AQT: Adversarial Query Transformers for Domain Adaptive Object Detection

Wei-Jie Huang[1] , **Yu-Lin Lu**[1] , **Shih-Yao Lin**[2*] , **Yusheng Xie**[3†] and **Yen-Yu Lin**[1,4]
[1]National Yang Ming Chiao Tung University
[2]Sony Corporation of America
[3]Amazon
[4]Academia Sinica

Abstract

Adversarial feature alignment is widely used in domain adaptive object detection. Despite the effectiveness on CNN-based detectors, its applicability to transformer-based detectors is less studied. In this paper, we present AQT (adversarial query transformers) to integrate adversarial feature alignment into detection transformers. The generator is a detection transformer which yields a sequence of feature tokens, and the discriminator consists of a novel *adversarial token* and a stack of cross-attention layers. The cross-attention layers take the adversarial token as the *query* and the feature tokens from the generator as the *key-value* pairs. Through adversarial learning, the adversarial token in the discriminator attends to the domain-specific feature tokens, while the generator produces domain-invariant features, especially on the attended tokens, hence realizing adversarial feature alignment on transformers. Thorough experiments over several domain adaptive object detection benchmarks demonstrate that our approach performs favorably against the state-of-the-art methods. Source code is available at https://github.com/weii41392/AQT.

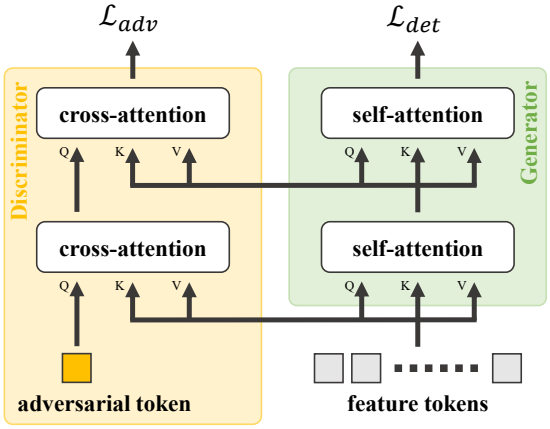


Figure 1: Our adversarial feature alignment for detection transformers. The discriminator in adversarial learning consists of a learnable adversarial token and cross-attention layers, while the generator is a detection transformer. For each cross-attention layer, the adversarial token serves as the *query* (Q) and the feature tokens as the *key-value* pair (K and V). To minimize the adversarial loss $\mathcal{L}_{adv}$, the adversarial token attends to the feature tokens discriminative for domain classification. On the contrary, the generator is forced to eliminate the domain-specific features to maximize $\mathcal{L}_{adv}$. As constrained by the detection loss $\mathcal{L}_{det}$, the generator prevents semantic collapse in the feature tokens in the meantime. Adversarial feature alignment is thus carried out.

1 Introduction

Object detection is active in computer vision and artificial intelligence researches because it is essential to a broad range of real-world applications, such as surveillance and self-driving cars. While the latest object detection methods [Ren *et al.*, 2015; Liu *et al.*, 2016; Tian *et al.*, 2019] have shown great success in several challenging benchmarks [Lin *et al.*, 2014], their capabilities largely rely on massive labeled data. More importantly, applying pretrained object detectors to new environments would result in performance degradation due to the distribution mismatch between training data and deployed environments.

Unsupervised domain adaptation (UDA) [Ganin and Lempitsky, 2015; Tzeng *et al.*, 2017] has been developed to mitigate these issues. In UDA, a model is usually trained with data from a *source* domain and a similar but different *target* domain, while the data labels are only available in the source domain. Under the covariate shift assumption, domain adaptation is typically carried out by minimizing cross-domain discrepancy so that the model supervised on the source domain can learn knowledge and representations generalizable to the target domain. Following the established practices for classification [Tzeng *et al.*, 2017] and segmentation [Hoffman *et al.*, 2018] tasks, many domain adaptive object detectors [Saito *et al.*, 2019; Chen *et al.*, 2020; Rezaeianaran *et al.*, 2021] adopt *adversarial feature alignment*. By introducing a domain classifier, a learnable measure of domain shifts can be used to impose minimax objectives and to force domain invariance. The resultant models tend to be free from biases towards the source domain.

*Work done outside of Sony
†Work done outside of Amazon

Despite its effectiveness on CNN-based detectors, adversarial feature alignment is less studied on transformer-based detectors [Carion *et al.*, 2020; Zhu *et al.*, 2021]. CNNs and transformers are intrinsically different. CNNs capture visual characteristics in features maps via convolutions, while transformers model token-wise relationships via the attention mechanism. The recent studies [Wang *et al.*, 2021a] also show that adversarial feature alignment on the CNN backbone of detection transformers brings only limited improvements for domain adaptation.

