SegDA: Maximum Separable Segment Mask with Pseudo Labels for Domain Adaptive Semantic Segmentation

Anant Khandelwal*
Applied Scientist, Amazon
anantkha@amazon.com

Abstract

Unsupervised Domain Adaptation (UDA) aims to solve the problem of label scarcity of the target domain by transferring the knowledge from the label rich source domain. Usually, the source domain consists of synthetic images for which the annotation is easily obtained using the well known computer graphics techniques. However, obtaining annotation for real world images (target domain) require lot of manual annotation effort and is very time consuming because it requires per pixel annotation. To address this problem we propose SegDA module to enhance transfer performance of UDA methods by learning the maximum separable segment representation. This resolves the problem of identifying visually similar classes like pedestrian/rider, sidewalk/road etc. We leveraged Equiangular Tight Frame (ETF) classifier inspired from Neural Collapse for maximal separation between segment classes. This causes the source domain pixel representation to collapse to a single vector forming a simplex vertices which are aligned to the maximal separable ETF classifier. We use this phenomenon to propose the novel architecture for domain adaptation of segment representation for target domain. Additionally, we proposed to estimate the noise in labelling the target domain images and update the decoder for noise correction which encourages the discovery of pixels for classes not identified in pseudo labels. We have used four UDA benchmarks simulating synthetic-to-real, daytime-to-nighttime, clear-to-adverse weather scenarios. Our proposed approach outperforms +2.2 mIoU on GTA → Cityscapes, +2.0 mIoU on Synthia → Cityscapes, +5.9 mIoU on Cityscapes → DarkZurich, +2.6 mIoU on Cityscapes → ACDC.

1. Introduction

With the success of Convolutional Neural Networks (CNN) [3, 23] and Vision Transformers [22, 57] based models on the task of semantic segmentation, there is an increase in interest in adopting the semantic segmentation models in production for autonomous vehicles. However, the success of these models has been shown on the synthetic datasets since obtaining per pixel annotation of synthetic datasets can be generated easily with computer graphics [32, 33], but obtaining these for real world is very costly since it requires lot of time for annotation of large number of images, in absence of which the deep neural network will not able to generalize for every type of scenario. There exists domain gaps between the synthetic and real world images like illumination, weather, and camera quality [9, 45, 49]. To achieve the generalization on real images without any labelled dataset, researchers resort to unsupervised domain adaptation (UDA) techniques either through network changes or data augmentation to source domain (synthetic) to be able to transfer learned knowledge from source domain to target domain environment.

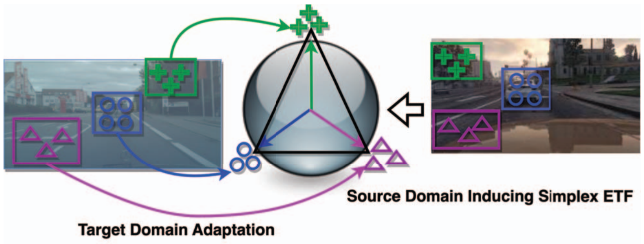

Figure 1: The idea of our proposed framework is to not only adapt the pixel representation in source domain to target domain but also make them aligned to the corresponding segment representation (collapsed representation) which is parallel to the classifier weight of corresponding classes and hence ensures maximum separability.

Existing works leveraged adversarial learning [26, 27, 29, 40, 44, 53], self-supervised learning [19, 28, 39, 55, 62, 65, 66] to learn domain invariant representations. Some minimizes this domain discrepancy at pixel level [13, 21, 49, 53], feature level [17, 25] or prediction level [29, 40, 41, 44]. Self-Supervised learning aims to mine the visual

[*] Work Done while author was in Amazon

knowledge from unlabelled images and pose the optimization objective to make these visual cues to be as closer to the ones in source domain, towards that some works have adopted augmentations to source domain like rotation [20], colorization [56], mixup [37] and random erasing [61]. However, a common issue with these methods is that their end-to-end network is very simple, relying either only on the data augmentation or techniques like use of variable dropout, teaching the network using supervision from discriminator to generate the consistent prediction between source and target domain. However, this does not resolve the problem of confusion between classes of similar visual appearances like *road/sidewalk* or *pedestrian/rider*. As shown in Figure 3 the ground truth label of sidewalk is incorrectly predicted in the SOTA segmentation models named DAFormer[15] and HRDA[16]. To solve this problem we propose the use Equiangular Tight Frame (ETF classifier inspired from Neural Collapse [30] ensuring the maximal separability between classes. The phenomenon states that when the neural network trained towards zero loss, the terminal layer features of each collapse to forming a ETF simplex and the corresponding collapsed feature vectors for each class is aligned with classifier weights. With the same phenomenon, we adapt the image encoder of neural network for target domain using the segment representation obtained from the target domain (along with collapsed representation of source domain) and align it with the ETF classifier weights (as shown in Figure 1). This helps to measure the noise remains in the pixel decoder and apply the noise correction training for pixel decoder. We additionally introduced the pixel discovery training for the possibility of pixel belonging to the new class and keep on introducing them via the pseudo labels obtained from the moving average based teacher network. This complete setting enables us to achieve the +2.2 mIoU on GTA → Cityscapes, +2.0 mIoU on Synthia → Cityscapes, +5.9 mIoU on Cityscapes → DarkZurich, +2.6 mIoU on Cityscapes → ACDC. The UDA benchmark of DarkZurich and ACDC correspond to images in nighttime and adverse weather conditions, achieving improvement in these UDA benchmarks proved the efficiency of our approach.

