Supplementary Material for Cooperative Self-Training for Multi-Target Adaptive Semantic Segmentation

The supplementary material is organized as follows: Sec. A summarizes the notations used. Sec. B describes the experimental details of our work. Sec. C reports the ablation study on hyperparameter sensitivity. Sec. D lists detailed quantitative comparisons on various configurations. Finally, Sec. E reports the qualitative visualizations pertaining to our method.

A. Notation

We summarize in Table A1 the notation used throughout the paper:

Notation	Description
$\mathcal{D}^{\mathrm{S}} = \{(\mathbf{x}_n^{\mathrm{S}}, \mathbf{y}_n^{\mathrm{S}})\}_{n=1}^{N^{\mathrm{S}}}$	Source data set
$\mathcal{D}^{\mathrm{T}_i} = \{\mathbf{x}_n^{\mathrm{T}_i}\}_{n=1}^{N^{\mathrm{T}_i}}$	i^{th} target data set
$\mathbf{x}^{\mathrm{S}} \in \mathbb{R}^{H\times W\times 3}$	Source inputs
$\mathbf{y}^{\mathrm{S}} \in \mathbb{R}^{H\times W\times K}$	Source labels
$\mathbf{x}^{\mathrm{T}_i} \in \mathbb{R}^{H\times W\times 3}$	Inputs from i^{th} target domain
f	Model function
Φ	Encoder network
C^{T_i}	i^{th} target domain-specific classifier
C^{A}	Domain-agnostic classifier
D^{T_i}	i^{th} target domain-specific discriminator
$\hat{\mathbf{p}}^{\mathrm{T}_i} \in \mathbb{R}^{H\times W\times K}$	Prediction from i^{th} target domain-specific classifier
$\mathbf{e}_k$	One-hot encoding operator
$\hat{\mathbf{y}}^{\mathrm{T}_i} \in \mathbb{R}^{H\times W}$	Pseudo-label for i^{th} target domain sample
$\mathbf{z}_l^{\mathrm{T}_i} = \Phi_l(\mathbf{x}^{\mathrm{T}_i})$	Latent feature of $\mathbf{x}^{\mathrm{T}_i}$ at l^{th} layer in Φ^{th} layer
$\boldsymbol{\mu}^{\mathrm{T}_i}, \boldsymbol{\sigma}^{\mathrm{T}_i}$	Style vectors (channel-wise mean and standard deviation) for i^{th} target domain input $\mathbf{x}^{\mathrm{T}_i}$
$\mathbf{z}_l^{\mathrm{T}_{i\rightarrow j}}$	Latent stylized feature of $\mathbf{x}^{\mathrm{T}_i}$ with content from $\mathbf{x}^{\mathrm{T}_i}$ and style from $\mathbf{x}^{\mathrm{T}_j}$
w_i	Averaged rectification weight for the sample $\mathbf{x}^{\mathrm{T}_i}$

Table A1: Notation used throughout the paper

B. Experimental Details

Datasets. We evaluate our method on two benchmarks previously used in the literature. These benchmarks are based on four datasets:

- *GTA5* [7] is collected from the video game GTA5. The dataset contains 24966 labeled images in total where the image resolution is 1914 $\times$ 1052. The synthetic nature of this dataset makes it very relevant for domain adaptation experiments.
- *Cityscapes* [1] is a large-scale dataset that has 2975 training and 500 validation labeled images collected mainly in German cities.
- *Mapillary* [5] contains 18000 training and 2000 validation high-resolution images collected from all over the world. Compared to Cityscapes, this dataset is more diverse.
- *IDD* [9] is collected on Indian roads and it has 6993 and 981 finely annotated images in training and validation sets respectively. *IDD* is very challenging since India cities visually differ from the cities depicted in the other datasets.

Implementation details. In the warm-up stage, we employ the hyper-parameters as [8] except that we extend the warm-up stage from 20K to 60K iterations to get better initial pseudo-labels for the self-training stage. In the second stage, we use Stochastic Gradient Descent optimizer with learning rate 1.0×10^{-4} to train the model for another 60K iterations. In all the experiments in the *7-classes* and *19-classes* settings, we use random crop of size 320×160 and 512×256 respectively to accelerate the training. In the second stage, we use strong data-augmentation and update the pseudo-labels every 10K iterations.

C. Ablation Study of Hyperparameters

In the final objective of our proposed CoaST, we weigh all the constituent losses and set other hyperparameters with a value that equals to 1. This disposes off the need to have a target validation set, which indeed is not available for any UDA setting. Nevertheless, below we study the sensitivity of CoaST with respect to two hyperparameters over ranges of possible values.

Ratio of Pair-wise Losses. We perform an ablation study on the weighing hyperparameter λ that weighs the pair-wise losses: consistency loss $\mathcal{L}_{\text{cst}}$ and the rectified segmentation losses $\bar{\mathcal{L}}_{\text{pl}}^{\text{sty}}$. The weighted training objective of our CoaST, first introduced in Eqn. **??** of the main paper, is written as:

$$
\begin{aligned}
\mathcal{L}_{\textit{CoaST}} = & \sum_{(\mathbf{x}^{\mathrm{S}},\mathbf{y}^{\mathrm{S}})\in\mathcal{D}^{\mathrm{S}}} \mathcal{L}_{\mathrm{seg}}(\mathbf{x}^{\mathrm{S}},\mathbf{y}^{\mathrm{S}}) \\
& + \sum_{i=1}^{M} \sum_{\mathbf{x}^{\mathrm{T}_i}\in\mathcal{D}^{\mathrm{T}_i}} \Big[\frac{1}{M}\mathcal{L}_{\mathrm{kd}}(\mathbf{x}^{\mathrm{T}_i}) + \bar{\mathcal{L}}_{\mathrm{pl}}(\mathbf{x}^{\mathrm{T}_i}) \\
& + \lambda \frac{1}{M-1} \sum_{\substack{j=1\\ j\neq i}}^{M} \sum_{\mathbf{x}^{\mathrm{T}_j}\in\mathcal{D}^{\mathrm{T}_j}} \big(\bar{\mathcal{L}}_{\mathrm{pl}}^{\mathrm{sty}}(\mathbf{x}^{\mathrm{T}_i},\mathbf{x}^{\mathrm{T}_j}) \\
& \qquad + \mathcal{L}_{\mathrm{cst}}(\mathbf{x}^{\mathrm{T}_i},\mathbf{x}^{\mathrm{T}_j}) \big) \Big]
\end{aligned}
\tag{A1}
$$