To address this issue, we present *AQT (adversarial query transformer)* that combines adversarial learning and transformers for domain adaptive object detection. Since token-wise operations are implemented in transformers, it is logical to conduct token-wise feature alignment and to focus on the domain-specific tokens. Based on this observation, we develop a strategy for the proposed AQT. We illustrate its conceptual workflow in Figure 1. The generator is the original detection transformer (denoted by self-attention layers for simplicity). It extracts a sequence of feature tokens from the input image and further detects the objects from it. The discriminator is a learnable *adversarial token* and a stack of cross-attention layers. In each cross-attention layer, the adversarial token serves as the query and the intermediate feature tokens from the generator as the key-value pair. To minimize the adversarial loss, the adversarial token tends to attend to the domain-specific feature tokens. On the contrary, the generator is forced to generate domain-invariant features, especially for the attended key-value pairs, to maximize the loss. As constrained by the detection loss $\mathcal{L}_{det}$, the generator also has to prevent semantic collapse in the feature tokens. Adversarial feature alignment is thus carried out.

The proposed mechanism is flexible. First, it is a *plug-and-adapt* module and can work with many existing transformer-based detectors [Carion *et al.*, 2020; Zhu *et al.*, 2021]. Second, it can adversarially align features of different levels. Feature tokens in Figure 1 can correspond to patches, feature maps, or detected objects. Namely, our AQT can realize space-, channel-, and instance-level feature alignment.

The main contribution of this work is three-fold. First, we propose a novel approach AQT, which integrates adversarial feature alignment into a detection transformer via an adversarial token to identify the feature tokens hard to align at that moment. Second, the proposed AQT is simple and flexible. It can work with many existing transformer-based detectors and align features of diverse levels in a unified way. Third, the proposed AQT performs favorably against state-of-the-art methods on the benchmarks, including *Cityscapes* [Cordts *et al.*, 2016] *to Foggy Cityscapes* [Sakaridis *et al.*, 2018] and *Sim10k* [Johnson-Roberson *et al.*, 2017] *to Cityscapes*.

2 Related Work

2.1 Object Detection

As a fundamental topic in computer vision, object detection has been actively studied for decades. Recent advances in object detection can be mainly attributed to CNNs, and categorized by whether region-of-interest proposals are extracted (two-stage) or not (one-stage). While two-stage detectors [Ren *et al.*, 2015] are considered to be more accurate, one-stage detectors [Liu *et al.*, 2016] are benefited from their simple structure and can perform faster.

Different from CNN-based detectors, transformer-based detectors [Carion *et al.*, 2020; Zhu *et al.*, 2021] explore token-wise dependencies for context modeling and eliminate the need for many hand-crafted components, such as anchor generation and non-maximum suppression. DETR [Carion *et al.*, 2020] first introduces transformers to object detection and yields competitive performance. To mitigate the slow convergence and prohibitive memory usage of DETR, Deformable DETR [Zhu *et al.*, 2021] adopts a learnable sparse attention to speed up convergence and to process multi-scale feature maps efficiently. While these methods focus on supervised learning, we aim at generalizing a detection transformer to the unlabeled target domain by utilizing cross-domain data.

