

# Characterizing Adversarial Examples Based on Spatial Consistency Information for Semantic Segmentation

Chaowei Xiao<sup>1</sup>, Ruizhi Deng<sup>2</sup>, Bo Li<sup>3</sup>, Fisher Yu<sup>3</sup>, Mingyan Liu<sup>1</sup>,  
Jinfeng Yi<sup>4</sup>, and Dawn Song<sup>3</sup>

<sup>1</sup>University of Michigan    <sup>2</sup>Simon Fraser University    <sup>3</sup>UC Berkeley    <sup>4</sup>JD AI  
Research

## 1 Adversarial Examples For Cityscapes and BDD Datasets Against DRN and DLA models

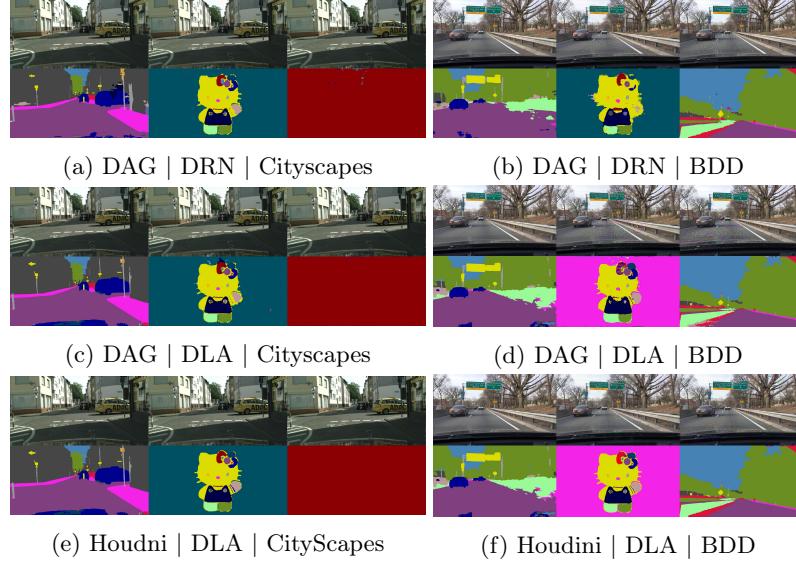


Fig. 1: Samples of benign and adversarial examples. We use the format “attack method | attack model | dataset” to label the settings of each adversarial examples. Within each subfigure, the first column shows benign images and corresponding segmentation results, the second and third columns show adversarial examples with different adversarial targets (targeting on Kitty/Pure in (a),(c), (d) and on Kitty and Scene in (b),(d),(f)).

Figure 1 shows the benign and adversarial examples targeting at diverse adversarial targets: “Hello Kitty” (Kitty) and random pure color (Pure) on

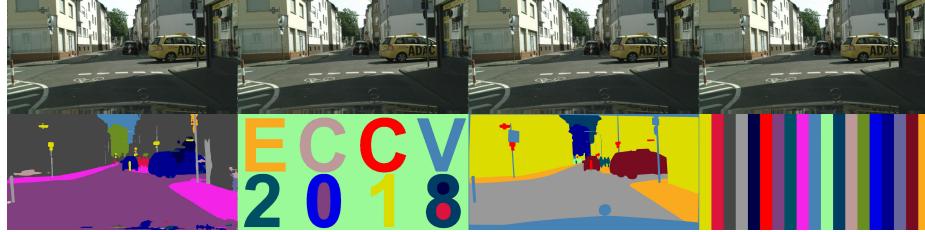


Fig. 2: Attack results of additional targets on Cityscapes. The first column shows benign instance, while 2-4 columns show adversarial examples with target “ECCV 2018”, “Remapping”, and “Color strip”, respectively.

Cityscapes [2]; and “Hello Kitty” (kitty) and a real scene without any cars (Scene) on BDD [?] dataset against DRN [4] and DLA [5] segmentation models. In order to increase the diversity of our target set, we also apply different colors for the background of “Hello Kitty” on BDD dataset against DLA model.

Figure 2 shows the additional adversarial targets, including “ECCV 2018”, “Remapping”, and “Color strip”. Here remapping means we generate an adversarial target by shifting the numerical label of each class in the ground truth by a constant offset. This way, we can guarantee that each target has no overlap with the ground truth mask. For “Color strip”, we divide the target into 19 strips evenly, each of which is filled with a class label, aiming to mitigate possible bias for different classes.

## 2 Spatial Consistency Based Method

---

### Algorithm 1: Patch Selection Algorithm (`getOverlapPatches`)

---

```

input : patch size  $s$   

        image width  $w$   

        image height  $h$   

        bound  $b_{\text{low}}, b_{\text{upper}}$   

output: Two random patches  $P_1$  and  $P_2$ :  $(u_1, u_2, u_3, u_4), (v_1, v_2, v_3, v_4)$   

1 Generate two random integer numbers  $I_1, I_2$ , where  

 $0 < I_1 < w - s - b_{\text{upper}}, 0 < I_2 < h - s - b_{\text{upper}}.$  ;  

2 Generate two random integer numbers  $I_3, I_4$ , where  $b_{\text{low}} < I_3, I_4 < b_{\text{upper}}.$  ;  

Return:  $(I_1, I_2, I_1 + s, I_2 + s), (I_1 + I_3, I_2 + I_4, I_1 + I_3 + s, I_2 + I_4 + s)$ 

```

---

### 2.1 Spatial Context Analysis

Algorithm 1 describes the algorithm of `getOverlapPatches`. Figure 3 shows the heatmaps of the per-pixel self-entropy on Cityscapes and BDD dataset against

Method	Model	mIOU	Detection				Detection Adap			
			DAG		Houdini		DAG		Houdini	
			Scene	Kitty	Scene	Kitty	Scene	Kitty	Scene	Kitty
Scale (std=0.5)	DRN (16.4M)	54.5	96%	100%	99%	100%	69%	89%	46%	91%
Scale (std=3.0)			100%	100%	100%	100%	31%	89%	1%	48%
Scale (std=5.0)			100%	100%	100%	100%	8%	84%	0%	36%
Scale (std=0.5)	DLA (18.1M)	46.29	96%	88%	99%	99%	89%	90%	80%	58%
Scale (std=3.0)			100%	100%	100%	100%	66%	88%	11%	26%
Scale (std=5.0)			98%	100%	99%	100%	32%	78%	2%	12%
Spatial (K=1)	DRN (16.4M)	54.5	98%	100%	99%	99%	89%	99%	89%	99%
Spatial (K=5)			100%	100%	100%	100%	100%	100%	100%	100%
Spatial (K=10)			100%	100%	100%	100%	100%	100%	100%	100%
Spatial (K=50)			100%	100%	100%	100%	99%	100%	99%	100%
Spatial (K=1)	DLA (18.1M)	46.29	98%	99%	95%	95%	96%	99%	98%	95%
Spatial (K=5)			100%	100%	98%	98%	99%	100%	99%	96%
Spatial (K=10)			100%	100%	99%	99%	99%	100%	99%	96%
Spatial (K=50)			100%	100%	99%	99%	100%	100%	99%	93%

Table 1: Detection results (AUC) of image spatial (Spatial) and scale consistency (Scale) based methods on BDD dataset. The number in parentheses of the “Model” shows the number of parameters for the target mode, and “mIOU” shows the performance of segmentation model on pristine data. We color all the AUC less than 80% with red.

