Domain-Adaptive Semantic Segmentation with Memory-Efficient Cross-Domain Transformers

Ruben Mascaro
rmascaro@ethz.ch
Lucas Teixeira
lteixeira@mavt.ethz.ch
Margarita Chli
chlim@ethz.ch

Vision for Robotics Lab
ETH Zurich, Switzerland
University of Cyprus, Cyprus

Abstract

Unsupervised Domain Adaptation (UDA), a process by which a model trained on a well-annotated source dataset is adapted to an unlabeled target dataset, has emerged as a promising solution for deploying semantic segmentation models in scenarios where annotating extensive amounts of data is cost-prohibitive. Although the recent development of UDA strategies exploiting Transformer-based architectures has represented a major advance in the field, current approaches struggle to effectively learn context dependencies in the target domain, leading to suboptimal semantic label predictions. Aiming at addressing this issue, in this work we introduce a generic three-branch Transformer block that combines self- and cross-attention mechanisms for better source and target feature alignment. We then show how the proposed architecture can be seamlessly incorporated into state-of-the-art self-training UDA schemes for semantic segmentation, yielding enhanced adaptation capabilities without increasing the GPU memory footprint during training. The resulting framework significantly outperforms its baseline on benchmarking datasets for synthetic-to-real (+1.4 mIoU on GTA→Cityscapes and +1.1 mIoU on SYNTHIA→Cityscapes) and clear-to-adverse-weather (+3.4 mIoU on Cityscapes→ACDC) UDA. In addition, it achieves superior robustness compared to using existing cross-domain Transformer architectures that require substantially more GPU memory for training.

Code – https://github.com/VIS4ROB-lab/MemCDT

1 Introduction

Semantic segmentation is a fundamental task in computer vision that aims at assigning a class label to each pixel in an image. By doing so, it provides a detailed understanding of the image content, which is often key to enabling high-level reasoning in various downstream applications, such as autonomous driving and robotics. Despite the outstanding progress achieved in the field through the application of deep learning techniques [4, 13, 27], current methods still rely on the availability of abundant labeled data for training and are highly sensitive to domain shifts. This poses a substantial challenge to the deployment of semantic segmentation models in a wide range of applications and environments, as collecting large-scale, pixel-level annotated datasets reproducing the conditions expected during deployment is an extremely labor-intensive and time-consuming process.

A promising approach to overcome this problem involves the transfer of knowledge acquired from well-annotated source data to unlabeled target data, which is commonly referred to as Unsupervised Domain Adaptation (UDA) in the literature. In the context of semantic segmentation, UDA methods have witnessed remarkable performance improvements over the past few years, especially since the incorporation of strategies that leverage modern Transformer-based architectures [10]. Built upon the self-attention mechanism [23], these models have demonstrated greater success in modeling context relationships compared to traditional Convolutional Neural Networks (CNNs), leading to increased generalization capabilities. Nonetheless, a considerable performance gap still exists between UDA and supervised training. This stems from the fact that, while the learning of context dependencies can be guided by ground truth in supervised learning, the unsupervised losses in UDA typically lack the power to facilitate effective learning of such information in the target domain. As a result, the adapted models tend to focus on irrelevant regions and struggle to provide adequate support for accurate semantic label prediction [25].

In this work, we address this issue by explicitly leveraging cross-domain context relationships during training. Drawing inspiration from recent work exploiting cross-domain attention for UDA in image classification [28], we rethink how such a mechanism can be introduced within Transformer backbones for semantic segmentation to promote the learning of robust, domain-invariant features. The resulting architecture can be effortlessly integrated into current state-of-the-art self-training UDA pipelines, yielding a simple, yet effective scheme that exhibits increased generalization capabilities across diverse synthetic-to-real and clear-to-adverse-weather UDA benchmarks. Furthermore, the proposed approach does not increase the GPU memory footprint of the underlying UDA framework, offering a more accessible solution compared to existing cross-domain Transformer designs [25, 28] that require higher-end hardware for training.

