DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation

Lukas Hoyer
ETH Zurich
lhoyer@vision.ee.ethz.ch

Dengxin Dai
MPI for Informatics
ddai@mpi-inf.mpg.de

Luc Van Gool
ETH Zurich & KU Leuven
vangool@vision.ee.ethz.ch

Abstract

As acquiring pixel-wise annotations of real-world images for semantic segmentation is a costly process, a model can instead be trained with more accessible synthetic data and adapted to real images without requiring their annotations. This process is studied in unsupervised domain adaptation (UDA). Even though a large number of methods propose new adaptation strategies, they are mostly based on outdated network architectures. As the influence of recent network architectures has not been systematically studied, we first benchmark different network architectures for UDA and newly reveal the potential of Transformers for UDA semantic segmentation. Based on the findings, we propose a novel UDA method, DAFormer. The network architecture of DAFormer consists of a Transformer encoder and a multi-level context-aware feature fusion decoder. It is enabled by three simple but crucial training strategies to stabilize the training and to avoid overfitting to the source domain: While (1) Rare Class Sampling on the source domain improves the quality of the pseudo-labels by mitigating the confirmation bias of self-training toward common classes, (2) a Thing-Class ImageNet Feature Distance and (3) a learning rate warmup promote feature transfer from ImageNet pretraining. DAFormer represents a major advance in UDA. It improves the state of the art by 10.8 mIoU for GTA→Cityscapes and 5.4 mIoU for Synthia→Cityscapes and enables learning even difficult classes such as train, bus, and truck well. The implementation is available at https://github.com/lhoyer/DAFormer.

1. Introduction

In the last few years, neural networks have achieved overwhelming performance on many computer vision tasks. However, they require a large amount of annotated data in order to be trained properly. For semantic segmentation, annotations are particularly costly as every pixel has to be labeled. For instance, it takes 1.5 hours to annotate a single image of Cityscapes [12] while, for adverse weather conditions, it is even 3.3 hours [58]. One idea to circumvent this issue is training with synthetic data [55, 57]. However, commonly used CNNs [38] are sensitive to domain shift and generalize poorly from synthetic to real data. This issue is addressed in unsupervised domain adaptation (UDA) by adapting the network trained with source (synthetic) data to target (real) data without access to target labels.

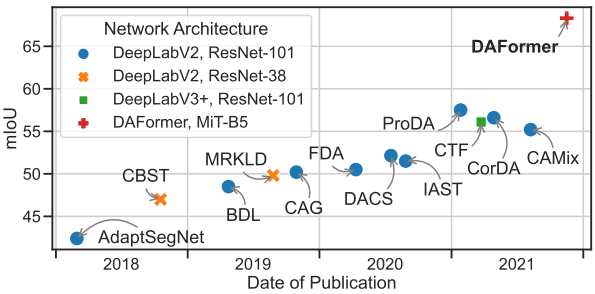

Figure 1. Progress of UDA over time on GTA→Cityscapes. Most previous UDA methods are evaluated using the outdated DeepLabV2 architecture. We rethink the design of the network architecture as well as its training strategies for UDA and propose DAFormer, significantly outperforming previous methods.

Previous UDA methods mostly evaluated their contributions using a DeepLabV2 [6] or FCN8s [46] network architecture with ResNet [24] or VGG [60] backbone in order to be comparable to previously published works. However, even their strongest architecture (DeepLabV2+ResNet101) is outdated in the area of supervised semantic segmentation. For instance, it only achieves a supervised performance of 65 mIoU [68] on Cityscapes while recent networks reach up to 85 mIoU [64, 86]. Due to the large performance gap, it stands to question whether using outdated network architectures can limit the overall performance of UDA and can also misguide the benchmark progress of UDA. In order to answer this question, this work studies the influence of the network architecture for UDA, compiles a more sophisticated architecture, and successfully applies it to UDA with a few simple, yet crucial training strategies. Naively using a

more powerful network architecture for UDA might be suboptimal as it can be more prone to overfitting to the source domain. Based on a study of different semantic segmentation architectures evaluated in a UDA setting, we compile DAFormer, a network architecture tailored for UDA (Sec 3.2). It is based on recent Transformers [14,83], which have been shown to be more robust than the predominant CNNs [3]. We combine them with a context-aware multi-level feature fusion, which further enhances the UDA performance. To the best of our knowledge, DAFormer is the first work to reveal the significant potential of Transformers for UDA semantic segmentation.

Since more complex and capable architectures are more prone to adaptation instability and overfitting to the source domain, in this work, we introduce three training strategies to UDA to address these issues (Sec. 3.3). First, we propose Rare Class Sampling (RCS) to consider the long-tail distribution of the source domain, which hinders the learning of rare classes, especially in UDA due to the confirmation bias of self-training toward common classes. By frequently sampling images with rare classes, the network can learn them more stably, which improves the quality of pseudo-labels and reduces the confirmation bias. Second, we propose a Thing-Class ImageNet Feature Distance (FD), which distills knowledge from diverse and expressive ImageNet features in order to regularize the source training. This is particularly helpful as the source domain is limited to only a few instances of certain classes (low diversity), which have a different appearance than the target domain (domain shift). Without FD this would result in learning less expressive and source-domain-specific features. As ImageNet features were trained for thing-classes, we restrict the FD to regions of the image that are labeled as a thing-class. And third, we introduce learning rate warmup [22] newly to UDA. By linearly increasing the learning rate up to the intended value in the early training, the learning process is stabilized and features from ImageNet pretraining can be better transferred to semantic segmentation.

DAFormer outperforms previous methods by a large margin (see Fig. 1) supporting our hypothesis that the network architecture and appropriate training strategies play an important role for UDA. On GTA→Cityscapes, we improve the mIoU from 57.5 [88] to 68.3 and on Synthia→Cityscapes from 55.5 [88] to 60.9. In particular, DAFormer learns even difficult classes that previous methods struggled with. For instance, we improve the class *train* from 16 to 65 IoU, *truck* from 49 to 75 IoU, and *bus* from 59 to 78 IoU on GTA→Cityscapes. Overall, DAFormer represents a major advance in UDA. Our framework can be trained in one stage on a single consumer RTX 2080 Ti GPU within 16 hours, which simplifies its usage compared to previous methods such as ProDA [88], which requires training multiple stages on four V100 GPUs for several days.

