Ensemble-based Blackbox Attacks on Dense Prediction (Supplementary Material)

Summary

In the supplementary material, we provide additional results and analyses on joint attacks for multiple blackbox models of different dense prediction tasks, attacks on object detection, attacks on semantic segmentation, and corresponding visualization of adversarial examples for qualitative evaluation. We also report runtime and resource usage.

A. Joint attack for multiple blackbox models

In this section, we provide additional visualization results for joint (targeted) blackbox attacks against object detection and semantic segmentation models.

More Visualization of adversarial examples. We visualize some adversarial examples in Fig. S1. In Fig. S1a, we show an example where our method generates a single perturbed image to map the bicycle on the right-hand-side to train. In object detection results we see the label for the Bicycle bounding box has been changed to Train, and for the segmentation map, the corresponding region has changed to teal color encoding for Train as well. In Fig. S1b, the generated perturbed image maps the car in the middle to traffic light. Note that the bounding box for the Car in the middle changed to Traffic Light for the object detector and the same area in the semantic segmentation map changed the color to orange (corresponding to Traffic Light label).

B. Attacks against object detection

Attacks using different surrogate models. In our previous experiments (Tab. 1 and Tab. S1), we follow the model selection in [4] for a fair comparison. We can easily replace the surrogate models with different ones and expect its effectiveness across different settings. For example, we can replace YOLOv3 with Deformable DETR (denoted as Deform) and get similar results, as shown in Tab. S2 below. The experiment setup and victim models are same as reported in Tab. S2 for $\ell_\infty = 20$.

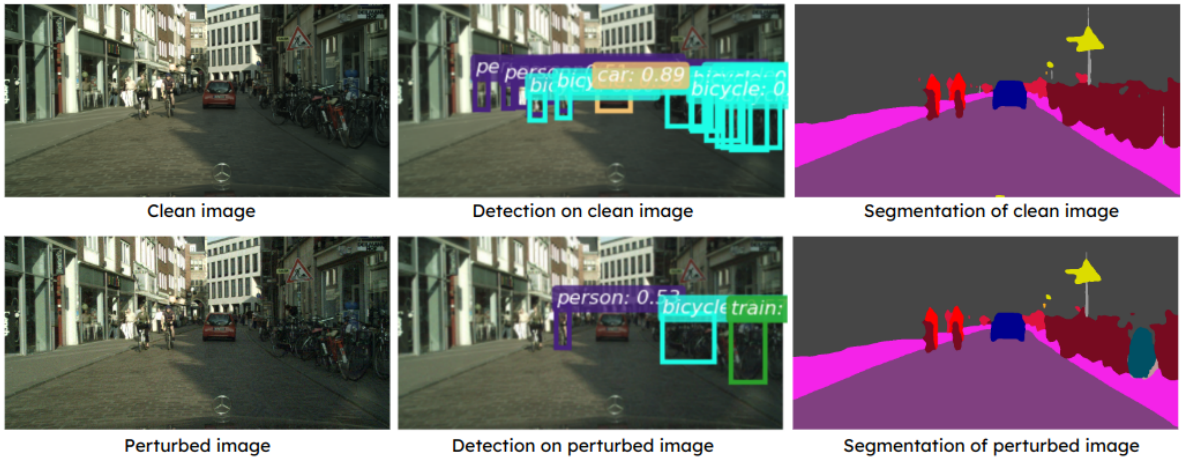
Comparisons with zero-query attacks. Zero-Query attack (ZQA) [2] does not rely on any feedback from the victim. It assesses the attack success probability on the surrogate model before launching a single and most promising attack against the victim. Due to these differences in problem setting, we do not directly compare with this method in the main paper. Here we compare the numbers reported from corresponding manuscripts in Tab. S2. ZQA uses a single surrogate model without any feedback from the victim model. It performs worse than the few-query attacks [4] with 3–5 queries, and our method clearly outperforms both of them.

Comparison with conventional query-based attacks. Existing query-based methods, including GARSDC [27] and PRFA [28], require thousands of queries (which is prohibitive) and they are only applicable for untargeted attacks. Furthermore, their perturbations are clearly visible, see Fig. 5 in [27], while our perturbations remain imperceptible. For these key differences, we did not include their comparison in the main paper, but here we provide a mAP score comparison with them. We use 5 surrogate models from Tab. 2 and perform vanishing attacks on ATSS [58] model, we show in Tab. S3 that our method can achieve a near-zero mAP within just a few queries (Q).

