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

Abstract

The performance of modern object detectors drops when the test distribution differs from the training one. Most of the methods that address this focus on object appearance changes caused by, e.g., different illumination conditions, or gaps between synthetic and real images. Here, by contrast, we tackle geometric shifts emerging from variations in the image capture process, or due to the constraints of the environment causing differences in the apparent geometry of the content itself. We introduce a self-training approach that learns a set of geometric transformations to minimize these shifts without leveraging any labeled data in the new domain, nor any information about the cameras. We evaluate our method on two different shifts, i.e., a camera's field of view (FoV) change and a viewpoint change. Our results evidence that learning geometric transformations helps detectors to perform better in the target domains.

1. Introduction

While modern object detectors [1, 2, 17, 23, 24] achieve impressive results, their performance decreases when the test data depart from the training distribution. This problem arises in the presence of appearance variations due to, for example, differing illumination or weather conditions. Considering the difficulty and cost of acquiring annotated data in the test (i.e., target) domain, Unsupervised Domain Adaptation (UDA) has emerged as the standard strategy to address such scenarios [3, 4, 9, 26, 38].

In this context, much effort has been made to learn domain invariant features, such that the source and target distributions in this feature space are similar. This has led to great progress in situations where the appearance of the objects changes drastically from one domain to the other, as in case of real-to-sketch adaptation (e.g., Pascal VOC [10] to Comics [15]), or weather adaptation (e.g., Cityscapes [6] to Foggy Cityscapes [27]). Nevertheless, such object appearance changes are not the only sources of domain shifts. They can also have geometric origins. For example, as shown in Fig. 1, they can be due to a change in camera viewpoint or field-of-view (FoV), or a change of object scale due to different scene setups. In practice, such geometric shifts typically arise from a combination of various factors, including but not limited to the ones mentioned above.

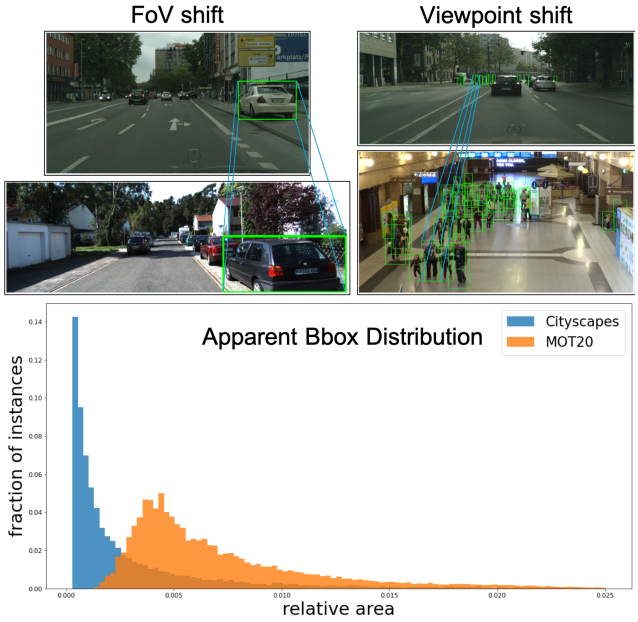

Figure 1. **Geometric shifts.** **(Left)** Due to a different FoV, the cars highlighted in green, undergo different distortions even though they appear in similar image regions. **(Right)** Different camera viewpoints (front facing vs downward facing) yield different distortions and occlusion patterns for pedestrian detection. **(Bottom)** The distributions of pedestrian bounding box sizes in Cityscapes [6] and MOT [8] differ significantly as the pedestrians are usually far away or in the periphery in Cityscapes. The top images are taken from Cityscapes [6], and the bottom-left and right ones from KITTI [12] and MOT [8], respectively.

In this paper, we introduce a domain adaptation approach tackling such geometric shifts. To the best of our knowledge, the recent work of [13] constitutes the only attempt at considering such geometric distortions. However, it introduces a method solely dedicated to FoV variations, assuming that the target FoV is fixed and known. Here, we de-

velop a more general framework able to cope with a much broader family of geometric shifts.

To this end, we model geometric transformations as a combination of multiple homographies. We show both theoretically and empirically that this representation is sufficient to encompass a broad variety of complex geometric transformations. We then design an *aggregator* block that can be incorporated to the detector to provide it with the capacity to tackle geometric shifts. We use this modified detector to generate pseudo labels for the target domain, which let us optimize the homographies so as to reduce the geometric shift.

Our contributions can be summarized as follows. (i) We tackle the problem of general geometric shifts for object detection. (ii) We learn a set of homographies using unlabeled target data, which alleviates the geometric bias arising in source-only training. (iii) Our method does not require prior information about the target geometric distortions and generalizes to a broad class of geometric shifts. Our experiments demonstrate the benefits of our approach in several scenarios. In the presence of FoV shifts, our approach yields similar performance to the FoV-dedicated framework of [13] but without requiring any camera information. As such, it generalizes better to other FoVs. Furthermore, we show the generality of our method by using it to adapt to a new camera viewpoint in the context of pedestrian detection.Our implementation can be accessed at https://github.com/vidit09/geoshift.