From the Fig. A1 (left), we can see that, the mIoU remains fairly stable over a wide operating window of λ. The performance starts to drop only when we increase the value of λ to large values. This is reasonable because when $\lambda = 10$, the $\mathcal{L}_{\mathrm{cst}}$ and the $\bar{\mathcal{L}}_{\mathrm{pl}}^{\mathrm{sty}}$ starts to dominate the other losses in Eqn A1. We observe a well-behaved training dynamics when we set the value of λ to standard value of 1, or $\log \lambda = 0$.

Temperature. The rectification weight described in the Eqn. **??** of the main paper is obtained by applying an exponential operation on the consistency score. To recap, the exp(.) function is used to bound the KL-divergence consistency score between]0,1], which otherwise is unbounded. The rectification weight can be regulated by using a *temperature* hyperparameter γ, that controls the steepness of the exp(.) curve. In other words, higher the value of γ, more quickly the curve goes to zero, and vice-versa. The rectification weight which is a function of γ is given as:

$$
w_i = \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} \exp\left(-\gamma \mathcal{L}_{\mathrm{kl}}(\hat{\mathbf{p}}^{\mathrm{T}_i}, \hat{\mathbf{p}}^{\mathrm{T}_{i\to j}})\right) \tag{A2}
$$

It can be observed from the Fig. A1 (right) that the performance of CoaST does not vary much while changing the temperature γ. Indeed, we see that the average mIoU remains in a tight range of 70.5% to 71.3%, even for extreme values of γ. Note that we vary the value of γ between 0.01 and 10 in our ablation study, whereas we report the logarithmic values of γ in Fig. A1 on the x-axis for clarity of the plot.

D. Quantitative Comparison.

D.1. Detailed Results of the Synthetic to Real Settings

In the Tab. **??** of the main paper, we reported the summary of the performances on all the settings with GTA5 as the source domain. In this section, we report the detailed class-wise results for those settings. The Tab. A2, A3 and A4 detail the results on the *7-class* setting while the Tab. A5 detail the results on the *19-class* setting. Note that the detailed results of G2CI are already shown in Tab. **??** and the Tab. **??** of the main paper.

In Tab. A2, A3 and A4, we can see that our CoaST outperforms all the baselines and *MTKT* [8] in *7-class* benchmark for most of the classes. These results are in-line with the summarized results reported in the main paper and confirm the consistent gain provided by our CoaST for the majority of the classes. In Tab. A5, we show the detailed comparison with *Individual* and *MTKT* in *19-class* benchmark. Note that, the detailed comparison with scores reported for every class is not reported in paper introducing *CCL* [3] and *ADAS* [4]. Since their codes are not publicly available, we could not provide the detailed class-wise scores. The comparison with *CCL* [3] and *ADAS* [4] could only be reported as in Tab. **??** of the main paper.

D.2. Synthetic to Real scenario: summary of all the Settings.

In the main paper, we report in Tab. **??** the average mIoU considering all the possible target configurations on the *19-classes* Benchmark in the *Synthetic to Real* scenario. We now report the results on the *7-classes* Benchmark in Tab. A6. In short, we observe that *CoaST* obtains performance on par with *ADAS* [4]. *CoaST* obtains the pest average performance in three configurations over four. These experiments demonstrate again the robustness of our approach.

D.3. Detailed Results of the Real to Real Settings

Here we show the comparison with *MTKT* in all the Real to Real settings on the *7-class* benchmark. We observe from the Tab. A7 that *CoaST* can clearly outperform *MTKT* in all the real to real configurations. This again proves the versatility of *CoaST* as it can yield better performance when trained on both synthetic and real source domains.

D.4. Comparison with other MTDA methods.

In this section, we compare our methods with other MTDA methods in the literature that have been proposed for object recognition. Following *CCL* [3], we report the numbers of *CoaST* on the *19-class* benchmark in the Tab. A8. Note that only *CCL* [3] and *CoaST* are specifically designed for semantic segmentation. The baselines in the MTDA setting [2, 6] that are designed for object recognition perform