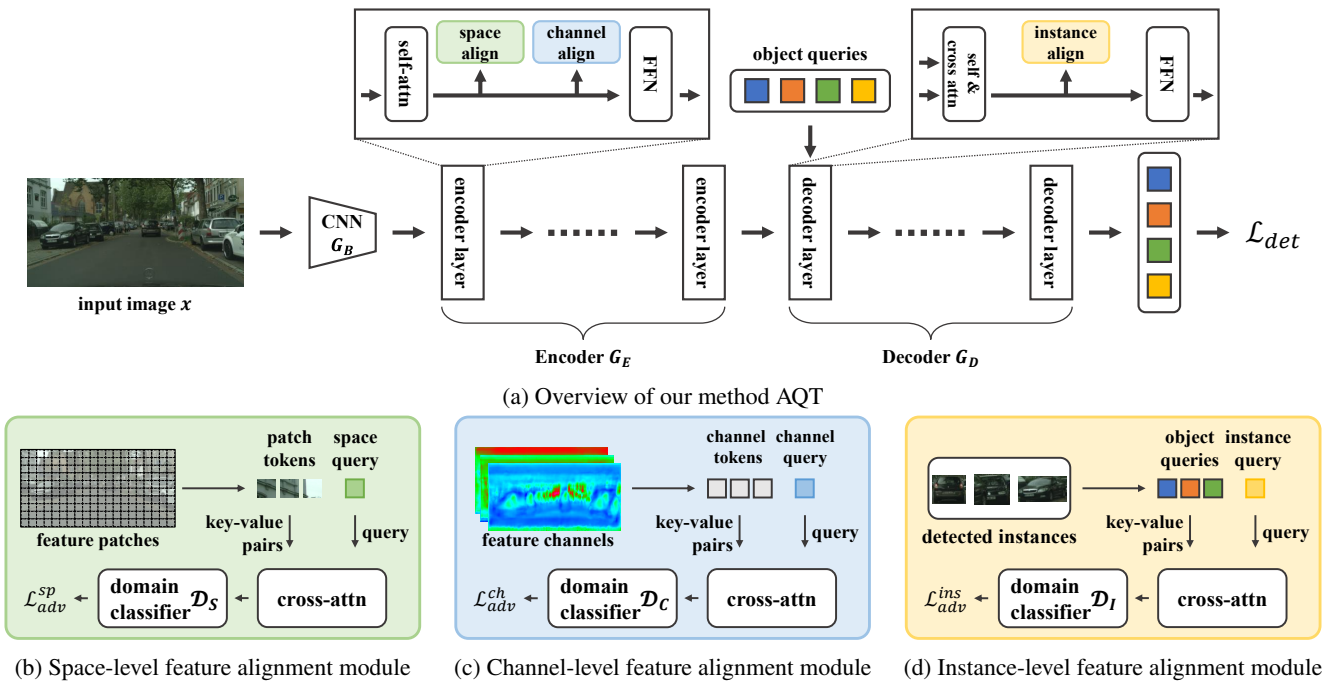
2.2 Domain Adaptive Object Detection

To circumvent performance degradation caused by distribution shifts, the research of domain adaptive object detection has drawn attention recently. The seminal work [Chen *et al.*, 2018] investigates adversarial domain adaptation [Ganin and Lempitsky, 2015] for Faster R-CNN [Ren *et al.*, 2015]. Inspired by it, many following works [Saito *et al.*, 2019; Xu *et al.*, 2020a; Hsu *et al.*, 2020; VS *et al.*, 2021; Wang *et al.*, 2021b] employ domain classifiers on different aspects of cross-domain features. Other groups of works utilize self-training [Kim *et al.*, 2019; Munir *et al.*, 2021], mean teacher framework [Cai *et al.*, 2019; Deng *et al.*, 2021], and image-to-image translation [Chen *et al.*, 2020]. Although these methods are not categorized in adversarial domain adaptation, some of them also adopt adversarial learning as a part of their algorithms.

The aforementioned methods are usually constrained to CNN-based detectors, such as Faster R-CNN [Ren *et al.*, 2015], SSD [Liu *et al.*, 2016], and FCOS [Tian *et al.*, 2019]. Due to the inherent differences between CNNs and transformers, they may be inapplicable to or suboptimal for transformer-based detectors. While detection transformers [Carion *et al.*, 2020; Zhu *et al.*, 2021] have revealed their potentials, less efforts have been made for their adaptation. A recent study [Wang *et al.*, 2021a] empirically finds that existing approaches bring only limited improvements for detection transformers. The authors attribute this finding to the lack of domain invariance in sequential features, and thus propose domain query-based feature alignment. As their *domain query* belongs to the generator, it can aggregate global features, but not necessarily domain-specific ones. In contrast, our *adversarial token* works for the discriminator and attends to the hard-to-align feature tokens. Thus, the generator is encouraged to eliminate domain-specific features and produce more domain-invariant ones. In addition, thanks to the flexible adversarial token, our AQT can carry out space-, channel-, and instance-level feature alignment in a unified manner.

3 Proposed Method

This section describes our proposed approach. We first give a method overview and then elaborate the space-, channel-, and instance-level feature alignment, respectively.



(a) Overview of our method AQT

(b) Space-level feature alignment module

(c) Channel-level feature alignment module

(d) Instance-level feature alignment module

Figure 2: Overview of our proposed Adversarial Query Transformers (AQT). (a) Our AQT framework with (b) space-level, (c) channel-level, and (d) instance-level feature alignment.

3.1 Overview

Given a source dataset $S = \{x_s^i, y_s^i\}_{i=1}^{N_s}$ and an unlabeled target dataset $T = \{x_t^i\}_{i=1}^{N_t}$ where x denotes an image and y represents the ground truth for object detection, we train an adaptive detection transformer F with both S and T, and evaluate it on data in the target domain. To this end, the proposed AQT integrates adversarial feature alignment into detection transformer, and carries out domain invariance in the space, channel, and instance levels.

An overview of our framework is shown in Figure 2a. The framework consists of a generator which is an object detector and three discriminators which perform feature alignment in the space, channel, and instance levels, respectively. The detector includes a CNN backbone G_B for feature extraction and a detection transformer with an encoder G_E and a decoder G_D. Given an input image x, we apply the backbone G_B and get its feature maps $z \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels, and H and W denote the map height and width, respectively. The feature maps z are then flattened into patch tokens $z_p \in \mathbb{R}^{C\times L}$, where $L = H \times W$ is the number of tokens. The encoder G_E aggregates features for each token in z_p via the self-attention mechanism. The decoder G_D carries out object detection by taking the *object queries* as input. A detection loss $\mathcal{L}_{det}$ is applied to derive the network.

For adversarial feature alignment, each layer of the encoder G_E consists of not only a self-attention module and a FFN but also a space-level alignment module (green-shaded region in Figure 2a) and a channel-level alignment module (blue-shaded region in Figure 2a). Similarly, each layer in the decoder G_D is associated with an additional instance-level alignment module (yellow-shaded region in Figure 2a). The three modules act as the discriminators, and perform space-, channel-, and instance-level feature alignment, respectively.