2. Related Work

Unsupervised Domain Adaptation (UDA). UDA for image recognition [11, 21, 22, 30, 32, 35, 36] and object detection [3, 4, 9, 20, 26, 38] has made a great progress in the past few years. The common trend in both tasks consists of learning domain invariant features. For object detection, this entails aligning the global (e.g., illumination, weather) and local (foreground objects) features in the two domains. In this context, [3, 5, 26, 28] align image- and instance-level features in the two domains via adversarial learning [11]; [33] learns category-specific attention maps to better align specific image regions; [38] clusters the proposed object regions using k-means clustering and uses the centroids for instance-level alignment. While this successfully tackles domain shifts caused by object appearance variations, it fails to account for the presence of shifts due to the image capture process itself, such as changes in camera intrinsics or viewpoint. The only initial step at considering a geometric shift is the work of [13], which shows the existence of an FoV gap in driving datasets [6, 12] and proposes a Position Invariant Transform (PIT) that corrects the distortions caused specifically by an FoV change. In essence, PIT undistorts the images by assuming knowledge of the target FoV. By contrast, here, we introduce an approach that generalizes to a broad family of geometric shifts by learning transformations without requiring any camera information.

Self-training. Self-training, generally employed in the semi-supervised setting, offers an alternative to learning domain-invariant features and utilize unlabeled data to improve a detector's performance. In this context, [29] uses a student-teacher architecture where the teacher model is trained with supervised data and generates pseudo-labels on unannotated data. These pseudo-labels are then used to train a student model. While effective in the standard semi-supervised learning scenario, the quality of the pseudo-labels obtained with this approach tends to deteriorate when the labeled and unlabeled data present a distribution shift. [9, 20] have therefore extended this approach to domain adaptation by using the Mean Teacher strategy of [31] to generate reliable pseudo-labels in the target domain. Other approach include the use of CycleGAN [37] generated images to train an unbiased teacher model [9], and that of different augmentation strategies to generate robust pseudo-labels [20]. Our approach also follows a self-training strategy but, while these works focus on object appearance shifts, we incorporate learnable blocks to address geometric shifts. As shown in our experiment, this lets us outperform the state-of-the-art AdaptTeacher [20].

Learning Geometric Transformations. End-to-end learning of geometric transformations has been used to boost the performance of deep networks. For example, Spatial Transformer Networks (STNs) [16] reduce the classification error by learning to correct for affine transformations; deformable convolutions [7] model geometric transformations by applying the convolution kernels to non-local neighborhoods. These methods work well when annotations are available for supervision, and make the network invariant to the specific geometric transformations seen during training. Here, by contrast, we seek to learn transformations in an unsupervised manner and allow the network to generalize to unknown target transformations.

3. Modeling Geometric Transformations

In the context of UDA, multiple geometric differences can be responsible for the gap between the domains. Some can be characterized by the camera parameters, such as a change in FoV (intrinsic) or viewpoint (extrinsic), whereas others are content specific, such as a difference in road width between different countries. Ultimately, the geometric shift is typically a combination of different geometric operations. Since the parameters of these operations are unknown, we propose to bridge the domain gap by learning a geometric transform. Specifically, we aggregate the results of multiple perspective transforms, i.e., homographies, to obtain a differentiable operation that can emulate a wide variety of geometric transforms.

3.1. Theoretical Model

Let us first show that, given sufficiently many homographies, one can perfectly reproduce any mapping between $\mathbb{R}^2 \setminus (0,0)$ and $\mathbb{R}^2$.

Single homography for a single point. First, we show that a single homography with 4 degrees of freedom can map a point $p \in \mathbb{R}^2 \setminus (0,0)$ to any other point in $\mathbb{R}^2$. To this end, let

$$H = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ l_x & l_y & 1 \end{bmatrix} \quad (1)$$

be a homography, with (s_x, s_y) the scaling factors on the x- and y-axis, respectively, and (l_x, l_y) the perspective factors in x and y, respectively. For any destination point $d \in \mathbb{R}^2$, there exists a set of parameters (s_x, s_y, l_x, l_y) such that $d = H \times p$. One such set is $(\frac{d_x}{p_x}, \frac{d_y}{p_y}, 0, 0)$.

Emulating any geometric transformation Now that we have shown that a single homography can move a point to any other point in $\mathbb{R}^2$, we describe a simple protocol to emulate any geometric transform. Given an unknown geometric transform $T : \mathbb{R}^2 \setminus (0,0) \rightarrow \mathbb{R}^2$, we aim to emulate T with a set of homographies. In general, for an image $\mathbf{I} \in \mathbb{R}^{3 \times h \times w}$, we can restrict the domain of T to only image coordinates. To this end, we can define a set of homographies $H_i \in \mathbb{H}$ for i in $\{1, 2, 3, ..., h \times w\}$, where the parameters of H_i are chosen to mimic the transform T for location i of the image. In this protocol, the aggregation mechanism is trivial since each homography is in charge of remapping a single pixel coordinate of the original space.

While this works in theory, this is of course not viable in practice since it would require too many homographies. With a smaller number of homographies, each transform needs to remap multiple points, and a more sophisticated aggregation mechanism is required. Specifically, the aggregation mechanism needs to select which transform is in charge of remapping which point. In the next section, we empirically show that this strategy lets us closely approximate the spherical projection mapping used in PIT [13].

