Supplementary Material: Learning Transformations To Reduce the Geometric Shift in Object Detection

Vidit Vidit[1] Martin Engilberge[1] Mathieu Salzmann[1,2]
CVLab, EPFL[1], ClearSpace SA[2]
firstname.lastname@epfl.ch

A.1. Transformations through Homography

We use homography to introduce varied perspective transformations so that they can distort the same image regions differently as seen in Fig. A.1. This helps the detector to learn robust object features and simultaneously optimize an aggregator with a different set of homographies which can bridge the gap between two domains.

A.2. Feature Maps Activation

We show in Fig. A.2 how different homographies generate activation in the feature maps. Not all homographies look at the same image region, therefore the task of the aggregator is to bring in the activations from different transformations together.

A.3. Other Aggregator Architecture

We implement aggregator using standard functions to combine $\{\mathcal{F}_{\mathcal{H}_i}\}_{i=1}^{N}$. Tab. A.1 illustrates this study for FoV adaptation, where the training is done under mean teacher formalism to learn $|\mathcal{T}| = N = 5$. We see that these non-learnable aggregators are able to outperform MT baseline (Sec. 5.4, in the main paper) suggesting that including transformations helps to bridge the geometric shifts.

Function	Car AP@0.5
sum	$78.1_{\pm 0.14}$
mean	$78.7_{\pm 0.05}$
max	$78.7_{\pm 0.12}$
min+max	$78.9_{\pm 0.43}$
MT	78.3
Ours	$79.9_{\pm 0.14}$

Table A.1. Aggregator Architecture without learnable parameters

A.4. Why learning transformations?

We compare our approach against (a) a fixed set of random transformations used throughout the training and inference; (b) sampling random homographies throughout training and inference; (c) sampling random homographies throughout training and identity homographies during inference. We use the FoV adaptation task with $N = 5$ homographies and keep the original training step (i) and step (ii) (Sec. 4.2 in main paper) unchanged for all cases. Our approach achieves 79.9 AP vs (a) 78.2, (b) 79.3, (c) 77.7. This shows that the choice of homographies significantly impacts performance. Interestingly, (b) can be seen as an ensemble method that outperforms the **MT** and **AT** baselines (Tab. 1 in main paper). Our proposed approach nonetheless achieves better performance by learning the transformations. This study further evidences the importance of transformations and the need to learn them. Additionally, we can achieve better inference speed w.r.t randomly sampling transformations.

A.5. FoV Decreasing Results

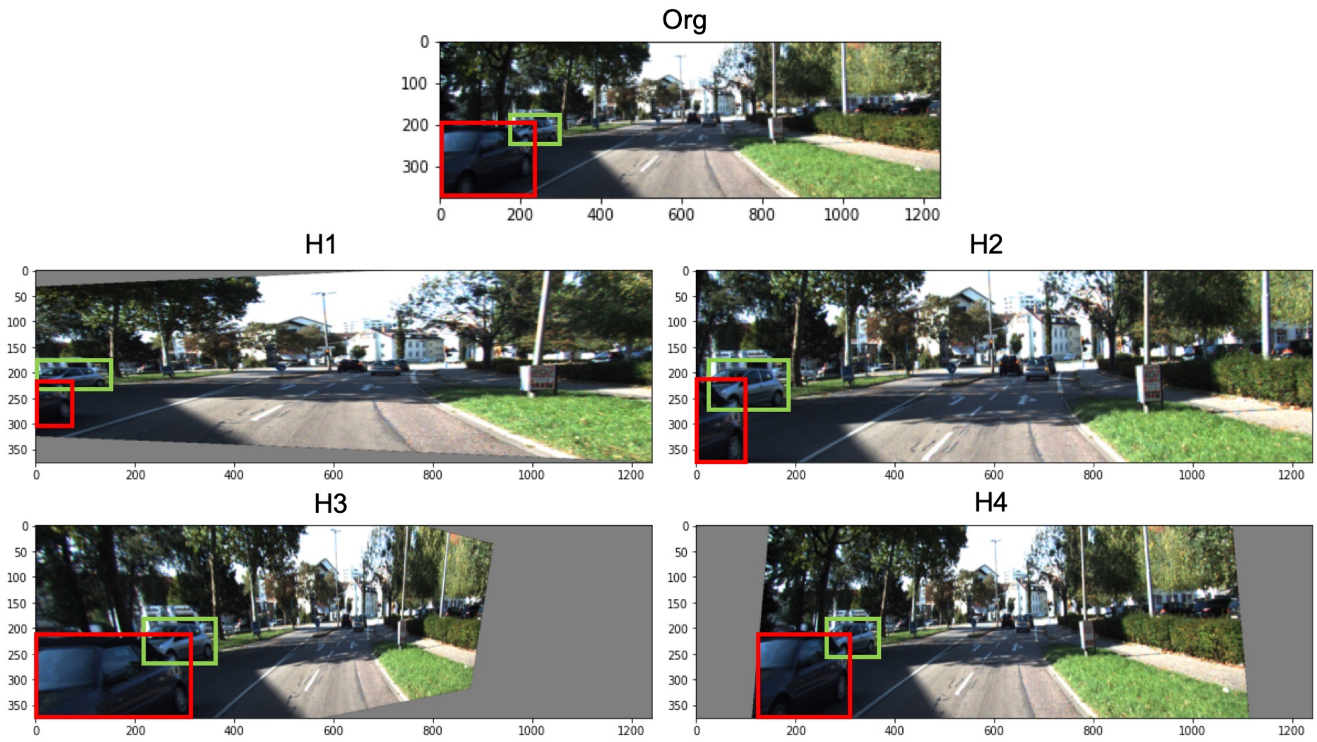
We provide additional results for the decreasing-FoV case, i.e., KITTI(source) to Cityscapes(target): (a) **MT**: 47.1, (b) **MT+PIT**: 48.5, (c) **Ours** ($N = 5$): 49.3. These results further show the effectiveness of our method.