2. Related Work

Unsupervised Domain Adaptation (UDA) aims to solve the label scarcity problem for target domain with the successful transfer of knowledge from label rich source domain. Some CycleGAN[64] based methods [13, 48] does exactly the task of visual style transfer from source domain to the target domain. These methods belong to a major category of adversarial learning methods[26, 27, 29, 40, 44, 53], which aims to learn the domain invariant representation based on min max optimization strategy, where a feature extractor is trained to fool a discriminator and thus helps to obtain the adapted feature representations. However, as shown in [60], this type of training is unstable leading to suboptimal performance. This is followed by another line works of training the segmentation network on target domain with pseudo label which can be pre-computed either offline [53, 65] or updated online during training iterations [15, 39]. Irrespective of the way of generating pseudo labels, there is inevitable noise (due to underlying difference in data distribution between domains) in the pseudo labels which make the training noisy and leading to sub optimal performance. Some adopted the use of high confidence pseudo labels [66, 65], some conducted domain alignment for reliable pseudo labels [58] and some works leveraging uncertainty estimation [59] and efficient sampling [28]. Apart from the works mentioned above researchers also adopted combining adversarial learning and self training with specialized entropy minimization schemes[4, 44], semantic prototype based contrastive learning method for class alignment [51], visual pretraining [46], contrastive learning between features using different saliency masks [42]. Our proposed method is orthogonal to all the above approaches and adds value on top of SOTA methods as proved through qualitative and quantitative analysis over four UDA benchamrks.

3. Methods

We start by introducing problem statement and the basic understanding of semantic segmentation along with the corresponding loss functions in supervised setting and domain adaptation setting. Following this we described our proposed model i.e. *SegDA* and the modelling of maximum separable segments under the label noise implicit in pseudo labels. Finally, we discussed the the utility of loss functions in identifying the regions not highlighted in pseudo labels and the corrected loss for segmentation under label noise.

Problem Statement: Given the source domain data containing images $\mathcal{X}^S = \{x_k^S\}_{k=1}^{N_S}$, labelled by $\mathcal{Y}^S = \{y_k^S\}_{k=1}^{N_S}$ and the unlabelled target domain $\mathcal{X}^T = \{x_k^T\}_{k=1}^{N_T}$, where N_S and N_T are the number of images in source and target domain respectively. The label map of source domain $\mathcal{Y}^S$ contains C categories. The setting for domain adaptive semantic segmentation requires to learn the function able to map the unlabelled images $\mathcal{X}^T$ to their semantic segmentation labels $\mathcal{Y}^T$ without the supervision of ground truth target domain labels.

Semantic Segmentation: A neural network supervised training on labelled images follows the existing works [58, 66] and the supervised loss is formulated as:

$$\mathcal{L}_k^S = \mathcal{H}(f_\theta(x_k^S), y_k^S), \mathcal{L}^S = \frac{1}{N_S}\sum_{k=1}^{N_S}\mathcal{L}_k^S \qquad (1)$$

$$\mathcal{H}(\hat{y}, y) = -\sum_{i=1}^{H}\sum_{i=1}^{W}\sum_{i=1}^{C} y_{ijc} \log \hat{y}_{ijc} \quad (2)$$

However, this setting can only be applied to source domain where labelled data is accessible. For the target domain in the absence of ground-truth labels, predictions from source trained model on target domain does not show the similar performance as on the source domain because of underlying difference in the dataset distribution like source domain consists of synthetic images while the target domain contains the images from real world. This requires to adapt the model trained on source domain to the unlabelled target domain $\mathcal{X}^T$. Similar to [39, 63] we also handled the problem of label scarcity with pseudo labels $\tilde{\mathcal{Y}}_T = \{\tilde{Y}_k^T \in \{0, 1\}^{H \times W \times C}\}_{k \in N_T}$ by teacher network $\bar{f}_\theta$ updated during training of student f_θ with exponential moving average of weights of student network at each training iteration[38, 60]. The loss function with pseudo labels is given as follows:

$$\mathcal{L}_k^T = \mathcal{H}(f_\theta(x_k^T), \bar{Y}_k^T),\ \mathcal{L}_T = \frac{1}{N_T}\sum_{k=1}^{N_T} \mathcal{L}_k^T \quad (3)$$

But however the training with pseudo labels is noisy, and hence in practice [11] we also account for only confident predictions (greater than the threshold τ_h) to contribute in the loss function. However, we additionally implemented the noise estimation in the segmentation loss (denoted as $\mathcal{L}_{corr}$) for segment classes identified from pseudo labels (denoted as C'). To be able to discover the new segment classes not present in pseudo labels we consider the pixels having $f_\theta(x_k^T) < \tau_l$ (lower threshold) and $\bar{y}_k^T = \text{argmax} f_\theta(x_k^T) > C'$ incorporated in the loss function $\mathcal{L}_{dis}$. Further, we ensure the maximum separability of each of the segment using the feature collapse property of Neural Collapse [30] and use the representation obtained from segment decoder to adapt the pixel classifier to the target domain representation (incorporated in the loss $\mathcal{L}_{dapt}$). To avoid forgetting the source domain, the collapsed segments representation from source domain is plugged in $\mathcal{L}_{mem}$ along with domain adaptation loss $\mathcal{L}_{dapt}$. The end-to-end network for domain adaptive semantic segmentation consists of pixel level module, segment level module and joint classifier for both pixel and segment as shown in Figure 2. Only pixel level module and classifier collectively called f_θ is the adapted semantic segmentation network and hence deployed in production for inference after domain adaptation training.

Pixel-Level Module outputs the d-dimensional representation for each pixel in an image of size $H \times W$. It consists of an encoder which generates the low resolution image feature map denoted as $\mathcal{F} \in R^{C_\mathcal{E} \times \frac{H}{S} \times \frac{W}{S}}$ where $C_\mathcal{E}$ is the number of channels and S is the stride of the feature map. The feature map $\mathcal{F}$ is then gradually up sampled by the pixel decoder to output the d-dimensional pixel level feature map $\mathcal{E}_{pixel}^{d \times H \times W}$. Any existing pixel classification based segmentation model[16, 15, 5, 3, 57] fits this module, but however we described the encoder and pixel decoder outputs so as to leverage these in obtaining the segment representation (using segment decoder) and the (pixel, segment) classification using joint classifier.

Segment Level Module: To obtain the equivalent segment representation we convert the Transformer decoder[43] with N- positional embeddings as queries to the decoder with text embeddings obtained from CLIP text encoder [31] as queries. Specifically, the text embeddings are calculated for C' segment labels identified from the pseudo labels. These C' embeddings at each position as query is not trainable and the image features $\mathcal{F}$ as key and values are used to generate the segment representation $\mathcal{S}^{d_S \times C'}$ at the output of transformer decoder.

