

**Abstract**

Recent works on two-stage cross-domain detection have widely explored the local feature patterns to achieve more accurate adaptation results. These methods heavily rely on the region proposal mechanisms and ROI-based instance-level features to design fine-grained feature alignment modules with respect to the foreground objects. However, for one-stage detectors, it is hard or even impossible to obtain explicit instance-level features in the detection pipelines. Motivated by this, we propose an Implicit Instance-Invariant Network (I\(^3\)Net), which is tailored for adapting one-stage detectors and implicitly learns instance-invariant features via exploiting the natural characteristics of deep features in different layers. Specifically, we facilitate the adaptation from three aspects: (1) Dynamic and Class-Balanced Reweighting (DCBR) strategy, which considers the coexistence of intra-domain and intra-class variations to assign larger weights to those sample-scarce categories and easy-to-adapt samples; (2) Category-aware Object Pattern Matching (COPM) module, which boosts the cross-domain foreground objects matching guided by the categorical information and suppresses the uninformative background features; (3) Regularized Joint Category Alignment (RJCA) module, which jointly enforces the category alignment at different domain-specific layers with a consistency regularization. Experiments reveal that I\(^3\)Net exceeds the state-of-the-art performance on benchmark datasets.

1. Introduction

Object detection has achieved remarkable progress due to the unprecedented development of deep convolutional networks (CNNs) and the existence of large-scale annotated datasets. However, collecting large amounts of instance-level annotated data in various domains for object detection is prohibitively costly. An alternative would be applying the off-the-shelf detection model trained on the source domain to a new target domain. However, deep object detectors suffer from performance degradation when applied to a new domain under the presence of domain shift [41]. This problem has inspired the research on Unsupervised Domain Adaptation (UDA) [27], which aims to bridge the distribution discrepancy between source and target domains via knowledge transfer. Numerous approaches, such as moment matching [11, 6, 23, 25, 49] and adversarial learning [7, 42, 37, 24, 44], have been proposed for cross-domain image classification and semantic segmentation.
Compared to the conventional UDA problems, cross-domain object detection is a more sophisticated and challenging problem since the adaptation of classification and regression should be simultaneously considered. Current methods [4, 53, 34, 1, 14, 3, 46, 52, 45] mostly resort to the adversarial feature adaptation to explore discriminative feature patterns at local-level, global-level, and instance-level for adapting two-stage detectors (see top of Fig. 1), Faster R-CNN [33]. However, they heavily rely on the region proposal mechanisms and ROI-based instance-level features to design fine-grained feature alignment modules with respect to the foreground objects. For example, Zhu et al. [53] mine the target discriminative regions based on the region proposals derived from the RPN. Cai et al. [1] regularize the relational graphs by using the ROI-based features. Chen et al. [3] and Xu et al. [45] assist the instance-level feature alignment by the contextual or categorical regularization.

One-stage object detectors, such as SSD [22] and RetinaNet [21], have the merits of being faster and simpler in real-world applications. Unfortunately, it is unrealistic to obtain explicit instance-level features in the one-stage detectors due to the lack of region proposal step. Hence, how to adapt one-stage detectors is vital for practical scenarios but yet to be thoroughly studied. The motivation of this paper is shown in the bottom of Fig. 1. Deep features in the standard CNNs must eventually transition from general to specific along the network [48]. Inspired by this, in one-stage detectors, we can reasonably envision that the features at lower layers (e.g., color, corner, edge, and illumination) are expected to be mostly instance-uninformative, while the features at higher layers (e.g., object categories) are instance-informative. Therefore, we need to alleviate the negative influence of uninformative features and promote the alignment of informative features, i.e., suppress redundant (such as background) information from the lower layers and enhance the cross-domain semantic correlation of foreground objects at the higher layers.

In this paper, we propose an Implicit Instance-Invariant Network (I²Net) that removes the need for requiring explicit instance-level features. Instead, we implicitly learn instance-invariant features via the alignment of transferable regions and images while preserving the inter-domain class relationships. To be specific, we facilitate the adaptation of one-stage detectors from three aspects. Firstly, upon observing that there exist two conceptually orthogonal distribution variations hidden in the target data, i.e., intra-domain and intra-class variations, we propose a Dynamic and Class-Balanced Reweighting (DCBR) strategy to dynamically reweight each target sample based on its adaptation difficulty, which is measured by the degree of class imbalance and the prediction uncertainty of a multi-label classifier. Secondly, considering that object with the same category label but from different domains will share similar object patterns, we design a Category-aware Object Pattern Matching (COPM) module to boost cross-domain foreground objects matching guided by the categorical information and suppress the uninformative background features at lower layers. Finally, we develop a Regularized Joint Category Alignment (RJCA) module to enable category alignment by considering complementary effect of different domain-specific layers and further incorporate a consistency regularization term with respect to the average prediction of different detection heads. Experimental results show that the proposed I²Net significantly improves the state-of-the-art performance of one-stage cross-domain object detection on three benchmarks.

