Attention Decomposition for Cross-Domain Semantic Segmentation

Liqiang He and Sinisa Todorovic

Oregon State University, Corvallis, OR 97330, USA
{heli,sinisa}@oregonstate.edu

Abstract. This work addresses cross-domain semantic segmentation. While recent CNNs and proposal-free transformers led to significant advances, we introduce a new transformer with a lighter encoder and more complex decoder with query tokens for predicting segmentation masks, called ADFormer. The domain gap between the source and target domains is reduced with two mechanisms. First, we decompose cross-attention in the decoder into domain-independent and domain-specific parts to enforce the query tokens interact with the domain-independent aspects of the image tokens, shared by the source and target domains, rather than domain-specific counterparts which induce the domain gap. Second, we use the gradient reverse block to control back-propagation of the gradient, and hence introduce adversarial learning in the decoder of ADFormer. Our results on two benchmark domain shifts – GTA to Cityscapes and SYNTHIA to Cityscapes – show that ADFormer outperforms SOTA proposal-free methods with significantly lower complexity. The implementation is available at https://github.com/helq2612/ADFormer.

Keywords: Cross-domain settings · Semantic segmentation

1 Introduction

This paper addresses cross-domain semantic segmentation. We are given images from the source and target domains, which share semantic classes, but significantly differ in imaging conditions and appearance of these classes – the difference referred to as the domain gap. Only the source-domain images are annotated with semantic segmentation masks. The goal is to use the provided source-domain supervision to perform semantic segmentation in the target domain. This problem arises in a wide range of applications, including autonomous driving under different weather conditions [19, 26].

A direct application of SOTA models for standard (single-domain) semantic segmentation, e.g., transformers [4, 5, 43], to cross-domain settings typically yields poor results. To reduce the domain gap, prior work has studied several frameworks: (a) decomposition of features into domain-independent and domain-specific parts [39–41]; (b) adversarial learning via a gradient reverse layer (GRL) [17]; (c) self-training; and (d) hybrids thereof [11, 12, 20]. For each of these frameworks, prior work has typically used *proposal-free* models which have a *heavy encoder*, either

CNNs [30, 36, 44, 46] or transformers [13–15]. By proposal-free, we mean models that do not explicitly use proposals to obtain the final semantic segmentation. For example, a proposal-free transformer does not have query tokens in its decoder. Also, by heavy encoder, we mean a complex image-encoding network pre-trained on a very large dataset, like ImageNet [8]. We find that such proposal-free models may not be the most suitable for our problem, because their pretraining is done for image classification – a task different from semantic segmentation – on a large dataset which may be unrelated to the source and target domains of interest. While there are attempts to address this issue – e.g., "thing-class feature distance" regularization [13] – they poorly generalize to all possible cross-domain settings – e.g., the target domain may not have "thing" classes of ImageNet [35].

This motivates us to use a *proposal-based* transformer in which learnable proposals (a.k.a. query tokens) serve as feature prototypes of semantic classes, and where domain alignment is performed in the decoder. The decoder is aimed at decoding the proposals into their semantic segmentation in the input image. As the decoder is learned on both the source and target datasets, we can directly learn how to align the proposals across the two domains. Therefore, our model choice seems more suitable for the downstream cross-domain setting than proposal-free models. Among recent proposal-based transformers [1,4,5,45,47], Mask2Former [4] has shown great success in single-domain semantic segmentation. In the decoder of Mask2Former, the "abstracted" query tokens are gradually refined and tuned to the input image, by the means of cross-attention with the image tokens, such that each query can predict a class and its segmentation mask in the image at the output. We extend Mask2Former such that the query tokens learn domain-independent class prototypes, as desired in cross-domain settings. To the best of our knowledge, proposal-based transformers have never been used in cross-domain semantic segmentation.

The cross-attention in Mask2Former estimates similarity between the query tokens and image tokens. In cross-domain settings, the latter are characterized by both domain-independent (DI) and domain-specific (DS) features. Consequently, the cross-attention consists of DI and DS attentions. The DS cross-attention tuned to the source domain may negatively affect the refinement of the query tokens on the target-domain images. To mitigate this issue, prior work [11] has directly decomposed the query and image tokens into DI and DS features, at a high computational cost. In contrast, as shown in Fig. 1, our new attention decomposition transformer, called *ADFormer*, splits the Mask2Former's semantic-segmentation head into two, parallel, DI and DS processing branches, for a more direct estimation of the two types of cross-attention, with low complexity.

To learn to decompose the cross-attention, following [13, 30], we train our ADFormer with self-learning on both source and *domain-mixed* images, where the latter are synthetic images composed of select parts from the source- and target-domain images, as illustrated in Fig. 2. The supervision for self-learning represents the known masks of the corresponding source-domain and target-domain regions in the domain-mixed image. Such training allows each query token of our ADFormer to predict a quadruplet *(class, mask, domain, domain-*

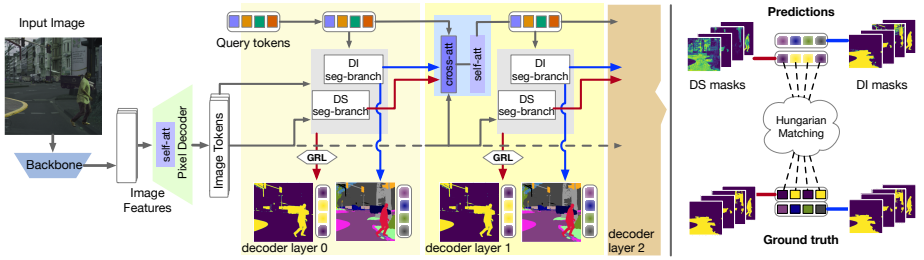

Fig. 1: ADFormer consists of a backbone, pixel-decoder, and transformer-decoder. The transformer-decoder has two parallel domain-independent (DI) and domain-specific (DS) branches that decompose the cross-attention into DI and DS components, and enable every query token to predict a quadruplet *(class, mask, domain, domain-mask)*. The DI segmentation head outputs the semantic class and its segmentation mask in the image, and the DS segmentation head outputs the domain class (source or target) and its domain-mask in the image. The predicted quadruplets are matched with the ground truth to estimate loss. The gradient reverse layer (GRL) is conveniently introduced for the loss on DS outputs to reduce the domain gap.

mask). As shown in Fig. 1, the first pair in the quadruplet is output by the DI head for the semantic class and its segmentation mask in the image, and the second pair is output by the DS head for the domain class (source or target) and its segmentation mask in the image. The predicted quadruplets are matched with the ground truth to estimate loss. The gradient reverse layer (GRL) is conveniently introduced for the loss on DS outputs to reduce the domain gap.

