

Enhancing targeted transferability via feature space fine-tuning: supplementary material

Hui Zeng, Biwei Chen, and Anjie Peng

The supplementary document consists of four parts of content: A) Ablation study on N_{ft} and k ; B) Visual comparison; C) Date-free targeted Universal adversarial perturbation (UAP); D) Alternative methods for calculating aggregate gradient.

A Ablation study

1) **Influence of the iteration number N_{ft} of fine-tuning.** We study the influence of N_{ft} on the transfer success rate in the single-model, random-target transfer scenario, with the source model fixed as Res50. The optimal N_{ft} values vary from 10 to 15 when the baseline attack is CE (Fig. 1(a)) and from 5 to 10 when the fine-tuning is based on Logit (Fig. 1(b)). This can be explained as follows. A relatively weak attack, e.g., CE, has greater potential for improvement and thus needs more iterations of fine-tuning. In contrast, a relatively strong attack, Logit or model-ensemble, is more suitable for less fine-tuning. In our study, we set $N_{ft} = 10$ for all attacks and in all scenarios for simplicity. It is observed from Fig. 1 that the choice of $N_{ft} = 10$ is almost always dominant $N_{ft} = 0$ that represents no fine-tuning.

2) **Influence of the fine-tuning layer k .** Next, we study the influence of target layer k in fine-tuning on the transfer success rate. In this experiment, we fix the other parameters of the proposed method and select a few internal layers for each source model. Fig. 2(a), (b), and (c) report the transferability of adversarial examples crafted on source models Res50,

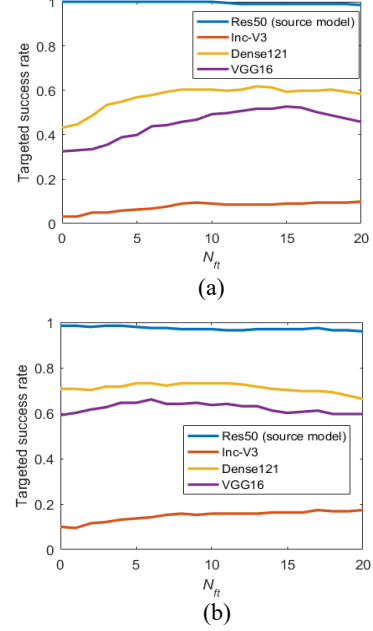


Fig. 1. Effect of N_{ft} on AEs' transferability. The source model is Res50. The baseline attack is CE (a) and Logit (b).

Dense121, and Inc-V3, respectively. The main takeaway is that fine-tuning on a middle layer is helpful to transferability. This finding is consistent with previous works that early layers are usually data-specific, whereas later ones are model-specific. Based on the above considerations, we select to attack *Mixed 6b* for Inc-v3, the last layer of the third block for Res50 and Dense121, and *Conv4_3* for VGG16 in this study.

