Supplementary Material for "D^{2}ADA: Dynamic Density-aware Active Domain Adaptation for Semantic Segmentation"

Tsung-Han Wu[1], Yi-Syuan Liou[1], Shao-Ji Yuan[1], Hsin-Ying Lee[1]
Tung-I Chen[1], Kuan-Chih Huang[1], and Winston H. Hsu[1,2]

[1]National Taiwan University [2]Mobile Drive Technology

1 Proof

We would like to prove Eq. 4 in Sec. 3.2 of the main paper.

$$
\begin{aligned}
&D_{\mathrm{KL}}(p_T(c,z) \;||\; p_S(c,z)) \\
&\quad = \mathbb{E}_{p_T(c,z)}[\log p_T(c,z) - \log p_S(c,z)] \\
&\quad = \mathbb{E}_{p_T(c,z)}[\log p_T(c) + \log p_T(z|c) - \log p_S(c) - \log p_S(z|c)] \\
&\quad = \mathbb{E}_{p_T(c,z)}[\log p_T(c) - \log p_S(c)] \\
&\qquad\qquad + \mathbb{E}_{p_T(c,z)}[\log p_T(z|c) - \log p_S(z|c)] \\
&\quad = \mathbb{E}_{p_T(c)}[\log p_T(c) - \log p_S(c)] \\
&\qquad\qquad + \mathbb{E}_{p_T(c)}[\mathbb{E}_{p_T(z|c)}[\log p_T(z|c) - \log p_S(z|c)]] \\
&\quad = D_{\mathrm{KL}}(p_T(c) \;||\; p_S(c)) + \mathbb{E}_{p_T(c)}[D_{\mathrm{KL}}(p_T(z|c) \;||\; p_S(z|c))]
\end{aligned}
$$

2 Implementation Details

We conducted all experiments on an 8-core CPU personal computer with an NVIDIA RTX3090 GPU. The following section elaborates on the implementation details of UDA warm-up, density-aware selection, dynamic scheduling policy, and network fine-tuning. Note that the following symbols are identical to those in Sec. 3 of the main paper. The whole pipeline is presented in Algorithm 1.

2.1 UDA warm-up

For the two tasks, GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes, we utilized a conventional UDA method [10] to train an initial model. For both DeepLabV2 and DeepLabV3+ network backbones, we apply the SGD optimizer with an initial learning rate of 2.5e-4 and a decay rate of 0.9. Following [10], we warm up the network with adversarial training for about 100k steps (about 8 hours).

2.2 Density-aware selection

As mentioned in Sec. 3.2 of the main paper, we utilized a set of Gaussian Mixture Models (GMMs) to model the conditional probability distributions of the two domains. The domain density $p_S(z|c)$ and $p_T(z|c)$ are the likelihood of sampling the region feature z from the source domain and target domain given the predicted category c respectively.

The feature z is a vector of 256 dimensions. For the DeepLabV3+ network backbone, the feature is extracted before the final linear classification layer. For the DeepLabV2 network backbone, we slightly modify the network to make the output of Atrous Spatial Pyramid Pooling (ASPP) as a 256-dimensional feature vector and add the final classification layer after ASPP.

In our implementation, the number of mixtures in GMM is proportional to the number of regions of the category and is clipped in the range of 1 to 10. The process of constructing GMMs can be efficiently completed by offline and parallel execution, which takes about 0.009 seconds per region with four parallel processes. We believe the density estimator could be replaced by other methods and is worth further investigation, which is beyond the scope of this paper.

2.3 Dynamic Scheduling Policy

As explained in Sec. 3.3 of the main paper, we use two hyper-parameters, α and β, to dynamically schedule the labeling budgets of different active selection methods. For the GTA $\rightarrow$ Cityscapes task, we set $\alpha = 1, \beta = 1$. For the SYNTHIA $\rightarrow$ Cityscapes task, we set $\alpha = 0.5, \beta = 1$. The selection of (α, β) is not difficult: (1) a larger α indicates a larger domain gap and (2) simply setting β = 1 (half-decay scheduling) performs well on both tasks. The selection of α and β (half-decay) are discussed in Sec. 4.2 and Sec. 4.3 respectively. The computational cost for this step, including the labeling budget decision and uncertainty selection, consumes about 0.001 for each region.