2. Related Work

Unsupervised Domain Adaptation (UDA) UDA methods have attracted much attention for alleviating the distributional variations between two distinct domains in image classification, semantic segmentation, and object detection. For UDA, a typical solution is to match the source and target feature distributions in the common space by embedding disparity measures into deep architectures, such as Maximum Mean Discrepancy (MMD) [43, 23], Correlation Alignment (CORAL) [40], Central Moment Discrepancy (CMD) [49], and transport distance [20, 47]. Inspired by the success of Generative Adversarial Nets (GAN) [12], a large amount works [8, 42, 35, 30, 44, 2, 50, 17] have been done by adversarially learning domain-invariant representations with extra categorical regularization.

Object Detection Object Detection is one of the most fundamental computer vision problems in the past few decades [54]. Our work focuses on how to adapt object detectors, so we only review several representative two-stage and one-stage detectors. The series of region-based convolutional networks (i.e., R-CNN [10], Fast R-CNN [9], and Faster R-CNN [33]) have achieved compelling results in terms of detection accuracy. They count on the region proposal mechanisms to classify region of interest (ROI) independently [10], or share the convolution features with ROI pooling layer [9], or produce the region proposals based on a Region Proposal Network (RPN) [33]. On the other hand, one-stage detectors, such as SSD [22], YOLO [31, 32], and RetinaNet [21] have shown a clear superiority on the inference speed by directly carrying out the category confidence prediction and the bounding box regression.

UDA for Object Detection Domain Adaptive Faster R-CNN [4] is a pioneering two-stage cross-domain detection method that reduces the distributional shift by adversarially learning domain-invariant features on both image-level and instance-level. Considered the local nature of object detection task, most recent efforts [53, 34, 1, 14, 3, 46, 52, 45, 15, 39, 51] are devoted to capture the local feature patterns
and explicitly align them at multiple levels. For instance, Chen et al. [3] propose to hierarchically calibrate the transferability of different level features (i.e., local-region, image, and instance) to improve the discriminability of detectors; Xu et al. [46] and Zheng et al. [52] draw motivation from the cross-domain prototype alignment [44, 2, 28] to align the foreground objects with the same category between domains. However, these methods cannot be simply extended to the one-stage detectors since they highly rely on the region proposals and pooled instance-level features. The study on adapting one-stage object detectors is very limited. A pioneering attempt [19] presents a weak self-training strategy by simultaneously reducing the false positives and false negatives during the hard negative mining. However, self-training-based methods may be vulnerable to the error accumulation problem, especially on the sophisticated cross-domain detection scenarios. Moreover, how to learn instance-invariant feature representations without the guidance of region proposal mechanisms, which is crucial for adapting one-stage detectors, still remains unclear.

3. Methodology

In the task of cross-domain object detection, we are given a source domain \( D_s = \{ (x_i^s, y_i^s, b_i^s) \}_{i=1}^{N_s} \) of \( N_s \) labeled samples, and a target domain \( D_t = \{ x_j^t \}_{j=1}^{N_t} \) of \( N_t \) unlabeled samples. \( D_s \) and \( D_t \) are drawn from different data distributions, but share an identical group of classes (\( K \) classes in all). The objective of this paper is to transfer knowledge from \( D_s \) to \( D_t \) and achieve good detection results in \( D_t \).

Framework Overview. To this end, we propose an Implicit Instance-Invariant Network (I\(^3\)Net), which is comprised of three components, namely, Dynamic and Class-Balanced Reweighting (DCBR), Category-aware Object Pattern Matching (COPM), and Regularized Joint Category Alignment (RJCA). The overview of I\(^3\)Net is demonstrated in Figure 2. The basic idea is to utilize the inherent characteristics of representations at different layers of the detector to compensate for the lack of explicit instance-level features. DCRB reweights target samples based on the adaptation difficulty with respect to the intra-domain and intra-class variations, COPM captures the foreground object patterns and suppresses redundant background information, and RJCA promotes the cross-domain category alignment in different domain-specific layers (connected with detection heads) with a consistency regularization. Following the prior work on adapting one-stage detectors [19], our I\(^3\)Net is based on the SSD [22] framework.