3.2. Approximating PIT with Homographies

To demonstrate the possibility offered by aggregating multiple homographies, we design an approximation of PIT using only homographies. PIT proposes to correct for an FoV gap by remapping images to a spherical surface. During this transformation, regions further from the center of a scene are compressed with a higher ratio. This variable compression of the space cannot be reproduced by a single homography transformation. To overcome this limitation, we combine the results of multiple homographies that all have different compression rates (scaling parameters). For the aggregation mechanism, we use the optimal strategy by selecting for each pixel the homography that approximates best the PIT mapping. As shown in Fig. 2, this combination closely approximates the PIT results with only 5 homographies. Further analysis of these experiments is available in the supplementary material in Fig. A.3.

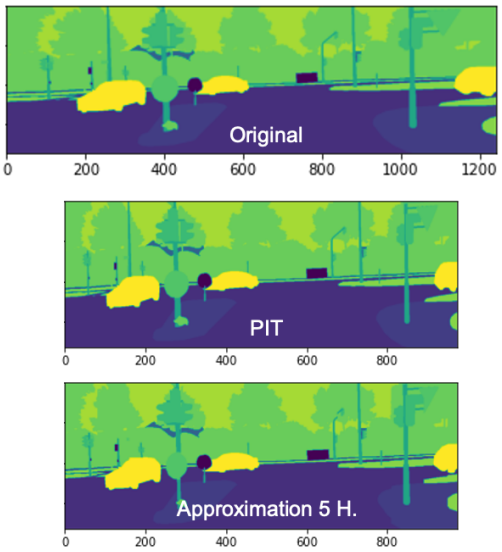

Figure 2. **Approximating PIT with homographies.** We show the original image (top), the PIT [13] correction (middle), and our approximation of PIT using 5 homographies. Note that 5 homographies are sufficient to closely match the PIT spherical correction.

3.3. Homographies in a Learning Setup

In the two previous sections, we have demonstrated both theoretically and empirically the flexibility of aggregating homographies. This makes this representation an ideal candidate for domain adaptation since the geometric shift between the domains is unknown and can be a combination of different transforms, such as FoV change, viewpoint change, camera distortion, or appearance distortion. As will be discussed in the next section, by learning jointly the set of perspective transforms and the aggregation mechanism on real data, our model can reduce the geometric shift between the two domains without prior knowledge about this domain gap.

4. Method

Let us now introduce our approach to reducing the geometric shift in object detection. Following the standard UDA setting, let $D_s = \{(I_s, B_s, C_s)\}$ be a labeled source dataset containing images $I_s = \{I_s^i\}_1^{N_s}$ with corresponding object bounding boxes $B_s = \{b_s^i\}_1^{N_s}$ and object classes $C_s = \{c_s^i\}_1^{N_s}$. Furthermore, let $D_t = \{I_t\}$ denote an unlabeled target dataset for which only images $I_t = \{I_t^i\}_1^{N_t}$

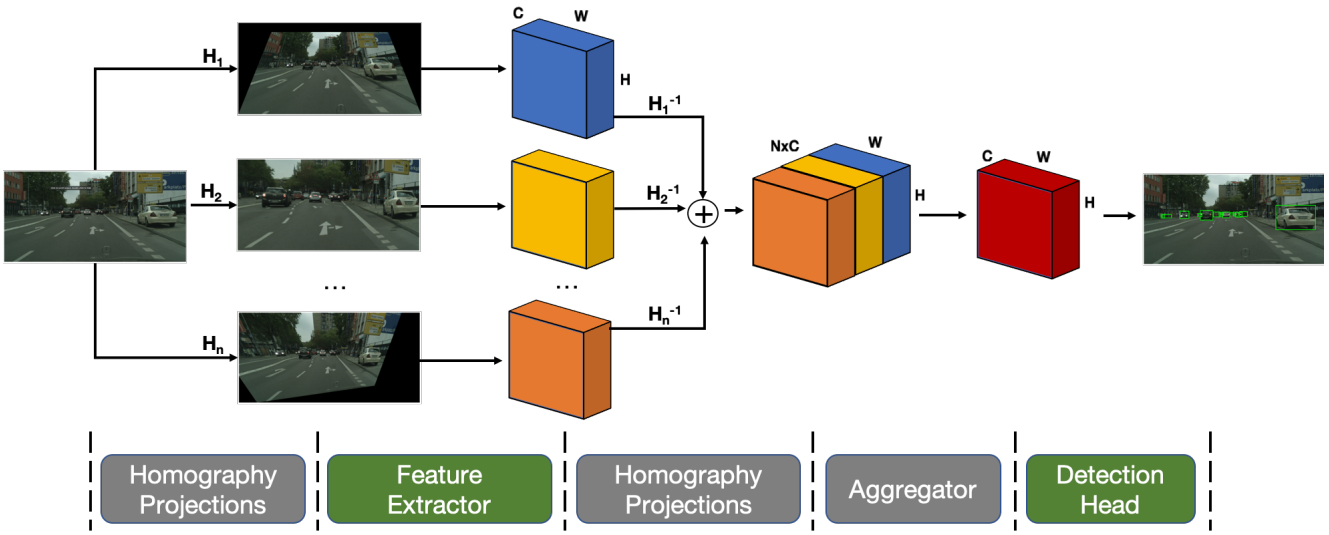

Figure 3. **Architecture**: The input image is first transformed by a set of trainable homographies. The feature maps extracted from the transformed images are then unwarped by the inverse homographies to achieve spatial consistency. We then combine the unwarped feature maps using a trainable *aggregator*, whose output is passed to a detection head. The blocks shown in green correspond to standard FasterRCNN operations. The $\oplus$ symbol represents the concatenation operation.

