Supplementary Materials for "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

Overview

In this supplementary material, we provide additional explanations, experiments, and analyses to support our findings. We also provide PyTorch code, demo and pre-trained models (under untargeted and targeted attack settings with varying ℓ_∞ constraints) at: https://github.com/MisterRpeng/EGS-TSSA.

Contents

A. Additional Explanation of the Generation of Structural Sparse Perturbation

Gradient Descent based Sparsification. First of all, we will show that the gradient descent based optimization algorithm used by TSAA is incompatible with our concerned structural sparse constraint problem. Given $(i, j) \in \mathcal{G}_{p,q}$, the partial derivative of $\mathcal{L}_{sparse}$ defined in Eq. 4 w.r.t. $m_{i,j}$ can be computed as follows:

$$\frac{\partial \mathcal{L}_{sparse}(\boldsymbol{m})}{\partial m_{i,j}} = \lambda_1 \text{sgn}(m_{i,j}) + \lambda_2 \frac{m_{i,j}}{||\boldsymbol{m}_{\mathcal{G}_{p,q}}||_2}.$$

Based on this, the perturbation position $m_{i,j}$ can be iteratively updated via gradient descent:

$$m_{i,j}^{(t+1)} = m_{i,j}^{(t)} - \alpha^{(t)} \frac{\partial \mathcal{L}_{sparse}(\boldsymbol{m})}{\partial m_{i,j}},$$

where $\alpha^{(t)}$ is the adapative learning rate for t-th iteration.

*Corresponding Author: Xin Feng

Since partial derivatives of ℓ_1-norm and group ℓ_{21}-norm w.r.t. $m_{i,j}$ are mixed with each other, group sparsity w.r.t. $\mathcal{G}_{p,q}$ cannot be obtained via gradient descent. To be specific, whenever the partial derivative of group ℓ_{21}-norm (*i.e.*, $\frac{\alpha\lambda_2 m_{i,j}}{||\boldsymbol{m}_{\mathcal{G}_{p,q}}||_2}$) updates $\boldsymbol{m}_{\mathcal{G}_{p,q}}$ to a vector with all zero values, the partial derivative of ℓ_1-norm w.r.t. a certain index $(i,j) \in \mathcal{G}_{p,q}$ (*i.e.*, $\alpha\lambda_1 \text{sgn}(m_{i,j})$) can change the value of $m_{i,j}$ back to nonzero, leading to $||\boldsymbol{m}_{\mathcal{G}_{p,q}}||_2 > 0$.

Cascaded Sparsification. In practice, it is difficult to find appropriate values of hyperparameters λ_1 and λ_2 that can generate the exact group sparsity in the optimization framework of gradient descent. On the other hand, the augmented Lagrange multiplier is the widely used method to solve multiple sparsity-induced penalties. However, the issue of convergence persists when alternatively updating between the generator parameter (via gradient descent) and the Lagrange multiplier (via linear update rule). Thus, the optimization problem (3) does not have the trivial solution.

For the purpose of effectively resolving this issue, we introduce an auxiliary variable $\boldsymbol{\rho}$ to decouple the optimization between two sparsity-induced penalties, propose the masked quantization network Q to connect original variable $\boldsymbol{m}$ and auxiliary variable $\boldsymbol{\rho}$, and further incorporate exact k group sparsity (generated based on group feature importance) into our end-to-end training procedure.

Overall, the generation of structural sparse perturbation is decomposed into two-step cascaded sparsification through masked quantization network: (i) coarse-grained sparsity at the group level is guaranteed by $\boldsymbol{c} \odot \boldsymbol{\rho}$ where $\boldsymbol{c}$ is the group mask (see Eqs. 7-10), after element-wise multiplication only k groups in $\boldsymbol{\rho}$ are kept, *i.e.*, $||\boldsymbol{m}||_{20}^{\mathcal{G}} = k$; (ii) fine-grained sparsity at the pixel level is guaranteed by ℓ_1 penalty $||\boldsymbol{m}||_1$, resulting in sparse values of $\boldsymbol{m}$ in those remaining k groups, *e.g.*, $m_{i,j} = 0$, $(i,j) \in \mathcal{G}_{p,q}$, $\boldsymbol{m}_{\mathcal{G}_{p,q}} \neq 0_{S^2}$.