Figure 2b illustrates the space-level alignment module, which includes a cross-attention layer. A space-level adversarial query, or *space query* for short, iterates through the layers in G_E. The pixels in the feature maps yield the feature tokens and serve as the key-value pairs. A domain classifier D_S is adopted in the space-level alignment module of the last encoder layer, and is learned to predict the domain of the space query. By changing the feature tokens from pixels to channels and detected objects, Figure 2c and Figure 2d show the channel-level and instance-level alignment modules, respectively. We elaborate the three modules as follows.

3.2 Space-level Feature Alignment

As mentioned in [Wang *et al.*, 2021a], direct feature alignment on the CNN backbone results in suboptimal performance on detection transformers. To address this problem, we introduce the discriminator, *i.e.* the *adversarial token* and cross-attention layers, for adversarial feature alignment.

The adversarial token, similar to the class token [Dosovitskiy *et al.*, 2021], is trainable and derived to fulfill some objective. For the class token, the objective is to find discriminative features for object recognition, while for adversarial token, the objective is to identify hard-to-align feature tokens for domain classification. In attention mechanisms, a query Q attends to a set of keys K and maps itself into a linear combination of the corresponding values V depending on attention weights. As a result, whichever tokens the query gives a high attention weight on, these tokens are likely to be discriminative for domain classification, *i.e.* being domain-specific.

In each layer i of the transformer encoder G_E, we embed a space-level alignment module after the original self-attention layer. This module consists of a cross-attention layer and a linear mapping layer (simplified from a FFN). As shown in Figure 2b, in the cross-attention layer, the *adversarial token* acts as a query, while the patch tokens z_p, which can be viewed as the output of the generator, act as the key-value pairs. Since this process is to align space-level features, we term the adversarial token as the space-level adversarial query, or space query q_s. Let q_s^i and z_p^i denote the space query and patch tokens which the alignment module in the i-th encoder layer takes as input, where q_s^1 is a randomly initialized C-dimensional vector and $z_p^1 = z_p$. The module in the i-th encoder layer maps the space query q_s^i to its successor q_s^{i+1} w.r.t. z_p^i by

$$q_s^{i+1} = \mathrm{Linear}(\mathrm{MultiHeadAttn}(q_s^i, z_p^i)), \tag{1}$$

where the multi-head attention function $\mathrm{MultiHeadAttn}$ defined in [Zhu *et al.*, 2021] takes two inputs, query and key. Note that we omit positional embeddings of the keys, normalization layers, and residual connections for simplicity. In practice, the key-value pairs pass by a gradient reversal layer [Ganin and Lempitsky, 2015] first to reverse the gradients backpropagated to the generator. This process continues until the end of G_E, where i is the number of encoder layers N. The space-level domain classifier D_S then identifies the domain of the image given the output of the final-layer space query q_s^{N+1}. This domain classification task is optimized by minimizing the following binary cross-entropy loss

$$\begin{aligned}\mathcal{L}_{adv}^{sp} = &- d \log D_S(q_s^{N+1}) - \\ &(1-d)\log(1 - D_S(q_s^{N+1})),\end{aligned} \tag{2}$$

where d denotes the domain label. It takes value 0 for source images, and 1 otherwise. Derived to minimize this adversarial loss $\mathcal{L}_{adv}^{sp}$, the space query q_s and the space-level alignment module manage to identify domain-specific local patches. The generator, *i.e.* the original layers in detection transformer, with an aim to maximize $\mathcal{L}_{adv}^{sp}$, is forced to generate more domain-invariant features. As adversarial learning progresses, space-level cross-domain features are adapted.

3.3 Channel-level Feature Alignment

Although space-level feature alignment effectively suppresses domain-specific local patches, it alone does not eliminate the global biases in the encoder G_E. The main reason is that attention maps are relatively sparse. Only the keys of the most discriminative patches are assigned high attention weights. To solve this problem, we also adopt channel-level feature alignment in G_E.

We define *channel tokens* $z_c \in \mathbb{R}^{\hat{L} \times C}$ as a different view of patch tokens z_p. Each channel token comes from a feature map. $\hat{L}$ is a hyperparameter, and $\hat{L} = C \ll L$ empirically. The reason of resizing is to handle images of different sizes and reduce parametrization overloads. In our implementation, we reshape z_p back into $z \in \mathbb{R}^{C \times H \times W}$, and pool z to yield $\hat{z} \in \mathbb{R}^{C \times P \times P}$, where $P^2 = \hat{L}$. Thus, we obtain z_c by flattening and transposing $\hat{z}$.

To enable channel-level alignment, we leverage a channel-level adversarial query, or channel query q_c. Similar to the space-level alignment module, we embed a channel-level alignment module in each layer of G_E. Let q_c^i and z_c^i denote the channel query and channel tokens which the i-th alignment module takes as input, where q_s^1 is a randomly initialized $\hat{L}$-dimensional vector and z_c^i comes from z_p^i. The module in the i-th encoder layer maps the channel query q_c^i to its successor q_c^{i+1} w.r.t. z_c^i via

$$q_c^{i+1} = \mathrm{Linear}(\mathrm{MultiHeadAttn}(q_c^i, z_c^i)). \tag{3}$$

Again, this process continues until the end of G_E. The channel-level domain classifier D_C then identifies the domain of the image given the final-layer channel query q_c^{N+1}. The channel-level adversarial loss thus yields

$$\begin{aligned}\mathcal{L}_{adv}^{ch} = &- d \log D_C(q_c^{N+1}) - \\ &(1-d)\log(1 - D_C(q_c^{N+1})).\end{aligned} \tag{4}$$

3.4 Instance-level Feature Alignment

Although we make the encoder G_E domain-adaptive using both space-level and channel-level alignment, the decoder G_D remains biased towards the source domain. Thus, we remedy this issue and show the flexibility of our method by extending it to instance-level feature alignment.