ADFormer achieves SOTA performance on two benchmark domain shifts, GTA [27] → Cityscapes [7]) and SYNTHIA [28] → Cityscapes [7], with lower complexity than prior work.

Our contributions are summarized below:

1. We are the first to design and evaluate a proposal-based model, ADFormer, and integrate adversarial learning via GRL into self-learning of a proposal-based model for cross-domain semantic segmentation.
2. ADFormer's decoder explicitly decomposes cross-attention into DI and DS components, which enforces the query tokens to focus their learning of class prototypes on the semantically relevant DI cues in the image.

In the following, Sec. 2 reviews related work, Sec. 3 briefly outlines Mask2Former, Sec. 4 specifies ADFormer, Sec. 5 presents our experimental results, and Sec. 6 concludes the paper.

2 Related Work

Transformer-based segmentation. Recent work shows that transformers [2, 4, 5, 43] outperform CNNs in semantic segmentation. For example, Segformer [43] avoids using complex decoders by unifying transformers with lightweight

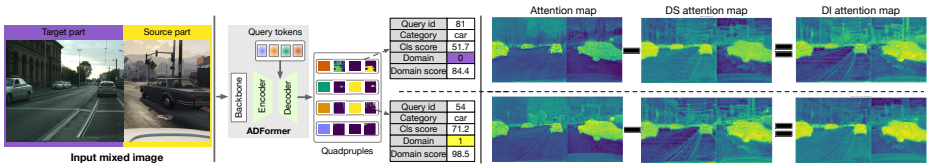

Fig. 2: For a domain-mixed image consisting of source- and target-domain parts, every query token predicts the quadruplet *(class, mask, domain, domain-mask)*. The figure visualizes the cross-attention of two sample query tokens and the corresponding attention decomposition, enabled by the DI and DS heads in ADFormer (see Fig. 1). The top query predicts "target" as the domain class, and the bottom query, "source". Before the decomposition, the top query's cross-attention has higher values on objects (e.g., cars) from the target domain. Similarly, the bottom query's cross-attention emphasizes objects from the source domain. After the decomposition, the resulting DI cross-attention for both queries has high values on objects from both source and target domains.

multi-layer perception (MLP) decoders. MaskFormer [5] estimates the query-based attention and mask classification, while its successor Mask2Former [4] additionally filters image background with the masked attention. Recent cross-domain semantic segmentation methods usually adopt the MIt-B5 encoder from Segformer, whose complexity is very high ($>$ 85M parameters, see Sec. 5). Our ADFormer extends Mask2Former, with significantly lighter encoder (called pixel-decoder), to better suit cross-domain semantic segmentation.

Unsupervised domain adaptation (UDA) for semantic segmentation can be divided into adversarial-training [31, 32, 37] and self-training approaches [13–15, 30]. The former use GRL to reverse the sign of loss and thus rather maximize than minimize the loss for outputs deemed domain-specific. This effectively suppresses learning on features that induce the domain gap. In self-training, a student model is trained on both source- and target-domain images, where supervision in the target domain comes based form pseudo-ground-truth labels generated by a teacher model. The teacher model is updated from the student model usually through exponential moving average (EMA). Recent domain-bridging methods [3, 30] perform training on domain-mixed images composed of source and target parts. In this paper, we follow these well-established learning frameworks, and train ADFormer using adversarial-training via GRL, self-training via student-teacher models, and domain-bridging via domain-mixed images.

Domain Decomposition. Cross-domain learning typically focuses on feature decomposition. For example, CNN features can be disentangled into DI and DS parts with an orthogonal constraint [41], or mutual-information loss [40], or a cyclic-disentanglement [39]. Also, images can be disentangled into DI structure and DS texture representations [38]. In contrast, we focus on attention decomposition, rather than feature decomposition.

3 Brief Review of Mask2Former

ADFormer extends Mask2Former [4], which consists of a backbone network, pixel decoder (also called "encoder" in DETR-family detectors [1, 10, 25, 47]), (query) decoder, and a semantic segmentation branch. The encoder embeds image features, extracted with a backbone network, into a sequence of image tokens, $\mathcal{X} = \{x_i\}_{i=1}^{N}$, $x_i \in \mathbb{R}^d$, where $N = h \times w$ is the sequence length, and $[h, w]$ are the height and width of the image-feature map. The decoder introduces a sequence of query tokens, $\mathcal{Z} = \{z_j\}_{j=1}^{N_\mathrm{q}}$, $z_j \in \mathbb{R}^d$, aimed at learning class prototypes. In the decoder, every query token z_j is correlated with all image tokens $\mathcal{X}$ to compute the semantic-segmentation responses of z_j:

$$\mathcal{M}_j = \{\sigma(m_{ij}) : m_{ij} = x_i^\top \mathrm{FFN}(z_j),\ i = 1, \ldots, N\},\ j = 1, \ldots, N_\mathrm{q}\ , \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function, and $\mathrm{FFN}(\cdot)$ is a feedforward three-layer non-linear projection. For every z_j, the decoder's classification head, denoted as $\mathrm{class}(\cdot)$, predicts a class distribution, $c_j = \mathrm{softmax}(\mathrm{class}(z_j)) \in [0, 1]^C$, over C classes. Thus, the decoder of Mask2Former can be viewed as decoding the image tokens into class-mask responses $\{(c_j, \mathcal{M}_j)\}_{j=1}^{N_\mathrm{q}}$, which can be mapped into the final pixel-wise prediction:

$$y_i = \mathrm{softmax}(\sum_{j=1}^{N_\mathrm{q}} \sigma(m_{ij}) c_j) \in [0, 1]^C, \quad i = 1 \ldots N\ . \tag{2}$$

The decoder usually has several layers, $l = 1, 2, \ldots$, aimed at refining $\mathcal{Z}$ through cross-attention and self-attention before making the final prediction given by (2). The refinement of $\mathcal{Z}^l$ at every layer l, by the means of cross-attention, is important since it focuses every "abstracted" class prototype z_j^l to tune to visual cues captured by the image tokens as:

$$\mathcal{Z}^l = (A^l(\mathcal{X} W_V)) W_O, \quad \mathcal{Z}^l \in \mathbb{R}^{N_\mathrm{q} \times d}, \quad \mathcal{X} \in \mathbb{R}^{N \times d}\ , \tag{3}$$

where we slightly abuse the notation for $\mathcal{Z}$ and $\mathcal{X}$ and treat them as matrices, $W_V, W_O \in \mathbb{R}^{d \times d}$ are two linear projection matrices, and $A^l \in \mathbb{R}^{N_\mathrm{q} \times N}$ is the cross-attention at the decoder layer l estimated as

$$A^l = (\mathcal{Z}^l W_Q)(\mathcal{X} W_K)^\top + M^{l-1}, \quad M(i,j) = \begin{cases} 0\ , & \text{if } \sigma(m_{ij}) > 0.5 \\ -\infty\ , & \text{otherwise} \end{cases}\ , \tag{4}$$

where $W_Q, W_K \in \mathbb{R}^{d \times d}$ are another two linear projection matrices, and M^{l-1} is the attention mask for filtering out image tokens predicted as background by the previous layer $(l-1)$ in (1). M enforces that the query tokens are refined only based on the foreground image tokens.

