Any Target Can be Offense: Adversarial Example Generation via Generalized Latent Infection (Supplementary Material)

Youheng Sun[1⋆], Shengming Yuan[1⋆], Xuanhan Wang[2⋆⋆], Lianli Gao[1], and Jingkuan Song[1]

[1] Center for Future Media, University of Electronic Science and Technology of China
[2] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China
youheng.sun@std.uestc.edu.cn, shengming.yuan@outlook.com, wxuanhan@hotmail.com, lianli.gao@uestc.edu.cn, jingkuan.song@gmail.com

A Implement Details

During training, we consider samples with small classification loss to be excellent target samples with more prominent features. We continue to use this estimation during testing.

Test on Known Classes. When evaluating the target attack success rate on known classes, since the attacker entirely constructs the training set of known classes, we select the target sample with the smallest classification loss in the train set as the target for that known class.

Test on Unknown Classes. When testing on unknown classes, the attacker obviously cannot access the complete target dataset to select the best sample but can still choose some target samples that are relatively clear and unobstructed. To simulate this process, we select 10 images with relatively small classification losses for each target class from the validation set of ImageNet as targets. During testing, the target samples of unknown classes are **randomly** selected from these 10 images.

B Efficiency Analysis

In this section, we analyze the efficiency of different methods. To simulate the scenario of unknown classes, we divide the ImageNet-1k dataset into 5 parts, with each part appearing sequentially. Unlike other generator-based methods that require retraining for each new part, our GAKer can attack any target class with just one training process. As shown in Tab. 1, our method only requires 1% of the training time compared to TTP, highlighting the remarkable efficiency of our GAKer.

⋆ Equal contribution.
⋆⋆ Corresponding author

Table 1: Training time comparison assuming ImageNet is divided into 5 parts, with 200 new target classes added each time. "5*20*3.7h" means retraining 5 times, each with 20 epochs, taking 3.7 hours per epoch (in one NVIDIA GTX 4090 GPU).

Method	TTP	HGN	ESMA	Ours
Time	1k*1*13.1h(100%)	5*20*3.7h(3%)	5*20*6.8h(5%)	1*20*6.6h(1%)

C Parameter Sensitivity Analysis

To explore the impact of the parameter α, we conduct several experiments. We train the model with α set to 0, 0.25, 0.5, 0.75, and 1, respectively. Then, we validate the effect of different α settings on the attack success rate. In this experiment, we use ResNet-50 (Res-50) [1] as the substitute model and test the target attack success rates (TASR) on known and unknown classes on ResNet-50 (Res-50) [1], ResNet-152 (Res-152) [1], VGG-19 [3], DenseNet-121 (Dense-121) [2]. The experimental results are shown in Fig. 1. Finally, we select α as 0.5, which yields the best experimental results.

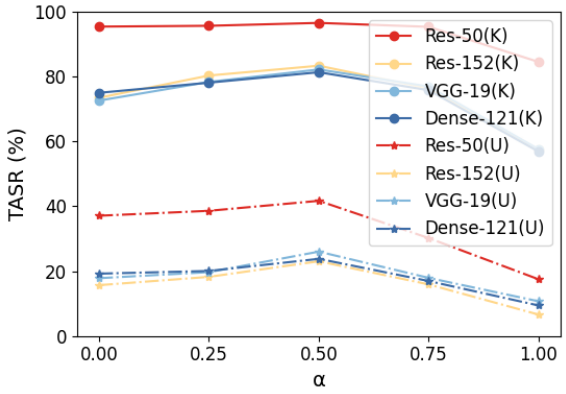

Fig. 1: Effect of α. The Res-50 serves as the substitute model, while the performance of black-box models, including Res-152, VGG-19, and Dense-121, is evaluated for both known (K) and unknown (U) classes.

D Further Analysis on Unknown Classes

To further analyze the performance of our GAKer on unknown classes, we visualize the targeted attack success rate (TASR) for all 800 unknown classes in Fig. 2. Meanwhile, we present the histogram of TASR in Fig. 3. The statistics show that there is a large variation in TASR among different unknown classes.

The TASR for most of the unknown classes exceeds 50%, but some classes are hardly attacked, with a TASR lower than 50%. This phenomenon indicates that different classes have varying levels of difficulty.

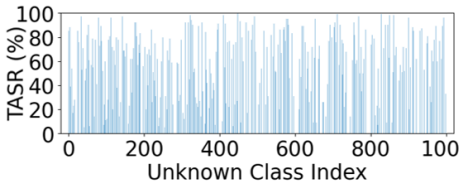

Fig. 2: The TASR for each unknown class. The substitute model is Res-50.