B. Additional Explanation of the Difference between Hard and Soft Smoothness Losses

Hard Smoothness Loss. Since all the trainable variables are constrained by a mask, hard smoothness loss can only learn the structural sparsity within the group mask $\boldsymbol{c}$:

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

where pseudo labels (already quantized to 0 and 1 values) in $\boldsymbol{m}$ beyond the group mask are reset to exact zeros after element-wise multiplication w.r.t. $\boldsymbol{c}$.

Soft Smoothness Loss. On the contrary, soft smoothness loss has a certain freedom to learn the structural sparsity of some positions beyond the group mask $\boldsymbol{c}$:

$$\begin{aligned}\mathcal{L}_{smooth}^{(soft)}(\boldsymbol{\rho}, \boldsymbol{m}, \boldsymbol{c}) &= ||\boldsymbol{\rho} - \boldsymbol{c} \odot \boldsymbol{m}||_2^2 \\ &= ||\boldsymbol{c} \odot \boldsymbol{\rho} + (1 - \boldsymbol{c}) \odot \boldsymbol{\rho} - \boldsymbol{c} \odot \boldsymbol{m})||_2^2 \\ &= ||\boldsymbol{c} \odot (\boldsymbol{\rho} - \boldsymbol{m})||_2^2 + ||(1 - \boldsymbol{c}) \odot \boldsymbol{\rho}||_2^2 + 2\big(\boldsymbol{c} \odot (\boldsymbol{\rho} - \boldsymbol{m})\big)^T \big((1 - \boldsymbol{c}) \odot \boldsymbol{\rho}\big) \\ &= ||\boldsymbol{c} \odot (\boldsymbol{\rho} - \boldsymbol{m})||_2^2 + ||(1 - \boldsymbol{c}) \odot \boldsymbol{\rho}||_2^2,\end{aligned}$$

where the 3rd term in the 3rd row is equal to exact zero due to the complementary property between $1 - \boldsymbol{c}$ and $\boldsymbol{c}$.

The Difference of Smoothness Losses. According to the above derivations, the explicit relation between hard and soft smoothness losses can be obtained:

$$\mathcal{L}_{smooth}^{(soft)}(\boldsymbol{\rho}, \boldsymbol{m}, \boldsymbol{c}) = \mathcal{L}_{smooth}^{(hard)}(\boldsymbol{\rho}, \boldsymbol{m}, \boldsymbol{c}) + ||(1 - \boldsymbol{c}) \odot \boldsymbol{\rho}||_2^2.$$

It can be seen that the soft smoothness loss is a more generalized loss term including the hard smoothness loss.

As a result, the soft smoothness loss can learn the structural sparsity not only within the group mask $\boldsymbol{c}$ via the first loss term (same as hard smoothness loss) but also beyond the group mask $\boldsymbol{c}$ via the second loss term. Besides, in the second loss term, only the ℓ_2-norm based penalty is enforced on the elements in $\boldsymbol{\rho}$ beyond the group mask $\boldsymbol{c}$ (*i.e.*, $1 - \boldsymbol{c}$), which is also known as weight decay that can shrinkage the large values back to our predefined ranges.

C. Additional Experiments

C.1. Discussion of Convergence Issue

To achieve our expected sparse results, the corresponding values of λ_1 and λ_2 result in a very large overall loss $\mathcal{L}$ that does not decrease effectively along with the iteration. To minimize the overall loss, the optimization focuses on reducing the number

of perturbation position as much as possible due to imbalanced losses. Finally, the perturbation position is not generated properly, leading to the training failure as shown in Fig. 1.

To resolve this issue, we introduce multi-stage optimization algorithm to train the sparse attack model, which can decrease the overall loss effectively by combining stage I training with stage II training.

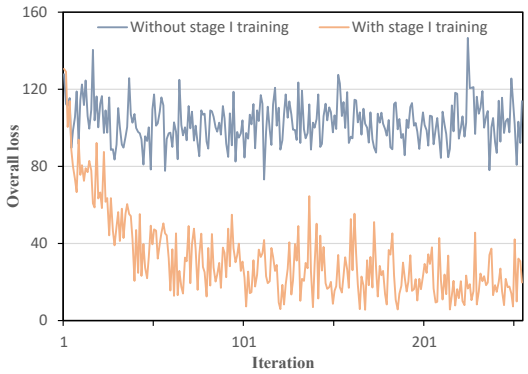


Figure 1. Comparison of the overall loss of EGS-TSSA (Ours) via different training strategies. To be minimized directly, the overall loss does not decrease through the training iteration. By using the multi-stage optimization algorithm, the overall loss starts to decrease effectively and the sparse attack model can be trained properly.

