SAM Encoder Breach by Adversarial Simplicial Complex Triggers Downstream Model Failures

Supplementary Material

Algorithm 1 Domain re-adaption Simplex Initialization Algorithm

Input: SAM image encoder $f_{im}(\cdot)$, A random subset of the SA-1B training data $\mathbf{tar}$, clean task-specific image x_τ,ℓ_1 Loss function $\mathcal{L}(\cdot)$ with domain re-adaption, number of mode's optimization iteration t,perturbation budget ϵ, decay factor μ, number of simplexes N,random patch augmentation $\mathcal{PT}(\cdot)$

Output: Vertices of the adversarial simplicial complex $\mathcal{K}$

1: Init. $x_0^* = x_\tau, g_0 = 0$

2: **for** $i = 0 \to t-1$ **do**

3: Input x_0^* and x_τ to $f_{im}(\cdot)$ and obtain the gradient $\nabla_x \mathcal{L}(f_{im}(x_\tau^*), f_{im}(x_\tau), \mathbf{tar})$; Use Eq(**??**)

4: Update g_{i+1} by accumulating the velocity vector in the gradient direction as

$$g_{i+1} = \mu \cdot g_t + \frac{\mathcal{L}(f_{im}(x_i^*), f_{im}(x_i))}{\|\nabla_x \mathcal{L}(f_{im}(x_i^*), f_{im}(x_i))\|_1};$$

5: Update x_{i+1}^* by applying the sign gradient as

$$x_{i+1}^* = x_i^* + \frac{\epsilon}{t} \cdot \text{sign}(g_{i+1});$$

6: **end for**

7: **return** Initial vertex of ASC $x_0^0 = x_t^*$

1. Initialization for ASC

In Section 3.2, we propose our adversarial simplicial complex (ASC)initialization method, a novel method that learns an adversarially strong initial vertex for the adversarial complex which is closer to the surrogate model's training data domain to enhance the attack potential of our simplicial complex. Specifically, we use MIM gradient optimization method, incorporating a domain re-adaptation regularization term into the optimization objective: ensuring that adversarial samples in feature space are distant from the semantics of clean images, while reducing the distribution discrepancy between adversarial samples and training data.This leads to further improved generalization performance of the attacks to downstream models,laying the groundwork for the optimization of ASC.The detailed implementation of Initialization is shown in Algorithm 1.

2. Algorithm for ASC

With the initial vertex in Section 3.2, we propose a novel Vertex-refine ASC optimization strategy. To enhance the diversity of the ASC, we designed a patch arrangement role tailored for the ViT architecture of the surrogate model. After augmenting the adversarial initial vertex, it serves as the starting point for each simplex in the complex. To ensure the stability (overall quality) of ASC, we opt for the Vertex-refine approach, which fixes known vertices and seeks the geometric center as the new initial position for optimization. To guarantee the attack transferability of the adversarial simplicial complex and to maximally explore the vulnerabilities of the surrogate model, we maximize the volume of the ASC while ensuring it remains in a high-loss region. The detailed implementation of Initialization is shown in Algorithm 2.

3. Visualization of predictions

We randomly selected images for each task to visualize their segmentation results under different methods, allowing for a direct comparison of performance in Fig. 1. This visualization demonstrates the severe threat posed by our VeSCA adversarial attack to downstream task models, significantly reducing the reliability of predictions from large foundational models using an independent input image. Notably, since SpaceNet consists of large high-altitude satellite images, to best showcase the attack effects, we randomly chose a section of it for magnification.