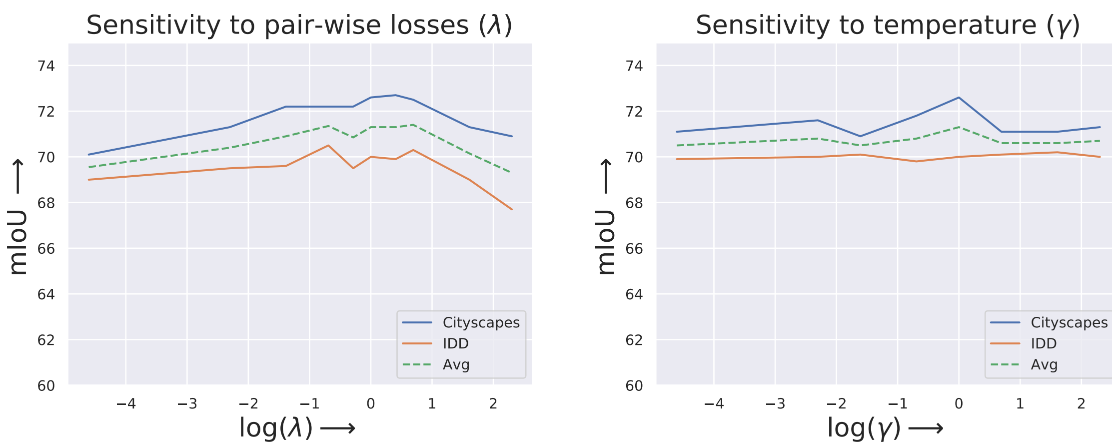

Figure A1: Sensitivity analysis of our proposed CoaST for the *7-class* MTDA configuration of GTA5 $\rightarrow$ Cityscapes + IDD. **Left**: we vary the pair-wise loss weight λ and evaluate the mIoU for the target domains. The performance curve remains stable over a wide operating window, and starts to degrade only for extreme values of the λ. **Right**: we vary the temperature γ and evaluate the mIoU for the target domains. We notice that the average mIoU varies slightly with γ. On the x-axis we plot the logarithmic values of the hyperparameters for clarity

GTA5 $\rightarrow$ Cityscapes + Mapillary										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
Individual [10]	C	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8	69.7
	M	89.5	72.6	31.0	75.3	94.1	50.7	73.8	69.6	
Data Comb. [10]	C	93.1	80.5	24.0	77.9	81.0	52.5	75.0	69.1	68.9
	M	90.0	71.3	31.1	73.0	92.6	46.6	76.6	68.7	
Multi-Dis [8]	C	94.5	80.8	22.2	79.2	82.1	47.0	79.0	69.3	69.5
	M	89.4	71.2	29.5	**76.2**	93.6	50.4	78.3	69.8	
MTKT [8]	C	95.0	81.6	23.6	80.1	**83.6**	53.7	79.8	71.1	70.9
	M	**90.6**	73.3	31.0	75.3	**94.5**	52.2	79.8	70.8	
ADAS [4](1024×512)	C	**96.4**	83.5	**35.1**	**83.6**	**84.9**	**62.3**	**81.3**	**75.3**	**73.9**
	M	88.6	73.7	**41.0**	**75.4**	**93.4**	**58.5**	77.2	**72.6**	
CoaST (Ours)	C	94.7	**84.4**	29.3	81.6	77.7	57.1	**81.3**	72.3	72.3
	M	89.2	**74.9**	37.5	74.6	89.2	57.9	**82.8**	72.3	

Table A2: The comparison of CoaST with the state-of-the-art on the *7-classes* benchmark using the GTA5 $\rightarrow$ Cityscapes + Mapillary configuration. We observe that CoaST outperforms MTKT on several classes and also on average

sub-optimally with respect *CCL* [3]. However, *CoaST* surpasses *CCL* [3] by a non-trivial margin, validating the importance of data driven image stylization for the MTDA in semantic segmentation.

D.5. Direct Transfer to Unseen Domains

Similar to [8], we directly test our adapted model on a new (or *unseen*) target domain to evaluate the generalization ability of our model. This setting is often referred to as *open-compound* domain adaptation in the literature. In the Tab. A9, we report the comparison of the generalization ability with other methods on *7-class* benchmark. We can observe that among considered MTDA baselines, our *CoaST* has the best generalization ability. This hints at the fact that our proposed cooperative self-training realized with feature stylization can induce better generalizability.

GTA5 → Mapillary + IDD										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
Individual [10]	M	89.5	72.6	31.0	75.3	94.1	50.7	73.8	69.6	67.4
	I	91.2	53.1	16.0	78.2	90.7	47.9	78.9	65.1	
Data Comb. [10]	M	89.6	71.0	**34.2**	74.5	92.9	47.3	78.6	69.7	67.9
	I	91.8	54.0	**17.4**	76.9	92.3	51.4	78.4	66.0	
Multi-Dis [8]	M	**89.9**	71.7	28.7	76.0	93.6	51.6	79.7	70.2	68.1
	I	91.4	54.9	14.6	78.5	**93.0**	51.1	79.0	66.1	
MTKT [8]	M	88.8	73.2	31.5	74.7	**94.1**	52.5	**79.9**	70.7	68.3
	I	91.4	55.9	13.5	76.7	92.1	**52.3**	79.4	65.9	
CoaST (Ours)	M	**90.5**	**75.9**	37.2	**73.6**	90.8	**57.5**	81.3	**72.4**	**70.6**
	I	**93.3**	**60.9**	19.8	**79.3**	91.2	54.1	**82.6**	**68.7**	

Table A3: The comparison of CoaST with the state-of-the-art on the *7-classes* benchmark using the GTA5 → Mapillary + IDD configuration. We observe that CoaST outperforms MTKT on several classes and also on average

GTA5 → Cityscapes + Mapillary + IDD										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
Individual [10]	C	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8	68.2
	M	89.5	72.6	31.0	75.3	94.1	50.7	73.8	69.6	
	I	91.2	53.1	16.0	78.2	90.7	47.9	78.9	65.1	
Data Comb. [10]	C	93.6	80.6	26.4	78.1	81.5	51.9	76.4	69.8	67.8
	M	89.2	72.4	32.4	73.0	92.7	41.6	74.9	68.0	
	I	92.0	54.6	15.7	77.2	90.5	50.8	78.6	65.6	
Multi-Dis [8]	C	**94.6**	80.0	20.6	79.3	84.1	44.6	78.2	68.8	68.2
	M	89.0	72.5	29.3	75.5	**94.7**	50.3	78.9	70.0	
	I	91.6	54.2	13.1	78.4	93.1	49.6	80.3	65.8	
MTKT [8]	C	**94.6**	**80.7**	23.8	79.0	84.5	51.0	79.2	70.4	69.1
	M	90.5	73.7	32.5	**75.5**	94.3	51.2	**80.2**	71.1	
	I	91.7	55.6	14.5	78.0	**92.6**	49.8	79.4	65.9	
ADAS [4](1024×512)	C	**95.8**	**82.4**	**38.3**	82.4	85.0	**60.5**	80.2	**74.9**	71.3
	M	89.2	71.5	**45.2**	**75.8**	92.3	56.1	75.4	**72.2**	
	I	89.9	52.7	**25.0**	78.1	92.1	51.0	77.9	66.7	
CoaST (Ours)	C	94.4	80.2	27.0	**82.6**	**88.3**	54.6	**81.0**	72.6	**71.7**
	M	**91.7**	**74.9**	36.2	73.9	92.0	**57.5**	79.5	**72.2**	
	I	**94.6**	**62.0**	21.0	**82.6**	**92.6**	**55.4**	**83.7**	**70.3**	

