Supplementary Material for CVPR 2023 Paper: Object Detection with Self-Supervised Scene Adaptation

Zekun Zhang[1] Minh Hoai[1,2]

[1]Stony Brook University, Stony Brook, NY 11794, USA
[2]VinAI Artificial Intelligence Application and Research JSC, Hanoi, Vietnam

{zekzhang,minhhoai}@cs.stonyrbook.edu

Abstract

In this supplementary document, we show more sample frames and distribution of camera locations of Scenes100 *to illustrate its diversity. We provide more implementation details and the running speed for the experiments in the main paper. We also look into the quality of the pseudo-labeling and model performance from the perspective of individual videos. Lastly we show the results of compound adaptation experiments. Please refer to https://github.com/cvlab-stonybrook/scenes100 for the dataset and code. It is recommended to read this document on a color screen, and zoom in for fine details in the figures.*

A. Sample Frames and Distribution of Camera Locations of Scenes100

In Fig. A we present sample frames along with their non-annotation masks and annotation bounding boxes from 12 videos in Scenes100 to show the diversity of the dataset. Please refer to the sub-captions for our comments for each of the scenes. The diversity of scenes shows the usefulness of the dataset. And since all videos share the same set of hyper-parameters in the adaptation experiments, it reiterates the effectiveness and robustness of the proposed self-supervised scene adaptive object detection method.

The locations of the cameras of the videos in Scenes100 is shown on a world map in Fig. B. Unlike most other object detection or scene understanding datasets, which are captured in a smaller range of locations, the videos in Scenes100 were recorded in places across the globe, giving great diversity.

B. More Implementation Details

B.1. Software Libraries, Hyper-parameters, and Training Details

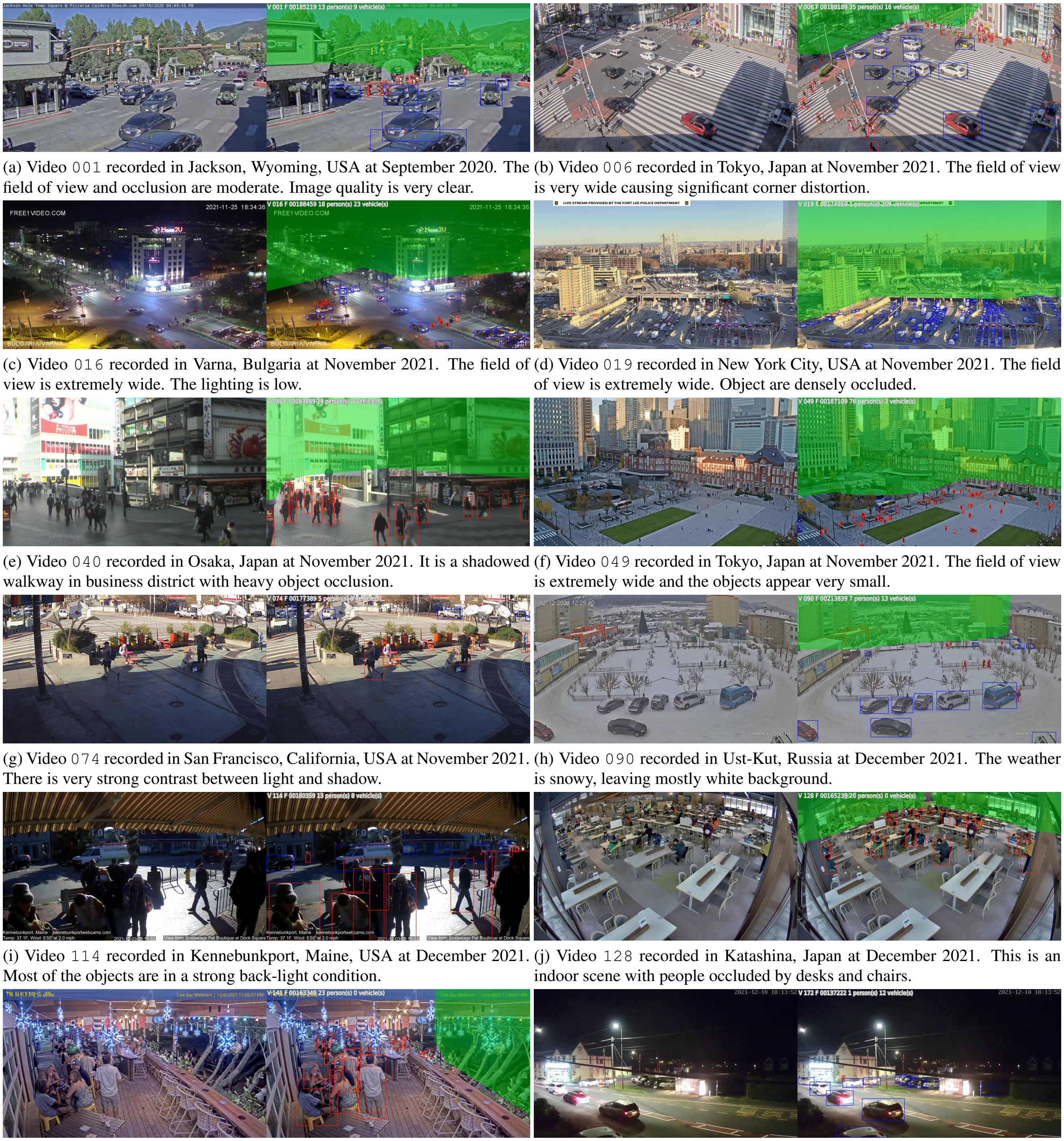
We start the finetuning from the models provided by the Detectron2 [5] model zoo. M1 and M2 are based on the configurations “COCO-Detection/ faster_rcnn_R_50_FPN_3x.yaml” and “COCO-Detection/ faster_rcnn_R_101_FPN_3x.yaml”, respectively. We keep the weights of the backbone and RPN, but re-initialize the weights of the new ROI heads, as the number of classes changes during the object categories remapping described in the main paper. Then the whole network is trained end-to-end on remapped MSCOCO training set. We use learning rate scheduling with base of 5×10^{-4}, image batch size of 4, and ROI batch size of 128. The models are trained for 15,000 iterations. We examine the models’ performance on validation set and losses periodically during training to ensure convergence.