2 Related Work

UDA for Semantic Segmentation. UDA methods are typically categorized into adversarial and self-training approaches. The former exploit learned domain discriminators to align the source and target domain distributions at input [8, 9], feature [22] or output level [22, 24]. The latter, on the contrary, train the network on the target domain using pseudo-labels. These can be either generated offline, requiring multiple training stages [29, 33], or predicted online during training, in which case pseudo-label prototypes [30] or consistency regularization techniques [1, 15, 21] are often used to promote stability of training. Most UDA methods have evaluated their contributions using CNN-based architectures so far, with self-training methods gradually outperforming the usually more unstable and computationally expensive adversarial approaches. Recently, with DAFormer [10] and HRDA [11], the Transformer architecture and additional strategies for self-training were introduced to the task of UDA in semantic segmentation, greatly improving performance over previous CNN-based methods. Building upon the former as an example of a modern, simple, and widely applicable UDA pipeline that can be deployed on consumer-grade GPUs (as opposed to the latter, which is specifically tailored to high-resolution input), here we aim at further enhancing source-target feature alignment through an additional network component that leverages the Transformer's attention mechanisms for mixing context-aware features across domains.

Vision Transformers. Motivated by their initial success in Natural Language Processing, Transformer-based architectures [23] have recently been adapted to computer vision tasks such as image classification [7, 12], object detection [3, 32], and semantic segmentation [27, 31], yielding state-of-the-art performance while exhibiting greater robustness than CNNs against distribution shifts [2, 27]. The key component of these novel architectures is the so-called self-attention module, which enables the integration of both local and global context information in the computed deep features as opposed to the convolution operation in CNNs, which only captures local information. Originating from self-attention, cross-attention has been deployed within Transformer architectures mainly for feature fusion in multi-modal tasks (e.g. language-vision [5, 26]). However, in the context of image classification, recent work has demonstrated that such mechanisms can also be used to enhance source-target feature alignment within Transformer-based UDA frameworks [28]. Inspired by this idea, we study effective ways to incorporate cross-domain attention within Transformer-based backbones for semantic segmentation and design a UDA training strategy that leverages the properties of these newly introduced modules to promote knowledge transfer. Compared to concurrent work that applies the architecture introduced in [28] to domain-adaptive semantic segmentation, our proposed approach achieves superior performance while being substantially more memory efficient.

3 Method

3.1 Preliminaries

In UDA, we are given a set of images from a source domain $\mathcal{X}_S = \{x_S^{(i)}\}_{i=1}^{N_S}$, together with its corresponding set of ground-truth, pixel-wise semantic annotations in the form of one-hot labels $\mathcal{Y}_S = \{y_S^{(i)}\}_{i=1}^{N_S}$, and a set of unlabeled images from the target domain $\mathcal{X}_T = \{x_T^{(i)}\}_{i=1}^{N_T}$. We assume that, while the images in $\mathcal{X}_S$ and $\mathcal{X}_T$ are sampled from different distributions, both domains share a common label space $\mathcal{C}$. The goal is then to train a neural network f parametrized by θ, i.e. f_θ, so that it delivers reliable performance on previously unseen images originating from the target domain.

Following DAFormer [10], we start by setting up a baseline UDA pipeline that adopts the online self-training paradigm, where the mean-teacher framework is used. This is comprised of a teacher network h_ϕ and a student network f_θ, both sharing the same architecture. The student model is used to backpropagate gradients and update weights based on the training loss, while the teacher model is used to produce pseudo-labels for the target images:

$$\hat{y}_T^{(h,w,c)} = \left[c = \operatorname*{argmax}_{c'} h_\phi\left(x_T\right)^{(h,w,c')}\right], \tag{1}$$

where $[\cdot]$ denotes the Iverson bracket. During training, the weights of the teacher model are updated based on the Exponential Moving Average (EMA) of the student's weights after each iteration t, i.e. $\phi_{t+1} \leftarrow \alpha\phi_t + (1-\alpha)\,\theta_t$.