Table A4: The comparison of CoaST with the state-of-the-art on the *7-classes* benchmark using the GTA5 → Cityscapes + Mapillary + IDD configuration. We observe that CoaST outperforms MTKT on several classes and also on average. Particularly, the gain in performance for CoaST over MTKT for the IDD is fairly substantial

GTA5 → Cityscapes + IDD

Method	Target	road	sidewalk	building	walk	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mIoU	Avg.
Individual [10]	C	**88.8**	23.8	**81.5**	27.7	27.3	31.7	33.2	22.9	83.1	27.0	76.4	58.5	**28.9**	84.3	**30.0**	36.8	**0.3**	27.7	**33.1**	43.3	43.5
	I	**94.1**	24.4	**66.1**	31.3	22.0	25.4	9.3	26.7	80.0	**31.4**	**93.5**	48.7	43.8	71.4	49.4	28.5	**0**	48.7	**34.3**	43.6	
MTKT	C	88.5	37.2	79.1	22.8	19.8	26.3	33.8	16.7	84.8	34.2	80.5	54.9	15.0	84.1	27.5	41.2	**0.3**	27.9	7.0	41.7	42.5
	I	82.7	24.2	54.2	29.3	22.0	24.8	8.4	52.0	78.7	18.2	92.1	38.6	51.0	72.5	60.8	27.6	**0**	54.5	14.1	43.3	
CoaST (Ours)	C	81.7	**38.3**	71.0	**33.3**	**30.7**	**35.1**	**38.2**	**37.6**	**86.4**	**46.9**	**81.9**	**63.4**	27.4	**84.5**	29.4	**45.6**	**0.3**	**32.6**	31.3	**47.1**	**48.2**
	I	85.7	**36.1**	65.1	**33.2**	**23.7**	**32.8**	**19.0**	**62.9**	**82.5**	29.5	91.8	**52.1**	**55.3**	**83.4**	**62.9**	**46.1**	**0**	**55.5**	18.5	**49.3**	

GTA5 → Cityscapes + Mapillary

Method	Target	road	sidewalk	building	walk	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mIoU	Avg.
Individual [10]	C	88.8	23.8	**81.5**	27.7	27.3	31.7	33.2	22.9	83.1	27.0	76.4	58.5	**28.9**	84.3	30.0	36.8	0.3	27.7	33.1	43.3	44.7
	M	81.1	18.6	74.8	23.9	**28.9**	30.3	35.7	33.7	**78.4**	40.7	93.3	49.5	**42.3**	80.4	35.1	34.2	**17.8**	41.8	**36.1**	46.1	
MTKT [8]	C	**89.2**	36.1	**81.5**	31.6	22.1	28.4	31.4	13.8	85.1	34.3	**83.5**	57.6	19.1	**86.1**	**36.0**	44.1	0.4	**32.5**	6.1	43.1	45.4
	M	**86.8**	38.7	78.7	27.0	28.4	29.5	37.3	34.6	**78.4**	42.3	**94.9**	53.7	37.9	**84.2**	41.1	34.5	15.5	**44.0**	18.0	47.6	
CoaST (Ours)	C	82.1	**36.2**	77.5	**47.4**	**34.9**	**36.7**	**42.0**	**36.6**	**87.2**	**38.6**	80.8	**60.6**	21.6	**86.1**	33.3	**45.7**	**2.5**	26.2	**34.7**	**47.9**	**49.9**
	M	84.7	**44.4**	**80.3**	**35.7**	27.7	**37.2**	**45.1**	**51.8**	73.8	**42.4**	93.7	**64.5**	42.2	83.9	**49.0**	**44.0**	10.0	38.5	35.6	**51.8**	

GTA5 → Mapillary + IDD

Method	Target	road	sidewalk	building	walk	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mIoU	Avg.
Individual [10]	M	81.1	18.6	74.8	23.9	28.9	30.3	35.7	33.7	78.4	40.7	93.3	49.5	**42.3**	80.4	35.1	34.2	**17.8**	**41.8**	36.1	46.1	44.9
	I	**94.1**	24.4	66.1	31.3	**22.0**	25.4	9.3	26.7	80.0	31.4	93.5	48.7	43.8	71.4	49.4	28.5	**0**	48.7	**34.3**	43.6	
MTKT [8]	M	**85.4**	**42.8**	78.6	28.9	**32.8**	31.0	32.8	35.4	**79.8**	**45.0**	**95.1**	54.1	34.3	82.5	36.2	34.1	9.5	40.6	**37.6**	48.2	45.4
	I	82.1	14.1	56.4	31.4	21.3	28.6	12.5	43.0	**81.0**	26.9	**93.6**	35.8	46.9	76.2	56.9	39.5	**0**	50.5	13.4	42.6	
CoaST (Ours)	M	79.4	35.3	**81.0**	**34.6**	30.9	**37.8**	**43.7**	**52.7**	74.1	**45.0**	93.4	**63.7**	**43.2**	**84.9**	**48.2**	**51.3**	5.3	39.8	36.6	**51.6**	**50.6**
	I	87.1	**30.1**	**66.3**	**34.7**	21.8	**34.5**	**18.9**	**66.0**	80.6	**41.7**	91.3	**52.8**	**55.8**	**83.7**	**58.4**	**48.0**	**0**	**55.6**	13.2	**49.5**	

