Supplementary For: Learning Cascaded Detection Tasks with Weakly-Supervised Domain Adaptation

Niklas Hanselmann[1,2], Nick Schneider[1], Benedikt Ortelt[1,3] and Andreas Geiger[2,4]

In this supplementary material we provide further implementation and training details, as well as additional discussion and results. We start by describing the used hyperparameters and experimental setup in more detail in Sec. I as well as measures to correct inconsistencies in label definitions between the different datasets in Sec. II. We then give an overview of the datasets' class distributions in Sec. III. Finally, in Sec. IV, we provide an additional visual comparison of all methods and show quantitative results for all random seeds of our main experiments, including per-class results.

I. Training and Implementation Details

As mentioned in the main paper, we use Mask-RCNN [2] as the cascaded detection framework, where we replace the mask predictor by the 3D bounding box regressor proposed in [4] for the monocular 3D detection task. All models are optimized using Stochastic Gradient Descent with a momentum of 0.9 and weight decay of 0.1. The learning rate is set to 10^{-2} with a linear warmup from 10^{-5} for the first 1000 iterations. Besides the initial warmup period, we use multi-step learning rate decay with task specific schedules as detailed below. We select a batch size of 8 for all experiments and compose each batch such that it contains an equal number of source and target domain images.

Instance Segmentation: In all instance segmentation experiments we train for a total of 24000 iterations, decaying the learning rate by a factor of 0.1 at 18000 iterations. For both loss terms $\mathcal{L}_{det}$ and $\mathcal{L}_{att}$, we use the original loss functions and weightings of [2]. For data augmentation, we apply horizontal flipping and scale augmentations, where we resize the image such that the shorter side is of a randomly sampled length between 800 and 1024 pixels.

Monocular 3D Detection: For Monocular 3D detection, we train for a total of 120000 iterations, decaying the learning rate by a factor of 0.1 at 72000 and again at 90000 iterations. For $\mathcal{L}_{det}$, we again use the original loss functions and weightings of [2]. For the 3D detection loss $\mathcal{L}_{att}$, we use the disentangled Huber loss as in the original work [4]. However, we found that equalizing the losses by individually reweighting the different disentangled terms leads to improved performance. We thus use the following weights for each of the disentangled terms: A factor of 2 for the depth and rotation and a factor of 100 for the projected center and box dimensions. We use no augmentations for the monocular 3D detection task.

II. Consistency of Label Definitions Across Domains

Benchmarking the ability of domain adaptation algorithms to mitigate distribution shifts can be challenging in practice due to inconsistencies in the definitions of labels between commonly used source and target domain datasets. Although these inconsistencies are not always trivially resolved, there are some that can be dealt with manually, which we describe in the following.

The "Rider" Class in Synscapes and Cityscapes: One key inconsistency in labeling policies between Synscapes and Cityscapes stems from the definitions of the rideable vehicle classes (e.g. "bicycle" and "motorcycles"). In Synscapes, these classes include the rider, while in Cityscapes there exists a separate "rider" class, a problem that effects both instance segmentation and monocular 3D detection. For instance segmentation, this can be resolved by utilizing Synscapes' semantic segmentation ground-truth: Since here, riders are annotated as "person", we can accurately isolate the vehicle and its rider in all instance masks by performing a pixel-wise lookup for the "person" class in the corresponding semantic segmentation map. Although in principle one could use the obtained instance masks to also correct the object's 3D bounding box dimensions, this only works for non-occluded objects. Since the variance in dimensions for these classes is low in Synscapes, we instead heuristically correct them by cropping their height by 0.65m and 0.6m for "bicycle" and "motorcycle", respectively, which we found to work well in practice.

Articulated Vehicles: Jointed vehicles like articulated buses, trams and trucks can be difficult to describe with a single 3D bounding box when the angles between segments are large. In Cityscapes, each segment is hence annotated with a separate bounding box in these cases, while in Synscapes, these classes are always annotated with a single bounding box. We therefore remove the "bus" and "train" classes from consideration for our monocular 3D detection experiments, but keep the "truck" class, since the number of jointed trucks in Cityscapes is low.