In a detection transformer, object queries $z_o \in \mathbb{R}^{C \times L_o}$, are L_o learned embeddings that decode object representations from the encoder output. To align object representations in z_o, we introduce the instance-level adversarial query, or instance query q_i. Similar to the alignment modules built in G_E, we embed an instance-level module in each layer of G_D. Let q_i^j and z_o^j denote the instance query and object queries which the j-th alignment module takes as input, where q_i^1 is a randomly initialized C-dimensional vector and $z_o^1 = z_o$. The module in the j-th decoder layer maps the instance query q_i^j to its successor q_i^{j+1} w.r.t. z_o^j by

$$q_i^{j+1} = \mathrm{Linear}(\mathrm{MultiHeadAttn}(q_i^j, z_o^j)). \tag{5}$$

This process continues until the end of G_D, where j is the number of decoder layers M. The instance-level adversarial loss is thus yielded

$$\begin{aligned}\mathcal{L}_{adv}^{ins} = &- d \log D_I(q_i^{M+1}) - \\ &(1-d)\log(1 - D_I(q_i^{M+1})).\end{aligned} \tag{6}$$

Overall Objective. We summarize the overall objective of the proposed Adversarial Query Transformer. The overall adversarial loss function is formulated as

$$\mathcal{L}_{adv} = \lambda_{sp}\mathcal{L}_{adv}^{sp} + \lambda_{ch}\mathcal{L}_{adv}^{ch} + \lambda_{ins}\mathcal{L}_{adv}^{ins}, \tag{7}$$

where λ_{sp}, λ_{ch}, and λ_{ins} denote the balancing weights for the corresponding terms in the loss function. $\mathcal{L}_{det}$ denotes the detection loss of the adopted detection transformer. The overall loss function of the proposed AQT is

$$\mathcal{L}_{AQT} = \mathcal{L}_{det} - \mathcal{L}_{adv}, \tag{8}$$

and the optimization objective for our domain adaptive detection transformer F is

$$F^* = \arg\min_{F} \min_{\substack{G_B, G_E \\ G_D}} \max_{\substack{D_S, D_C \\ D_I}} \mathcal{L}_{AQT}. \tag{9}$$

Method	Backbone	prsn	rider	car	truck	bus	train	motor	bike	mAP
Source Only (Faster R-CNN)	R-50	26.9	38.2	35.6	18.3	32.4	9.6	25.8	28.6	26.9
DA-Faster [Chen *et al.*, 2018]	R-50	29.2	40.4	43.4	19.7	38.3	28.5	23.7	32.7	32.0
RPA [Zhang *et al.*, 2021]	V-16	33.3	45.6	50.5	30.4	43.6	42.0	29.7	36.8	39.0
HTCN [Chen *et al.*, 2020]	V-16	33.2	47.5	47.9	31.6	47.4	40.9	32.3	37.1	39.8
ICCR-VDD [Wu *et al.*, 2021]	V-16	33.4	44.0	51.7	33.9	52.0	34.7	34.2	36.8	40.0
DSS [Wang *et al.*, 2021b]	R-50	42.9	51.2	53.6	33.6	49.2	18.9	**36.2**	41.8	40.9
KTNet [Tian *et al.*, 2021]	V-16	46.4	43.2	60.6	25.8	41.2	40.4	30.7	38.8	40.9
UMT [Deng *et al.*, 2021]	V-16	33.0	46.7	48.6	**34.1**	**56.5**	46.8	30.4	37.3	41.7
MeGA-CDA [VS *et al.*, 2021]	V-16	37.7	49.0	52.4	25.4	49.2	46.9	34.5	39.0	41.8
ViSGA [Rezaeianaran *et al.*, 2021]	R-50	38.8	45.9	57.2	29.9	50.2	**51.9**	31.9	40.9	43.3
Source Only (FCOS)	R-50	36.9	36.3	44.1	18.6	29.3	8.4	20.3	31.9	28.2
EPM [Hsu *et al.*, 2020]	R-50	44.2	46.6	58.5	24.8	45.2	29.1	28.6	34.6	39.0
SSAL [Munir *et al.*, 2021]	V-16	45.1	47.4	59.4	24.5	50.0	25.7	26.0	38.7	39.6
Source Only (Deformable DETR)	R-50	37.7	39.1	44.2	17.2	26.8	5.8	21.6	35.5	28.5
SFA† [Wang *et al.*, 2021a]	R-50	47.1	46.4	62.2	30.0	50.3	35.5	27.9	41.2	42.6
AQT† (Ours)	R-50	**49.3**	**52.3**	**64.4**	27.7	53.7	46.5	36.0	**46.4**	**47.1**

Table 1: Results of *Cityscapes to Foggy Cityscapes*. "prsn", "motor", and "bike" denote "person", "motorcycle", and "bicycle", respectively.