4 Specification of ADFormer

Fig. 1 shows the main components of ADFormer. In training, we use both source and domain-mixed images. This allows for self-learning of ADFormer by predicting

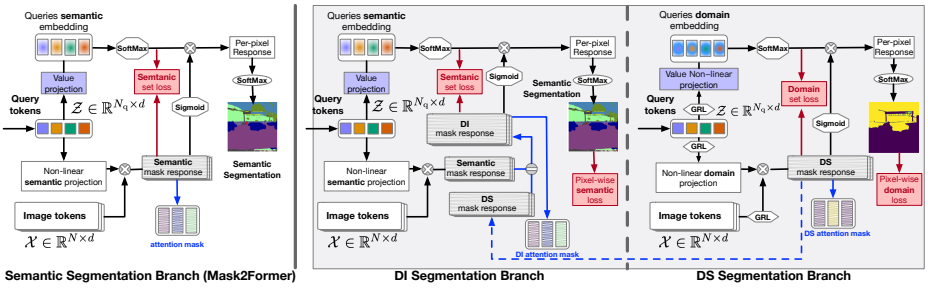

Fig. 3: (*Left*): The single semantic segmentation branch of Mask2Former. (*Middle* and *Right*) The decoder of ADFormer disentangles the class-mask and domain-mask predictions through the respective domain-independent (DI) and domain-specific (DS) segmentation branches.

not only the semantic mask of each class but also the domain mask for the input image. The encoder produces the image tokens, $\mathcal{X}$, which are passed to the decoder for refining the query tokens $\mathcal{Z}$ and decoding $\mathcal{X}$ with $\mathcal{Z}$ into the quadruplet *(class, mask, domain, domain-mask)* responses, $\{(c_j, \mathcal{M}_j^{\text{DI}}, d_j, \mathcal{M}_j^{\text{DS}})\}_{j=1}^{N_q}$. As depicted in more detail in Fig. 3, every z_j is passed to two parallel branches: the DI branch for predicting the semantic segmentation mask $\mathcal{M}_j^{\text{DI}}$ and the DS branch for estimating the domain segmentation mask $\mathcal{M}_j^{\text{DS}}$. In the DS branch, several GRLs are used to reverse the gradient flow in back-propagation to reduce the domain gap. In the following, we specify the DI and DS branches, whose detailed architecture is shown in Fig. 3.

4.1 DS Segmentation Branch

Since both $\mathcal{X}$ and $\mathcal{Z}$ carry information about the domain, regions belonging to the target or source domain in the input image are identified by correlating every z_j with $\mathcal{X}$ as

$$\mathcal{M}_j^{\text{DS}} = \{\sigma(m_{ij}^{\text{DS}}) : m_{ij}^{\text{DS}} = x_i^\top \text{FFN}^{\text{DS}}(z_j),\ i = 1, \dots, N\},\ j = 1, \dots, N_q\ , \quad (5)$$

For every z_j, the binary domain classification head, denoted as domain$(\cdot)$, predicts a domain-class distribution, $d_j = \text{softmax}(\text{domain}(z_j)) \in [0,1]^2$, over two domain classes source and target. The domain$(\cdot)$ is implemented as a FFN. Thus, the DS segmentation branch can be viewed as decoding $\mathcal{X}$ into domain-mask responses $\{(d_j, \mathcal{M}_j^{\text{DS}})\}_{j=1}^{N_q}$, which can be mapped into the pixel-wise domain classification:

$$y_i^{\text{DS}} = \text{softmax}(\sum_{j=1}^{N_q} \sigma(m_{ij}^{\text{DS}}) d_j) \in [0,1]^2, \quad i = 1 \dots N\ . \quad (6)$$

Importantly, the refinement of $\mathcal{Z}$ via cross-attention is not computed in the DS branch, since the query tokens are supposed to learn DI class prototypes.

The DS branch uses GRL to reduce the domain gap, as shown in Fig. 3. In the forward pass, GRL is the identity function. In back-propagation, GRL reverses the sign of the gradient of loss in order to "fool" the DS branch such that errors in predicting the domain classes are encouraged. The hyperparameter λ_{GRL} of the reversed gradient is empirically set to a very small value ($\lambda_{\text{GRL}} = 10^{-4}$). Thus, on the one hand, in back-propagation, GRL effectively makes the network ignore domain differences and focus more on the DI visual cues. On the other hand, the loss incurred on the per-query predictions $\{(d_j, \mathcal{M}_j^{\text{DS}})\}_{j=1}^{N_{\text{q}}}$ and pixel-wise domain classification $\{y_i^{\text{DS}}\}_{i=1}^{N}$ make the network learn the domain differences. In this way, GRL enables adversarial learning of the DS branch.

4.2 DI Segmentation Branch

The DI segmentation branch decodes $\mathcal{X}$ into DI class-mask responses. While the DS branch leverages the domain information entangled in both $\mathcal{X}$ and $\mathcal{Z}$ for the pixelwise domain classification in (6), the DI branch seeks to estimate DI correlation between $\mathcal{X}$ and $\mathcal{Z}$. Following prior work, we make two common assumptions about the DI and DS components of the image tokens and query tokens. First, they can be decomposed as:

$$x_i = x_i^{\text{DI}} + x_i^{\text{DS}}, \quad z_j = z_j^{\text{DI}} + z_j^{\text{DS}} \ . \tag{7}$$

Second, they are uncorrelated (orthogonal):

$$(x_i^{\text{DS}})^\top z_j^{\text{DI}} = 0, \quad (x_i^{\text{DI}})^\top z_j^{\text{DS}} = 0 \ . \tag{8}$$

From (7), (8), the DI correlation between every x_i and z_j can be computed as

$$\begin{aligned} x_i^\top z_j &= (x_i^{\text{DI}})^\top z_j^{\text{DI}} + (x_i^{\text{DS}})^\top z_j^{\text{DS}} + \cancel{(x_i^{\text{DI}})^\top z_j^{\text{DS}}} + \cancel{(x_i^{\text{DS}})^\top z_j^{\text{DI}}}, \\ \Rightarrow \quad & (x_i^{\text{DI}})^\top z_j^{\text{DI}} = x_i^\top z_j - (x_i^{\text{DS}})^\top z_j^{\text{DS}}. \end{aligned} \tag{9}$$

From (9), the DI semantic-segmentation responses of the query tokens can be estimated as

$$\mathcal{M}_j^{\text{DI}} = \{\sigma(m_{ij}^{\text{DI}}) : m_{ij}^{\text{DI}} = m_{ij} - m_{ij}^{\text{DS}},\ i = 1, \ldots, N\},\ j = 1, \ldots, N_q \ . \tag{10}$$

where m_{ij} is given by (1) and m_{ij}^{DS} is computed in (5).