For self-supervised adaptation training, the training video portions are down-sampled uniformly to 5 frames per second, which gives 27,000 training frames per video. All spatial resolutions of the frames are kept. For pseudo-labeling, we set $\lambda_{det} = 0.5$, $\lambda_{sot} = 0.9$, and $\lambda_{iou} = 0.85$. For location-aware mixup, we set $p_{mixup} = 0.3$, $r_{mixup} = 0.5$, and $\alpha_{cover} = 0.65$. For dynamic background extraction we set $T_{bg} = 90s$. For fusion models training, we use average pooling for feature pyramid fusion, and set $\alpha_{mid} = \alpha_{late} = 0.5$. In adaptation training, we use learning rate scheduling with base of 10^{-4}, image batch size of 4, and ROI batch size of 128. The models are trained for 20,000 iterations. We examine the models’ performance on validation set and losses periodically during training to ensure convergence.

B.2. Implementation Details of Compared Baselines

The official implementation[1] of **Self-Train (ST)** [3] does not include the code for detection, tracking, and hard nega-

[1] https://github.com/AruniRC/detectron-self-train


Jackson Hole Town Square @ Pizzeria Caldera ©SeeJH.com 09/10/2020 04:49:15 PM
V 001 F 00185219 13 person(s) 9 vehicle(s)
V 006 F 00188189 35 person(s) 16 vehicle(s)
FREE1VIDEO.COM
2021-11-25 18:34:36
V 016 F 00188459 18 person(s) 23 vehicle(s)
BULGARIA/VARNA
LIVE STREAM PROVIDED BY THE FORT LEE POLICE DEPARTMENT
V 019 F 00174959 3 person(s) 209 vehicle(s)
V 040 F 00163889 29 person(s) 0 vehicle(s)
V 049 F 00187109 76 person(s) 3 vehicle(s)
V 090 F 00213839 7 person(s) 13 vehicle(s)
V 114 F 00180359 13 person(s) 8 vehicle(s)
Kennebunkport, Maine kennebunkportwebcams.com
Temp: 37.1F, Wind: ESE at 2.0 mph
View from Spatawags Pet Boutique at Dock Square
V 128 F 00165239 20 person(s) 0 vehicle(s)
Live Bar Webcam | 12/6/2021 11:05:57 PM
V 141 F 00163349 23 person(s) 0 vehicle(s)
2021-12-10 18:13:52
V 172 F 00137222 1 person(s) 12 vehicle(s)

(a) Video `001` recorded in Jackson, Wyoming, USA at September 2020. The field of view and occlusion are moderate. Image quality is very clear.