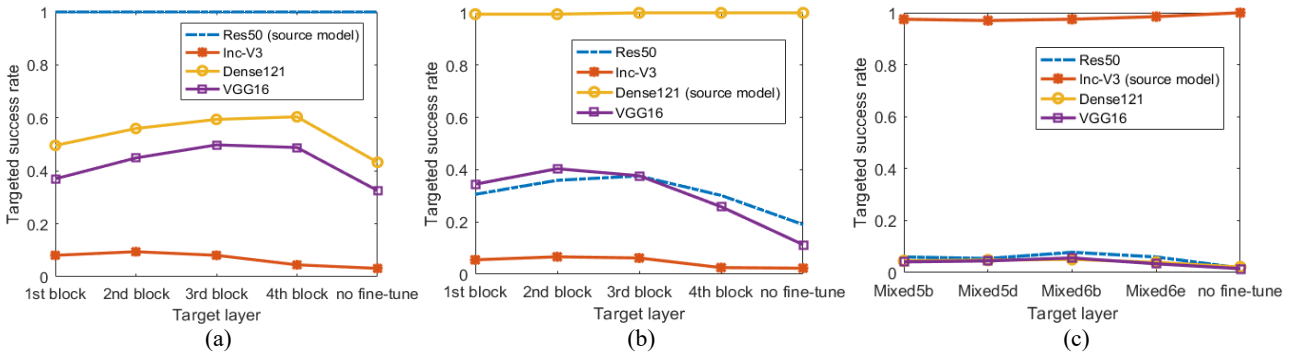


Fig. 2. Effect of target layer on AEs' transferability. The baseline attack is CE. The source models are Res50 (a), Dense121 (b), and Inc-V3 (c), respectively.

B Visual comparison

Besides the example in the paper, we provide additional examples in this file. Fig. 3 shows AEs targeted to ‘grey owl,’ and Fig. 4 shows AEs targeted to ‘hippopotamus.’ While the

perturbation introduced by the iterative methods resembles noise, that introduced by TTP is more semantically-aligned.

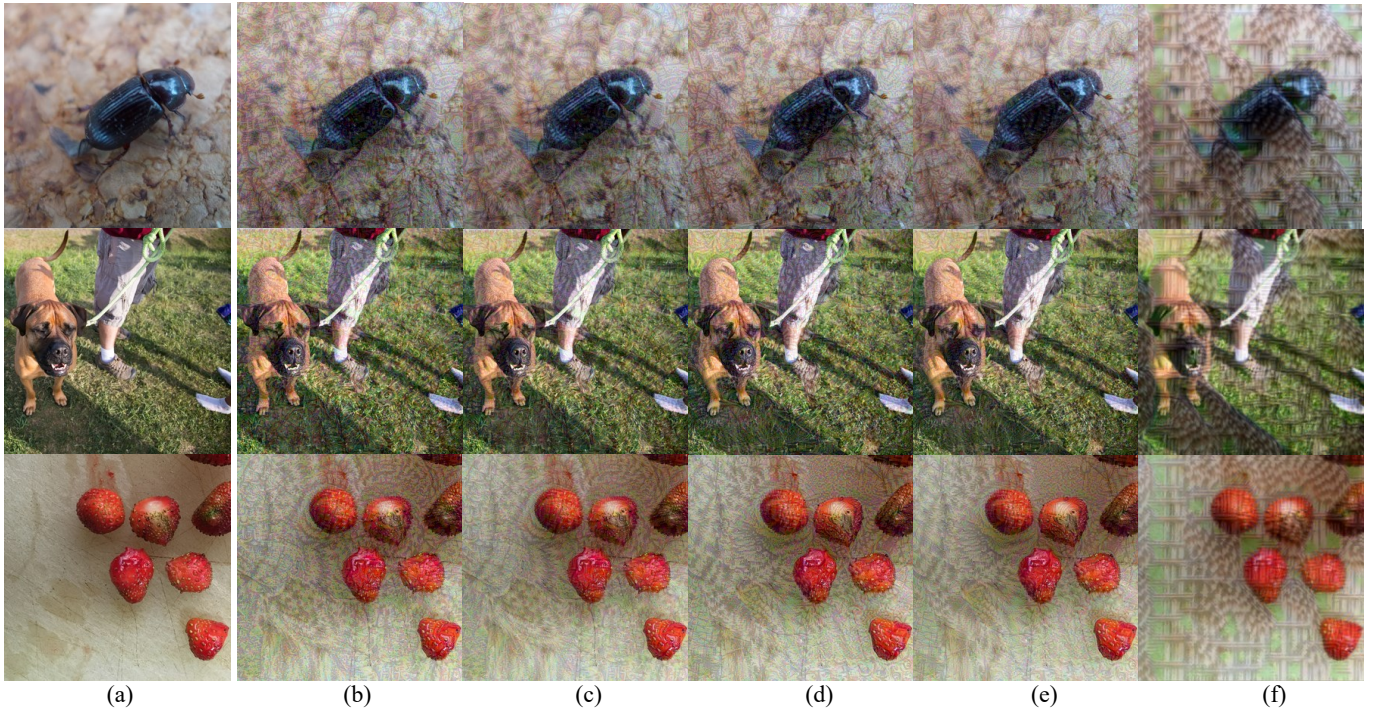


Fig. 3. The visual comparison of the AEs generated by different methods, $\epsilon = 16$. The target class is ‘grey owl’. (a) Original image, (b) CE, (c) CE+ft (proposed), (d) Po+Trip, (e) Po+Trip+ft (proposed), (f) TTP.