Note that $\mathcal{M}_j^{\text{DI}}$ in (10) does not require an explicit decomposition of features of the image and query tokens. It suffices to simply subtract the domain-mask prediction $\mathcal{M}_j^{\text{DS}}$ from the entangled semantic mask $\mathcal{M}_j$ which can be predicted as in Mask2Former (1).

As in Mask2Former, the DI branch has a classification head, denoted as class$(\cdot)$, which predicts a class distribution, $c_j = \text{softmax}(\text{class}(z_j)) \in [0, 1]^C$. Thus, the DI segmentation branch outputs semantic-segmentation responses $\{(c_j, \mathcal{M}_j^{\text{DI}})\}_{j=1}^{N_{\text{q}}}$, which can be mapped into the pixel-wise semantic classification:

$$y_i = \text{softmax}(\sum_{j=1}^{N_{\text{q}}} \sigma(m_{ij}^{\text{DI}}) c_j) \in [0, 1]^C, \quad i = 1 \ldots N \ . \tag{11}$$

Similar to Mask2Former, ADFormer's decoder refines $\mathcal{Z}^l$ through layers l, by the means of cross-attention, only in the DI segmentation branch, because we want the class prototypes in $\mathcal{Z}^l$ to capture DI properties of the image tokens as:

$$\mathcal{Z}^l = ((A^{\mathrm{DI}})^l(\mathcal{X}W_V))W_O, \quad \mathcal{Z}^l \in \mathbb{R}^{N_{\mathrm{q}} \times d}, \quad \mathcal{X} \in \mathbb{R}^{N \times d}\,, \tag{12}$$

where $(A^{\mathrm{DI}})^l \in \mathbb{R}^{N_{\mathrm{q}} \times N}$ is the cross-attention of the DI branch at layer l:

$$(A^{\mathrm{DI}})^l = (\mathcal{Z}^l W_Q)(\mathcal{X}W_K)^\top + (M^{\mathrm{DI}})^{l-1}, \quad M^{\mathrm{DI}}(i,j) = \begin{cases} 0\,, & \text{if } \sigma(m_{ij}^{\mathrm{DI}}) > 0.5 \\ -\infty\,, & \text{otherwise} \end{cases}, \tag{13}$$

where M^{DI} is the DI attention mask for enforcing the refinement of the query tokens based only on the DI foreground image tokens.

4.3 Loss Functions

ADFormer predicts per-query quadruplets *(class, mask, domain, domain-mask)* and pixel-wise semantic and domain classification. This incurs the set loss and the pixel-wise loss, as described below.

Set loss is estimated with the one-to-one Hungarian matching between the quadruplet predictions $\{(c_j, \mathcal{M}_j^{\mathrm{DI}}, d_j, \mathcal{M}_k^{\mathrm{DS}})\}_{j=1}^{N_{\mathrm{q}}}$ and the ground truth appropriately converted to ground-truth quadruplets $\{(c_k^*, \mathcal{M}_k^*, d_k^*, (\mathcal{M}_k^{\mathrm{DS}})^*)\}_{k=1}^{N_{\mathrm{gt}}}$. Let $h(k)$ denote the Hungarian mapping of every kth ground truth to its best matched jth prediction, $h(k) = j$. Then, the set loss is specified as

$$\mathcal{L}_{\mathrm{set}} = \sum_{k=1}^{N_{\mathrm{gt}}} \mathcal{L}_{\mathrm{M2F}}((c_k^*, \mathcal{M}_k^*), (c_{h(k)}, \mathcal{M}_{h(k)}^{\mathrm{DI}})) + \mathcal{L}_{\mathrm{DS}}((d_k^*, (\mathcal{M}_k^{\mathrm{DS}})^*), (d_{h(k)}, \mathcal{M}_{h(k)}^{\mathrm{DS}}))\,, \tag{14}$$

where $\mathcal{L}_{\mathrm{M2F}}$ is the same set loss used in Mask2Former, which includes the cross-entropy semantic classification loss and the regression loss for the semantic mask responses. Similarly, $\mathcal{L}_{\mathrm{DS}}$ includes the cross-entropy domain classification loss and the domain-mask regression loss. The regression losses for both $\mathcal{L}_{\mathrm{M2F}}$ and $\mathcal{L}_{\mathrm{DS}}$ include the pixel-wise binary cross-entropy loss and the dice loss.

Segmentation loss. In addition to the set loss, we take into account the pixel-wise supervision for both semantic segmentation and domain segmentation:

$$\mathcal{L}_{\mathrm{seg}} = \sum_{i=1}^{N} \mathcal{L}_{\mathrm{CE}}(y_i^*, y_i) + \mathcal{L}_{\mathrm{BCE}}((y_i^{\mathrm{DS}})^*, y_i^{\mathrm{DS}})\,, \tag{15}$$

where CE and BCE denote the pixel-wise cross-entropy loss and binary cross-entropy loss, and * denotes the corresponding ground truth at ith pixel.

Our cross-domain learning. We adopt the self-training framework of [13,30], and train the *student*-ADFormer with the overall loss:

$$\mathcal{L} = \lambda_{\mathrm{set}}(\mathcal{L}_{\mathrm{set}}^{\mathrm{source}} + \mathcal{L}_{\mathrm{set}}^{\mathrm{mix}}) + \lambda_{\mathrm{seg}}(\mathcal{L}_{\mathrm{seg}}^{\mathrm{source}} + \mathcal{L}_{\mathrm{seg}}^{\mathrm{mix}})\,, \tag{16}$$

where $\lambda_{\text{set}}, \lambda_{\text{seg}}$ are positive hyperparameters, and "source" and "mix" denote loss from the source-domain and domain-mixed images, respectively. The "mix" loss is evaluated with respect to pseudo-ground-truth labels, predicted by the *teacher*-ADFormer on the target-domain parts, and actual ground-truth labels for the source-domain parts in the domain-mixed images. The teacher-ADFormer updates its weights from the student-ADFormer via exponential moving average.