A.6. Diversity in T

In order to show that diverse transformations are learned, we set $\mathcal{H}_i = I$ and train our mean teacher formulation. Fig. A.4 shows diverse set of transformations learned in FoV adaptation task. Even though we do not enforce diversity among homographies, it is learned through our approach.

A.7. Evolution of T

We provide qualitative results for $\mathcal{T}$ learned in FoV and Viewpoint adaptation, Fig. A.5 and Fig. A.8, respectively. The qualitative results for the same adaptation task can be seen in Fig. A.6 and Fig. A.7, respectively.


Org

H1

H2

H3

H4

Figure A.1. **Transformations**: Here we demonstrate how the two objects in the original image undergo different perspective transformations. Our task is to learn robust object features under such transformations and use them to bring the two domains closer while being agnostic to the camera parameters. We train with a multiple set of transformations to change the same image region differently. With our trainable aggregator, we can then combine features from different regions to help in improving the detector's performance.

A.8. Hyperparameter details

Augmentations. We use Detectron2 [2]s implementation for random crop and torchvision [1] for color jittering.

Kind	Details
Random Crop	Relative Range: $[0.3, 1]$
Color Jitter	Brightness=.5, Hue=.3

Table A.2. Augmentations

FasterRCNN [1] training. We train our base network with random crop strategy on with only source data, which is Cityscapes for both the adaptation tasks. The trained model achieves 74.7 and 58.4 AP@0.5 score on the source domain validation set for *car* and *person* detection, respectively.

Mean Teacher Training For our mean teacher setup (Sec. 4.2, in the main paper), we choose $\tau = 0.6$ as the confidence threshold for the pseudo-labels and evaluate contribution of target domain loss for different λ. Fig. A.11 summarizes this study. We see that method performs worse when we have equal contribution from both source and target domain loss $\lambda = 1$, as the false positives in the target domain quickly deteriorate the training. Fig. A.12, evaluation for different values of τ.

A.9. Architecture details

Our aggregator architecture consists of three convolution layers along with BatchNorm and Relu layers after each convolution. Tab. A.3 shows the details of different layers. Here, $C = 1024$ corresponds to the output of the feature extractor.