[1]Mercedes-Benz AG, R&D, Stuttgart, Germany
[2]University of Tübingen, Tübingen, Germany
[3]Robert Bosch GmbH, Stuttgart, Germany
[4]Max Planck Institute for Intelligent Systems, Tübingen, Germany
Primary contact: niklas.hanselmann@daimler.com

Common Label Spaces: Since not all classes always exist in both the source and target domain datasets, we use the set of common classes for each scenario and task. Specifically, for instance segmentation we drop the classes "bicycle" and "train" in the VIPER to Cityscapes scenario, because they either do not exist or are extremely rare in the source domain dataset. Additionally, we merge the classes "person" and "rider" as VIPER does not distinguish between the two. For all other tasks and scenarios we use the full set of semantic classes available in Cityscapes, unless there are inconsistencies in label definitions that are not easily resolved, as described above.

III. Label Distributions

As discussed in the main paper, domain-adversarial feature alignment approaches often do not consider class information and can fail in the presence of label shift. In this section, we show the distributions of object classes for the considered datasets. We omit Foggy Cityscapes, since it has exactly the same label distribution as Cityscapes. As can be seen from Fig. 1 and 2, the relative frequencies of instance classes differ quite strongly between the datasets. In addition, the average number of instances per image for each class also varies significantly, suggesting a shift in scene compositions.

IV. Additional Qualitative and Quantitative Results

We provide further visual comparison of results on both tasks in Fig. 3 to 6. Additionally, we report the full quantitative results of our main experiments for each class over all three random seeds in Tab. I to V.

References

[1] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017.

[3] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

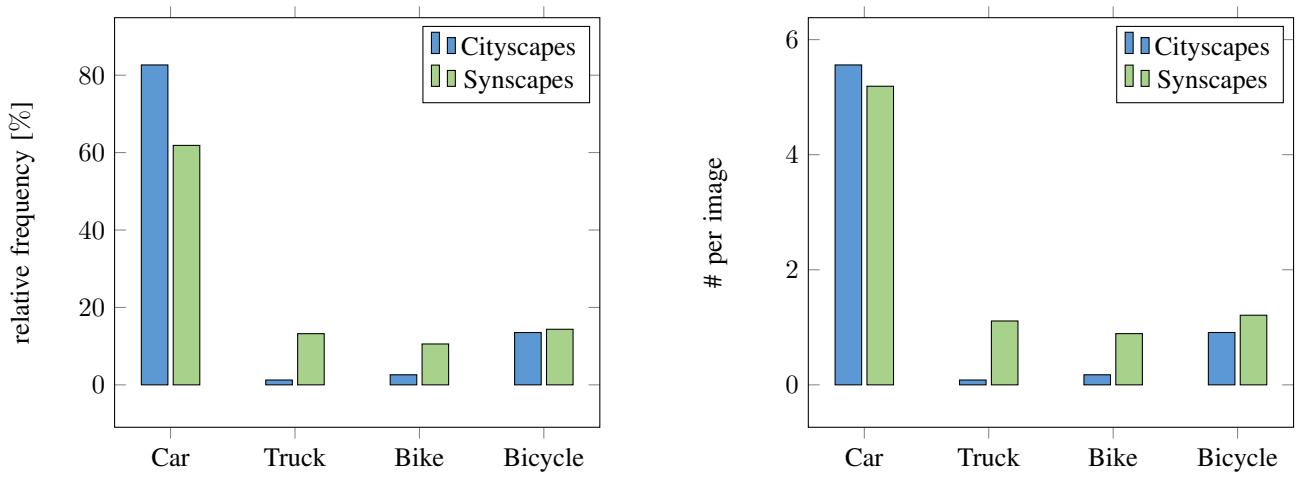


Fig. 1: **Class distributions for monocular 3D detection.** Relative occurence frequencies of each class overall (*left*) and average occurences per image (*right*).