2.4 Network Fine-tuning

After the active selection step, we acquire ground truth labels of the top-ranked regions in D_T^U and move them to D_T^L. Then, fine-tune the model on $D_S \cup D_T^L$ using cross-entropy loss in a supervised manner. For the fine-tuning step, we utilized the SGD optimizer with an initial learning rate of 2.5e-4 and a decay rate of 0.9. The fine-tuning step takes about 8 hours.

3 Baseline Active Learning Methods

We describe the implementation of 8 region-based active learning baselines used in our experiments. In the following section, we let R denote a region with N pixels within it, and θ denotes the fixed trained deep learning network.

Algorithm 1: The pipeline of **D²ADA**

Input: data pool: $(D_S, D_T = \{D_T^L, D_T^U\}$, where $D_T^L = \emptyset)$, hyper-parameters: (α, β), max_active_iterations = N, labeling budget for each active selection round: B

Output: the output model h_θ

Warm-up: $h_\theta \leftarrow (D_S, D_T^U, D_T^L)$ according to [10]

for $n \leftarrow 0$ **to** N **do**

```
// Construct source and target density estimators
```

$Z_S, C_S \leftarrow h_\theta(D_S)$

$Z_T, C_T \leftarrow h_\theta(D_T)$

$GMM_{\text{src}} \leftarrow$ ConstructGMMs(Z_S, C_S)

$GMM_{\text{trg}} \leftarrow$ ConstructGMMs(Z_T, C_T)

```
// Calculate the region importance metric π
```

$R^* \leftarrow$ a new empty list

for each c **in** $\{1, 2, ..., C\}$ **do**

$R_c \leftarrow$ Obtain all regions predicted as category c in D_T^U

for each region **in** R_c **do**

Obtain the domain density d_S, d_T by feeding the regional feature z and the predicted category c to GMM_{src} and GMM_{trg}

Calculate the importance score π according to Eq. 2

end

Rank all the regions in R_c in descending order based on the important score; then, append it to R^*

Calculate the categorical KL-divergence $D_{KL}(p_T(z|c) \mid\mid p_S(z|c))$ according to Eq. 5

end

```
// Dynamic Scheduling Policy
```

Determine B_n^u, B_n^d given (α, β, B) according to Eq. 7

```
// Class-Balanced Selection
```

Determine $B^{d,c}$ for each c given the categorical KL-divergence according to Eq. 6

```
// Label Acquisition
```

$S_n^u \leftarrow$ UncertaintySelection$(B_n^u,\ h_\theta, D_T^U)$ according to [11]

$S_n^d \leftarrow$ DensityAwareSelection$(B_n^d,\ R^*)$ `// Pick top-ranked regions based on the categorical budgets` $B^{d,c}$

$X_{\text{active}} \leftarrow S_n^u \bigcup S_n^d$

$Y_{\text{active}} \leftarrow$ Obtain ground-truth labels from the oracle given X_{active}

$D_T^L \leftarrow D_T^L \bigcup (X_{\text{active}}, Y_{\text{active}})$

$D_T^U \leftarrow D_T^U \setminus X_{\text{active}}$

```
// Supervised fine-tuning
```

$h_\theta \leftarrow$ Tune the model on $D_S \bigcup D_T^L$ with cross-entropy loss

end

return h_θ

RAND Randomly select few regions of images in the unlabeled target domain dataset D_T^U for label acquisition.

MAR [11] Wang *et al.* [11] proposed using model softmax margin as the indicator to select informative instances for labeling. Specifically, we produce the score for a region (S_R^{MAR}) by averaging the difference between the two most likely category labels for all pixels within the region, as shown in Eq. 1. After that, we acquire the ground truth labels of few regions with the smallest score, which means the smallest margin, in the unlabeled dataset for each category.

$$S_R^{\text{MAR}} = \frac{1}{N} \sum_{n=1}^{N} P(\hat{y_n^1}|R;\theta) - P(\hat{y_n^2}|R;\theta), \tag{1}$$

where $\hat{y_n^1}$ is the first most probable label category and $\hat{y_n^2}$ is the second most probable label category.

CONF [11] The main concept of the confidence selection strategy is to acquire labels for samples whose prediction has the least confidence [11,12]. As can be observed in Eq. 2, the score for a region (S_R^{CONF}) is produced by averaging the softmax confidence of all pixels within the region. After that, we select a portion of regions with the least confidence score in the unlabeled dataset for label acquisition.

$$S_R^{\text{CONF}} = \frac{1}{N} \sum_{n=1}^{N} P(\hat{y_n^1}|R;\theta), \tag{2}$$

where $\hat{y_n^1}$ is the softmax confidence value of the predicted category label.