References

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 2

[2] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2

[1] https://pytorch.org/vision/stable/transforms.html

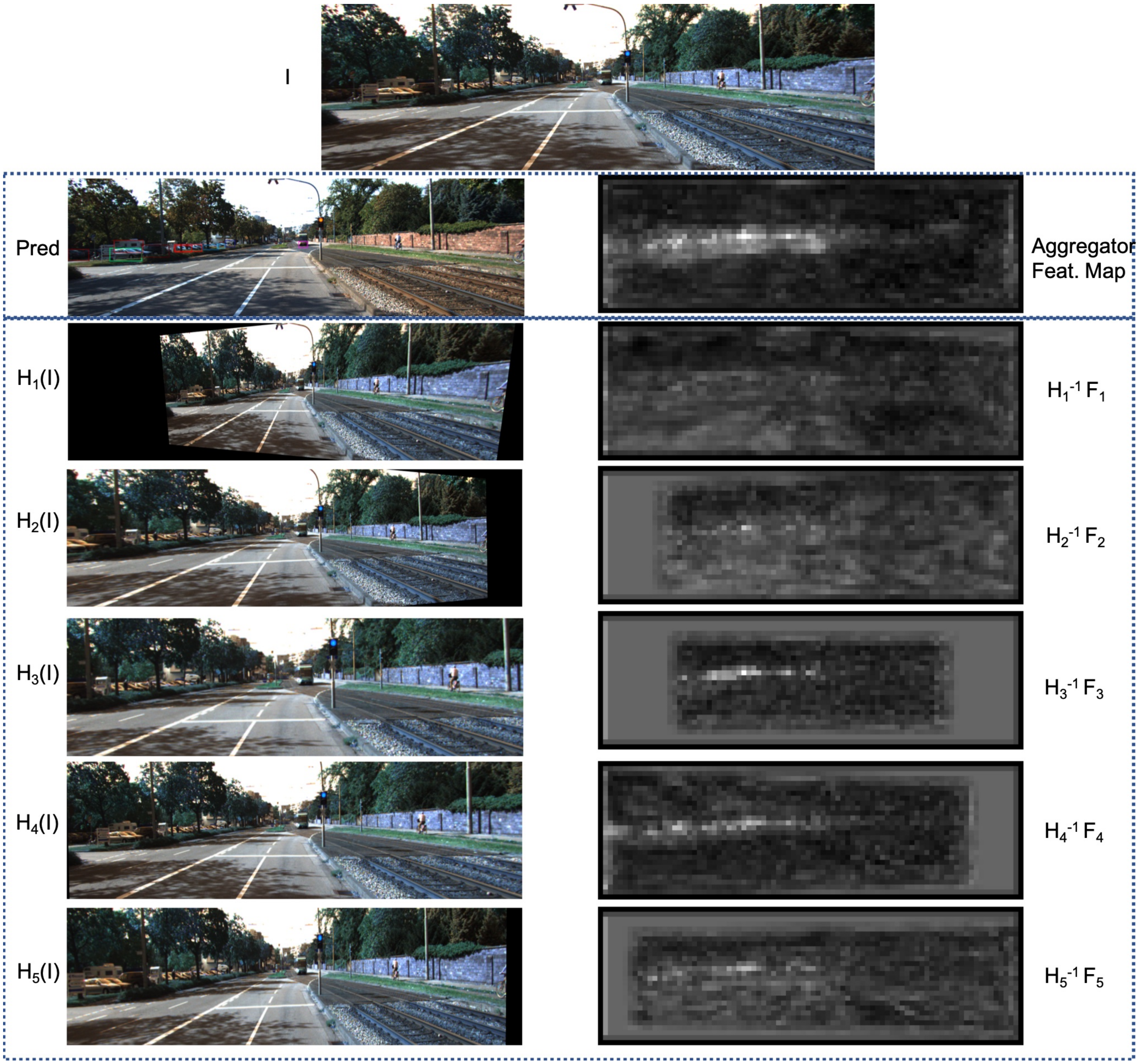


Figure A.2. **Feature Maps**: Top row: predictions of our network and feature map after aggregator. Left column: Image I, transformed by 5 learnt homographies; Right Column: Feature maps F warped by corresponding H^{-1} which are input to aggregator. Each transform distorts the image regions differently. Most of the *cars* are on the left side and of small size in the image. H_1 distorts the left side leading to no activation($H_1^{-1}F_1$) for the object. H_3 which causes zoom-in effect has the strongest activation as the smaller objects are visible better here. Overall aggregator feature map contains activation from the region where the objects exist. The aggregator has learnt how to combine regions with activations under different homographies. The feature maps are generated by taking maximum over channel dimension.

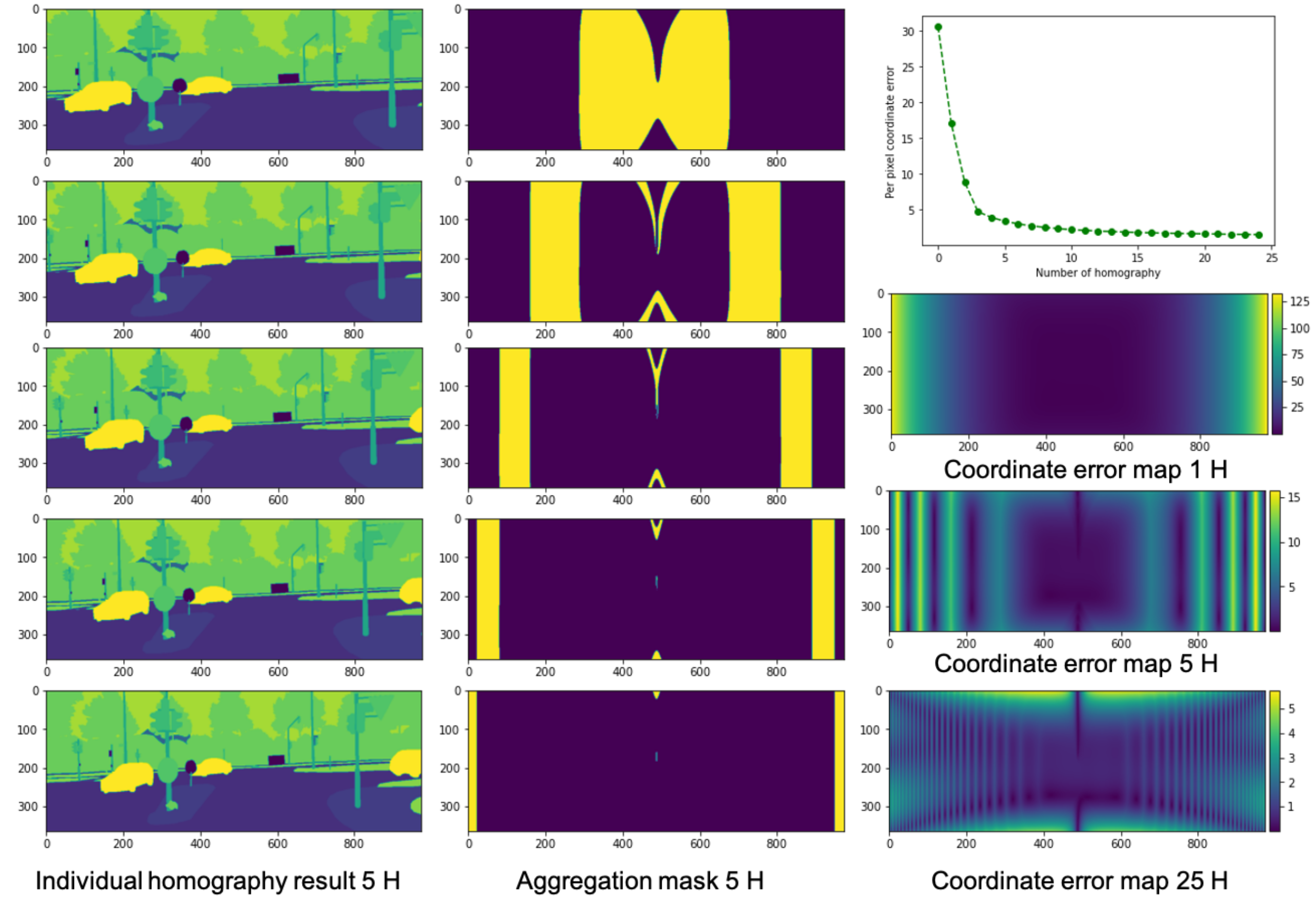


Figure A.3. **Approximating PIT with homographies.** Left column: Visualization of each homography use to approximate PIT with 5 transforms; the top one is the identitity, and the following ones are in order of increasing compression. Center column: Contribution of each homography to the final remapping. Right column: The top figure shows the per pixel coordinate error when compared to the PIT remapping as a function of the number of homographies used in the approximation; the three bottom figures depict the coordinate error maps for 1, 5, and 25 homographies used to approximate PIT (note the scale change in pixel coordinate error).