Joint Pixel and Segment Classifier: We proposed the neural collapse [30] inspired classifier capable of classifying both pixels and segment to the segment classes. Recent works have studied the practice of training DNN towards zero loss, this reveals that the classifier weights and last layer features collapse to form a geometric structure in the form of Equiangular Tight Frame (ETF). Essentially, the properties is stated as follows:

- ($\mathcal{NC}1$) **Variability Collapse**: Last layer features of a class collapse into within-class mean.
- ($\mathcal{NC}2$) **Convergence**: The within class means of all the classes converge to a vertices of a simplex ETF.
- ($\mathcal{NC}3$) **Classifier Convergence**: Within-class means aligned to their corresponding classifier weights and hence classifier will also converge to form a simplex ETF.

Neural collapse describes the optimal geometric structure of the classifier, following [30] we pre-fixed this optimality by fixing the learnable classifier structure to the simplex ETF. Therefore the segmentation network f_θ is consists of the pixel-level module denoted by $f_\theta^P \in R^{d \times H \times W}$ and the classifier $W_{ETF} \in R^{d \times C}$. The classifier weights are then initialized as per the simplex representation given as:

$$W_{ETF} = \sqrt{\frac{C}{C-1}}\mathbf{U}(\mathbf{I}_K - \frac{1}{C}\mathbf{1}_C\mathbf{1}_C^T) \quad (4)$$

where $W_{ETF} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3.......w_C] \in R^{d \times C}$, $\mathbf{U} \in R^{d \times C}$ allows the rotation, and satisfies $\mathbf{U}^T\mathbf{U} = \mathbf{I}_C$, $\mathbf{I}_C$ is an identity matrix and $\mathbf{1}_C$ is an all ones vector. This initialization offers W_{ETF} to be maximally pairwise separable. For any pair (c_1, c_2) of classifier W_{ETF} satisfies:

$$\mathbf{w}_{c_1}^T\mathbf{w}_{c_2} = \frac{C}{C-1}\delta_{c_1,c_2} - \frac{1}{C-1},\ \forall(c_1, c_2) \in [1, C] \quad (5)$$

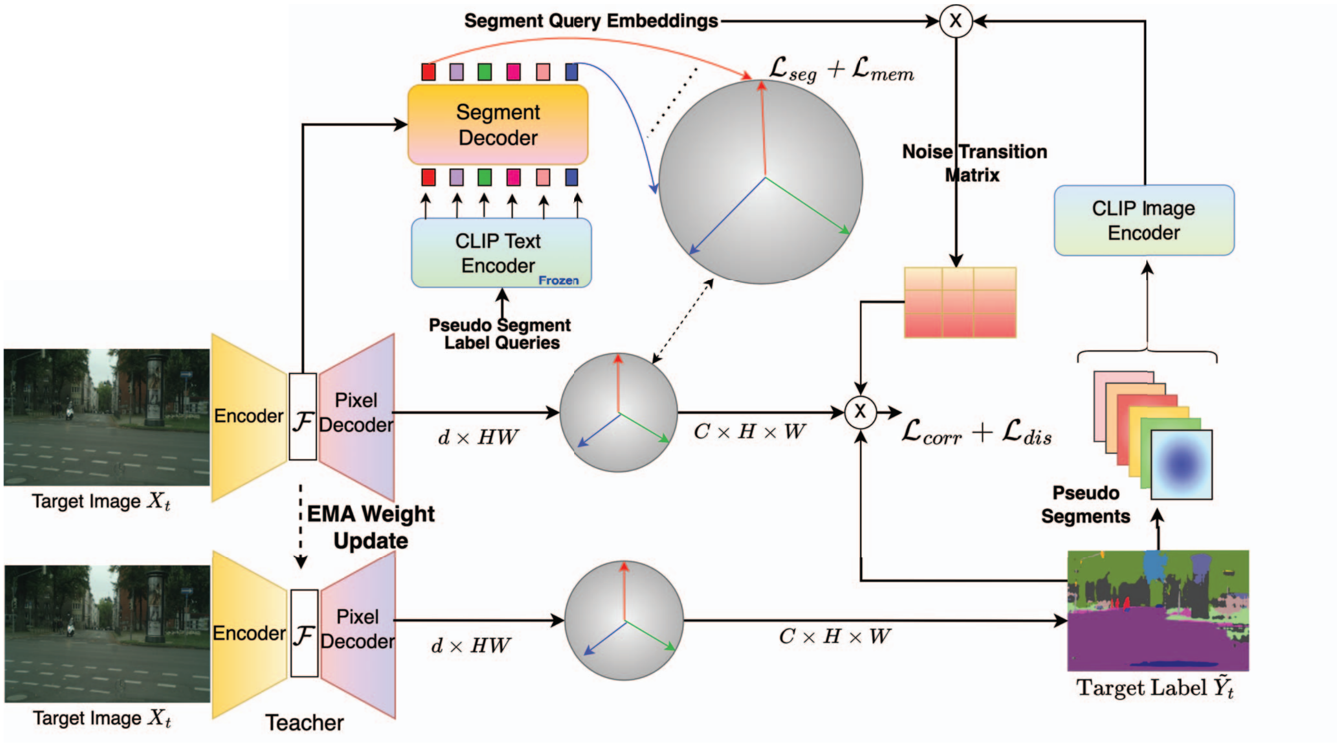

Figure 2: UDA with proposed method SegDA. The source domain trained model is adapted to target domain using domain adaptation loss $\mathcal{L}_{dapt}$, memory loss $\mathcal{L}_{mem}$ to retain the source information, noise correction loss $\mathcal{L}_{corr}$ and pixel discovery loss $\mathcal{L}_{dis}$. Overall SegDA enforces the noise correction in the pixel predictions made by an exponential moving average (EMA) teacher, where the noise is estimated using the adapted representations for each segment class which corresponds to the vector in a ETF Simplex, hence ensuring maximum separability.