C.2. Discussion of Smoothness Loss

In this section, we show adversarial examples crafted by EGS-TSSA$_{hard}$ and EGS-TSSA$_{soft}$, shown in Fig. 2. As can be seen from the visualization results, the perturbations generated by the soft smoothness loss are more spread out (beyond the constraint of group mask) than the hard smoothness loss (within the constraint of group mask).

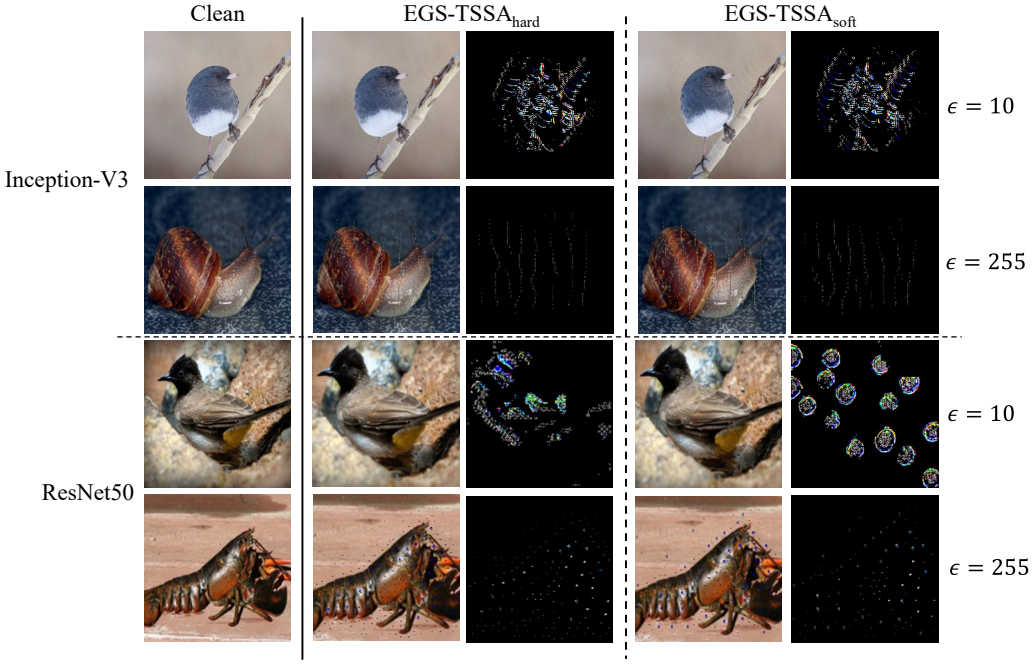
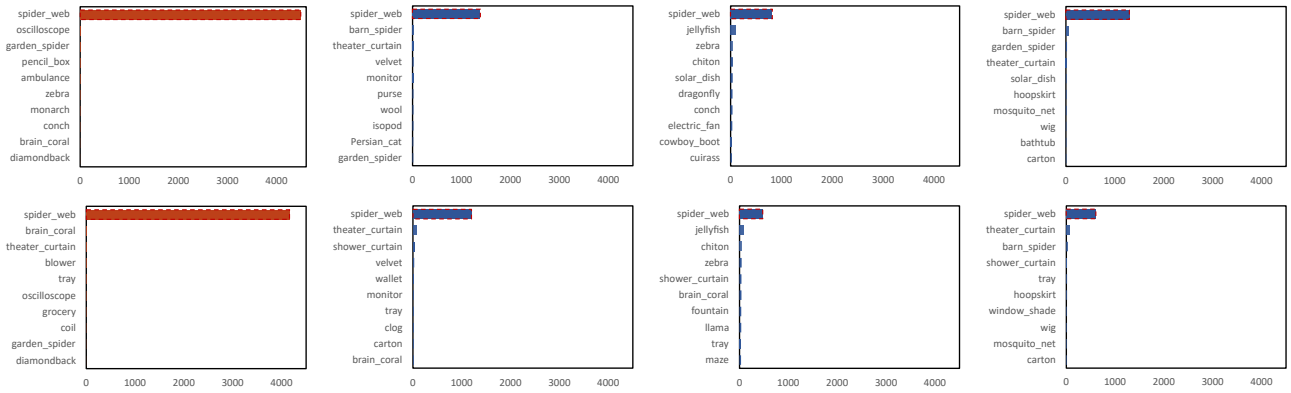


Figure 2. Comparison of adversarial examples crafted by EGS-TSSA (Ours) under hard and soft smoothness loss constraints. Surrogate models are Inception-V3 (top) and ResNet50 (bottom) respectively. When constrained by the hard smoothness loss, EGS-TSSA$_{hard}$ learns the structural sparse perturbation more concentrated around the group mask. Contrarily, constrained by the soft smoothness loss, EGS-TSSA$_{soft}$ learns the structural sparse perturbation with more freedom beyond the group mask. Furthermore, the sparse perturbation generated by EGS-TSSA$_{soft}$ has more structured and repetitive patterns than EGS-TSSA$_{hard}$.