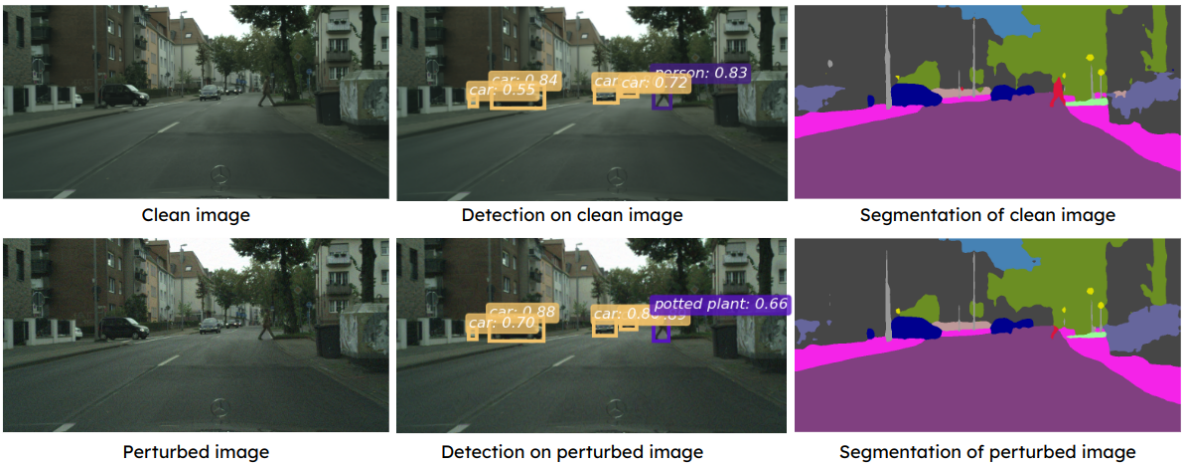
C. Attacks against semantic segmentation

Attacks on Pascal VOC dataset. We generate adversarial attacks using different sizes of ensemble and report mIoU scores on the Pascal VOC dataset in Tab. S5. Similar to the results on the Cityscapes dataset in Tab. 3, as we increase the number of surrogate models from 2 to 6, the attack performance improves (indicated by smaller mIoU scores). Attack performance of our method further improves with weight optimization (with $Q = 20$). These results show that by adjusting the weights of the surrogate ensemble, we can improve the attack performance. Our attack method with $N = 6$ surrogate models provides 27–29% improvement in mIoU scores compared to DS attack for the victim models `PSPNet-Res50` and `DeepLabV3-Res50`. Note that DS attack uses these two models as the whitebox surrogates as well victim models. In contrast, we keep all four victim models `PSPNet-Res50`, `DeepLabV3-Res50`, `PSPNet-Res101`, `DeepLabV3-Res101` out of our ensemble. Our surrogate ensemble consists of `FCN`, `UPerNet`, `PSANet`, `GCNe`, `ANN`, `EncNet` with `ResNet50` backbones, which reflects a more realistic setting where the victim blackbox model is different from any of the surrogate models.

Effect of backbones on attack performance. We note that for VOC dataset results in Tab. S5, our method provides high attack success for blackbox victim models with `ResNet50` backbone. However, the attack performance on victim models with `ResNet101` backbone degrades (as re-



(a) Generate perturbation to map the Bicycle on the right-hand-side to Train. Image id: `munster_000140_000019`



(b) Generate perturbation to map the Pedestrian to Potted Plant. Image id: `munster_000006_000019`

(c) Color encoding for segmentation maps in CityScapes dataset

Figure S1. Visual adversarial examples of our method that generates successful attacks to fool a blackbox object detector and a blackbox semantic segmentation model using a single perturbed image.

flected by large mIoU values). To further demonstrate the effectiveness of our attack, we replace the backbones of the surrogate models with the `ResNet101` backbones while keeping the rest of model architectures same as the original ensemble. Results reported in Tab. S6 show that if we replace surrogate models with `ResNet101` backbones (same backbone as the victim blackbox models), then our attack method provides significantly better results.