5 Experimental Results

Datasets. Following prior work [13–15], we evaluate ADFormer on two domain shifts: GTA [27] → Cityscapes [7], and SYNTHIA [28] → Cityscapes [7]. The source-domain datasets GTA and SYNTHIA consist of 24,996 and 9400 annotated images, respectively. The target-domain dataset Cityscapes consists of 2975 and 500 images for training and validation.

Evaluation metric. As in [13–15], ADFormer is evaluated with respect to the mean intersection-over-union (mIoU) of semantic segmentation in target-domain images on 19 classes of GTA → Cityscapes and 16 classes of SYNTHIA → Cityscapes.

Network architecture. The architecture of ADFormer is implemented using the mmsegmentation framework [6] of Mask2former [4]. As the backbone network, we use the lightweight Swin-S [21], pre-trained on the ImageNet-1K [8]. The encoder in ADFormer has six layers, and the decoder's mandatory zero layer is followed by six additional layer (i.e., there are 7 decoder layers).

Parameter setting. As in [13], we train ADFormer with AdamW [22], learning rates $\eta_{\text{base}} = 1e-4$ and $\eta_{\text{backbone}} = 1e-5$, weight decay of 0.05, linear learning rate warm-up $t_{warm} = 1.5$K iterations, and linear decay after that. The hyperparameter of the segmentation loss is $\lambda_{\text{seg}} = 10.0$ and the hyperparameter of the set loss is $\lambda_{\text{set}} = 0.025$. The gradient reverse hyperparameter in the GRLs of the DS branch is $\lambda_{\text{GRL}} = 0.0001$. As in [13], we adopt the Rare Class Sampling strategy to handle the class imbalance in the datasets, but do not use the "Thing-Class ImageNet Feature Distance" regularization. The experiments were conducted on a NVIDIA-H100.

Ablation Study. Tab. 1 shows how individual components of ADFormer and our design choices affect our performance on GTA→ Cityscapes. We organize the ablations as follows:

- **baseline**: Mask2Former is the baseline model that predicts *(class, mask)* for every query token.
- **DS**: The DS segmentation branch is added to Mask2Former, and the cross-attention is computed as in (4); the model predicts *(class, mask, domain, domain-mask)* for every query token.
- **AD**: The attention decomposition is performed, and the refinement of the query tokens via the DI cross-attention is computed as in (12) and (13).
- **GRL**: GRLs are used in the DS branch.
- **FD**: "Thing-class ImageNet feature distance" regularization of [13] is used.

– $\mathcal{L}_{\mathbf{seg}}$: In addition to the set loss, the model is trained on the pixel-wise segmentation loss $\mathcal{L}_{\text{seg}}$.

All of the ablated models and the baseline in Tab. 1 use the rare-class sampling from [13] and self-training of [30], and follow the same training strategies as in [13]. Interestingly, when we train ADFormer with the FD regularization, our mIoU decreases to 66.3. From Tab. 1, adding each proposed component to the baseline gradually improves our performance. ADFormer achieves the best performance for the configuration shown in the last bottom row 8 of Tab. 1, which is used further in the remaining experiments.

	baseline	DS	AD	GRL	FD	$\mathcal{L}_{\text{seg}}$	mIoU
1	✓	-	-	-	-	-	44.1
2	✓	-	-	-	-	✓	51.4
3	✓	✓	-	-	-	✓	56.6
4	✓	✓	✓	-	-	✓	57.5
5	✓	✓	-	✓	-	✓	63.1
6	✓	✓	✓	✓	-	-	65.0
7	✓	✓	✓	✓	✓	✓	66.3
8	✓	✓	✓	✓	-	✓	**69.2**

Table 1: Contribution of each proposed component to the results of ADFormer on GTA→Cityscapes. Baseline is Mask2Former, DS is the DS segmentation branch, AD is the attention decomposition, FD is the feature distance regularization of [13].

Complexity vs. Performance. Tab. 2 presents a trade-off between complexity and performance of ADFormer when using different backbone networks on GTA→Cityscapes. Complexity is evaluated in terms of the number of model parameters, and average running time in inference tested on a single NVIDIA-H100 machine. The table also compares DAFormer [13] with the variant of ADFormer which uses the same backbone MiT-B5 as DAFormer. Other SOTA approaches [14, 15] also use MiT-B5 as the backbone. ADFormer with Swin-S as the backbone has significantly fewer model parameters than DAFormer (76% #Param), and outperforms DAFormer in terms of mIoU (+0.9) and speed (1.7 times faster). When ADFormer uses the larger backbone Swin-B, mIoU improves from 69.2 to 70.4; however, at the cost of increasing both model and computational complexity. Therefore, we find that ADFormer with Swin-S represents our optimal trade-off between complexity and performance, and is used in the remaining experiments including the comparison with SOTA in Tab. 6.

Sensitivity to the loss hyperparameters. Tab. 3 exhaustively tests our sensitivity to the hyperparameters λ_{set} and λ_{seg}, which control the loss in (16), on the domain shift GTA→Cityscapes. From Tab. 3, a careful selection of λ_{set} and λ_{seg} is critical, where our performance is less sensitive to λ_{set} than λ_{seg}.

Sensitivity to the strength of the gradient reverse. In Tab. 4, we test the effect of λ_{GRL} on our performance on the domain shift GTA→Cityscapes.

	Model	Backbone	#Param(M)	speed	mIoU
	DAFormer [13]	MiT-B5 [43]	85.1	3.9fps	68.3
1	ADFormer	MiT-B5 [43]	96.2	2.9fps	70.1
2	ADFormer	ResNet101 [9]	58.5	7.5fps	65.6
3	ADFormer	Swin-S [21]	64.3	6.7fps	**69.2**
4	ADFormer	Swin-B [21]	102.5	2.7fps	70.4

Table 2: Trade-off between complexity and performance of ADFormer and DAFormer when using different backbone networks on the domain shift GTA→Cityscapes.

	λ_{set}	λ_{seg}	mIoU		λ_{set}	λ_{seg}	mIoU		λ_{set}	λ_{seg}	mIoU
1	0.01	5.0	58.7	4	0.025	5.0	61.3	7	0.05	5.0	59.8
2	0.01	10.0	65.4	5	0.025	10.0	**69.2**	8	0.05	10.0	67.6
3	0.01	20.0	62.1	6	0.025	20.0	62.4	9	0.05	20.0	62.9

Table 3: Sensitivity to the hyperparameters which control the loss in (16) on GTA→Cityscapes.