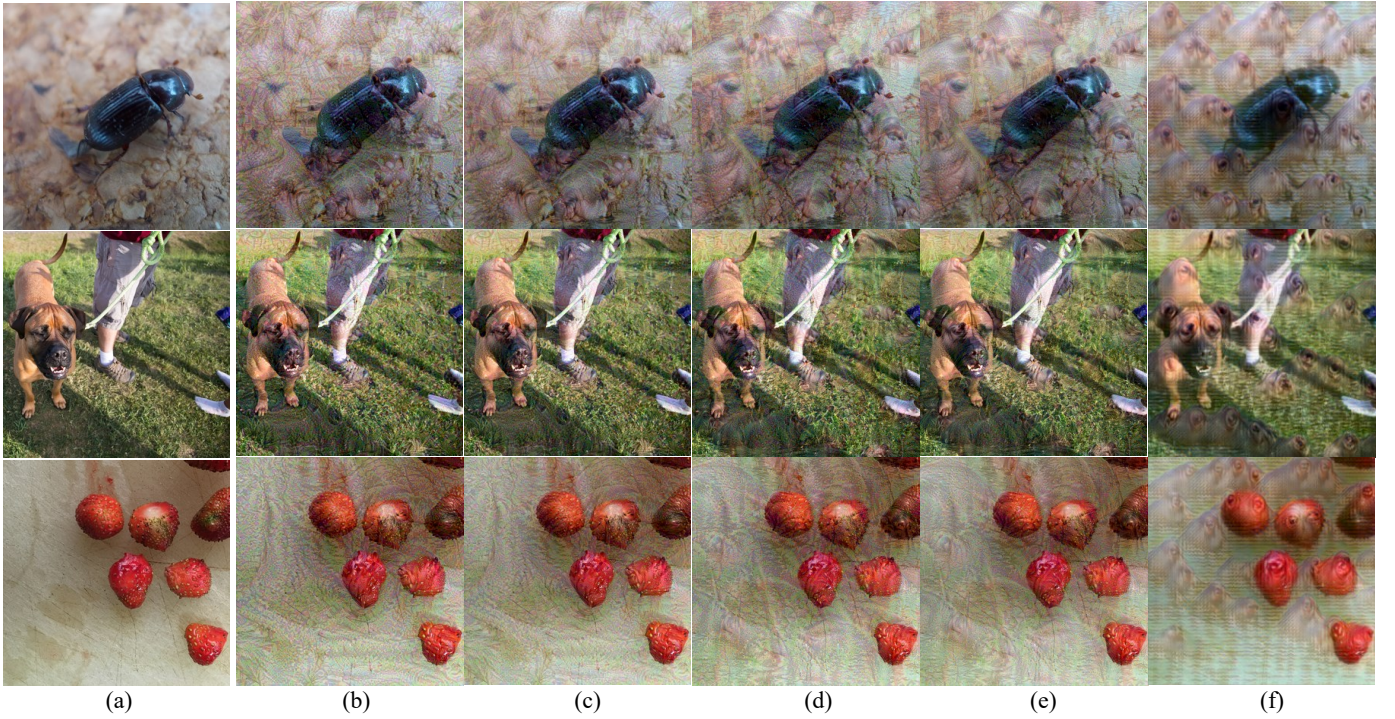


Fig. 4. The visual comparison of the AEs generated by different methods, $\epsilon = 16$. The target class is ‘hippopotamus’. (a) Original image, (b) Logit, (c) Logit+ft (proposed), (d) SupHigh, (e) SupHigh+ft (proposed), (f) TTP.

