

LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation

Inkyu Shin Dong-Jin Kim Jae Won Cho Sanghyun Woo Kwanyong Park In So Kweon
KAIST, South Korea.

{dlsrbgg33, djnjusa, chojw, shwoo93, pkyong7, iskweon77}@kaist.ac.kr

Abstract

*Unsupervised Domain Adaptation (UDA) for semantic segmentation has been actively studied to mitigate the domain gap between label-rich source data and unlabeled target data. Despite these efforts, UDA still has a long way to go to reach the fully supervised performance. To this end, we propose a **Labeling Only if Required** strategy, **LabOR**, where we introduce a human-in-the-loop approach to adaptively give scarce labels to points that a UDA model is uncertain about. In order to find the uncertain points, we generate an inconsistency mask using the proposed adaptive pixel selector and we label these segment-based regions to achieve near supervised performance with only a small fraction (about 2.2%) ground truth points, which we call “Segment based Pixel-Labeling (SPL).” To further reduce the efforts of the human annotator, we also propose “Point based Pixel-Labeling (PPL),” which finds the most representative points for labeling within the generated inconsistency mask. This reduces efforts from 2.2% segment label \rightarrow 40 points label while minimizing performance degradation. Through extensive experimentation, we show the advantages of this new framework for domain adaptive semantic segmentation while minimizing human labor costs.*

1. Introduction

Semantic segmentation enables understanding of image scenes at the pixel level, and is critical for various real-world applications such as autonomous driving [41] or simulated learning for robots [13]. Unfortunately, the pixel level understanding task in deep learning requires tremendous labeling efforts in both time and cost. Therefore, unsupervised domain adaptation (UDA) [15] addresses this problem by utilizing and transferring the knowledge of label-rich data (source data) to unlabeled data (target data), which can reduce the labeling cost dramatically [40]. According to the adaptation methodology, UDA can be largely divided into *Adversarial learning based* [27, 37, 51, 52, 54] DA and *Self-training based* [31, 34, 47, 57, 59] DA. While the former focuses on minimizing task-specific loss for

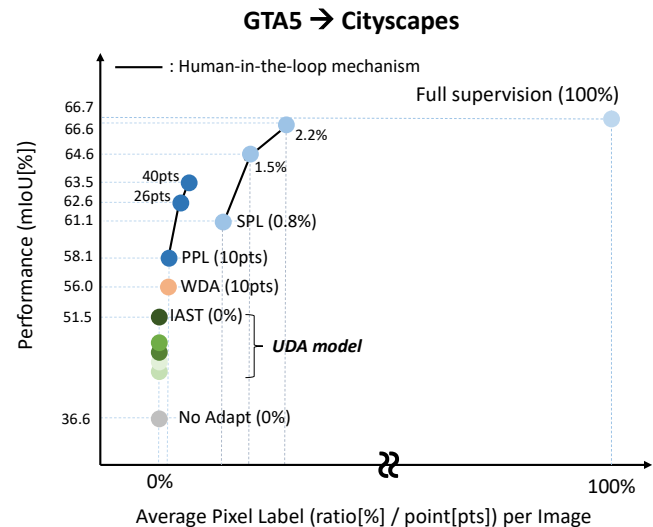


Figure 1. **Average Pixel Label per image vs. Performance.** Our novel human-in-the-loop framework, **LabOR** (PPL and SPL) significantly outperforms not only previous UDA state-of-the-art models (e.g., IAST [31]) but also DA model with few labels (e.g., WDA [38]). Note that our PPL requires negligible number of label to achieve such performance improvements (25 labeled points per image), and our SPL shows the performance comparable with fully supervised learning (0.1% mIoU gap). Detailed performance can be found in Table. 1 and Fig. 5.

source domain and domain adversarial loss, the self-training strategy retrains the model with generated target-specific pseudo labels. Among them, IAST [31] achieves state-of-the-art performance in UDA by effectively mixing adversarial based and self-training based strategies.

Despite the relentless efforts in developing UDA models, the performance limitations are clear as it still lags far behind the fully supervision model. As visualized in Fig. 1, the recent UDA methods remain at around (\sim 50% mIoU) which is far below the performance of full supervision (\sim 65% mIoU) on GTA5 [41] \rightarrow Cityscapes [8].

Motivated by the limitation of UDA, we present a new perspective of domain adaptation by utilizing a minute portion of pixel-level labels in an adaptive human-in-the-loop manner. We name this framework **Labeling Only** if

Required (LabOR), which is described in Fig. 2. Unlike conventional self-training based UDA that re-trains the target network with the pseudo labels generated from the model predictions, we utilize the model predictions to find uncertain regions that require human annotations and train these regions with ground truth labels in a supervised manner. In particular, we find regions where the two different classifiers mismatch in predictions. In order to effectively find the mismatched regions, we introduce additional optimization step to maximize the discrepancy between the two classifiers like [7, 43]. Therefore, by comparing the respective predictions from the two classifiers on a pixel level, we create a mismatched area that we call the *inconsistency mask* which can be regarded uncertain pixels. We call this framework the “Adaptive Pixel Selector” which guides a human annotator to label on proposed pixels. This results in the use of a very small number of pixel-level labels to maximize performance. Depending on how we label the proposed areas, we propose two different labeling strategies, namely “Segment based Pixel-Labeling (SPL)” and “Point based Pixel-Labeling (PPL).” While SPL labels every pixels on the inconsistency mask in a segment-like manner, PPL places its focus more on the labeling effort efficiency by finding the representative *points* within a proposed segment. We empirically show that the two proposed “Pixel-Labeling” options not only help a model achieve near supervised performance but also reduces human labeling costs dramatically.

We summarize our contributions as follows:

1. We design a new framework of domain adaptation for semantic segmentation, LabOR, by utilizing a small fraction of pixel-level labels with an adaptive human-in-the-loop pixel selector.
2. We propose two labeling options, Segment based Pixel-Labeling (SPL) and Point based Pixel-Labeling (PPL), and show that these methods are especially advantageous in performance compared to UDA and labeling efficiency respectively.
3. We conduct extensive experiments to show that our model outperforms previous UDA model by a significant margin even with very few pixel-level labels.