2. Related Work

Semantic Image Segmentation Since the introduction of Convolutional Neural Networks (CNNs) [38] for semantic segmentation by Long *et al*. [46], they have been dominating the field. Typically, semantic segmentation networks follow an encoder-decoder design [2, 46, 56]. To overcome the problem of the low spatial resolution at the bottleneck, remedies such as skip connections [56], dilated convolutions [5,85], or resolution preserving architectures [62] were proposed. Further improvements were achieved by harnessing context information, for instance using pyramid pooling [6, 7, 33, 89] or attention modules [17, 34, 78, 86]. Inspired by the success of the attention-based Transformers [71] in natural language processing, they were adapted to image classification [14, 66] and semantic segmentation [45, 83, 90] achieving state-of-the-art results. For image classification, CNNs were shown to be sensitive to distribution shifts such as image corruptions [27], adversarial noise [63], or domain shifts [26]. Recent works [3, 51, 53] show that Transformers are more robust than CNNs with respect to these properties. While CNNs focus on textures [19], Transformers put more importance on the object shape [3, 51], which is more similar to human vision [19]. For semantic segmentation, ASPP [7] and skip connections [56] were reported to increase the robustness [35]. Further, Xie *et al*. [83] showed that their Transformer-based architecture improves the robustness over CNN-based networks. To the best of our knowledge, the influence of recent network architectures on the UDA performance of semantic segmentation has not been systematically studied yet.

Unsupervised Domain Adaptation (UDA) UDA methods can be grouped into adversarial training and self-training approaches. Adversarial training methods aim to align the distributions of source and target domain at input [20, 29], feature [30, 68], output [68, 72], or patch level [69] in a GAN framework [18, 21]. Using multiple scales [8, 68] or category information [15, 48, 80] for the discriminator can refine the alignment. In self-training, the network is trained with pseudo-labels [39] for the target domain. Most of the UDA methods pre-compute the pseudo-labels offline, train the model, and repeat the process [13,84,92,93]. Alternatively, pseudo-labels can be calculated online during the training. In order to avoid training instabilities, pseudo-label prototypes [88] or consistency regularization [61,65] based on data augmentation [1,9,50] or domain-mixup [67, 91] are used. Several methods also combine adversarial and self-training [37,40,74], train with auxiliary tasks [32,73,75], or perform test-time UDA [76].

Datasets are often imbalanced and follow a long-tail distribution, which biases models toward common classes [79]. Strategies to address this problem are re-

sampling [23,25,81], loss re-weighting [42,59], and transfer learning [36, 43]. Also in UDA, re-weighting [49, 92] and class-balanced sampling for image classification [54] were applied. We extend class-balanced sampling from classification to semantic segmentation and propose Rare Class Sampling, which addresses the co-occurrence of rare and common classes in a single semantic segmentation sample. Further, we demonstrate that re-sampling is particularly effective to train Transformers for UDA.

Li *et al*. [41] have shown that knowledge distillation [28] from an old task can act as a regularizer for a new task. This concept was successfully deployed with ImageNet features to semi-supervised learning [31] and adversarial UDA [8]. We apply this idea to self-training, show that it is particularly beneficial for Transformers, and improve it by restricting the Feature Distance to image regions with thing-classes [4] as ImageNet mostly labels thing-classes.

3. Methods

3.1. Self-Training (ST) for UDA

First, we will give an overview over our baseline UDA method for evaluating different network architectures. In UDA, a neural network g_θ is trained using source domain images $\mathcal{X}_S = \{x_S^{(i)}\}_{i=1}^{N_S}$ and one-hot labels $\mathcal{Y}_S = \{y_S^{(i)}\}_{i=1}^{N_S}$ in order to achieve a good performance on target images $\mathcal{X}_T = \{x_T^{(i)}\}_{i=1}^{N_T}$ without having access to the target labels $\mathcal{Y}_T$. Naively training the network g_θ with a categorical cross-entropy (CE) loss on the source domain

$$\mathcal{L}_S^{(i)} = -\sum_{j=1}^{H\times W}\sum_{c=1}^{C} y_S^{(i,j,c)} \log g_\theta(x_S^{(i)})^{(j,c)} \tag{1}$$

usually results in a low performance on target images as the network does not generalize well to the target domain.

To address the domain gap, several strategies have been proposed that can be grouped into adversarial training [30, 68,74] and self-training (ST) [67,88,92] approaches. In this work, we use ST as adversarial training is known to be less stable and is currently outperformed by ST methods [67, 88]. To better transfer the knowledge from the source to the target domain, ST approaches use a teacher network h_ϕ (which we will describe later) to produce pseudo-labels for the target domain data

$$p_T^{(i,j,c)} = [c = \arg\max_{c'} h_\phi(x_T^{(i)})^{(j,c')}], \tag{2}$$

where $[\cdot]$ denotes the Iverson bracket. Note that no gradients will be backpropagated into the teacher network. Additionally, a quality / confidence estimate is produced for the pseudo-labels. Here, we use the ratio of pixels exceeding a threshold τ of the maximum softmax probability [67]

$$q_T^{(i)} = \frac{\sum_{j=1}^{H\times W}[\max_{c'} h_\phi(x_T^{(i)})^{(j,c')} > \tau]}{H\cdot W}. \tag{3}$$

The pseudo-labels and their quality estimates are used to additionally train the network g_θ on the target domain

$$\mathcal{L}_T^{(i)} = -\sum_{j=1}^{H\times W}\sum_{c=1}^{C} q_T^{(i)} p_T^{(i,j,c)} \log g_\theta(x_T^{(i)})^{(j,c)}. \tag{4}$$

The pseudo-labels can be generated either online [1, 67, 91] or offline [84, 92, 93]. We opted for online ST due to its less complex setup with only one training stage. This is important as we compare and ablate various network architectures. In online ST, h_ϕ is updated based on g_θ during the training. Commonly, the weights h_ϕ are set as the exponentially moving average of the weights of g_θ after each training step t [65] to increase the stability of the predictions

$$\phi_{t+1} \leftarrow \alpha\phi_t + (1-\alpha)\theta_t. \tag{5}$$

ST has been shown to be particularly efficient if the student network g_θ is trained on augmented target data, while the teacher network h_ϕ generates the pseudo-labels using non-augmented target data for semi-supervised learning [16, 61, 65] and unsupervised domain adaptation [1, 67] In this work, we follow DACS [67] and use color jitter, Gaussian blur, and ClassMix [52] as data augmentations to learn more domain-robust features.

