Bidirectional Alignment for Domain Adaptive Detection with Transformers

Liqiang He[1], Wei Wang[2], Albert Chen[2], Min Sun[2], Cheng-Hao Kuo[2], and Sinisa Todorovic[1]

[1]Oregon State University, Corvallis, OR, USA
{heli, sinisa}@oregonstate.edu
[2]Amazon, Bellevue, WA, USA
{wweiwan, aycchen, minnsun, chkuo}@amazon.com

Abstract

We propose a Bidirectional Alignment for domain adaptive Detection with Transformers (BiADT) to improve cross domain object detection performance. Existing adversarial learning based methods use gradient reverse layer (GRL) to reduce the domain gap between the source and target domains in feature representations. Since different image parts and objects may exhibit various degrees of domain-specific characteristics, directly applying GRL on a global image or object representation may not be suitable. Our proposed BiADT explicitly estimates token-wise domain-invariant and domain-specific features in the image and object token sequences. BiADT has a novel deformable attention and self-attention, aimed at bi-directional domain alignment and mutual information minimization. These two objectives reduce the domain gap in domain-invariant representations, and simultaneously increase the distinctiveness of domain-specific features. Our experiments show that BiADT achieves very competitive performance to SOTA consistently on Cityscapes-to-FoggyCityscapes, Sim10K-to-Citiscapes and Cityscapes-to-BDD100K, outperforming the strong baseline, AQT, by 1.9, 2.1, and 2.4 in mAP$_{50}$, respectively.

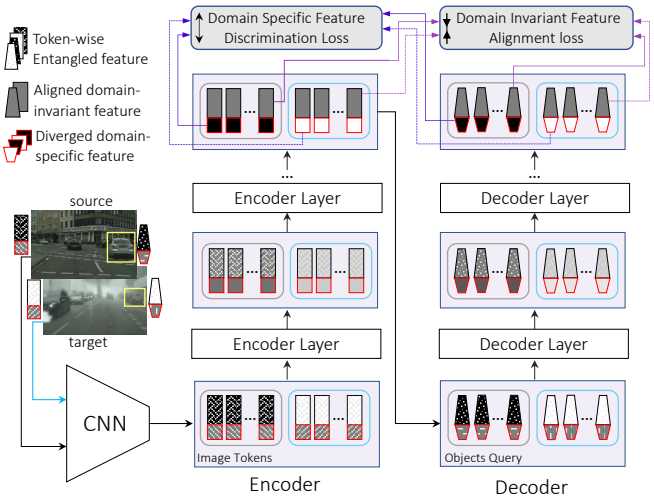

Figure 1. The proposed BiADT is a transformer that consists of the encoder and decoder (for clarity, we only show 2-encoder-layer and 2-decoder-layer). At the input, the encoder takes image-level entangled features (black-white pattern) as image tokens and gradually disentangles them into domain-specific features (marked black for the source and white for the target) and domain-invariant features (marked gray). The decoder decodes object queries on the image, and similarly disentangles the features. We say that the encoder and decoder perform bidirectional feature alignment. This means that they both seek to align domain-invariant features and increase distinctiveness of domain-specific features, at their respective image and object levels.

1. Introduction

This paper focuses on cross-domain object detection – an important problem for vision applications which requires a detector trained on source-domain images generalize well on target-domain images. We say that there is a "*domain gap*" (or "*domain shift*") between the source and target domains, since their respective images significantly differ in appearance and texture, while they do share the same object classes of interest. Despite recent advances in standard object detection [32, 26, 2, 49, 25], their direct application in the cross-domain settings typically yields poor results.

We address cross-domain object detection within the unsupervised domain adaptation (UDA) framework, where training of a detector has access to images of both source and target domains, but object annotations are only available for the source images. A common approach is to learn a domain agnostic feature space in which feature distributions from the source and target domains are aligned by metric learning [29, 15] or adversarial learning [46, 8, 30, 4].

However, we find that this unified feature alignment, at the image and object levels, may be not appropriate for non-trivial domain shifts which characterize most real-world settings. For example, when appearance of objects is the same in the source and target domains[1], these approaches tend to over-align features of such objects [21]. Also, enforcing domain alignment of both image and object features in a unified manner often leads to feature misalignments [21], which could remove critical contextual cues for detection.

To address the above limitation, in this paper, we present a Bidirectional Alignment for domain adaptive Detection with Transformers – BiADT. As shown in Fig. 1, it effectively disentangles features into domain-invariant ($\mathcal{I}$) and domain-specific ($\mathcal{D}$) features, and performs bidirectional alignment, which indicates BiADT seeks to align $\mathcal{I}$-features (i.e., reduce the domain gap in $\mathcal{I}$-features), and simultaneously increase distinctiveness of $\mathcal{D}$-features. Architecture-wise, BiADT leverages the recent successful family of transformer-based detectors – namely, Dab-Deformable-Detr [25] and Deformable-Detr [49] – and extends these architectures with the proposed bidirectional feature alignment in the *deformable-attention* and *self-attention* components. The former exists in both the encoder and decoder of the transformer, while the latter exists in the decoder only.

In addition to the above mentioned feature disentanglement and bidirectional feature alignment, BiADT has a domain embedding component in each unit of the image and object-query token sequences. This component seeks to integrate $\mathcal{D}$-features from the image and object embeddings, and can be easily supervised as the domain label of each training image is known.

The closest prior work that also uses the transformer architecture for domain-adaptive object detection is AQT [17]. AQT aligns features via three adversarial tokens space-wise, channel-wise and instance-wise. The corresponding attention modules in AQT guide the three adversarial tokens to align features for the entire image and the entire object sequence. Importantly, AQT does not disentangle features into the domain-invariant and domain-specific ones. Not only do we explicitly disentangle features, but we also do so for each token individually, at both the image level and the object level. Moreover, not only do we align $\mathcal{I}$-features, but also increase discriminativeness of $\mathcal{D}$-features for each token.

Recently, the teacher-student self-training (TSST) has been extended to address cross-domain object detection with adversarial learning and data augmentation [23, 13, 18]. TSST relies on generating reliable pseudo-labels, and model learning involves complex multi-stage training procedures. As our experiments demonstrate, our BiADT can be easily integrated in the TSST framework.

[1]See Fig. 5: the roads in the source image and the foggy target image look similar, so there is less need to align their features.

Below, we summarize our key contributions:

1) We are the first to seamlessly integrate token-wise domain alignment in the standard attention module of the transformer architecture. Specifically, we propose new designs of the deformable attention and self-attention in the transformer that explicitly disentangle features into domain-invariant and domain-specific features, as well as perform their bidirectional alignment.

2) We propose a new token-wise domain embedding, at both the image- and object-token sequences in the transformer, for predicting the domain label of images and objects. This facilitates extraction of domain-specific features.

3) Our experiments demonstrate that BiADT produces very competitive performance to SOTA on three benchmark cross-domain datasets: Cityscapes→FoggyCityscapes, Sim10K→Citiscapes and Cityscapes→BDD100K. We also test different ablations and variants of BiADT, including its integration with the AQT alignments and self-training.