2. Related Work

Unsupervised Domain Adaptation. Domain Adaptation is a classic computer vision problem that aims to mitigate the performance degradation due to a distribution mismatch across domains and has been investigated in image classification problems through both conventional methods [10, 14, 15, 24, 26] and deep CNN-based methods [11, 12, 25, 29, 32, 35, 44]. Domain adaptation has recently been studied in other vision tasks such as ob-

ject detection [5], depth estimation [1], and semantic segmentation [17]. With the introduction of the automatically annotated GTA dataset [40], *unsupervised* domain adaptation (UDA) for semantic segmentation has been extensively studied. Adversarial learning approaches have aimed to minimize discrepancy between source and target feature distributions and this approach has been studied on three different levels in practice: input-level alignment [6, 17, 33, 36, 48], intermediate feature-level alignment [18, 19, 28, 30, 52], and output-level alignment [51].

Domain Adaptation with Few Labels. Despite extensive studies in UDA, the performance of UDA is known to be much lower than that of supervised learning [42]. In order to mitigate this limitation, various works have tried to leverage ground truth labels for the target dataset. For example, semi-supervised domain adaptation, which utilize randomly selected image-level labels per class as the labeled training target examples, has been recently studied for image classification [42], semantic segmentation [53], and image captioning [4, 20]. However, these naive semi-supervised learning approaches do not consider which target images should be labeled given a fixed budget size. Similar to semi-supervised domain adaptation, some works have used active learning [45] to give labels to a small portion of the dataset [50, 39]. These works leverage a model to find data points that would increase the performance of the model the most. Furthermore, in order to reduce the labeling effort per image for target images in domain adaptation, a method to leverage weak labels, several points per image, has also been studied [38].

In contrast, our work differentiates itself by allowing the model to automatically pinpoint to the human annotator which points to label on a pixel-level that would have the best potential performance increase instead of randomly picking labels which can possibly be already easy for the model to predict. In addition, unlike the semi-supervised model which has random annotations prior to training, we allow the model to let the annotator know which points in an image are best to increase performance. Although at first glance our method may seem similar to active learning in the human-in-the-loop aspect, our work is the first to propose a method on the *pixel-level* instead of image-level. Overall, our pixel-level sampling approach is not only efficient, but also orthogonal to the existing active, weak label, or semi-supervised domain adaptation frameworks.

3. Proposed Method

In this section, we introduce our method from inconsistency mask generation to adaptive pixel labeling.

3.1. Problem Definition: Domain Adaptation

Let us denote $g_\phi(\cdot)$ as the network backbone with the parameter ϕ that generates features from an input x . Then,

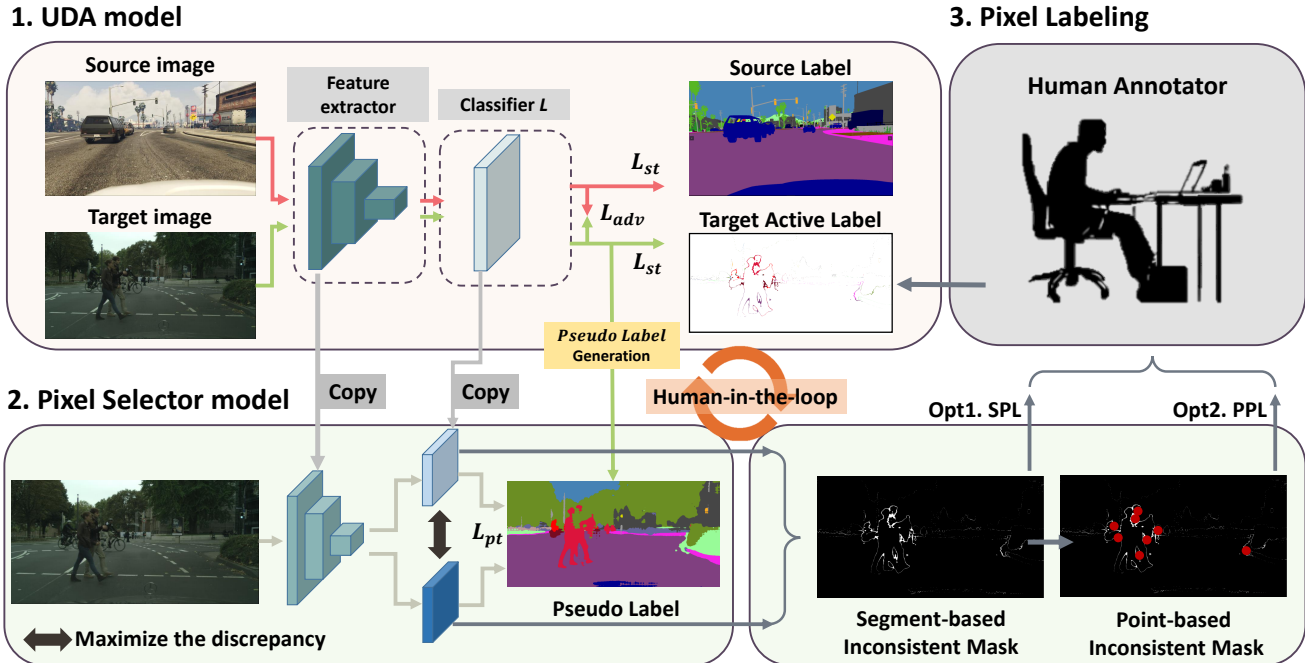


Figure 2. **The overview of the proposed adaptive pixel-basis labeling, LabOR.** This framework is made up of two models: UDA model and Pixel selector model. The UDA model initially trained from conventional adversarial learning forwards target image to generate pseudo label. Different from normal self-training training scheme [31] that utilizes the generated label to retrain the model directly, we instead train a pixel selector model to brings out inconsistent mask where human annotator is guided to label. In this process, we use pseudo label training loss, L_{pt} which contains pseudo label cross entropy loss and classifiers’ discrepancy loss. With those human labels, we return to the original UDA model for training that uses L_{st} .