DRN and DLA models. It is clearly shown that the adversarial instances have higher entropy than benign ones. Table 1 shows that the detection results (AUC) based on spatial consistency method with fix patch size. It demonstrates that the spatial consistency information can help to detect adversarial examples with AUC nearly 100% on BDD dataset. Table 4 5 show the results on additional targets on Cityscapes and BDD datasets. Table 2 3 show the detection results (AUC) based on spatial consistency method with random patch size. They show that random patch sizes achieve the similar detection result.

## 2.2 Scale Consistency Analysis

We applied image scaling to the adversarial examples generated by Houdini [1] and DAG [3] on Cityscapes and BDD datasets against DRN and DLA models. The result shows in Figure 4. We can find the same phenomenon that when we applying Gaussian blurring with high std (3 and 4), adversarial perturbation is harmed and segmentation result are no longer adversarial targets.

Table 1 shows that the method based on image scale information can achieve similarly AUC compared with spatial consistency based method on BDD.

Method (Spatial)	Model	mIOU	Detection				Detection Adap			
			DAG		Houdini		DAG		Houdini	
			Pure	Kitty	Pure	Kitty	Pure	Kitty	Pure	Kitty
K=1	DRN (16.4M)	66.7	91 ± 0.1%	88 ± 0.1%	91 ± 0.3%	90 ± 0.1%	97 ± 0%	92 ± 0.1%	90 ± 2.5%	93 ± 0.1%
K=5			99 ± 0.1%	99 ± 0.1%	100 ± 0.0%	99 ± 0.1%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=10			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=50			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=1	DLA (18.1M)	74.5	96 ± 0.1%	98 ± 0.1%	96 ± 0.1%	96 ± 0.1%	99 ± 0.3%	99 ± 0.1%	98 ± 0.4%	99 ± 0.1%
K=5			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=10			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=50			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%

Table 2: Detection results (AUC) of image spatial (Spatial) based method with random patch size on Cityscapes dataset.

Method (Spatial)	Model	mIOU	Detection				Detection Adap			
			DAG		Houdini		DAG		Houdini	
			Pure	Kitty	Pure	Kitty	Pure	Kitty	Pure	Kitty
K=1	DRN (16.4M)	54.5	91 ± 0.1%	88 ± 0.1%	91 ± 0.3%	90 ± 0.1%	97 ± 0%	92 ± 0.1%	90 ± 2.5%	93 ± 0.1%
K=5			99 ± 0.1%	99 ± 0.1%	100 ± 0.0%	99 ± 0.1%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=10			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=50			100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%
K=1	DLA (18.1M)	46.29	98 ± 0%	96 ± 0.1%	95 ± 0.1%	92 ± 0%	98 ± 0.1%	94 ± 0%	97 ± 0.1%	90 ± 0%
K=5			100 ± 0%	99 ± 0%	99 ± 0%	98 ± 0%	99 ± 0%	98 ± 0%	99 ± 0%	90 ± 0%
K=10			100 ± 0%	100 ± 0%	100 ± 0%	98 ± 0%	100 ± 0%	99 ± 0%	100 ± 0%	96 ± 0%
K=50			100 ± 0%	100 ± 0%	100 ± 0%	99 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%	100 ± 0%

Table 3: Detection results (AUC) of image spatial (Spatial) based method with random patch size on BDD dataset.

Method (Spatial)	Model	mIOU	Detection				Detection Adap			
			DAG		Houdini		DAG		Houdini	
			ECCV	Remap	Strip	ECCV	Remap	Strip	ECCV	Remap
K=1	DRN (16.4M)	66.7	93%	91%	91%	91%	91%	91%	90%	92%
K=5			99%	100%	99%	99%	100%	99%	100%	100%
K=10			100%	100%	100%	100%	100%	100%	100%	100%
K=50			100%	100%	100%	100%	100%	100%	100%	100%
K=1	DLA (18.1M)	74.5	96%	99%	97%	95%	97%	96%	99%	99%
K=5			100%	100%	100%	100%	100%	100%	100%	100%
K=10			100%	100%	100%	100%	100%	100%	100%	100%
K=50			100%	100%	100%	100%	100%	100%	100%	100%

Table 4: Detection results (AUC) of spatial consistency (Spatial) based method on Cityscapes dataset for additional targets.

Method (Spatial)	Model	mIOU	Detection				Detection Adap			
			DAG		Houdini		DAG		Houdini	
			ECCV	Remap	Strip	ECCV	Remap	Strip	ECCV	Remap
K=1	DRN (16.4M)	54.5	99%	99%	99%	99%	99%	99%	99%	98%
K=5			100%	100%	100%	100%	100%	100%	100%	98%
K=10			100%	100%	100%	100%	100%	100%	100%	99%
K=50			100%	100%	100%	100%	100%	100%	100%	97%
K=1	DLA (18.1M)	46.29	99%	99%	99%	98%	97%	98%	99%	97%
K=5			100%	100%	100%	100%	100%	100%	100%	99%
K=10			100%	100%	100%	100%	100%	100%	100%	99%
K=50			100%	100%	100%	100%	100%	100%	100%	99%

Table 5: Detection results (AUC) of spatial consistency (Spatial) based method on BDD dataset for additional targets.