2. Related Work

2.1. Transformer based Object Detection

Recently, Carion *et al*. [2] proposed an end-to-end detection model DETR that breaks new ground for object detection. DETR is composed of a CNN backbone followed by a Transformer encoder-decoder [37]. The encoder applies a series of transformer layers on 2D flattened image features from the CNN backbone, while the decoder takes a set of learnable object queries as input and tries to fill the queries with the encoding features from detected objects. DETR does not rely on anchors, and there is no need for non-maximum suppression. This makes the whole framework end-to-end optimizable. These attractive properties inspired many following researches to further improve its performance. For example, conditional DETR [27] decouples the query into two parts of content and position, enforcing a correspondence between a query and a specific spatial embedding. Deformable-Detr [49] directly treats 2D reference points as queries to perform cross-attention. DAB-Detr [25] interprets queries as 4-D anchor boxes and optimize them progressively. In this paper, the combined Dab-Deformable-Detr is used as our base detector.

2.2. UDA based Object Detection

UDA effectively bridges the "domain gap" between labeled training data and unlabeled target data. Adversarial learning uses a domain classifier to predict which domain the input comes from, and uses a gradient reverse layer to confuse the classifier and extract domain invariant features. Adversarial-learning based UDA has been widely used in recent approaches to tackle cross domain object detection [33, 14, 45, 16, 39, 17]. For example, [4] first uses Faster R-CNN as a detector and applies image and object domain

classifiers to align cross-domain features via adversarial-learning. Other kinds of feature alignments have also been proposed, including *e.g.*, multi-scale [14], contextual [11], spatial attention [5], topological relation [10], local prototypes [28], object localizer [24, 20], semantic adaptation [44], category-consistency [45], domain-specific suppression [39], strong-weak alignment [33], multi-level alignments [14, 16], and teacher-student self-training [3, 13, 23]. With the emergence of transformer-based detectors, DETR family of models have also been applied to UDA. For example, AQT [17], SFA [38], O2Net [9] and MTTrans [18] aggregate and align features from the entire token sequence of the two domains.

2.3. Domain Feature Decomposition

Domain feature decomposition has been explored in recent cross-domain object detection. For example, VDD [43] disentangles CNN deep features into domain-invariant and domain-specific features via an orthogonal constraint. PDN [42] designs a progressive disentangled convolutional network to extract instance-invariant features with a three-stage training mechanism, and uses a mutual information based loss for feature disentanglement. Single-DGOD [41] uses a cyclic-disentangled module in a CNN to decouple domain-invariant and domain-specific features with a contrastive loss, and improves object detection accuracy with a self-distillation module. Feature disentanglement has also been used in other domain-adaptive tasks, *e.g.*, semantic segmentation [40] and depth estimation [35]. Our main difference to existing works is that our feature disentanglement is done individually for each image and object token in their respective sequences with transformers, resulting in different intensities of alignment on distinct image parts. We also modify the mutual-information loss from [42] to account for the difference in channel size of our domain-invariant and domain-specific features.

3. BiADT

BiADT seeks to decompose input features of either the source domain, $\mathcal{S}$, or the target domain, $\mathcal{T}$, into domain-invariant $\mathcal{I}$ features and domain-specific $\mathcal{D}$ features, progressively layer-by-layer of its transformer-based architecture, as illustrted in Fig. 1. BiADT consists of the encoder and decoder. The encoder incorporates long-range visual information into each image token, and the decoder decodes objects on the image context and outputs predictions. As shown in Fig. 1, the feature decomposition occurs at both image level, $\boldsymbol{Y}$, and object level, $\boldsymbol{X}$, resulting in $\mathcal{I}$-features and $\mathcal{D}$-features for images from the encoder, as well as $\mathcal{I}$-features and $\mathcal{D}$-features for objects from the decoder. We make two changes in the standard transformer, as follows.

First, we design a new token-wise domain-specific embedding $\mathcal{D}$ at the image level in the encoder and at the object level in the decoder. This means that the standard content and position embedding ($\boldsymbol{c}$, $\boldsymbol{p}$) in an image (or object) token is disentangled into $\mathcal{I}$ and $\mathcal{D}$ components:($\boldsymbol{c}_\mathcal{I}$, $\boldsymbol{p}_\mathcal{I}$) and ($\boldsymbol{c}_\mathcal{D}$, $\boldsymbol{p}_\mathcal{D}$). Second, we propose two new attention modules in the transformer: (1) *DABA* – deformable-attention with bi-alignment present in the encoder and decoder; and (2) *SABA* – self-attention with bi-alignment present in the decoder only.

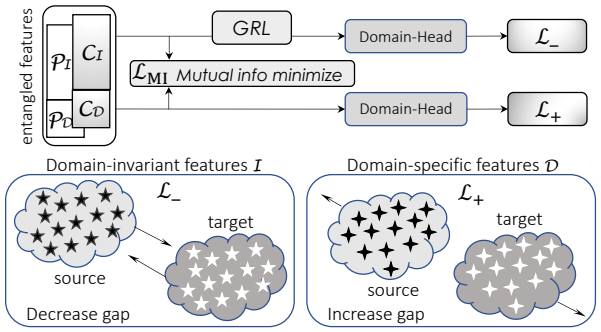

Figure 2. Three losses used for our bidirectional feature alignment. For each image/object token represented by entangled features at the input, we use a GRL (gradient reverse layer) followed by a domain head (DH) to reduce the domain gap, resulting in the domain-invariant feature $\mathcal{I}$ (bottom left). We pass the entangled features to a DH to increase the domain gap, resulting in domain-specific features $\mathcal{D}$ (bottom right). We also use a mutual information (MI) loss to further disentangle these two types of features.

As shown in Fig 2, the proposed bi-alignment is driven by minimizaing three losses, $\mathcal{L}_-$, $\mathcal{L}_+$, and mutual information $\mathcal{L}_{\text{MI}}$ between $\mathcal{I}$-features and $\mathcal{D}$-features. $\mathcal{L}_-$ is aimed at minimizing the domain gap with GRL, and $\mathcal{L}_+$ at maximizing the domain gap, using a domain discriminator (domain head). By minimizing mutual information between $\mathcal{I}$-features and $\mathcal{D}$-features, our goal is to make them independent and thus improve their disentanglement. Importantly, the three losses are used token-wise at both the image and object levels, instead of globally for the entire image and entire object sequence in [17, 38].

3.1. Review of Deformable Attention

BiADT uses Dab-Deformable-Detr [25] as a base detector. Deformable attention modules [49] in Dab-Deformable -Detr attend only to a small set of key sampling points around a reference. This enables fast convergence in training. Also, Dab-Deformable-Detr lends itself for a seamless extension with our own design. Before we introduce our DABA and SABA attention modules, we briefly review deformable attention.

For a given image token sequence $\boldsymbol{Y} \in \mathbb{R}^{N\times Z}$, with the length $N = H \cdot W$ and the number of feature channels Z, deformable attention module aggregates features from a subset of $\boldsymbol{Y}$. Specifically, for every query token $[\boldsymbol{c}_q, \boldsymbol{p}_q]$, where $\boldsymbol{c}_q \in \mathbb{R}^{1\times Z}$ is the query content embed-

ding, and $\boldsymbol{p}_q \in \mathbb{R}^{1\times Z}$ is the query positional embedding, the deformable attention is defined as:

$$\text{DeformAttn}\left(\boldsymbol{c}_q, \boldsymbol{p}_q, \boldsymbol{Y}\right) = \left[\boldsymbol{a}_q \cdot \boldsymbol{V}_q\right] \boldsymbol{W}^o \quad (1)$$

where $\boldsymbol{V} = \boldsymbol{Y}\boldsymbol{W}^v$ denotes the value embeddings of $\boldsymbol{Y}$ with the learnable projection matrix $\boldsymbol{W}^v \in \mathbb{R}^{Z\times Z^v}$. $\boldsymbol{V}_q \in \mathbb{R}^{n'\times Z^v}$ is a subset of $\boldsymbol{V}$ sampled at $n'<N$ positions at certain offsets $\boldsymbol{\Delta}_q$ from the query position, *i.e.*, $(x, y)_q+\boldsymbol{\Delta}_q$. The offsets are adaptively computed by a learnable linear projection function F^{offset} as

$$\boldsymbol{\Delta}_q = F^{\text{offset}}(\boldsymbol{c}_q + \boldsymbol{p}_q) \in \mathbb{R}^{n'\times 2}. \quad (2)$$