C Date-free targeted UAP

Targeted UAP is a particular type of perturbation that can drive multiple clean images into a given class y_t . Among the methods for crafting targeted UAP, we are particularly interested in the data-free approach, which does not require additional training data. Precisely, we use a mean image (all entrances of which equal 0.5) as the start point and mount a targeted attack for 200 iterations to obtain a targeted UAP ($\epsilon = 16$) with different simple iterative methods. Then, the obtained UAP is applied to all 1000 images in our dataset.

Table 1 reports the success rates averaged over 100 classes ($y_t = 0:99$). It is observed that feature space fine-tuning consistently improves the baseline attacks. Combining the results of the paper, we can conclude that the proposed fine-tuning scheme improves AEs’ transferability not only across models but also across input images. Fig. 5 presents examples of targeted UAPs generated with different methods. It is observed that the UAPs are less noisy after feature space fine-tuning.

D Alternative aggregate gradients

This subsection investigates the effect of the method of calculating aggregate gradients on the proposed fine-tuning scheme. Fig. 6 compares the transferability of AEs (under the random-target and most difficult-target scenarios, $\epsilon = 16$) when the aggregate gradient is generated with FIA [1] and RPA [2]. Unlike FIA, which adopts a pixel-wise mask, RPA adopts a patch-wise mask in calculating aggregate gradients. For a fair comparison, we set the ensemble number $N=30$ for both FIA and RPA. Our results show that the more advanced RPA indeed improves the transferability slightly in most cases. This result

Table 1. Success rates (%) of the data-free UAPs with $\epsilon = 16$. Without/with fine-tuning. Dominant results are in **bold**.

| Attack | Res50 | Dense121 | VGG16 | Inc-v3 |
|--------|-------------------|-------------------|-------------------|-----------------|
| CE | 8.1/ 15.1 | 8.0/ 13.1 | 19.2/ 34.6 | 1.9/ 2.4 |
| Logit | 20.7/ 24.6 | 17.5/ 18.8 | 64.9/ 66.3 | 3.6/ 4.7 |

indicates the proposed method can be further improved by incorporating more advanced aggregate gradient methods. In the paper, we use FIA to generate the aggregate gradient for simplicity.

[1] Z. Wang, H. Guo, Z. Zhang, et. al., “Feature importance-aware transferable adversarial attacks,” ICCV2021, pp. 7639–7648.

[2] Y. Zhang, Y. Tan, T. Chen, et. al., “Enhancing the transferability of adversarial examples with random patch,” IJCAI2022, pp. 1672–1678..