GTA5 → Cityscapes + Mapillary + IDD

Method	Target	road	sidewalk	building	walk	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mIoU	Avg.
Individual [10]	C	**88.8**	23.8	**81.5**	27.7	27.3	31.7	33.2	22.9	83.1	27.0	76.4	58.5	**28.9**	84.3	30.0	36.8	0.3	27.7	**33.1**	43.3	
	M	81.1	18.6	74.8	23.9	28.9	30.3	35.7	33.7	**78.4**	40.7	93.3	49.5	42.3	80.4	35.1	34.2	17.8	**41.8**	**36.1**	46.1	43.3
	I	**94.1**	24.4	**66.1**	31.3	**22.0**	25.4	9.3	26.7	80.0	**31.4**	**93.5**	48.7	43.8	71.4	49.4	28.5	**0**	48.7	**34.3**	43.6	
MTKT [8]	C	85.9	33.7	81.2	30.2	20.0	31.3	32.7	17.5	84.1	33.2	80.8	56.1	16.8	83.2	26.0	39.2	**10.9**	24.4	13.7	42.2	
	M	**85.9**	42.1	76.1	29.1	28.7	30.3	35.1	34.4	76.6	43.1	**93.7**	55.2	31.1	82.7	37.8	31.8	**20.8**	37.8	17.3	46.8	43.9
	I	83.1	16.9	55.7	**34.7**	21.3	27.6	8.0	48.6	77.7	26.9	91.9	36.7	46.2	74.2	56.1	41.3	**0**	48.7	15.7	42.7	
CoaST (Ours)	C	88.4	**43.0**	80.0	**30.9**	**29.4**	**37.6**	**36.9**	**42.1**	**86.2**	**40.9**	**81.5**	**60.1**	15.4	**85.5**	**33.5**	**44.5**	4.6	**30.7**	26.7	47.2	
	M	82.8	**44.8**	**79.5**	**32.3**	**37.9**	**38.3**	**38.2**	**52.4**	76.0	**45.5**	92.9	**65.2**	**39.2**	**85.8**	**51.0**	**43.1**	6.2	38.2	27.3	**51.4**	**49.1**
	I	86.9	**29.0**	64.1	31.2	20.2	**36.7**	**14.8**	**51.9**	**82.3**	**48.2**	92.7	**51.8**	**53.6**	**83.8**	**60.7**	**46.6**	**0**	**50.5**	20.6	**48.7**	

Table A5: The detailed class-wise comparison of *CoaST* in the *19-class* setting with the existing state-of-the-art methods. In all the experiments, GTA5 is considered as the source domain and the various combinations of the other benchmarks are considered as the target domains. In all the configurations our CoaST clearly outperforms the existing baselines by a clear margin

Target C I M	method	mIoU C	I	M	mIoU Avg.
	***7-classes* Benchmark**				
√ √ -	*MTKT* [8]	70.4	65.9	-	68.2
	ADAS [4](1024 × 512)	**75.4**	66.9	-	71.2
	CoaST (Ours)	72.6	**70.0**	-	**71.3**
√ - √	*MTKT* [8]	71.1	-	70.8	71.0
	ADAS [4](1024 × 512)	**75.3**	-	**72.6**	**73.9**
	CoaST (Ours)	72.3	-	72.3	72.3
- √ √	*MTKT* [8]	-	65.9	70.7	68.3
	ADAS [4](1024 × 512)	-	-	-	-
	CoaST (Ours)	-	**68.7**	**72.4**	**70.6**
√ √ √	*MTKT* [8]	70.4	65.9	71.1	69.1
	ADAS [4](1024 × 512)	**74.9**	66.7	**72.2**	71.3
	CoaST (Ours)	**72.6**	70.3	**72.2**	**71.7**

Table A6: Summary of performances obtained on the *7-classes* Benchmark with different dataset configurations. Cityscapes, IDD and Mapillary are referred to as C, I and M respectively. We report the mIoU averaged over the target domains.

E. Visualization

To get insights about the cooperative rectification used in our *CoaST*, we provide some visualization of the uncertainty map estimated by our approach in Sec. E.1 and report a qualitative comparison with the state-of-the-art in Sec. E.2.

E.1. Pseudo-label Uncertainty Maps

In the Fig. A2 we show the uncertainty maps estimated by *CoaST* for random training images on the *7-classes* benchmark. More precisely, we visualize the consistency scores $\mathcal{L}_{KL}$ used in Eqn. **??** to estimate the rectification weights, which we call as uncertainty maps. In addition, we show the pseudo-label (PL) maps and the corresponding error map. The error map is computed by subtracting the PL map from the ground truth. From the Fig. A2 we can see that in most cases, regions with high uncertainty (warm colors) correspond to errors while regions that are correctly segmented have low uncertainty (dark blue). For example, in the first part of the Fig. A2, which show the visualizations from the *Cityscapes* data set, we can see that the most uncertain regions correspond to the object boundaries between classes. Since the object boundaries are the most challenging regions for the model, the uncertainty maps are especially bright for such regions. We can notice in the last part of Fig. A2 that for the IDD data set the regions with multiple occlusions have high uncertainty.

Similarly, in Fig. A3 we also report the uncertainty map obtained in the case of the *19-classes* benchmark. Coherent with the observations found for the *7-classes* benchmark, we again see that uncertain regions usually correspond to to regions with errors.