To train the student network, a supervised branch for the source domain and an unsupervised branch for the target domain are employed. The supervised branch is trained with a categorical cross-entropy loss using the available ground-truth labels for the source domain images:

$$\mathcal{L}_S = -\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C} y_S^{(h,w,c)} \log p_S^{(h,w,c)}, \tag{2}$$

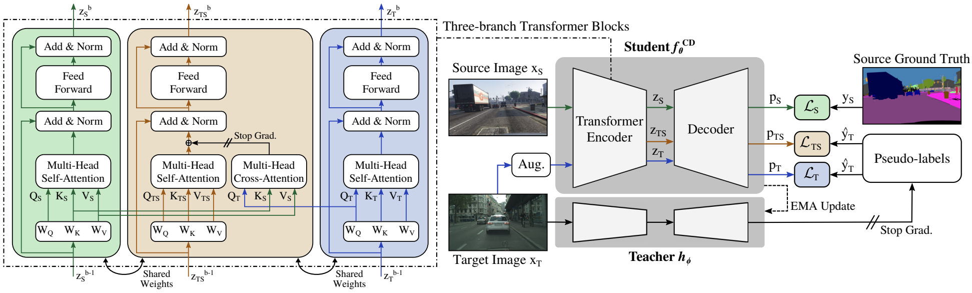


Figure 1: Overview of the proposed UDA framework for semantic segmentation with memory-efficient cross-domain Transformers. The three-branch, weight-sharing Transformer block architecture introduced in Sec. 3.2 is illustrated on the left, while the adapted UDA self-training strategy described in Sec. 3.3 is shown on the right.

while the pseudo-labels are used to train the unsupervised branch on the target domain:

$$\mathcal{L}_T = -\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C} q_T \hat{y}_T^{(h,w,c)} \log p_T^{(h,w,c)} \ . \tag{3}$$

Here, p_S and p_T are the pixel-wise softmax class probabilities predicted by the source and the target branches, respectively, i.e. $p_S = f_\theta(x_S)$ and $p_T = f_\theta(x_T)$, while q_T represents a confidence estimate for the pseudo-labels, which is determined by the proportion of pixels whose maximum softmax probability surpasses a threshold τ [10, 21]:

$$q_T = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\left[\max_{c'} h_\phi(x_T)^{(h,w,c')} > \tau\right] . \tag{4}$$

In practice, to boost the efficiency of training, the student network is trained on augmented target data, while the teacher network generates pseudo-labels using non-augmented images. As in DACS [21] and DAFormer [10], we use color jitter, Gaussian blur, and ClassMix [16] for data-augmentation-based consistency regularization. Additional training strategies introduced in DAFormer, such as Rare Class Sampling and ImageNet Feature Distance, are also adopted in our framework.

3.2 Memory-Efficient Cross-Domain Transformer Backbone

Given the baseline UDA pipeline described in the previous section, we study how the intrinsic properties of Transformer-based architectures can be leveraged to further enhance the learning of domain-robust features. Transformer-based models build upon the so-called self-attention mechanism [23], which produces a representation of an input sequence based on dynamically computed relationships among all the elements in it. Formally, the self-attention block takes as input a sequence of N flattened image patches or feature embeddings and projects them into three vectors, namely queries $Q \in \mathbb{R}^{N\times d_k}$, keys $K \in \mathbb{R}^{N\times d_k}$ and values $V \in \mathbb{R}^{N\times d_v}$, with d_k and d_v indicating their dimensions. The output is computed as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \ . \tag{5}$$

One of the key advantages of the self-attention mechanism is that it allows the network to integrate both local and global context relationships into their features. However, while the learning of such information can be effectively guided by ground truth in supervised learning, the lack of ground-truth supervision for the target domain in UDA typically causes self-attention to be noisy and focus on less informative regions when applied to target images. To mitigate this issue, we design a novel backbone architecture that helps the model bridge distributional shifts in attention across domains and makes it more robust in the presence of noisy attention maps.