3.2. DAFormer Network Architecture

Previous UDA methods mostly evaluate their contributions using a (simplified) DeepLabV2 network architecture [6, 68], which is considered to be outdated. For that reason, we compile a network architecture that is tailored for UDA to not just achieve good supervised performance but also provide good domain-adaptation capabilities.

For the encoder, we aim for a powerful yet robust network architecture. We hypothesize that robustness is an important property in order to achieve good domain adaptation performance as it fosters the learning of domain-invariant features. Based on recent findings [3, 51, 53] and an architecture comparison for UDA, which we will present in Sec. 4.2, Transformers [14, 66] are a good choice for UDA as they fulfill these criteria. Although the self-attention from Transformers [71] and the convolution both perform a weighted sum, their weights are computed differently: in CNNs, the weights are learned during training but fixed during testing; in the self-attention mechanism, the weights are dynamically computed based on the similarity or affinity between every pair of tokens. As a consequence, the self-similarity operation in the self-attention mechanism provides modeling means that are potentially more adaptive and general than convolution operations.

In particular, we follow the design of Mix Transformers (MiT) [83], which are tailored for semantic segmentation. The image is divided into small patches of a size of 4×4 (instead of 16×16 as in ViT [14]) in order to preserve

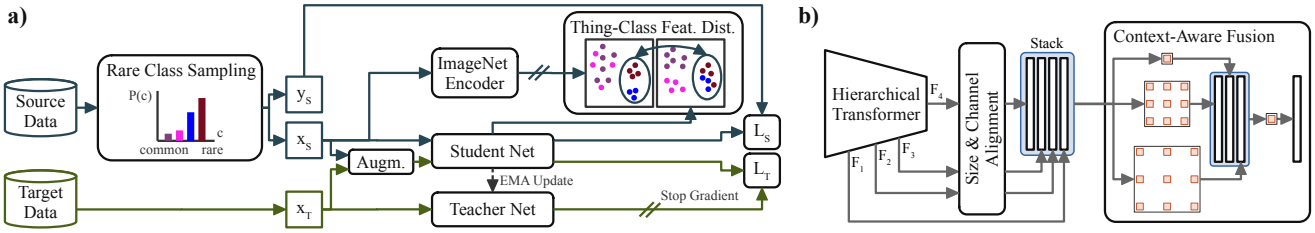

Figure 2. Overview of our UDA framework with Rare Class Sampling, Thing-Class Feature Distance, and DAFormer network.

details for semantic segmentation. To cope with the high feature resolution, sequence reduction [77] is used in the self-attention blocks. The transformer encoder is designed to produce multi-level feature maps $F_i \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i}$. The downsampling of the feature maps is implemented by overlapping patch merging [83] to preserve local continuity.

Previous works on semantic segmentation with Transformer backbones usually exploit only local information for the decoder [77, 83, 90]. In contrast, we propose to utilize additional context information in the decoder as this has been shown to increase the robustness of semantic segmentation [35], a helpful property for UDA. Instead of just considering the context information of the bottleneck features [6,7], DAFormer uses the context across features from different encoder levels as the additional earlier features provide valuable low-level concepts for semantic segmentation at a high resolution, which can also provide important context information. The architecture of the DAFormer decoder is shown in Fig. 2 (b). Before the feature fusion, we embed each F_i to the same number of channels C_e by a 1×1 convolution, bilinearly upsample the features to the size of F_1, and concatenate them. For the context-aware feature fusion, we use multiple parallel 3×3 depthwise separable convolutions [10] with different dilation rates [85] and a 1×1 convolution to fuse them, similar to ASPP [7] but without global average pooling. In contrast to the original use of ASPP [7], we do not only apply it to the bottleneck features F_4 but use it to fuse all stacked multi-level features. Depthwise separable convolutions have the advantage that they have a lower number of parameters than regular convolutions, which can reduce overfitting to the source domain.

3.3. Training Strategies for UDA

One challenge of training a more capable architecture for UDA is overfitting to the source domain. To circumvent this issue, we introduce three strategies to stabilize and regularize the UDA training: Rare Class Sampling, Thing-Class ImageNet Feature Distance, and learning rate warmup. The overall UDA framework is shown in Fig. 2 (a).

Rare Class Sampling (RCS) Even though our more capable DAFormer is able to achieve better performance on difficult classes than other architectures, we observed that the UDA performance for classes that are rare in the source dataset varies significantly over different runs. Depending on the random seed of the data sampling order, these classes are learned at different iterations of the training or sometimes not at all as we will show in Sec. 4.4. The later a certain class is learned during the training, the worse is its performance at the end of the training. We hypothesize that if relevant samples containing rare classes only appear late in the training due to randomness, the network only starts to learn them later, and more importantly, it is highly possible that the network has already learned a strong bias toward common classes making it difficult to 're-learn' new concepts with very few samples. This is further reinforced as the bias is confirmed by ST with the teacher network.

