Multi-Source Domain Adaptation for Object Detection with Prototype-based Mean Teacher

Atif Belal[1], Akhil Meethal[1], Francisco Perdigon Romero[2], Marco Pedersoli[1], Eric Granger[1]

[1] LIVIA, Dept. of Systems Engineering, ETS Montreal, Canada
[2] GAIA Montreal, Ericsson Canada

atif.belal.1@ens.etsmtl.ca, akhilpm135@gmail.com, francisco.perdigon.romero@ericsson.com,
{marco.pedersoli, eric.granger}@etsmtl.ca

Abstract

Adapting visual object detectors to operational target domains is a challenging task, commonly achieved using unsupervised domain adaptation (UDA) methods. Recent studies have shown that when the labeled dataset comes from multiple source domains, treating them as separate domains and performing a multi-source domain adaptation (MSDA) improves the accuracy and robustness over blending these source domains and performing a UDA. For adaptation, existing MSDA methods learn domain-invariant and domain-specific parameters (for each source domain). However, unlike single-source UDA methods, learning domain-specific parameters makes them grow significantly in proportion to the number of source domains. This paper proposes a novel MSDA method called Prototype-based Mean Teacher (PMT), which uses class prototypes instead of domain-specific subnets to encode domain-specific information. These prototypes are learned using a contrastive loss, aligning the same categories across domains and separating different categories far apart. Given the use of prototypes, the number of parameters required for our PMT method does not increase significantly with the number of source domains, thus reducing memory issues and possible overfitting. Empirical studies indicate that PMT outperforms state-of-the-art MSDA methods on several challenging object detection datasets. Our code is available at https://github.com/imatif17/Prototype-Mean-Teacher

1. Introduction

Object detection (OD), one of the fundamental tasks in computer vision, has made significant progress in recent years [2, 24, 38]. However, this progress, typically measured by performance on curated benchmark datasets such as MS-COCO [16] or Pascal VOC [6], may decline significantly with changes in the data distribution between train-

ing (source) and testing (target) domain data [31, 37]. This distribution shift can be caused by several factors, like variations in capture conditions, e.g., weather, illumination, resolution, geographic locations, and size of objects. Since it is too costly to annotate data collected from every operational target domain, researchers have explored a multitude of techniques for unsupervised domain adaptation (UDA), which seeks to find a common representation space between data distributions of a labelled source domain and an unlabeled target domain [3, 15, 22].

Among the state-of-art UDA strategies for OD, feature alignment [3, 39] and pseudo-labeling of target data [12, 13] are the most common. Recently, impressive results were obtained by combining both strategies, where the popular mean-teacher [23] framework is used to pseudo-label target domain images, and a gradient reversal layer (GRL) [7] integrated domain discriminator is used to align features in an adversarial way [1, 5, 15]. Mainstream UDA methods rely on a single source domain with labeled data during adaptation. However, in a practical setting, several source datasets may be available, and the datasets may also be collected from multiple domains, e.g., sensors, environment, etc. The simplest way to use UDA with multiple sources of training data is by blending source domain data to form a single labeled source dataset, as shown in Fig. 1a. However, considering each domain as a separate source allows the OD model to address domain discrepancies between individual source distributions explicitly, and aligning sources during adaptation has been found to provide a higher level of recognition accuracy and robustness [36]. This setting is referred to as multi-source domain adaptation (MSDA).