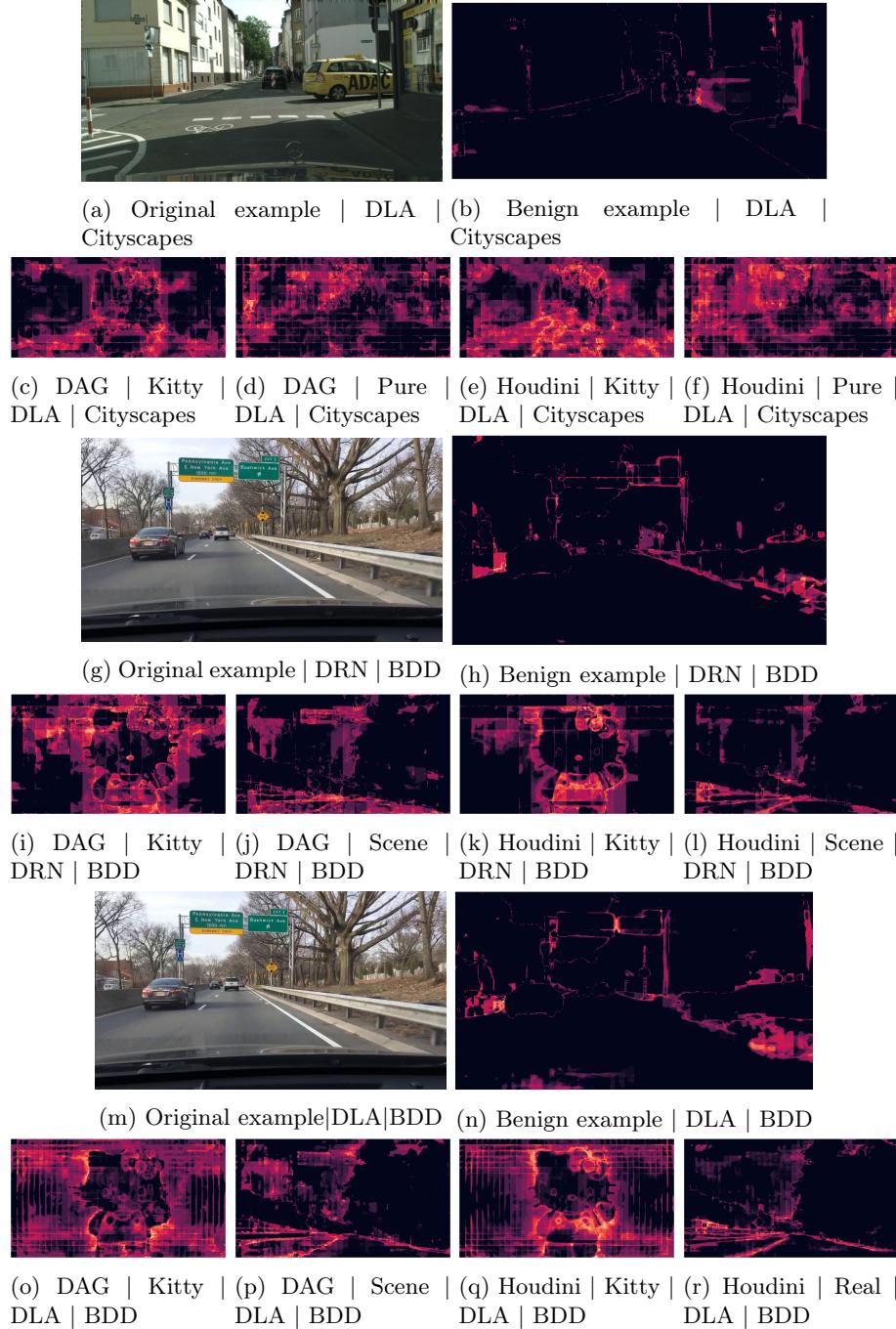
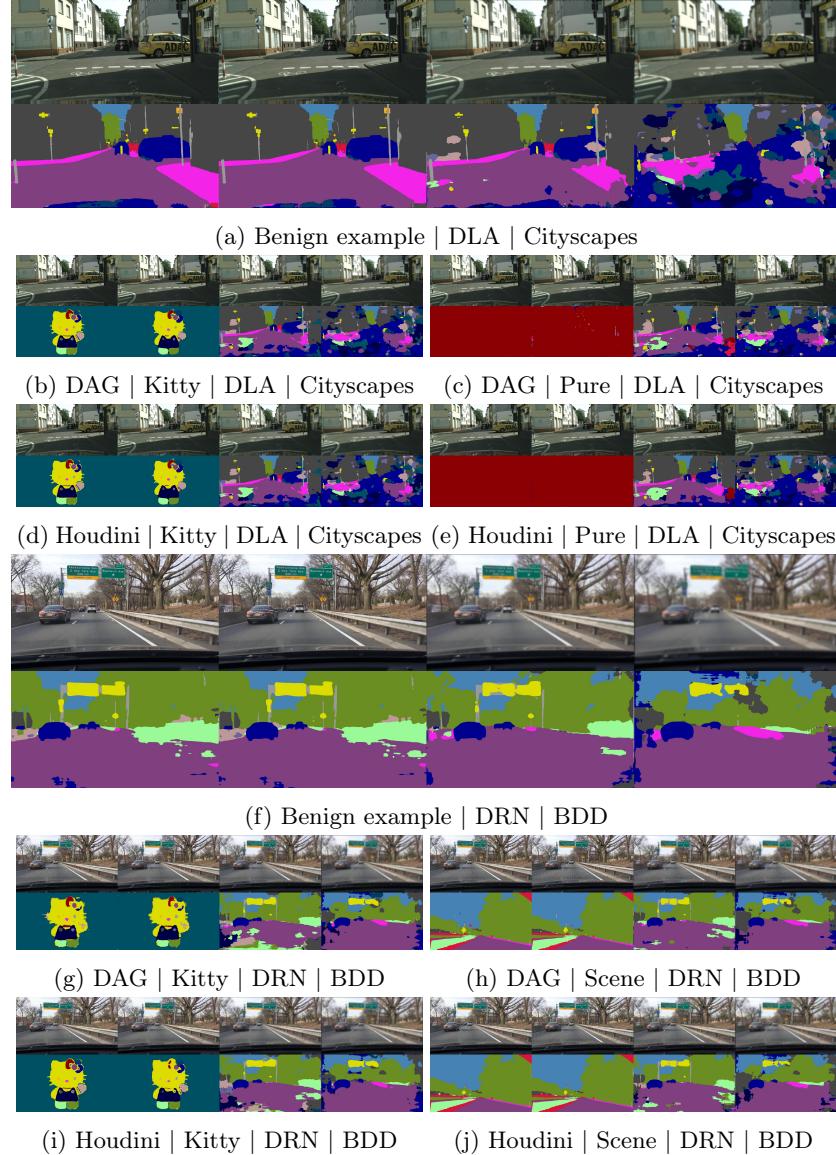


Fig. 3: Heatmap of per-pixel self-entropy. (a), (b), (g), (h), (m) and (n) show benign images and its corresponding per-pixel self-entropy heatmaps. We use the format “examples | attack model | dataset” to label them. For the rest, we use the format “attack method | target label | attack model | dataset” to label each subcaption.



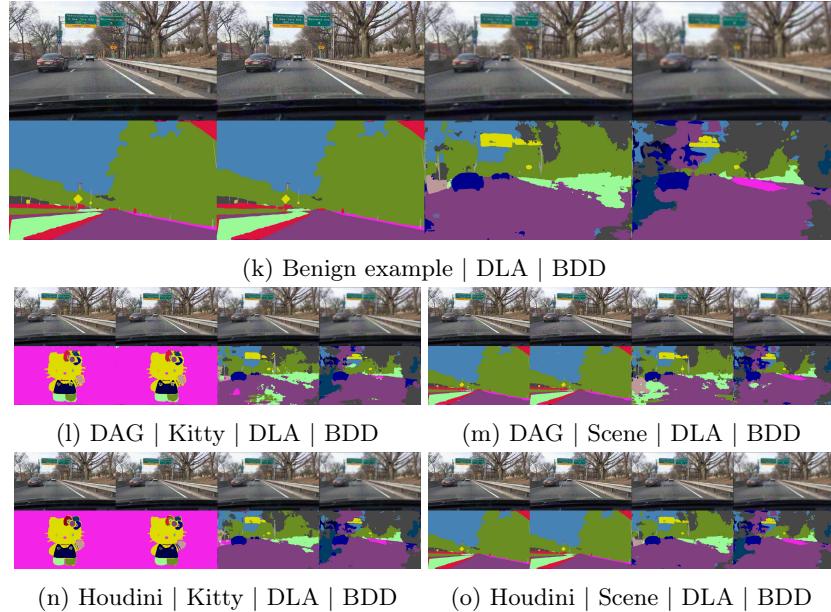
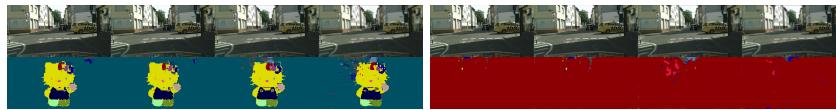


Fig. 4: Examples of images and corresponding segmentation results before/after image scaling. For each subfigure, the first column shows benign/adversarial images, while the following columns represent images after scaling by applying Gaussian kernel with std as 0.5, 3, and 5, respectively. (a),(f) and (k) show benign images before/after image scaling and the corresponding segmentation results and we use the format “example | attack model | dataset” to identify the corresponding model and dataset; (b)-(e), (g)-(j) and (l)-(o) present similar results for adversarial images and we use the format “attack method | target label | attack model | dataset” to label the settings of each image.