ENT [11] In the field of information theory, entropy is a widely used metric to evaluate the information of a probability distribution [7]. The idea of this type of selection strategy is to select regions with the largest entropy for labeling [11]. As shown in Eq. 3, the score for a region (S_R^{ENT}) is formed by averaging the softmax entropy of all pixels in a region. After that, a portion of regions with the largest entropy in the unlabeled dataset is selected for label acquisition.

$$S_R^{\text{ENT}} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{c} [P(y_n^i|R;\theta) \cdot \log[P(y_n^i|R;\theta)]], \tag{3}$$

where c is the number of categories, and $P(y_n^i|R;\theta)$ represents the softmax probability that the model predicts pixel n as class i.

BADGE [1] Ash *et al.* proposed selecting a batch of diverse and uncertain samples for labeling with a designed gradient embedding space. Specifically, the method first calculates the gradient embedding of each sample to indicate its uncertainty and then selects diverse samples for label acquisition with k-means++. In our implementation, we produce the regional gradient embedding by averaging the value of all pixels within it. Then, we follow its original implementation to cluster the regions with the k-means++ algorithm.

ReDAL [13] Wu *et al.* proposed acquiring a batch of diverse point cloud regions with high uncertainty for labeling by entropy, 3D characteristics, and greedy diversity selection. In our implementation, we carefully replaced the 3D characteristics term as detected 2D edges and followed the rest of the algorithm.

AADA [9] Su *et al.* presented the first active domain adaptation approach for image classification. The concept of this method is to leverage the softmax entropy and the domain discriminator to select uncertain samples that are far from the source domain distribution. In our implementation, we calculated the region-level softmax entropy and domain discriminator result and followed the rest of the algorithm.

CLUE [4] Prabhu *et al.* presented another active domain adaptation method by clustering the uncertainty-weighted embeddings. The same, we treated a region as the fundamental label query unit and followed the original implementation.

4 More Experimental Results and Analyses

In this section, we first discuss the effectiveness of UDA warm-up and hyper-parameter selections. Then, we analyze the influence of inaccurately predicted categories. Finally, we report the raw tables of Fig. 3 in the main paper in Tab. 4, 5 and show the per-class performance of our active learning strategy in Tab. 6.

4.1 Effectiveness of UDA Warm-up

Tab. 1 shows the mIoU scores after applying the UDA method [10] as the warm-up step. The result shows that the performance of the UDA method is still far from that of full supervision and our method.

We further investigate the effectiveness of UDA warm-up on the GTA5 $\rightarrow$ Cityscapes task. With 1% target domain labeled regions, our method can reach $64.0 \sim 64.1$ mIoU with and without warm-up. This shows that UDA warm-up plays little role in our improvement. The main reason for using warm-up in our experiments is to follow prior domain adaptation works [3,8,14].

Table 1. mIoU scores of UDA warm-up [10] on the two tasks.

(a) GTA $\rightarrow$ Cityscapes		
	DeepLabV2	DeepLabV3+
mIoU (%)	44.61	45.51
(b) SYNHTIA $\rightarrow$ Cityscapes		
	DeepLabV2	DeepLabV3+
mIoU (%)	39.95	43.04

4.2 Effectiveness of Initial Balance Coefficient

As mentioned in Sec. 3.3 of the main paper, the balance coefficient α is designed to balance between density-aware and the uncertainty-based method at the first active selection round. We investigate the effectiveness of α with the DeepLabV3+ model backbone for the two tasks.

As can be observed in Tab. 2, with the aid of partial or full acquired annotations by our designed density-aware method, models are able to obtain higher mIoU scores compared with only using conventional uncertainty-based method ($\alpha = 0$). For the GTA5 $\rightarrow$ Cityscapes task, only adopting density-aware selection strategy at the beginning achieve the best result; while for the SYNTHIA $\rightarrow$ Cityscapes task, choosing $\alpha = 0.5$ to combine density-aware and uncertainty-based methods obtain the best performance. The experimental results confirm the effectiveness of our density-aware selection in severe domain shift.

Table 2. We report the mIoU scores with different balance coefficients α. We found that using only the uncertainty-based method, *i.e.*, $\alpha = 0$, obtained the worst results among all combinations. The results show that using some or all of the obtained annotations through density-aware selection can improve model performance.

