Transferable Sparse Adversarial Attack against Deep Learning Models

Wanpeng Xu

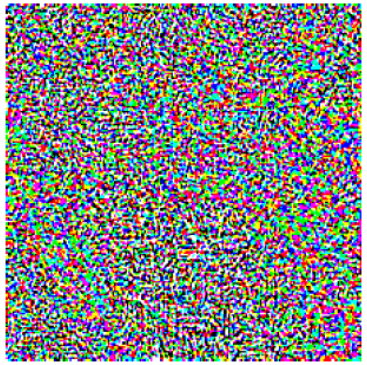
Contents

Seminal Work

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Research Status of Adversarial Attacks

White-box Attack

- FGSM
- DeepFool
- C&W
- PGD / PGD_0
- JSMA
- SparseFool
- …

Black-box Attack

- ZOO
- Boundary Attack
- UAP
- One Pixel Attack
- Pointwise Attack
- CornerSearch
- …

JUST TRY IT! JSMA!

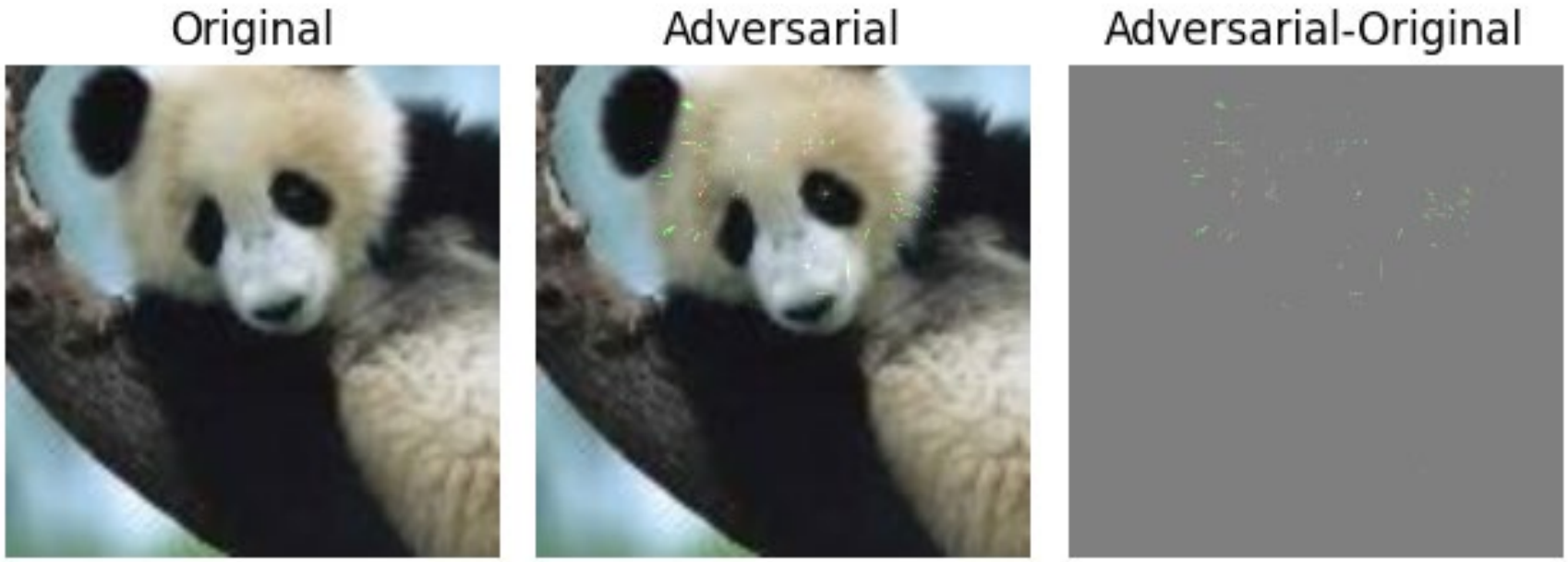
GPU: 5 seconds or less
CPU: 1 minute or less


cropped_panda.jpg


JSMA.ipynb

```
for epoch in range(epochs):
    output = model(img)
    label = np.argmax(output.data.cpu().numpy())
    loss = loss_func(output, target)
    print('epoch={} label={} loss={}'.format(epoch, label, loss))
    if label == target_label:
        break               # 攻击成功
    zero_gradients(img)     # 梯度清零
    idx, pix_sign = saliency_map(output, img, target_label, mask)
    # 添加扰动
    img.data[idx] = img.data[idx] + pix_sign * theta * (max_ - min_)
    # 达到极限的点不再参与更新
    if (img.data[idx] <= min_) or (img.data[idx] >= max_):
        print('idx={} over {}'.format(idx, img.data[idx]))
        mask[idx] = 0
        img.data.cpu()[idx] = np.clip(img.data.cpu()[idx], min_, max_)
```