(a) Benign example | DRN | Cityscapes



(b) DAG | Kitty | DRN | Cityscapes

(c) DAG | Pure | DRN | Cityscapes

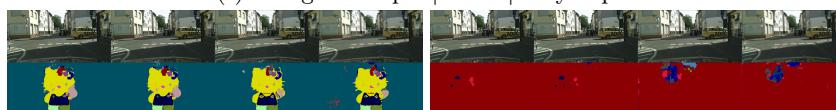


(e) Houdini | Pure | DRN | Cityscapes

(d) Houdini | Kitty | DRN | Cityscapes



(f) Benign example | DLA | Cityscapes



(g) DAG | Kitty | DLA | Cityscapes

(h) DAG | Pure | DLA | Cityscapes



(i) Houdini | Kitty | DLA | Cityscapes (j) Houdini | Pure | DLA | Cityscapes



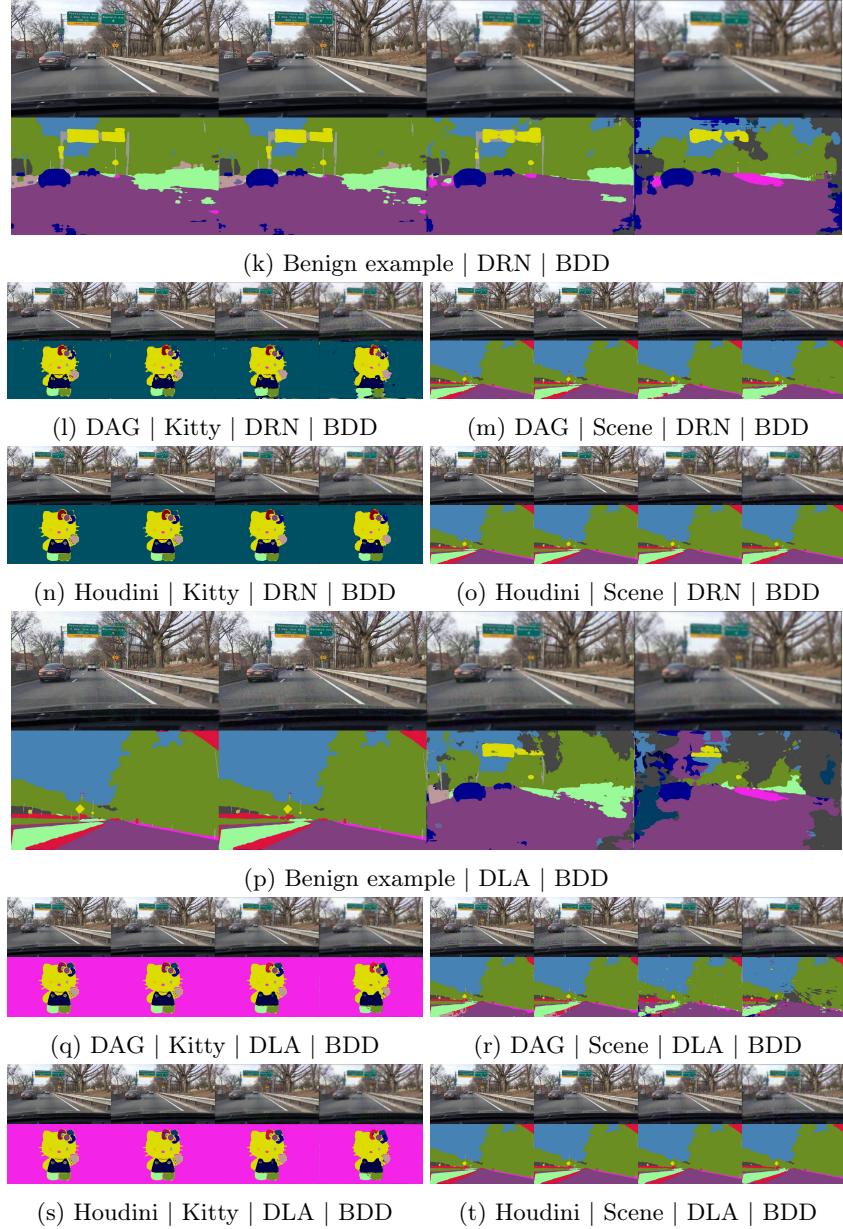


Fig. 5: Examples of images and corresponding segmentation results for adaptive attack against image scaling. For each subfigure, the first column shows benign/adversarial images, while the following columns show images after scaling by applying Gaussian kernel with std as 0.5, 3, and 5, respectively. (a), (f), (k) and (p) show benign images before/after image scaling and the corresponding segmentation results and The format “example | attack model | dataset” uses to identify the corresponding model and dataset; (b)-(e), (g)-(j), (l)-(o) and (q)-(t) present similar results for adaptive adversarial images and we describe by the format “attack method | target label | attack model | dataset”.

### 2.3 Adaptive Attack Evaluation

---

**Algorithm 2:** DAG adaptive attack against spatial consistency method

---

```

input: Input image  $\mathbf{X}$  ;
Number of attack patches  $K$ ;
Patch size  $s$ ;
Segmentation model  $f$ ;
 $L^2$  bound  $\mathbf{b}$ ;
Recognition targets  $\mathcal{T} = \{t_1, t_2, \dots, t_N\}$  ;
Adversarial label set  $\mathbf{Y} = \{Y_{t_1}, Y_{t_2}, \dots, Y_{t_N}\}$ ;
Maximal iteration  $M_0$  ;
output: Adversarial perturbation  $\mathbf{r}$ ;

Initialization :  $\mathbf{X}_0 \leftarrow \mathbf{X}$ ,  $\mathbf{r} \leftarrow 0$ ,  $m \leftarrow 0$ ,  $\mathcal{T}_0 \leftarrow \mathbf{Y}$ ,  $w = \mathbf{X}.width$ ,  $h = \mathbf{X}.height$ ;
1 while  $m < M_0$  &  $\text{l2norm}(\mathbf{r}) < \mathbf{b}$  do
2    $\mathcal{T}_m = \{t_n | \text{argmax}_c \{f_c(\mathbf{X}_m, t_n)\} \neq Y_{t_n}\}$ ;
3    $\mathbf{r}_m \leftarrow \sum_{t_n \in \mathcal{T}_m} \nabla_{\mathbf{x}_m} f_{Y_{t_n}}(\mathbf{X}_m, t_n)$  ;
      /* Attack C random patches */;
4 for  $k \leftarrow 0$  to  $K$  do
5   Generate two random integers  $I_1, I_2$  where  $0 < I_1 < w - s$  and
         $0 < I_2 < h - s$ ;
6    $P_k = \mathbf{X}[I_1 : I_1 + s, I_2 : I_2 + s]$ ;
7    $\mathcal{T}_k = \{t_{n'} | \text{argmax}_c \{f_c(P_k, t_{n'})\} \neq Y_{t_{n'}} \& t_{n'} \in \mathcal{T}_0\}$ ;
8    $\mathbf{r}_m \leftarrow \sum_{t_{n'} \in \mathcal{T}_k} \nabla_{P_k} f_{Y_{t_{n'}}}(P_k, t_{n'})$ 
9 end
10   $\mathbf{r}'_m \leftarrow \frac{\gamma}{\|\mathbf{r}_m\|_\infty} \mathbf{r}_m$ ;
11   $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{r}'_m$ ;
12   $\mathbf{X}_{m+1} \leftarrow \mathbf{X}_m + \mathbf{r}'_m$ ;
13   $m \leftarrow m + 1$ ;
14 end

Return:  $\mathbf{r}$ 

```

---

Here we illustrate the adaptive attack algorithm based on DAG and Houdini against spatial consistency method in Algorithm 2 and Algorithm 3. Instead of only attacking the benign image, the adaptive attack here will randomly pick some patches and attack them with the whole image together.