3.1. Dynamic and Class-Balanced Reweighting

To date cross-domain detection methods [4, 34, 46, 52] mainly focus on the feature-level adaptation and treat all the target samples equally, while they neglect the distributional characteristics of the target data, which are crucial for the adaptation process. By contrast, the proposed DCRB strategy explicitly explores the intra-domain and intra-class variations within the unlabeled target domain to assign larger weights to those sample-scarce categories and easy-to-adapt samples. We analysis these two variations in the following.

Intra-Domain Variations. The class-imbalance probl-
lem [26], which refers to the inequality among the number of examples belonging to different classes, commonly exists in the object detection. Prior efforts, such as Focal Loss [21] and hard example mining [22, 36], are devoted to tackle the foreground-background class imbalance, which is irrelevant to the number of examples per class in a single domain. In cross-domain detection, we argue that the foreground-background class imbalance, which is dataset-relevant and may be different between domains, is prone to deteriorate the adaptation performance since the adaptation of each class will be affected by the number of examples per class in both domains, i.e., the adaptation difficulty of different categories may be distinct.

**Intra-Class Variations.** Owing to the difference of background, object co-occurrence, and scene layouts across domains, excessively align the source and target features in the full dataset will result in negative transfer, i.e., some target samples may be less transferable or even non-transferable. However, most leading cross-domain detection methods treat the target domain as a whole without considering the structures of intra-class data distributions. Motivated by this, we assume that the adaptation difficulty of different samples within the same class may be distinct. An intuitive solution is to utilize re-weighting techniques. However, this solution suffers a critical limitation in the context of cross-domain object detection. In contrast to the classification problem, where a single image usually contains only one semantic category, there exist multiple instances within the same image in the detection problem. Thus, how to measure the adaptation difficulty of an input target sample in cross-domain detection remains unclear.

Based on the above discussions, we formally provide the details of the proposed DCRB. The DCRB consists of two steps: (i) estimate the adaptation difficulty of each individual target sample and each target class; (ii) reweight target samples based on the estimation results. Technically, the adaptation difficulty of an target example \( x^t \) (for ease of denotation, we omit the subscript of \( x^t \) and \( x^s \) when they apply) is measured by using an image-level multi-label classifier \( F_{mlc} \) in Fig. 2. We first train \( F_{mlc} \) based on the labeled source samples for initialization. The multi-label classification loss on the source domain is formulated as:

\[
\mathcal{L}_{mlc} = \sum_{k=1}^{K} y_k^s \cdot \log(\hat{y}_k^s) + (1 - y_k^s) \cdot \log(1 - \hat{y}_k^s) \tag{1}
\]

where \( y_k^s \) is the \( k^{th} \) \( (k = \{1, 2, ..., K\}) \) element of \( y^s \) and \( \hat{y}_k^s = F_{mlc}(G_1(x^s))^k \) \((G_1 \ is \ a \ feature \ extractor \ connected \ to \ F_{mlc}). \ (G_1) \ is \ a \ feature \ extractor \ connected \ to \ F_{mlc} \). \( y_k^s = 1 \) means that there exists at least one object of class \( k \) in \( x^s \); otherwise, \( y_k^s = 0 \) indicates that \( x^s \) does not contain the object of class \( k \). For each target sample \( x^t \), we denote the prediction of its multi-label classification as \( \hat{y}_k^t = F_{mlc}(G_1(x^t))^k \). Then, we define the weight function of a target sample \( x^t \) w.r.t. the intra-class variation by using its multi-label classification output:

\[
w_1^t = \frac{1}{K'} \sum_{k=1}^{K} \mathbb{I}(\hat{y}_k^t(x^t) > \tau) \cdot \hat{y}_k^t(x^t) + 1 \tag{2}
\]

where \( K' = \sum_{k=1}^{K} \mathbb{I}(\hat{y}_k^t(x^t) > \tau), \) and \( \tau \) is a threshold. \( \mathbb{I}(a) \) is an indicator function which is \( 1 \) if \( a \) is true and \( 0 \) otherwise. By doing so, target samples with higher classification confidence scores will be assigned larger weights since they are more similar with source domain. Note that the value of \( w_1^t \) increases continuously since the source and target distributions are getting closer as training proceeds.

To estimate the number of examples per class in \( D_t \), we resort to the classification output for roughly dividing \( D_t \) into \( K \) classes. \( x^t \) is added into the target domain of the class \( D_t^k \) if \( k' = \arg \max_k \hat{y}_k^t(x^t) \). Then, the unlabeled target samples \( D_t \) are split into \( K \) classes, i.e., \( D_t = \{D_t^k\}_{k=1}^{K} \). To this end, we are able to assign larger weights to those sample-scarce categories. The weight function of \( x^t \) w.r.t. the intra-domain variation is formulated as,

\[
w_2^t = e^{(1 - N_t^k/N_k)} \tag{3}
\]

where \( N_t^k \) denotes the number of samples in class \( k \).