4. Parameter analysis

To explore VeSCA's sensitivity to hyper parameters, we investigated the impact of several key parameters on ISTD datasets. As shown in Figure 2, we analyzed how different hyper parameters affect VeSCA's performance and identified their optimal values for our previous experiments.First, the number of iterations t in the initialization of the adversarial simplex increases BER as it grows, the performance converges at $t = 10$. Second, the Monte Carlo trials H shows an evident improvement in BER with its increase , peaking at $H = 6$

For N simplex vertices increasing, BER rises, peaking at $n = 4, 5$, then slowly falls. This may be due to the expanding simplicial complex pushing random samples further from the training domain. When the number of simplex is constant, increasing the number of vertices M per simplex boosts BER, peaking at $M = 5$, then stabilizing.BER

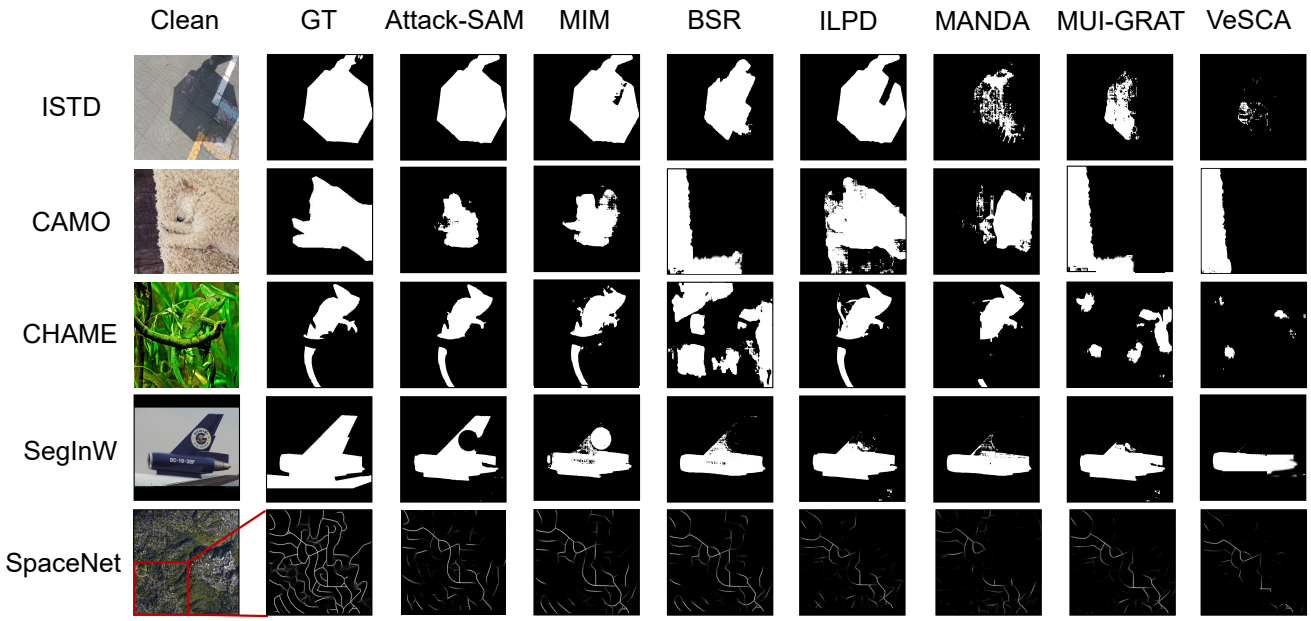

Figure 1. The visualized adversarial attack results in ISTD, CAMO, CHAME,SegInW, SpaceNet datasets by different attacks.

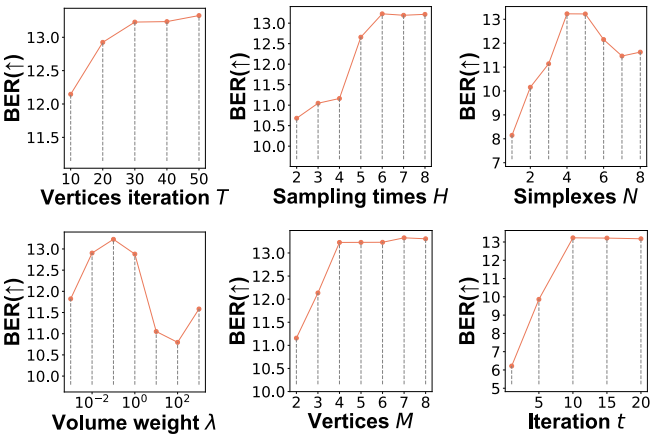

Figure 2. The BER(↑) of adversarial examples generated by VeSCA on ISTD dataset with the change of six hyperparameters

peaks at a volume weight $\lambda = 0.1$, indicating optimal adversarial impact. Below 0.1, BER rises with increasing λ, while above 10, BER declines, suggesting diminishing returns on adversarial effectiveness as λ moves away from the optimum.When $\lambda = 100$, BER increases but does not exceed the value at $\lambda = 0.1$.