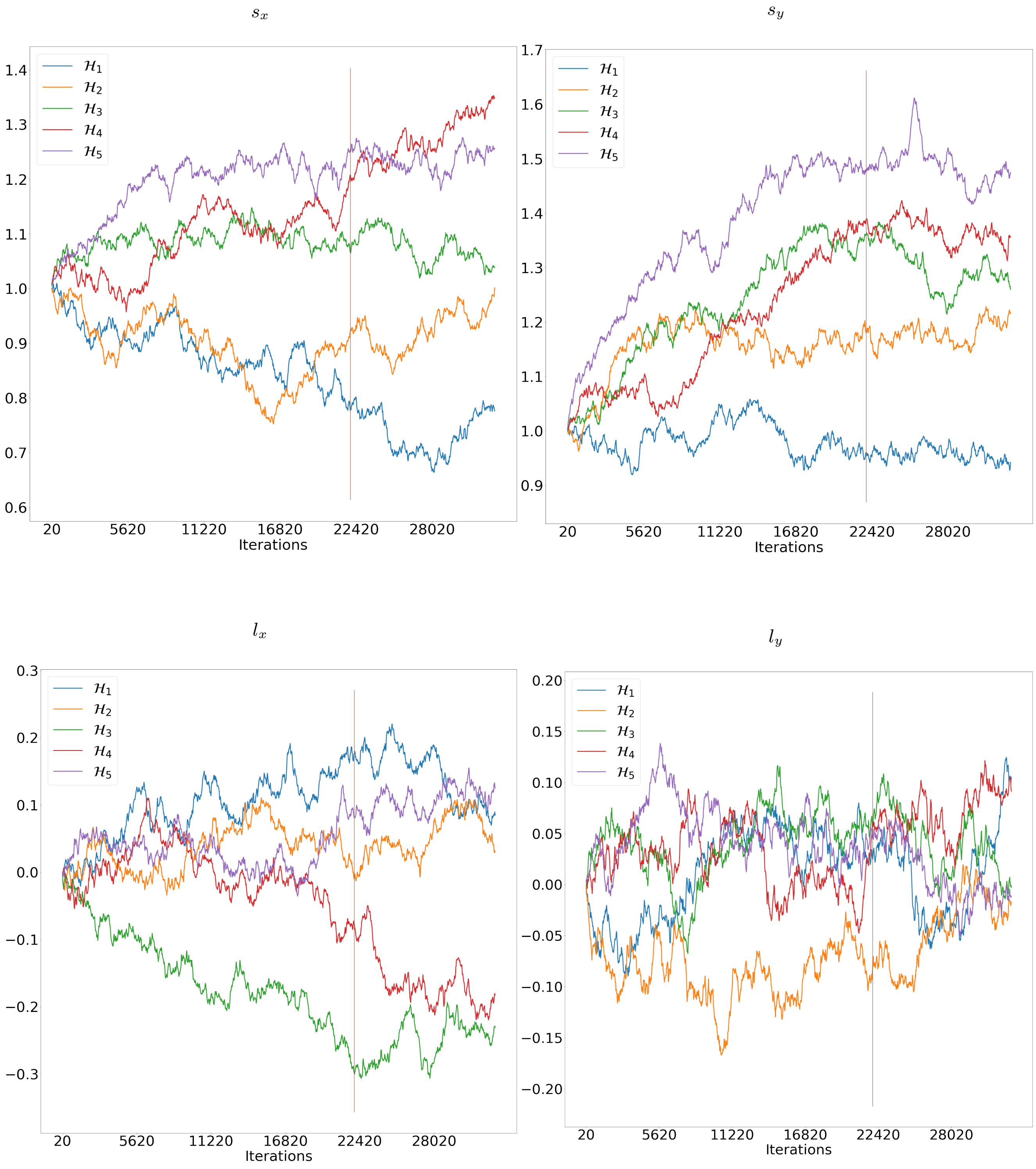


Figure A.4. **Diversity in $\mathcal{T}$:** We train $|\mathcal{T}| = 5$ initialized with $\mathcal{H}_i = I$. Homographies parameterized by s_x, s_y, l_x, l_y evolve as the training proceeds and tend to become diverse. Each homography is shown in different color. Even though we do not enforce any diversity, our approach learns diverse set of transformations. With these learned homorgraphies, we achieve 79.5 AP@0.5 score for FoV adaptation task. The best score is achieved at iteration = 22k shown with the vertical line.