Based on Eq. (2) and Eq. (3), the overall weight function of a target sample \( x^t_j \) is formulated as follows,

\[
w^t = \theta w_1^t + (1 - \theta) w_2^t \tag{4}
\]

where \( \theta \) is a hyper-parameter to balance \( w_1^t \) and \( w_2^t \). After adding the weights to all target samples, the adversarial loss of image-wise domain discriminator \( D_g \) can be written as:

\[
\mathcal{L}_{adv} = -\frac{1}{N_t} \sum_{i=1}^{N_t} \log(D_g(G_2(x^t_i))) -\frac{1}{N_s} \sum_{j=1}^{N_s} w_j^t \cdot \log(1 - D_g(G_2(x^t_j))) \tag{5}
\]

where \( G_2 \) is a feature extractor that is connected to \( D_g \).

### 3.2. Category-Aware Object Pattern Matching

As we discussed in Section 1, the feature representations at lower layers contain various redundant information (e.g., background) and should not be fully aligned. Previous works [4, 34], which strictly matching the low-level features, may result in inferior performance especially on the one-stage detection. During the exploration, we observe that objects with the same category label but from different domains will own similar object patterns. Object pattern, which refers to the discriminative features of foreground objects, can provide rich semantic information w.r.t. the objects, such as object category, shape, size, etc. Driven
by this finding, we propose a Category-aware Object Pattern Matching (COPM) module to boost cross-domain foreground objects matching guided by the categorical information and suppress the uninformative background features.

Suppose that we have a CNN layer (e.g., Conv 4,3 in SSD300) and its corresponding activation tensor \( A \in \mathbb{R}^{C \times H \times W} \), which consists of \( C \) feature planes and has height of \( H \) and width of \( W \). An intuitive idea for local feature alignment is to extract attention maps from both motivations from the randomized multilinear map \( R \). Note that we incorporate a pixel-level domain discriminator \( \Phi \) to estimate the tensor product via Hadamard product, i.e., \( R^p \) in SSD300) and its corresponding activation tensor \( \Phi \).

After fusion, we align the source and target object patterns by minimizing distance between the them, \( \Phi(z_{st}^l, z_{st}^l) \), which can be written as:

\[
\Phi(z_{st}^l, z_{st}^l) = \frac{1}{n_k^s} \sum_{j=1}^{n_k^s} \sum_{m=1}^{N} y_{mk}^{lmk} \cdot z_{st}^l
\]

where \( n_k^s \) denotes the number of source objects labeled with class \( k \), \( m \) is the pixel index in \( z \), \( y_{mk}^{lmk} \in \{0, 1\} \) is an indicator for determining whether the current pixel is predicted as class \( k \). The source global prototype of each class is computed at the beginning of training. Let the prediction of detection head w.r.t. a target object be represented by \( \hat{p}(z_{st}^l) \).

To reduce computational cost, we flatten the source and target attention maps to vectors, which are denoted as \( f^s \) and \( f^t \). Finally, we align the source and target object patterns by minimizing distance between them, the loss function is formulated as:

\[
\mathcal{L}_{ls} = \sqrt{H \times W} \cdot \Phi \left( \frac{f^s}{\|f^s\|_2}, \frac{f^t}{\|f^t\|_2} \right)
\]

where \( \Phi(x, x') = \|x - x'\|_2 \) is the Euclidean distance. Note that we incorporate a pixel-level domain discriminator (i.e., \( D_l \) in Fig. 2) into COPM to further reduce the low-level feature disparity. Thus, the objective of COPM is formulated as: \( \mathcal{L}_{coppm} = \mathcal{L}_{ls} + \mathcal{L}_{adv} \), where \( \mathcal{L}_{adv} \) is a vanilla pixel-wise domain adversarial training loss.

3.3. Regularized Joint Category Alignment

Prototype\(^1\)-based feature alignment has been widely explored to measure the category-level feature discrepancy in UDA \cite{44, 2, 29} and been applied to the two-stage cross-domain detection \cite{52, 46}. However, considering the dense prediction property of one-stage detectors, prototype alignment may be error-prone in this case compared to adapting two-stage detectors where most negative proposals will be filtered out. Moreover, prior efforts only implement the prototype alignment in a certain high-level feature layer without considering the potential complementary effect of different domain-specific layers. Motivated by this, we propose a Regularized Joint Category Alignment (RJCA) module to achieve the category alignment at different domain-specific layers and regularize the average prediction consistency of different layers with respect to the same category.