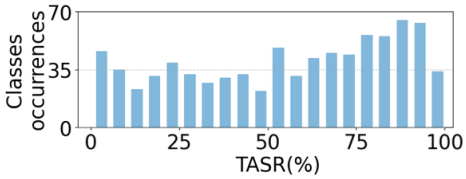

Fig. 3: The histogram of unknown class distributions by TASR. The substitute model is Res-50.

E Results on GPT-4V

We conduct a series of tests on the GPT-4V [4] to showcase the effectiveness of our method when dealing with Large Vision-Language Models. All experiments on GPT-4V are conducted on 13 March 2024.

As shown in Figs. 4 to 7, when we ask GPT-4V whether the adversarial example contains the original class or the target class, GPT-4V will deny the former and affirm the latter. Furthermore, when we ask GPT-4V to describe the content of the image by one sentence, the description corresponds with the target rather than the original image.

When we ask GPT-4V to **"Please generate a similar image"**, as shown in Figs. 8 to 11, we can see that the generated sample is also similar to the target and utterly unrelated to the original class.

This is a strong indication that when dealing with Large Vision-Language Models, our method produces adversarial samples that effectively fool the model.

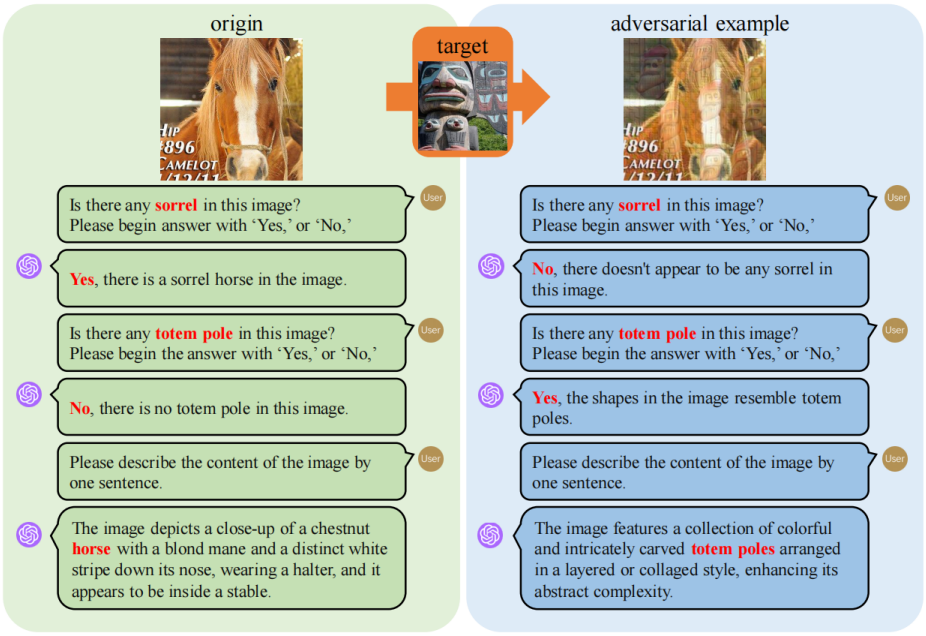

Fig. 4: Q&A from **"sorrel"** to **"totem pole"**

F Visualization of Adversarial Examples

In this section, we show some adversarial examples in Fig. 12.