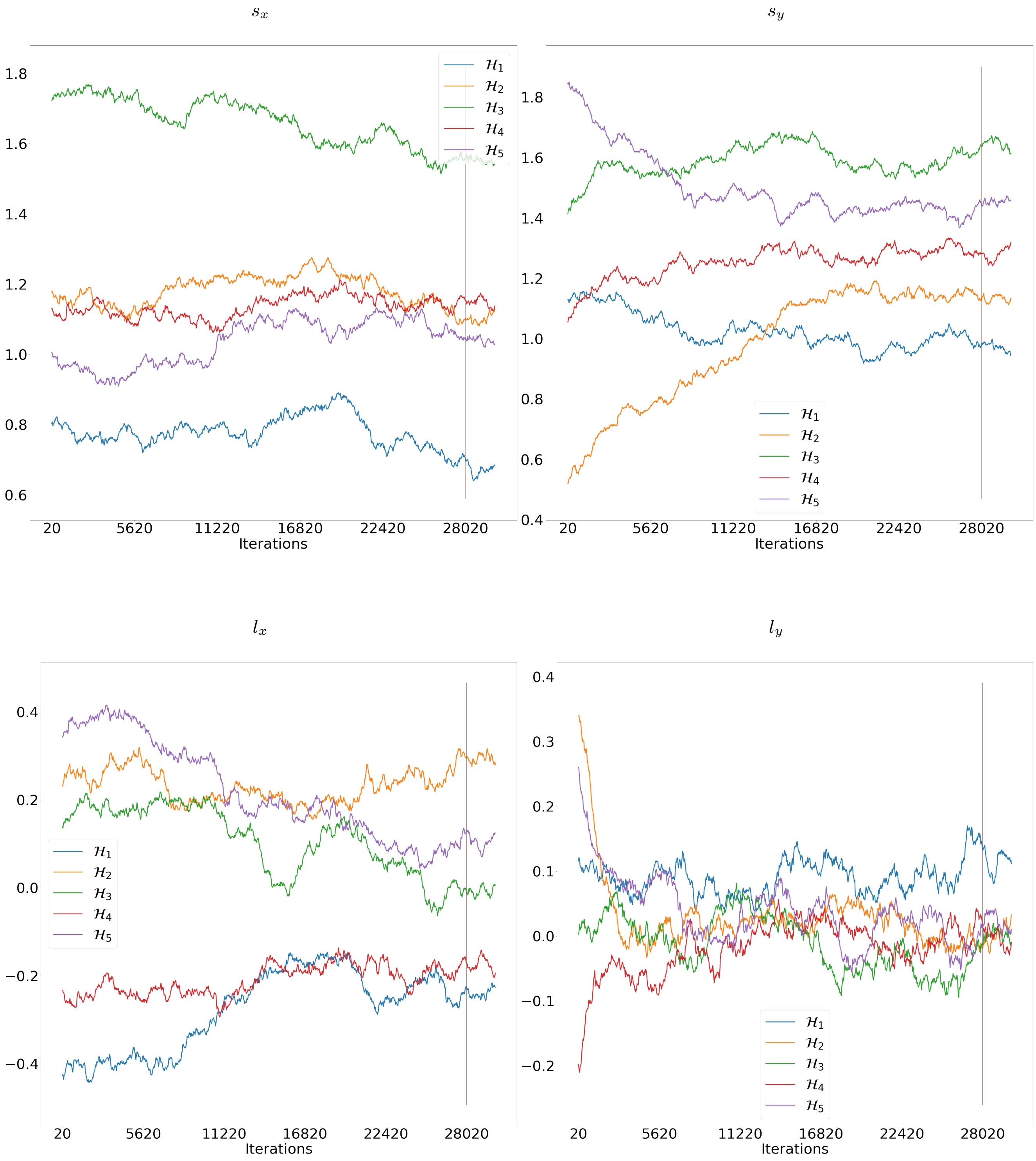


Figure A.5. Quantitative results for the corresponding results in Figure A.6. The randomly initialized transforms, parameterized by s_x, s_y, l_x, l_y, evolve to achieve the best score at 28k iterations (shown by the vertical bar). The colors represent different homographies. Some set of parameters converges to similar value but overall each homography is unique.