with the classification layer including softmax activation $f_{\theta}(\cdot)$ with the parameter θ , a class prediction (probability) is computed ($\hat{\mathbf{Y}} = p(\mathbf{Y}|\mathbf{x}; \theta, \phi) = f_{\theta} \circ g_{\phi}(\mathbf{x}) \in \mathbb{R}^{W \times H \times K}$, where W and H are width and height of the segmentation map, and K is the total number of classes). The combined network $f_{\theta} \circ g_{\phi}(\cdot)$ can be implemented with typical semantic segmentation generators [2, 3]. A typical semantic segmentation model is trained with cross-entropy loss $\text{CE}(\cdot, \cdot)$ with the ground truth label $\mathbf{Y} \in \mathbb{R}^{W \times H}$. Furthermore, let us denote $\mathcal{S} = \{(\mathbf{x}_s, \mathbf{Y}_s)\}_{s=1}^S$ as the labeled images from the source dataset and $\mathcal{T} = \{\mathbf{x}_t\}_{t=1}^T$ as the unlabeled images from the target dataset. Unsupervised Domain Adaptation (UDA) tries to leverage both the abundant labeled source dataset and the small number of unlabeled target dataset to train a deep neural network.

Recent unsupervised domain adaptive semantic segmentation use self-training methods [31, 60] and have shown state-of-the-art performances and are optimized as follows: In practice, the model alternates between generating pseudo-labels $\hat{\mathbf{Y}}_t(\mathbf{x}_t) \in \mathbb{R}^{W \times H}$ for an image \mathbf{x}_t based on the model prediction $p(\mathbf{Y}|\mathbf{x}; \theta, \phi)$ and retraining the model on the target dataset with the generated pseudo labels. The goal of self-training based domain adaptation [31, 60] is to devise an effective loss function and a way to generate pseudo labels. Specifically, CRST [60] propose class-

balanced pseudo label generation strategy and confident region KLD minimization to prevent overfitting on pseudo labels. IAST [31] tackles the class-balanced pseudo label generation which ignores the individual attributes of instance to design an instance adaptive selector. Moreover, IAST adds an entropy minimization approach on unlabeled pixels. Self-training based domain adaptation far underperforms a fully supervised model. This can be attributed to two reasons. First, cutting out unconfident pixels and retraining with the thresholded labels is not intuitive as the model forced to be trained with only the pixels that model itself is confident in. Second, existing pseudo label generation commonly originates from specific manually set hyper-parameters, causing incorrect pseudo labels which degrades the performance. To address this issue, we propose a new perspective of self-training based domain adaptation with a human-in-the-loop approach by using a human annotator to label a small number of informative *pixels*. As the human annotator annotates the pixels where the model is uncertain, the labeled pixels ultimately act as a guide for the model. We call this method **Labeling Only if Required (LabOR)**. In order to minimize the efforts of the human annotator, we must answer the key question “*what is an informative pixel to label?*” In other words, our goal is to find the pixels where the model is uncertain. To this end, we propose to

select the pixels that show the highest *classifier discrepancy* motivated by the classifier discrepancy based domain adaptation method, MCDDA [43].

3.2. Generating Inconsistency Mask

Fig. 2 illustrates an overview of our proposed method. First, we pre-train a model with the labeled source dataset \mathcal{S} by minimizing supervised cross-entropy loss:

$$\mathcal{L}_s(\theta, \phi) = \mathbb{E}_{(\mathbf{x}_s, \mathbf{Y}_s) \in \mathcal{S}} [\text{CE}(\mathbf{Y}_s, p(\mathbf{Y}|\mathbf{x}_s; \theta, \phi))]. \quad (1)$$

Following this, in order to improve the effectiveness of self-training, we utilize warm-up with adversarial training [31] before moving on to self-training.

$$\mathcal{L}_{adv}(\theta, \phi) = \mathbb{E}_{\mathbf{x}_s \in \mathcal{S}, \mathbf{x}_t \in \mathcal{T}} [\text{Adv}(p(\mathbf{x}_s; \theta, \phi), p(\mathbf{x}_t; \theta, \phi))]. \quad (2)$$

Then we copy the parameters of the backbone and the classifier (twice for classifier) (i.e., $\theta'_1 \leftarrow \theta, \theta'_2 \leftarrow \theta, \phi' \leftarrow \phi$) to create our Adaptive Pixel Selector model ($f_{\theta'_1}, f_{\theta'_2}, g_{\phi'}$). This model is only used for the purposes of pixel selection and has no effect on the performance. Using this newly created model, we optimize the model with the two auxiliary classifiers and increase the discrepancy in relation to each other. After this, we propose to find the pixels where the two classifiers have different output class predictions. Using the different output class predictions, we create a mask consisting of pixels that are inconsistent $M(\mathbf{x}_t; \phi', \theta'_1, \theta'_2) \in \mathbb{R}^{W \times H}$, and we call this the *inconsistency mask*. The mask generation would be formulated as follows:

$$M(\mathbf{x}_t) = \left[\arg \max_K f_{\theta'_1} \circ g_{\phi'}(\mathbf{x}_t) \neq \arg \max_K f_{\theta'_2} \circ g_{\phi'}(\mathbf{x}_t) \right]. \quad (3)$$

For simplicity, we abuse the notation $M(\mathbf{x}_t; \phi', \theta'_1, \theta'_2)$ as $M(\mathbf{x}_t)$. We conjecture that if the two classifiers trained on the same dataset generate different predictions for the same region, then it means the model prediction shows a high variance in that input region. Therefore we conclude that this *inconsistency mask* represents the pixels the model is the most unsure about. In other words, we hypothesize that by giving ground truth labels for these pixels to guide the model, the model would more easily bridge the gap between the domains and improve the generalizability of the model. The detailed method on giving ground truth labels will be described in the next subsection.