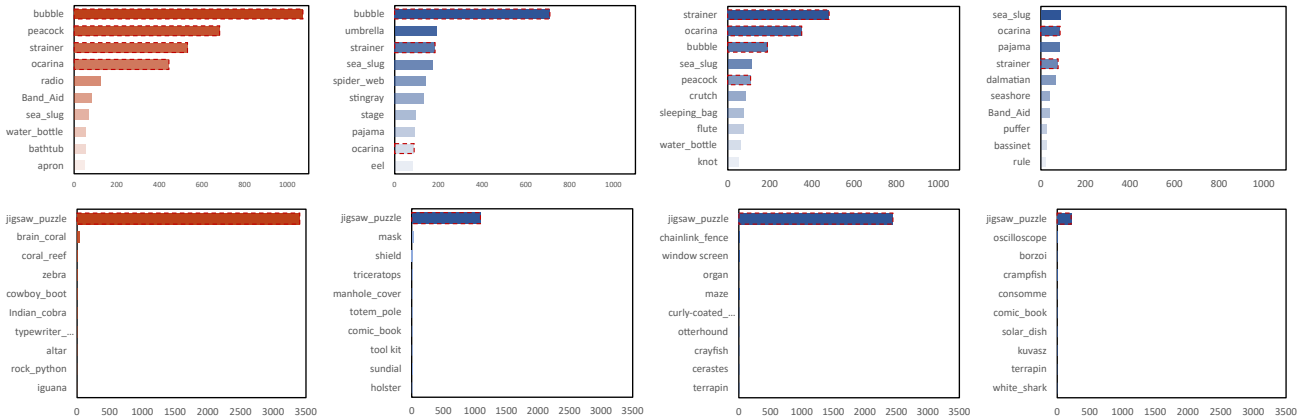
C.3. Analysis of the Label Distribution of Structural Sparse Perturbation

The classification results of all the adversarial examples are analyzed for different threat models. For surrogate models where the generated perturbation positions are located in classification-relevant regions, such as Inception-v3, our approach does not change this perturbation pattern. In contrast to TSAA, our EGS-TSSA achieves better top-1 label domination in Fig. 3. For ResNet50 where the generated perturbation positions are located near the image boundary, our method changed the perturbation pattern completely to concentrate more on classification-relevant regions. It is shown in Fig. 4 that our adversarial examples do not attack threat models towards a single label, but towards a variety of labels. Thus, our EGS-TSSA method is more inclined to destroy classification-relevant features in the image.



(a) Inception-V3 (b) VGG16 (c) ResNet50 (d) DenseNet161

Figure 3. The label distribution of adversarial examples crafted by TSAA and our EGS-TSSA (the surrogate model: Inception-V3) under the constraint of $\ell_\infty = 10$, where the x-axis represents the number of adversarial examples that are classified to the corresponding label and the y-axis represents the classification labels (only top 10 labels are shown for the comparison). Red bars indicate white-box attacks and blue bars indicate black-box attacks. In contrast to TSAA (2nd row), our EGS-TSSA method (1st row) achieves better top-1 label domination from surrogate model to target models due to the proposed constraint of structural sparsity.