As described in Sec. 4.1, λ_{GRL} controls the strength of the gradient of loss with the reversed sign in back-propagation. Note that $\lambda_{\text{GRL}} = -1$ means that we remove GRLs from the DS branch. For $\lambda_{\text{GRL}} = -1$, Tab. 4 shows very poor results, since the DS branch is encouraged to increase the domain gap. $\lambda_{\text{GRL}} = 0$ means that FFN^{DS} and the FFN of the domain classifier will be learned while the other components of the DS branch will receive zero gradient. At $\lambda_{\text{GRL}} = 0$, the result in Tab. 4 suggests that the query tokens are successfully refined based on DI properties of the image tokens. Finally, $\lambda_{\text{GRL}} > 0$ means that the DS branch is trained via adversarial learning, which gives the best performance at $\lambda_{\text{GRL}} = 1e{-}4$. However, when the reversed gradient is too high, e.g. $\lambda_{\text{GRL}} \geq 1e{-}3$, the adversarial learning starts hurting our performance.

λ_{GRL}	-1	-0.1	0	1e-5	1e-4	1e-3	1e-2
mIoU	57.5	59.4	65.4	66.8	**69.2**	67.1	63.4

Table 4: Ablation about gradient reverse control

Sensitivity to the number of decoder layers. Tab. 5 presents the model complexity and performance of ADFormer for different numbers of decoder layers, including the mandatory layer zero. From Tab. 5, the higher the number of decoder layers the better mIoU, but at the cost of increasing model complexity.

Comparison with SOTA. Tab. 6 presents a comparison of ADFormer with SOTA approaches on the two domain shifts. The table splits the methods into the proposal-free and proposal-based groups. Specifically, DAFormer [13], HRDA [14],

# decoder layers	1+1	1+3	1+6	1+9
#param (M)	56.4	59.6	64.3	69.1
mIoU	66.7	67.8	69.2	69.7

Table 5: The number of model parameters and mIoU of ADFormer for the varying total number of decoder layers, including the mandatory layer zero, on GTA→Cityscapes.

MIC [15], CDAC [34] and DiGA [29] fall in the proposal-free group, since they use complex encoders which require pre-training. The other approaches in the table including ADFormer align the domain gap mainly in their CNN-based or transformer-based decoders, and hence belong to the proposal-based group. For a fair comparison, MIC [15], CDAC [34] and DiGA [29] are considered without the high-resolution domain adaptation (HRDA) [14]. From the table, ADFormer outperforms the second best DAFormer on both domain shifts by 0.7 and 1.2 in mIoU.

Qualitative results. Fig. 4 illustrates our results in training on an example domain-mixed image. The figure visualizes the quadruplet predictions of four query tokens which got matched by the Hungarian algorithm to the ground truth segments in the domain-mixed image. The rightmost column of the figure shows that ADFormer succeeds in recognizing and segmenting the very challenging "sidewalk", as well as in classifying that "sidewalk" belongs to the target domain.

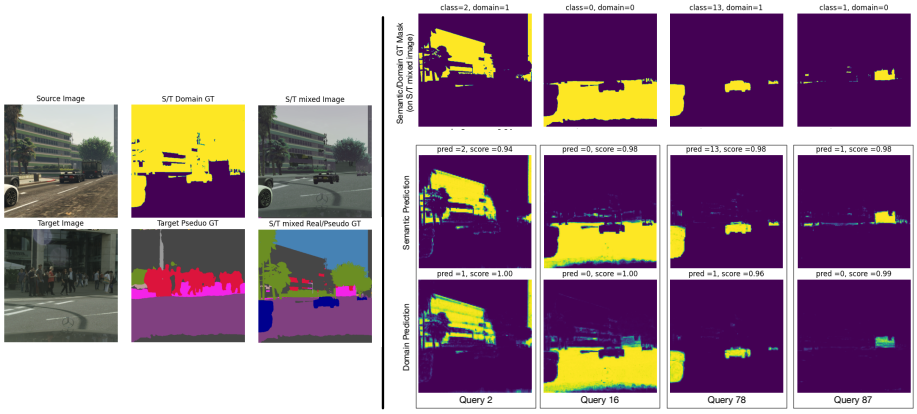

Fig. 4: Visualization of our training. Results of ADFormer on an example training domain-mixed image (3rd column from the left), which represents a collage of parts taken from the source and target images (1st and 2nd columns from the left). On the right-hand-side we visualize predictions of four query tokens matched by the Hungarian algorithm to the ground truth segments (top row on the right). Note that the ground-truth semantic mask and domain mask are equivalent for the domain-mixed image, by construction. Each query predicts the quadruplet *(class, mask, domain, domain-mask)* shown in the middle and bottom rows on the right.