Given $\phi', \theta'_1, \theta'_2$, we first apply the self-training loss function with the pseudo labels (one-hot vector labels generated from $\hat{\mathbf{Y}}_t = p(\mathbf{Y}|\mathbf{x}_t; \theta, \phi)$), which has been utilized in various tasks [21, 22, 31, 49, 60]:

$$\begin{aligned} \mathcal{L}_{\text{self}}(\phi', \theta'_1, \theta'_2) &= \mathbb{E}_{\mathbf{x}_t \in \mathcal{T}} [\text{CE}(\arg \max_K \hat{\mathbf{Y}}_t, p(\mathbf{Y}|\mathbf{x}_t; \theta'_1, \phi')) \\ &\quad + \text{CE}(\arg \max_K \hat{\mathbf{Y}}_t, p(\mathbf{Y}|\mathbf{x}_t; \theta'_2, \phi'))]. \end{aligned} \quad (4)$$

The detailed design choices for the pseudo labels for $\mathcal{L}_{\text{self}}$ is discussed in the supplementary materials. Then, in order to optimize the two auxiliary classifiers to increase the discrepancy in relation to each other, we introduce an additional training stage to optimize the auxiliary classifiers to increase the distance between the classifiers' outputs. In addition, we also minimize the classifier discrepancy with respect to the backbone feature extractor $g_{\phi'}$, which results in a similar formulation to the classifier discrepancy maximization in MCDDA [43]:

$$\begin{aligned} &\min_{\phi'} \max_{\theta'_1, \theta'_2} \mathcal{L}_{\text{dis}}(\phi', \theta'_1, \theta'_2) \\ &= \min_{\phi'} \max_{\theta'_1, \theta'_2} \mathbb{E}_{\mathbf{x}_t \in \mathcal{T}} \left[\|f_{\theta'_1} \circ g_{\phi'}(\mathbf{x}_t) - f_{\theta'_2} \circ g_{\phi'}(\mathbf{x}_t)\|_1 \right]. \end{aligned} \quad (5)$$

Note that the goal of classifier discrepancy maximization in MCDDA is to create tighter decision boundaries in order to align the latent feature distributions between the source and the target domains. In contrast, we maximize the classifier discrepancy for the sole purposes of generating a more representative inconsistency mask so that the human annotator can give ground truth labels to pixels that truly require labels. After optimizing the auxiliary classifiers ($\theta'_1, \theta'_2, \phi'$), we utilize the different outputs from these classifiers and compare them in a pixel-to-pixel manner using (3) to obtain $M(\mathbf{x}_t)$. After the human annotator gives ground truth labels to the uncertain pixels based on $M(\mathbf{x}_t)$, the model (f_{θ}, g_{ϕ}) is then trained with the target dataset \mathcal{T} with the given ground truth labeled pixels $\tilde{\mathbf{Y}}_t(\mathbf{x}_t)$:

$$\mathcal{L}_t(\theta, \phi) = \mathbb{E}_{\mathbf{x}_t \in \mathcal{T}} [\text{CE}(\tilde{\mathbf{Y}}_t(\mathbf{x}_t), p(\mathbf{Y}|\mathbf{x}_t; \theta, \phi))]. \quad (6)$$

Then the process starting from copying ($\theta'_1 \leftarrow \theta, \theta'_2 \leftarrow \theta, \phi' \leftarrow \phi$), optimizing $\mathcal{L}_{\text{self}}(\phi', \theta'_1, \theta'_2)$ and $\mathcal{L}_{\text{dis}}(\phi', \theta'_1, \theta'_2)$, to inconsistency generation $M(\mathbf{x}_t)$ is repeated. The overall method is summarized in Alg. (1). We repeat the process 3 times as we empirically found that the number of uncertain pixels and the model performance converges after 3 stages.

3.3. Adaptive Pixel Labeling

Given an inconsistency mask $M(\mathbf{x}_t)$, the question arises as how to give labels to the pixels. With this in mind, we propose two different methods for giving ground truth annotations with different focuses and strengths.

Segment based Pixel-Labeling (SPL). As the inconsistency mask shows all pixels that the model is uncertain about, we consider giving ground truth annotations for all the pixels selected. We call this methods the Segment based Pixel-Labeling (SPL). In SPL, no further calculations are needed after the inconsistency mask has been generated, and after the pixels are annotated, the model $p(\mathbf{Y}|\mathbf{x}; \theta, \phi)$ is further trained. Empirically, we find that the inconsistency mask for each stage averages in percent of pixel of

Algorithm 1: Pixel Selector Model

Input: Source data \mathcal{S} , Target data \mathcal{T} , Initialized model $f_\theta \circ g_\phi(\cdot)$
Output: The model with adapted weights on target dataset $f_\theta \circ g_\phi(\cdot)$

```
1 begin
2   Pre-train the model on the source dataset.
3   And, initial adapt with adversarial learning.
4    $\theta, \phi \leftarrow \arg \min_{\theta, \phi} \mathcal{L}_s(\theta, \phi) + \mathcal{L}_{adv}(\theta, \phi)$ 
   (Eq. (1))
5   for 3 Stages do
6     Define auxiliary layers and copy weights
7      $\theta'_1 \leftarrow \theta, \theta'_2 \leftarrow \theta, \phi' \leftarrow \phi$ 
8     Apply self-training (Eq. (4))
9      $\phi', \theta'_1, \theta'_2 \leftarrow \arg \min_{\phi', \theta'_1, \theta'_2} \mathcal{L}_{self}(\phi', \theta'_1, \theta'_2)$ 
10    Maximize classifier discrepancy (Eq. (5))
11     $\phi', \theta'_1, \theta'_2 \leftarrow \arg \min_{\phi'} \max_{\theta'_1, \theta'_2} \mathcal{L}_{dis}(\phi', \theta'_1, \theta'_2)$ 
12    for  $\mathbf{x}_t \in \mathcal{T}$  do
13      Generate  $M(\mathbf{x}_t; \phi', \theta'_1, \theta'_2)$  with Eq. (3).
14      if SPL then
15        Annotate inconsistency mask
16         $\tilde{\mathbf{Y}}_t(\mathbf{x}_t) \leftarrow M(\mathbf{x}_t) \odot \mathbf{Y}_t$ 
17      else if PPL then
18        Select representative points
19         $P(\mathbf{x}_t) = \text{SelectPt}(M, p(\mathbf{Y}|\mathbf{x}_t; \theta, \phi))$ 
20        Annotate the points
21         $\tilde{\mathbf{Y}}_t(\mathbf{x}_t) \leftarrow P(\mathbf{x}_t) \odot \mathbf{Y}_t$ 
22      Train the model with the pseudo labels
23       $\theta, \phi \leftarrow \arg \min_{\theta, \phi} \mathcal{L}_t(\theta, \phi)$  (Eq. (6))
```

total pixels per image at 1% and totals to 2.2% at the final stage as some uncertain pixels are overlapped. The performance of SPL achieves near supervised learning, and it far exceeds the performance of our next method, which is more focused on drastically reducing human annotation labor.