To address this, we propose Rare Class Sampling (RCS). It samples images with rare classes from the source domain more often in order to learn them better and earlier. The frequency f_c of each class c in the source dataset can be calculated based on the number of pixels with class c

$$f_c = \frac{\sum_{i=1}^{N_S} \sum_{j=1}^{H \times W} [y_S^{(i,j,c)}]}{N_S \cdot H \cdot W}. \tag{6}$$

The sampling probability $P(c)$ of a certain class c is defined as a function of its frequency f_c

$$P(c) = \frac{e^{(1-f_c)/T}}{\sum_{c'=1}^{C} e^{(1-f_{c'})/T}}. \tag{7}$$

Therefore, classes with a smaller frequency will have a higher sampling probability. The temperature T controls the smoothness of the distribution. A higher T leads to a more uniform distribution, a lower T to a stronger focus on rare classes with a small f_c. For each source sample, a class is sampled from the probability distribution $c \sim P$ and an image is sampled from the subset of data containing this class $x_S \sim \text{uniform}(\mathcal{X}_{S,c})$. Eq. 7 allows to over-sample images containing rare classes ($P(c) \geq 1/C$ if f_c is small). As a rare class (small f_c) usually co-occurs with multiple common classes (large f_c) in a single image, it is beneficial to sample rare classes more often than common classes ($P(c_{rare}) > P(c_{common})$) to get closer to a balance of the

re-sampled classes. For example, the common class road co-occurs with rare classes such as bus, train, or motorcycle and is therefore already covered when sampling images with these rare classes. When decreasing T, more pixels of classes with small f_c are sampled but also fewer pixels of classes with medium f_c. The temperature T is chosen to reach a balance of the number of re-sampled pixels of classes with small and medium f_c by maximizing the number of re-sampled pixels of the class with the least.

Thing-Class ImageNet Feature Distance (FD) Commonly, the semantic segmentation model g_θ is initialized with weights from ImageNet classification to start with meaningful generic features. Given that ImageNet also contains real-world images from some of the relevant high-level semantic classes, which UDA often struggles to distinguish such as train or bus, we hypothesize that the ImageNet features can provide useful guidance beyond the usual pretraining. In particular, we observe that the DAFormer network is able to segment some of the classes at the beginning of the training but forgets them after a few hundred training steps as we will show in Sec. 4.5. Therefore, we assume that the useful features from ImageNet pretraining are corrupted by $\mathcal{L}_S$ and the model overfits to the synthetic source data.

In order to prevent this issue, we regularize the model based on the Feature Distance (FD) of the bottleneck features F_θ of the semantic segmentation UDA model g_θ and the bottleneck features $F_{ImageNet}$ of the ImageNet model

$$d^{(i,j)} = ||F_{ImageNet}(x_S^{(i)})^{(j)} - F_\theta(x_S^{(i)})^{(j)}||_2 \,. \quad (8)$$

However, the ImageNet model is mostly trained on thing-classes (objects with a well-defined shape such as car or zebra) instead of stuff-classes (amorphous background regions such as road or sky) [4]. Therefore, we calculate the FD loss only for image regions containing thing-classes $\mathcal{C}_{things}$ described by the binary mask M_{things}

$$\mathcal{L}_{FD}^{(i)} = \frac{\sum_{j=1}^{H_F \times W_F} d^{(i,j)} \cdot M_{things}^{(i,j)}}{\sum_j M_{things}^{(i,j)}} \,, \quad (9)$$

This mask is obtained from the downscaled label $y_{S,small}$

$$M_{things}^{(i,j)} = \sum_{c'=1}^{C} y_{S,small}^{i,j,c'} \cdot [c' \in \mathcal{C}_{things}] \,. \quad (10)$$

To downsample the label to the bottleneck feature size, average pooling with a patch size $\frac{H}{H_F} \times \frac{W}{W_F}$ is applied to each class channel and a class is kept when it exceeds the ratio r

$$y_{S,small}^{c} = [\text{AvgPool}(y_S^c, H/H_F, W/W_F) > r] \,. \quad (11)$$

This ensures that only bottleneck feature pixels containing a dominant thing-class are considered for the feature distance.

The overall UDA loss $\mathcal{L}$ is the weighted sum of the presented loss components $\mathcal{L} = \mathcal{L}_S + \mathcal{L}_T + \lambda_{FD}\mathcal{L}_{FD}$.

Learning Rate Warmup for UDA Linearly warming up the learning rate [22] at the beginning of the training has successfully been used to train both CNNs [24] and Transformers [14, 71] as it improves network generalization [22] by avoiding that a large adaptive learning rate variance distorts the gradient distribution at the beginning of the training [44]. We newly introduce learning rate warmup to UDA. We posit that this is particularly important for UDA as distorting the features from ImageNet pretraining would deprive the network of useful guidance toward the real domain. During the warmup period up to iteration t_{warm}, the learning rate at iteration t is set $\eta_t = \eta_{base} \cdot t/t_{warm}$.

4. Experiments

4.1. Implementation Details

Datasets For the target domain, we use the Cityscapes street scene dataset [12] containing 2975 training and 500 validation images with resolution 2048×1024. For the source domain, we use either the GTA dataset [55], which contains 24,966 synthetic images with resolution 1914×1052, or the Synthia dataset [57], which consists of 9,400 synthetic images with resolution 1280×760. As a common practice in UDA [68], we resize the images to 1024×512 pixels for Cityscapes and to 1280×720 pixels for GTA.

Network Architecture Our implementation is based on the mmsegmentation framework [11]. For the DAFormer architecture, we use the MiT-B5 encoder [83], which produces a feature pyramid with $C = [64, 128, 320, 512]$. The DAFormer decoder uses $C_e = 256$ and dilation rates of 1, 6, 12, and 18. All encoders are pretrained on ImageNet-1k.

Training In accordance with [45, 83], we train DAFormer with AdamW [47], a learning rate of η_{base}=6×10^{-5} for the encoder and 6×10^{-4} for the decoder, a weight decay of 0.01, linear learning rate warmup with t_{warm}=1.5k, and linear decay afterwards. It is trained on a batch of two 512×512 random crops for 40k iterations. Following DACS [67], we use the same data augmentation parameters and set α=0.99 and τ=0.968. The RCS temperature is set T=0.01 to maximize the sampled pixels of the class with the least pixels. For FD, r=0.75 and λ_{FD}=0.005 to induce a similar gradient magnitude into the encoder as $\mathcal{L}_S$.