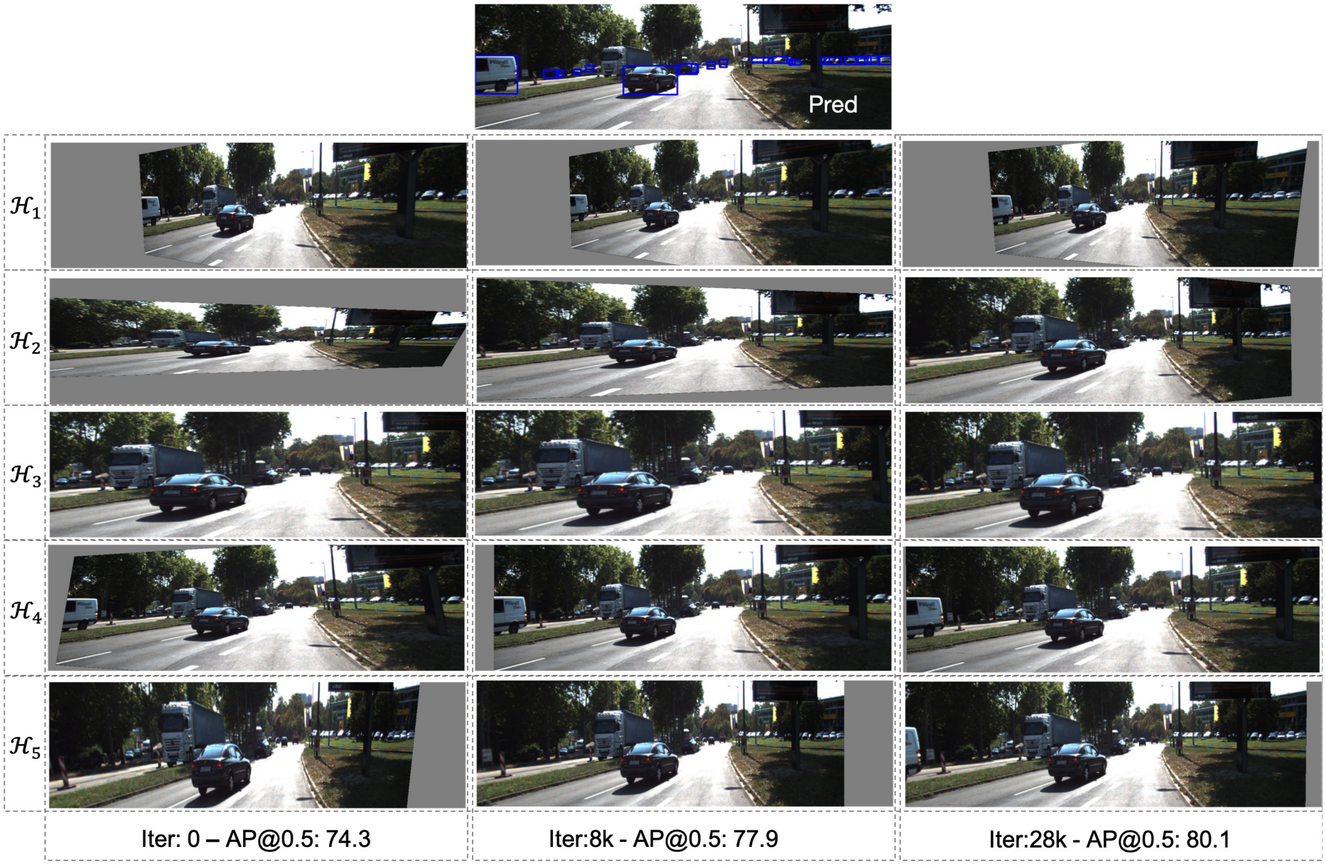


Figure A.6. **FoV adaptation:** The randomly initialized homographies evolve as the training progresses to improve the overall AP score. We train with 5 homographies and show how they transform an image for the corresponding FoV adaptation task.

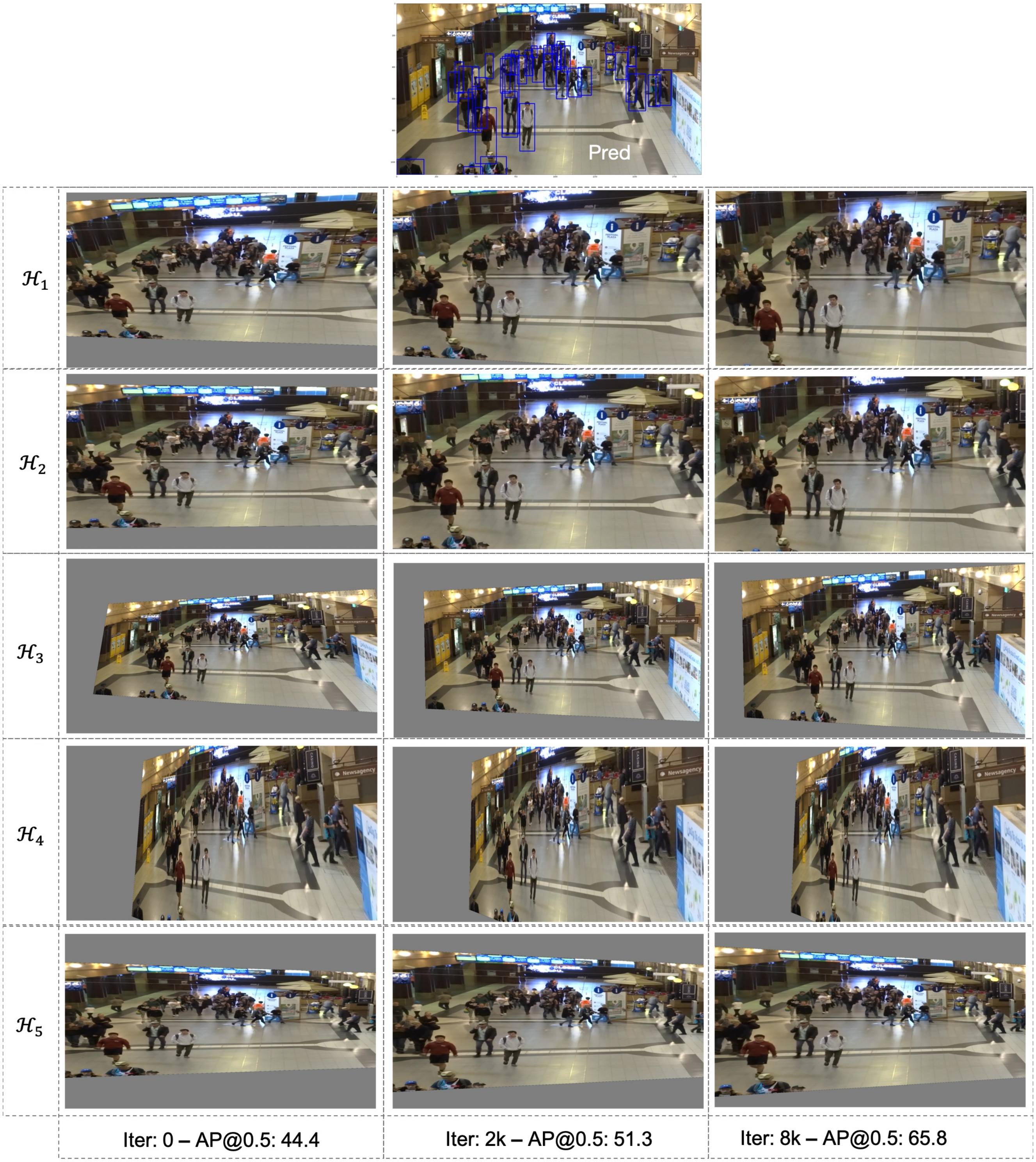


Figure A.7. **Viewpoint adaptation:** The randomly initialized homographies evolve as the training progresses to improve the overall AP score. We train with 5 homographies and show how they transform an image for the corresponding viewpoint adaptation task.