Attack performance on different backbones. We performed additional experiments using `FCN` and `PSPNet` methods and `MobileNetV2` and `ResNeSt` (denoted as -mv2 and -s101 in Tab. S4) backbones for victim models. The attack setting corresponds to Tab. 4. Due to the great difference in backbones across surrogate and victim, the attack performance drops. Nevertheless, the attack performance improves significantly as we increase the ensemble size and optimize ensemble weights. Results are reported in Tab. S4.

Attack performance on surrogate models. For the sake of completeness, we also report attack performance on the whitebox surrogate models for both untargeted and targeted attacks in Tab. S7 and Tab. S8. We observe that as we increase the number of models in the ensemble from $N = 1$ to $N = 5$, we can achieve better attack performance on all the whitebox and blackbox victim models we tested. Attacks that are successful on blackbox victim models are almost always successful on all surrogate models.

Visualization of adversarial examples. We present some

Table S1. Targeted attack success rate (%) for different methods on COCO dataset. Similar setting as in Tab. 1.

Perturbation Budget	Weight Balancing	Weight Optimization	**Surrogate Ensemble** FRCNN	**Surrogate Ensemble** YOLOv3	**Blackbox Victim Models** (ASR ↑) RetinaNet	Libra	Fovea	Free	DETR
$\ell_\infty = 10$	✗	✗	19.6	79.7	4.6	5.0	4.4	6.6	2.6
	✗	✓	49.8	**97.8**	13.5	16.2	14.4	22.6	8.4
	✓	✗	57.2	65.3	16.2	17.6	16.6	24.0	5.4
	✓	✓	**78.0**	86.1	**31.7**	**32.0**	**32.3**	**41.6**	**15.4**
	Context-aware Attack [4]		41.2	54.4	12.0	11.2	18.6	25.0	10.8
$\ell_\infty = 20$	✗	✗	25.8	82.2	8.9	9.8	8.4	13.2	5.6
	✗	✓	62.4	**98.2**	23.0	32.2	22.4	32.2	13.2
	✓	✗	68.8	75.8	25.5	28.0	27.1	38.0	13.8
	✓	✓	**88.9**	94.5	**48.5**	**53.8**	**49.5**	**65.6**	**31.0**
	Context-aware Attack [4]		64.4	70.0	20.8	22.2	35.4	40.8	20.0
$\ell_\infty = 30$	✗	✗	29.0	82.2	8.8	9.4	13.3	14.6	6.4
	✗	✓	69.0	**99.3**	27.9	34.6	31.2	43.6	17.6
	✓	✗	72.7	78.6	32.5	33.8	34.1	41.6	14.8
	✓	✓	**91.7**	95.5	**57.6**	**64.4**	**58.3**	**71.2**	**36.6**
	Context-aware Attack [4]		68.6	75.4	27.2	27.2	39.2	46.2	21.2

Table S2. Replacing YOLOv3 with Deformable DETR. Correspond to Tab. 1, perturbation budget $\ell_\infty = 20$.

Weight Balancing	Weight Optimization	**Surrogate Ensemble** FRCNN	**Surrogate Ensemble** Deform	**Blackbox Victim Models** (ASR ↑) Retina	Libra	Fovea	Free	DETR
✗	✗	8.5	69.5	10.6	4.0	8.0	10.5	12.0
✗	✓	34.6	94.3	36.7	25.0	33.5	53.0	38.5
✓	✗	68.0	80.5	47.2	38.5	37.5	57.5	26.5
✓	✓	**88.1**	**95.0**	**74.9**	**70.5**	**73.0**	**84.0**	**56.0**
ZQA [2]		88.2	-	44.0	51.4	53.4	-	-

Table S3. Comparison with conventional query-based attacks.

Method	ATSS [58] mAP ↓	ATSS [58] Q ↓
Clean	0.54	N/A
PRFA [28]	0.20	3500
GARSDC [27]	0.04	1837
Ours	**0.00**	**10**

Table S4. Semantic segmentation targeted pixel success ratio (PSR) (%) for blackbox victim models with different backbones.