4.2. Comparison of Network Architectures for UDA

First, we compare several semantic segmentation architectures with respect to their UDA performance (see Sec. 3.1) on GTA→Cityscapes in Tab. 1. Additionally, we also provide the performance of the networks trained only with augmented source data (domain generalization) as well as the oracle performance trained with target labels (supervised learning). In all cases, the model is evaluated on the Cityscapes validation set and the performance is provided as mIoU in %. To compare how well a network is suited for

Table 1. Comparison of the mIoU (%) on the Cityscapes val. set of different segmentation architectures for source-only (GTA), UDA (GTA→Cityscapes), and oracle (Cityscapes) training. Additionally, the relative UDA performance (Rel.) wrt. the oracle mIoU is provided. Mean and SD are calculated over 3 random seeds.

Architecture	Src-Only	UDA	Oracle	Rel.
DeepLabV2 [6]	34.3 ±2.2	54.2 ±1.7	72.1 ±0.5	75.2%
DA Net [17]	30.9 ±2.1	53.7 ±0.2	72.6 ±0.2	74.0%
ISA Net [34]	32.3 ±2.1	53.3 ±0.4	72.0 ±0.5	74.0%
DeepLabV3+ [7]	31.0 ±1.4	53.7 ±1.0	75.6 ±0.9	71.0%
SegFormer [83]	**45.6** ±0.6	**58.2** ±0.9	**76.4** ±0.2	**76.2%**

Table 2. Ablation of the SegFormer encoder and decoder.

Encoder	Decoder	UDA	Oracle	Rel.
MiT-B5 [83]	SegF. [83]	58.2 ±0.9	76.4 ±0.2	76.2%
MiT-B5 [83]	DLv3+ [7]	56.8 ±1.8	75.5 ±0.5	75.2%
R101 [24]	SegF. [83]	50.9 ±1.1	71.3 ±1.3	71.4%
R101 [24]	DLv3+ [7]	53.7 ±1.0	75.6 ±0.9	71.0%

UDA, we further provide the relative performance (Rel.), which normalizes the UDA mIoU by the oracle mIoU. Note that the oracle mIoU is generally lower than reported in the literature on supervised learning as for UDA the images of Cityscapes are downsampled by a factor of two, which is a necessary common practice in UDA to fit images from both domains and additional networks into the GPU memory.

The majority of works on UDA use DeepLabV2 [6] with ResNet-101 [24] backbone. Interestingly, a higher oracle performance does not necessarily increase the UDA performance as can be seen for DeepLabV3+ [7] in Tab. 1. Generally, the studied more recent CNN architectures, do not provide a UDA performance gain over DeepLabV2. However, we identified the Transformer-based SegFormer [83] as a powerful architecture for UDA. It increases the mIoU for source-only / UDA / oracle training significantly from 34.3 / 54.2 / 72.1 to 45.6 / 58.2 / 76.4. We believe that especially the better domain generalization (source-only training) of SegFormer is valuable for the improved UDA performance.

To get a better insight into why SegFormer works well for UDA, we swap its encoder and decoder with ResNet101 and DeepLabV3+. As the MiT encoder of SegFormer has an output stride of 32 but the DeepLabV3+ decoder is designed for an output stride of 8, we bilinearly upsample the SegFormer bottleneck features by $\times 4$ when combined with the DeepLabv3+ decoder. Tab. 2 shows that the lightweight MLP decoder of SegFormer has a slightly higher relative UDA performance (Rel.) than the heavier DLv3+ decoder (76.2% vs 75.2%). However, the crucial contribution to good UDA performance comes from the Transformer MiT encoder. Replacing it with the ResNet101 encoder leads to a significant performance drop of the UDA performance. Even though the oracle performance drops as well due to the smaller receptive field of the ResNet encoder [83], the drop for UDA is over-proportional as shown by the relative performance decreasing from 76.2% to 71.4%.

Table 3. Influence of the encoder on UDA performance.

Enc.	Dec.	Src-Only	UDA	Oracle	Rel.
R50 [24]	DLv2 [6]	29.3	52.1	70.8	73.6%
R101 [24]	DLv2 [6]	36.9	53.3	72.5	73.5%
S50 [87]	DLv2 [6]	27.9	48.0	67.7	70.9%
S101 [87]	DLv2 [6]	35.5	53.5	72.2	74.1%
S200 [87]	DLv2 [6]	35.9	56.9	73.5	77.4%
MiT-B3 [83]	SegF. [83]	42.2	50.8	76.5	66.4%
MiT-B4 [83]	SegF. [83]	44.7	57.5	77.1	74.6%
MiT-B5 [83]	SegF. [83]	46.2	58.8	76.2	77.2%

Table 4. Influence of learning rate warmup on UDA performance.

Architecture	LR Warmup	UDA	Oracle	Rel.
DeepLabV2 [6]	–	49.1 ±2.0	67.4 ±1.7	72.8%
DeepLabV2 [6]	✓	54.2 ±1.7	72.1 ±0.5	75.2%
SegFormer [83]	–	51.8 ±0.8	72.9 ±1.6	71.1%
SegFormer [83]	✓	58.2 ±0.9	76.4 ±0.2	76.2%

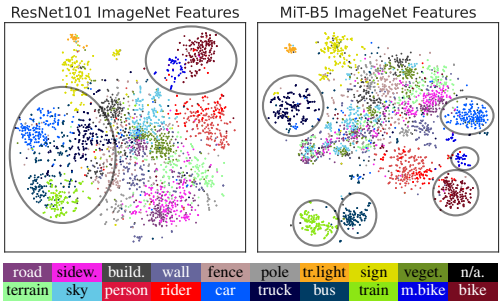

Figure 3. T-SNE [70] embedding of the bottleneck features after ImageNet pre-training for ResNet101 [24] and MiT-B5 [83] on the Cityscapes val. set, showing a better vehicle separability for MiT.