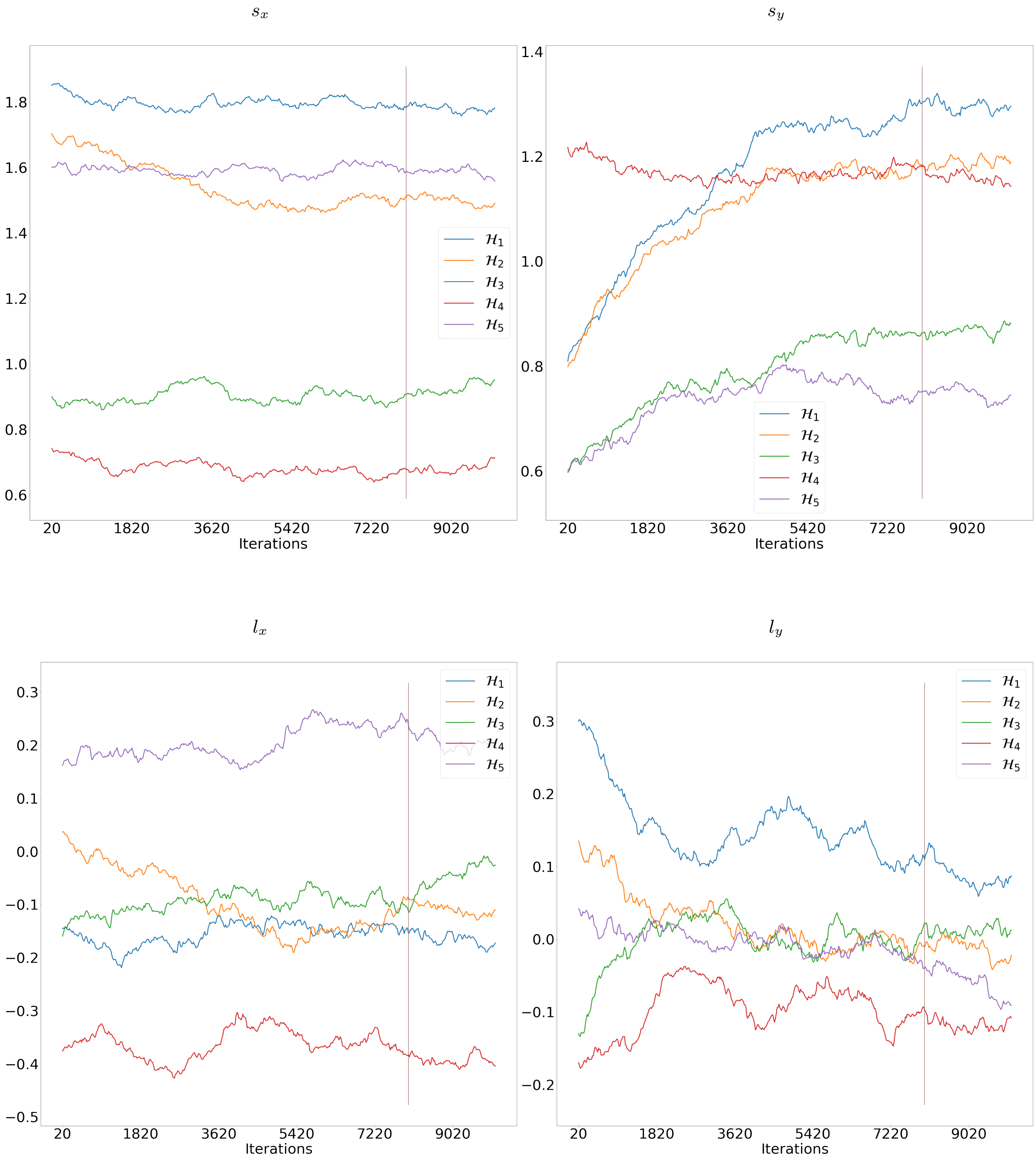


Figure A.8. Quantitative results for the corresponding results in Figure A.7. The randomly initialized transforms, parameterized by s_x, s_y, l_x, l_y, evolve to achieve the best score at 8k iterations (shown by the vertical bar). The colors represent different homographies. Some s_y parameters start at a similar value but eventually diverge.

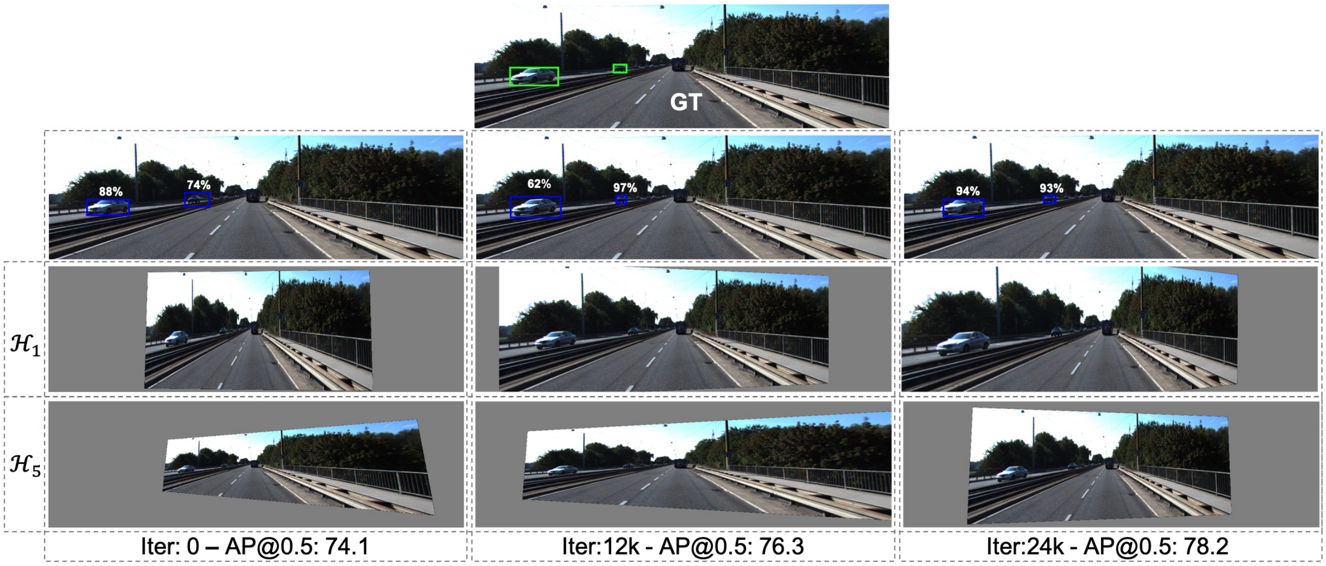


Figure A.9. **Evolution of $\mathcal{T}$.** We showcase how two homographies, $\mathcal{H}_1$ and $\mathcal{H}_5$, evolve across the training iterations and influence the prediction scores. Starting from random homographies at iteration 0, the transformations converge to homographies suited for FoV adaptation. The detection scores consequently increase throughout the training process. Moreover, this increase in detection score is reflected in the overall AP@0.5 score, which jumps from 74.1 to 78.2.