E.2. State-of-the-art Qualitative Comparison

In addition to the qualitative comparisons provided in the Fig. **??** of the main paper, we also report qualitative comparisons on the *7-classes* benchmark in the GTA5 → Cityscapes + Mapillary + IDD configuration. We compare our proposed *CoaST* with MTKT in the Fig. A4. In-line with the observation from the main paper, we also find that *CoaST* is better at segmenting small obscure objects from the *human* class in the IDD data set. Success on the hard classes eventually leads *CoaST* to reach improved numbers over the existing state-of-the-art methods.

Cityscapes → Mapillary + IDD										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
MTKT[8]	M	88.3	70.4	31.6	75.9	**94.4**	50.9	77.0	69.8	69.0
	I	93.6	54.9	18.6	84.0	**94.5**	53.4	79.2	68.3	
CoaST (Ours)	M	**90.2**	**73.4**	**37.2**	**78.8**	92.3	**59.2**	**84.1**	**73.6**	**72.6**
	I	**95.1**	**58.0**	**26.6**	**85.4**	93.0	**59.0**	**83.9**	**71.6**	
Mapillary → Cityscapes + IDD										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
MTKT[8]	C	94.7	81.9	35.6	83.0	84.7	57.0	83.9	74.4	72.7
	I	95.2	61.6	24.6	85.4	94.3	55.7	81.1	71.1	
CoaST (Ours)	C	**95.6**	**84.4**	**36.7**	**83.9**	**88.2**	**58.2**	**85.8**	**76.1**	**74.8**
	I	**95.5**	**64.6**	**31.1**	**85.8**	**94.6**	**58.2**	**84.7**	**73.5**	
IDD → Cityscapes + Mapillary										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
MTKT[8]	C	**96.7**	82.8	31.0	**84.7**	**89.8**	**60.2**	85.1	75.8	73.9
	M	90.4	71.2	33.8	79.1	95.8	55.3	79.0	72.1	
CoaST (Ours)	C	96.5	**84.3**	**33.6**	**84.7**	89.1	58.3	**85.8**	**76.0**	**75.7**
	M	**91.0**	**76.3**	**39.7**	**82.2**	**96.0**	**59.0**	**83.1**	**75.3**	

Table A7: The detailed comparison of CoaST with the state-of-the-art on the *7-classes* benchmark in all Real to Real scenarios. CoaST clearly outperforms MTKT in all the real to real configuration.

GTA5 → Cityscapes + IDD				
Setting	**Method**	**mIoU** C	I	**Avg.**
DG	*Yue et al.* [11]	42.1	42.8	42.5
MTDA	*MTDA-ITA* [2]	40.3	41.2	40.8
	MT-MTDA [6]	43.2	44.0	43.6
	CCL [3]	45.0	46.0	45.5
	CoaST (Ours)	**47.1**	**49.3**	**48.2**

Table A8: The quantitative comparison of our CoaST with different MTDA methods on the *19-class* benchmark for the GTA5 → Cityscapes + IDD configuration. CoaST outperforms the considered MTDA baselines that are designed for either object recognition or semantic segmentation. DG stands for domain generalization setting and a method designed for such a setting is also under performed by CoaST

Direct Transfer to an Unseen Target Domain										
Setup	**Method**	**Test**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**
G → C + I	*Data Comb.* [10]	M	88.4	71.0	31.0	72.4	92.0	37.4	74.7	66.7
	Multi-Dis[8]		89.2	72.1	21.7	73.8	**94.0**	34.8	75.9	65.9
	MTKT[8]		89.8	**74.0**	30.4	74.1	93.6	52.6	79.4	70.6
	CoaST (Ours)		**91.6**	73.9	**34.8**	**77.8**	93.0	**57.7**	**81.9**	**72.9**
G → C + M	*Data Comb.*[10]	I	91.6	54.7	13.9	76.5	90.9	48.3	77.5	64.8
	Multi-Dis[8]		91.2	54.6	12.9	77.7	**92.5**	50.3	78.6	65.4
	MTKT[8]		91.5	56.1	12.3	76.1	90.9	51.4	79.2	65.4
	CoaST (Ours)		**93.2**	**59.7**	**17.1**	**80.1**	91.0	**51.7**	**81.2**	**67.7**

Table A9: The quantitative comparison for direct transfer to new (or *unseen*) domains in *7-class* benchmark. GTA5, Cityscapes, Mapillary and IDD are referred to as G, C, M and I, respectively. The **Test** column denotes the unseen target domain where the models have been evaluated