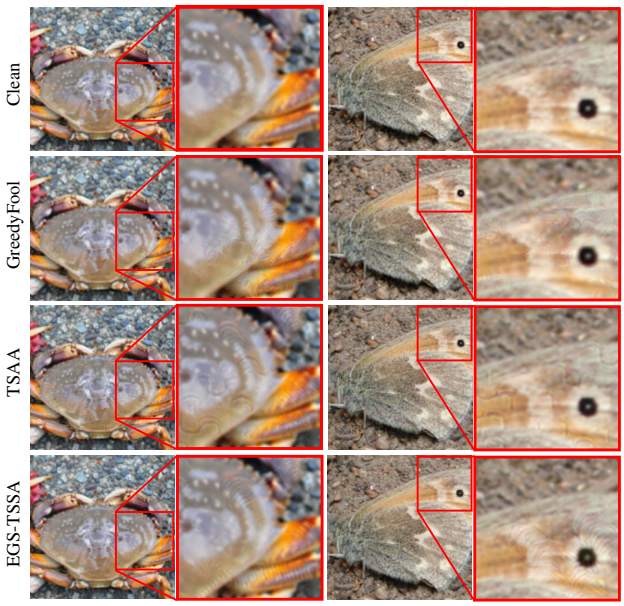


(a) ResNet50 (b) VGG16 (c) DenseNet161 (d) Inception-V3

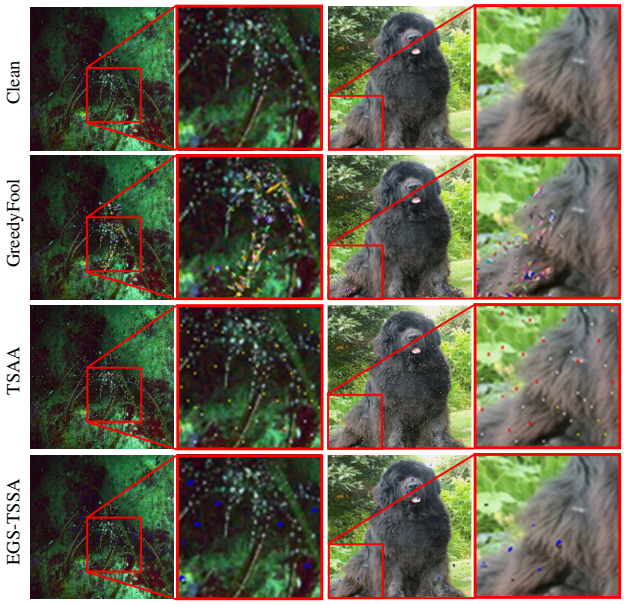
Figure 4. The label distribution of adversarial examples crafted by TSAA and our EGS-TSSA (the surrogate model: ResNet50) under the constraint of $\ell_\infty = 10$, where the x-axis represents the number of adversarial examples that are classified to the corresponding label and the y-axis represents the classification labels (only top 10 labels are shown for the comparison). Red bars indicate white-box attacks and blue bars indicate black-box attacks. In contrast to TSAA (2nd row), our EGS-TSSA method (1st row) fails to achieve the top-1 label domination from surrogate model to target models since the sparse perturbation pattern of ResNet50 is completely changed. However, the number of adversarial examples crafted by EGS-TSSA in other labels becomes larger than TSAA. As a result, the sparse pattern of our EGS-TSSA method that concentrates on classification-relevant regions has a higher transferability (see Table 1).

C.4. Comparison of Adversarial Examples on Image Classification

In Fig. 5, perturbations crafted by our EGS-TSSA method can blend more naturally into classification-relevant regions, thus making adversarial examples undetectable.