Inspired by recent work introducing cross-domain Transformers for UDA in image classification [28], we aim at extending Transformer-based semantic segmentation backbones with cross-domain attention mechanisms in order to capture common context relationships among images from different domains. Contrary to self-attention, cross-domain attention takes as input query vectors from an image of one domain and key, value vectors from an image of the other domain. Namely, following Eq. 5, we compute target-to-source attention as $Attn_{TS} = \text{Attention}\left(Q_T, K_S, V_S\right)$. By leveraging this operation to transfer relevant context relationships from the source to the target images, our method effectively softens the boundary between the two domains.

Fig. 1 shows how we apply cross-domain attention within Transformer backbones for UDA in semantic segmentation. Specifically, we replace every self-attention block in the original architecture with the illustrated three-branch, weight-sharing attention mechanism. From left to right, we name these branches as *source* (S), *target-to-source* (TS) and *target* (T), according to the type of attention they are comprised of. In the b-th transformer block, the S and T branches apply self-attention to the input source and target embeddings, respectively, producing output source and target feature representations, i.e. z_S^b and z_T^b. Cross-attention mechanisms, on the other hand, are incorporated in the TS branch, which is designed to produce mixed feature representations for the target image, i.e. z_{TS}^b, by attending on both intra- and cross-domain similar patches. More precisely, the TS branch extracts initial features from the target images and produces output based on the combination of self and target-to-source attention. To fuse features extracted from the self- and cross-attention modules, we simply compute the average of both.

It is worth noting that, differently from related methods employing cross-domain Transformers in UDA [25, 28], our design introduces both self- and cross-attention modules within the TS branch. In our approach, cross-domain attention is used to generate a representation for the target image based on affine features retrieved from the source image. Fusing such a representation with the output of self-attention, the TS branch in the proposed architecture eventually produces a feature representation for the target image that encodes relevant contextual information extracted from both the source and the target domains. This enforces an effective transfer of contextual dependencies from the source to the target domain during training. Furthermore, as cross-domain attention might produce noisy signals, we use gradient stopping on the cross-attentive features. A key implication of this design is that it removes the need to backpropagate gradients through multiple branches simultaneously, resulting in a much more memory-efficient architecture compared to existing cross-domain Transformers in the literature.

3.3 UDA Self-Training with Cross-Domain Transformer Backbones

By replacing the self-attention blocks in a given Transformer architecture f_θ with the proposed three-branch attention modules, we obtain an augmented version of the original network, denoted as f_θ^{CD}. As shown in Fig. 1, during the forward pass, the three-branch Transformer backbone f_θ^{CD} takes as input a batch of source-target image pairs and produces three different sets of output features, i.e. one per branch, that we denote as z_S, z_{TS} and z_T. These are then fed to the decoder, resulting in three pixel-wise softmax segmentation maps p_S, p_{TS} and p_T. The learning of domain-invariant features is achieved in our approach by enforcing consistency between the semantic labels predicted by the the TS branch and the pseudo-labels predicted by the teacher network:

$$\mathcal{L}_{TS} = -\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C} q_T \hat{y}_T^{(h,w,c)} \log p_{TS}^{(h,w,c)} \ . \tag{6}$$

Combined with the standard losses used for the source S and target T branches in self-training UDA (i.e. Eq. 2 and Eq. 3), the total training loss is obtained as:

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_T + \mathcal{L}_{TS} \ . \tag{7}$$

3.4 Inference for the Target Domain

Since f_θ^{CD} keeps the same underlying architecture and number of parameters as the original model it inherits from, i.e. f_θ, the learned parameters θ during UDA training can be directly loaded into f_θ to perform inference on images from the target domain. Therefore, despite reducing training speed due to the additional calculations for the TS branch, the proposed strategy does not alter throughput nor the amount of resources required during inference.