The attention weights $\boldsymbol{a}_q$ in Eq. (1) are estimated by another learnable linear projection function F^{atten}:

$$\boldsymbol{a}_q = F^{\text{atten}}(\boldsymbol{c}^q + \boldsymbol{p}^q) \in \mathbb{R}^{1\times n'}. \quad (3)$$

Finally, the attention weights $\boldsymbol{a}_q$ is used to aggregate features from the sampled value embeddings of $\boldsymbol{V}_q$. Also, the learnable projection matrix $\boldsymbol{W}^o \in \mathbb{R}^{Z_v\times Z}$ in Eq. (1) is used to compute the final deformable attention feature.

Transformers usually have a multi-head architecture for the above attention computing, and we have reviewed only one deformable attention head, for simplicity. Our BiADT actually uses multi-head deformable attentions, as Dab-Deformable-Detr [25].

3.2. Feature Disentanglement and Alignment

A standard token in the DETR family of transformers consists of content embedding $\boldsymbol{c}$ and position embedding $\boldsymbol{p}$, where $\boldsymbol{c} = [\boldsymbol{c}_x, \boldsymbol{c}_y] \in \mathbb{R}^{2d}$ and $\boldsymbol{p} = [\boldsymbol{p}_x, \boldsymbol{p}_y] \in \mathbb{R}^{2d}$ ($d = 128$). The indices x and y represent the corresponding two image axes along which $\boldsymbol{c}$ is estimated. As introduced earlier, we propose to extend the standard token with explicit domain-invariant and domain-specific embeddings, $\boldsymbol{c}_\mathcal{I} \in \mathbb{R}^{2d}$ and $\boldsymbol{c}_\mathcal{D} \in \mathbb{R}^{d}$ and $\boldsymbol{p}_\mathcal{I} \in \mathbb{R}^{2d}$ and $\boldsymbol{p}_\mathcal{D} \in \mathbb{R}^{d}$, resulting in:

$$[[\boldsymbol{c}_\mathcal{I}, \boldsymbol{c}_\mathcal{D}], [\boldsymbol{p}_\mathcal{I}, \boldsymbol{p}_\mathcal{D}]]. \quad (4)$$

Since the image's domain is known, we can readily assign the $\boldsymbol{p}_\mathcal{D}^\mathcal{S} = \{-1\}^d$ for the source domain $\mathcal{S}$, and $\boldsymbol{p}_\mathcal{D}^\mathcal{T} = \{1\}^d$ for the target domain $\mathcal{T}$.

Deformable Attention with Bi-Alignment. Details of our DABA and SABA modules are shown in Fig. 3. The figure also illustrates our key differences from AQT [17]. Let $\boldsymbol{Y} = [\boldsymbol{Y}_\mathcal{I}, \boldsymbol{Y}_\mathcal{D}] \in \mathbb{R}^{N\times 3d}$ denote the image token sequence, and $\boldsymbol{C} = [\boldsymbol{C}_\mathcal{I}, \boldsymbol{C}_\mathcal{D}] \in \mathbb{R}^{n\times 3d}$, $\boldsymbol{P} = [\boldsymbol{P}_\mathcal{I}, \boldsymbol{P}_\mathcal{D}] \in \mathbb{R}^{n\times 3d}$ denote the content and positional embeddings of the query token sequences. Since the deformable attention is used in both encoder and decoder, the query token sequence can be either the image token sequence $\boldsymbol{Y}$ (in the encoder) or the object token sequence $\boldsymbol{X} = [\boldsymbol{X}_\mathcal{I}, \boldsymbol{X}_\mathcal{D}]$ (in the decoder). As shown in Fig. 3 (middle), BiADT computes $\Delta\boldsymbol{C} = [\Delta\boldsymbol{C}_\mathcal{I}, \Delta\boldsymbol{C}_\mathcal{D}]$ in every DABA module.

From Fig. 3 (middle), the residual of alignable part $\Delta\boldsymbol{C}_\mathcal{I}$ can be estimated as

$$\Delta\boldsymbol{C}_\mathcal{I} = \text{DeformAttn}\left(\boldsymbol{C}_\mathcal{I}, \boldsymbol{P}_\mathcal{I}, \boldsymbol{Y}_\mathcal{I}\right) = \left[\boldsymbol{A}_\mathcal{I} \cdot \boldsymbol{V}_\mathcal{I}\right] \boldsymbol{W}_\mathcal{I}^o, \quad (5)$$

where $\boldsymbol{V}_\mathcal{I} \in \mathbb{R}^{n'\times 2d}$ is a subset of the image value embeddings $\boldsymbol{Y}_\mathcal{I}\boldsymbol{W}_\mathcal{I}^v \in \mathbb{R}^{N\times 2d}$ sampled at $n' < N$ positions at offsets $\boldsymbol{\Delta}$ from the query positions (ref to Eq.(2)); $\boldsymbol{A}_\mathcal{I} \in \mathcal{R}^{n\times n'}$ is the attention matrix between n queries and $n' < N$ sampled points given by (3); and $\boldsymbol{W}_\mathcal{I}^v, \boldsymbol{W}_\mathcal{I}^o \in \mathbb{R}^{2d\times 2d}$ are both learnable linear projection matrices.

Also from Fig. 3 (middle), the residual of domain-specific part can be estimated as

$$\Delta\boldsymbol{C}_\mathcal{D} = \text{DeformAttn}\left(\boldsymbol{C}, \boldsymbol{P}, \boldsymbol{Y}\right) = \left[\boldsymbol{A}_\mathcal{D} \cdot \boldsymbol{V}_\mathcal{D}\right] \boldsymbol{W}_\mathcal{D}^o, \quad (6)$$

where $\boldsymbol{V}_\mathcal{D} \in \mathbb{R}^{n'\times 3d}$ is a subset of the image value embedding $[\text{GRL}(\boldsymbol{Y}_\mathcal{I}), \boldsymbol{Y}_\mathcal{D}]\boldsymbol{W}_\mathcal{D}^v \in \mathbb{R}^{N\times 3d}$ with $n' < N$. $\boldsymbol{W}_\mathcal{D}^v \in \mathbb{R}^{3d\times 3d}$ and $\boldsymbol{W}_\mathcal{D}^o \in \mathbb{R}^{3d\times d}$ are both learnable linear projection matrices. $\boldsymbol{V}_\mathcal{D}$ is sampled at $n' < N$ positions at the same offsets $\boldsymbol{\Delta}$ from the query positions as in (5).

Note that GRL operates as an identity function in the feed forward computation, and reverses the gradient in backpropagation. In this way, it "fools" the domain discriminator and reduces the domain gap in $\boldsymbol{C}_\mathcal{I}$.