are available, without annotations. Here, we tackle the case where the two domains differ by geometric shifts but assume no knowledge about the nature of these shifts. Below, we first introduce the architecture we developed to handle this and then our strategy to train this model.

4.1. Model Architecture

The overall architecture of our approach is depicted in Fig. 3. In essence, and as discussed in Sec. 3, we characterize the geometric changes between the source and target data by a set of transformations $\mathcal{T} = \{\mathcal{H}_i\}_1^N$. Each $\mathcal{H}_i$ in $\mathcal{T}$ is a homography of the same form as in Eq. (1). For our method to remain general, we assume the transformations to be unknown, and our goal, therefore, is to learn $\mathcal{T}$ to bridge the gap between the domains. This requires differentiability w.r.t. the transformation parameters, which we achieve using the sampling strategy proposed in [16].

As shown in Fig. 3, the input image is transformed by the individual homographies in $\mathcal{T}$, and the transformed images are fed to a modified FasterRCNN [24] detector. Specifically, we extract a feature map $\mathcal{F}_{\mathcal{H}_i} \in \mathbb{R}^{H\times W\times C}$ for each transformed image via a feature extractor shared by all transformations. To enforce spatial correspondence between the different $\mathcal{F}_{\mathcal{H}_i}$s, we unwarp them with $\mathcal{H}_i^{-1}$.

We then introduce an *aggregator* $\mathcal{A}_{\theta_g}$, parameterized by θ_g, whose goal is to learn a common representation given a fixed number of unwarped feature maps $\mathcal{F}'_{\mathcal{H}_i}$. To achieve this, the aggregator takes as input

$$\mathcal{G} = \mathcal{F}'_{\mathcal{H}_1} \oplus \mathcal{F}'_{\mathcal{H}_2} \oplus ... \oplus \mathcal{F}'_{\mathcal{H}_N} \in \mathbb{R}^{H\times W\times C\times N} , \quad (2)$$

where $\oplus$ represents concatenation in the channel dimension. The aggregator outputs a feature map $\mathcal{A}_{\theta_g}(\mathcal{G}) \in \mathbb{R}^{H\times W\times C}$, whose dimension is independent of the number of transformations. This output is then passed to a detection head to obtain the objects' bounding boxes and class labels.

4.2. Model Training

Our training procedure relies on three steps: (i) Following common practice in UDA, we first train the FasterRCNN detector with source-only data; (ii) We then introduce the aggregator and train it so that it learns to combine different homographies using the labeled source data; (iii) Finally, we learn the optimal transformations for adaptation using both the source and target data via a Mean Teacher [31] strategy.

Aggregator Training. To train the aggregator, we randomly sample a set of homographies $\mathcal{T} \in \mathbb{R}^{N\times 4}$ in each training iteration.[1] This gives the aggregator the ability to robustly combine diverse input transformations but requires strong supervision to avoid training instabilities. We, therefore, perform this step using the source data.

The loss function for a set of transformed images $\mathcal{T}(I_s)$ is then defined as in standard FasterRCNN training with a combination of classification and regression terms [24]. That is, we train the aggregator by solving

$$\min_{\theta_g} \mathcal{L}_{cls}(\mathcal{T}(I_s)) + \mathcal{L}_{reg}(\mathcal{T}(I_s)) , \quad (3)$$

[1] As our homographies involve only 4 parameters, with a slight abuse of notation, we say that $\mathcal{H}_i \in \mathbb{R}^4$.

where

$$\mathcal{L}_{cls}(\mathcal{T}(I_s)) = \mathcal{L}^{rpn}_{cls} + \mathcal{L}^{roi}_{cls} \ , \tag{4}$$

$$\mathcal{L}_{reg}(\mathcal{T}(I_s)) = \mathcal{L}^{rpn}_{reg} + \mathcal{L}^{roi}_{reg} \ . \tag{5}$$

$\mathcal{L}^{rpn}_{\cdot}$ and $\mathcal{L}^{roi}_{\cdot}$ correspond to the Region Proposal Network (RPN) loss terms and the Region of Interest (RoI) ones, respectively. During this process, we freeze the parameters θ_b of the *base* network, i.e, feature extractor and detection head, which were first trained on the source data without aggregator. Ultimately, the aggregator provides the network with the capacity to encode different transformations that are not seen in the source domain. The third training step then aims to learn the best transformation for successful object detection in the target domain.