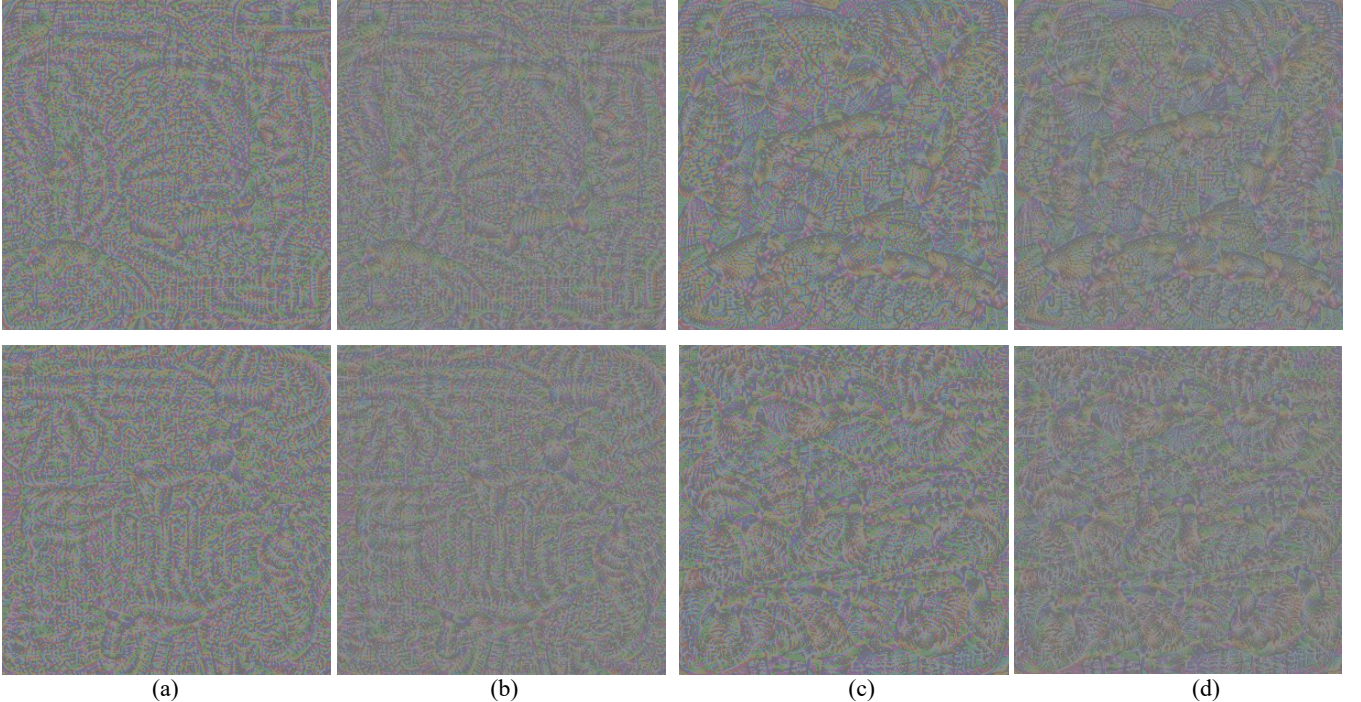


Fig. 5. Data-free targeted UAPs ($\epsilon = 16$, VGG16) generated by different methods. The target class is ‘tench’ for the first row, and ‘goose’ for the second row. (a) CE, (b) CE+ft, (c) Logit, (d) Logit+ft (proposed).

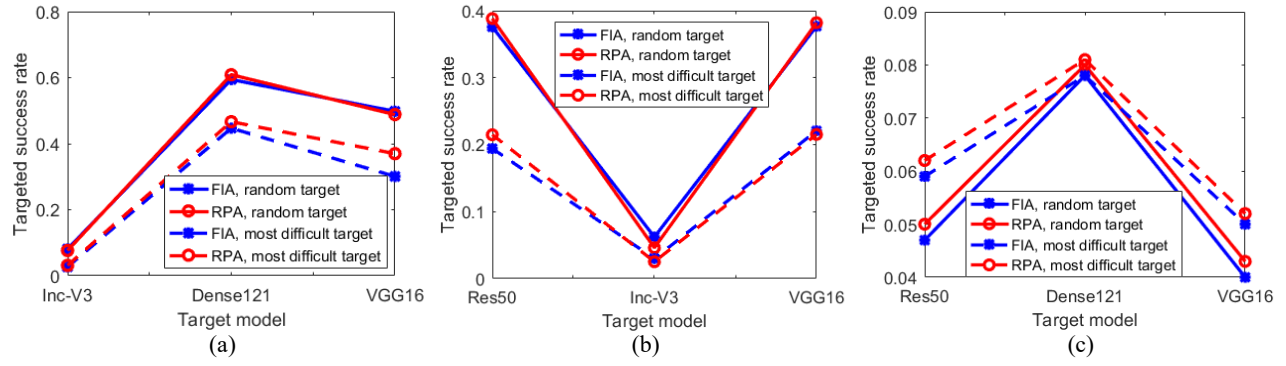


Fig. 6. Comparison AEs' transferability when the aggregate gradients are generated with FIA and RPA. The baseline attack is CE. The source models are Res50 (a), Dense121 (b), and Inc-V3 (c), respectively.