Point based Pixel-Labeling (PPL). We also propose another Pixel-Labeling method that sets its focus on minimizing human annotation costs; we call this method the Point based Pixel-Labeling (PPL). Although PPL receives an inconsistency mask like SPL, we propose to label only the most *representative* pixels in the inconsistency mask instead of labeling all the pixels. Among the most representative pixels, we deliberately choose to maximize diversity by selecting all unique classes present in the inconsistency mask.

Given a set of uncertain pixels (inconsistency mask $M(\mathbf{x})$) and a model’s output probability prediction for all the pixels $\hat{\mathbf{Y}} = \{\hat{\mathbf{Y}}_{i,j} \in \mathbb{R}^K | i \in [1, W], j \in [1, H]\}$, we

first cluster the pixels that the model $p(\mathbf{Y}|\mathbf{x}; \theta, \phi)$ predicts to be the same class. We define the set of uncertain pixels \mathcal{D}^k for class k as follows:

$$\mathcal{D}^k = \{(i, j) \in M(\mathbf{x}) | k = \arg \max_K \hat{\mathbf{Y}}_{i,j}\}. \quad (7)$$

Then we compute the class prototype vector μ^k for each class k as the mean vectors of \mathcal{D}^k :

$$\mu^k = \frac{1}{|\mathcal{D}^k|} \sum_{(i,j) \in \mathcal{D}^k} \hat{\mathbf{Y}}_{i,j} \in \mathbb{R}^K. \quad (8)$$

Finally, we select the points that has the most similar probability vector for each prototype vector to construct the set of selected points P :

$$P(\mathbf{x}) = \{\arg \min_{(i,j) \in \mathcal{D}^k} d(\mu^k, \hat{\mathbf{Y}}_{i,j})\}_{k=1}^K. \quad (9)$$

We use cosine distance for a distance measure $d(\cdot, \cdot)$. Note that as \mathcal{D}^k can be a null set for some classes, $0 \leq |P(\mathbf{x}_t)| \leq K$, if the model fails to predict a certain class. At each stage, on average, the model generates 12 clusters, and cumulatively we average on giving 40 ground truth labels per target image \mathbf{x}_t in an image of size 640×1280 . This calculates to a $\approx 0.0049\%$ of the image being given ground truth labels. In comparison to SPL, which averages ≈ 18022 pixels $\rightarrow 2.2\%$ of entire image, we further reduce the human labeling costs by 0.2%. Due to the drastically reduced amount of ground truth annotations, PPL naturally underperforms in relation to SPL. Nevertheless, we empirically show that the performance gain of PPL over other UDA or weakly supervised DA methods is still significant.

4. Experiments

In this section, we conduct extensive experiments to analyze our methods both quantitatively and qualitatively.

4.1. Dataset

We evaluate our model on the most common adaptation benchmark of GTA5 [41] to Cityscapes [8]. Following the standard protocols from previous works [31, 30], we adapt the model to the Cityscapes training set and evaluate the performance on the validation set.

4.2. Implementation details

To push the state-of-the-art benchmark performances, we test our method Labor on the IAST framework [31]. For our backbones, we use ResNet-101 [16] for the feature extractor and Deeplab-v2 [2] for the segmentation model. We utilize source domain to pretrain model and adversarial training to initially reduce domain shift. We train the model for a total of 3 stages. In each stage, the proposed iterative human-in-the-loop mechanism is performed. We follow IAST’s implementation details for fair comparison.

GTA5 \rightarrow Cityscapes																				
Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
No Adapt	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
AdaptSegNet [51]	86.5	36.0	79.9	23.4	23.3	35.2	14.8	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
ADVENT [52]	89.9	36.5	81.2	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
SIMDA [54]	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
LTIR [23]	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2
PCEDA [55]	91.0	49.1	85.6	37.2	29.7	33.7	38.1	39.2	85.4	35.4	85.1	61.1	32.8	84.1	45.6	46.9	0.0	34.2	44.5	50.5
FDA [56]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
CBST [58]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
CRST(MRKL) [60]	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1
TPLD [47]	94.2	60.5	82.8	36.6	16.6	39.3	29.0	25.5	85.6	44.9	84.4	60.6	27.4	84.1	37.0	47.0	31.2	36.1	50.3	51.2
IAST [31]	93.8	57.8	85.1	39.5	26.7	26.2	43.1	34.7	84.9	32.9	88.0	62.6	29.0	87.3	39.2	49.6	23.2	34.7	39.6	51.5
WDA [38] (Point)	94.0	62.7	86.3	36.5	32.8	38.4	44.9	51.0	86.1	43.4	87.7	66.4	36.5	87.9	44.1	58.8	23.2	35.6	55.9	56.4
Ours (PPL: Point)	96.1	71.8	88.8	47.0	46.5	42.2	53.1	60.6	89.4	55.1	91.4	70.8	44.7	90.6	56.7	47.9	39.1	47.3	62.7	63.5
Ours (SPL: Segment)	96.6	77.0	89.6	47.8	50.7	48.0	56.6	63.5	89.5	57.8	91.6	72.0	47.3	91.7	62.1	61.9	48.9	47.9	65.3	66.6
Supervised	96.9	77.1	89.8	45.6	49.9	47.4	55.8	64.1	90.0	58.2	92.8	71.9	46.9	91.4	60.3	65.8	54.3	44.6	64.7	66.7

Table 1. **Experimental results on GTA5 \rightarrow Cityscapes.** While our PPL method already surpass previous UDA state-of-the-art models (e.g., IAST [31]) and DA model with few labels(e.g., WDA [38]) by only leveraging (around 40 labeled points per image), our SPL method shows the performance comparable with fully supervised learning (only 0.1% mIoU gap).