Let  $\mathbf{X}$  be an image and  $K$  is the number of patches selected from  $\mathbf{X}$  to perform adaptive attack. We define  $\mathcal{T} = \{t_1, t_2, \dots, t_N\}$  as the set comprising the coordinates of the recognition target pixels.  $f$  denotes the segmentation network, and we use  $f(\mathbf{X})$  to denote the classification score of the entire image and  $f(\mathbf{X}, t_n)$  to denote the classification score vector at pixel  $n$ .  $\mathbf{Y} = Y_{t_1}, Y_{t_2}, \dots, Y_{t_N}$  denotes the adversarial label set where  $Y_{t_n}$  represents the adversarial label of pixel  $n$ . Given an input tensor  $\mathbf{r} \in \mathbb{R}^{w \times h \times c}$ , the function returns its  $L^2$  norm defined to be  $\|\mathbf{r}\|_2 = \sqrt{\frac{\sum_{i=1}^w \sum_{j=1}^h \sum_{k=1}^c r_{ijk}^2}{w \cdot h \cdot c}}$ . In Algorithm 3, we follow the same

**Algorithm 3:** Houdini adaptive attack against spatial consistency method

---

**input:** Input image  $\mathbf{X}$  ;  
Number of attack patches  $K$ ;  
Patch size  $s$ ;  
Segmentation model  $f$ ;  
 $L^2$  bound  $\mathbf{b}$ ;  
Adversarial target label  $\mathbf{Y}$ ;  
Maximal iteration  $M_0$  ;  
Houdini loss  $\ell_H$ ;

**output:** Adversarial perturbation  $\mathbf{r}$ ;

**Initialization :**  $\mathbf{X}_0 \leftarrow \mathbf{X}, \mathbf{r} \leftarrow 0, m \leftarrow 0, \mathcal{T} \leftarrow \mathbf{Y}, w \leftarrow \mathbf{X}.width, h \leftarrow \mathbf{X}.height$ ;

1 **while**  $m < M_0$  **do**

2    $\mathbf{r}_m \leftarrow 0$ ;  
/\* get the gradient of the perturbation from the objective \*/;

3    $\mathbf{r}_m \leftarrow \mathbf{r}_m + \nabla_{\mathbf{r}} \ell_H(f(\mathbf{X} + \mathbf{r}), \mathbf{Y})$ ;

4    $\mathbf{r}_m \leftarrow \mathbf{r}_m + \nabla_{\mathbf{r}} \max(\text{l2norm}(\mathbf{r}) - b, 0)$  ;

5   **for**  $k \leftarrow 0$  **to**  $K$  **do**

6     Generate two random integer numbers  $I_1, I_2$  where  
 $0 < I_1 < w - s, 0 < I_2 < h - s$ ;

7      $\mathbf{r}_m \leftarrow \mathbf{r}_m + \nabla_{\mathbf{r}} \ell_H(f((\mathbf{X} + \mathbf{r})[i : i + s, j : j + s]), \mathbf{Y}[i : i + s, j : j + s])$ ;

8   **end**

9    $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{r}_m$ ;

10    $m \leftarrow m + 1$ ;

11 **end**

**Return:**  $\mathbf{r}$

---

definition of Houdini loss  $\ell_H$  as proposed in the work [1]. We set the maximal number of iteration to be approximately 300<sup>1</sup> in all settings. We set  $L^2$  bound to be 0.06 for simplicity.

The detection results in term of AUC of the spatial consistency based method against such adaptive attacks on BDD and Cityscapes are shown in Table 1 2 3 4 5. Even against such strong adaptive attacks, the spatial consistency based method can still achieve nearly 100% AUC. Figure 6 shows the confusion matrix of detection result for adversaries and detection method choosing various  $K$ . It is clear that when we choose  $K = 50$ , it is already sufficient to detect sophisticated attacks on Cityscapes dataset against DLA model and can also achieved 100% detection rate with false positive rate 95% on BDD.

The detection results of the image scaling based method against adaptive attacks on BDD are shown in Table 1. For image scaling based detection method, it can be easily attacked (AUC drops drastically). Figure 5 shows the qualitative results. It is obvious that even under Gaussian kernel with large standard deviation, the adversarial example can still be fooled into predicting different ma-

---

<sup>1</sup> 300 is approximately three times the average number of iterations in non-adaptive attack.

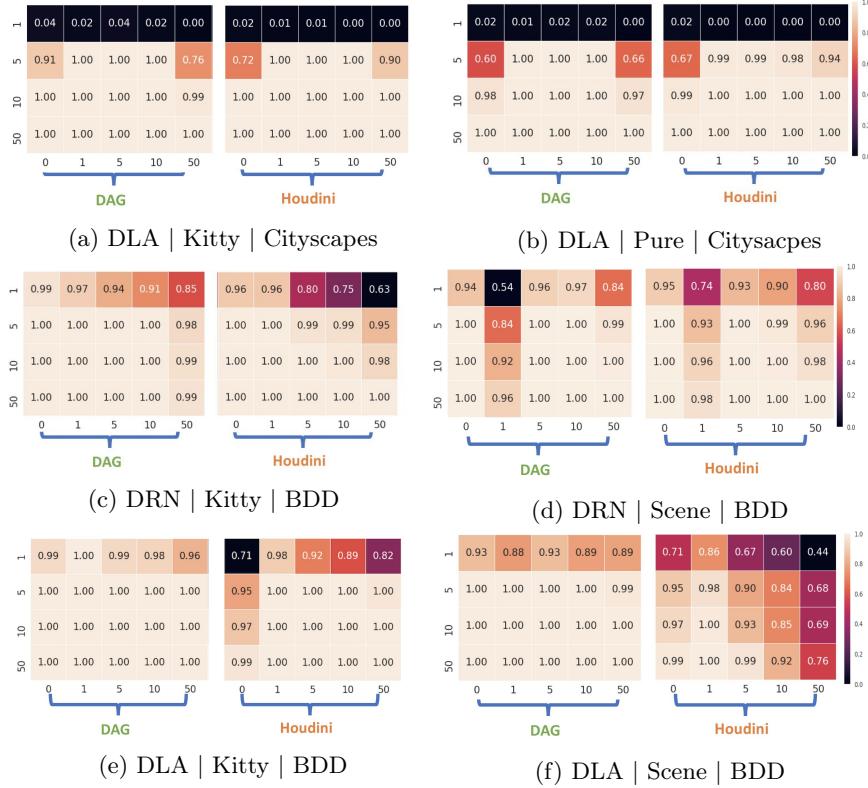


Fig. 6: Detection performance of spatial consistency based method against adaptive attack with different  $K$ . We use the format “attack model | target label | dataset” to label the settings for each figure. X-axis indicates the number of patches selected to perform the adaptive attack (0 means regular attack). Y-axis indicates the number of overlapping regions selected during detection. We select the minimal mIOU from benign patches as our threshold on Cityscapes, and the one which guarantees accuracy as above 95% on benign images for BDD.