(b) Video `006` recorded in Tokyo, Japan at November 2021. The field of view is very wide causing significant corner distortion.

(c) Video `016` recorded in Varna, Bulgaria at November 2021. The field of view is extremely wide. The lighting is low.

(d) Video `019` recorded in New York City, USA at November 2021. The field of view is extremely wide. Object are densely occluded.

(e) Video `040` recorded in Osaka, Japan at November 2021. It is a shadowed walkway in business district with heavy object occlusion.

(f) Video `049` recorded in Tokyo, Japan at November 2021. The field of view is extremely wide and the objects appear very small.

(g) Video `074` recorded in San Francisco, California, USA at November 2021. There is very strong contrast between light and shadow.

(h) Video `090` recorded in Ust-Kut, Russia at December 2021. The weather is snowy, leaving mostly white background.

(i) Video `114` recorded in Kennebunkport, Maine, USA at December 2021. Most of the objects are in a strong back-light condition.

(j) Video `128` recorded in Katashina, Japan at December 2021. This is an indoor scene with people occluded by desks and chairs.

(k) Video `141` recorded in St John, U.S. Virgin Islands at December 2021. It is an indoor scene with heavy object occlusion.

(l) Video `172` recorded in Ammanford, Wales at December 2021. The lighting is very low, and motion blur of the objects is strong.

Figure A. Sample frames from Scenes100 with their non-annotation masks and annotation bounding boxes. As described in the sub-captions, the videos cover a variety of locations, weather, lighting conditions, image qualities, camera perspectives, and indoor/outdoor environments. The diversity of our dataset makes it representative for various scenes and thus useful for the understudied scene adaptive object detection task.

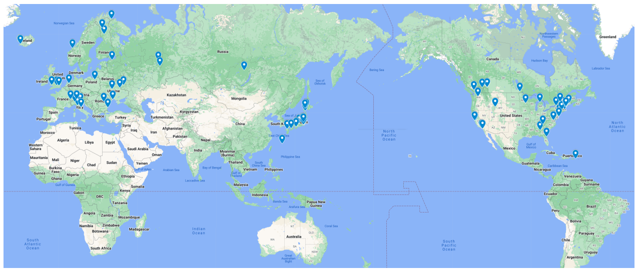


Figure B. The locations of the cameras in Scenes100 on the world map. Antarctica and Arctic regions are not included. Each location is represented by a blue pin. The number of pins is smaller than 100, as some of the videos are captured in the same city. Map is created using tools provided by Google Maps. Please see https://support.google.com/maps/answer/3145721 for its conventions on region names and borders.

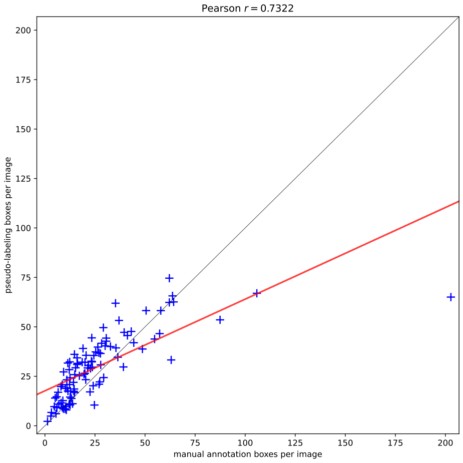


Figure C. Compare the number of object bounding boxes per image of manual annotation and pseudo-labeling. Each scatter point + represents one video. The red line — indicates the least-square linear model fit to the scatter. The Pearson correlation coefficient is displayed.

tive mining, so we re-implemented it in our framework.

We take the code of the core algorithm from the official implementation[2] of **STAC** [4], which is the image augmentation function set, and integrate it into our framework.

We directly use the official implementation[3] of **Adaptive Teacher (AT)** [2] and simply replace the initial base model and data loader with ours.

We directly use the official implementation[4] of **H^{2}FA R-CNN** [6], and simply replace the initial base model and data loader with ours.