Q	N	**Blackbox Victim Models** (PSR ↑) FCN-mv2	FCN-s101	PSP-mv2	PSP-s101
0	1	33.26	1.01	3.96	2.71
	3	30.39	1.39	5.82	6.94
	5	38.92	3.12	7.84	8.32
20	3	50.31	22.79	24.09	54.06
	5	**53.09**	**34.57**	**30.20**	**60.43**

visual examples of untargeted attacks in Fig. S2 and targeted attacks in Fig. S4. We observe that the attacks generated by surrogate model do not transfer to the victim model for untargeted or targeted cases (i.e., $Q = 0$). The attacks generated after weight optimization (i.e., $Q = 20$) succeed for untargeted and targeted attacks. Our targeted attack setup is visually explained in Fig. S3. Instead of mapping every pixel prediction to an arbitrary target label, we focus on attacking a single object y in the original prediction (e.g."road" in Fig. S3a with white bounding-box). We select the target label $y^\star$ as the class that appears most frequently as the least-likely label of the pixels in the selected region. For example, Fig. S3b shows class "building" in grey color as the least likely class in the target region. Finally, we generate attack to replace the entire selected region in the original prediction to its target label (Fig. S3c).

D. Runtime and resource usage

We performed experiments on a single RTX 3090 GPU. Average time per query to attack an object detector for a 375×500 image with ensemble size $N = \{2, 5\}$ is $\{0.5, 1\}$sec. Average time per query to attack a segmentation model for a 512×1024 image with ensemble size $N = \{2, 5\}$ is $\{2.5, 5.5\}$sec.

Table S5. mIoU scores (%) for untargeted attacks on semantic segmentation models with Pascal VOC dataset. The lower value indicates better attack performance. Surrogate of ensemble sizes $N = 2, 4, 6$. We compare $Q = 0$ (i.e. direct transfer attack) with $Q = 20$ ensemble attack performance. Results show enabling the ensemble query introduced attack performance increments. Blue numbers represent whitebox attacks.

Method	**Whitebox Surrogate**	**Blackbox Victim Models** (mIoU ↓)			
		PSPNet-Res50	PSPNet-Res101	DeepLabV3-Res50	DeepLabV3-Res101
Clean Images	-	76.78	78.47	76.17	78.70
Baseline	PSPNet-Res50	5.09	37.06	6.57	38.98
	DeepLabV3-Res50	3.63	22.01	3.14	22.58
DS	PSPNet-Res50	2.07	16.10	2.56	18.57
	DeepLabV3-Res50	2.31	**12.32**	2.15	**13.64**
Ours ($Q = 0$)	$N = 2$	14.33	35.47	12.31	35.31
	$N = 4$	8.74	29.41	7.92	28.01
	$N = 6$	7.28	24.28	6.75	24.63
Ours ($Q = 20$)	$N = 2$	5.56	27.49	4.43	28.46
	$N = 4$	2.23	22.24	2.09	20.34
	$N = 6$	**1.69**	18.07	**1.53**	17.61

Table S6. mIoU scores (%) for untargeted attacks on semantic segmentation models with Pascal VOC dataset. The lower value indicates better attack performance. Surrogate of ensemble sizes $N = 1$ to 6. We compare the performance of `ResNet50` and `ResNet101` backbones in the ensemble. The attack performance on `ResNet101` backbone victim models increases if we use the surrogate models with `ResNet101` backbone. Note there is no weight optimization for $N = 1$.