The domain-specific attention matrix $A_\mathcal{D}$ is as

$$A_\mathcal{D} = F_\mathcal{D}^{\text{atten}}([\text{GRL}(\boldsymbol{C}_\mathcal{I} + \boldsymbol{P}_\mathcal{I}), \boldsymbol{C}_\mathcal{D} + \boldsymbol{P}_\mathcal{D}]), \quad (7)$$

where the function $F_\mathcal{D}^{\text{atten}}$ is given by (3). With this design, we want to distill domain-specific features from $\boldsymbol{C}_\mathcal{I}$ into $\boldsymbol{C}_\mathcal{D}$. In turn, this facilitates learning of $\boldsymbol{C}_\mathcal{I}$ to be more domain invariant.

Self-Attention with Bi-Alignment. SABA module exists in the transformer decoder only, and is shown in Fig. 3 (right). Let $\boldsymbol{X} = [\boldsymbol{X}_\mathcal{I}, \boldsymbol{X}_\mathcal{D}] \in \mathbb{R}^{N\times 3d}$ denote the object token sequence, and $\hat{\boldsymbol{C}} = [\hat{\boldsymbol{C}}_\mathcal{I}, \hat{\boldsymbol{C}}_\mathcal{D}] \in \mathbb{R}^{n\times 3d}$, $\hat{\boldsymbol{P}} = [\hat{\boldsymbol{P}}_\mathcal{I}, \hat{\boldsymbol{P}}_\mathcal{D}] \in \mathbb{R}^{n\times 3d}$ denote the content and positional embeddings of the object query token sequences.

The architecture of SABA is similar to that of DABA. Since SABA aims to capture the objects relationships in the object-query sequence, it does not use deformable attention but uses a complete attention instead, as the standard self-attention in the Detr-family of transformer detectors [2, 27, 49]. Again, BiADT also attempts to compute $\Delta\hat{\boldsymbol{C}} = [\Delta\hat{\boldsymbol{C}}_\mathcal{I}, \Delta\hat{\boldsymbol{C}}_\mathcal{D}]$ in every SABA module, and splits the embedding features $[\hat{\boldsymbol{C}}_\mathcal{I}, \hat{\boldsymbol{C}}_\mathcal{D}]$ at object level.

From Fig. 3 (right), the domain-invariant self-attention residual features are computed as

$$\Delta\hat{\boldsymbol{C}}_\mathcal{I} = (\hat{\boldsymbol{A}}\hat{\boldsymbol{V}}_\mathcal{I})\hat{\boldsymbol{W}}_\mathcal{I}^o \quad (8)$$

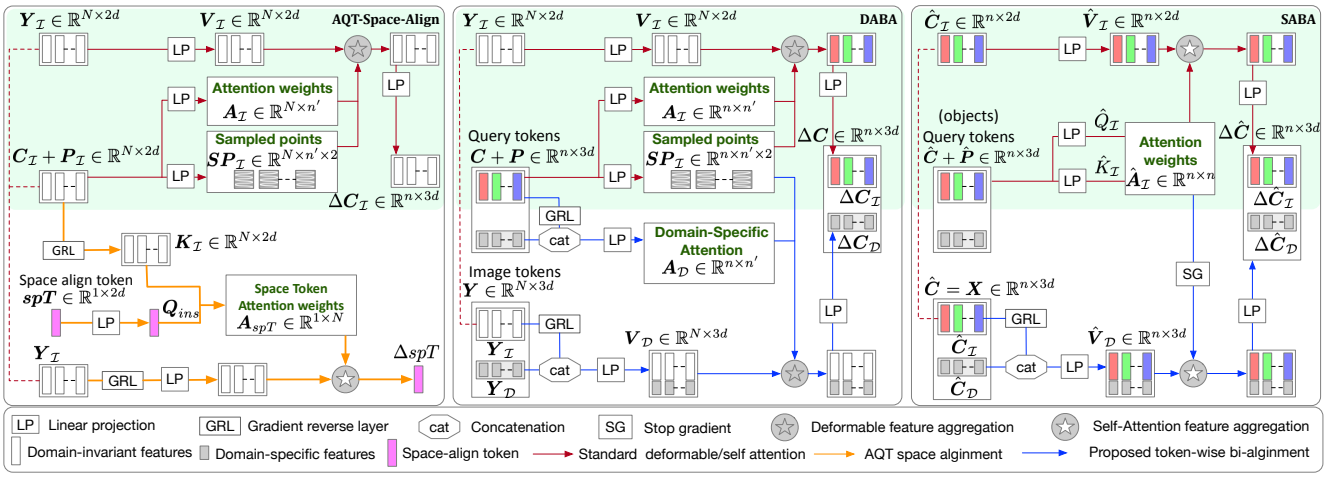

Figure 3. Differences between AQT [17] and our BiADT. (*Left*) Space-alignment in AQT aligns features over all image tokens. (*Middle*) Our DABA module is used in the encoder for self-attention and in the decoder for cross-attention. (*Right*) Our SABA module in the decoder captures inter-object relationships in the object-query sequence. Notice that the top regions (colored as light blue) are standard deformable attention and self attention, to which our proposed DABA and SABA are seamlessly integrated to align features of every token.

where $\hat{\boldsymbol{A}} = \text{softmax}\frac{\hat{\boldsymbol{Q}}_{\mathcal{I}}\hat{\boldsymbol{K}}_{\mathcal{I}}^{\top}}{\sqrt{2d}}$ is a standard self-attention matrix, and $\hat{\boldsymbol{Q}}_{\mathcal{I}}$, $\hat{\boldsymbol{K}}_{\mathcal{I}}$ are standard projected embeddings of the queries and keys the same as in [27, 49]; $\hat{\boldsymbol{V}}_{\mathcal{I}} = \hat{\boldsymbol{C}}_{\mathcal{I}}\hat{\boldsymbol{W}}_{\mathcal{I}}^{v}$ is the projected value embeddings for the domain-invariant features with $\hat{\boldsymbol{W}}_{\mathcal{I}}^{v} \in \mathbb{R}^{2d\times 2d}$ being the projection matrix. Similar to DABA, we directly use the attention matrix generated by the domain-invariant features $\hat{\boldsymbol{C}}_{\mathcal{I}}$ to both branches, so as to encourage the feature exchanges based on similarity of objects' domain-invariant characteristics.

Also from Fig. 3 (right), the domain-specific residual feature of objects is computed as:

$$\Delta\hat{\boldsymbol{C}}_{\mathcal{D}} = (\hat{\boldsymbol{A}}\hat{\boldsymbol{V}}_{\mathcal{D}})\hat{\boldsymbol{W}}_{\mathcal{D}}^{o} \tag{9}$$

with the linear projection matrix $\hat{W}_{\mathcal{D}}^{o} \in \mathbb{R}^{3d\times d}$. In (9), the domain-specific value embedding $\hat{\boldsymbol{V}}_{\mathcal{D}}$ is computed from the entire object feature as $[\text{GRL}(\hat{\boldsymbol{C}}_{\mathcal{I}}), \hat{\boldsymbol{C}}_{\mathcal{D}}] \cdot \hat{\boldsymbol{W}}_{\mathcal{D}}^{v}$, where $\hat{\boldsymbol{W}}_{\mathcal{D}}^{v} \in \mathbb{R}^{3d\times 3d}$ is the projection matrix, and GRL serves to reduce the domain gap for the domain-invariant object feature $\hat{\boldsymbol{C}}_{\mathcal{I}}$ in backprogagation.

Finally, the residual feature $\Delta\hat{\boldsymbol{C}} = [\Delta\hat{\boldsymbol{C}}_{\mathcal{I}}, \Delta\hat{\boldsymbol{C}}_{\mathcal{D}}]$ is used to update $\hat{\boldsymbol{C}}$ in each SABA module.