During domain adaptation the source and target classes will remain same and hence the classifier prototypes trained for source domain and adapted for target domain. We utilize the dot-regression (DR) loss [54] for source domain training with W_{ETF} instead of cross-entropy (CE) loss since CE contains both the PUSH and PULL term, where PUSH term separates the feature vector of a class with classifier prototypes of different classes (but is inaccurate as highlighted in [30]) there we live only with PULL term which bring closer the feature vector of a class and the corresponding classifier prototype. The DR loss is formulated as:

$$\min_{\theta_P} \mathcal{L}(f^P_{\theta_i}, W_{ETF}) = \frac{1}{2}(\mathbf{w}^T_{c_i} f^P_{\theta_i} - 1)^2 \tag{6}$$

where $\mathbf{w}^T_{c_i}$, is the classifier prototype corresponding to class c_i and θ_P are the parameters of pixel module. The feature vector for each pixel is batch normalized using the batch normalization layer as the last layer. The loss in equation 6 is summed over each batch input x_i. The gradient of loss in equation 6 ($f^P_{\theta_i}$ is considered as optimization variables as in [7]) $\partial\mathcal{L}/\partial f^P_{\theta_i} = -(1 - \cos\angle(f^P_{\theta_i}, \mathbf{w}_{c_i}))\mathbf{w}_{c_i}$, which is effectively pulling the feature towards the classifier prototype for class c_i and hence converge to the simplex ETF classifier weights W_{ETF} resulting in collapsed representation for each class. The prediction score for all classes for a particular pixel representation is given as $\langle f^P_{\theta_i}, \mathbf{w}_{c_k} \rangle \forall c_k \in [1, C]$, results in predicted feature map $\mathcal{E}^{C\times H\times W}_{pred}$. To be able to domain adapt the classifier without forgetting the source domain we maintain the memory for each class collapsed features as the mean of all representations as per variability collapse in $\mathcal{NC}1$.

$$\mathcal{M}_{c_i} = \frac{1}{N_{c_i}} \sum_{i=1}^{N_{c_i}} f^P_{\theta_i} \tag{7}$$

N_{c_i} denotes the number of samples (pixels) across all the images in the training dataset. This results in memory $\mathcal{M} = [\mathcal{M}_{c_1}, \mathcal{M}_{c_2}, ..., \mathcal{M}_C]$. These memory vectors along the target domain segment representations are then used to adapt the classifier prototypes for each segment without forgetting the source domain.

3.1. Segment Representation Adaptation

The feature for every pixel belonging to the corresponding segment class collapse to within-class means, this collapsed representation is effectively the segment representation corresponding to each segment class, and hence the memory representations we have obtained for the source domain in equation 9 are the segment representation for corresponding class in the source domain. These source representations are aligned with the corresponding classifier prototypes of W_{ETF}. We adapt the target domain segment representations (obtained from segment level module) using these classifier prototypes. The idea of introducing the segment module is to denoise the pixel decoder to obtain pixel representation for target domain without forgetting the source domain and eliminating the requirement of keeping all source domain samples in the memory. For each of the pseudo label class $c \in [1, C']$ (identified in the input image from target domain) their corresponding label representation from CLIP Text Encoder [31] is obtained and used as query in the segment decoder as shown in Fig. 2. The domain adaptation loss for segment representation is formulated as:

$$\min_{\theta_e,\theta_{sd}} \mathcal{L}_{dapt} = \frac{1}{2}(\mathbf{w}_{c_i}^T \mathcal{S}_{c_i} - 1)^2,\ \forall c_i \in [1, C'] \qquad (8)$$

where θ_e, θ_{sd} are the parameters for encoder and segment decoder respectively. This loss will be summed over each batch containing sample x_i from target domain images. Since the training of encoder and segment decoder with target domain completely wipes out the source domain information, we add the memory loss formulated as:

$$\min_{\theta_P} \mathcal{L}_{mem} = \frac{1}{2}(\mathbf{w}_{c_i}^T \mathcal{M}_{c_i} - 1)^2,\ \forall c_i \in [1, C'] \qquad (9)$$

where parameters θ_P comprises the encoder parameters θ_e and pixel decoder parameters θ_{pd}.

3.2. Noise Estimation and Pixel Class Discovery

Training with memory loss and adaptation loss ensures the encoder to retain the information for source domain along with learning for target domain. However, pixel decoder contains only the source domain information and hence produces noisy pixel class distributions for target domain and hence it requires to denoise the pixel decoder. The error can eb of two types: 1) the noise between the classes $c \in [1, C']$ identified from the pseudo labels and 2) the noise due to incorrect prediction where the actual ground truth belongs to class category outside C'. For (1) we estimate the noise transition matrix for each segment class $c \in [1, C']$ identified from the pseudo labels. For (2) we propose a new loss which facilitates the discovery of pixels belong to class category $c \in [C' + 1, C]$. For each of segment masks identified from the pseudo labels we cropped the C' images from the predicted segments in each image. We obtain the noisy segment representation $\mathcal{S}_{noisy}^{d\times C'}$ from the cropped images for target domain using CLIP Image Encoder [31], where d is the embedding dimension. Effectively the noise transition matrix is given as $\mathcal{N} = \mathcal{S}^T \mathcal{S}_{noisy} \in R^{C'\times C'}$, where $d_S = d$. For the pseudo labels obtained from the mean teacher we use the max across class scores to predict the best class if it is greater than high threshold τ_h given as, $\text{argmax}_{c_k} \langle e_{c_k} > \tau_h \rangle\ \forall e_{c_k} \in \mathcal{E}_{pred}^{C\times H\times W}$ effectively resulting in C' classes forming different segments in a given image $\mathbf{x}_k$ denoted by $\tilde{y}_k^T \in \{0,1\}^{H\times W\times C'}$. For these C' classes the student predictions is denoted as $\mathcal{E}_{st,C'}$. Only the predictions belong to C' classes (from the student) can be noise corrected hence $\mathcal{E}_{st,C'}$ is noise corrected as $(\mathcal{N} \cdot \mathcal{E}_{st,C'})$, and the corrected loss is formulated as:

$$\mathcal{L}_{corr} = -\sum_{k\in N_T} \mathcal{H}(\mathcal{N} \cdot \mathcal{E}_{st,C'}, \tilde{y}_k^T)\ \forall c \in [1, C'] \qquad (10)$$