(a) GTA $\rightarrow$ Cityscapes with 1% Target Labels					
α	0	0.25	0.5	0.75	1.0
mIoU (%)	59.57	61.95	63.30	63.73	**64.03**
(b) SYNHTIA $\rightarrow$ Cityscapes with 1% Target Labels					
α	0	0.25	0.5	0.75	1.0
mIoU (%)	61.56	61.98	**62.47**	62.07	61.87

4.3 Effectiveness of Different Scheduling Policies

As discussed in Sec. 3.3 of the main paper, due to the rapid domain shift reduction, we design a dynamic scheduling policy to half decay the labeling budget of the domain exploration and gradually put more emphasis on the uncertainty-based method. Here we discuss the effectiveness of five budget scheduling policies on the GTA5 → Cityscapes task with DeepLabV3+ model backbone, including pure density, pure uncertainty, even distribution, linear decay, and half decay.

We classify these five policies based on the λ value in Eq. 7 in the main paper. Pure density and pure uncertainty refer to $\lambda = 1$ and $\lambda = 0$ respectively. Even distribution means evenly assigning the labeling budgets to our density-aware method and uncertainty-based approach for each active selection round, *i.e.*, $\lambda = 0.5$. Linear decay refers to the linear decrease of the labeling budgets assigned to the density-aware method. In our implementation, the proportion of density-aware selection method is initialized as 1.0 and linearly decreases by 0.2 for each step, *i.e.*, $\lambda = 1.0 - 0.2(n-1)$. Half decay is our budget scheduling policy described in Sec. 3.3 of the main paper, *i.e.*, $\lambda = \alpha \cdot 2^{-\beta(n-1)}, (\alpha, \beta) = (1, 1)$.

As shown in Tab. 3, our half-decay approach obtains the best performance with 1%, 3%, and 5% budgets. Overall, the result suggests that our density-aware technique and traditional uncertainty-based method complement each other to reach better adaptability under our dynamic scheduling policy.

Table 3. We compare different label budget scheduling strategies on the GTA → Cityscapes task. The result shows that our designed half-decay method performs the best among all strategies.

Scheduling Policy	mIoU		
	1%	3%	5%
Pure Density ($\lambda = 1$)	**64.03**	69.38	70.69
Pure Uncertainty ($\lambda = 0$)	59.97	68.79	70.70
Even Distribution ($\lambda = 0.5$)	63.30	69.66	71.15
Linear Decay ($\lambda = 1.0 - 0.2(n-1)$)	**64.03**	69.49	71.14
Our Half Decay ($\lambda = 2^{-(n-1)}$)	**64.03**	**69.86**	**71.25**

4.4 Influence of Inaccurately Predicted Categories

As mentioned in Sec. 3.2 in the main paper, our density-aware selection estimates the domain density with the extracted region features and the corresponding predicted categories. Since the predicted category might be inaccurate, especially in the first stage in the ADA, we conducted an experiment to verify whether our method is robust under this issue.

The result shows that indeed the initially predicted category might indeed be inaccurate, but our method can recall more of these mispredicted data for labeling. According to the statistics, about 56% of "target bus regions" were predicted as other classes by the initial model. Still, our method recalled 25% of these mispredicted regions to re-label, while the uncertainty-based methods could only recall 4% of them. Overall, we show our approach is effective even with noisy initial labels in this experiment.

Table 4. Results of mIoU performance (%) on GTA [5] → Cityscapes [2] with DeepLabV3+ network backbone.

% Target Labels	RAND	MAR	CONF	ENT	BADGE	ReDAL	AADA	CLUE	$\mathbf{D^2ADA}$
1	58.81	57.95	59.90	59.57	62.67	61.97	55.84	60.13	**64.03**
2	61.11	64.33	65.72	66.08	65.62	65.93	61.71	62.96	**67.65**
3	62.23	67.08	68.83	68.79	67.73	67.12	64.93	64.99	**69.86**
4	63.62	68.47	69.55	69.83	68.66	68.55	65.74	65.88	**70.66**
5	63.50	69.44	70.70	70.70	70.50	68.30	66.01	67.13	**71.25**

Table 5. Results of 16-classes mIoU performance (%) on SYNTHIA [6] → Cityscapes [2] with DeepLabV3+ network backbone.