3.3. Loss Functions

BiADT uses multi "domain heads", GRL and mutual information constraint to optimize the image token sequence in encoder, *i.e.*, $\boldsymbol{C}^{Y} = [\boldsymbol{C}_{\mathcal{I}}^{Y}, \boldsymbol{C}_{\mathcal{D}}^{Y}] \in \mathbb{R}^{N\times 3d}$, and object token sequence in decoder, *i.e.*, $\boldsymbol{C}^{X} = [\boldsymbol{C}_{\mathcal{I}}^{X}, \boldsymbol{C}_{\mathcal{D}}^{X}] \in \mathbb{R}^{n\times 3d}$. Driven by such multiple forces, the features at both image and object level can be well decomposed.

DyHinge loss – Existing adversarial learning methods usually make use of BCE (binary cross-entropy) and GRL to reduce the domain gap in features. For example, AQT uses BCE on the domain label prediction of the three alignment tokens. Similarly, we apply BCE loss on the $\boldsymbol{C}_{\mathcal{D}}$ branch that aims to absorb domain-specific features and in turn reduce the domain gap in the domain-invariant features. In some cases, we notice it is hard to distinguish the domain from certain local area of the image, *e.g.* the middle part of the road in Fig. 5. Hence, applying BCE loss on $\boldsymbol{C}_{\mathcal{I}}$ branch may be too strict for certain tokens, causing an excessive alignment for some of the the domain-invariant features in $\boldsymbol{C}_{\mathcal{I}}$ branch. We propose a more relaxed dynamic hinge loss:

$$\mathcal{L}_{\text{DyHinge}}(z) = \begin{cases} z, & \text{if } z_{GT} = 0 \\ \max\{0, m - z\}, & \text{if } z_{GT} = 1 \end{cases} \tag{10}$$

where $m = \max(F_{\mathcal{D}}^{Y}(\boldsymbol{C}_{\mathcal{D}}^{Y})^{\mathcal{SG}}, 0.5)$ is the dynamic margin. It is derived from the domain confidence score of the $\boldsymbol{C}_{\mathcal{D}}$ branch (see 12, 14) representing the intensity of "domain shift". A smaller value of m thus indicates milder alignment on the corresponding $\boldsymbol{C}_{\mathcal{I}}$ feature and vice versa. $\mathcal{SG}$ is the stop-gradient operation to optimize the two branches independently. z_{GT} is the ground truth, 0 for source and 1 for target. Please note when the domain label prediction on $\boldsymbol{C}_{\mathcal{D}}$ branch is as confident as 1, the above DyHinge loss becomes standard hinge loss with margin of 1.

Losses for Bi-Alignments. As shown in the bottom part of Fig. 2, we define four distinct losses for supervising the domain invariant and specific parts in each token sequence. As shown in the equations below, $F_{\mathcal{I}}$ and $F_{\mathcal{D}}$ denote the domain heads implemented as MLP layers. GT is the ground truth of the domain class, GT=0 for source and GT=1 for target domains. L_{BCE} is the binary cross-entropy loss, and

L_{DyHinge} is the hinge loss. The "+/-" signs denote increase or decrease the domain gap in respective features.

$$\mathcal{L}_{\text{-}}^{Y} = \mathcal{L}_{\text{DyHinge}}(F_{\mathcal{I}}^{Y}(\text{GRL}(\boldsymbol{C}_{\mathcal{I}}^{Y})), \text{GT}), \tag{11}$$

$$\mathcal{L}_{\text{+}}^{Y} = \mathcal{L}_{\text{BCE}}(F_{\mathcal{D}}^{Y}(\boldsymbol{C}_{\mathcal{D}}^{Y}), \text{GT})), \tag{12}$$

$$\mathcal{L}_{\text{-}}^{X} = \mathcal{L}_{\text{DyHinge}}(F_{\mathcal{I}}^{X}(\text{GRL}(\boldsymbol{C}_{\mathcal{I}}^{X})), \text{GT}), \tag{13}$$

$$\mathcal{L}_{\text{+}}^{X} = \mathcal{L}_{\text{BCE}}(F_{\mathcal{D}}^{X}(\boldsymbol{C}_{\mathcal{D}}^{X}), \text{GT})), \tag{14}$$

The overall adaptation loss, $\mathcal{L}_{\text{da}}$, is the summation of the two bi-alignment losses at X and Y:

$$\mathcal{L}_{\text{da}} = (\lambda_{\text{+}}^{Y}\mathcal{L}_{\text{+}}^{Y} + \lambda_{\text{-}}^{Y}\mathcal{L}_{\text{-}}^{Y}) + (\lambda_{\text{+}}^{X}\mathcal{L}_{\text{+}}^{X} + \lambda_{\text{-}}^{X}\mathcal{L}_{\text{-}}^{X}) \tag{15}$$

where $\lambda_{\text{+}}^{Y}, \lambda_{\text{-}}^{Y}, \lambda_{\text{+}}^{X}$ and $\lambda_{\text{-}}^{X}$ are four positive coefficients.

Mutual Information Minimization. Along with the bi-alignment losses previously defined, we also minimize the mutual information (MI) loss between $\boldsymbol{C}_{\mathcal{I}}$ features and $\boldsymbol{C}_{\mathcal{D}}$ features, making them more independent. We follow [42] and use the Mutual Information Neural Estimator (MINE) [1] to compute the MI loss based on the Monte-Carlo integration [31]:

$$\mathcal{L}_{\text{MI}}(\boldsymbol{C}_{\mathcal{I}}, \boldsymbol{C}_{\mathcal{D}}) = \frac{1}{n}\sum_{j=1}^{n} T_{\theta}(\boldsymbol{c}_{\mathcal{I}}, \boldsymbol{c}_{\mathcal{D}}) - \log(\frac{1}{n}\sum_{j=1}^{n} e^{T_{\theta}(\boldsymbol{c}_{\mathcal{I}}, \boldsymbol{c}'_{\mathcal{D}})}) \tag{16}$$

where $(\boldsymbol{c}_{\mathcal{I}}, \boldsymbol{c}_{\mathcal{D}})$ is sampled from the joint distribution of the image token sequence or the object token sequence, and $\boldsymbol{c}'_{\mathcal{D}}$ is sampled from the marginal distribution. T_{θ} is a network consisting of three fully-connected layers.

Total Loss. The final training objective is defined as:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{da}} + \lambda_{\text{MI}}\mathcal{L}_{\text{MI}} \tag{17}$$

where $\mathcal{L}_{\text{det}}$ is the detection loss from Deformable-Detr [49], $\mathcal{L}_{\text{da}}$ is the domain bi-alignment loss given by Eq.(15), and $\mathcal{L}_{\text{MI}}$ is the mutual information constraint loss.

3.4. Differences from AQT

Differences between our BiADT and AQT [17] – our strong baseline that achieves competitive performance – are illustrated in Fig. 3. The key differences include: (1) AQT uses $2d$-dim features $\boldsymbol{c}_{\mathcal{I}}$ to represent domain-invariant image/object features, and we create additional d-dim features $\boldsymbol{c}_{\mathcal{D}}$ associated to each of them to represent its domain-specific feature. (2) AQT uses three alignment queries (also known as adversarial tokens), *i.e.*, spatial, channel and instance queries, to reduce the domain gap via extra attention modules. The domain specific patterns in the overall token sequences are loosely captured by such three adversarial tokens without considering the individual difference. In contrast, our bi-alignments can be seamlessly integrated into the standard attention layers, so every image/object token is explicitly split into the domain-invariant and domain-specific parts, resulting more fine-grained token-wise domain alignment for object detection. BiADT has comparable model complexity to AQT. The overall number of parameters of inference in BiADT model is 45.4M, whereas AQT is 46.6M. As mentioned earlier, the three AQT alignments can be readily integrated in our model as well.