Learning the Transformations. As we have no annotations in the target domain, we exploit a Mean Teacher (MT) strategy to learn the optimal transformations. To this end, our starting point is the detector with a trained aggregator and a set of random transformations $\mathcal{T}$. The MT strategy is illustrated in Fig. 4. In essence, MT training [31] involves two copies of the model: A student model, with parameters $\theta^{st} = \{\mathcal{T}^{st}, \theta^{st}_b, \theta^{st}_g\}$, that will be used during inference, and a teacher model, with parameters $\theta^{te} = \{\mathcal{T}^{te}, \theta^{te}_b, \theta^{te}_g\}$, that is updated as an Exponentially Moving Average (EMA) of the student model. That is, the student's parameters are computed with standard backpropagation, whereas the teacher's ones are updated as

$$\theta_{te} \leftarrow \alpha\theta_{te} + (1-\alpha)\theta_{st} \ . \tag{6}$$

The student model is trained using both source and target detection losses. Since the target domain does not have annotations, the teacher model is used to generate pseudo-labels. These labels might be noisy, and hence we only keep the predictions with a confidence score above a threshold τ. Furthermore, non-maxima suppression (NMS) is used to remove the highly-overlapping bounding box predictions.

Formally, given a source image I_s and a target image I_t, the student model is trained by solving

$$\min_{\mathcal{T}^{st}, \theta^{st}_g, \theta^{st}_b} \mathcal{L}_{det}(\mathcal{T}(I_s)) + \lambda \mathcal{L}_{det}(\mathcal{T}(I_t)) \ , \tag{7}$$

where λ controls the target domain contribution and

$$\mathcal{L}_{det}(\mathcal{T}(I_s)) = \mathcal{L}_{cls}(\mathcal{T}(I_s)) + \mathcal{L}_{reg}(\mathcal{T}(I_s)) \ , \tag{8}$$

$$\mathcal{L}_{det}(\mathcal{T}(I_t)) = \mathcal{L}_{cls}(\mathcal{T}(I_t)) \ . \tag{9}$$

Similarly to [18,20], we update the student model with only the classification loss in the target domain to help stabilize training.

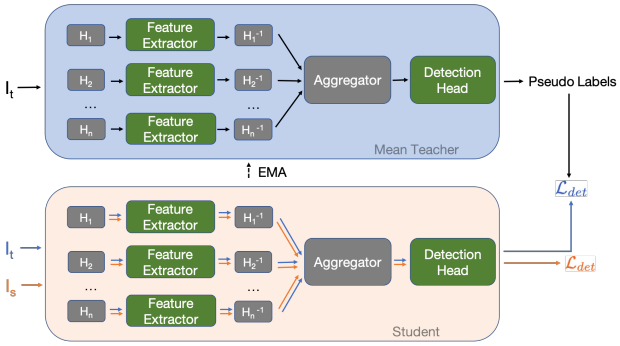

Figure 4. **Mean Teacher formalism.** The student model is trained with ground-truth labels in the source domain and pseudo labels in the target one. These pseudo labels are produced by the teacher model, which corresponds to an exponentially moving average (EMA) of the student network.

5. Experiments

We demonstrate the effectiveness and generality of our method on different geometric shifts. First, to compare to the only other work that modeled a geometric shift [13], we tackle the problem of a change in FoV between the source and target domain. Note that, in contrast to [13], we do not assume knowledge of the target FoV. Furthermore, while [13] was dedicated to FoV adaptation, our approach generalizes to other geometric shifts. We demonstrate this on the task of pedestrian detection under a viewpoint shift. We compare our method with the state-of-the-art Adapt-Teacher [20], which also uses a Mean Teacher, but focuses on appearance shifts. In the remainder of this section, we describe our experimental setup and discuss our results.

5.1. Datasets

Cityscapes [6] contains 2975 training and 500 test images with annotations provided for 8 categories (*person*, *car*, *train*, *rider*, *truck*, *motorcycle*, *bicycle* and *bus*). The average horizontal (FoVx) and vertical (FoVy) FoVs of the capturing cameras are 50°and 26°, respectively. We use this dataset as the source domain for both FoV adaptation and viewpoint adaptation.

KITTI [12] is also a street-view dataset containing 6684 images annotated with the *car* category. The horizontal (FoVx) and vertical (FoVy) FoVs of the camera are 90°and 34°, respectively. We use this dataset as target domain for FoV adaptation, as the viewpoint is similar to that of Cityscapes. Following [13], we use 5684 images for unsupervised training and 1000 images for evaluation.

MOT [8] is a multi-object tracking dataset. We use the indoor mall sequence, MOT20-02, consisting of 2782 frames annotated with the *person* category. We employ this dataset as target domain for viewpoint adaptation. We use the first

2000 frame for unsupervised training and last 782 for evaluation.

5.2. Adaptation Tasks and Metric

FoV adaptation. As in [13], we consider the case of an increasing FoV using Cityscapes as source domain and KITTI as target domain. The horizontal and vertical FoVs increase from (50°, 26°) in Cityscapes to (90°, 34°) in KITTI. Therefore, as can be seen in Fig. 1, the KITTI images have a higher distortion in the corners than the Cityscapes ones. Similarly to PIT [13], we use the *car* category in our experiments.