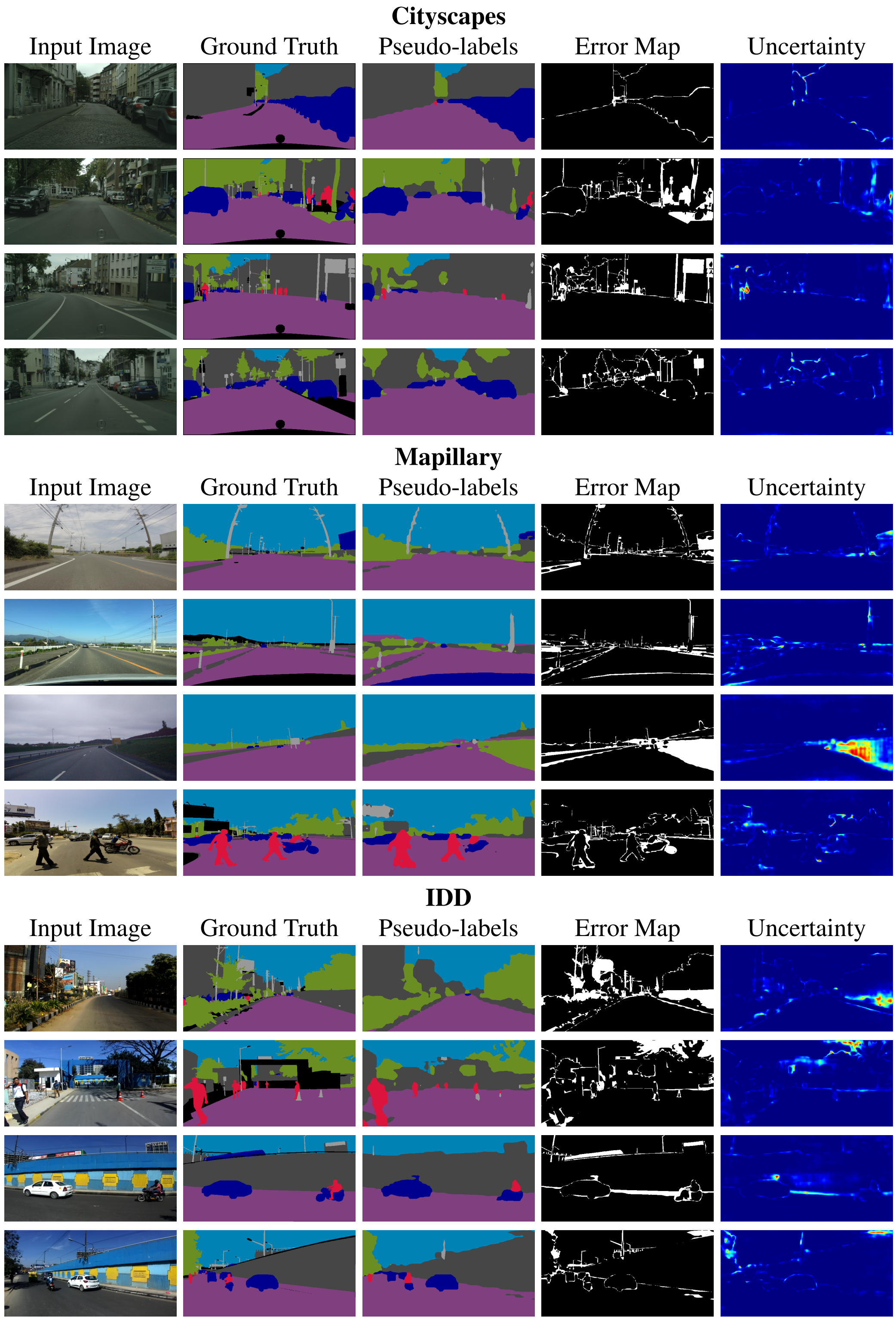

Figure A2: Visualization of uncertainty maps (or *rectification weights*) estimated by CoaST for the *7-class* benchmark in the GTA5 $\rightarrow$ Cityscapes + Mapillary + IDD configuration. The error maps are computed by subtracting the peusdo-labels from the ground truth. The uncertainty maps can be seen to be correctly correlated with the error maps