4. Experimental Results

Datasets and Metrics. Evaluation is conducted on public benchmark datasets with significant domain gaps. For each dataset, we follow the same protocol of existing works [17, 13, 23], and report the average precision (AP_{50}) of each class and the mean AP over all classes for object detection.

Cityscapes→FoggyCityscapes **(C2F)**: Cityscapes [6] data set consists of 2,975 training images and 500 validation images of urban street scenes under normal weather conditions from 50 cities. Foggycityscapes [34] is synthesized by adding artificial fog to the Cityscapes images. So, they have the same training-validation split. The Cityscapes training set and the unlabeled Foggycityscapes training set are used for training and the validation set of FoggyCityscapes is used for evaluation. In this setting, the domain adaption is from normal to foggy weather conditions.

Sim10K→Cityscapes **(S2C)**: Sim10K[19] data set consists of 10,000 synthetic images, used as the source dataset. The target dataset is Ciytscapes. Following AQT [17], only the object "car" is considered. In this setting, the domain adaption is from synthetic to real world images.

Cityscapes→BDD100k daytime **(C2B)**: BBD100k [47] is a large-scale, ego-centric dataset. Its daytime subset includes 36,278 and 5,258 images for training and validation, respectively. In this setting, the Cityscapes training set is used as a smaller source domain, and the daytime subset of BBD100K is used as a large unlabeled target set. This setting evaluates our domain adaptation between two datasets with different visual attributes and different dataset sizes. Following existing methods [17, 3], we report results on the seven object classes shared by both datasets.

Implementation Details. Our object detector is Dab-Deformable-Detr [25] with ResNet50 [12] as backbone pre-trained on ImageNet [7]. It has 6 encoder layers and 6 decoder layers. In training, we use the same detection loss as in [25]. In the C2F setting defined above, we set $\lambda_{\text{+}}^{Y}$ / $\lambda_{\text{+}}^{X}$ to 10^{-2}; and $\lambda_{\text{-}}^{Y}$ / $\lambda_{\text{-}}^{X}$ to 10^{-1}. In the other two settings, S2C and C2B, we set $\lambda_{\text{+}}^{Y}$ / $\lambda_{\text{+}}^{X}$ to 10^{-2} and 10^{-5}; and $\lambda_{\text{-}}^{Y}$ / $\lambda_{\text{-}}^{X}$ to 10^{-1} and 10^{-4}. λ_{MI} is set to 5e-5. The learning rate for the backbone and the transformer are set to 2e-5 and 2e-4, respectively, and the batch size is set to 16. The learning rate decay is set to 0.1, and applied after 40 epochs for the 50-epoch training schedule.

Method	Backbone	Detector	Pseudo-Label	person	rider	car	truck	bus	train	motor	bike	mAP
MTTrans [18] *ECCV'22*	R50	Deform-Detr	Yes	47.7	49.9	65.2	25.8	45.9	33.8	32.6	46.5	43.4
PT [3] *ICML'22*	V16	Faster R-CNN	Yes	43.2	52.4	63.4	33.4	56.6	37.8	41.3	48.7	47.1
TDD [13] *CVPR'22*	R50	Faster R-CNN	Yes	50.7	53.7	68.2	35.1	53.0	45.1	38.9	49.1	49.2
AT [23] *CVPR'22*	V16	Faster R-CNN	Yes	45.5	55.1	64.2	35.0	56.3	54.3	38.5	51.9	50.9
AT* [23] *CVPR'22*	V16	Faster R-CNN	Yes	44.1	54.2	62.7	33.6	54.4	51.9	39.2	49.2	49.5
PDN [42] *TPAMI'21*	R101	Faster R-CNN	No	32.8	44.4	49.6	33.0	46.1	38.0	29.9	35.3	38.6
ICCR-VDD [43] *ICCV'21*	R50	Faster R-CNN	No	33.4	44.0	51.7	33.9	52.0	34.7	34.2	36.8	40.0
SFA [38] *ACM MM'21*	R50	Deform-Detr	No	46.5	48.6.	62.6	25.1	46.2	29.4	28.3	44.0	41.3
MGADA [48] *CVPR'22*	R101	FCOS	No	43.1	47.3	61.5	30.2	53.2	50.3	27.9	36.9	43.8
SIGMA [22] *CVPR'22*	R50	FCOS	No	44.0	43.9	60.3	31.6	50.4	51.5	31.7	40.6	44.2
O2Net [9] *ACM MM'22*	R50	Deform-Detr	No	48.7	51.5	63.6	31.1	47.6	47.8	38.0	45.9	46.8
AQT [17] *IJCAI'22*	R50	Deform-Detr	No	49.3	52.3	64.4	27.7	53.7	46.5	36.0	46.4	47.1
AQT* [17] *IJCAI'22*	R50	DAB-Deform-Detr	No	49.8	54.2	65.8	29.0	56.2	37.5	38.9	48.2	47.4
BiADT	R50	DAB-Deform-Detr	No	50.3	56.4	66.5	32.5	52.3	47.8	40.1	48.3	49.3
BiADT+AQT	R50	DAB-Deform-Detr	No	50.1	55.4	67.9	31.5	56.1	46.8	38.6	49.3	49.6
BiADT+TS	R50	DAB-Deform-Detr	Yes	52.2	58.9	69.2	31.7	55.0	45.1	42.6	51.3	50.8

Table 1. Comparison to the SOTA in the Cityscapes→FoggyCityscapes setting. The symbol * denotes our results using the official Github repository. For fair comparison, our BiADT is also trained for 50 epochs the same as AQT [17].

Method	Backbone	Detector	P-L	car mAP
PT[3]	V16	Faster R-CNN	Yes	55.1
MTTrans[18]	R50	Deform-Detr	Yes	57.9
TDD [13]	V16	Faster R-CNN	Yes	63.3
MGADA [48]	R101	FCOS	No	54.1
SFA [38]	R50	Deform-Detr	No	52.6
SIGMA [22]	R50	FCOS	No	53.7
AQT [17]	R50	Deform-Detr	No	53.4
AQT* [17]	R50	DAB-Deform-Detr	No	53.7
O2Net [9]	R50	Deform-Detr	No	54.1
BiADT	R50	DAB-Deform-Detr	No	55.8
BiADT + AQT	R50	DAB-Deform-Detr	No	56.6

Table 2. Comparison to the SOTA in the Sim10K→Cityscapes.

4.1. Comparison with the State-of-the-art

Tables 1, 2, and 3 compare our BiADT to the state of the art methods (SOTA) in the three domain-adaptation settings – C2F, S2C, and C2B. The SOTA uses the following detectors: Faster R-CNN [32], FCOS[36], Deformable-Detr[49], and Dab-Deformable-Detr[25]. As for AQT, other than the accuracy reported in AQT [17] using Deformable-Detr, we also compare with an accuracy obtained by our own using the official AQT github repository with using Dab-Deformable-Detr as the detector for a fair comparison, named AQT*. As explained in Section. 3.4, we can add the three AQT alignments to our BiADT easily, which gives the approach called BiADT+AQT. Finally, when we integrate BiADT into the teacher-student learning framework (*i.e.*, TSST) with strong data augmentation, we get the approach called BiADT+TS.