In the light of fully convolutional and multi-level prediction characteristics of one-stage detectors, we aim at jointly enforcing the cross-domain category alignment in different layers. First of all, assume that the deep networks will generate the activations in different layers as \( \{z_{t1}^1, ..., z_{tL}^L\}_{i=1}^{N} \) and \( \{z_{t1}^1, ..., z_{tL}^L\}_{i=1}^{N} \), where \( l \in L \) and \( z \in \mathbb{R}^{C \times H \times W} \). Then, we resort to the per-pixel prediction to compute the prototype of each source class in layer \( l \), which can be written as:

\[
\hat{Z}_{k}^{l|m|} = \frac{1}{n_k^s} \sum_{j=1}^{n_k^s} \sum_{m=1}^{N} y_{mk}^{lmk} \cdot z_{t|m}^l
\]

where \( n_k^s \) denotes the number of source objects labeled with class \( k \), \( m \) is the pixel index in \( z \), \( y_{mk}^{lmk} \in \{0, 1\} \) is an indicator for determining whether the current pixel is predicted as class \( k \). The source global prototype of each class is computed at the beginning of training. Let the prediction of detection head w.r.t. a target object be represented by \( \hat{p}(z_{st}^l) \).

The target local prototype is computed by:

\[
\hat{z}_{t}^{l|m|} = \frac{1}{n_{t}^l} \sum_{j=1}^{n_{t}^l} \sum_{m=1}^{N} y_{tm}^{lmk} \cdot z_{t|m}^l
\]

where \( n_{t}^l \) denotes the number of objects that are assigned with pseudo label \( k \) and \( B_k \) is the mini-batch samples of the target domain. Similarly, we can obtain a set of source local prototypes \( \{\hat{z}_{s}^{l|m|}\}_{k=1}^{K} \). The objective function of joint category alignment is formulated as follows:

\[
\mathcal{L}_{jca} = \sum_{l} \left( \sum_{k} d(\hat{Z}_{k}^{l|m|}, \hat{Z}_{k}^{l|m|}) + \sum_{m,n|m \neq n} h(\hat{Z}_{k}^{l|m|}, \hat{Z}_{k}^{l|m|}) \right)
\]

where \( d \) and \( h \) are two different similarity functions to measure the distance between prototypes. In our case, we instantiate Eq. (11) by the contrastive loss as defined in \cite{13}.

\(^{1}\)Prototype is the mean feature of the samples within the same class.
During training, the global prototype in Eq. (11) is updated by the local prototype in a moving average manner,

$$Z_k^{[l]} \leftarrow \rho Z_k^{[l]} + (1 - \rho) z_k^{[l]}$$

(12)

where $\rho$ is set to 0.7 in all experiments. In addition, we regularize the prediction consistency of different layers w.r.t. the same class $k$ by respectively minimizing their symmetrized Kullback–Leibler (KL) divergence, which is formulated as:

$$L_{pr} = \frac{1}{K} \sum_{l} \sum_{k=1}^{K} \frac{1}{2} D_{KL}(\tilde{p}(z_k^{[l]})) \| \hat{p}(z_k^{[l]}) )$$

\[
+ D_{KL}(\tilde{p}(z_k^{[l]})) \| \hat{p}(z_k^{[l]}) )], \text{ where } l_a, l_b \in L.
\]

where $\tilde{p}(z_k^{[l]})$ and $\hat{p}(z_k^{[l]})$ stand for the average prediction w.r.t. the class $k$ in different layers. Here, to smooth the prediction, we add a temperature variate $T$ ($T = 2$ in all experiments) to the softmax function. To this end, the objective of the proposed RJCA can be written as: $L_{rjca} = L_{jca} + \gamma L_{pr}$, where $\gamma$ is set to 0.1 in all experiments.

3.4. Training Loss

Suppose that the detection loss is denoted as $L_{det}$, which includes the classification and regression losses. Joint all the presented parts, the overall objective function of I3Net is formulated as follows,

$$L_{I3Net} = L_{det} + \lambda_1 L_{dcbr} + \lambda_2 (L_{copm} + L_{rjca})$$

(14)

where $\lambda_1$ and $\lambda_2$ are hyper-parameters for balancing different loss components.