Therefore, we further investigate the influence of the encoder architecture on UDA performance. In Tab. 3, we compare different encoder designs and sizes. It can be seen that deeper models achieve a better source-only and relative performance demonstrating that deeper models generalize/adapt better to the new domain. This observation is in line with findings on the robustness of network architectures [3]. Compared to CNN encoders, the MiT encoders generalize better from source-only training to the target domain. Overall, the best UDA mIoU is achieved by the MiT-B5 encoder. To gain insights on the improved generalization, Fig. 3 visualizes the ImageNet features of the target domain. Even though ResNet structures stuff-classes slightly better, MiT shines at separating semantically similar classes (e.g. all vehicle classes), which are usually particularly difficult to adapt. A possible explanation might be the texture-bias of CNNs and the shape-bias of Transform-

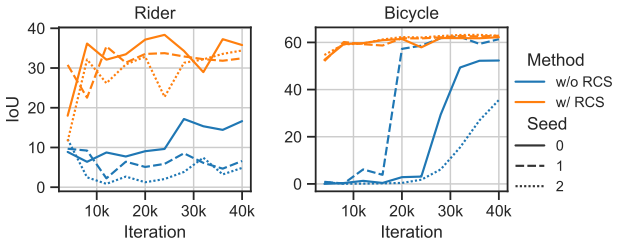

Figure 4. SegFormer UDA performance for the rare classes rider and bicycle without and with Rare Class Sampling (RCS).

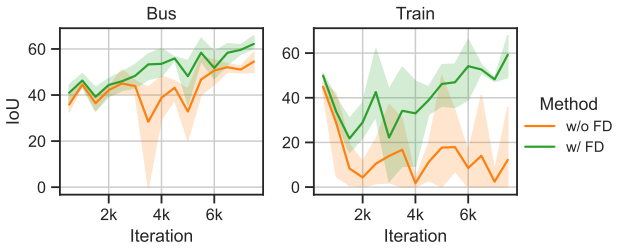

Figure 5. SegFormer UDA performance in the beginning of the training with and without ImageNet Feature Distance (FD).

Table 5. Ablation of the components of the UDA framework.

	Network	Warmup	RCS	FD	Misc.	UDA
1	SegF. [83]	–	–	–	–	51.8 ±0.8
2	SegF. [83]	✓	–	–	–	58.2 ±0.9
3	SegF. [83]	✓	✓(T=∞)	–	–	62.0 ±1.5
4	SegF. [83]	✓	✓	–	–	64.0 ±2.4
5	SegF. [83]	✓	–	✓(all $\mathcal{C}$)	–	58.8 ±0.4
6	SegF. [83]	✓	–	✓	–	61.7 ±2.6
7	SegF. [83]	✓	✓	✓	–	66.2 ±1.0
8	SegF. [83]	✓	✓	✓	Crop PL, $\alpha\uparrow$	67.0 ±0.4
9	DLv2 [6]	–	–	–	–	49.1 ±2.0
10	DLv2 [6]	✓	✓	✓	Crop PL, $\alpha\uparrow$	56.0 ±0.5

ers [3]. Before we study the context-aware fusion decoder of DAFormer in Sec. 4.6, we will first discuss how to stabilize the training of MiT with the default SegFormer decoder.

4.3. Learning Rate Warmup

Tab. 4 shows that learning rate warmup significantly improves both UDA and oracle performance. UDA benefits even more than supervised learning from warmup (see column Rel.), showing its particular importance for UDA by stabilizing the beginning of the training, which improves difficult classes (cf. row 1 and 2 in Fig 6). As warmup is essential for a good UDA performance across different architectures, it was already applied in the previous section.

4.4. Rare Class Sampling (RCS)

When training SegFormer for UDA, we observe that the performance of some classes depends on the random seed for data sampling as can be seen for the blue IoU curves

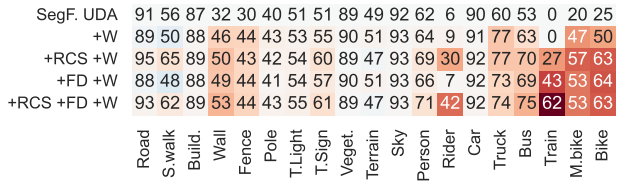

Figure 6. Comparison of the class-wise IoU of Warmup (W), RCS and FD. The color visualizes the IoU difference to the baseline.

in Fig. 4. The affected classes are underrepresented in the source dataset as shown in the supplement. Interestingly, the IoU for the class bicycle starts increasing at different iterations for different seeds. We hypothesize that this is caused by the sampling order, in particular when the relevant rare classes are sampled. Further, the later the IoU starts improving, the worse is the final IoU of this class, probably due to the confirmation bias of self-training that was accumulated over earlier iterations. Therefore, for UDA, it is especially important to learn rare classes early.

In order to address this issue, the proposed RCS increases the sampling probability of rare classes. Fig. 4 (orange) shows that RCS results in an earlier increase of the IoU of rider/bicycle and a higher final IoU independent of the data sampling random seed. This confirms our hypothesis that an (early) sampling of rare classes is important for learning these classes properly. RCS improves the UDA performance by +5.8 mIoU (cf. row 2 and 4 in Tab. 5). The highest IoU increase is observed for the rare classes rider, train, motorcycle, and bicycle (cf. row 2 and 3 in Fig. 6). RCS also outperforms its special case $T = \infty$, which corresponds to 'class-balanced sampling' (cf. row 3 and 4 in Tab. 5), as class-balanced sampling does not consider the co-occurrence of multiple classes in semantic segmentation.

4.5. Thing-Class ImageNet Feature Distance (FD)

While RCS gives a performance boost, the performance for thing-classes (e.g. bus and train) could still be further improved as some of the object classes that are fairly well separated in ImageNet features (see Fig. 3 right) are mixed together after the UDA training. When investigating the IoU during the early training (see Fig. 5 orange), we observe an early performance drop for the class train. We assume that the powerful MiT encoder overfits to the synthetic domain. When regularizing the training with the proposed FD, the performance drop is avoided (see Fig. 5 green). Also other difficult classes such as bus, motorcycle, and bicycle benefit from the regularization (cf. row 2 and 4 in Fig. 6). Overall the UDA performance is improved by +3.5 mIoU (cf. row 2 and 6 in Tab. 5). Note that applying FD only to thing-classes, which the ImageNet features were trained on, is important for its good performance (cf. row 5 and 6).