We want to ensure the discovery of pixels belonging to the classes outside C' and hence we consider the pixel scores from teacher $e_{c_k} < \tau_l$ (lower threshold) and the one-hot pseudo label is $\tilde{y}_k^o = \text{argmax}_{c_k}(e_{c_k}) > C'$. The threshold $\tau_h = 0.8$ and $\tau_l = 0.2$ in our case. The pixel discovery loss is then formulated as:

$$\mathcal{L}_{dis} = -\sum_{k\in N_T} \mathcal{H}(\mathcal{E}_{st} \setminus \tilde{y}_k^T, \tilde{y}_k^o)\ \forall c \in [C' + 1, C] \qquad (11)$$

where $\mathcal{E}_{st} \setminus \tilde{Y}_k^T$ denotes the student predictions outside the C' classes.

3.3. Overall Optimization Scheme

After training the pixel level module for source domain governed by loss given in equation 6, the source trained pixel module is then adapted to target domain governed by loss functions given in equation 8, 9, 10 and 11. The combined loss is given as:

$$\min_{\theta_P,\theta_{sd}} \mathcal{L}_{dapt} + \mathcal{L}_{mem} + \mathcal{L}_{corr} + \mathcal{L}_{dis} \qquad (12)$$

The teacher network ϕ_P (pixel module) is implemented as an EMA teacher [38]. Its weights are the exponential moving average (EMA) of the weights of the (student) network θ_P.

$$\phi_{P,t+1} \leftarrow \alpha\phi_{P,t} + (1 - \alpha)\theta_{P,t} \qquad (13)$$

where t is the training step. The EMA teacher effectively an ensemble of student models at different training steps, which is a most widely used learning strategy in semi-supervised setting [10, 14, 36, 38] and UDA [1, 15, 16, 39]. As the training grows the teacher is updated from student θ_P obtaining more context of what could be the stable pseudo labels based on the noise correction loss and the segment adaptation resulting in increased domain adapted performance on target domain.

4. Experiments

Datasets: We study the domain adaptation setting considering various realistic scenarios for street scenes i.e. synthetic-to-real, clear-to-adverse weather, and day-to-nighttime. There are public datasets available for both synthetic as well as realistic environments. For synthetic dataset, we use GTA [32] containing 24,966 training images (1914×1052 pixels) and Synthia [33] containing 9,400 images (1280 × 760 pixels). For clear weather, we use Cityscapes (CS) [6] consisting of 2,975 and 500 images (2048 × 1024 pixels) for training and validation respectively. For nighttime we use DarkZurich [34] with 2,416 and 151 images (19201080 pixels) for training and test respectively. For adverse weather (fog, night, rain, and snow) we use ACDC [35] containing 1,600, 406 and 2,000 images (1920 × 1080 pixels) for training, validation and test respectively. The training resolution as per the used UDA pixel level module (half-resolution for DAFormer [15] and full resolution for HRDA [16]).

Structure Details: We adopt the pixel level module following recent SOTA UDA setting [15, 50, 63] based on DAFormer network [15] consists of a MiT-B5 encoder [15, 52] pretrained on ImageNet-1k [8]. Following HRDA [16] we used the context aware feature fusion decoder (from DAFormer embedding dimension 768) and for scale attention decoder we use SegFormer MLP decoder [52] with an embedding dimension of 768 matching the dimension of Segment Decoder (Transformer Decoder [43] as in DETR[2]) and CLIP Image Encoder [31]. Specifically, to compare **SegDA** on various setting we used DAFormer[15], HRDA[16] and a DeepLabV2 [3] (with a ResNet-101 [12] backbone).

Implementation Details: We train our network on Titan RTX GPU for 40K training iterations and a batch size of 2. We adopted the multi-resolution training strategy from HRDA [16]. We adopted ADAMW [24] optimizer with a learning rate of 6×10^{-5} for encoder and 6×10^{-4} for the decoder with linear learning rate warmup. The applicable strategies like DACS [39] data augmentation, Rare Class Sampling [15], and ImageNet Feature Distance [15] is used as it is along with corresponding set of parameters from the respective UDA methods. For EMA teacher update we have used $\alpha = 0.999$. Following the setting [15, 16] we also adopted color augmentation (brightness, contrast, saturation, hue, and blur) during source domain training. We used mean intersection-over-union (mIoU) as the metric to evaluate our UDA method.

Reproducibility: Our code is based on Pytorch [18] and will be publicly available to reproduce all the results.

4.1. Comparisons with State-of-the-art Methods

To facilitate the comparison of SegDA with various SOTA methods, we first evaluated SegDA with different existing network architectures and UDA methods for domain adaptive semantic segmentation on GTA → CS. As shown in Table 1, with SegDA all the network architectures and UDA methods perform consistently better (ranging from +1.3 upto +4.9 mIoU) than without SegDA counterparts. This implies that the proposed domain adaptive framework SegDA not only benefit the CNN based architectures (like DeepLabV2 [3]) but able to perform better with Transformer based architectures as well like DAFormer [15]. As expected the performance improvement with advanced transformer architectures is less since the base performance of UDA w/o SegDA is already high in these cases.