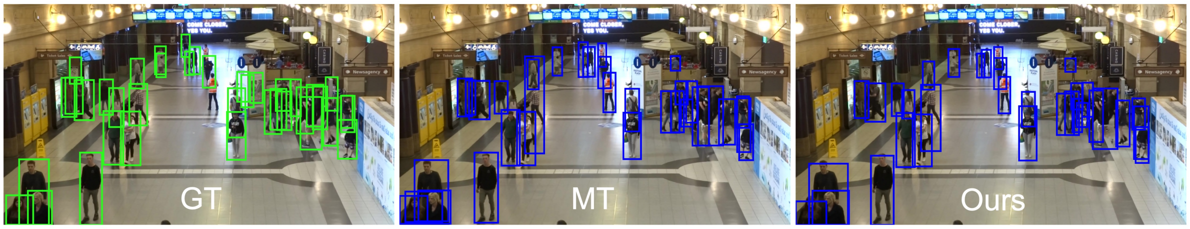


Figure A.10. **Viewpoint Adaptation: Qualitative Results.** We visualize results for viewpoint adaptation between Cityscapes and MOT20-02. The left image depicts the ground truth, the middle one the results of Mean Teacher adaptation, and the right one those of our approach. Our approach recovers more detections (e.g., the woman near the stroller in the center-left) while having fewer false positives (overlapping box in bottom-left corner of the MT results).

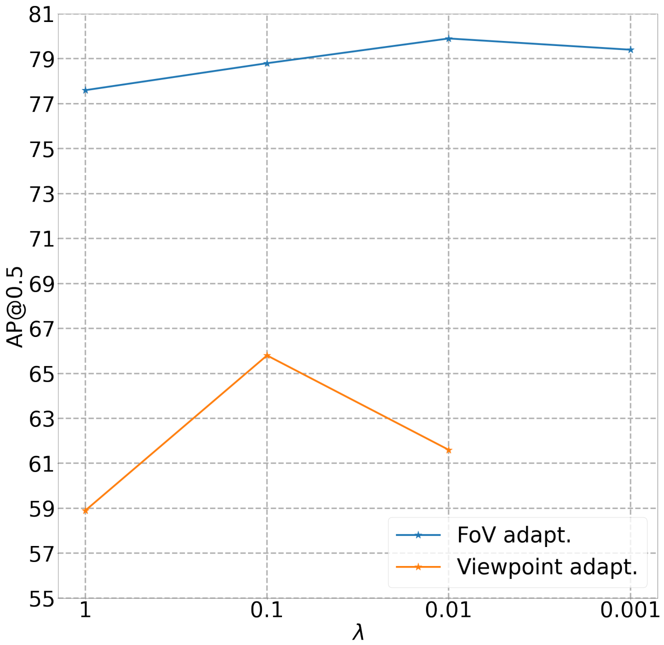


Figure A.11. **Study on** λ for $\tau = 0.6$, $|\mathcal{T}| = 5$

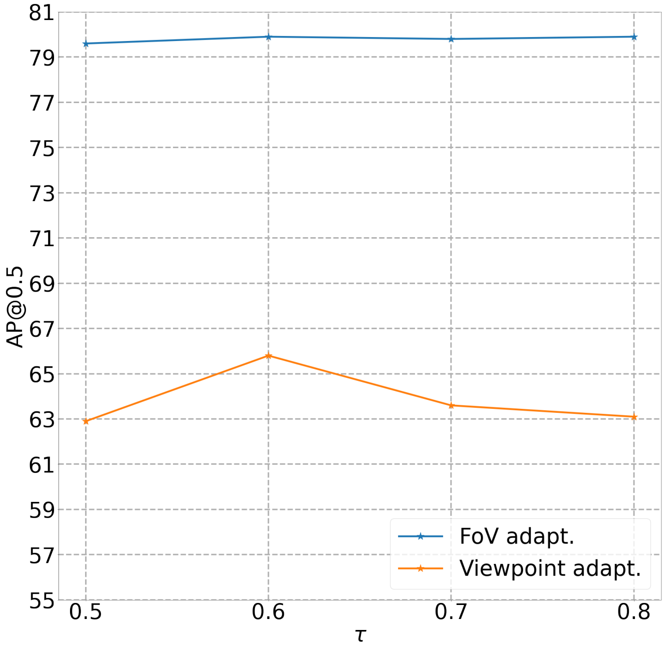


Figure A.12. **Study on** τ for FoV and Viewpoint adaptation using $\lambda = 0.01, 0.1$, respectively. Here ,$|\mathcal{T}| = 5$ is used for the study.

Table A.3. Aggregator Architecture for $|\mathcal{T}| = N$

	# Channels	
Layer	Input	Output
Conv2d 3×3	$N \times C$	$N \times C/2$
BatchNorm + Relu	$N \times C/2$	$N \times C/2$
Conv2d 3×3	$N \times C/2$	C
BatchNorm + Relu	C	C
Conv2d 1×1	C	C
BatchNorm + Relu	C	C