FoV generalization. Following PIT [13], we study the generalization of our approach to new FoVs by cropping the KITTI images to mimic different FoV changes in the horizontal direction (FoVx). Specifically, we treat FoVx = 50° as the source domain and the cropped images with FoVx = {70°, 80°, 90°} as different target domains. We evaluate our approach on *car* on these different pairs of domains.

Viewpoint adaptation. This task entails detecting objects seen from a different viewpoint in the source and target domains. We use the front-facing Cityscapes images as source domain and the downward-facing MOT ones as target one. As the MOT data depicts pedestrians, we use the bounding boxes corresponding to the *person* category in Cityscapes.[2]

Metric. In all of our experiments, we use the Average Precision (AP) as our metric. Specifically, following [13], we report the AP@0.5, which considers the predictions as true positives if they match the ground-truth label and have an intersection over union (IOU) score of more than 0.5 with the ground-truth bounding boxes.

5.3. Implementation Details

We use the Detectron2 [34] implementation of FasterRCNN [24] with a ResNet50 [14] backbone as our *base* architecture. In all of our experiments, the images are resized so that the shorter side has 800 pixels while maintaining the aspect ratio. The base network is first trained on source-only images with random cropping and random flipping augmentation for 24k iterations with batch size 8. We use the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01, scaled down by a 0.1 factor after 18k iterations. We use ImageNet [25] pretrained weights to initialize the ResNet50 backbone.

We then incorporate the *aggregator* in the trained base architecture. The aggregator architecture contains three convolutional layers with a kernel size of 3×3, and one 1×1 convolutional layer. We first train the aggregator on the source data with the base frozen and using random transformations $\mathcal{T}$. The transformations are generated by randomly sampling each $\mathcal{H}_i$ parameters as $s_x, s_y \sim \mathcal{U}_{[0.5,2.0]}, \mathcal{U}_{[0.5,2.0]}$ and $l_x, l_y \sim \mathcal{U}_{[-0.5,0.5]}, \mathcal{U}_{[-0.5,0.5]}$. We train the aggregator for 30k iterations using a batch size of 8 and the SGD optimizer with a learning rate of $1e^{-4}$.

The student and teacher models are then initialized with this detector and the random $\mathcal{T} = \{\mathcal{H}_i\}_{i=1}^{N}$. We optimize $\mathcal{T}$ using Adam [19], while the base and aggregator networks are optimized by SGD. The learning rate is set to $1e^{-3}$ and scaled down by a factor 0.1 after 10k iterations for the SGD optimizer. For the first 10k iterations in FoV adaptation and for 2k iterations for viewpoint adaptation, we only train $\mathcal{T}$ keeping base and aggregator frozen. The α coefficient for the EMA update is set to 0.99; the confidence threshold $\tau = 0.6$; $\lambda = \{0.01, 0.1\}$ for FoV and viewpoint adaptation, respectively. The Mean Teacher framework is trained using both the source and target data. We set $N = 5$, unless otherwise specified, and use a batch size of 4, containing 2 source and 2 target images. We apply random color jittering on both the source and target data as in [20, 31]. All of our models are trained on a single NVIDIA V100 GPU. A detailed hyper-parameter study is provided in the supplementary material.

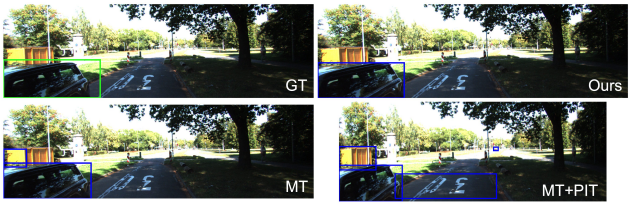

Figure 5. **FoV Adaptation: Qualitative Results.** We visualize a car detection result in the Cityscapes-to-KITTI FoV adaptation scenario. The top left image corresponds to the ground truth, the bottom left to the Mean Teacher result, which confuses the orange container with a car, the bottom right to the Mean Teacher adaptation + PIT FoV adaptation result, which also mistakes the orange container for a car and further detects the speed limit on the road. Our approach, on the top right, correctly matches the ground truth.

5.4. Comparison with the State of the Art

We compare our approach with the following baselines[3]. **FR**: FasterRCNN trained only on the source data with random crop augmentation; **AT**: AdaptTeacher [20]; **MT**: Mean Teacher initialized with FR and trained with random color jittering on both the source and target data (i.e., this corresponds to our mean teacher setup in Sec. 4.2 but without the aggregator and without transformations $\mathcal{T}$); **FR+PIT**: Same setup as FR but with the images corrected with PIT [13]; **MT+PIT**: Same setup as MT but with the

[2] In Cityscapes, a person may be labeled as either *person* or *rider*. Since the *rider* label is used for people riding a vehicle, we omit these cases.

[3] We re-implement PIT baselines with our detector.

Method	Car AP@0.5
FR [24]	76.1
AT [20]	77.2
FR+PIT	77.6
MT	78.3
MT+PIT [13]	79.7
Ours	**80.4** $\pm$ 0.15

Table 1. FoV Adaptation.

