Supplementary Material for "Leveraging Local Patch Differences in Multi-Object Scenes for Generative Adversarial Attacks"

Abhishek Aich, Shasha Li, Chengyu Song, M. Salman Asif,
Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury
University of California, Riverside, USA
{aaich001@, sli057@, csong@cs., sasif@ece., krish@cs., amitrc@ece.}ucr.edu

Baselines. We use three state-of-the-art generative attack methods (GAP [1], CDA [2], and TDA [3]) as our baselines. We change the cross-entropy loss in GAP [1] and CDA [2] with binary cross-entropy loss to adapt it for multi-object surrogate classifier.

Implementation Details. Following [2], we use the ResNet architecture introduced in [4] as the generator network for $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$, except we replace the ReLU [5] activation with Fused Leaky ReLU [6] activation to stabilize training (negative slope = 0.2, scale set = $\sqrt{2}$). We use Adam optimizer [7] with a learning rate 0.0001, batch size 32, and exponential decay rates between 0.5 and 0.999. All images are resized to 224×224 and normalized with mean and standard deviation before feeding to generator. Further, similar to gaussian smoothing in CDA [2], in order to make the perturbations more transferable, we clamp the perturbed image between 0 to 1. This clamping trick helps in increasing the transferability of the perturbations. For fair comparison, we apply this strategy to all attacks. Perturbation generators are trained for 20 epochs. We use PyTorch [8] in all experiments. Training time is 1 hr for Pascal-VOC dataset and 10 hrs for MS-COCO dataset on two NVIDIA GeForce RTX 3090 GPUs. For all experiments, number of patches for $\mathcal{L}_{\text{LPCL}}$ is set to 128.

White-box, black-box, and strict black-box attacks. We sanalyze white-box and black-box attack (attack in same distribution as adversary) performance of **LPD-Attack** in Table 1(a). This attack tests the strength of perturbations on the same type of task (*i.e.* multi-object classification) as training. Our proposed method shows a stronger attack than GAP [1] and CDA [2]. In comparison to TDA [3], our attack shows comparable performances (for cases where TDA [3] does better, the difference is very small) in most cases even though we do not need to manually choose a specific layer for each classifier to train the perturbation generator. Choosing a particular mid-layer for every classifier does not always guarantee better transferability of perturbations. A similar observation can be made in Table 1(b).

Table 1: *Generative Attack Comparison when $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ is trained with MS-COCO:* Gray colored cells represent the white-box attacks. $\boldsymbol{f}(\cdot)$ in both Table 1(a) and Table 1(b) are pre-trained on MS-COCO.

(a) *Setting* 1 and *Setting* 2

$\boldsymbol{f}(\cdot)$	Method	MS-COCO Trained Victim Models		
		Res152	**VGG19**	**Dense169**
		67.95%	66.49%	67.60%
Res152	GAP [1]	44.98%	34.61%	43.91%
	CDA [2]	45.00%	34.91%	44.36%
	TDA [3]	39.60%	29.41%	39.66%
	LPD-Attack	41.02%	30.18%	40.33%
VGG19	GAP [1]	44.83%	34.67%	44.10%
	CDA [2]	44.41%	30.63%	43.53%
	TDA [3]	39.81%	23.04%	38.96%
	LPD-Attack	40.09%	24.23%	39.11%
Dense169	GAP [1]	44.55%	34.28%	43.61%
	CDA [2]	44.92%	34.86%	44.24%
	TDA [3]	42.69%	31.96%	40.30%
	LPD-Attack	41.96%	30.19%	39.47%

(b) *Setting* 3

$\boldsymbol{f}(\cdot)$	Method	Pascal-VOC Trained Victim Models		
		Res152	**VGG19**	**Dense169**
		83.12%	83.18%	83.73%
Res152	GAP [1]	58.80%	48.67%	60.80%
	CDA [2]	58.67%	48.66%	60.92%
	TDA [3]	54.05%	43.29%	57.55%
	LPD-Attack	55.76%	43.88%	58.15%
VGG19	GAP [1]	59.09%	48.61%	61.17%
	CDA [2]	58.44%	45.20%	60.35%
	TDA [3]	55.06%	38.41%	57.49%
	LPD-Attack	55.24%	40.71%	58.10%
Dense169	GAP [1]	58.45%	48.18%	60.47%
	CDA [2]	58.69%	48.68%	61.04%
	TDA [3]	56.65%	45.52%	58.54%
	LPD-Attack	56.08%	43.46%	57.23%

References

[1] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative Adversarial Perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431. IEEE, 2018.

[2] Muzammal Naseer, Salman H Khan, Harris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-Domain Transferability of Adversarial Perturbations. *arXiv preprint arXiv:1905.11736*, 2019.

[3] Mathieu Salzmann et al. Learning transferable adversarial perturbations. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[5] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*. ICML, 2010.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410. IEEE, 2019.

[7] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pages 8026–8037. NeurIPS, 2019.