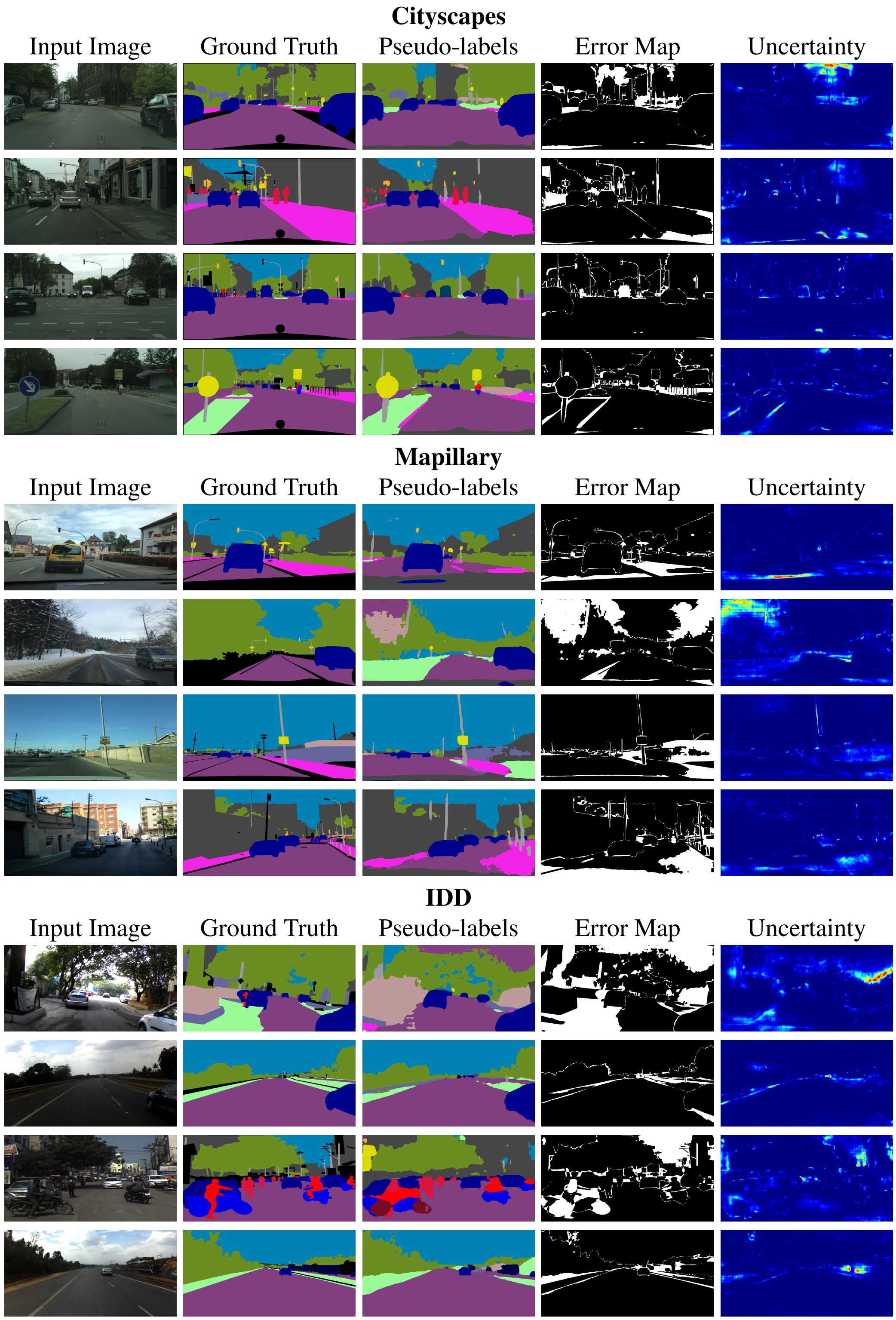

Figure A3: Visualization of uncertainty maps (or *rectification weights*) estimated by CoaST for the *19-class* benchmark in the GTA5 $\rightarrow$ Cityscapes + Mapillary + IDD configuration. The error maps are computed by subtracting the peusdo-labels from the ground truth. The uncertainty maps can be seen to be correctly correlated with the error maps

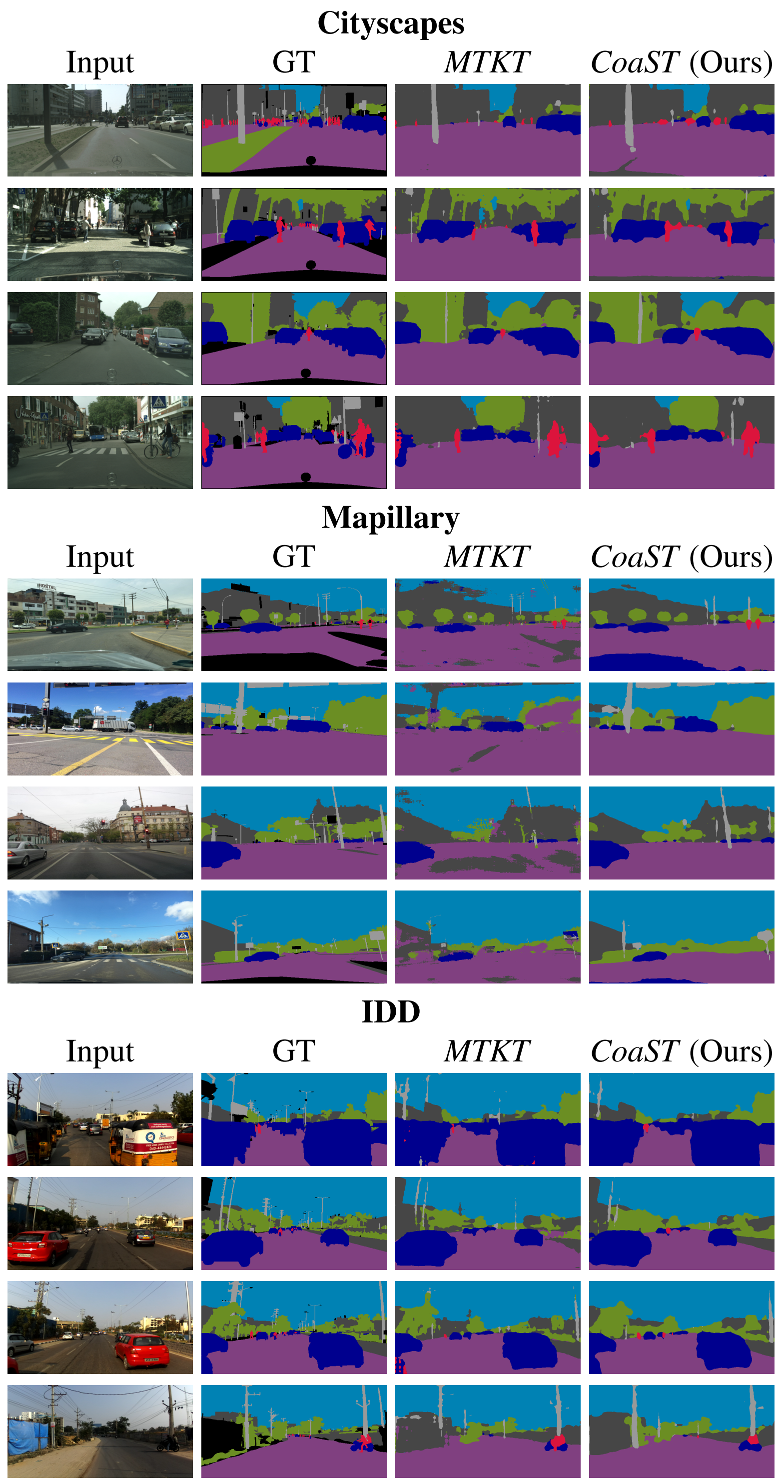

Figure A4: Qualitative comparisons of state-of-the-art predictions for the *7-classes* benchmark in the GTA5 → Cityscapes + Mapillary + IDD configuration

References

[1] 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 (CVPR)*, pages 3213–3223, 2016.

[2] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing (TIP)*, 29:3993–4002, 2020.

[3] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8187–8196, 2021.

[4] Seunghun Lee, Wonhyeok Choi, Changjae Kim, Minwoo Choi, and Sunghoon Im. Adas: A direct adaptation strategy for multi-target domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19196–19206, 2022.

[5] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4990–4999, 2017.

[6] Le Thanh Nguyen-Meidine, Atif Belal, Madhu Kiran, Jose Dolz, Louis-Antoine Blais-Morin, and Eric Granger. Unsupervised multi-target domain adaptation through knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1339–1347, 2021.

[7] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 102–118, 2016.

[8] Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9072–9081, 2021.

[9] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1743–1751, 2019.

[10] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2517–2526, 2019.

[11] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2100–2110, 2019.