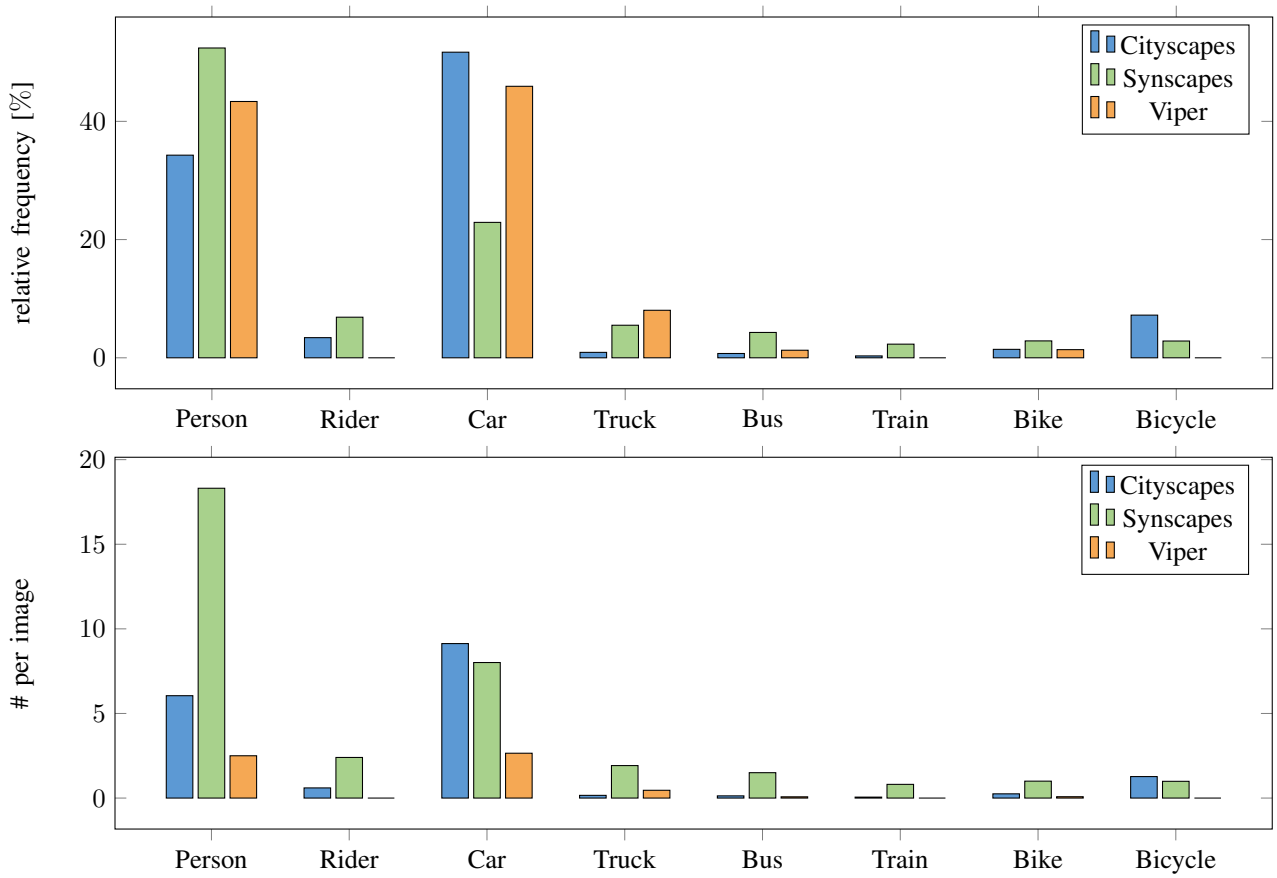
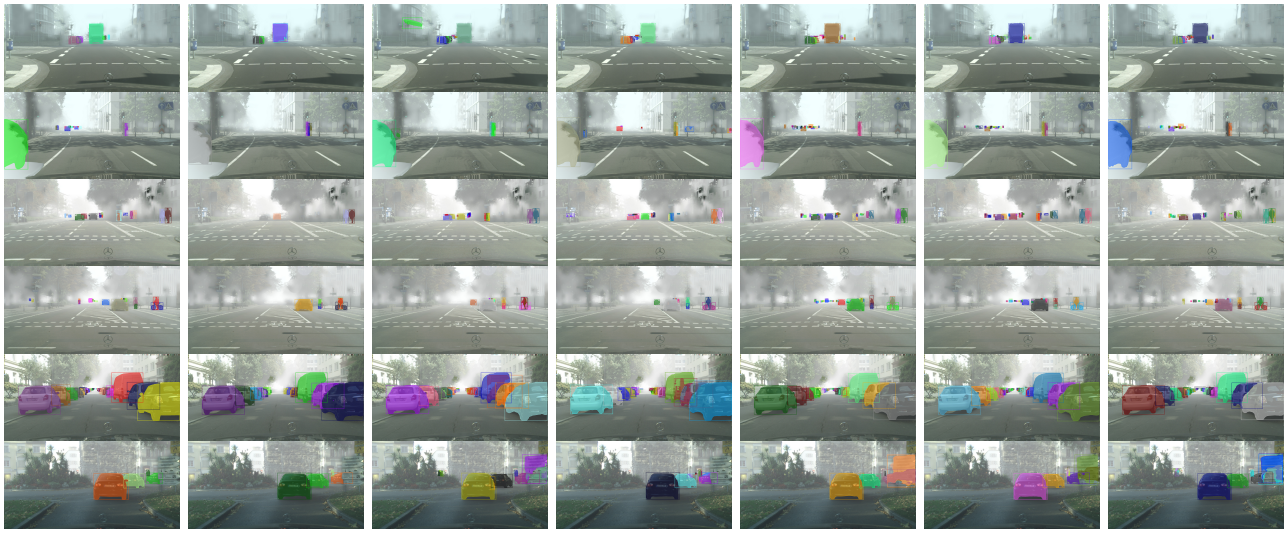