Method	Road	S.walk	Build	Wall	Fence	Pole	Tr.L	Sign	Veg.	Terr	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
GTA → Cityscapes																				
DAFormer† [13]	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
DAFormer*†	96.0	70.6	89.2	54.4	49.3	50.7	56.1	60.9	88.9	42.5	91.7	71.7	43.1	92.6	77.8	71.8	54.9	58.3	64.5	67.6
HRDA† [14]	96.4	74.4	91.0	61.6	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79.0	52.9	93.9	84.1	85.7	75.9	63.9	67.5	73.8
MIC*† [15]	96.7	75.0	90.0	58.2	50.4	51.1	56.7	62.1	90.2	51.3	92.9	72.4	47.1	92.8	78.9	83.4	75.6	54.2	62.6	70.6
CLAN [23]	88.7	35.5	80.3	27.5	25.0	29.3	36.4	28.1	84.5	37.0	76.6	58.4	29.7	81.2	38.8	40.9	5.6	32.9	28.8	45.5
CBST [48]	89.6	58.9	78.5	33.0	22.3	41.4	48.2	39.2	83.6	24.3	65.4	49.3	20.2	83.3	39.0	48.6	12.5	20.3	35	47.0
FADA-MST [33]	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
FDA [44]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
DACS† [30]	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
IAST-MST [24]	94.1	58.8	85.4	39.7	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	24.7	35.2	40.2	52.2
ProCA [18]	91.9	48.4	87.3	41.5	31.8	41.9	47.9	36.7	86.5	42.3	84.7	68.4	43.1	88.1	39.6	48.8	40.6	43.6	56.9	56.3
CorDA [36]	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 [46]	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
CaCo‡ [16]	93.8	64.1	85.7	43.7	42.2	46.1	50.1	54.0	88.7	47.0	86.5	68.1	2.9	88.0	43.4	60.1	31.5	46.1	60.9	58.0
SePiCo†† [42]	95.2	67.8	88.7	41.4	38.4	43.4	55.5	63.2	88.6	46.4	88.3	73.1	49.0	91.4	63.2	60.4	0.0	45.2	60.0	61.0
ADFormer†	96.7	75.1	88.8	57.5	45.9	45.6	55.4	59.8	90.2	45.6	92.1	70.8	43.0	91.0	78.9	79.3	68.7	52.7	65.0	69.2
SYNTHIA → Cityscapes																				
DAFormer† [13]	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
DAFormer*†	89.2	51.1	87.4	31.5	7.8	47.7	53.1	49.3	83.9	-	84.3	73.4	47.0	88.3	-	56.1	-	55.6	60.8	60.4
HRDA† [14]	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	-	92.9	79.4	52.8	89.0	-	64.7	-	63.9	64.9	65.8
MIC*† [15]	83.0	40.9	88.2	37.6	9.0	52.4	56.0	56.5	87.6	-	93.4	74.2	51.4	87.1	-	59.6	-	57.9	61.2	62.2
DiGA*† [29]	85.2	41.4	88.2	42.6	7.5	52.1	57.5	47.7	87.8	-	90.8	75.0	50.8	87.8	-	58.0	-	58.5	63.0	62.1
CDAC*† [34]	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
FDA [44]	84.2	35.1	78.0	6.1	0.44	27.0	8.5	22.1	77.2	-	79.6	55.5	19.9	74.8	-	24.9	-	14.3	40.7	40.5
CBST [48]	53.6	23.7	75.0	12.5	0.3	36.4	23.5	26.3	84.8	-	74.7	67.2	17.5	84.5	-	28.4	-	15.2	55.8	42.5
FADA [33]	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	-	84.0	53.5	22.6	85.4	-	43.7	-	26.8	27.8	45.2
CaCo‡ [16]	87.4	48.9	79.6	8.8	0.2	30.1	17.4	28.3	79.9	-	81.2	56.3	24.2	78.6	-	39.2	-	28.1	48.3	46.0
DACS† [30]	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
IAST-MST [24]	81.9	41.5	83.3	17.7	4.6	32.3	30.9	28.8	83.4	-	85.0	65.5	30.8	86.5	-	38.2	-	33.1	52.7	49.8
CorDA [36]	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	50.0
ProCA [18]	90.5	52.1	84.6	29.2	3.3	40.3	37.4	27.3	86.4	-	85.9	69.8	28.7	88.7	-	53.7	-	14.8	54.8	53.0
ProDA [46]	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
SePiCo†† [42]	77.0	35.3	85.1	23.9	3.4	38.0	51.0	55.1	85.6	-	80.5	73.5	46.3	87.6	-	69.7	-	50.9	66.5	58.1
ADFormer†	91.8	53.6	87.0	40.5	5.2	46.8	52.1	54.9	88.4	-	92.6	72.5	45.7	86.1	-	61.6	-	50.4	64.4	62.1

Table 6: Comparison of ADFormer with SOTA on GTA→Cityscapes and SYNTHIA→Cityscapes. The methods are divided into the proposal-free group (top) and proposal-based group (bottom). We use the official code to reproduce the results of DAFormer [13] without the ImageNet Feature Distance regularization (DAFormer*). For a fair comparison, MIC [15], CDAC [34] and DiGA [29] are considered without the high-resolution domain adaptation [14] (MIC*, CDAC*, and DiGA*). †, ‡, and †† indicate the domain-mix techniques introduced in [30], [16] and [42], respectively.

Fig. 5 and Fig. 6 compare the inference results of ADFormer and DAFormer on a few example validation images from the target Cityscapes domain. These examples support our general observation that ADFormer outperforms DAFormer on challenging classes which are very similar to "thing" classes in ImageNet. One possible explanation for our better performance on such classes is that ADFormer does not use DAFormer's feature distance regularization which favors detection of "thing" classes from ImageNet.

6 Conclusion

We have specified a new proposal-based transformer, ADFormer, for cross-domain semantic segmentation. ADFormer extends Mask2Former with an additional DS segmentation branch aimed at predicting the source-domain or target-domain class at every pixel, which allows for self-learning on the domain-mixed images. In

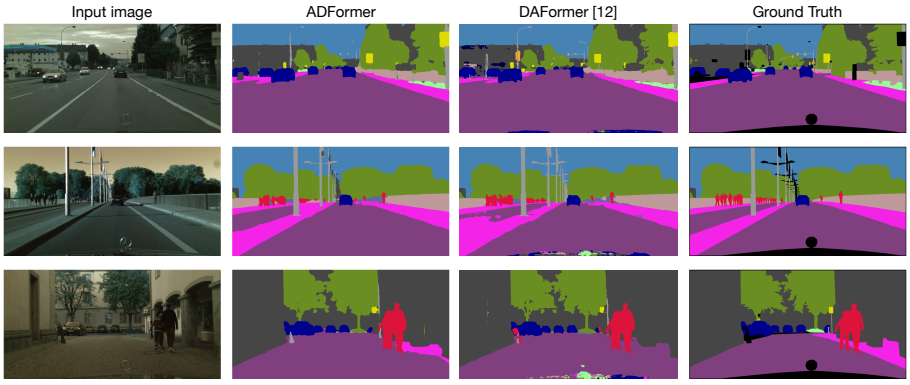

Fig. 5: Inference on example validation images from Cityscapes when ADFormer and DAFormer [13] are trained on GTA→Cityscapes. DAFormer tends to confuse "sidewalk" with "road" (middle) and "person" with "pole" (bottom), whereas ADFormer makes fewer of such mistakes.