5. Information of Datasets

In Section 4.1, we use five datasets for different downstream tasks. The datasets include ISTD which contains 1870 image triplets of shadow images, CHAMELEON that contains 76 camouflaged images, and the CAMO dataset which contains 1500 camouflaged object images.The CityScale dataset contains 180 satellite images of 20 U.S. cities.The SegInW dataset for zero-shot segmentation which contains 2330 images from 25 open classes.

6. Impacts of work

Our method generates highly transferable adversarial samples using an open-source SAM base model to compromise its downstream task models. Unlike prior efforts, we focus on identifying vulnerable spaces that pose risks to downstream tasks by constructing an Adversarial Simplicial Complex (ASC). This allows for a more comprehensive assessment of the foundational model's risk propagation.

Experiments across multiple downstream models and datasets demonstrate that the base model's security risks pose significant potential threats to downstream tasks. The numerous adversarial samples within ASCs which have large volume successfully attack downstream tasks, highlighting the urgent need for robust defense mechanisms to protect these models from adversarial threats.

7. Hardware

Tests on downstream task models are performed on NVIDIA GeForce RTX 3090 GPUs, except that experiments on generating downstream adversarial samples by various attack methods are conducted on two NVIDIA H800 PCIe GPUs with 80 GB memory due to their large memory requirements.

Algorithm 2 Vertex-Refining Simplicial Complex Attack Algorithm

Input: SAM encoder $f_{im}(\cdot)$, task-specific image x_τ, Loss function $\mathcal{L}(\cdot)$, Initial vertex x_0^0, PAR augmentation $\mathcal{PAR}(\cdot)$, segmentation scale ns, random angle θ, number of vertex's optimization iteration$T$,perturbation budget ϵ, decay factor μ, number of simplexes N,number of vertices M, regularization parameter λ^*, Monte Carlo times$H$.

Output: ASC $\mathcal{K}_{adv}$

$x_0^i = \mathcal{PAR}(x_0^0, ns), i = 1, 2, \ldots, N-1$

Get Initial ASC as

$$\mathcal{K}(S_0(x_0^0), S_1(x_0^1), \ldots, S_{N-1}(x_0^{N-1}))$$

for $n = 0 \to N-1$ **do**

  $S'_n = S_n$

  for $m = 0 \to M-2$ **do**

    Init. $g_0 = 0$ and $x_0^* = \frac{x_0^n + \sum_{t=1}^{m+1} x_t^n}{m+2}$

    for $k = 0 \to T-1$ **do**

      Input x_k^* and x_τ to $f_{im}(\cdot)$,and sample H vertices$x_{k,h}^*$ from $S = S'_n \cup x_k^*$

      if $m = 0$ **then**

        $\mathcal{L}_{op} = \frac{\sum_{h=1}^{H} \mathcal{L}(f_{im}(x_{k,h}^*), f_{im}(x_\tau))}{H}$

      else

        $\mathcal{L}_{op} = \frac{\sum_{h=1}^{H} \mathcal{L}(f_{im}(x_{k,h}^*), f_{im}(x_\tau))}{H} + \frac{\lambda^*}{\log \mathcal{V}(S'_n)} * \log \mathcal{V}(S)$

      end if

      Update g_{t+1} and x_{k+1}^*

$$g_{k+1} = \mu \cdot g_k + \frac{\nabla_x \mathcal{L}_{op}}{\|\nabla_x \mathcal{L}_{op}\|_1};$$

$$x_{k+1}^* = x_k^* + \frac{\epsilon}{T} \cdot \text{sign}(g_{k+1});$$

    end for

    $x_{m+1}^n = x_T^*$ and $S'_n = S'_n \cup x_{m+1}^n$

  end for

end for

return $\mathcal{K}_{adv}$