Table 1 shows the experimental results of the C2F setting, our BiADT achieves 49.3 mAP, which outperforms most of the recent works, even the ones with using pseudo-label. It only slightly lower than AT [23] (mAP=50.9, top-1 in literature to our best knowledge, our reproduction of AT is mAP=49.5), but if combining with pseudo-label, our BiADT+TS also achieves comparable accuracy at mAP=50.8. Compare to AQT or AQT*, both are not using pseudo-label, our BiADT achieves a higher mAP with a clear margin $>= 1.9$. Besides, adding the AQT alignments to BiADT further improves this margin to 2.2. For the S2C and the C2B settings, Tables 2 and 3 show that our BiADT and BiADT+AQT give the best results in comparison to the SOTA methods without using pseudo-labeling in training.

4.2. Ablations and Detectors

Ablations. Table 4 evaluates the contribution of each proposed module in BiADT. The experiment is based on the C2F setting. In the first row, all of the proposed domain alignments are disabled except for the backbone alignment. Such a most basic version gives the mAP at 38.3. Adding the object level domain alignments (rows 2-4 in Table 4) improves mAP to 41.2, 42.3 and 42.9, indicating that the proposed object alignments indeed reduce the domain gap in the object domain-invariant features. Adding the image-level alignments (rows 5-7 in Table 4) gives better results than using only the object-level alignments, improves mAP to 43.4, 43.3 and 44.4, respectively. This suggests that the image alignments are more important for cross-domain detection. While the domain-specific and domain-invariant features are further disentangled with the proposed DyHinge loss and the mutual information loss (MI), BiADT achieves the best mAP to 49.3, indicating all the designs are not overlap but complement to each other.

Transformer Detectors. Table 5 compares the accuracy numbers using Deformable-Detr, Dab-Deformable-Detr, or Conditional-Detr as the base object detector. For both AQT and BiADT, Dab-Deformable-Detr gives the best results. This could be explained that Dab-Deformable-Detr uses dynamic anchor boxes to learn the object query position. Both AQT and BiADT perform worse using Conditional-Detr,

Method	Backbone	Detector	Pseudo-Label	person	rider	car	truck	bus	motor	bike	mAP
MTTrans [18] *ECCV'22*	R50	Deform-Detr	Yes	44.1	30.1	61.5	25.1	26.9	17.7	23.0	32.6
PT [3] *ICML'22*	V16	Faster R-CNN	Yes	-	-	-	-	-	-	-	34.9
TDD [13] *CVPR'22*	R50	Faster R-CNN	Yes	57.9	47.4	74.5	31.5	27.5	32.0	36.5	43.9
ICR-CCR [45] *CVPR'20*	V16	Faster R-CNN	No	31.4	31.3	46.3	19.5	18.9	17.3	23.8	26.9
SFA [38] *ACM MM'21*	R50	Deform-Detr	No	40.2	27.6	57.5	19.1	23.4	15.4	19.2	28.9
AQT [17] *IJCAI'22*	R50	Deform-Detr	No	38.2	33.0	58.4	17.3	18.4	16.9	23.5	29.4
AQT* [17] *IJCAI'22*	R50	DAB-Deform-Detr	No	39.5	33.2	58.5	17.8	18.3	17.5	23.3	30.3
O2Net [9] *ACM MM'22*	R50	Deform-Detr	No	40.4	31.2	58.6	20.4	25.0	14.9	22.7	30.5
BiADT	R50	DAB-Deform-Detr	No	42.0	34.5	59.9	17.2	19.2	17.8	24.4	32.7
BiADT+AQT	R50	DAB-Deform-Detr	No	42.1	34.0	60.9	17.4	19.5	18.2	25.7	33.6

Table 3. Comparison to the SOTA in the Cityscapes→BDD100k daytime setting. (please see the caption of Tab. 1 for *)

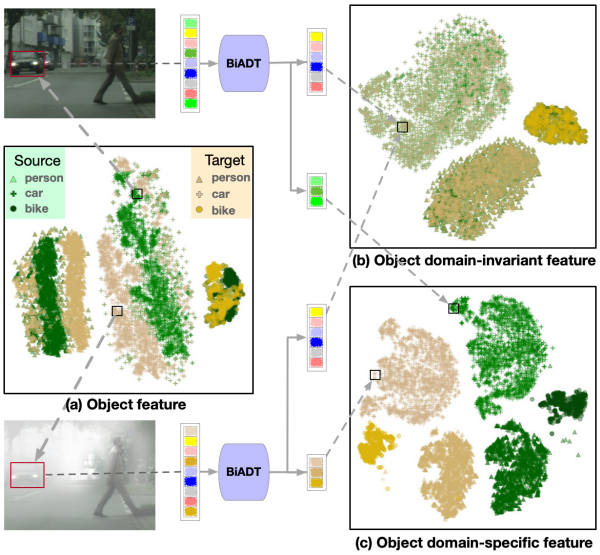

Figure 4. The t-SNE visualization of object features that belong to three object classes, *i.e.*, person, car, and bike, of the test images in the Cityscapes (source – dark greens) and FoggyCityscapes (target – dark yellows) datasets.

Row	obj $X+$	obj $X-$	img $Y+$	img $Y-$	DyHinge	MI	mAP
1							38.3
2	✓						41.2
3		✓					42.3
4	✓	✓					42.9
5			✓				43.4
6				✓			43.3
7			✓	✓			44.4
8	✓	✓	✓	✓			48.2
9	✓	✓	✓	✓	✓		48.7
10	✓	✓	✓	✓	✓	✓	49.3

Table 4. Ablations of the proposed components in BiADT for the Cityscapes→FoggyCityscapes setting.

i.e., a single scale transformer-based detector, than the other two multi-scale detectors.

4.3. Visualizations

Object feature distribution: As the feature disentanglement shown in Fig. 4, the separated $\mathcal{I}$ features in (b) are well aligned cross two domains. In contrast, the $\mathcal{D}$ in (c) features show a clear margin for the two domains between domain-specific features. Besides, we observe that the category-wise properties are preserved quite well in the resultant feature space.

Detector	Method	mAP	Method	mAP
Conditional-Detr	AQT	25.4	BiADT	27.8
Deform-Detr	AQT	47.1	BiADT	48.9
Dab-Deform-Detr	AQT	47.4	BiADT	49.3

Table 5. The effect of using different transformer-based detectors for the Cityscapes →FoggyCityscapes setting.

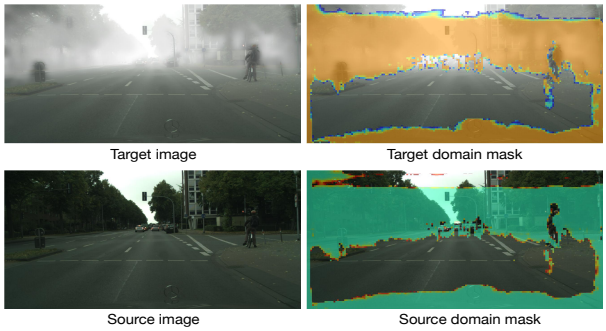

Figure 5. An example of the predicted domain mask by BiADT. The top row shows an image from the target domain and the predicted domain mask (colored in yellow). The bottom row shows the corresponding image from the source domain and the predicted domain mask (colored in green). The predicted domain mask correctly identifies the cause of the domain gap – the foggy region.