The official implementation[5] of **TIA** [7] is based on another framework not compatible with our dataset and model architecture. So we implement it following the core logic in the official code base. We have to reduce the coefficient of auxiliary consistency losses by a factor of 10 to avoid training divergence. Number of training iterations is also reduced to 250, for longer training schedule leads to strong performance degradation.

The official implementation[6] of **LODS** [1] is based on another framework not compatible with our dataset and model architecture. So we implement it following the core logic in the official code base. Number of training iterations is also reduced to 250 due to performance degradation.

B.3. Inference Speed of Different Fusion Models

We show the inference throughput of vanilla faster-RCNN, early-fusion faster-RCNN, mid-fusion faster-RCNN, and late-fusion faster-RCNN in Tab. A. All models use the same R-101 backbone, and are ran on the same NVIDIA RTX 4090 GPU using the same set of images. The dataloader is carefully optimized to eliminate any possible CPU bottleneck. Please note that although mid-fusion and late-fusion introduces significantly more parameters by duplicating the RPN and ROI networks, only the branch for the fused feature pyramid will be deployed for inference. Thus they do not enlarge the model compared to vanilla faster-RCNN at inference time.

fusion model	inference throughput (images/s)
vanilla model	17.78
early-fusion	17.33
mid-fusion	11.93
late-fusion	11.96

Table A. Inference throughput of different fusion models in term of images per second. As stated in the main paper, early-fusion adds very limited computational cost, while mid-fusion and late-fusion impact the speed more significantly, but they give higher precision.

B.4. Effect of Pseudo-Labeling Hyper-parameters

We try to exclude tracking pseudo bounding boxes from training. We also test only using M2 to generate the detection results. The results are shown in Tab. B. Including tracking bounding boxes in pseudo-labeling provides consistent performance gain. The results of only using M2 in detection also show that ensemble of models is beneficial.

We change the hyper-parameters λ_{det} and λ_{iou} for pseudo-labeling, and see how they effect the performance of the adapted models. To avoid interfering of other factors, location-aware mixup and object mask fusion are not

[2] https://github.com/google-research/ssl_detection
[3] https://github.com/facebookresearch/adaptive_teacher
[4] https://github.com/XuYunqiu/H2FA_R-CNN
[5] https://github.com/MCG-NJU/TIA
[6] https://github.com/Flashkong/Source-Free-Object-Detection-by-Learning-to-Overlook-Domain-Style

detectors	tracking	APG_{co}^{m}	APG_{co}^{50}	APG_{w}^{m}	APG_{w}^{50}
M1+M2	✓	**+0.95**	**+0.54**	**+1.67**	**+1.55**
M1+M2	✗	+0.73	+0.22	+1.49	+1.31
M2	✓	+0.58	-0.30	+1.21	+0.67

Table B. Effects of tracking and ensemble of models on adaptation performance, in term of averaged AP gain. ^{m} and ^{50} stand for mean over IoU thresholds and $IoU = 50\%$, respectively. $_{co}$ and $_{w}$ stand for standard classes mean and proposed classes-weighted mean. ✓ and ✗ mean being and not being applied, respectively.

used. The results are shown in Tab. C. The results show that increasing λ_{det} to certain level can lead to increase of APG^{m}, but decrease of APG^{50}. The performance will be degraded if λ_{det} is too high. Lower λ_{iou} leads to slightly higher performance.

λ_{det}	λ_{iou}	APG_{co}^{m}	APG_{co}^{50}	APG_{w}^{m}	APG_{w}^{50}
0.5	0.85	+0.95	+0.54	+1.67	+1.55
0.7	0.85	**+1.47**	+0.43	+1.79	+0.78
0.9	0.85	+0.33	-3.02	-0.12	-4.09
0.5	0.75	+1.28	**+0.60**	**+2.03**	**+1.75**
0.5	0.95	+0.36	+0.50	+1.00	+1.51

Table C. Effects of pseudo-labeling hyper-parameters on adaptation performance, in term of averaged AP gain. ^{m} and ^{50} stand for mean over IoU thresholds and $IoU = 50\%$, respectively. $_{co}$ and $_{w}$ stand for standard classes mean and proposed classes-weighted mean. λ_{det} is the minimum score for a detected object to be included in the pseudo bounding boxes. λ_{iou} is the IoU for 2 bounding boxes to be merged during refinement.