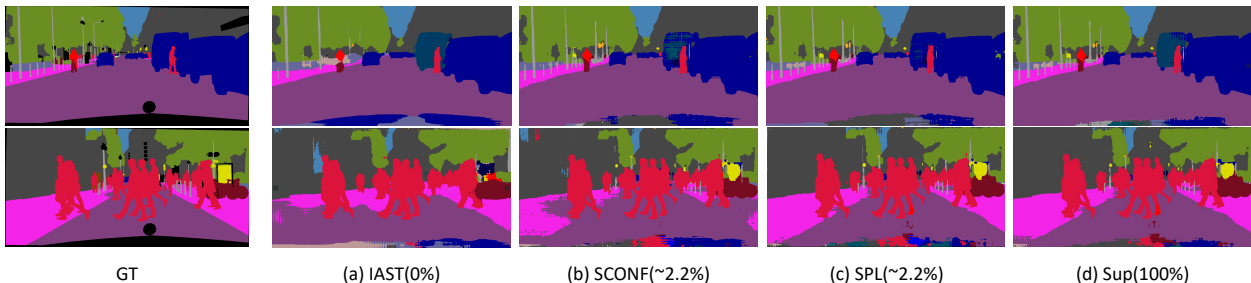


Figure 3. **Qualitative result of our SPL** While the state-of-the-art UDA method, i.e., IAST [31], and a naive way to label regions, SCONF baseline, show erroneous segmentation results, the proposed method, SPL, shows the correct segmentation result similar to the fully supervised approach.

4.3. Experimental Results on GTA5 \rightarrow Cityscapes

We show our quantitative results of both of our methods PPL and SPL compared to other state-of-the-art UDA methods [30, 51, 52, 58, 60] in Table 1. Although out of our scope, we compare our method to Weak-label DA (WDA) [38] to show the competitiveness of our approach. To truly understand the capabilities of our approach, we also include the result of the fully supervised model. Table 1 shows that our LabOR SPL outperforms all state-of-the-art UDA or WDA approaches in all cases by a large margin. Even when compared to the fully supervised method, SPL is only down by 0.1 mIoU in comparison. In some classes such as “Wall, Fence, Pole, TL, PR, Rider, Car, Truck, Motor, Bike,” SPL even outperforms the supervised model. We believe this is a remarkable finding that can potentially be explored to hopefully surpass the performance of fully supervised methods. Even though our LabOR PPL only utilized point level supervision for the target dataset, PPL also shows significant performance gains over previous state-of-the-art UDA or WDA methods. In comparison to the best

performing UDA model IAST [31], PPL gains an 12% increase in mIoU and the performance only degrades by 3.1% when compared to SPL. Even when compared to WDA that utilizes point labels similar to PPL, our PPL has a 7.1% increase in performance. Note that WDA labels average around 10-15 pixels per image, and although ours does give 3 times more pixels, we are able to increase the performance drastically while further reducing human interference.

4.4. Further Discussion

Segment based Pixel-Labeling Strategies. To understand the performance gains from SPL as an uncertainty measure, we compare SPL with several other uncertainty metrics motivated from active learning research. The comparison result between our SPL and the above baselines is demonstrated in Fig. 5 (a). **Random (RAND)** is a passive learning strategy that labels pixels according to a uniform distribution over an image region. **Softmax Confidence (SCONF)** [9] queries pixels for which a model has the least confidence in its most likely generated sequence: $1 - \max_K \hat{Y}_{i,j}$. **Entropy (ENT)** [46] queries pixels that

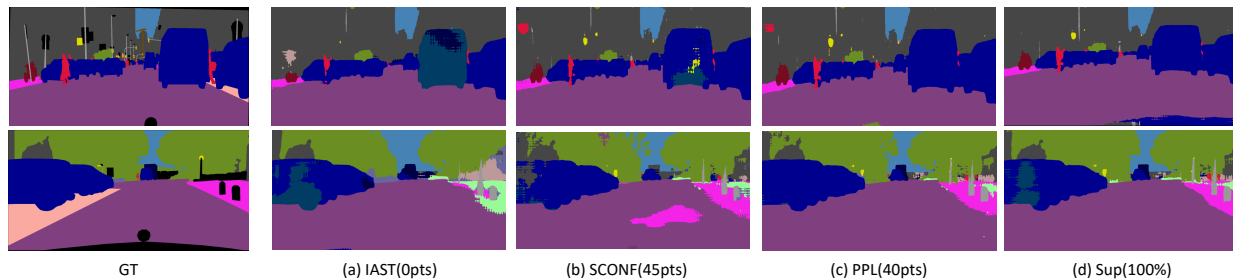


Figure 4. **Qualitative result of our PPL** While the state-of-the-art UDA method, i.e., IAST [31], and a naive way to label regions, SCONF baseline, show erroneous segmentation results, the proposed method, PPL, shows the correct segmentation result similar to the fully supervised approach.

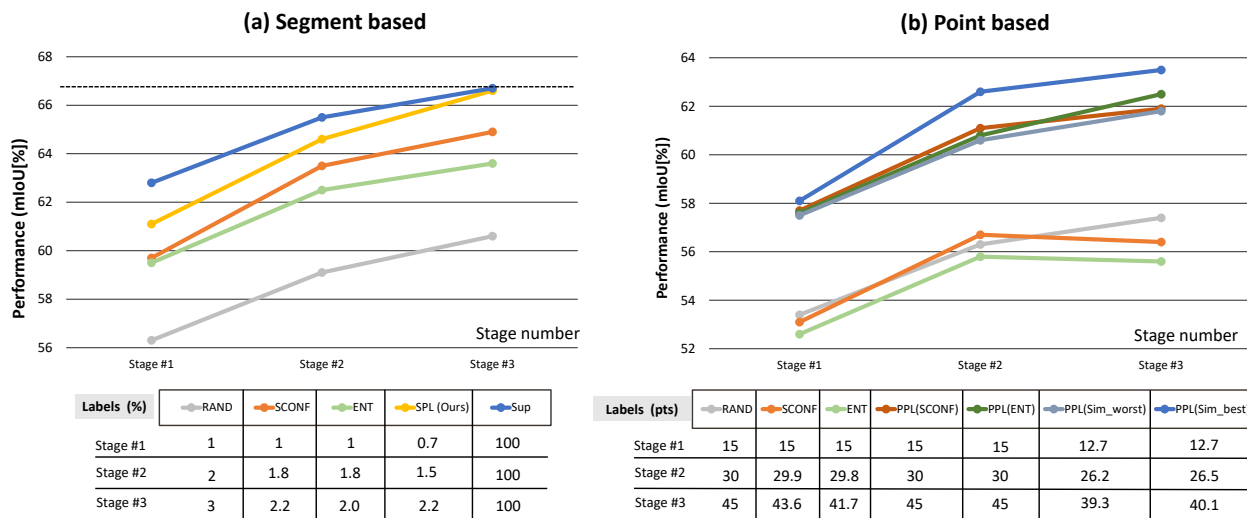


Figure 5. **The performance of (a) segment based and (b) point based pixel labeling strategies.** (a) Our method, SPL, significantly outperforms all the methods among the uncertainty metrics, and our method shows the performance comparable to that of fully supervised training method at the final stage. (b) Among the point based strategies, our final model, (PPL-Sim(best)), shows the best performance.