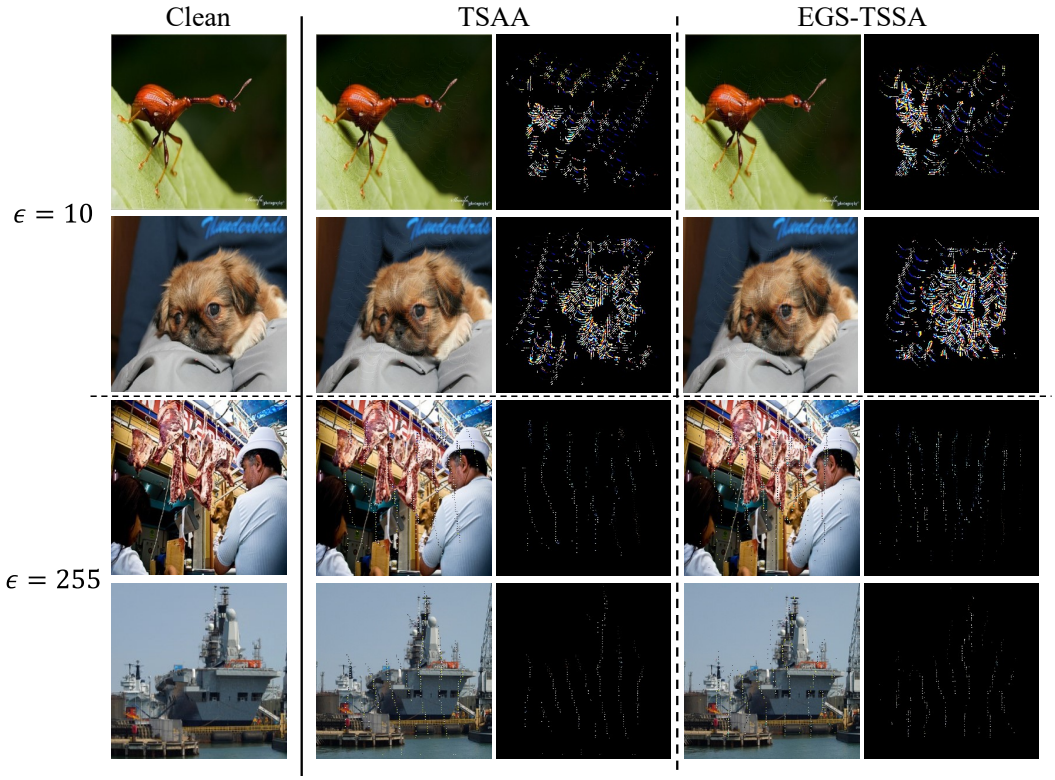
(a) $\epsilon = 10$



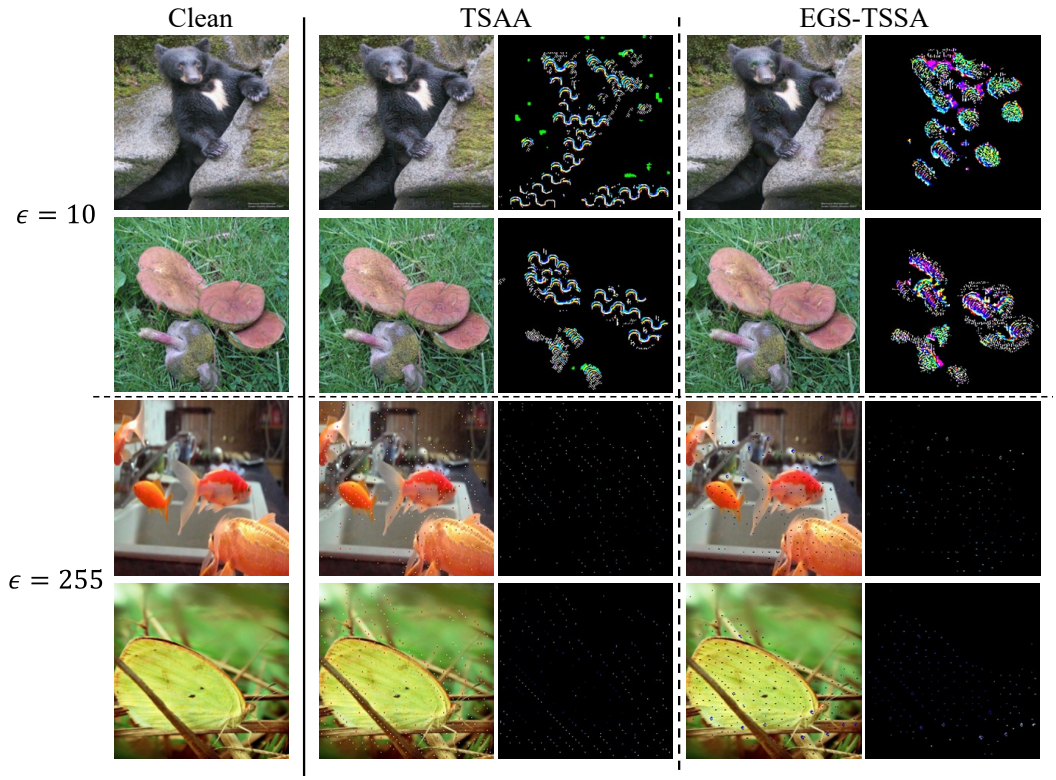
(b) $\epsilon = 255$

Figure 5. Visualization of adversarial examples crafted by GreedyFool, TSAA and our EGS-TSSA (the surrogate model: ResNet50) under the constraints of $\ell_\infty = 10$ and $\ell_\infty = 255$. As can be seen, the perturbations of our EGS-TSSA method are more structural sparse and more imperceptible.

We further investigate adversarial examples generated by TSAA and EGS-TSSA (Ours) using different surrogate models under the constraints of $\ell_\infty = 10$ and $\ell_\infty = 255$, respectively. In Fig. 6, TSAA fails to attack all the benign examples, but our EGS-TSSA attacks them all successfully. Thus it can be seen that the perturbations generated by our proposed EGS-TSSA method are more structured and more concentrated in the classification-relevant important regions.