B.5. Effect of Mixup Hyper-parameters

We change the hyper-parameters p_{mixup}, r_{mixup}, and α_{cover} for our proposed location-aware object mixup, and see how they effect the performance of the adapted models. To avoid interfering of other factors, object mask fusion is not used. The results are shown in Tab. D. It can be seen that increasing p_{mixup} or r_{mixup}, meaning stronger mixup, can improve the APs slightly.

B.6. Discussion

Please note that the results shown in Tab. C and Tab. D should not be viewed as a full-scale hyper-parameter tuning. The hyper-parameters in pseudo-labeling and location-aware mixup can interfere with each other. The situation is more complicated when object mask fusion is used. A proper tuning will require a search in the full hyper-parameter space, which is far beyond the computational capacity we have. It can also be reasonably expected that each video in Scenes100 has it own set of optimal hyper-parameters. Nevertheless, we argue that our proposed methods are mostly insensitive to the hyper-parameters, and can still perform well even no video-specific tuning is applied.

C. Individual Video Based Analysis

In the main paper, all the results are given in the form of the average over all the videos in Scenes100. However, due to their diversity, it is natural that our proposed methods perform differently on different videos. Here we take a deeper look into the individual performance of our methods.

C.1. Quantitative Assessment of Pseudo-Labeling

In Fig. C, we compare the number of manually labeled object bounding boxes per image with the number of pseudo bounding boxes after the proposed pseudo-labeling procedure. The same set of hyper-parameters are used as in the experiments. All the pseudo boxes with at least 1 corner inside the non-evaluation mask are removed for consistency. Please note that the mask is not used in the adaptation training experiments. It is clear the number of bounding boxes from pseudo-labeling is correlated with actual number of bounding boxes from human annotation. Please note that the comparison is not precise, since the frames used for pseudo-labeling and human annotation come from different parts of the videos, which means the density of objects can change. When the object density is not very high (less than 50 object per image), the correlation is strong, showing the proposed pseudo-labeling can identify most of the objects. However, when the object density is very high, meaning that there is heavy occlusion or the field of view is extremely wide, the scene becomes more difficult for object detectors and the number of pseudo bounding boxes is smaller compared to the actual number.

C.2. Correlation of AP Gains of Methods

We take a framework either from the baseline methods or from our ablation study, pair it with our best proposed method (pseudo-labeling + location-aware mixup + object mask mid-fusion, indicated by PL+MX+MF). we plot the individual AP gain after adaptation on each video of the 2 models in the pair as a scatter and calculate the Pearson correlation coefficient. The results are shown in Fig. D.

It is clear that only ST and LODS, which give moderate AP gain during adaptation, is weakly correlated with the

p_{mixup}	r_{mixup}	α_{cover}	APG_{co}^{m}	APG_{co}^{50}	APG_{w}^{m}	APG_{w}^{50}
0.3	0.5	0.65	+1.72	+1.67	+2.25	+2.53
0.5	0.5	0.65	+1.77	+1.72	+2.41	+2.75
0.7	0.5	0.65	**+2.06**	**+2.16**	**+2.47**	**+2.90**
0.3	0.3	0.65	+1.67	+1.67	+2.15	+2.47
0.3	0.7	0.65	+1.60	+1.59	+2.20	+2.57
0.3	0.5	0.45	+1.81	+1.70	+2.29	+2.52
0.3	0.5	0.85	+1.73	+1.74	+2.23	+2.60

Table D. Effects of locate-aware mixup hyper-parameters on adaptation performance, in term of averaged AP gain. ^{m} and ^{50} stand for mean over IoU thresholds and $IoU = 50\%$, respectively. $_{co}$ and $_{w}$ stand for standard classes mean and proposed classes-weighted mean. p_{mixup} is the probability that a frame is selected for mixup. r_{mixup} is the probability a pseudo bounding box in the mixup source frame is copied and pasted. α_{cover} is the threshold that a covered pseudo bounding box to be removed if exceeded.

proposed method. STAC, AT, H^{2}FA, and TIA all are not correlated with the proposed method. However, the combinations PL (pseudo-labeling), PL+MX (pseudo-labeling + location-aware mixup), PL+EF (pseudo-labeling + object mask early-fusion), PL+MF (pseudo-labeling + object mask mid-fusion), and PL+LF (pseudo-labeling + object mask late-fusion) are all strongly correlated with the best method. This indicates that our proposed methods are consistent in scene adaptation performance. The scenes that all methods perform poorly can be regarded as hard samples.