Fig. 2: **Class distributions for instance segmentation.** Relative occurence frequencies of each class overall (*top*) and average occurences per image (*bottom*).



(a) Ground-truth (b) Source only (c) DARCNN [1] (d) SWDA [3] (e) Ours (f) Ours + CWFA (g) Oracle

Fig. 3: **Additional instance segmentation results (CS $\rightarrow$ FCS).** Best viewed zoomed in.

(a) Ground-truth (b) Source only (c) DARCNN [1] (d) SWDA [3] (e) Ours (f) Ours + CWFA (g) Oracle

Fig. 4: **Additional monocular 3D detection results (SYN $\rightarrow$ CS).** Best viewed zoomed in.

(a) Ground-truth (b) Source only (c) DARCNN [1] (d) SWDA [3] (e) Ours (f) Ours + CWFA (g) Oracle

Fig. 5: **Additional monocular 3D detection results (CS $\rightarrow$ FCS).** Best viewed zoomed in.

(a) Ground-truth (b) Source only (c) DARCNN [1] (d) SWDA [3] (e) Ours (f) Ours + CWFA (g) Oracle

Fig. 6: **Additional monocular 3D detection results (SYN $\rightarrow$ CS).** Best viewed zoomed in.

Method	Run	mAP	Person	Rider	Car	Truck	Bus	Train	Bike	Bicycle
Source Only	1	16.22	23.85	10.46	38.70	5.70	33.66	5.97	4.91	6.52
	2	17.03	23.95	11.28	38.57	6.89	31.25	12.50	5.26	6.52
	3	15.84	23.75	10.94	37.51	5.55	31.48	5.96	4.97	6.54
DAFRCNN	1	17.54	24.17	12.86	38.91	8.20	35.10	8.42	6.21	6.46
	2	17.40	24.48	11.48	37.80	7.97	31.42	14.07	5.81	6.17
	3	16.98	23.87	11.81	38.37	5.37	34.70	9.68	5.87	6.15
SWDA	1	17.36	23.62	10.97	37.99	7.34	37.60	9.23	6.01	6.09
	2	17.53	25.05	12.68	38.52	9.17	34.00	10.35	4.59	5.85
	3	15.89	22.95	10.80	37.77	6.45	32.22	6.64	4.09	6.21
Ours	1	31.66	30.92	18.88	48.23	**35.49**	56.38	36.59	12.80	13.96
	2	31.91	31.65	18.93	47.95	33.70	55.12	**38.24**	**15.58**	14.14
	3	**32.05**	**31.94**	19.13	48.08	31.68	57.02	37.79	15.45	**15.32**
Ours + CWFA	1	30.83	30.44	20.72	48.26	29.99	55.54	32.59	15.24	13.90
	2	30.78	29.92	21.19	48.09	31.04	55.61	30.95	14.93	14.50
	3	31.25	30.51	**21.69**	**48.58**	31.33	54.53	35.72	13.70	13.92
Oracle	1	33.30	31.89	24.98	51.94	30.14	52.21	37.39	18.36	19.51
	2	33.61	31.92	25.33	51.93	28.86	54.49	39.26	18.04	19.08
	3	33.62	32.05	25.77	51.82	32.18	52.54	34.65	20.33	19.60