4. Experiments

4.1.Datasets

We conduct experiments based on Pascal VOC [5], Clipart1k, Watercolor2k, and Comic2k [16] datasets. Following the previous one-stage method [19], we utilize the Pascal VOC2007-trainval and VOC2012-trainval datasets as the source domain, and Clipart1k, Watercolor2k, and Comic2k as the target domain respectively. The Pascal VOC [5], which is a real-world image dataset, contains 16,551 images with 20 distinct object categories. Clipart1k [16], which is a graphical image dataset with complex backgrounds, consists of 1K images and has the same 20 categories as Pascal VOC. We utilize all images of Clipart1k as the target domain for both training and testing. Watercolor2k and Comic2k [16] contain 2K images respectively (i.e., 1K as the train set and the other 1K as the test set). They share 6 identical categories with the Clipart1k dataset, i.e., bicycle, bird, cat, car, dog, and person. Following the prior practice [19], we leverage the train set for training and the test set for evaluation.

4.2. Implementation Details

The base detection model in our experiments follows the same setting in [16, 19] that utilize SSD300 [22] framework with VGG-16 [38] architectures. The parameters of VGG-16 is fine-tuned from the model that has been pre-trained on ImageNet. In all experiments, the input images are resized to 300 × 300 and we conduct all augmentations used in [22, 19]. The batch size is selected as 32 (16 source images and 16 target images) to fit the GPU memory. We evaluate the cross-domain detection performance by reporting mean average precision (mAP) with an IoU threshold of 0.5 on the target domain. We adopt the stochastic gradient descent (SGD) optimizer for the detection network training with a momentum of 0.9, an initial learning rate of 0.001, weight decay of $5 \times 10^{-4}$. The learning rate is decreased to 0.0001 after 50 epochs. Note that the multi-label classifier $F_{mle}$ is pre-trained on the label source domain and keeps fixed when training our adaptation network. Without specific notation, we set $\tau = 0.5$ in Eq. (2) and $\theta = 0.5$ in Eq. (4). For the $L$ in RJCA, we set $L = \{Conv7, Conv9\}$ for the I3Net model based on SSD. We set $\lambda_1 = 0.05$ and $\lambda_2 = 1$ in Eq. (14) for all experiments. Our experiments are implemented with the Pytorch deep learning framework.

4.3. Comparisons with State-of-the-Arts

State-of-the-arts. We make comparison to the state-of-the-art cross-domain object detection methods, including Domain Adversarial Neural Networks (DANN) [8], adversarial Background Score Regularization + Weak Self-Training (BSR+WST) [19], Strong-Weak Distribution Alignment (SWDA) [34], and Hierarchical Transferability Calibration Network (HTCN) [3]. The quantitative results of DANN, BSR, WST, and BSR+WST are cited from the original paper [19]. We reproduce the complete SWDA model on our one-stage scenarios. Moreover, we remove the context-aware instance-level alignment component from the HTCN model and re-implement the rest modules in our experiments. Note that mainstream cross-domain detection methods (e.g., [4, 53, 1, 14, 46, 52, 45]) are tailored for two-stage detector and cannot be simply extended to one-stage-based experiments since they highly count on the region proposal mechanisms.

Results on Clipart1k. Table 1 displays the adaptation results on Pascal VOC → Clipart1k. Source Only denotes that the baseline SSD is trained on the source domain and directly tested on the target domain without any adaptation. The proposed I3Net significantly outperforms all the compared methods in terms of mAP and improves over state-of-the-art by +2.0% (35.8% to 37.8%). It is noteworthy that all components of the proposed I3Net are designed appropriately and when we remove one of these components,
Table 1: Results of adapting PASCAL VOC to Clipart1k (%). mAP is reported on Clipart1k.

<table>
<thead>
<tr>
<th>Methods</th>
<th>bike</th>
<th>bird</th>
<th>car</th>
<th>cat</th>
<th>dog</th>
<th>person</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [22]</td>
<td>27.3</td>
<td>60.4</td>
<td>17.5</td>
<td>16.0</td>
<td>14.5</td>
<td>43.7</td>
<td>32.0</td>
</tr>
<tr>
<td>DANN [8]</td>
<td>24.1</td>
<td>52.6</td>
<td>27.5</td>
<td>18.5</td>
<td>20.3</td>
<td>59.3</td>
<td>37.4</td>
</tr>
<tr>
<td>DT+P/ w/o label [16]</td>
<td>16.8</td>
<td>53.7</td>
<td>19.7</td>
<td>31.9</td>
<td>21.3</td>
<td>39.3</td>
<td>39.8</td>
</tr>
<tr>
<td>WST [19]</td>
<td>30.8</td>
<td>65.5</td>
<td>18.7</td>
<td>23.0</td>
<td>24.9</td>
<td>57.5</td>
<td>40.2</td>
</tr>
<tr>
<td>BSR [19]</td>
<td>26.3</td>
<td>56.8</td>
<td>21.9</td>
<td>20.0</td>
<td>24.7</td>
<td>55.3</td>
<td>42.9</td>
</tr>
<tr>
<td>SWDA [34]</td>
<td>29.0</td>
<td>60.7</td>
<td>25.0</td>
<td>20.4</td>
<td>24.6</td>
<td>55.4</td>
<td>36.1</td>
</tr>
<tr>
<td>BSR+WST [19]</td>
<td>28.0</td>
<td>64.5</td>
<td>23.9</td>
<td>19.0</td>
<td>21.9</td>
<td>64.3</td>
<td>43.5</td>
</tr>
<tr>
<td>HTCNI [3]</td>
<td>28.7</td>
<td>67.7</td>
<td>25.3</td>
<td>16.1</td>
<td>28.7</td>
<td>56.0</td>
<td>38.9</td>
</tr>
</tbody>
</table>