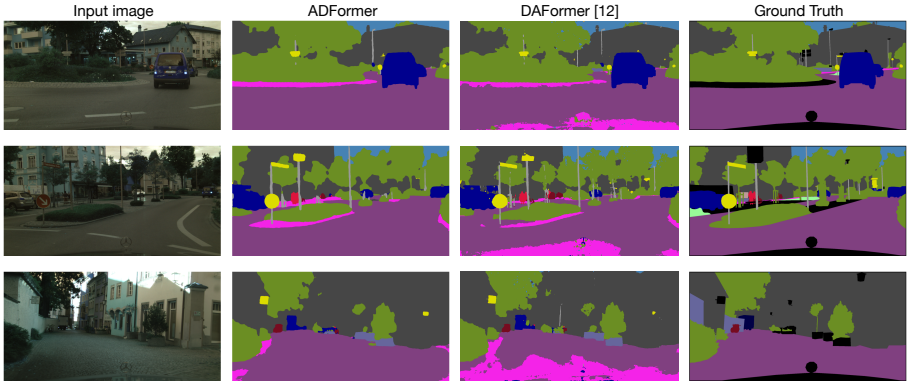

Fig. 6: Inference on example validation images from Cityscapes when ADFormer and DAFormer [13] are trained on SYNTHIA→Cityscapes. DAFormer has a higher false positive rate of "sidewalk" for the ground-truth "road" than ADFormer.

addition, ADFormer decomposes cross-attention into the DI and DS components, and uses only the DI cross-attention for refining the query tokens through seven decoder layers. The attention decomposition and GRLs in the DS branch jointly reduce the domain gap. Evaluation is performed on two benchmark domain shifts: GTA→Cityscapes, and SYNTHIA→ Cityscapes. Our extensive ablation study demonstrates that adding each proposed component to the baseline Mask2Former gradually improves performance, and that ADFormer with Swin-S as the backbone network achieves a good trade-off between complexity and performance. With significantly fewer model parameters, ADFormer outperforms all SOTA approaches on the test domain shifts.

Acknowledgement: This work has been supported by USDA NIFA award No.2021-67021-35344 (AgAID AI Institute).

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
3. Chen, L., Wei, Z., Jin, X., Chen, H., Zheng, M., Chen, K., Jin, Y.: Deliberated domain bridging for domain adaptive semantic segmentation. Advances in Neural Information Processing Systems **35**, 15105–15118 (2022)
4. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1290–1299 (2022)
5. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34**, 17864–17875 (2021)
6. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation` (2020)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. He, L., Todorovic, S.: Destr: Object detection with split transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9377–9386 (2022)
11. He, L., Wang, W., Chen, A., Sun, M., Kuo, C.H., Todorovic, S.: Bidirectional alignment for domain adaptive detection with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18775–18785 (2023)
12. He, M., Wang, Y., Wu, J., Wang, Y., Li, H., Li, B., Gan, W., Wu, W., Qiao, Y.: Cross domain object detection by target-perceived dual branch distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9570–9580 (2022)
13. Hoyer, L., Dai, D., Van Gool, L.: DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9924–9935 (2022)
14. Hoyer, L., Dai, D., Van Gool, L.: Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In: European Conference on Computer Vision. pp. 372–391. Springer (2022)
15. Hoyer, L., Dai, D., Wang, H., Van Gool, L.: MIC: Masked image consistency for context-enhanced domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11721–11732 (2023)

16. Huang, J., Guan, D., Xiao, A., Lu, S., Shao, L.: Category contrast for unsupervised domain adaptation in visual tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1203–1214 (2022)
17. Huang, W.J., Lu, Y.L., Lin, S.Y., Xie, Y., Lin, Y.Y.: Aqt: Adversarial query transformers for domain adaptive object detection. In: International Joint Conference on Artificial Intelligence (IJCAI) (2022)
18. Jiang, Z., Li, Y., Yang, C., Gao, P., Wang, Y., Tai, Y., Wang, C.: Prototypical contrast adaptation for domain adaptive semantic segmentation. In: European conference on computer vision. pp. 36–54. Springer (2022)
19. Larsson, M., Stenborg, E., Hammarstrand, L., Pollefeys, M., Sattler, T., Kahl, F.: A cross-season correspondence dataset for robust semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9532–9542 (2019)
20. Li, Y.J., Dai, X., Ma, C.Y., Liu, Y.C., Chen, K., Wu, B., He, Z., Kitani, K., Vajda, P.: Cross-domain adaptive teacher for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7581–7590 (2022)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
22. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. ICLR (2018)
23. Luo, Y., Liu, P., Zheng, L., Guan, T., Yu, J., Yang, Y.: Category-level adversarial adaptation for semantic segmentation using purified features. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(8), 3940–3956 (2021)
24. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16. pp. 415–430. Springer (2020)
25. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional DETR for fast training convergence. arXiv preprint arXiv:2108.06152 (2021)
26. Mușat, V., Fursa, I., Newman, P., Cuzzolin, F., Bradley, A.: Multi-weather city: Adverse weather stacking for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2915 (2021)
27. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 102–118. Springer (2016)
28. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3234–3243 (2016)
29. Shen, F., Gurram, A., Liu, Z., Wang, H., Knoll, A.: Diga: Distil to generalize and then adapt for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15866–15877 (2023)
30. Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: DACS: Domain adaptation via cross-domain mixed sampling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1379–1389 (2021)

31. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7472–7481 (2018)
32. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2517–2526 (2019)
33. Wang, H., Shen, T., Zhang, W., Duan, L.Y., Mei, T.: Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In: European conference on computer vision. pp. 642–659. Springer (2020)
34. Wang, K., Kim, D., Feris, R., Betke, M.: Cdac: Cross-domain attention consistency in transformer for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11519–11529 (2023)
35. Wang, T., Sankari, P., Brown, J., Paudel, A., He, L., Karkee, M., Thompson, A., Grimm, C., Davidson, J., Todorovic, S.: Automatic estimation of trunk cross sectional area using deep learning. Collaborative Robotics & Intelligent Systems Institute, Oregon State University (2023)
36. Wang, Y., Zhang, R., Zhang, S., Li, M., Xia, Y., Zhang, X., Liu, S.: Domain-specific suppression for adaptive object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9603–9612 (2021)
37. Wang, Z., Yu, M., Wei, Y., Feris, R., Xiong, J., Hwu, W.m., Huang, T.S., Shi, H.: Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12635–12644 (2020)
38. Wei-Lun, C., Hui-Po, W., Wen-Hsiao, P., Wei-Chen, C.: All about structure: Adapting structural information across domains for boosting semantic segmentation. In: CVPR (2019)
39. Wu, A., Deng, C.: Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 847–856 (2022)
40. Wu, A., Han, Y., Zhu, L., Yang, Y.: Instance-invariant domain adaptive object detection via progressive disentanglement. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
41. Wu, A., Liu, R., Han, Y., Zhu, L., Yang, Y.: Vector-decomposed disentanglement for domain-invariant object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9342–9351 (2021)
42. Xie, B., Li, S., Li, M., Liu, C.H., Huang, G., Wang, G.: Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 9004–9021 (2023)
43. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021)
44. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4085–4095 (2020)
45. Zhang, H., Li, F., Xu, H.S., Huang, S., Liu, S., shuan Ni, L.M., Zhang, L.: MP-Former: Mask-piloted transformer for image segmentation. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)

46. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12414–12424 (2021)
47. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
48. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European conference on computer vision (ECCV). pp. 289–305 (2018)