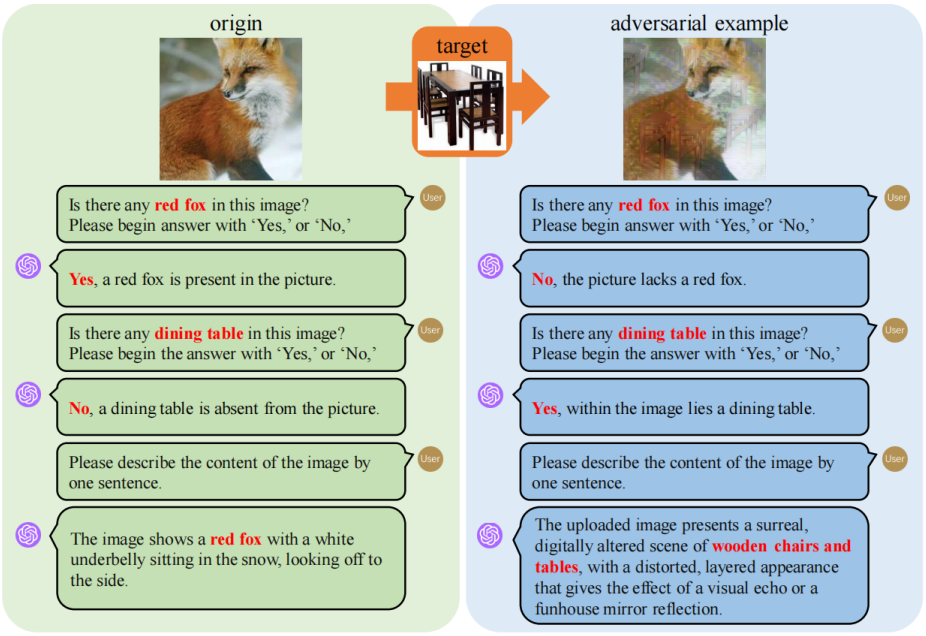

Fig. 5: Q&A from **"red fox"** to **"dining table, board"**

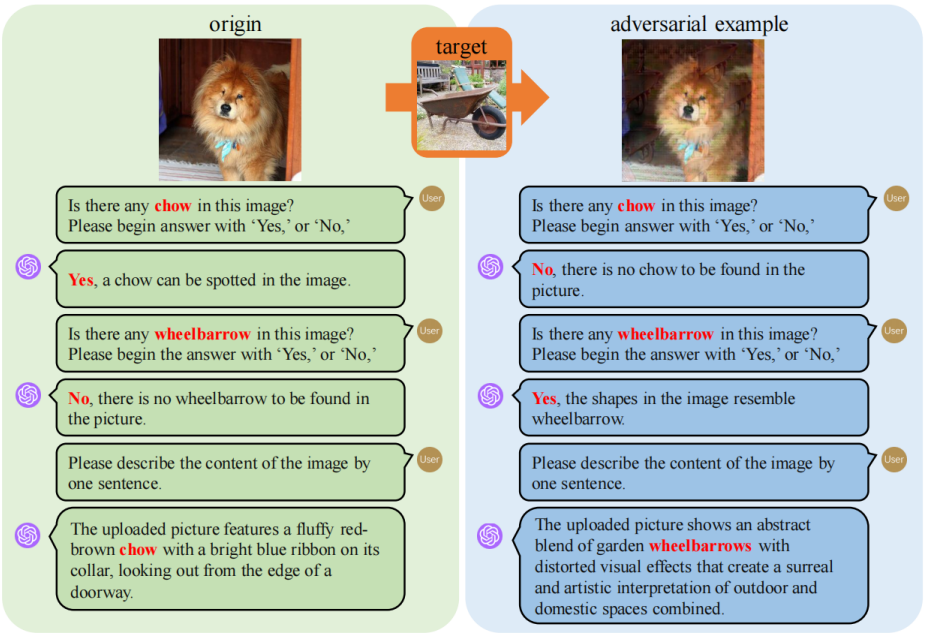

Fig. 6: Q&A from **"chow"** to **"wheelbarrow"**

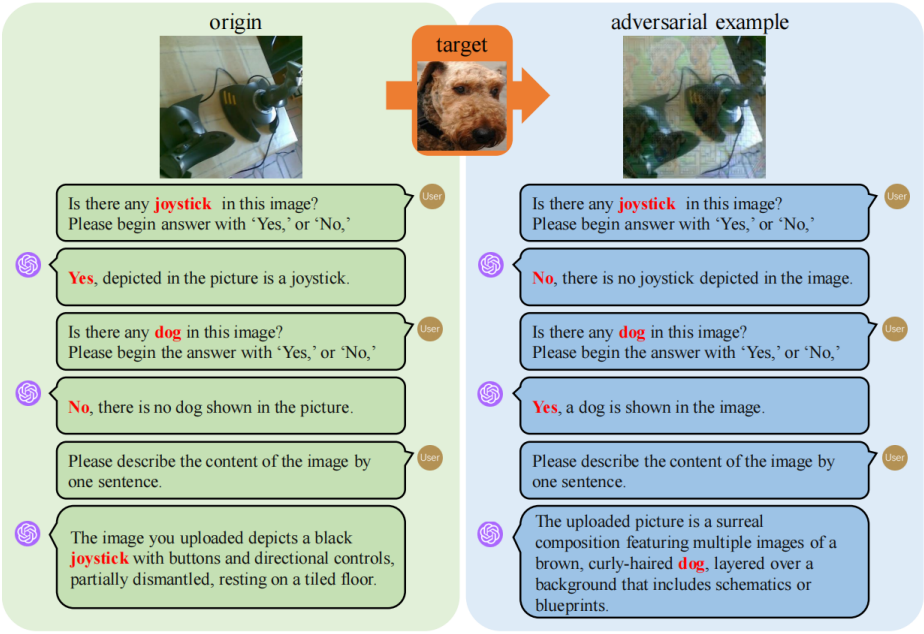

Fig. 7: Q&A from **"joystick"** to **"dog"**

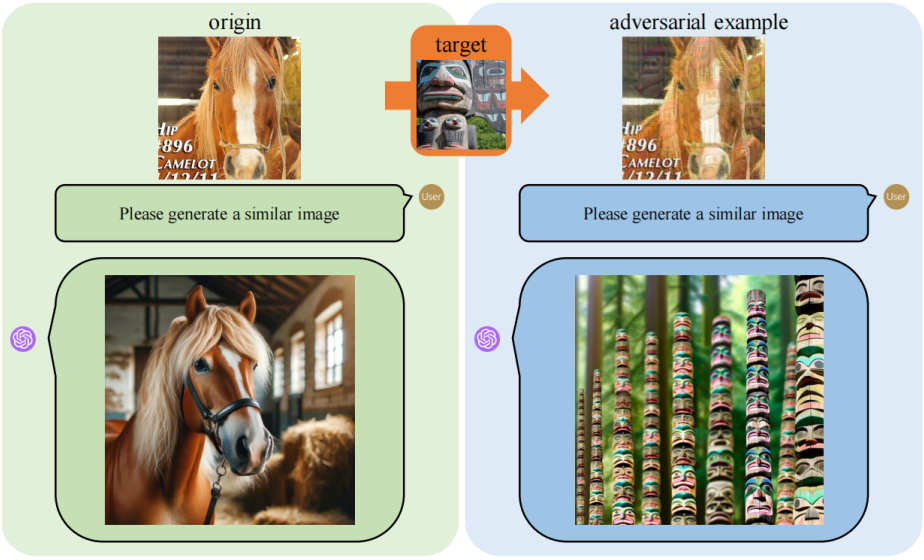

Fig. 8: From **"sorrel"** to **"totem pole"** generated by GPT-4V