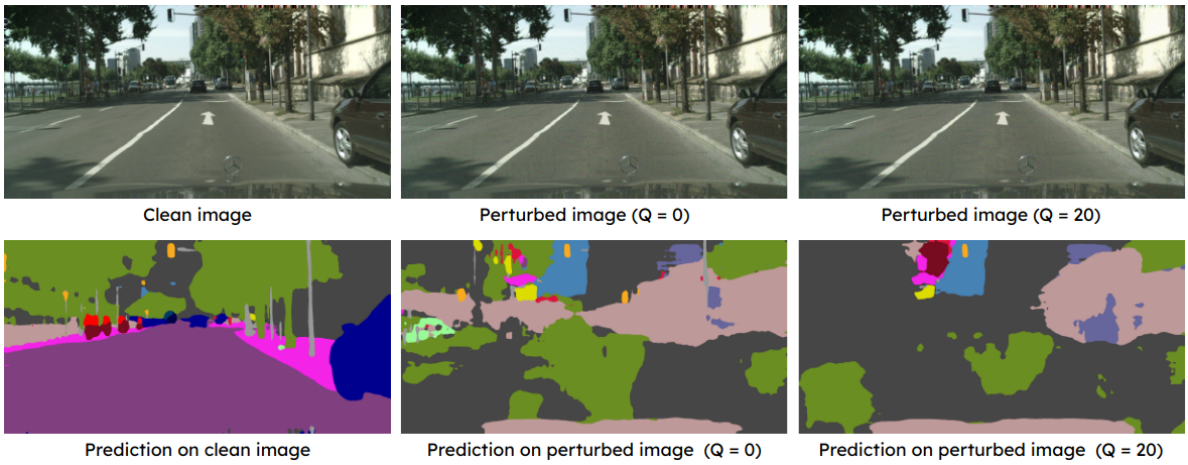
Q	N	**Blackbox Victim: PSPNet-Res101** (mIoU ↓)		**Blackbox Victim: DeeplabV3-Res101** (mIoU ↓)	
		Ensemble backbone: Res50	Ensemble backbone: Res101	Ensemble backbone: Res50	Ensemble backbone: Res101
0	1	38.51	24.76	38.66	25.98
	2	35.47	21.50	35.31	21.54
	3	31.95	17.65	32.39	18.02
	4	29.41	14.53	28.01	14.32
	5	25.82	13.67	24.79	12.28
	6	24.28	12.49	24.63	12.35
20	2	27.49	8.78	28.46	8.80
	3	24.80	5.15	22.55	5.69
	4	22.24	5.49	20.34	4.49
	5	19.62	**3.27**	18.31	**3.13**
	6	**18.07**	4.04	**17.61**	3.32

Table S7. Semantic segmentation untargeted attack mIoU scores (%) for blackbox victim models and whitebox surrogate models with different ensemble sizes (N). The lower value indicates better attack performance. Experiment with CityScapes dataset, $\ell_\infty \leq 8$. PSP-r50, PSP-r101, DL3-r50, DL3-r101 stands for `PSPNet` and `DeepLabV3` built on `ResNet50`, `ResNet101` backbone respectively.

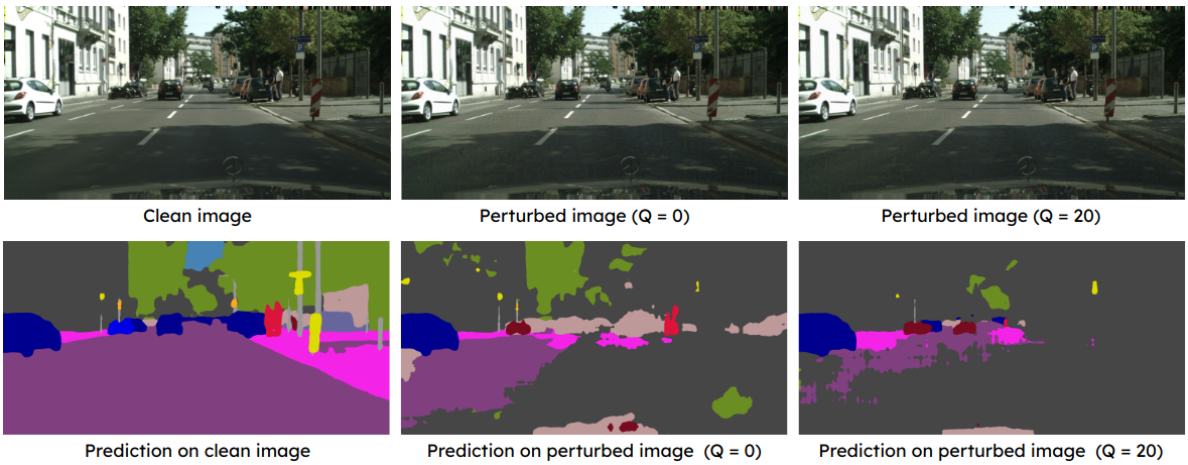
N	**Surrogate Ensemble**						**Blackbox Victim Models** (mIoU ↓)			
	FCN	UPerNet	PSANet	GCNet	ANN	EncNet	PSP-r50	PSP-r101	DL3-r50	DL3-r101
1	2.42	-	-	-	-	-	2.68	6.92	5.16	10.13
2	1.28	1.06	-	-	-	-	1.38	2.88	1.15	3.50
3	1.45	1.06	1.05	-	-	-	1.13	2.39	0.95	2.67
4	1.25	0.97	0.87	0.91	-	-	**0.79**	2.04	**0.73**	1.80
5	1.18	0.96	0.93	0.91	1.14	-	0.78	1.69	0.89	2.09
6	1.26	1.08	1.08	1.05	1.25	1.16	0.90	**1.55**	0.94	**1.09**