C.3. Success and Failure Cases Study

In Fig. E we present the videos that the best proposed method performs extraordinary well or bad in term of APG_{w}^{m}, and discuss the possible causes. Since it is the AP gains being examined, good performance means not only that the adapted model achieves high precision, but also that the base model cannot perform very well so there is room for improvement. Bad performance means the adaptation process actually degrade the detection ability of the base model.

Please note that the case study is qualitative and empirical. In our future work we will carry more systematic analysis on the performance, and improve our methods based on the observations.

D. Performance of Compound Models

Here we treat Scenes100 in a manner closer to the standard domain adaptive object detection problem. All 100 videos are regarded as a whole target domain, and the models are trained on all the training frames from them, resulting a generic (compound) model for all videos. For consistency reasons, we keep all the hyper-parameters and settings during training unchanged from the individual adaptation settings used in the main paper, only to increase the number of training iterations by 20× to incorporate larger training set. After training, we still use the same independent evaluation protocol as in the main paper, and report the average AP gains in Tab. E.

Interestingly, different methods performs very differently under this compound adaptation setting compared to individual adaptation setting. ST is the only method the performs noticeably better, implying that it benefits from higher variance in the training data. AT and TIA perform significantly worse. Our proposed method sees about 1-point drop in the APs, but is still the best one by a considerable margin. This shows that the proposed method is more suitable for a fine-grained scene adaptive learning setting compared to more generic domain adaptive one. Trying to explain the vast difference between methods under different settings can be an interesting direction of research in our future work.

References

[1] Shuaifeng Li, Mao Ye, Xiatian Zhu, Lihua Zhou, and Lin Xiong. Source-free object detection by learning to overlook domain style. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2022. 3, 6

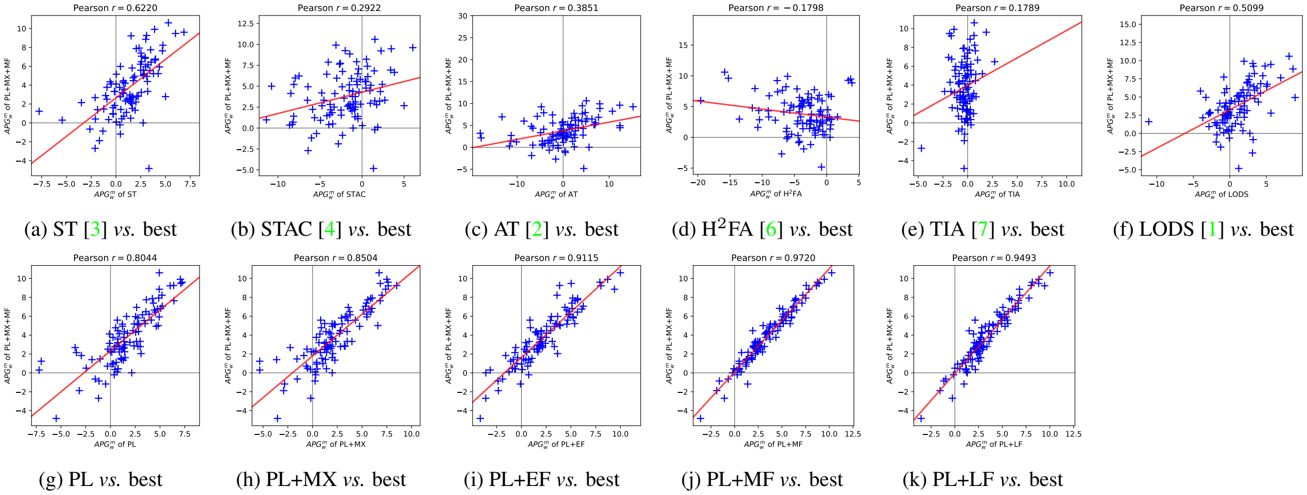
[2] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2022. 3, 6

[3] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2019. 1, 6

[4] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. In *arXiv*, 2020. 3, 6

[5] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 1

[6] Yunqiu Xu, Yifan Sun, Zongxin Yang, Jiaxu Miao, and Yi Yang. H^{2}FA R-CNN: Holistic and hierarchical feature alignment for cross-domain weakly supervised object detec-