Domain prediction: Fig. 5 shows example test images from two domains and their corresponding domain masks predicted by BiADT from the encoder. As we can see, the target domain mask (top right) is very similar to the source domain mask (bottom right), and these masked regions indeed indicate the cause of the domain gap – the fog. The middle part of the road is correctly predicted as being more domain-invariant because of less fog. This also indicates that the encoder feature disentanglement is more challenging, as it operates on the image token sequence, where most of the tokens from background may lack domain-specific characteristics (e.g., tokens related to "road" in Fig. 5).

5. Conclusion

We have specified BiADT for cross-domain object detection. Our key contributions include: (1) decomposing the image token and object token into their respective domain-invariant features and domain-specific features; (2) deformable attention bi-alignment and self-attention bi-alignment, in which their corresponding domain-specific features are learned by attending all of the context in both image and object token sequences, resulting in the reduced domain gap in the domain-invariant features. Our BiADT significantly outperforms the strong baseline model AQT, and achieves very competitive performance to state-of-the-art on multiple benchmark domain shift scenarios. In comparison to pseudo-label methods that usually require complex multistage training, our training is just one stage, giving a superior model than some latest pseudo-label approaches on the multiple cross-domain datasets. Moreover, our experiments show that BiADT also can be trained with the more complex training procedures of pseudo-label approaches, giving further performance improvements.

For limitations, relative to AQT, BiADT uses additional features to represent domain-specific characteristics of the image/object tokens. However, BiADT has the same complexity as AQT in performing object detection. As any vision system, ours could be misused for malicious human monitoring and violations of privacy.

6. Acknowledgement

This work has been supported in part by Amazon internship and USDA NIFA award No.2021-67021-35344 (AgAID AI Institute).

References

[1] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018. 6

[2] 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*, pages 213–229. Springer, 2020. 1, 2, 4

[3] Meilin Chen, Weijie Chen, Shicai Yang, Jie Song, Xinchao Wang, Lei Zhang, Yunfeng Yan, Donglian Qi, Yueting Zhuang, Di Xie, et al. Learning domain adaptive object detection with probabilistic teacher. *arXiv preprint arXiv:2206.06293*, 2022. 3, 6, 7, 8

[4] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 1, 2

[5] Li Congcong, Du Dawei, Zhang Libo, Wen Longyin, Luo Tiejian, Wu Yanjun, and Zhu Pengfei. Spatial attention pyramid network for unsupervised domain adaptation. In *ECCV*, 2020. 3

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[8] Tzeng Eric, Hoffman Judy, Saenko Kate, and Darrell Trevor. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1

[9] Kaixiong Gong, Shuang Li, Shugang Li, Rui Zhang, Chi Harold Liu, and Qiang Chen. Improving transferability for domain adaptive detection transformers. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1543–1551, 2022. 3, 7, 8

[10] Chen Haoqi, Li Jiongcheng, Zheng Zebiao, Huang Yue, Ding Xinghao, and Yu Yizhou. Dual bipartite graph learning: A general approach for domain adaptive object detection. In *ICCV*, 2021. 3

[11] Chen Haoqi, Zheng Zebiao, Ding Xinghao, Huang Yue, and Dou Qi. Harmonizing transferability and discriminability for adapting object detectors. In *CVPR*, 2020. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[13] Mengzhe He, Yali Wang, Jiaxi Wu, Yiru Wang, Hanqing Li, Bo Li, Weihao Gan, Wei Wu, and Yu Qiao. Cross domain object detection by target-perceived dual branch distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9570–9580, 2022. 2, 3, 6, 7, 8

[14] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6668–6677, 2019. 2, 3

[15] Yan Hongliang, Ding Yukang, Li Peihua, Wang Qilong, Xu Yong, and Zuo Wangmeng. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, 2017. 1

[16] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *European Conference on Computer Vision*, pages 733–748. Springer, 2020. 2, 3

[17] Wei-Jie Huang, Yu-Lin Lu, Shih-Yao Lin, Yusheng Xie, and Yen-Yu Lin. Aqt: Adversarial query transformers for domain adaptive object detection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. 2, 3, 4, 5, 6, 7, 8

[18] Yu Jinze, Liu Jiaming, Wei Xiaobao, Zhou Haoyi, Nakata Yohei, Gudovskiy Denis, Okuno Tomoyuki, Li Jianxin,

Keutzer Kurt, and Zhang Shanghang. Mttrans: Cross-domain object detection with mean-teacher transformer. In *ECCV*, 2022. 2, 3, 7, 8

[19] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016. 6

[20] Jiang Junguang, Chen Baixu, Wang Jianmin, and Long Mingsheng. Decoupled aaaptation for cross domain object detection. In *ICLR*, 2022. 3

[21] Shuai Li, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Category dictionary guided unsupervised domain adaptation for object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1949–1957, 2021. 2

[22] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5300, 2022. 7

[23] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7581–7590, 2022. 2, 3, 6, 7

[24] Zhao Liang and Wang Limin. Task-specific inconsistency alignment for domain adaptive object detection. In *CVPR*, 2022. 3

[25] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022. 1, 2, 3, 4, 6, 7

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *Proceedings of the European conference on computer vision (ECCV)*, pages 21–37, 2016. 1

[27] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. *arXiv preprint arXiv:2108.06152*, 2021. 2, 4, 5

[28] Xu Minghao, Wang Hang, Ni Bingbing, Tian Qi, and Zhang Wenjun. Cross-domain detection via graph-induced prototype alignment. In *CVPR*, 2020. 3

[29] Long Mingsheng, Cao Yue, Wang Jianmin, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1

[30] Long Mingsheng, Cao Zhangjie, Wang Jianmin, and I. Jordan Michael. Conditional adversarial domain adaptation. In *NIPS*, 2018. 1

[31] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pages 5102–5112. PMLR, 2019. 6

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 7

[33] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 2, 3

[34] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 6

[35] Lo Shao-Yuan, Wang Wei, Thomas Jim, Zheng Jingjing, M. Patel Vishal, and Kuo Cheng-Hao. Learning feature decomposition for domain adaptive monocular depth estimation. In *IROS*, 2022. 3

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 7

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[38] Wen Wang, Cao Yang, Zhang Jing, He Fengxiang, Zha Zheng-jun, Wen Yonggang, and Tao Dacheng. Exploring sequence feature alignment for domain adaptive detection transformers. In *ACM MultiMedia*, 2021. 3, 7, 8

[39] Yu Wang, Rui Zhang, Shuo Zhang, Miao Li, Yangyang Xia, XiShan Zhang, and ShaoLi Liu. Domain-specific suppression for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9603–9612, 2021. 2, 3

[40] Chang Wei-Lun, Wang Hui-Po, Peng Wen-Hsiao, and Chiu Wei-Chen. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *CVPR*, 2019. 3

[41] Aming Wu and Cheng Deng. 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*, pages 847–856, 2022. 3

[42] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3, 6, 7

[43] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. Vector-decomposed disentanglement for domain-invariant object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9342–9351, 2021. 3, 7

[44] Li Wuyang, Liu Xinyu, Yao Xiwen, and Yuan Yixuan. Scan: Cross domain object detection with semantic conditioned adaptation. In *AAAI*, 2022. 3

[45] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adap-

tive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020. 2, 3, 8

[46] Ganin Yaroslav and S. Lempitsky Victor. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 1

[47] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 6

[48] Wenzhang Zhou, Dawei Du, Libo Zhang, Tiejian Luo, and Yanjun Wu. Multi-granularity alignment domain adaptation for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9581–9590, 2022. 7

[49] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2021. 1, 2, 3, 4, 5, 6, 7