Table S8. Semantic segmentation targeted pixel success ratio (PSR) (%) for blackbox victim models and whitebox surrogate models with different ensemble sizes (N). The higher value indicates better attack performance. Experiment with CityScapes dataset, $\ell_\infty \leq 8$. PSP-r50, PSP-r101, DL3-r50, DL3-r101 stands for `PSPNet` and `DeepLabV3` built on `ResNet50`, `ResNet101` backbone respectively.

N	**Surrogate Ensemble**					**Blackbox Victim Models** (PSR ↑)			
	FCN	UPerNet	PSANet	GCNet	ANN	PSP-r50	PSP-r101	DL3-r50	DL3-r101
1	69.51	-	-	-	-	39.15	10.21	35.02	7.58
2	84.62	89.30	-	-	-	83.97	51.80	82.70	46.95
3	79.64	85.48	82.89	-	-	88.88	64.63	85.55	60.88
4	83.82	88.50	87.00	88.12	-	91.51	64.28	87.19	63.88
5	86.55	91.10	89.75	90.00	87.82	**92.91**	**69.09**	**88.95**	**69.65**

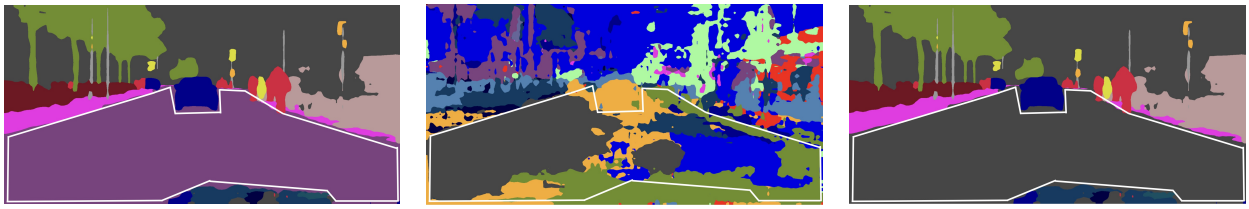


(a) Generate perturbation to maximize prediction error. We use the $N = 2$ surrogate ensemble to generate this attack against blackbox victim `PSPNet-Res50`. Image id: `frankfurt_000001_005703`



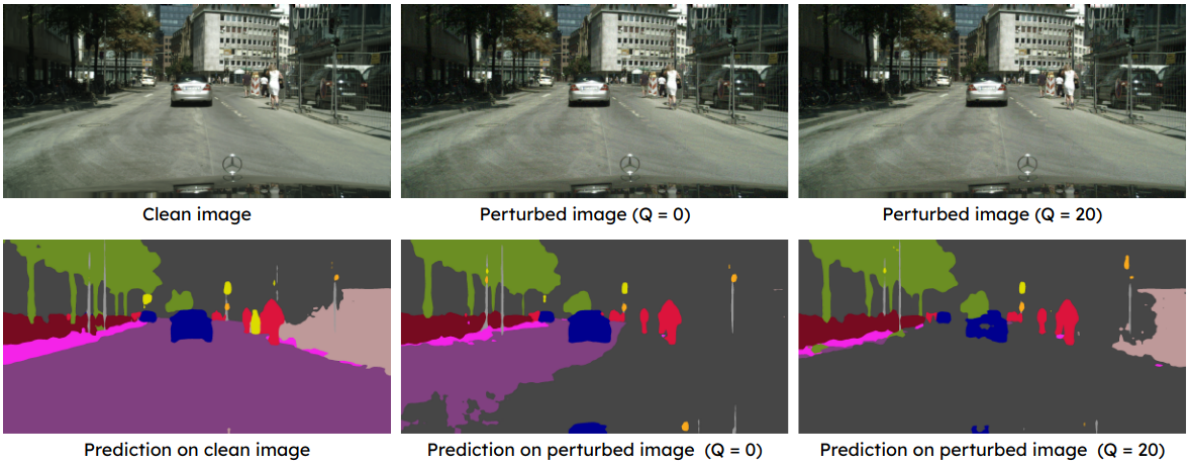
(b) Generate perturbation to maximize prediction error. We use the $N = 2$ surrogate ensemble to generate this attack against blackbox victim `DeepLabV3-Res101`. Image id: `frankfurt_000000_022797`