When combining RCS and FD, we observe a further im-

Table 6. Comparison with state-of-the-art methods for UDA. The results for DAFormer are averaged over 3 random seeds.

	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
GTA5 → Cityscapes																				
CBST [92]	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
DACS [67]	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
CorDA [75]	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47.0	89.7	66.7	35.9	90.2	48.9	57.5	0.0	39.8	56.0	56.6
ProDA [88]	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	**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**
Synthia → Cityscapes																				
CBST [92]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	–	78.3	60.6	28.3	81.6	–	23.5	–	18.8	39.8	42.6
DACS [67]	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
CorDA [75]	**93.3**	**61.6**	85.3	19.6	5.1	37.8	36.6	42.8	84.9	–	90.4	69.7	41.8	85.6	–	38.4	–	32.6	53.9	55.0
ProDA [88]	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	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**

Table 7. Comparison of decoder architectures with MiT encoder and UDA improvements (DSC: depthwise separable convolution).

Decoder	C_e	#Params	UDA	Oracle	Rel.
SegF. [83]	768	3.2M	67.0 ±0.4	76.8 ±0.3	87.2%
SegF. [83]	256	0.5M	67.1 ±1.1	76.5 ±0.4	87.7%
UperNet [82]	512	29.6M	67.4 ±1.1	**78.0** ±0.2	86.4%
UperNet [82]	256	8.3M	66.7 ±1.2	77.4 ±0.3	86.2%
ISA [34] Fusion	256	1.1M	66.3 ±0.9	76.3 ±0.4	86.9%
Context only at F_4	256	3.2M	67.0 ±0.6	76.6 ±0.2	87.5%
DAFormer w/o DSC	256	10.0M	67.0 ±1.5	76.7 ±0.6	87.4%
DAFormer	256	3.7M	**68.3** ±0.5	77.6 ±0.2	**88.0%**

provement to 66.2 mIoU showing that they complement each other (see row 7 in Tab. 5). We notice pseudo-label drifts originating from image rectification artifacts and the ego car. As these regions are not part of the street scene segmentation task, we argue that it is not meaningful to produce pseudo-labels for them. Therefore, we ignore the top 15 and bottom 120 pixels of the pseudo-label. As Transformers are more expressive, we further increase α to 0.999 to introduce a stronger regularization from the teacher. This mitigates the pseudo-label drifts and improves the UDA mIoU (cf. row 7 and 8 in Tab. 5). The overall improvement is +15.2 mIoU for SegFormer (cf. row 1 and 8) and +6.9 mIoU for DeepLabV2 (cf. row 9 and 10). When comparing both architectures, it can be seen that SegFormer benefits noticeably more, supporting our initial hypothesis that the architecture choice can limit the effectiveness of UDA methods.

4.6. DAFormer Decoder

After regularizing and stabilizing the UDA training for a MiT encoder and a SegFormer decoder, we come back to the network architecture and investigate our DAFormer decoder with the context-aware feature fusion. Tab. 7 shows that it improves the UDA performance over the SegFormer decoder from 67.0 to 68.3 mIoU (cf. row 1 and 8). Further, DAFormer outperforms a variant without depthwise separable convolutions (cf. last two rows) and a variant with ISA [34] instead of ASPP for feature fusion (cf. row 5 and 8). This shows that a capable but parameter-effective decoder with an inductive bias of the dilated depthwise separable convolutions is beneficial for good UDA performance. When the context is only considered for bottleneck features, the UDA performance decreases by -1.3 mIoU (cf. row 6 and 8), revealing that the context clues from different encoder stages used in DAFormer are more domain-robust. We further compare DAFormer to UperNet [82], which iteratively upsamples and fuses the features and was used together with Transformers in [45]. Even though UperNet achieves the best oracle performance, it is noticeably outperformed by DAFormer on UDA, which confirms that it is necessary to study and design the decoder architecture, along with the encoder architecture, specifically for UDA.

Tab. 6 shows that DAFormer outperforms previous methods by a significant margin. On GTA→Cityscapes, it improves the performance from 57.5 to 68.3 mIoU and on Synthia→Cityscapes from 55.5 to 60.9 mIoU. In particular, DAFormer learns even difficult classes well, which previous methods struggled with such as train, bus, and truck.

Further details are provided in the supplement, including RCS statistics, parameter sensitivity of RCS/FD, ablation of ST, a runtime and GPU memory benchmark, a comprehensive qualitative analysis, and a discussion of limitations.

5. Conclusions

We presented DAFormer, a network architecture tailored for UDA, which is based on a Transformer encoder and a context-aware fusion decoder, revealing the potential of Transformers for UDA. Additionally, we introduced three training policies to stabilize and regularize UDA, further enabling the capabilities of DAFormer. Overall, DAFormer represents a major advance in UDA and improves the SOTA performance by 10.8 mIoU on GTA→Cityscapes and 5.4 mIoU on Synthia→Cityscapes. We would like to highlight the value of DAFormer by superseding DeepLabV2 to evaluate UDA methods on a much higher performance level.