	Car AP@0.5 for FoVx			
Method	50°	70°	80°	90°
FR [24]	94.3	90.2	86.8	80.6
FR+PIT [13]	93.6	91.4	89.2	85.9
Ours-*h*	94.1 $\pm$ 0.16	93.1 $\pm$ 0.33	91.8 $\pm$ 0.40	88.8 $\pm$ 0.21

Table 2. FoV Generalization.

Method	Pedestrian AP@0.5
FR [24]	43.7
AT [20]	63.5
MT	64.7
Ours	**65.3** $\pm$ 0.37

Table 3. Viewpoint Adaptation.

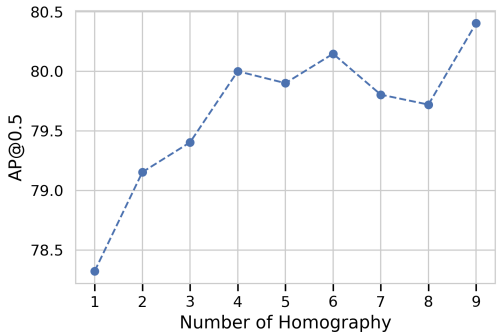

Figure 6. **Varying the number of homographies.** We evaluate the effect of N on the FoV adaptation task.

images corrected with PIT. We refer to our complete approach (Sec. 4.2) as **Ours**. For the task of FoV generalization, we report our results as **Ours-*h*** to indicate that we only optimize the homographies (5×4 parameters) in $\mathcal{T}$ to adapt to the new FoVs while keeping the base and aggregator networks frozen. This matches the setup of PIT [13], which also corrects the images according to the new FoVs. As **Ours** and **Ours-*h*** are trained with randomly initialized $\mathcal{T}$, we report the average results and standard deviations over three independent runs.

FoV adaptation. The results of Cityscapes $\rightarrow$ KITTI FoV adaptation are provided in Tab. 1. Both MT+PIT and *Ours* both bridge the FoV gap, outperforming the MT baseline. Note, however, that we achieve this by learning the transformations, without requiring any camera-specific information, which is needed by PIT. Note also that MT outperforms FR by learning a better representation in the target domain, even though FR is trained with strong augmentation, such as random cropping. AT underperforms because its strong augmentation strategy fails to generalize for datasets having prominent geometric shifts. Our improvement over MT evidences that learning transformations helps to overcome geometric shifts. We optimize with $N = 9$, homographies in this setup. Fig. 5 shows a qualitative example. Different homographies look into different image regions and the aggregator learns how to combine the activations corresponding to objects as depicted in Fig. 7. In supp. material Sec. A.3, we show effectiveness of a learned aggregator over the others. Additionally, we provide results on FoV decreasing adaptation in supp. material Sec. A.5.

FoV generalization. Tab. 2 summarizes the results obtained by using different FoVs as target domains while fixing the source FoV to 50°. Since both the source and target images are taken from KITTI, the domain gap is only caused by a FoV change. Note that the performance of FR drops quickly as the FoV gap increases. *Ours-h* outperforms FR+PIT by a growing margin as the FoV gap increases. This shows that learning transformations helps to generalize better to different amounts of geometric shifts.

Viewpoint adaptation. As shown in Fig. 1, a change in the camera viewpoint yields differences in the observed distortions and type of occlusions. The results in Tab. 3 show the benefits of our method over MT in this case. Note that PIT, which was designed for FoV changes, cannot be applied to correct for a viewpoint change. Other baselines outperform FR, as they use pseudo labels to fix the difference in bounding box distribution, as shown in Fig. 1. These results illustrate the generality of our method to different kinds of geometric shifts. Qualitative results for this task can be found in Fig. A.10.

5.5. Additional Analyses

Variable number of homographies. Let us now study the influence of the number of homographies in $\mathcal{T}$. To this end, we vary this number between 1 and 9. In Fig. 6, we plot the resulting APs for the Cityscapes-to-KITTI FoV adaptation task. Increasing the number of transformations results in a steady increase in performance, which nonetheless tends to plateau starting at 4 homographies. This is similar to our theoretical observation in Fig. A.3 of supp. material. Due to limited compute resources, we couldn't run ex-

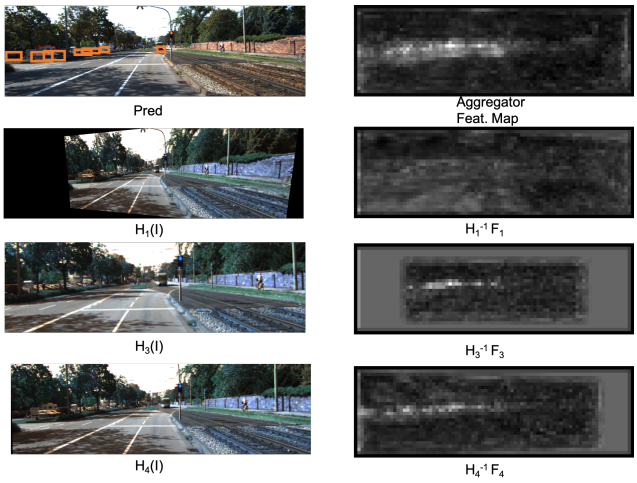

Figure 7. **Feature Maps**: Top row: predictions of our network and feature map after aggregator. Left column: Image I, transformed by learned homographies; Right Column: Feature maps F warped by corresponding H^{-1} which are input to the aggregator. Each transform distorts the image regions differently. 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 the zoom-in effect has the strongest activation as the smaller objects are visible better here. These maps are generated by taking maximum over channel dimension.

periments with more than 9 homographies. This confirms the intuition that a higher number of perspective transformations can better capture the geometric shift between two domains. Therefore, we conducted all experiments with the maximum number of homographies allowed by our compute resources.