Figure S2. Visual adversarial examples of our method for untargeted attacks to fool a blackbox semantic segmentation model.

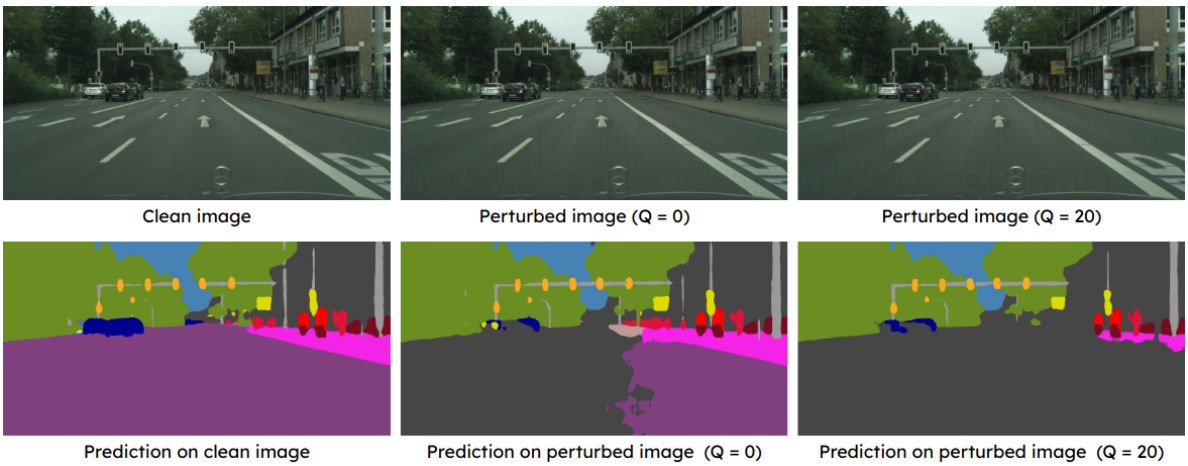


(a) Original Prediction **(b)** Least-likely Prediction **(c)** Attack Target

Figure S3. Our segmentation targeted attack setup. We select an object region y in the original prediction from surrogate `FCN` (Fig. S3a). Identify the targeted label $y^\star$ from Fig. S3b and craft the attack target Fig. S3c. Image id: `frankfurt_000001_007857`



(a) Generate perturbation to map the Road region to Building. We use the $N = 2$ surrogate ensemble to generate this attack against blackbox victim `PSPNet-Res50`. For the direct transfer attack (at $Q = 0$), only 67.08% of the pixels of the target region are successfully mapped to the desired class. After weight optimization (with $Q = 20$), pixel success rate increases to 99.77%. Image id: `frankfurt_000001_007857`



(b) Generate perturbation to map the Road region to Building. We use the $N = 6$ surrogate ensemble to generate this attack against blackbox victim `DeepLabV3-Res50`. For the direct transfer attack (at $Q = 0$), only 63.06% of the pixels of the target region are successfully mapped to the desired class. After weight optimization (with $Q = 20$), pixel success rate increases to 99.98%. Image id: `munster_000003_000019`

Figure S4. Visual adversarial examples of our method for targeted attacks to fool a blackbox semantic segmentation model.