Attack Result



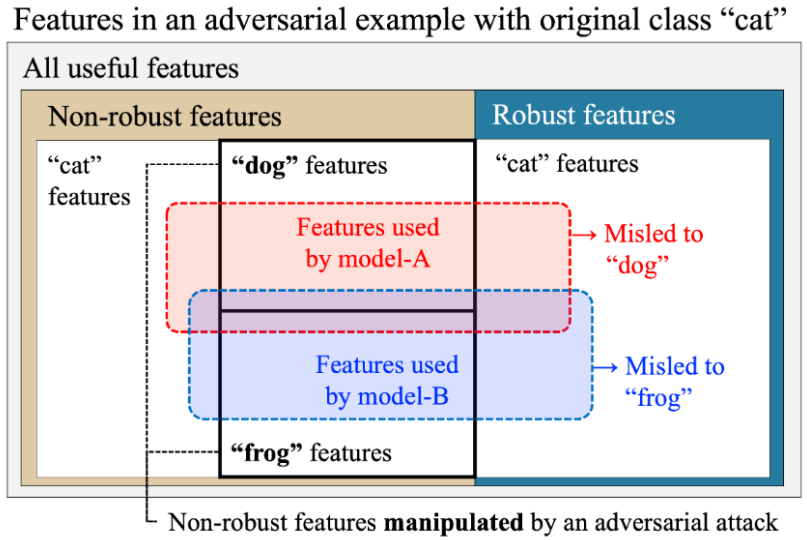
Research Status of Adversarial Training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta \in B(x,\varepsilon)} \mathcal{L}_{ce}(\theta, x+\delta, y) \right]$$

- Adversarial Regularization
- Curriculum-based Adversarial Training
- Ensemble Adversarial Training
- Adversarial Training with Adaptive ϵ
- Adversarial Training with Semi/Unsupervised Learning
- Efficient Adversarial Training
- …

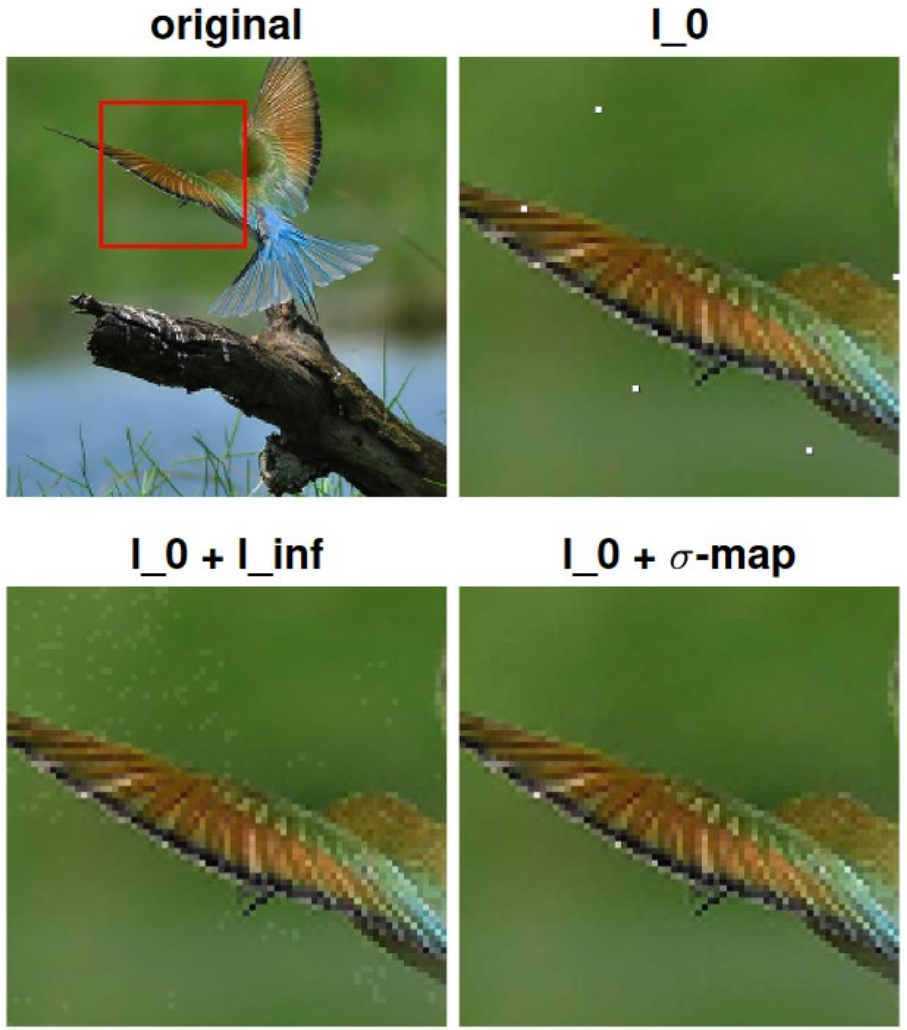
Our Work — Transferable Sparse Adversarial Attack