Network	UDA Method	w/o SegDA	w/ SegDA	Diff.
DeepLabV2 [3]	Entropy Min. [44]	44.3	49.2	+4.9
DeepLabV2 [3]	DACS [39]	53.9	56.5	+2.6
DeepLabV2 [3]	DAFormer [15]	56.0	59.8	+3.8
DeepLabV2 [3]	HRDA [16]	63.0	64.3	+1.3
DAFormer [15]	DAFormer [15]	68.3	70.8	+2.5
DAFormer [15]	HRDA [16]	73.8	76.0	+2.2

Table 1: Performance (mIoU in %) comparison of different UDA methods with and without SegDA on GTA → CS

Going forward, we evaluate the performance of SegDA with the highest performing UDA method HRDA [16] for further comparison with SOTA methods on different UDA scenarios namely: synthetic-to-real (GTA → CS and SYNTHIA → CS), clear-to-adverse weather (CS → ACDC) and day-to-nighttime (CS → DarkZurich). The quantitative comparison among different SOTA methods has been shown in Table 2, and the qualitative comparison is shown (in Figure 3) in the form of a visual difference between the image, ground truth, two latest SOTA transformer based methods [15, 16] and the proposed method SegDA. Summarizing results from Table 2 SegDA outperforms both CNN based and Transformer based architectures including the recent transformer based SOTA methods namely, DAFormer [15] and HRDA [16]. It improves the state-of-the-art performance by +2.2 mIoU on GTA → Cityscapes(CS), +2.0 mIoU on SYNTHIA → Cityscapes, +5.9 mIoU on Cityscapes → DarkZurich, +2.6 mIoU on Cityscapes → ACDC. Moreover, SegDA performs better than SOTA on class-wise IoU as well on most of the classes. Specifically, it outperforms for all the classes in GTA → Cityscapes except on Wall which is a general class and mostly been occluded with different objects. Similarly, the performance per class on Cityscapes → ACDC outperforms the SOTA performance on each class, this proves the performance gain owning to ETF classifier and noise correction which handles the noise and separability even in the presence of adverse weather. Across UDA benchmarks, the classes that are on most advantage with SegDA are *Fence*, *Pole*, *Traffic Light*, *Terrain* and *Rider*. ETF classifier and noise cor-

Method	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
Synthetic-to-Real: GTA→Cityscapes																				
ADVENT [44]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
DACS [39]	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
ProDA [55]	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
DAFormer [15]	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
HRDA [16]	96.4	74.4	91.0	**61.6**	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79.0	52.9	93.9	84.1	85.7	75.9	63.9	67.5	73.8
SegDA+ HRDA	**97.7**	**80.1**	**91.4**	**61.6**	**56.9**	**59.8**	**66.1**	**71.4**	**91.8**	**51.6**	**94.5**	**79.9**	**56.2**	**94.7**	**85.5**	**90.4**	**80.5**	**64.5**	**68.5**	**76.0**
Synthetic-to-Real: Synthia→Cityscapes																				
ADVENT [44]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	–	84.1	57.9	23.8	73.3	–	36.4	–	14.2	33.0	41.2
DACS [39]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	–	90.8	67.6	38.3	82.9	–	38.9	–	28.5	47.6	48.3
ProDA [55]	**87.8**	45.7	84.6	37.1	0.6	44.0	54.6	37.0	88.1	–	84.4	74.2	24.3	88.2	–	51.1	–	40.5	45.6	55.5
DAFormer [15]	84.5	40.7	88.4	41.5	6.5	50.0	55.0	54.6	86.0	–	89.8	73.2	48.2	87.2	–	53.2	–	53.9	61.7	60.9
HRDA [16]	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	–	92.9	79.4	52.8	89.0	–	64.7	–	63.9	64.9	65.8
SegDA+ HRDA	87.2	**50.7**	**89.4**	**49.6**	**8.2**	**59.6**	**66.8**	**63.6**	**88.2**	-	**94.6**	**81.0**	**58.9**	**90.2**	-	**64.7**	-	**67.1**	**64.9**	**67.8**
Day-to-Nighttime: Cityscapes→DarkZurich																				
ADVENT [44]	85.8	37.9	55.5	27.7	14.5	23.1	14.0	21.1	32.1	8.7	2.0	39.9	16.6	64.0	13.8	0.0	58.8	28.5	20.7	29.7
MGCDA [34]	80.3	49.3	66.2	7.8	11.0	41.4	38.9	39.0	64.1	18.0	55.8	52.1	53.5	74.7	66.0	0.0	37.5	29.1	22.7	42.5
DANNet [47]	90.0	54.0	74.8	41.0	21.1	25.0	26.8	30.2	72.0	26.2	84.0	47.0	33.9	68.2	19.0	0.3	66.4	38.3	23.6	44.3
DAFormer [15]	93.5	65.5	73.3	39.4	19.2	53.3	44.1	44.0	59.5	34.5	66.6	53.4	52.7	82.1	52.7	9.5	89.3	50.5	38.5	53.8
HRDA [16]	90.4	56.3	72.0	39.5	19.5	57.8	52.7	43.1	59.3	29.1	70.5	60.0	58.6	84.0	75.5	11.2	90.5	51.6	40.9	55.9
SegDA+ HRDA	**94.8**	**75.2**	**84.1**	**55.3**	**28.7**	**62.1**	**52.7**	**52.7**	59.3	**46.9**	70.5	**65.4**	**61.8**	**84.1**	**75.6**	**18.5**	**91.3**	**52.7**	**44.3**	**61.8**
Clear-to-Adverse-Weather: Cityscapes→ACDC																				
ADVENT [44]	72.9	14.3	40.5	16.6	21.2	9.3	17.4	21.2	63.8	23.8	18.3	32.6	19.5	69.5	36.2	34.5	46.2	26.9	36.1	32.7
MGCDA [34]	73.4	28.7	69.9	19.3	26.3	36.8	53.0	53.3	75.4	32.0	84.6	51.0	26.1	77.6	43.2	45.9	53.9	32.7	41.5	48.7
DANNet [47]	84.3	54.2	77.6	38.0	30.0	18.9	41.6	35.2	71.3	39.4	86.6	48.7	29.2	76.2	41.6	43.0	58.6	32.6	43.9	50.0
DAFormer [15]	58.4	51.3	84.0	42.7	35.1	50.7	30.0	57.0	74.8	52.8	51.3	58.3	32.6	82.7	58.3	54.9	82.4	44.1	50.7	55.4
HRDA [16]	88.3	57.9	88.1	55.2	36.7	56.3	62.9	65.3	74.2	57.7	85.9	68.8	45.7	88.5	76.4	82.4	87.7	52.7	60.4	68.0
SegDA+ HRDA	**90.8**	**67.4**	**89.3**	**55.3**	**40.5**	**57.2**	**62.9**	**68.5**	**76.4**	**61.9**	**87.1**	**71.4**	**49.5**	**89.8**	**76.5**	**86.8**	**89.2**	**56.9**	**63.3**	**70.6**