Method	prsn	rider	car	truck	bus	motor	bike	mAP
Source Only (Faster R-CNN)	28.8	25.4	44.1	17.9	16.1	13.9	22.4	24.1
DA-Faster [Chen *et al.*, 2018]	28.9	27.4	44.2	19.1	18.0	14.2	22.4	24.9
ICR-CCR-SW [Xu *et al.*, 2020a]	32.8	29.3	45.8	**22.7**	20.6	14.9	**25.5**	27.4
Source Only (FCOS)	38.6	24.8	54.5	17.2	16.3	15.0	18.3	26.4
EPM [Hsu *et al.*, 2020]	39.6	26.8	55.8	18.8	19.1	14.5	20.1	27.8
Source Only (Deformable DETR)	38.9	26.7	55.2	15.7	19.7	10.8	16.2	26.2
SFA [Wang *et al.*, 2021a]	**40.2**	27.6	57.5	19.1	**23.4**	15.4	19.2	28.9
AQT (Ours)	38.2	**33.0**	**58.4**	17.3	18.4	**16.9**	23.5	**29.4**

Table 2: Results of *Cityscapes to BDD100k daytime*. "prsn", "motor", and "bike" denote "person", "motorcycle", and "bicycle", respectively. All competing methods are developed upon ResNet-50.

4 Experimental Results

4.1 Datasets and Experimental Settings

Cityscapes to Foggy Cityscapes. Cityscapes [Cordts *et al.*, 2016] is an urban scene dataset containing 2,975 training images and 500 validation images. Foggy Cityscapes [Sakaridis *et al.*, 2018] is synthesized from and shared annotations with Cityscapes. We take the highest fog density images following [Rezaeianaran *et al.*, 2021]. In this setting, Cityscapes is used as the source domain, while Foggy Cityscapes is used as the target domain. 8 categories are considered.

Cityscapes to BDD100k daytime. BDD100k [Yu *et al.*, 2020] is a large-scale driving dataset with diverse scenarios. In this setting, Cityscapes is used as the source domain, while the daytime subset of BDD100k is selected as the target domain. Following [Xu *et al.*, 2020a], the common 7 categories are considered.

Sim10k to Cityscapes. Sim10k [Johnson-Roberson *et al.*, 2017] is a synthetic driving dataset containing 10,000 images. In this setting, Sim10k is used as the source domain, while Cityscapes is used as the target domain. Following [Chen *et al.*, 2018], only the category car is considered.

4.2 Implementation Details

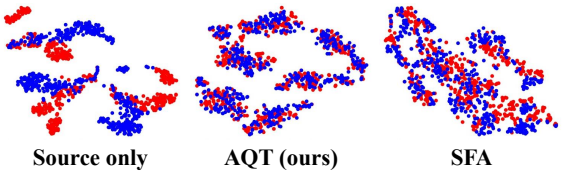
We select Deformable DETR [Zhu *et al.*, 2021] as our object detector with a ResNet-50 backbone pre-trained on ImageNet [Deng *et al.*, 2009]. We inherit most hyperparameters and training settings from Zhu *et al.*, including the detection loss $\mathcal{L}_{det}$ and Xavier initialization [Glorot and Bengio, 2010]. In *Cityscapes to Foggy Cityscapes*, all λ_{sp}, λ_{ch}, and λ_{ins} are set to 10^{-1}. In the other settings, following [Saito *et al.*, 2019], we adopt local alignment on the backbone and weak alignment using the focal loss [Lin *et al.*, 2017]. The λ_{sp}, λ_{ch} and λ_{ins} are set to 10^{-1}, 10^{-5}, and 10^{-4}, respectively. The batch size is set to 8 in all experiments.

4.3 Comparing with State-of-the-arts

From Table 1 to Table 3, we compare the proposed AQT with the existing methods based on Faster R-CNN [Ren *et al.*, 2015], FCOS [Tian *et al.*, 2019], or Deformable DETR [Zhu *et al.*, 2021] on three adaptation settings. "Source Only" indicates the baselines trained with source data only; "V-16" and "R-50" indicate the backbone is VGG-16 [Simonyan and Zisserman, 2015] and ResNet-50 [He *et al.*, 2016]. † indicates iterative bounding box refinement [Zhu *et al.*, 2021].

Cityscapes to Foggy Cityscapes. After adaptation, our method improves the baseline by 18.6% and outperforms all

(a) An input image and its corresponding attention map.



(b) Feature distribution visualizations using t-SNE.

Figure 3: Visualizations to analyze the proposed AQT.