% Target Labels	RAND	MAR	CONF	ENT	BADGE	ReDAL	AADA	CLUE	$\mathbf{D^2ADA}$
1	59.05	60.67	61.94	61.56	61.64	61.08	54.16	60.17	**62.47**
2	60.96	67.23	68.20	67.75	66.29	66.58	59.84	64.71	**69.35**
3	63.65	69.68	69.70	70.23	69.18	69.30	63.12	66.50	**71.01**
4	65.53	70.43	71.17	71.28	70.26	69.97	66.09	67.46	**72.40**
5	66.09	71.37	71.50	71.76	71.08	71.01	67.03	68.37	**72.74**

Table 6. Complete experimental results of our proposed $\mathbf{D^2ADA}$ on (a) GTA5 → Cityscapes and (b) SYNTHIA → Cityscapes with different percentage of acquired target labels.

(a) GTA5 → Cityscapes

	% Target Labels	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
$\mathbf{D^2ADA}$ (DeepLabV2)	1%	93.63	59.90	86.92	39.59	40.95	44.04	51.78	53.87	88.34	45.30	86.60	71.09	46.82	89.81	57.58	69.71	58.65	52.54	68.45	63.45
	2%	95.34	69.08	88.54	48.24	49.26	45.21	54.41	59.61	89.00	52.73	90.64	73.09	50.23	91.20	69.36	73.00	59.99	56.03	69.57	67.61
	3%	96.11	72.52	88.98	48.33	50.52	46.42	55.35	62.08	89.55	53.86	90.69	74.11	52.69	91.47	67.90	77.01	65.13	59.20	70.75	69.09
	4%	96.27	73.91	89.28	49.03	52.66	47.12	56.44	63.54	89.73	56.52	91.76	74.49	53.74	91.66	68.25	76.29	62.99	59.08	71.21	69.68
	5%	96.29	73.57	89.26	50.01	52.26	47.94	56.91	64.65	89.27	53.94	92.25	73.91	52.86	91.84	69.67	78.87	62.70	57.65	71.05	69.73
$\mathbf{D^2ADA}$ (DeepLabV3+)	1%	93.19	59.06	87.50	37.95	43.54	45.43	53.63	47.59	88.23	44.72	89.73	72.04	48.58	91.11	63.40	68.98	58.56	54.88	68.47	64.03
	2%	95.50	69.38	88.91	43.63	50.05	48.77	56.19	58.97	89.39	51.66	90.68	73.94	51.31	91.65	66.52	72.15	58.69	57.48	70.40	67.65
	3%	96.27	73.91	89.37	47.65	52.37	50.12	57.14	64.29	89.50	55.64	91.50	75.03	53.03	92.28	69.97	77.16	63.13	57.35	71.54	69.86
	4%	96.77	76.58	89.75	47.28	53.79	52.33	57.92	65.41	89.90	56.69	92.27	75.31	53.01	92.09	68.77	76.43	67.25	58.82	72.16	70.66
	5%	96.97	77.83	89.97	45.98	55.04	52.74	58.69	65.80	90.37	58.94	92.14	75.69	54.36	92.26	69.04	78.01	68.51	59.05	72.33	71.25

(b) SYNTHIA → Cityscapes

	% Target Labels	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Sky	PR	Rider	Car	Bus	Motor	Bike	mIoU	mIoU*
$\mathbf{D^2ADA}$ (DeepLabV2)	1%	93.82	61.50	85.21	26.15	19.21	40.75	46.41	53.16	86.11	87.27	70.78	46.77	86.54	34.08	48.49	66.21	59.53	66.64
	2%	94.95	67.30	87.79	37.92	42.04	44.09	53.45	61.05	88.17	90.11	73.64	53.37	89.98	66.26	53.54	69.36	67.07	73.00
	3%	95.73	71.75	88.48	38.68	44.08	46.4	54.48	64.64	88.68	90.18	74.49	53.99	90.72	73.27	57.48	70.86	68.99	74.98
	4%	96.21	73.95	88.93	41.23	48.24	47.45	55.31	65.65	89.23	91.24	74.59	54.39	91.07	73.37	58.21	71.55	70.04	75.67
	5%	96.41	74.57	89.09	42.51	47.70	47.99	55.64	66.46	89.47	91.73	75.10	55.15	91.37	76.97	57.97	71.77	70.62	76.28
$\mathbf{D^2ADA}$ (DeepLabV3+)	1%	92.45	55.44	86.75	34.94	29.07	44.90	48.97	54.43	87.09	90.27	73.66	49.39	88.98	40.74	52.85	69.64	62.47	68.51
	2%	95.54	71.45	88.78	38.97	45.69	50.34	55.57	64.78	89.46	92.06	75.61	53.38	91.14	69.67	55.49	71.70	69.35	74.97
	3%	96.13	74.04	89.11	39.64	49.52	52.58	56.24	65.91	89.89	92.80	76.38	54.63	92.20	76.06	58.77	72.31	71.01	76.50
	4%	96.19	74.40	89.91	48.48	50.71	53.60	58.10	66.88	90.01	93.29	77.15	56.28	92.25	78.39	59.54	73.27	72.40	77.36
	5%	96.67	76.76	90.27	48.73	51.06	54.24	58.28	67.99	90.41	93.37	77.37	56.41	92.53	77.53	58.88	73.29	72.74	77.67