Table 2: Results on adaptation from Pascal VOC to Watercolor2k (%). mAP is reported on the Watercolor2k test set.

<table>
<thead>
<tr>
<th>Methods</th>
<th>bike</th>
<th>bird</th>
<th>car</th>
<th>cat</th>
<th>dog</th>
<th>person</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [22]</td>
<td>77.5</td>
<td>46.1</td>
<td>44.6</td>
<td>30.0</td>
<td>26.0</td>
<td>58.6</td>
<td>47.1</td>
</tr>
<tr>
<td>DANN [8]</td>
<td>73.4</td>
<td>41.0</td>
<td>32.4</td>
<td>28.6</td>
<td>22.1</td>
<td>51.4</td>
<td>41.5</td>
</tr>
<tr>
<td>BSR [19]</td>
<td>82.8</td>
<td>43.2</td>
<td>49.8</td>
<td>29.6</td>
<td>27.6</td>
<td>58.4</td>
<td>48.6</td>
</tr>
<tr>
<td>WST [19]</td>
<td>77.8</td>
<td>48.0</td>
<td>45.2</td>
<td>30.4</td>
<td>29.5</td>
<td>64.2</td>
<td>49.2</td>
</tr>
<tr>
<td>SWDA [34]</td>
<td>73.9</td>
<td>48.6</td>
<td>44.3</td>
<td>36.2</td>
<td>31.7</td>
<td>62.1</td>
<td>49.5</td>
</tr>
<tr>
<td>BSR+WST [19]</td>
<td>75.6</td>
<td>45.8</td>
<td>49.3</td>
<td>34.1</td>
<td>30.3</td>
<td>64.1</td>
<td>49.9</td>
</tr>
<tr>
<td>HTCNI [3]</td>
<td>78.6</td>
<td>47.5</td>
<td>45.6</td>
<td>35.4</td>
<td>31.0</td>
<td>62.2</td>
<td>50.1</td>
</tr>
</tbody>
</table>

Table 3: Results on adaptation from Pascal VOC to Comic2k (%). mAP is reported on the Comic2k test set.

<table>
<thead>
<tr>
<th>Methods</th>
<th>bike</th>
<th>bird</th>
<th>car</th>
<th>cat</th>
<th>dog</th>
<th>person</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only [22]</td>
<td>43.3</td>
<td>9.4</td>
<td>23.6</td>
<td>9.8</td>
<td>10.9</td>
<td>34.2</td>
<td>21.9</td>
</tr>
<tr>
<td>DANN [8]</td>
<td>33.3</td>
<td>11.3</td>
<td>19.7</td>
<td>13.4</td>
<td>19.6</td>
<td>37.4</td>
<td>22.5</td>
</tr>
<tr>
<td>BSR [19]</td>
<td>45.2</td>
<td>15.8</td>
<td>26.3</td>
<td>9.9</td>
<td>15.8</td>
<td>39.7</td>
<td>25.5</td>
</tr>
<tr>
<td>WST [19]</td>
<td>45.7</td>
<td>9.3</td>
<td>30.4</td>
<td>9.1</td>
<td>10.9</td>
<td>46.9</td>
<td>25.4</td>
</tr>
<tr>
<td>SWDA [34]</td>
<td>50.6</td>
<td>13.6</td>
<td>31.0</td>
<td>7.5</td>
<td>16.4</td>
<td>41.4</td>
<td>26.8</td>
</tr>
<tr>
<td>HTCNI [3]</td>
<td>50.3</td>
<td>15.0</td>
<td>27.1</td>
<td>9.4</td>
<td>18.9</td>
<td>46.2</td>
<td>27.8</td>
</tr>
</tbody>
</table>

Table 4: Ablation of I³Net on three transfer tasks (%).