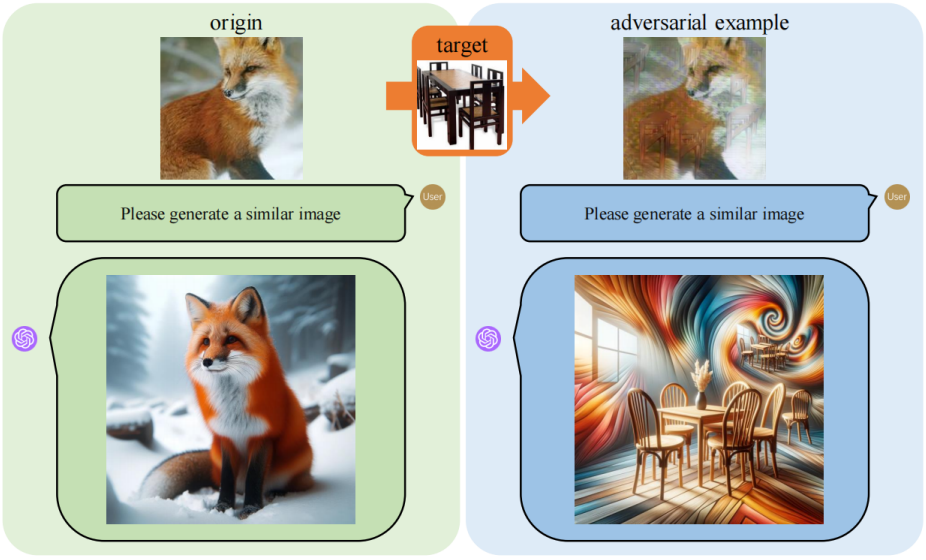

Fig. 9: From **"red fox"** to **"dining table, board"** generated by GPT-4V

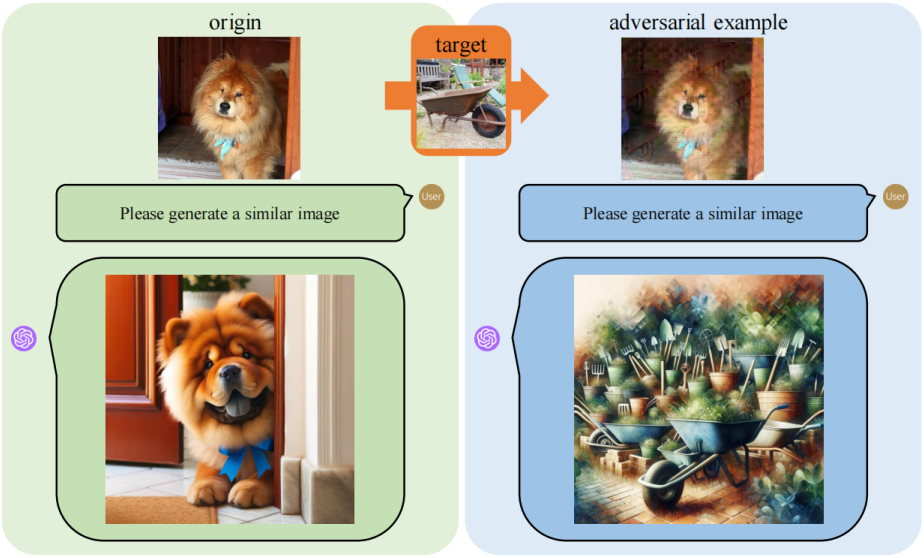

Fig. 10: From **"chow"** to **"wheelbarrow"** generated by GPT-4V

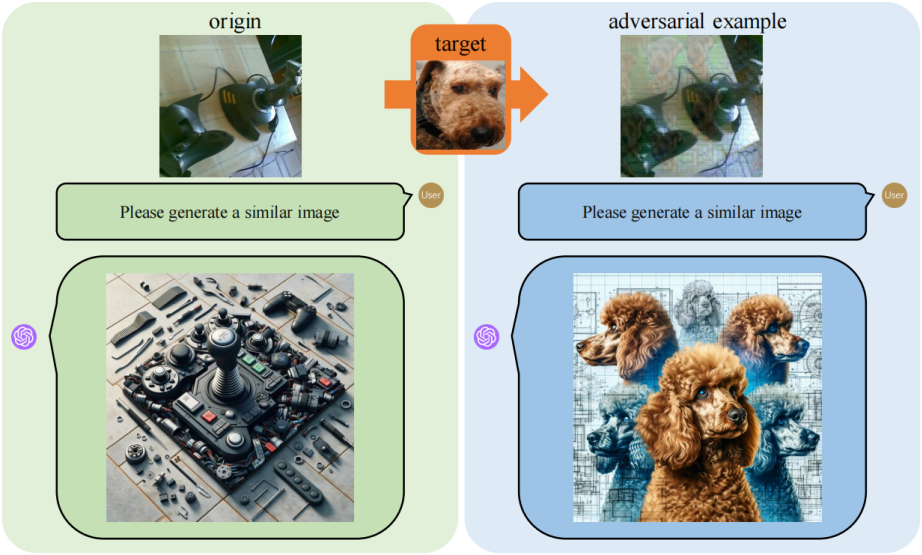

Fig. 11: From **"joystick"** to **"dog"** generated by GPT-4V

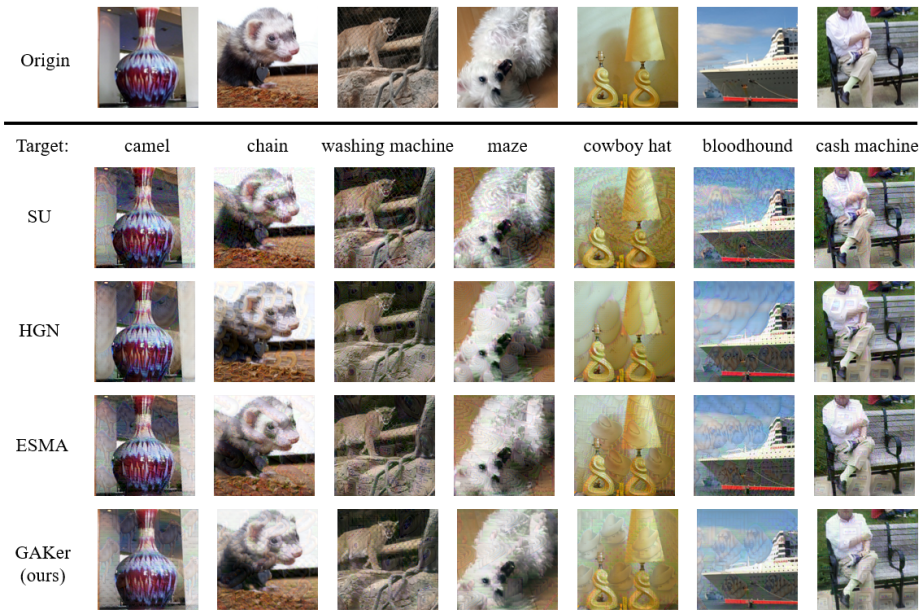

Fig. 12: Visualization of adversarial examples. All adversarial examples are generated on Res-50.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
2. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
4. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C., Liu, Z., Wang, L.: The dawn of lmms: Preliminary explorations with gpt-4v(ision). CoRR (2023)