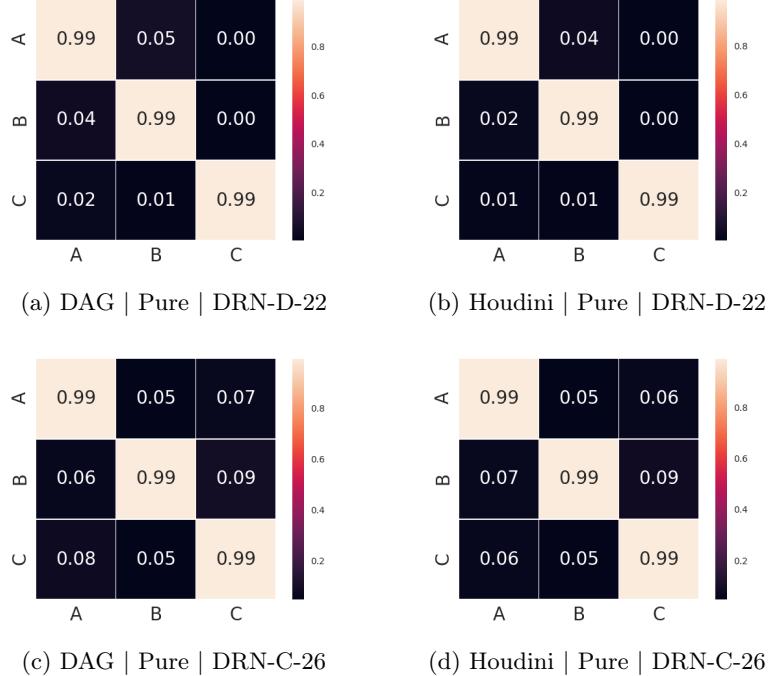
licious targets (“Kitty”, “Pure”, “Scene”) on Cityscapes and BDD dataset against DRN and DLA models.

### 3 Additional Results for Transferability Analysis

We present additional results for transferability analysis in Figure 7 to Figure 11.

Figure 7 to Figure 8 show the transferability analysis results for segmentation models. We report pixel-wise attack success rate for the pure target and normalized mIoU after eliminating K classes with the lowest IoU values for other targets. We set K to be 13 for CityScapes dataset and 5 for BDD dataset. Additional qualitative results are presented in Figure 9 to Figure 10.

Fig. 11 shows the transferability experiments results for classification models under targeted attack. The adversarial images are generated using iterative FGSM method from MNIST and CIFAR10 datasets. The caption of each sub-figure indicates the dataset and the attacked classification model.



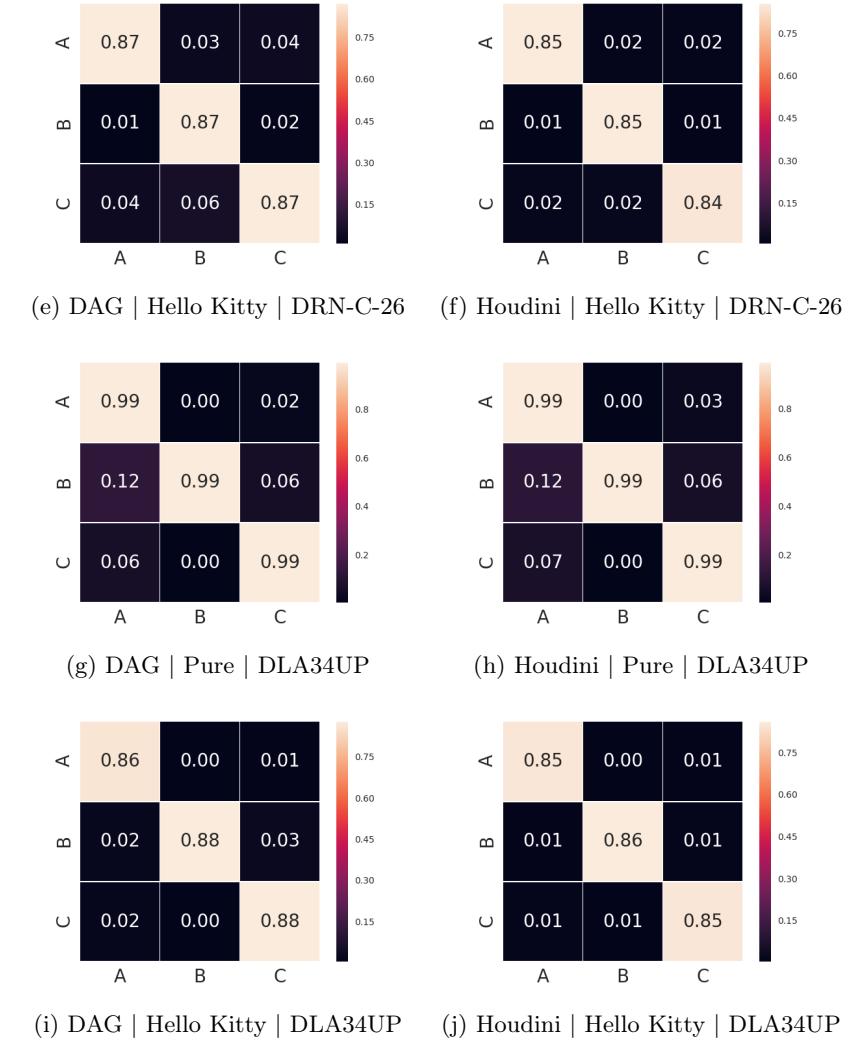
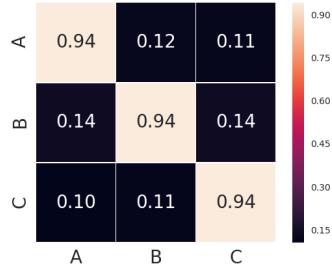
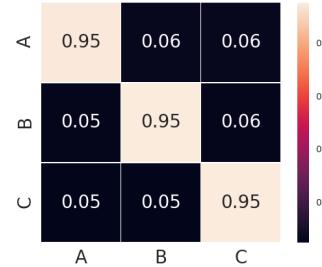


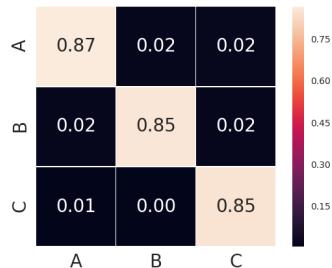
Fig. 7: Transferability analysis on CityScapes dataset: cell  $(i, j)$  shows the normalized mIoU value or pixel-wise attack success rate of adversarial examples generated against model  $j$  and evaluate on model  $i$ . Model A,B,C have the same architecture (DRN-C-26 or DLA34UP) with different initialization. We use format “attack method | attack target | model” to denote the caption of each sub-figure.