<table>
<thead>
<tr>
<th>Source Target</th>
<th>Clipart1k</th>
<th>Pascal VOC Watercolor2k</th>
<th>Comic2k</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCBR w/o Dynamic</td>
<td>37.3</td>
<td>51.4</td>
<td>29.2</td>
</tr>
<tr>
<td>DCBR w/o CB</td>
<td>37.1</td>
<td>51.0</td>
<td>29.3</td>
</tr>
<tr>
<td>COPM w/o C</td>
<td>36.8</td>
<td>51.1</td>
<td>29.0</td>
</tr>
<tr>
<td>COPM w/ MMD</td>
<td>34.9</td>
<td>48.4</td>
<td>27.0</td>
</tr>
<tr>
<td>COPM w/ Adv</td>
<td>37.0</td>
<td>50.7</td>
<td>29.8</td>
</tr>
<tr>
<td>RJCA w/o J</td>
<td>36.6</td>
<td>50.8</td>
<td>29.1</td>
</tr>
<tr>
<td>RJCA w/o PR</td>
<td>37.4</td>
<td>51.5</td>
<td>29.4</td>
</tr>
</tbody>
</table>

the final performance will drop accordingly.

Results on Watercolor2k and Comic2k. Results on the tasks of Pascal VOC → Watercolor2k and Pascal VOC → Comic2k are reported on Table 2 and Table 3 respectively. I³Net achieves better performance on most object categories, indicating that I³Net is capable of learning more transferable representations and for different cross-domain detection scenarios. It is noteworthy that I³Net substantially exhibits better adaptation performance on the challenging transfer task (27.8% to 30.1%), i.e., Pascal VOC → Comic2k, where the domain discrepancy is substantially large between source and target data.

4.4. Further Empirical Analysis

Ablation Study. We verify the effect of the proposed DCBR, COPM, and RJCA by evaluating variants of I³Net. The results are reported in Table 4. (1) DCBR w/o Dynamic and DCBR w/o CB denote that we remove $w_1$ and $w_2$ from Eq. (4) respectively. (2) COPM w/o C denotes that we remove the non-linear fusion step (Eq. (6)) and directly match the source and target vectorized attention maps. COPM w/ MMD and COPM w/ Adv denote that we replace the $L_2$ distance in Eq. (8) by MMD [23] and domain adversarial loss [7] respectively. (3) RJCA w/o J is the variant that only conducts the category alignment in one layer. RJCA w/o PR is the variant without prediction regularization (Eq. (13)). The results of COPM w/ MMD and COPM w/ Adv reveal that $L_2$ distance is able to better preserve the structured information (i.e., object patterns). The results of RJCA w/o J verify the significance of considering the complementary effect of different domain-specific layers.

Visualization of COPM. Figure 3 visualizes the attention maps generated by Source Only [22], HTCNI [3], and I³Net (Ours). The brighter the color is, the larger the weight value is. It is notable that the proposed I³Net is capable of (i) capturing the discriminative regions which contain rich semantic information, (ii) highlighting the foreground objects.
even with small object size, and (iii) suppressing the redundant background information.

**Qualitative detection results.** Figure 4 demonstrates the example of detection results on the three target domains, i.e., Clipart1k, Watercolor2k, and Comic2k. The proposed $I^3$Net consistently and significantly outperforms both Source Only [22] and HTCN† [3] models in different transfer tasks. Owing to the introduction of DCBR, $I^3$Net is capable of precisely detecting the sample-scarce categories (e.g., (a), (d), and (e)). $I^3$Net is able to detect those obscured objects and provide accurate bounding box predictions since we explicitly encourage the alignment of cross-domain object patterns via the proposed COPM (e.g., (a), (b), (c), and (f)). In addition, due to the presence of RJCA, $I^3$Net is able to ensure the cross-domain semantic consistency, and thus significantly reduce the false positive results and enhance the classification accuracy (e.g., (d) and (e)).

5. Conclusion

In this paper, we proposed the Implicit Instance-Invariant Network ($I^3$Net) to solve the cross-domain object detection problem based on the one-stage detectors without requiring explicit instance-level features. The key idea of our method is to implicitly learn instance-invariant features via exploiting the natural characteristics of deep features in different layers, i.e., suppressing redundant information from the lower layers and enhancing the cross-domain semantic correlation of foreground objects at the higher layers. Experiments on three standard cross-domain detection benchmarks verified the effectiveness of our method.

Acknowledgement This work was partially supported by National Key Research and Development Program of China (No.2020YFC2003900) and the National Natural Science Foundation of China under Grants U19B2031, 61971369.
References


[34] KuniakiSaito, YositakaUshiku, TatsuyaHarada, and KateSaenko. Strong-weak distribution alignment for adaptive ob-
object detection. In CVPR, pages 6956–6965, 2019. 2, 3, 4, 6, 7