(a) Inception-V3



(b) ResNet50

Figure 6. Comparison of adversarial examples crafted by TSAA and EGS-TSSA (Ours) under the constraints of $\ell_\infty = 10$ and $\ell_\infty = 255$. Surrogate models are Inception-V3 (a) and ResNet50 (b) respectively. Due to the constraint of structural sparsity, all the above results are attacked successfully by our EGS-TSSA method but the TSAA method fails.

C.5. Comparison of Adversarial Examples on Object Detection

In Fig. 7, we evaluate the performance of the proposed EGS-TSSA method on the object detection task. Compared to TSAA, our EGS-TSSA causes the object detection model to falsely ignore more correct objects and detect more incorrect objects, which demonstrates that our approach leads to a more severe destruction on object detection.

Figure 7. Comparison of adversarial examples crafted by TSAA and EGS-TSSA (Ours) on the object detection task. As compared to original detected results (see w/o attack) by the SSD300 model, our EGS-TSSA obtains lower AP and AR than TSAA.

C.6. Comparison of Adversarial Examples on Semantic Segmentation

In Fig. 8, we further verify the effectiveness of the proposed EGS-TSSA method on the semantic segmentation task. Adversarial examples that are crafted by our EGS-TSSA method can severely fool the prediction of correct semantic segmentation labels as compared to TSAA, *e.g.*, some foreground labels are falsely predicted as the background.

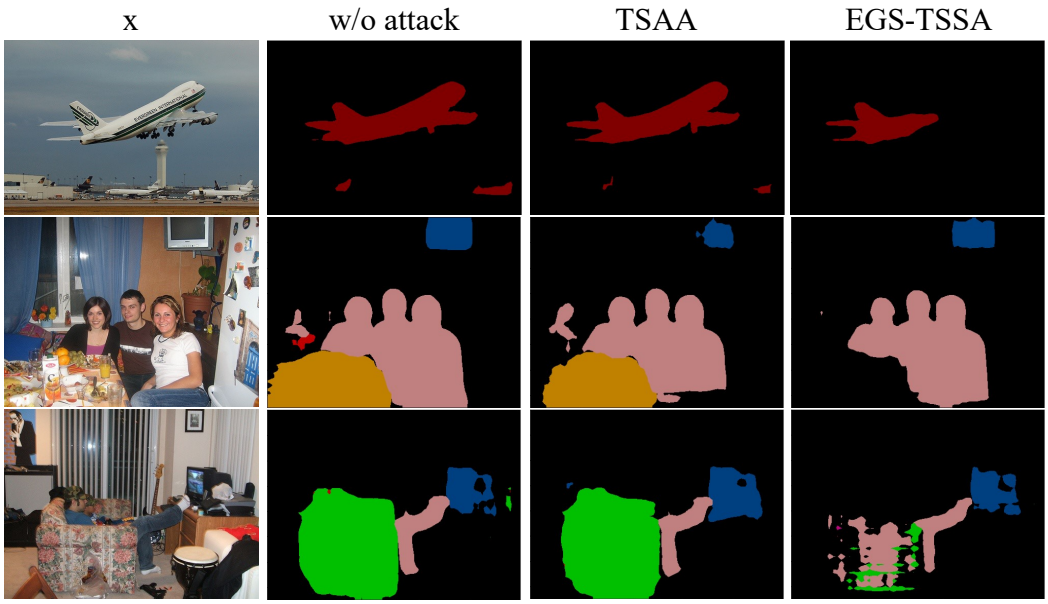


Figure 8. Comparison of adversarial examples crafted by TSAA and EGS-TSSA (Ours) on the semantic segmentation task. As compared to original segmented results (see w/o attack) by the FCN model, our EGS-TSSA obtains the lower mIoU than TSAA.

C.7. Visualization of the Pattern of Structural Sparse Perturbation

In Fig. 9, we visualize the pattern of sparse perturbations. Due to the constraint of ℓ_1-norm alone, TSAA generates the sparse perturbation globally. Clearly, our EGS-TSSA generates a more structured and sparse perturbation from the perspective of coarse-grained and fine-grained levels.