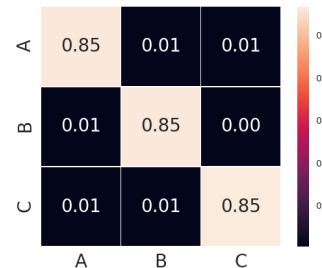
(a) DAG | Scene | DRN-D-22



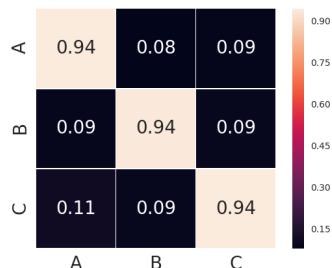
(b) Houdini | Scene | DRN-D-22



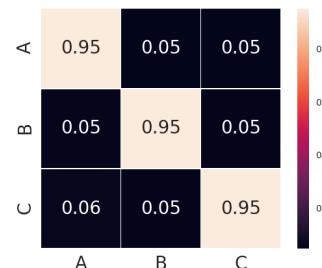
(c) DAG | Hello Kitty | DRN-D-22



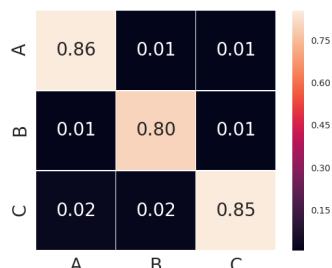
(d) Houdini | Hello Kitty | DRN-D-22



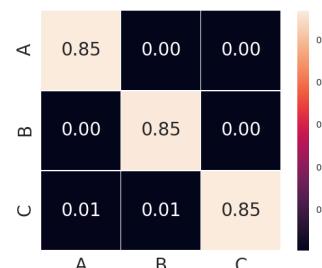
(e) DAG | Scene | DRN-C-26



(f) Houdini | Scene | DRN-C-26



(g) DAG | Hello Kitty | DRN-C-26



(h) Houdini | Hello Kitty | DRN-C-26

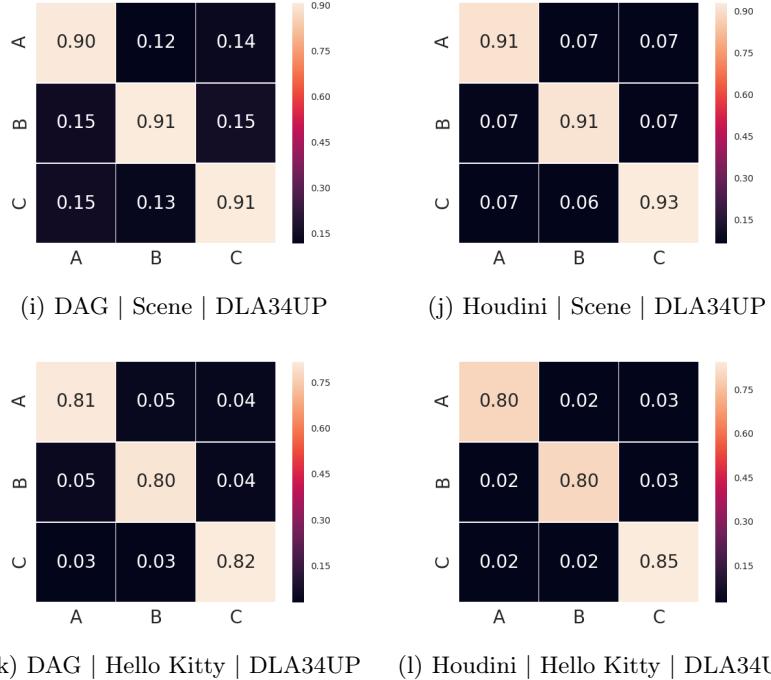
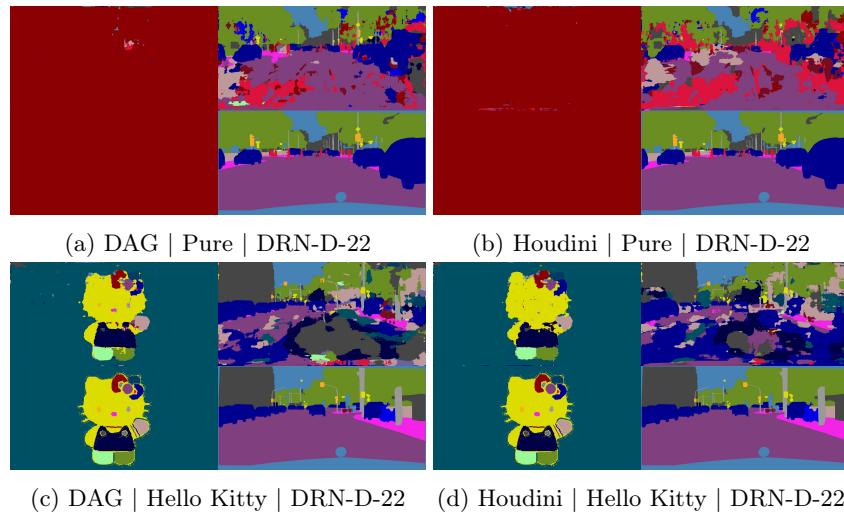


Fig. 8: Transferability analysis on BDD dataset.



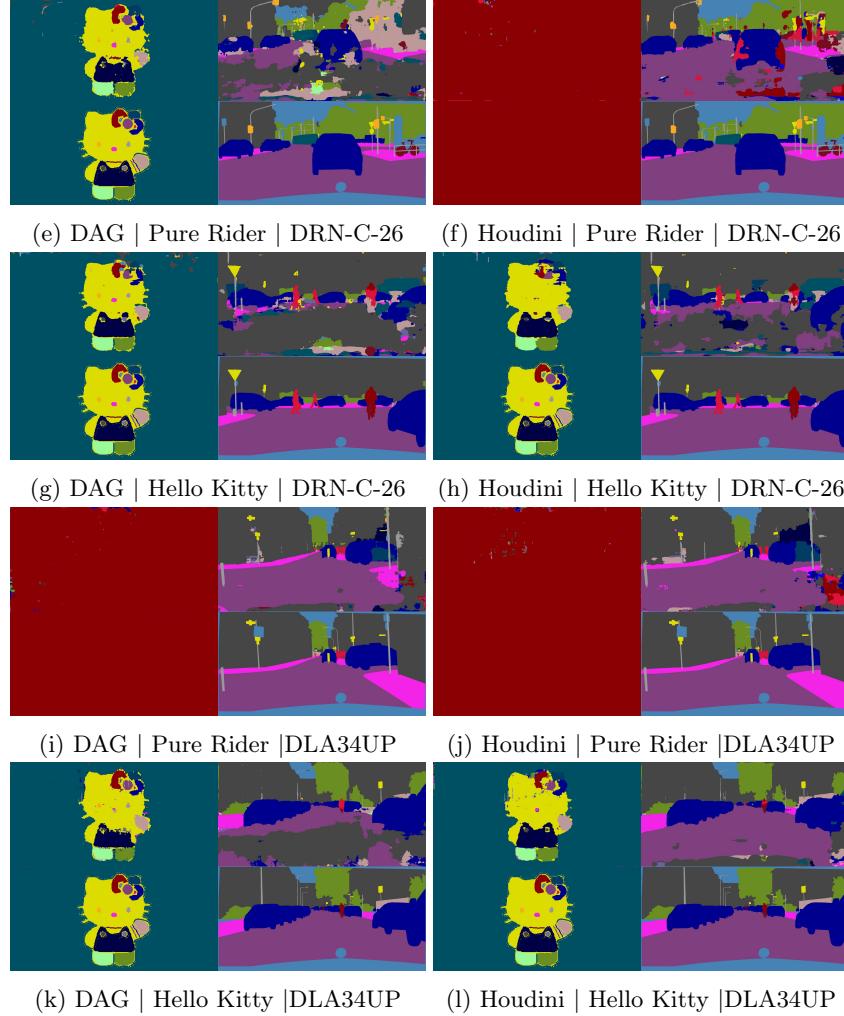
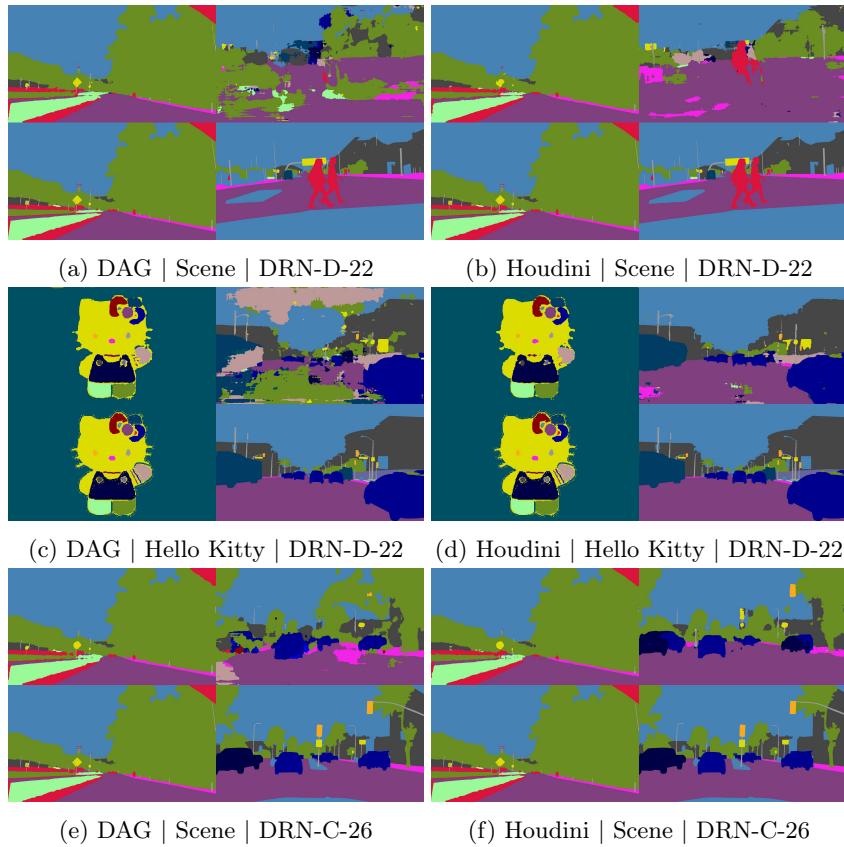