4 Experiments

4.1 Datasets

We test our approach on synthetic-to-real and clear-to-adverse-weather UDA using benchmarking semantic segmentation datasets of street scenes. As synthetic datasets, we use GTA [17] and SYNTHIA [18]. The former contains 24,966 training images with resolution of 1914 × 1052 pixels, while the latter features 9,400 training images with resolution of 1280 × 760 pixels. As real-world datasets, we take Cityscapes [6], with 2,975 training and 500 validation images of resolution 2048 × 1024 pixels for clear weather, and ACDC [19], with 1,600 training, 406 validation and 2,000 test images of resolution 1920 × 1080 for adverse weather (i.e. fog, night, rain, and snow). Following standard practice in UDA [10], we resize GTA images to 1280 × 720 pixels, Cityscapes images to 1024 × 512 pixels, and ACDC images to 960 × 540 pixels.

4.2 Implementation and Training Details

The UDA framework developed in this work builds upon the DAFormer [10] training pipeline as noted in Sec. 3.1 and uses the same encoder-decoder architecture formed by the MiT-B5 [27] backbone and the DAFormer head [10]. For UDA training, we modify the original MiT-B5 forward pass to exploit cross-attention between the source and the target images as described in Sec. 3.2 and incorporate the additional consistency loss introduced in Sec. 3.3 to supervise the cross-domain branch. In all of our experiments, the MiT-B5 encoder is pre-trained on ImageNet-1k.

	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Tr.Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	*mIoU*
GTA → Cityscapes (Val.)																				
ADVENT [24]	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 [21]	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 [30]	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 [10]	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
CDTDA [25]	**96.5**	**73.9**	89.5	**56.8**	48.9	50.7	55.8	**63.3**	89.9	**49.1**	91.2	**72.2**	**45.4**	**92.7**	78.3	**82.9**	67.5	55.2	**63.4**	69.6
Ours	96.3	73.7	**89.9**	56.2	**49.7**	**52.0**	**56.8**	62.7	**90.0**	**49.1**	91.5	71.5	44.6	92.5	**79.4**	77.8	**71.6**	**56.8**	63.2	**69.7**
SYNTHIA → Cityscapes (Val.)																				
ADVENT [24]	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 [21]	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 [30]	**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 [10]	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
CDTDA [25]	83.7	42.9	87.4	39.8	7.5	**50.7**	55.7	53.5	85.9	–	**90.9**	**74.5**	47.2	86.0	–	60.2	–	**57.8**	60.8	61.5
Ours	86.0	44.9	**88.7**	**44.0**	**7.9**	50.3	**56.0**	54.0	85.6	–	88.4	73.8	46.2	87.7	–	**61.5**	–	55.8	60.3	**62.0**
Cityscapes → ACDC (Test)																				
ADVENT [24]	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
FDA [29]	73.2	34.7	59.0	24.8	29.5	28.6	43.3	44.9	70.1	28.2	54.7	47.0	28.5	74.6	44.8	52.3	63.3	28.3	39.5	45.7
MGCDA [20]	**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
DAFormer [10]	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
CDTDA [25]	57.6	43.7	**85.1**	43.5	33.9	50.1	42.9	53.9	72.8	52.9	52.2	59.4	34.7	**83.6**	60.4	**68.7**	**84.3**	41.4	**53.0**	56.5
Ours	69.0	**53.1**	84.7	**45.8**	**36.0**	50.1	43.2	**57.0**	73.4	**54.2**	65.9	**59.9**	**37.0**	83.0	**65.8**	62.3	83.9	42.3	51.5	**58.8**

Table 1: Comparison with the state of the art on different UDA benchmarks. Following common practice, our reported results are averaged over 3 random seeds.