5 Qualitative Results

Due to space limitations, we show the qualitative results in the supplementary material. As shown in Fig. 1, we show five inference results of different approaches for the GTA5 → Cityscapes domain adaptation task, including success and failure cases.

The first three rows present the results that our method outperforms the UDA method [10] and the previous ADA approach [9]. As shown in the first row, our segmentation result is close to full supervision with clear boundaries. Compared with the prior UDA [10] and ADA [9] practices, our method can better capture the scene structure and significantly outperforms them. The second and the third rows show that our method can better recognize hard categories, such as “train” and “fences” (shown on the red bounding box).

The fourth and fifth rows present two failure cases of our method. As observed from the red bounding boxes, our method performs worse than full supervision in these pictures' corners or boundary areas. However, compared with the two existing approaches, our method still achieves better results. We believe that the problem of poor performance in the pictures' corners or boundary areas may be improved through a better active selection strategy, which is worthy of further research in the future.

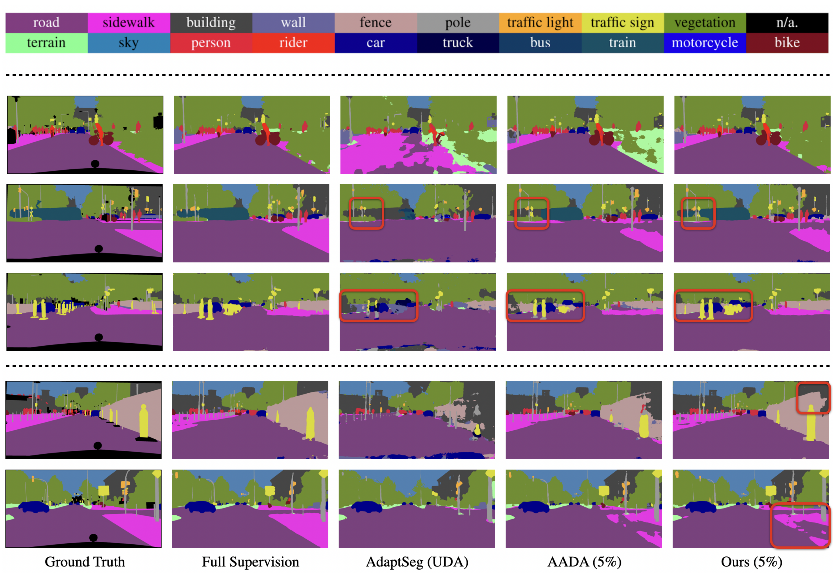

Fig. 1. Qualitative results of different approaches for the GTA5 → Cityscapes domain adaptation task. We present three success cases (in the top three rows) and two failure cases (in the bottom two rows) of our method. For more detailed explanation, please refer to Sec. 5.

References

1. Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2020. 5
2. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 8
3. Munan Ning, Donghuan Lu, Dong Wei, Cheng Bian, Chenglang Yuan, Shuang Yu, Kai Ma, and Yefeng Zheng. Multi-anchor active domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9112–9122, 2021. 5
4. Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8505–8514, 2021. 5
5. Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 8
6. German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 8
7. Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 4
8. Inkyu Shin, Dong-Jin Kim, Jae Won Cho, Sanghyun Woo, Kwanyong Park, and In So Kweon. Labor: Labeling only if required for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8588–8598, 2021. 5
9. Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 739–748, 2020. 5, 9
10. Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 1, 3, 5, 6, 9
11. D. Wang and Y. Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119, 2014. 3, 4
12. Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016. 4
13. Tsung-Han Wu, Yueh-Cheng Liu, Yu-Kai Huang, Hsin-Ying Lee, Hung-Ting Su, Ping-Chia Huang, and Winston H Hsu. Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15510–15519, 2021. 5

14. Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12414–12424, 2021. 5