TABLE I: **Instance segmentation.** Results for SYN $\rightarrow$ CS. We highlight the best runs, as reported in the main paper, in green.

Method	Run	mAP	Person	Car	Truck	Bus	Bike
Source Only	1	6.56	7.32	14.42	3.12	7.87	0.09
	2	6.90	9.20	14.60	3.65	6.84	0.19
	3	6.25	7.16	13.51	5.44	4.98	0.16
DAFRCNN	1	7.79	7.97	16.46	7.07	6.99	0.44
	2	9.06	10.27	17.36	9.50	7.83	0.35
	3	7.72	8.40	17.11	8.09	4.50	0.50
SWDA	1	6.66	7.98	11.97	4.69	8.62	0.04
	2	6.41	8.64	14.88	5.74	2.73	0.06
	3	6.63	7.22	13.73	5.09	6.76	0.38
Ours	1	**30.26**	20.32	38.66	**32.62**	48.79	10.9
	2	29.49	20.42	38.37	30.12	49.19	9.36
	3	30.07	**20.50**	38.57	30.49	47.15	**13.62**
Ours + CWFA	1	30.23	19.77	38.38	31.70	48.26	13.03
	2	29.03	**20.50**	38.08	29.72	45.10	11.73
	3	30.07	**20.50**	38.57	30.49	47.15	**13.62**
Oracle	1	35.87	25.24	52.17	32.03	52.28	17.64
	2	36.05	24.47	51.48	32.86	53.01	18.42
	3	35.77	24.96	51.73	31.11	52.19	18.86

TABLE II: **Instance Segmentation.** Results for VIPER $\rightarrow$ CS. We highlight the best runs, as reported in the main paper, in green.

Method	Run	mAP	Person	Rider	Car	Truck	Bus	Train	Bike	Bicycle
Source Only	1	13.10	16.38	13.93	25.81	8.96	19.29	4.22	7.06	9.16
	2	13.29	16.59	14.37	26.86	8.99	18.33	3.86	7.71	9.59
	3	13.92	16.51	13.00	27.97	12.09	23.10	3.91	6.12	8.68
DAFRCNN	1	21.86	22.50	18.57	38.58	17.77	32.91	20.34	9.98	14.24
	2	23.25	23.41	19.42	38.77	18.17	35.81	22.89	12.65	14.86
	3	21.22	22.38	18.57	37.99	18.83	34.01	13.07	10.44	14.43
SWDA	1	21.63	22.01	19.18	38.52	21.07	34.30	13.22	11.22	13.49
	2	21.98	23.09	19.51	38.48	18.42	35.12	16.76	10.52	13.94
	3	22.05	22.76	18.82	38.35	20.01	33.11	19.46	10.68	13.20
Ours	1	29.93	28.52	22.80	48.55	**28.99**	48.16	29.93	14.56	17.93
	2	29.39	28.57	22.36	48.17	26.82	44.35	30.17	**16.88**	17.82
	3	30.26	**29.15**	**23.58**	48.60	25.06	48.07	35.14	15.11	17.38
Ours + CWFA	1	30.31	28.47	21.90	**48.76**	27.71	48.76	33.80	15.16	17.92
	2	29.58	29.13	22.72	48.34	25.17	44.81	33.01	15.71	17.75
	3	29.67	29.06	22.66	48.52	27.68	45.20	30.90	15.20	18.11
Oracle	1	30.16	28.50	22.50	47.86	27.57	46.43	36.68	14.29	17.47
	2	30.12	28.56	21.69	48.42	27.57	47.91	34.11	14.28	18.39
	3	29.15	28.48	22.17	47.90	26.93	47.58	29.69	12.89	17.53