Method	Backbone	*car* AP
Source Only (Faster R-CNN)	R-50	39.4
DA-Faster [Chen *et al.*, 2018]	R-50	41.9
UMT [Deng *et al.*, 2021]	V-16	43.1
SWDA [Saito *et al.*, 2019]	R-50	44.6
MeGA-CDA [VS *et al.*, 2021]	V-16	44.8
RPA [Zhang *et al.*, 2021]	V-16	45.7
GPA [Xu *et al.*, 2020b]	R-50	47.6
ViSGA [Rezaeianaran *et al.*, 2021]	R-50	49.3
KTNet [Tian *et al.*, 2021]	V-16	50.7
Source Only (FCOS)	R-50	42.5
EPM [Hsu *et al.*, 2020]	R-50	47.3
SSAL [Munir *et al.*, 2021]	V-16	51.8
Source Only (Deformable DETR)	R-50	47.4
SFA [Wang *et al.*, 2021a]	R-50	52.6
AQT (Ours)	R-50	**53.4**

Table 3: Results of *Sim10k to Cityscapes*.

Space	Channel	Instance	Box-Refine	Two-Stage	mAP
					28.5
✓					40.6
	✓				36.2
		✓			36.8
✓	✓				41.4
✓		✓			40.9
	✓	✓			40.1
✓	✓	✓			44.8
✓	✓	✓	✓		**47.1**
✓	✓	✓	✓	✓	44.7

Table 4: Quantitative Ablation study of AQT on *Cityscapes to Foggy Cityscapes*. *Box-Refine* indicates iterative bounding box refinement and *Two-Stage* indicates two-stage Deformable DETR.

the other methods. We observe that our method performs worse on the categories "truck", "bus", and "train". We hypothesize this results from fewer instances in these categories, given that transformers rely on sufficient data more than CNNs.

Cityscapes to BDD100k daytime. Our method outperforms all the other methods in terms of mAP. All the methods reported in this experiment is developed upon ResNet-50. Again, our method performs worse on the categories with fewer instances.

Sim10k to Cityscapes. Due to the larger domain gap, our method outperforms the state-of-the-arts by a relatively small margin. However, our method still outperforms SFA [Wang *et al.*, 2021a].

Qualitative Result. We evaluate the detection quality of our method by comparing it with three existing methods, EPM [Hsu *et al.*, 2020], ViSGA [Rezaeianaran *et al.*, 2021], and SFA [Wang *et al.*, 2021a]. The results in Figure 4 show the superior performance of our method.

4.4 Ablation Study and Analysis

In this section, we provide detailed ablation study and analysis of AQT. Following [VS *et al.*, 2021], all these studies are conducted on *Cityscapes to Foggy Cityscapes*.

Where the Adversarial Token Looks at. To better understand how the adversarial token works, we provide an attention map of a space query on an input image, as shown in Figure 3a. The regions with strong responses in the attention map correspond to the foggiest regions in the input image. This demonstrates that the adversarial token tends to attend to the domain-specific features.

Visualizing Feature Distribution Alignment. We use t-SNE [van der Maaten and Hinton, 2008] to visualize the activations of different detection transformers, including the baseline (Source only), SFA [Wang *et al.*, 2021a], and ours. Figure 3b shows the results. Different colors represent the features in different domains. Empirically the aligned features via both AQT and SFA are more indistinguishable than the baseline. Interestingly, we notice the aligned feature distribution via AQT is more perceptually similar to the baseline.

Quantitative Ablation Study. We analyze the effect of the proposed three levels of alignment in Table 4, where *Box-Refine* and *Two-Stage* are inherited from Deformable DETR [Zhu *et al.*, 2021]. Based on this experiment, we conclude three major observations. First, each of the three levels of alignment leads to reasonable improvement. This demonstrates the effectiveness and flexibility of the proposed mechanism. Second, when a certain level of alignment is augmented with another one, a steady growth occurs. We can observe a rather considerable improvement when the three alignments are adopted, compared to the settings where only any two are adopted. Lastly, our framework can also benefit from other techniques. When using *Box-Refine*, our method achieves the best result. Thus, we take it as our default setting when compared with the state-of-the-arts.

Qualitative Ablation Study. In Figure 5, we show several detection results from the baseline (Source only), the

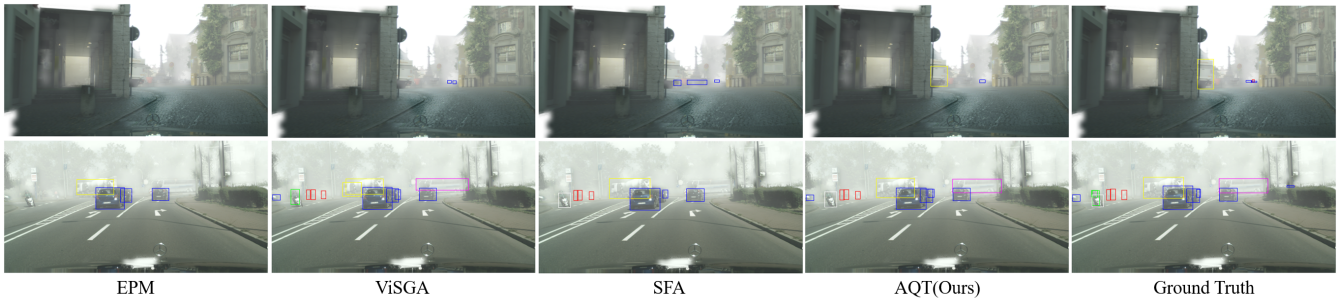


Figure 4: Our detection results compared with the state-of-the-art methods. Different categories are marked with different colors.