(a) ST [3] *vs.* best (b) STAC [4] *vs.* best (c) AT [2] *vs.* best (d) H^{2}FA [6] *vs.* best (e) TIA [7] *vs.* best (f) LODS [1] *vs.* best

(g) PL *vs.* best (h) PL+MX *vs.* best (i) PL+EF *vs.* best (j) PL+MF *vs.* best (k) PL+LF *vs.* best

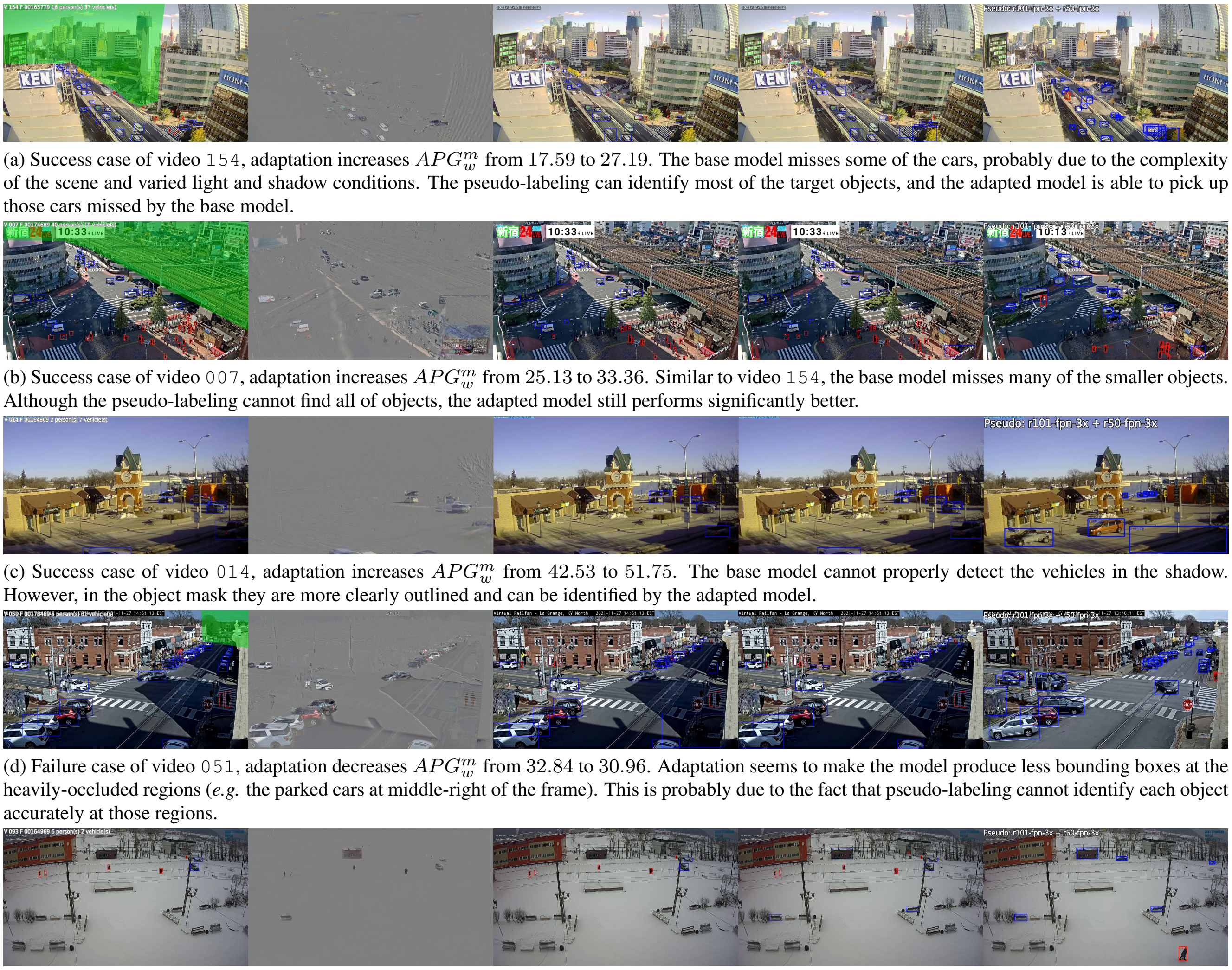
Figure D. Compare the AP gains of different frameworks against the proposed best model. Each scatter point + represents one video. The red lines — indicate the least-square linear models fit to the scatters. The Pearson correlation coefficients are displayed.

