2

[29] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Proceedings of the European Symposium on Security and Privacy*, pages 372–387, 2016. 1, 2

[30] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 2

[31] Md Aamir Raihan and Tor Aamodt. Sparse weight activation training. In *Advances in Neural Information Processing Systems*, pages 15625–15638, 2020. 4

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 1, 8

[33] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2016. 4

[34] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 2017. 1, 7

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 5

[36] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. 7

[37] 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. 1, 2

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 1, 5

[39] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9626–9635, 2019. 1, 8

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10347–10357, 2021. 7

[41] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7619–7628, 2021. 2

[42] Futa Waseda, Sosuke Nishikawa, Trung-Nghia Le, Huy H. Nguyen, and Isao Echizen. Closer look at the transferability of adversarial examples: How they fool different models differently. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1360–1368, 2023. 1

[43] Xingxing Wei and Shiji Zhao. Boosting adversarial transferability with learnable patch-wise masks. *IEEE Transactions on Multimedia*, 26:3778–3787, 2024. 2

[44] Zihao Xiao, Xianfeng Gao, Chilin Fu, Yinpeng Dong, Wei Gao, Xiaolu Zhang, Jun Zhou, and Jun Zhu. Improving transferability of adversarial patches on face recognition with generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11845–11854, 2021. 1

[45] 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. 1, 2

[46] 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*, pages 14521–14530, 2020. 1, 2

[47] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Data-free universal adversarial perturbation and black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7868–7877, 2021. 1, 2

[48] Qilong Zhang, Xiaodan Li, Yuefeng Chen, Jingkuan Song, Lianli Gao, Yuan He, and Hui Xue. Beyond imagenet attack: Towards crafting adversarial examples for black-box domains. In *International Conference on Learning Representations*, 2022. 2

[49] Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. Sparse and imperceptible adversarial attack via a homotopy algorithm. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12868–12877, 2021. 1, 2