Only optimizing $\mathcal{T}$. We also run the *Ours-h* baseline in the FoV and viewpoint adaptation scenarios. The resulting APs are 78.2 and 49.8, respectively. By learning only the 20 (5×4) homography parameters, our approach outperforms FR (in Tab. 1 and Tab. 3, respectively) by a large margin in both cases. This confirms that our training strategy is able to efficiently optimize $\mathcal{T}$ to bridge the geometric gap between different domains. We visualize in Fig. A.9 in the supplementary material some transformations learned for FoV adaptation by *Ours-h*. Note that they converge to diverse homographies that mimic a different FoV, correctly reflecting the adaptation task. Additionally in supp. material Sec. A.4, we ablate why the learning transform is useful. Specifically, learning not only provide better performance but provide faster inference w.r.t random transformations.

Diversity in $\mathcal{T}$. To show that our approach can learn a diverse set of transformations that help in the adaptation task, we initialize all the homographies with identity. Fig. 8 depicts the diversity of the learned homographies on the FoV adaptation task. Even though we do not enforce diversity, our approach learns a diverse set of transformations. With these learned homorgraphies, our model achieves 79.5 AP@0.5 score for the FoV adaptation task. We show additional results in the supplementary material Sec. A.6 and Sec. A.7.

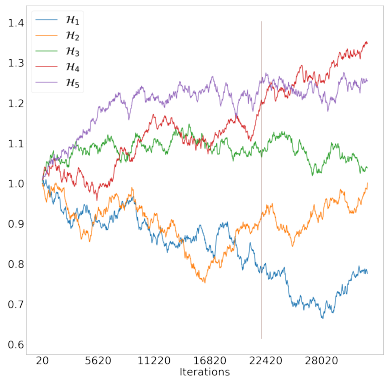

Figure 8. **Diversity in $\mathcal{T}$:** We train 5 homographies initialized as $\mathcal{H}_i = I$. We plot the evolution of s_x for different homograhies as training proceeds. Each homography is shown in a different color. Note that the values for the different homographies become diverse. The best score is achieved at iteration = 22k, indicated with the vertical line.

Limitations. Our approach assumes that the geometric gap between two domains can be bridged by a set of perspective transformations. We have shown that with enough transformations this is true. However, using a large number of homographies comes at a computational cost. The computational overhead leads to an increment in the inference time from $0.062s$ to 0.096s for $N = 5$ on an A100 Nvidia GPU with image dimension 402×1333. Nevertheless, our simple implementation shows promising results, and we will work on reducing this overhead in future work. Moreover since the optimization of the homography set is done at the dataset level, only certain transformations are beneficial to a given image. In the future, we therefore intend to condition the homography on the input image, which would reduce the total number of homographies needed.

6. Conclusion

We have introduced an approach to bridge the gap between two domains caused by geometric shifts by learning a set of homographies. We have shown the effectiveness our method on two different kinds of shifts, without relying on any annotations in the target domain, including information about the nature of the geometric shifts. Our analyses have evidenced that optimizing the transformations alone brings in improvement over the base detector and increasing the number of learnt homographies helps further. In the future, we plan to learn transformations that are conditioned on the input image to model image-dependent geometric shifts.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. Detr, https://github.com/facebookresearch/detr, 2020. 1

[3] Chaoqi Chen, Zebiao Zheng, Xinghao Ding, Yue Huang, and Qi Dou. Harmonizing transferability and discriminability for adapting object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8869–8878, 2020. 1, 2

[4] Chaoqi Chen, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. I3net: Implicit instance-invariant network for adapting one-stage object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[5] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2

[6] 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. 1, 2, 5

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2

[8] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 1, 5

[9] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4091–4101, 2021. 1, 2

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 2

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1, 2, 5

[13] Qiqi Gu, Qianyu Zhou, Minghao Xu, Zhengyang Feng, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Pit: Position-invariant transform for cross-fov domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8761–8770, 2021. 1, 2, 3, 5, 6, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[15] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018. 1

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 2, 4

[17] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, Oct. 2020. 1

[18] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6092–6101, 2019. 5

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 6, 7

[21] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2

[22] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 2

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[24] 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. 1, 4, 6, 7

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Chal-

lenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[26] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 1, 2

[27] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 1

[28] Zhiqiang Shen, Harsh Maheshwari, Weichen Yao, and Marios Savvides. Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. *arXiv preprint arXiv:1911.02559*, 2019. 2

[29] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2

[30] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 2

[31] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 2, 4, 5, 6

[32] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 2

[33] Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A Sindagi, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4516–4526, 2021. 2

[34] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[35] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, pages 5423–5432. PMLR, 2018. 2

[36] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020. 2

[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

[38] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019. 1, 2