References

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15384–15394, 2021. 2, 3

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017. 2

[3] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Int. Conf. Comput. Vis.*, pages 10231–10241, 2021. 2, 3, 6, 7

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1209–1218, 2018. 3, 5

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *Int. Conf. Learn. Represent.*, pages 834–848, 2015. 2

[6] 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 Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017. 1, 2, 3, 4, 6, 7

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis.*, pages 801–818, 2018. 2, 4, 6

[8] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7892–7901, 2018. 2, 3

[9] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 6830–6840, 2019. 2

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1251–1258, 2017. 4

[11] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 5

[12] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3213–3223, 2016. Dataset license: https://www.cityscapes-dataset.com/license/. 1, 5

[13] Dengxin Dai, Christos Sakaridis, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. *Int. J. Comput. Vis.*, 2019. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2020. 2, 3, 5

[15] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 982–991, 2019. 2

[16] Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *Brit. Mach. Vis. Conf.*, 2020. 3

[17] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154, 2019. 2, 6

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Research*, 17(1):2096–2030, 2016. 2

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *Int. Conf. Learn. Represent.*, 2018. 2

[20] Rui Gong, Wen Li, Yuhua Chen, Dengxin Dai, and Luc Van Gool. Dlow: Domain flow and applications. *Int. J. Comput. Vis.*, 129(10):2865–2888, 2021. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, pages 2672–2680, 2014. 2

[22] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2, 5

[23] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks*, pages 1322–1328, 2008. 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1, 5, 6

[25] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *Int. Conf. Comput. Vis.*, pages 6930–6940, 2021. 3

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Int. Conf. Comput. Vis.*, pages 8340–8349, 2021. 2

[27] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Int. Conf. Learn. Represent.*, 2019. 2

[28] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[29] 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 *Int. Conf. Mach. Learn.*, pages 1989–1998, 2018. 2

[30] 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. 2, 3

[31] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Köring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11130–11140, 2021. 3

[32] Lukas Hoyer, Dengxin Dai, Qin Wang, Yuhua Chen, and Luc Van Gool. Improving semi-supervised and domain-adaptive semantic segmentation with self-supervised depth estimation. *arXiv preprint arXiv:2108.12545*, 2021. 2

[33] Lukas Hoyer, Mauricio Munoz, Prateek Katiyar, Anna Khoreva, and Volker Fischer. Grid saliency for context explanations of semantic segmentation. In *Adv. Neural Inform. Process. Syst.*, pages 6462–6473, 2019. 2

[34] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv preprint arXiv:1907.12273*, 2019. 2, 6, 8

[35] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models with respect to common corruptions. *Int. J. Comput. Vis.*, 129(2):462–483, 2021. 2, 4

[36] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13896–13905, 2020. 3

[37] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12975–12984, 2020. 2

[38] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1, 2

[39] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Int. Conf. Mach. Learn. Worksh.*, 2013. 2

[40] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6936–6945, 2019. 2

[41] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2017. 3

[42] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Int. Conf. Comput. Vis.*, pages 2980–2988, 2017. 3

[43] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2970–2979, 2020. 3

[44] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Int. Conf. Learn. Represent.*, 2019. 5

[45] 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 *Int. Conf. Comput. Vis.*, pages 10012–1110022, 2021. 2, 5, 8

[46] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3431–3440, 2015. 1, 2

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2018. 5

[48] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2507–2516, 2019. 2

[49] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *Eur. Conf. Comput. Vis.*, pages 415–430, 2020. 3

[50] Luke Melas-Kyriazi and Arjun K. Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12435–12445, 2021. 2

[51] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing Properties of Vision Transformers. In *Adv. Neural Inform. Process. Syst.*, 2021. 2, 3

[52] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *IEEE Winter Conf. on Applications of Comput. Vis.*, pages 1369–1378, 2021. 3

[53] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021. 2, 3

[54] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 8558–8567, 2021. 3

[55] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Eur. Conf. Comput. Vis.*, pages 102–118, 2016. 1, 5

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Medical Image Computing and Computer-assisted Intervention*, pages 234–241, 2015. 2

[57] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3234–3243, 2016. Dataset license: CC BY-NC-SA 3.0. 1, 5

[58] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Int. Conf. Comput. Vis.*, pages 10765–10775, 2021. 1

[59] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 761–769, 2016. 3

[60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[61] 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. In *Adv. Neural Inform. Process. Syst.*, 2020. 2, 3

[62] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019. 2

[63] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[64] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020. 1

[65] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Adv. Neural Inform. Process. Syst.*, pages 1195–1204, 2017. 2, 3

[66] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Int. Conf. Mach. Learn.*, pages 10347–10357, 2021. 2, 3

[67] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain Adaptation via Cross-domain Mixed Sampling. In *IEEE Winter Conf. on Applications of Comput. Vis.*, pages 1379–1389, 2021. 2, 3, 5, 8

[68] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7472–7481, 2018. 1, 2, 3, 5

[69] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *Int. Conf. Comput. Vis.*, pages 1456–1465, 2019. 2

[70] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Research*, 9(11), 2008. 6

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 2, 3, 5

[72] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2517–2526, 2019. 2

[73] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 7364–7373, 2019. 2

[74] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *Eur. Conf. Comput. Vis.*, pages 642–659, 2020. 2, 3

[75] Qin Wang, Dengxin Dai, Lukas Hoyer, Olga Fink, and Luc Van Gool. Domain adaptive semantic segmentation with self-supervised depth estimation. In *Int. Conf. Comput. Vis.*, pages 8515–8525, 2021. 2, 8

[76] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[77] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Int. Conf. Comput. Vis.*, pages 568–578, 2021. 4

[78] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7794–7803, 2018. 2

[79] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Adv. Neural Inform. Process. Syst.*, pages 7032–7042, 2017. 2

[80] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12635–12644, 2020. 2

[81] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10857–10866, 2021. 3

[82] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Eur. Conf. Comput. Vis.*, pages 418–434, 2018. 8

[83] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Adv. Neural Inform. Process. Syst.*, 2021. 2, 3, 4, 5, 6, 7, 8

[84] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4085–4095, 2020. 2, 3

[85] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2, 4

[86] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Eur. Conf. Comput. Vis.*, pages 173–190, 2020. 1, 2

[87] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 6

[88] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12414–12424, 2021. 2, 3, 8

[89] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2881–2890, 2017. 2

[90] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6881–6890, 2021. 2, 4

[91] 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. In *IEEE Winter Conf. on Applications of Comput. Vis.*, pages 514–524, 2021. 2, 3

[92] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Eur. Conf. Comput. Vis.*, pages 289–305, 2018. 2, 3, 8

[93] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Int. Conf. Comput. Vis.*, pages 5982–5991, 2019. 2, 3