As in DAFormer, we train on batches of two 512×512 random crops for 40k iterations, using the AdamW [14] optimizer with a learning rate of 6×10^{-5} for the encoder and 6×10^{-4} for the decoder, a weight decay of 0.01, linear learning rate warm-up up to iteration 1.5k, and linear decay afterward. Hyper-parameter values for specific UDA training strategies such as DACS [21] augmentations, Rare Class Sampling (RCS), and ImageNet feature distance (FD) are kept as in the original configurations.

4.3 Comparison with the State of the Art

The performance of the proposed framework is evaluated on different domain adaptation scenarios: synthetic-to-real (GTA→Cityscapes and SYNTHIA→Cityscapes) and clear-to-adverse weather conditions (Cityscapes→ACDC). Tab. 1 shows the segmentation accuracy obtained with our approach compared to other methods from the state of the art. It is worth noting that DAFormer itself brings an unprecedented performance boost with respect to previous CNN-based methods [21, 24, 30], highlighting the benefits of leveraging modern Transformer architectures in UDA. Compared to the original DAFormer UDA pipeline, our method achieves significant improvements on all three benchmarks, showing that the newly introduced cross-domain branch effectively contributes to reducing the domain gap. Specifically, we improve performance by +1.4 mIoU on GTA→Cityscapes, by +1.1 mIoU on SYNTHIA→Cityscapes, and by +3.4 mIoU on Cityscapes→ACDC. Interestingly, results show that the increase in segmentation accuracy is remarkably higher in Cityscapes→ACDC than in the two synthetic-to-real scenarios. This arises from the fact that, as both Cityscapes and ACDC are recorded in the real world and partly in the same or very similar cities, the gap in distributions of context relationships is smaller in this benchmark, thus facilitating the transfer of domain-invariant features through cross-domain attention.

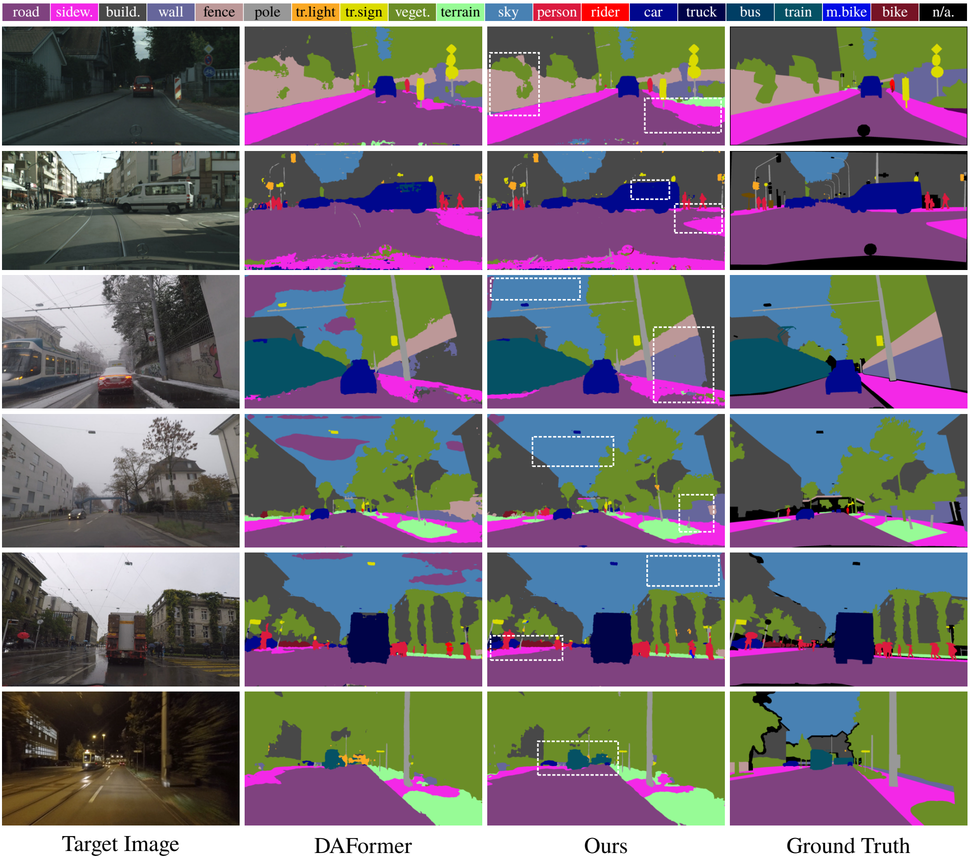
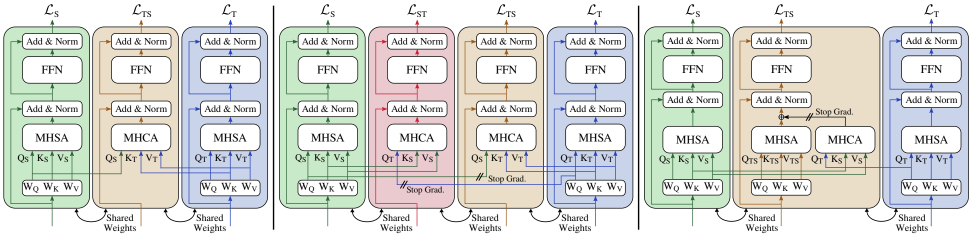