TABLE III: **Instance segmentation.** Results for CS $\rightarrow$ FCS. We highlight the best runs, as reported in the main paper, in green.

Method	Run	mDS	mAP	Car DS	Truck DS	Bike DS	Bicycle DS
Source Only	1	13.60	15.39	38.52	1.78	1.79	12.30
	2	13.73	15.55	37.25	7.7	7.32	9.57
	3	14.18	16.12	38.40	1.39	4.94	11.97
DAFRCNN	1	15.08	17.05	41.97	2.24	3.81	12.28
	2	15.30	17.37	41.58	1.97	5.49	12.16
	3	15.41	17.51	41.39	2.50	5.84	11.92
SWDA	1	14.82	16.92	38.00	1.62	8.87	10.78
	2	13.90	15.72	41.21	1.80	3.69	8.90
	3	14.56	16.49	41.17	1.63	4.30	11.16
Ours	1	20.39	23.33	44.98	7.6	13.8	15.19
	2	18.93	22.06	45.50	8.69	10.19	11.35
	3	19.02	21.84	45.02	9.68	6.34	15.02
Ours + CWFA	1	21.71	24.85	48.33	13.49	9.39	15.60
	2	**23.41**	**26.77**	**48.67**	**15.18**	11.92	**17.89**
	3	21.19	24.31	48.02	9.96	10.81	15.96
Oracle	1	23.34	25.20	56.90	9.20	10.56	16.70
	2	24.89	26.67	55.14	14.12	11.55	18.73
	3	25.03	26.09	55.87	9.26	16.08	18.90

TABLE IV: **Monocular 3D Detection.** Results for SYN $\rightarrow$ CS. We highlight the best runs, as reported in the main paper, in green. We also report the mAP for the 2D projections of the 3D bounding boxes onto the image plane.

Method	Run	mDS	mAP	Car DS	Truck DS	Bike DS	Bicycle DS
Source Only	1	12.80	13.93	31.44	4.00	4.58	11.18
	2	13.82	15.01	29.96	1.22	9.89	14.23
	3	13.71	14.77	32.78	3.02	6.60	12.43
DAFRCNN	1	19.46	21.11	44.82	9.10	6.33	17.58
	2	17.72	19.09	42.41	6.23	6.00	16.25
	3	18.50	19.86	43.46	6.87	7.72	15.96
SWDA	1	14.00	15.05	32.16	1.91	5.07	16.86
	2	17.01	18.44	40.16	3.82	8.30	15.77
	3	17.83	19.16	41.20	5.31	7.72	17.08
Ours	1	22.56	24.38	53.70	9.06	10.44	17.06
	2	21.95	23.68	50.43	**11.45**	9.60	16.33
	3	22.21	23.73	53.50	4.39	10.56	**20.39**
Ours + CWFA	1	23.82	25.77	**54.68**	10.96	10.98	18.67
	2	22.33	23.94	52.90	6.42	11.27	18.71
	3	**24.19**	**26.43**	53.00	7.73	**16.53**	19.50
Oracle	1	20.84	22.23	51.85	5.25	9.38	16.87
	2	24.66	26.25	52.50	9.10	17.60	19.45
	3	23.01	24.51	53.75	9.70	15.40	13.20

TABLE V: **Monocular 3D Detection.** Results for CS $\rightarrow$ FCS. We highlight the best runs, as reported in the main paper, in green. We also report the mAP for the 2D projections of the 3D bounding boxes onto the image plane.