Transferability

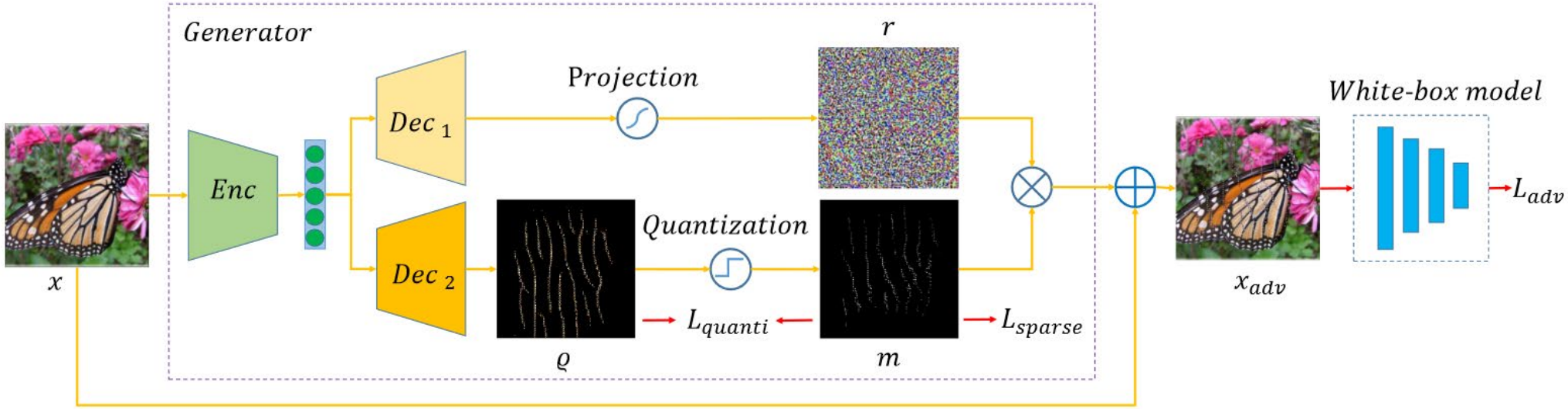


True Label	Plane	Car	Bird	Cat	Deer
Pred: Res-18	Truck	Frog	Cat	Frog	Bird
Pred: VGG-16	Deer	Frog	Frog	Ship	Horse

Sparsity



Trainable Generator Architecture



$$\boldsymbol{\delta} = \mathbf{r} \odot \mathbf{m}$$

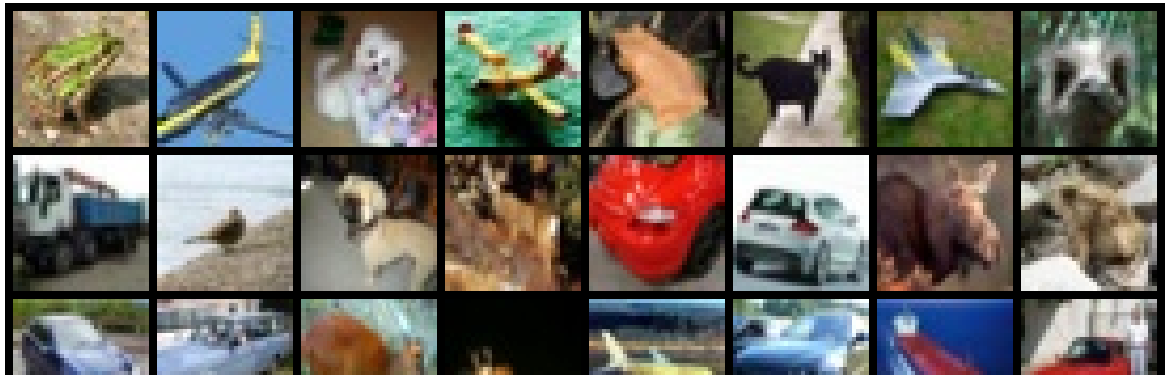
$$\begin{aligned} & \min_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_0 \\ \text{s.t.} \quad & \arg\max_c f(\mathbf{x}+\boldsymbol{\delta})_c \neq y \quad \Rightarrow \quad \arg\max_c f(\mathbf{x}_{adv})_c = y_t \\ & \|\boldsymbol{\delta}\|_\infty < \epsilon \end{aligned}$$