Table 2: Semantic Segmentation performance (mIoU in %) for four UDA benchmarks

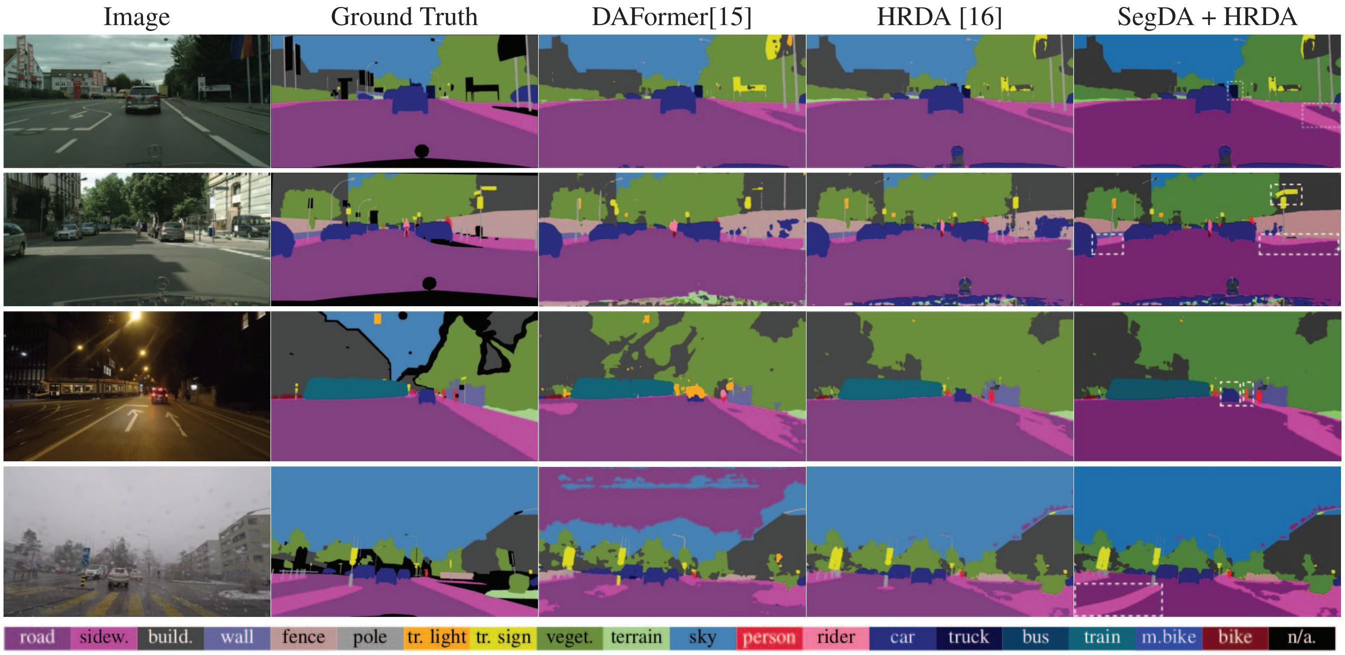

Figure 3: Qualitative comparison of SegDA with previous methods on GTA→CS (row 1 and 2), CS→ACDC (row 3), and CS→DarkZurich (row 4)

rection enforces the representation learned for these classes to be separable from commonly occurring classes, since they are dependent on context clues, HRDA models that better and ETF classifier helps to separate them from the most of the co-occurring classes. Most of the classes like *Road*, *Building* and *Vegetation* are showing the least improvements since they are pretty much general and hence easiest to identify therefore their IoU scores are already high by SOTA methods and hence very little improvement. The classes like *Bus* and *Vegetation* are not discoverable satisfactorily on Cityscapes → DarkZurich because of the night-time, it is not able to identify the green color for vegetation and color of vehicle as compared to black color which is present everywhere.

Qualitative Comparison: In Figure 3, the visual illustration of segmentation results shown to facilitate the comparison of the proposed method SegDA + HRDA over other two SOTA methods DAFormer[15] and HRDA[16] along with the corresponding ground truth for the image. Row 1 and 2 indicates segmentation results on GTA → CS, row 3 and 4 correspond to CS → ACDC and CS → DarkZurich respectively. The results highlighted by white dash boxes are the one captured correctly by SegDA as compared to other methods. Like n row 1, the SegDA is able to detect the *car* correctly while HRDA highlight the *car* along with *traffic sign* (blue along with yellow color). Also, we see that SegDA is able to identify the *traffic sign* correctly in row 2. All the segmentation results show the clear boundaries for the classes like *pole*, *fence*, *sidewalk* etc. This is because of the ETF classifier making the pixel representation to be maximally separable and noise correction further enhances the consistency at the pixel level.

Table 3: Ablation Study of SegDA with DAFormer[15] on GTA → CS

	ETF Classifier	Color Aug	EMA Teacher	Noise Correction	mIoU
1	-	-	-	-	68.3
2	✓	✓	✓	✓	70.8
3	-	✓	✓	✓	68.6
4	✓	-	✓	✓	70.6
5	✓	✓	-	✓	69.4
6	✓	✓	✓	-	66.8

4.2. Ablation Studies

In this section we present the analysis (in Table 3) of each of the individual components present in training of SegDA with DAFormer [15] (due to faster training) on GTA → CS. Training SegDA end-to-end with DAFormer [15](row 2) achieves +2.5 mIoU better than DAFormer alone (row 1). Further ablations in row 3-6 remove one component at each row indicated by '-', first we remove the ETF Classifier structure instead used the single layer MLP in place of that this setting reduces the performance by -2.2 mIoU resulting in almost the same performance as with DAFormer as the segment representations obtained from Transformer is not adapted and hence the noise handling is itself noisy which leads to EMA teacher being noisy and hence no advantage. Ablation with color augmentation reduces the performance only by -0.2 mIoU which is insignificant and hence not as much important as ETF classifier. Other components like Noise Correction (along with discovery) and EMA Teacher shows the performance reduction of -4.0 mIoU and -1.4 mIoU respectively. Indicating the highest importance of Noise correction if ETF classifier is present without which the domain adaptation is not satisfactorily since it guides how to discover the pixels for new classes and correct the confidence score of existing pixel predictions. The EMA teacher ablation confirms the stability of pseudo labels. With latest model in place of EMA Teacher the resultant domain adaptation reduces by -1.4 mIoU because of fluctuating predictions of pixels from one model to another and hence averaging makes the prediction consistently confident.