Figure 2: Qualitative semantic segmentation results of our method compared to the baseline DAFormer [10] on GTA→Cityscapes (row 1), SYNTHIA→Cityscapes (row 2), and Cityscapes→ACDC (rows 3-6). An extended analysis on example predictions is provided in the Supplementary Material.

We additionally compare our approach to concurrent work [25], dubbed CDTDA for convenience, that also leverages cross-domain Transformers for UDA in semantic segmentation. Specifically, CDTDA employs a bidirectional cross-domain Transformer backbone following the design of CDTrans [28] and supervises the target branch with an extra consistency loss on attention maps that we do not consider in our framework. Our overall training scheme, despite being simpler and requiring substantially fewer calculations, still achieves superior performance, indicating that our proposed cross-domain Transformer design does a better job at enforcing domain-robust feature learning.

4.4 Analysis of the Proposed Cross-Domain Transformer Design

To further verify the effectiveness of our proposed architecture for domain-adaptive semantic segmentation, we evaluate its performance against existing cross-domain Transformers in the literature, namely CDTrans [28] and CDTDA [25], both shown in Fig. 3. As the goal here is



(a) CDTrans (b) CDTDA (c) Ours

Figure 3: Different cross-domain Transformer variants for UDA. (a) corresponds to the CDTrans [28] design, originally developed for UDA in image classification. (b) is an adapted version of CDTrans which includes bidirectional cross-attention and gradient stopping on the query vectors in the cross-domain attention modules. (c) is our cross-domain Transformer block structure described in Sec. 3.2. $\mathcal{L}_S$ and $\mathcal{L}_{ST}$ are supervised cross-entropy losses using the ground-truth labels for the source domain, while $\mathcal{L}_T$ and $\mathcal{L}_{TS}$ are unsupervised cross-entropy losses using pseudo-labels produced by the EMA-teacher network.