Figure 5: Qualitative ablation studies on *Cityscapes to Foggy Cityscapes*. The images on the first to fourth columns are from the models trained with source data only, adopting channel-level alignment (denoted as "Ch."), adopting both channel- & instance-level alignment (denoted as "Ch.+Ins."), and adopting channel- & instance- & space-level alignment (denoted as "Ch.+Ins.+Sp."), respectively. The images in the last column are ground truth bounding boxes. Different categories are marked with different colors.

adapted models, and the ground truth bounding boxes. In the first row, the baseline barely discerns any object, while the adapted model with channel-level alignment ("Ch.") recognizes some objects. With the assistance of instance-level alignment, the adapted model ("Ch.+Ins.") can further localize the truck at the center of the image. It shows that channel-level alignment effectively eliminates cross-domain discrepancy, while the categories with few instances are still hard to detect. This drawback is compensated when adopting instance-level alignment due to reducing biases in the object queries. In the second row, the models in the left three columns fail to recognize the truck at the left of the image, while this false negative is resolved with space-level alignment ("Ch.+Ins.+Sp."). This study empirically shows different levels of alignment can be complementary.

5 Conclusion

In this paper, we present AQT (adversarial query transformers), an adaptation framework to integrate adversarial feature alignment into a detection transformer. It employs a novel adversarial token and a stack of cross-attention layers as the discriminator. As the query in each cross-attention layer, the adversarial token attends the feature tokens from the generator that are hard to align at that moment. Constrained by both the adversarial loss and the detection loss, the generator is forced to eliminate the domain-specific features while maintaining semantics in the feature tokens, hence realizing adversarial feature alignment on detection transformers. The proposed AQT demonstrates the flexibility of the proposed mechanism, combines the merits from the space-, channel-, and instance-level alignment, and yields a new state-of-the-art on several domain adaptive object detection benchmarks.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology (MOST) under grants 109-2221-E-009-113-MY3, 110-2628-E-A49-008, 111-2634-F-007-002, and 110-2634-F-006-022. This work was funded in part by Qualcomm and MediaTek. We thank the National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

References

[Cai *et al.*, 2019] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *CVPR*, 2019.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[Chen *et al.*, 2018] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018.

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

[Cordts *et al.*, 2016] 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 *CVPR*, 2016.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Deng *et al.*, 2021] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *CVPR*, 2021.

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

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

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hoffman *et al.*, 2018] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

[Hsu *et al.*, 2020] 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 *ECCV*, 2020.

[Johnson-Roberson *et al.*, 2017] 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? In *ICRA*, 2017.

[Kim *et al.*, 2019] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *ICCV*, 2019.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[Munir *et al.*, 2021] Muhammad Akhtar Munir, Muhammad Haris Khan, M. Sarfraz, and Mohsen Ali. Ssal: Synergizing between self-training and adversarial learning for domain adaptive object detection. In *NeurIPS*, 2021.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[Rezaeianaran *et al.*, 2021] Farzaneh Rezaeianaran, Rakshith Shetty, Rahaf Aljundi, Daniel Olmeda Reino, Shanshan Zhang, and Bernt Schiele. Seeking similarities over differences: Similarity-based domain alignment for adaptive object detection. In *ICCV*, 2021.

[Saito *et al.*, 2019] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 2019.

[Sakaridis *et al.*, 2018] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 2018.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Tian *et al.*, 2019] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[Tian *et al.*, 2021] Kun Tian, Chenghao Zhang, Ying Wang, Shiming Xiang, and Chunhong Pan. Knowledge mining and transferring for domain adaptive object detection. In *ICCV*, 2021.

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

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[VS *et al.*, 2021] Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A. Sindagi, and Vishal M. Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *CVPR*, 2021.

[Wang *et al.*, 2021a] Wen Wang, Yang Cao, Jing Zhang, Fengxiang He, Zheng-Jun Zha, Yonggang Wen, and Dacheng Tao. Exploring sequence feature alignment for domain adaptive detection transformers. In *ACM MM*, 2021.

[Wang *et al.*, 2021b] Yu Wang, Rui Zhang, Shuo Zhang, Miao Li, Yangyang Xia, Xishan Zhang, and Shaoli Liu. Domain-specific suppression for adaptive object detection. In *CVPR*, 2021.

[Wu *et al.*, 2021] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. Vector-decomposed disentanglement for domain-invariant object detection. In *ICCV*, 2021.

[Xu *et al.*, 2020a] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *CVPR*, 2020.

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

[Yu *et al.*, 2020] 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 *CVPR*, 2020.

[Zhang *et al.*, 2021] Yixin Zhang, Zilei Wang, and Yushi Mao. Rpn prototype alignment for domain adaptive object detector. In *CVPR*, 2021.

[Zhu *et al.*, 2021] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.