maximize the entropy of a model’s output: $H(\hat{Y}_{i,j})$, where $H(p) = -\sum_{k=1}^K p(k) \log p(k)$. For **RAND**, **SCONF**, and **ENT**, we need to set a constant number of pixels to label. As a result, we give labels to 1% pixel per image each stage for these baselines, so that the number of labeled pixels per stage is similar to that of our method. Note that unlike **RAND**, the pixel selection of **SCONF** and **ENT** is dependent on the model’s output during training, and this might cause the overlapping of some selected pixels over stages. As a result, although we give 1% pixel labels per each stage, the accumulated number of labeled pixels might be lower than $1\% \times (\text{Stage})$ as shown in Fig. 5 (a). **Fully Supervised (Sup)** leverages the all the ground truth labels in the target dataset for training. As shown in Fig. 5 (a), **SPL** significantly outperforms **SCONF**, which is the best performing method among the uncertainty metrics. In addition, even though **Sup** shows 1.67% mIoU gap in relation to **SPL** at Stage 1, our method shows only a 0.1% mIoU gap with **Sup** at the final stage (Stage 3). Furthermore, we tested the supervised baseline and our SPL to Stage 4, but the per-

formances of both models show the same as that of Stage 3, therefore, we make a decision to only train all methods only up to Stage 3. In summary, our **SPL** is the best option among various possible uncertain pixel selection methods in terms of the performance gain.

Point based Pixel-Labeling Strategies. For PPL, there are various options to select the *pixels* to label. We perform an additional experiment comparing our PPL with other several approaches in Fig. 5 (b). In addition to our PPL distance measures, we also evaluate other pixel selection methods, **RAND**, **SCONF**, and **ENT**, that are the exact same uncertainty metrics described in the previous paragraph, but we give labels to 15 pixels per image each stage for these baselines this time, so that the number of labeled pixels per stage is similar to that of our method. Given an inconsistency mask from Eq. (3), there are various options to select the representative points among the pixels other than measuring the distance with the class prototypes. **PPL-SCONF** queries pixels among the inconsistency mask for which a model has the least confidence in its most likely generated sequence. **PPL-ENT** queries pixels among the inconsis-

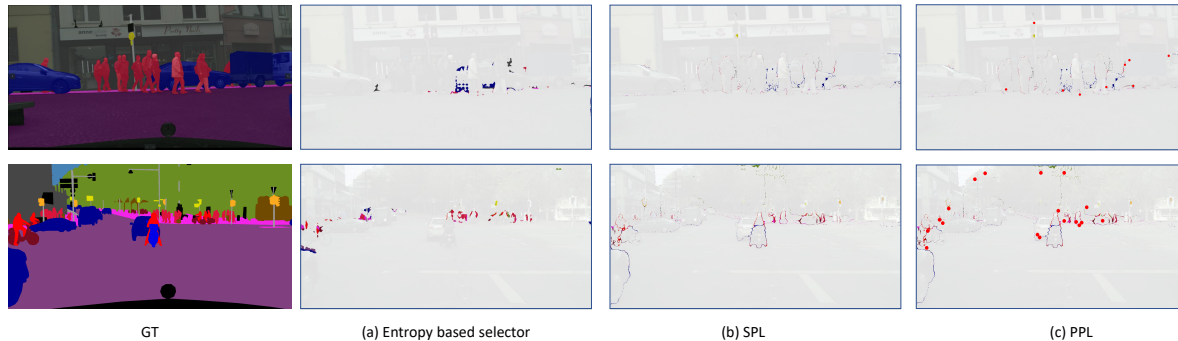


Figure 6. **The visualization of the generated regions to label** Compared to simple ENT baseline, our SPL and PPL are able to select more diverse points to give labels.

tency mask that maximize the entropy of a model’s output. Note that once again for **SCONF** and **ENT**, some uncertain pixels are overlapped, causing the number of pixels to be less than 15 each stage. After we measure the distance between the prototype vectors and the output prediction for the pixels, we can either select the point that is the nearest (**PPL-Sim(best)**) or far (**PPL-Sim(worst)**) from the prototype vectors. Fig. 5 (b) shows that our final PPL model, (**PPL-Sim(best)**), shows the best performance. Note that even the worst PPL distance measure of **PPL-Sim(worst)** far outperforms any of the other non-PPL based methods by a large margin. Interestingly, although **RAND** performs the best among the non-PPL based methods at Stage 3, **PPL-Sim(best)** even at Stage 1 outperforms the best performance of **RAND**. This result shows the importance of the strategy to pick pixels to label for a model’s performance.

Qualitative Results. Fig. 3 and Fig. 4 show the qualitative results of both our methods, SPL and PPL respectively, in comparison to the ground truth, the state-of-the-art UDA method, IAST [31], **SCONF** baseline for uncertain region selection, and **Sup** baseline as a performance upper bound. In Fig. 3, while IAST and **SCONF** baseline show erroneous segmentation results (e.g., the class “car” in the top result and the class “sidewalk” in the bottom result), the proposed method, SPL, shows the correct segmentation result similar to the supervised approach. In Fig. 4, IAST confuses the class “car” as “bus” and fails to classify the class “sidewalk.” **SCONF** baseline generates noisy segmentation result. In contrast, the proposed method, PPL, shows the correct segmentation result similar to the supervised approach.