Fig. 9: Transferability visualization on CityScapes dataset. In each sub-figure, the first row presents the segmentation results of adversarial example on model A (targeted model) and model B. The second row shows the adversarial target and the ground truth. We use format “attack method | attack target | model ” to denote the caption of each sub-figure.



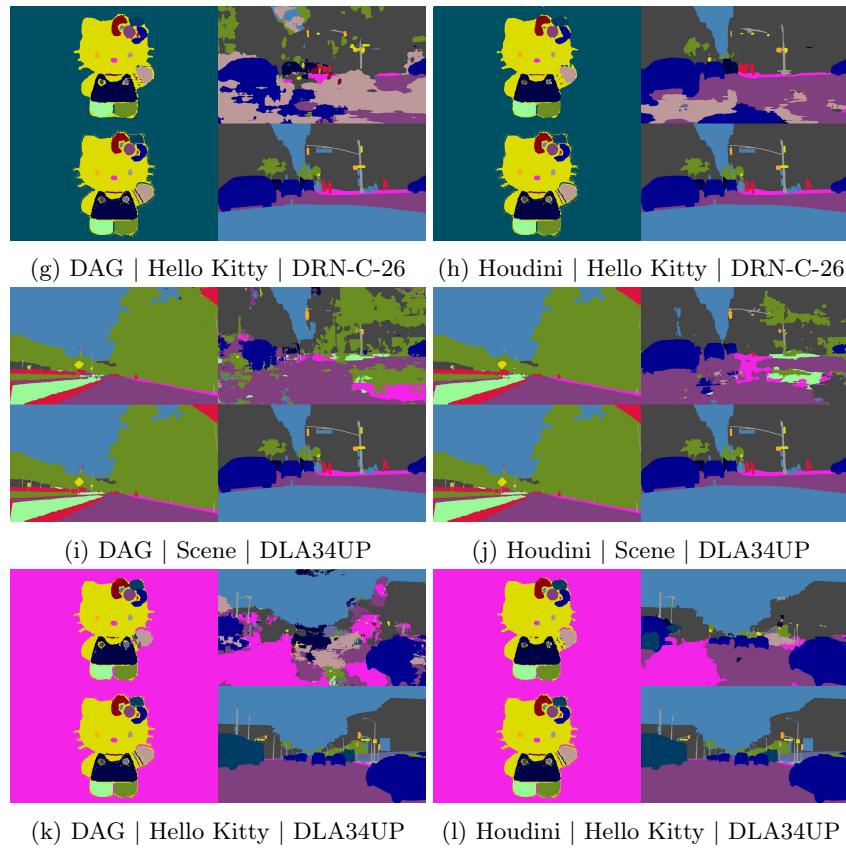


Fig. 10: Transferability visualization on BDD dataset.

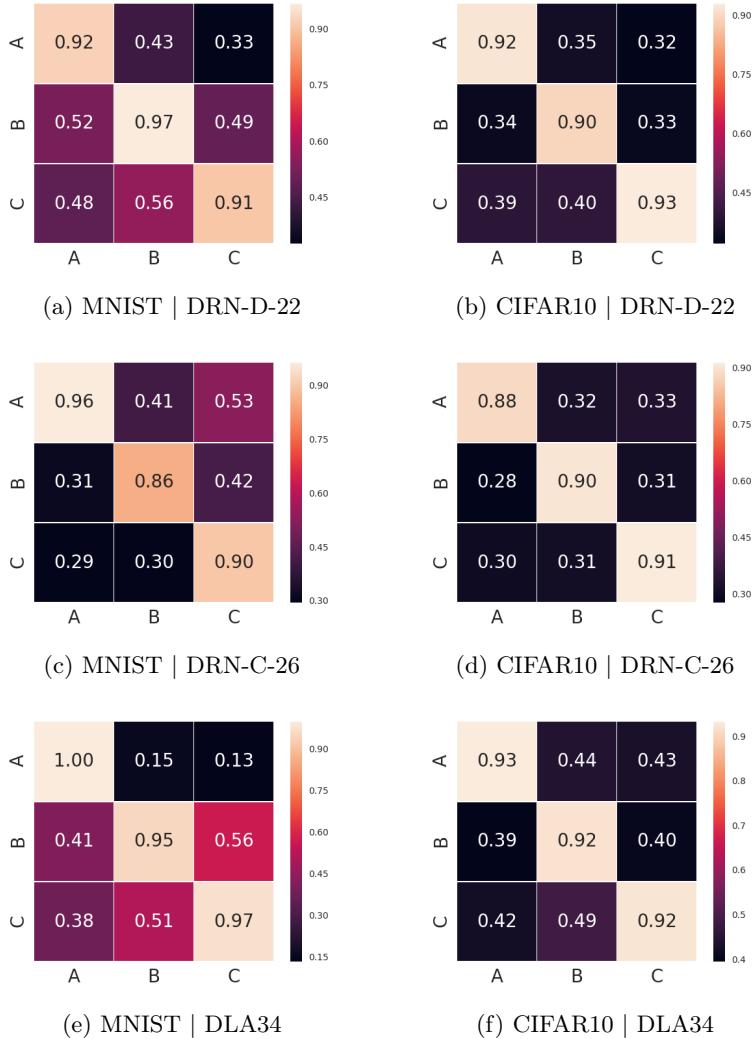


Fig. 11: Transferability analysis for classification models: cell  $(i, j)$  shows the attack success rate of the adversarial examples generated against Model  $j$  and evaluate on Model  $i$  under targeted attack setting. Model A,B,C are model with the same architecture (DRN-D-22, DRN-C-26 or DLA34UP) and different initialization. All the adversarial examples are generated using fast iterative gradient sign method. The caption of each sub-figure bear the “dataset | model”.

## References

1. Cisse, M., Adi, Y., Neverova, N., Keshet, J.: Houdini: Fooling deep structured prediction models. arXiv preprint arXiv:1707.05373 (2017)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
3. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: International Conference on Computer Vision. IEEE (2017)
4. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Computer Vision and Pattern Recognition (CVPR) (2017)
5. Yu, F., Wang, D., Darrell, T.: Deep layer aggregation. arXiv preprint arXiv:1707.06484 (2017)