$$\mathbf{r} = \epsilon \cdot \mathcal{D}_1(\mathbf{z})$$

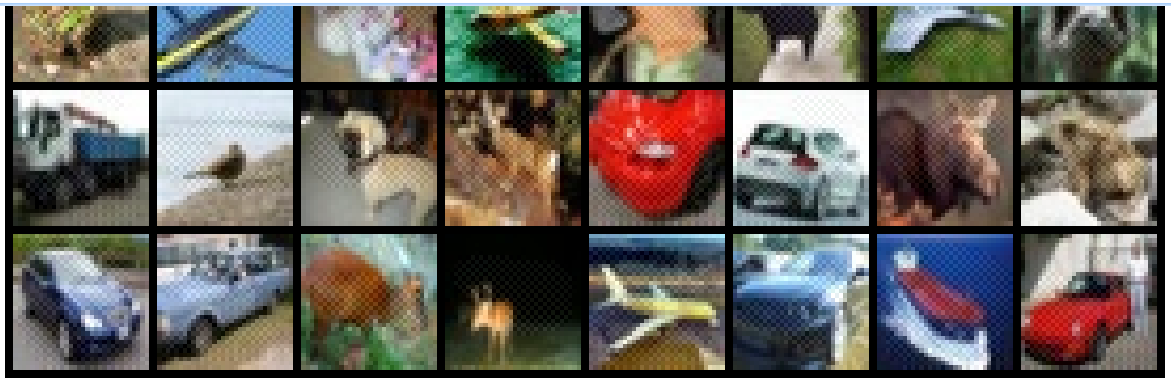
$$q(\varrho_{i,j}) = \begin{cases} 0 & \varrho_{i,j} \leqslant \tau \\ 1 & \varrho_{i,j} > \tau \end{cases}$$

$$P(X = k) = p^k (1-p)^{1-k}, k \in \{0, 1\}$$

Attack Result



```
root@autodl-container-90b111a9b0-190f96e6:~/autodl-tmp/TSAA_Capstone_Project/code# ./run.sh my_eval.py
Namespace(batch_size=10, eps=10, generator_weight_path='/root/autodl-tmp/TSAA_Capstone_Project_Big_Files/pretrained_generators/netG_-1_res50_eps10_epoch9_loss96.1.pth', model='res50', num_classes=10, target_class=-1, target_model='res18', target_model_we
ight='/root/autodl-tmp/TSAA_Capstone_Project/code/pytorch_cifar/checkpoint/res18/20230403_ckpt.pth', test_dir='/root/autodl-tmp/TSAA_Capstone_Project_Big_Files/dataset/CIFAR_10/cifar-10-batches-py/test')
Test data size: 10000
L0 norm: tensor(499.2936, device='cuda:0')
time: 0.0
Acc in Clean Image: 89.670%      Acc in Adversarial Image: 27.830%      Fooling Rate:71.880%
root@autodl-container-90b111a9b0-190f96e6:~/autodl-tmp/TSAA_Capstone_Project/code# ./run.sh my_eval.py
Namespace(batch_size=10, eps=10, generator_weight_path='/root/autodl-tmp/TSAA_Capstone_Project_Big_Files/pretrained_generators/netG_-1_res50_eps10_epoch9_loss96.1.pth', model='res50', num_classes=10, target_class=-1, target_model='res34', target_model_we
ight='/root/autodl-tmp/TSAA_Capstone_Project/code/pytorch_cifar/checkpoint/res34/20230403_ckpt.pth', test_dir='/root/autodl-tmp/TSAA_Capstone_Project_Big_Files/dataset/CIFAR_10/cifar-10-batches-py/test')
Test data size: 10000
L0 norm: tensor(499.2936, device='cuda:0')
time: 0.0
Acc in Clean Image: 90.990%      Acc in Adversarial Image: 29.210%      Fooling Rate:70.340%
130 root@autodl-container-90b111a9b0-190f96e6:~/autodl-tmp/TSAA_Capstone_Project/code# ./run.sh my_eval.py
Namespace(batch_size=10, eps=10, generator_weight_path='/root/autodl-tmp/TSAA_Capstone_Project_Big_Files/pretrained_generators/netG_-1_res50_eps10_epoch9_loss96.1.pth', model='res50', num_classes=10, target_class=-1, target_model='res50', target_model_we
ight='/root/autodl-tmp/TSAA_Capstone_Project/code/pytorch_cifar/checkpoint/res50/ckpt.pth', test_dir='/root/autodl-tmp/TSAA_Capstone_Project_Big_Files/dataset/CIFAR_10/cifar-10-batches-py/test')
Test data size: 10000
L0 norm: tensor(499.2936, device='cuda:0')
time: 0.0
Acc in Clean Image: 90.650%      Acc in Adversarial Image: 18.760%      Fooling Rate:81.050%
```



Thanks for Listening!