Method	Individually-adapted (main paper)				Compound adaptation			
	APG_{co}^{m}	APG_{co}^{50}	APG_{w}^{m}	APG_{w}^{50}	APG_{co}^{m}	APG_{co}^{50}	APG_{w}^{m}	APG_{w}^{50}
ST [3]	+0.80	+0.24	+1.39	+1.03	+1.69	+1.47	+1.69	+1.46
STAC [4]	-1.26	-5.12	-1.97	-6.64	-1.03	-4.81	-1.56	-6.03
AT [2]	-0.75	-1.11	+0.06	+0.04	-4.64	-6.92	-3.57	-5.50
H^{2}FA [6]	-3.10	-4.97	-3.77	-6.01	-3.78	-5.98	-3.99	-6.66
TIA [7]	-0.32	-0.37	-0.32	-0.33	-1.82	-2.76	-1.58	-2.34
LODS [1]	+0.45	+1.28	+1.02	+2.28	+0.59	+1.01	+0.69	+1.33
Proposed	**+3.76**	**+4.45**	**+3.78**	**+4.65**	**+2.77**	**+3.68**	**+2.56**	**+3.48**

Table E. Averaged AP gain of different compound adaptation models. The numbers of AP gain under individual adaptation setting are copied from the main paper for easy comparison. ^{m} and ^{50} stand for mean over IoU thresholds and $IoU = 50\%$, respectively. $_{co}$ and $_{w}$ stand for standard classes mean and proposed classes-weighted mean.

tion. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2022. 3, 6

[7] Liang Zhao and Limin Wang. Task-specific inconsistency alignment for domain adaptive object detection. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2022. 3, 6


V 154 F 00165779 18 person(s) 37 vehicle(s)
KEN
Pseudo: r101-fpn-3x + r50-fpn-3x
新宿24
10:33
10:13
Pseudo: r101-fpn-3x + r50-fpn-3x
V 014 F 00164969 2 person(s) 7 vehicle(s)
Pseudo: r101-fpn-3x + r50-fpn-3x
Virtual Railfan - La Grange, KY North 2021-11-27 14:51:13 EST
Pseudo: r101-fpn-3x + r50-fpn-3x
V 093 F 00164969 6 person(s) 2 vehicle(s)
Pseudo: r101-fpn-3x + r50-fpn-3x

(a) Success case of video 154, adaptation increases APG_w^m from 17.59 to 27.19. The base model misses some of the cars, probably due to the complexity of the scene and varied light and shadow conditions. The pseudo-labeling can identify most of the target objects, and the adapted model is able to pick up those cars missed by the base model.

(b) Success case of video 007, adaptation increases APG_w^m from 25.13 to 33.36. Similar to video 154, the base model misses many of the smaller objects. Although the pseudo-labeling cannot find all of objects, the adapted model still performs significantly better.

(c) Success case of video 014, adaptation increases APG_w^m from 42.53 to 51.75. The base model cannot properly detect the vehicles in the shadow. However, in the object mask they are more clearly outlined and can be identified by the adapted model.

(d) Failure case of video 051, adaptation decreases APG_w^m from 32.84 to 30.96. Adaptation seems to make the model produce less bounding boxes at the heavily-occluded regions (*e.g.* the parked cars at middle-right of the frame). This is probably due to the fact that pseudo-labeling cannot identify each object accurately at those regions.

(e) Failure case of video 093, adaptation decreases APG_w^m from 54.91 to 50.08. The base model already performs very well, likely because the background is covered by snow and the objects are well-separated. Pseudo-labeling introduces some false positive bounding boxes, which is learned by the adapted model. Tuning the pseudo-labeling hyper-parameters can probably fix this issue.

Figure E. Some success and failure cases of the best proposed method. For each case, from left to right, 5 images are presented: i) 1 sample frame from the evaluation split with its non-annotation masks and annotation bounding boxes, ii) its corresponding object mask image, iii) the detection output of the base model on the evaluation frame, iv) the detection output of the adapted model on the evaluation frame, and v) 1 sample frame from the training split with its pseudo-bounding boxes from pseudo-labeling. For the detection output, we only show object bounding boxes with confidence score higher than 0.5.