Table 4: Relative comparison of UDA GTA → CS and Supervised Training on CS. Rel. indicates mIoU$_{UDA}$/mIoU$_{Superv.}$

	mIoU$_{UDA}$	**mIoU$_{Superv.}$**	**Rel.**
DAFormer	68.3	77.6	88.0%
SegDA + DAFormer	70.8	77.8	91.0%
Improvement	+2.5	+0.2	+3.0%

Supervised Training: We compared the supervised and UDA performance of DAFormer with and without SegDA in Table 4. For SegDA with DAFormer on supervised setting the still the pseudo labels are generated and using EMA teacher and noise corrected towards the loss correction along with segement representation adaptation. This setting leads to very little improvement of +0.2 mIoU over the DAFormer alone. For UDA however the imporvement is +2.5 mIoU indicating the usefulness of segment representation adaption, ETF classifier, noise correction and EMA Teacher for daomin adaptation and generating stable pseudo labels. To quantify this relative improvement of eaach network setting for UDA and Supervised training we calculated Rel. as mIoU$_{UDA}$/mIoU$_{Superv.}$ indicated in last column of Table 4 indicating the DAFormer results in 88% improvement on UDA and 91% with SegDA applied over DAFormer. Overall there is 3% improvement by an addition of SegDA over DAFormer.

5. Conclusion

We proposed the UDA method SegDA, which is able to maximally separate the visually correlated classes with the method of noise correction in the pseudo labels as well as pixel discovery to the classes not present in the pseudo labels. Our method outperforms (on existing SOTA) by +2.2 mIoU on GTA → Cityscapes, +2.0 mIoU on Synthia → Cityscapes, +5.9 mIoU on Cityscapes → DarkZurich, +2.6 mIoU on Cityscapes → ACDC.

References

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15384–15394, 2021.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[3] 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.

[4] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2090–2099, 2019.

[5] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021.

[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.

[7] Hien Dang, Tan Nguyen, Tho Tran, Hung Tran, and Nhat Ho. Neural collapse in deep linear network: From balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*, 2023.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Jianwu Fang, Fan Wang, Peining Shen, Zhedong Zheng, Jianru Xue, and Tat-seng Chua. Behavioral intention prediction in driving scenes: A survey. *arXiv preprint arXiv:2211.00385*, 2022.

[10] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv preprint arXiv:1906.01916*, 2019.

[11] Xiaoqing Guo, Jie Liu, Tongliang Liu, and Yixuan Yuan. Simt: handling open-set noise for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7032–7041, 2022.

[12] 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.

[13] 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 *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[14] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Koring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11130–11140, 2021.

[15] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9924–9935, 2022.

[16] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 372–391. Springer, 2022.

[17] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 590–605, 2018.

[18] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021.

[19] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. *Advances in neural information processing systems*, 33:3569–3580, 2020.

[20] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International conference on learning representations (ICLR)*, 2018.

[21] 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.

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6778–6787, 2019.

[26] Yawei Luo, Ping Liu, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Category-level adversarial adaptation for semantic segmentation using purified features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):3940–3956, 2021.

[27] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[28] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 415–430. Springer, 2020.

[29] 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.

[30] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[32] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016.

[33] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[34] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3139–3153, 2020.

[35] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021.

[36] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[37] Chao Sun, Zhedong Zheng, Xiaohan Wang, Mingliang Xu, and Yi Yang. Self-supervised point cloud representation learning via separating mixed shapes. *IEEE Transactions on Multimedia*, 2022.

[38] 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.

[39] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021.

[40] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[41] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1456–1465, 2019.

[42] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10052–10062, 2021.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[44] 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.

[45] Tingyu Wang, Zhedong Zheng, Yaoqi Sun, Tat-Seng Chua, Yi Yang, and Chenggang Yan. Multiple-environment self-adaptive network for aerial-view geo-localization. *arXiv preprint arXiv:2204.08381*, 2022.

[46] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[47] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15769–15778, 2021.

[48] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[49] Zuxuan Wu, Xin Wang, Joseph E Gonzalez, Tom Goldstein, and Larry S Davis. Ace: Adapting to changing environments for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2121–2130, 2019.

[50] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[51] Binhui Xie, Kejia Yin, Shuang Li, and Xinjing Chen. Spcl: A new framework for domain adaptive semantic segmentation via semantic prototype-based contrastive learning. *arXiv preprint arXiv:2111.12358*, 2021.

[52] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[53] 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.

[54] Yibo Yang, Liang Xie, Shixiang Chen, Xiangtai Li, Zhouchen Lin, and Dacheng Tao. Do we really need a learnable classifier at the end of deep neural network? *arXiv preprint arXiv:2203.09081*, 2022.

[55] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021.

[56] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

[57] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[58] Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo. *arXiv preprint arXiv:1912.11164*, 2019.

[59] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision*, 129(4):1106–1120, 2021.

[60] Zhedong Zheng and Yi Yang. Adaptive boosting for domain adaptation: Toward robust predictions in scene segmentation. *IEEE Transactions on Image Processing*, 31:5371–5382, 2022.

[61] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[62] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Uncertainty-aware consistency regularization for cross-domain semantic segmentation. *Computer Vision and Image Understanding*, 221:103448, 2022.

[63] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Jiangmiao Pang, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Context-aware mixup for domain adaptive semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[64] 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.

[65] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[66] 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.