Fig. 6 visualizes the selected uncertain pixels to label from the **ENT** baseline and our methods SPL and PPL. We can see that unlike **ENT**, SPL is able to cover a much wider range of pixels across the image. **ENT** on the other hand tends to lump pixels that are nearby together. Furthermore, PPL is also shown to pick diverse pixels and not be grounded to a certain region of the image.

Effects of Entropy Regularization on SPL and PPL. Recent work [31] has proposed a regularizer in the form of

Method	Regularizer	Stage#1	Stage#2	Stage#3
SPL	×	61.1%(0.7%)	64.6%(1.5%)	66.6%(2.2%)
	Ent [31]	61.5%(0.7%)	64.9%(1.4%)	66.4%(2.1%)
PPL	×	58.1%(12.7pts)	62.6%(26.5pts)	63.5%(40.1pts)
	Ent [31]	58.9%(12.7pts)	62.3%(26.3pts)	63.9%(39.4pts)

Table 2. **Effect of self-training entropy regularization [31] on SPL and PPL.** While the entropy regularization does not improve the performance of our SPL, adding entropy regularize on our PPL slightly improves the performance.

entropy minimization for training in UDA to regularize uncertain points in an image. In light of this, we apply the entropy minimizer on both SPL and PPL to test its effect on performance. Table. 2 shows the effects of adding the entropy minimizer. Interestingly, on SPL, the entropy minimizer does not seem to have much impact. At Stages 1 and 2, the performance does increase slightly, but at Stage 3, the performance decreases. In contrast, for PPL, the entropy regularizer slightly improves the performance. We believe this might be the case as for SPL as the uncertain pixels of SPL are given ground truth labels for, so the regularizer has minimal effects. For PPL, as the number of ground truth pixels given are few, the regularizer helps in model training.

5. Conclusion

In this work, we tackle performance discrepancy of Unsupervised Domain Adaptation and proposed a new framework for domain adaptive semantic segmentation in a human-in-the-loop manner while generating the most informative pixel points that we call **Labeling Only if Required, LabOR**. Based on a self-training platform, we build our method to select the most *informative* pixels and introduce two pixel selection methods that we call “Segment based Pixel-Labeling” and “Point based Pixel-Labeling.”

Acknowledgement This work was supported in part by Samsung Electronics Co., Ltd (G01200447), and also under the framework of international cooperation program managed by the National Research Foundation of Korea(NRF-2020M3H8A1115028, FY2021)

References

- [1] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2810, 2018.
- [2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.
- [3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [4] Tseng-Hung Chen, Yuan-Hong Liao, Ching-Yao Chuang, Wan-Ting Hsu, Jianlong Fu, and Min Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *Proc. of Int’l Conf. on Computer Vision (ICCV)*, 2017.
- [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 *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 3339–3348, 2018.
- [6] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [7] Jae Won Cho, Dong-Jin Kim, Yunjae Jung, and In So Kweon. Mcdal: Maximum classifier discrepancy for active learning. *arXiv preprint arXiv:2107.11049*, 2021.
- [8] 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.
- [9] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, 2005.
- [10] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proc. of Int’l Conf. on Computer Vision (ICCV)*, pages 2960–2967, 2013.
- [11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.
- [12] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 597–613. Springer, 2016.
- [13] Florian Golemo, Adrien Ali Taiga, Aaron Courville, and Pierre-Yves Oudeyer. Sim-to-real transfer with neural-augmented robot simulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 817–828. PMLR, 29–31 Oct 2018.
- [14] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.
- [15] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011.
- [16] 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.
- [17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proc. of Int’l Conf. on Machine Learning (ICML)*, pages 1989–1998, 2018.
- [18] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.
- [19] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [20] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image captioning with very scarce supervised data: Adversarial semi-supervised learning approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.
- [21] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, Youngjin Yoon, and In So Kweon. Disjoint multi-task learning between heterogeneous human-centric tasks. In *Proc. of Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [22] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2020.
- [23] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020.
- [24] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 1785–1792. IEEE, 2011.
- [25] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proc. of Int’l Conf. on Computer Vision (ICCV)*, pages 5542–5550, 2017.
- [26] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. Domain generalization and adaptation using low rank exemplar svms. In *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, volume 40, pages 1114–1127. IEEE, 2017.

- [27] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.
- [28] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Transferable representation learning with deep adaptation networks. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 41(12):3071–3085, 2018.
- [29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.
- [30] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [31] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. *arXiv preprint arXiv:2008.12197*, 2020.
- [32] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proc. of Int’l Conf. on Computer Vision (ICCV)*, pages 5715–5725, 2017.
- [33] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4500–4509, June 2018.
- [34] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.
- [35] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proc. of Int’l Conf. on Computer Vision (ICCV)*, pages 754–763, 2017.
- [36] Kwanyong Park, Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Preserving semantic and temporal consistency for unpaired video-to-video translation. In *Proc. of the ACM International Conference on Multimedia (ACM MM)*, pages 1248–1257, 2019.
- [37] Kwanyong Park, Sanghyun Woo, Inkyu Shin, and In-So Kweon. Discover, hallucinate, and adapt: Open compound domain adaptation for semantic segmentation. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2020.
- [38] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schuster, Amit K. Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. In *European Conference on Computer Vision (ECCV)*, 2020.
- [39] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings, 2020.
- [40] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017.
- [41] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proc. of European Conf. on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.
- [42] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy, 2019.
- [43] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.
- [44] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Proc. of Neural Information Processing Systems (NeurIPS)*, pages 2110–2118, 2016.
- [45] Burr Settles. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114, 2012.
- [46] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017.
- [47] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *European Conference on Computer Vision*, pages 532–548. Springer, 2020.
- [48] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 2107–2116, 2017.
- [49] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. 2020.
- [50] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation, 2020.
- [51] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schuster, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 7472–7481, 2018.
- [52] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.
- [53] Zhonghao Wang, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-Mei Hwu, Thomas S. Huang, and Humphrey Shi. Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation, 2020.
- [54] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen mei Hwu, Thomas S. Huang, and Humphrey Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation, 2020.

- [55] Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9011–9020, 2020.
- [56] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.
- [57] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.
- [58] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 289–305, 2018.
- [59] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.
- [60] Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.