		Training	
Architecture	mIoU	Throughput	GPU Mem.
DAFormer [11]	68.1 ± 0.7	0.70 it/s	9.81 GB
CDTrans [28]	68.8 ± 0.4	0.44 it/s	17.51 GB
CDTDA [25]	68.9 ± 0.6	0.37 it/s	13.35 GB
Ours	69.7 ± 0.4	0.52 it/s	9.81 GB

Table 2: Throughput and memory consumption of DAFormer and the cross-domain Transformer variants in Fig. 3 during training on a NVIDIA A10G GPU, together with the achieved segmentation accuracy averaged over 3 random seeds. Results are obtained by integrating each of the evaluated architectures in our baseline UDA framework.

to compare the architecture designs, we integrate each of the aforementioned cross-domain Transformer variants in our UDA framework and report their performance on the challenging GTA→Cityscapes benchmark. For reference, we also report the performance achieved using the original DAFormer architecture with our baseline UDA pipeline (i.e. only using the standard source and target branches, without leveraging cross-domain attention). Results in Tab. 2 prove that, while all cross-domain Transformer variants tend to boost UDA performance, our novel design leads to better adaptation capabilities (+0.9 mIoU compared to CDTrans and +0.8 mIoU compared to CDTDA). Tab. 2 additionally reports the runtime and GPU memory footprint of the original DAFormer architecture and the three evaluated cross-domain Transformer designs during training. It is worth noting that both the CDTrans and the CDTDA architectures lead to a considerable increase in GPU memory consumption when compared to DAFormer (+78.5% and 36.1%, respectively), as they require gradient backpropagation through multiple branches simultaneously. Our design, on the contrary, does not suffer from this drawback (note that we backpropagate gradients through each branch separately) and only increases training time compared to DAFormer, as it requires an additional forward/backward pass for the TS branch. During inference, none of the methods lead to computation overhead as images are forwarded through the model's original self-attention branch.

5 Conclusion

In this work, we present a novel framework for robust domain-adaptive semantic segmentation with Transformer-based architectures. Targeting applications where annotated data for the target domain is not available, we introduce a generic three-branch Transformer block that combines self- and cross-attention mechanisms to boost learning of domain-invariant features. In addition, we show how such an architecture can be seamlessly integrated into a state-of-the-art self-training UDA scheme, resulting in a framework that, despite its simplicity, leads to learned models with better generalization capabilities. An extensive evaluation on benchmarking datasets reveals that the proposed framework consistently achieves better results than its baselines without increasing GPU memory consumption, thus comprising a significant step towards achieving more robust models under data-scarce scenarios and limited computational resources. Future work will investigate the application of the proposed cross-domain Transformer design to boost UDA capabilities in other computer vision problems, such as object detection and panoptic segmentation.

Acknowledgements

This work was supported by the Amazon Research Awards program and the Swiss National Science Foundation (SNF) through the National Centre of Competence in Research on Digital Fabrication (NCCR DFAB).

References

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

[4] 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 (TPAMI)*, 40(4):834–848, 2018.

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *European Conference on Computer Vision (ECCV)*, 2020.

[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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[8] Rui Gong, Wen Li, Yuhua Chen, Dengxin Dai, and Luc Van Gool. DLOW: Domain flow and applications. *International Journal of Computer Vision (IJCV)*, 2021.

[9] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

[10] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[11] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. HRDA : Context-aware high-resolution domain-adaptive semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2022.

[12] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[15] Luke Melas-Kyriazi and Arjun K. Manrai. PixMatch: Unsupervised domain adaptation via pixelwise consistency training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. ClassMix: Segmentation-based data augmentation for semi-supervised learning. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[17] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, 2016.

[18] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[19] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[20] 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 (TPAMI)*, 44(6): 3139–3153, 2022.

[21] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain adaptation via cross-domain mixed sampling. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[22] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uskoreit, Lilion Jones, Aidan N. Gomes, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[24] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Kaihong Wang, Donghyun Kim, Regerio Feris, Kate Saenko, and Margrit Betke. Exploring consistency in cross-domain transformer for domain adaptive semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[26] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. Multi-modality cross attention network for image and sentence matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] 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 *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[28] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. CDTrans: Cross-domain transformer for unsupervised domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2022.

[29] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[30] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[31] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[32] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

[33] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V.K.Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.