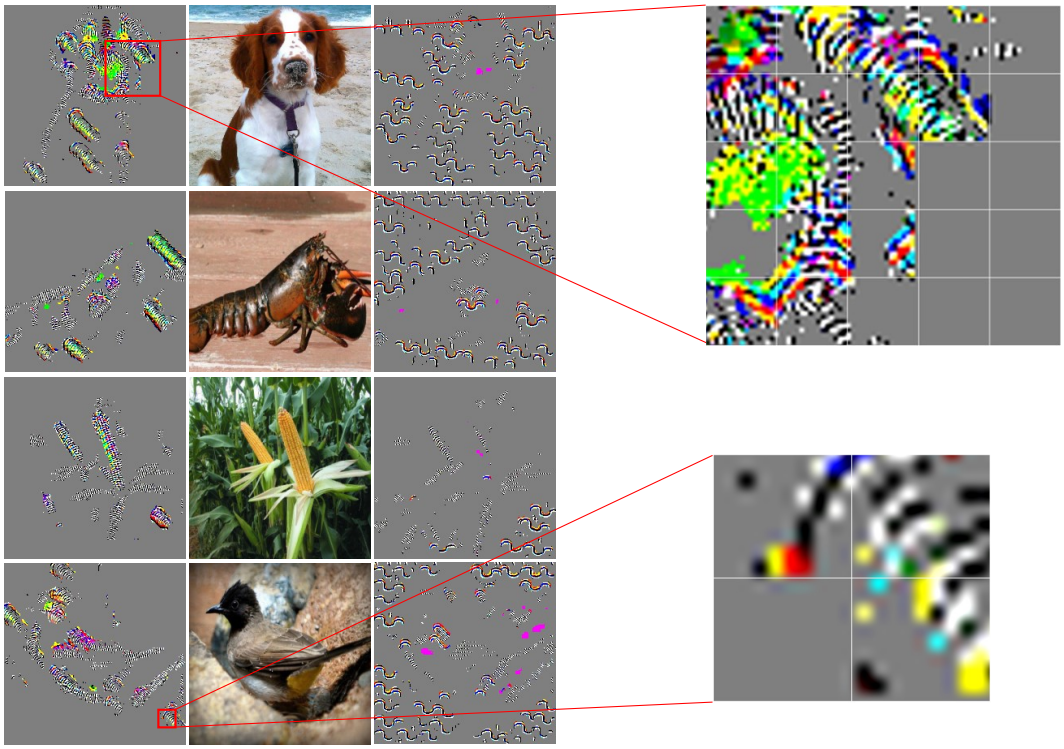


Figure 9. Visualization of the pattern of structural sparse perturbations crafted by TSAA and EGS-TSSA (Ours). As can be seen from the figure, our EGS-TSSA (1st column) learns not only the coarse-grained sparsity at group level but also the find-grained sparsity at pixel level, thus generating more structured and sparse perturbations than TSAA (3rd column).

C.8. Analysis of the Distribution of Structural Sparse Perturbation

In Fig. 10, we further analyze the distribution of sparse perturbations by ranking the importance of all subgroups of $\boldsymbol{GFI}$ and dividing them sequentially into 10 regions, each representing 10% of the entire $\boldsymbol{GFI}$. It is obvious that the crafted perturbation is highly overlapped with classification-relevant important regions.

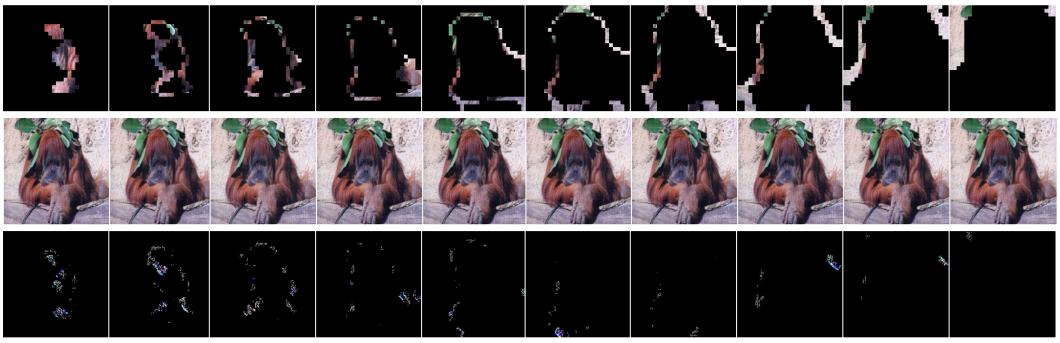


Figure 10. Visualization of the distribution of structural sparse perturbations crafted by EGS-TSSA (Ours). The 1st row is the 10 regions sequentially selected by $\boldsymbol{GFI}$, the 2nd row is the generation of adversarial examples using each above region, and the 3rd row is the corresponding visualization of the crafted perturbation.