Two methods have been proposed for MSDA in OD – Divide-and-Merge Spindle Network (DMSN) [33] and Target-Relevant Knowledge Preservation (TRKP) [27]. The overall architecture of both methods can be divided into domain-general and domain-specific model parameters. The initial part of the network learns a domain-invariant
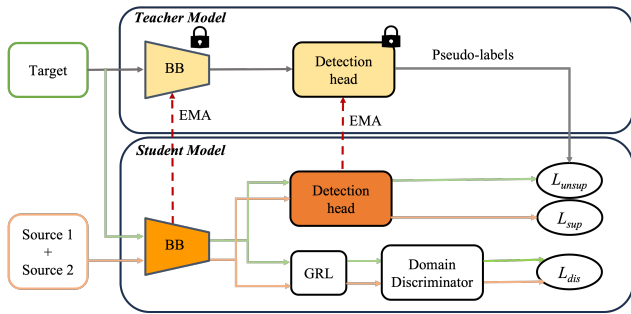
(a) **Mean-teacher with blending of sources** [15]. A standard mean-teacher model with a single student and teacher network, trained by blending data from multiple sources. Domain-invariant features are obtained from the backbone trained with a discriminator. No domain-specific information is considered.
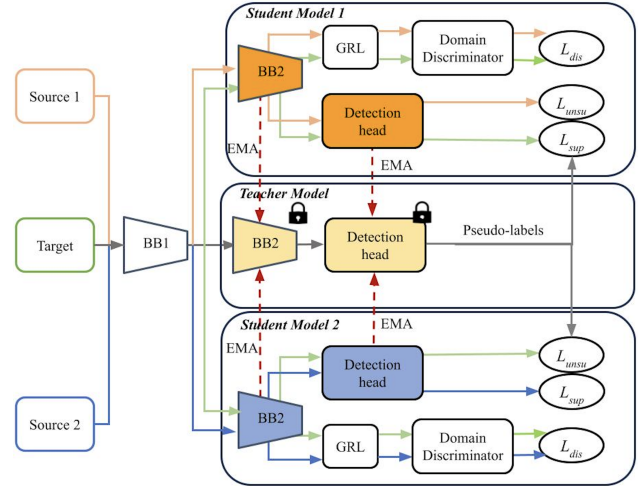
(b) **Divide-and-Merge Spindle Network (DMSN)** [33]. MSDA with multiple student networks and a single teacher network. Domain-invariant features are obtained from the first half of the backbone (BB1). Domain-specific features are obtained from BB2 and the detection head for each source domain.
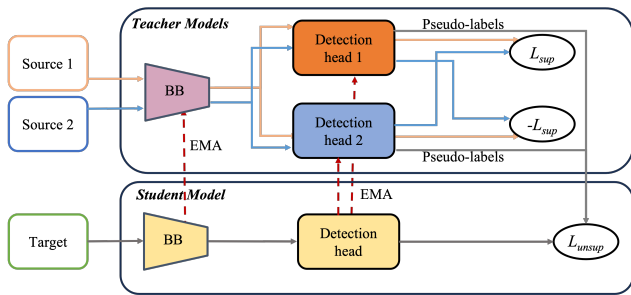
(c) **Target-Relevant Knowledge Preservation (TRKP)** [27]. MSDA with multiple teacher networks and a single student network. Domain-invariant features are obtained from the backbone by reversing detection head gradients. Domain-specific features are obtained from the detection heads.
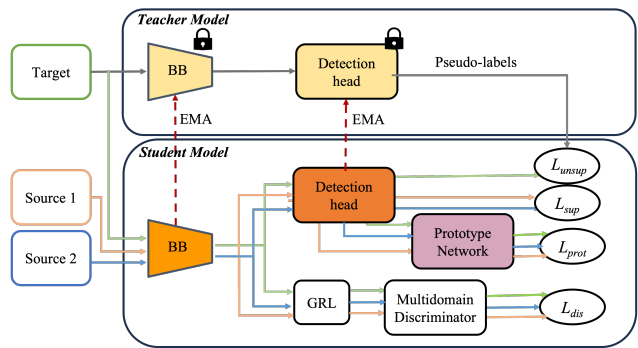
(d) **Our proposed PMT method**. The original mean teacher framework is used with one student and one teacher. Domain-invariant features are obtained from the backbone trained with a discriminator. Domain-specific information is obtained from domain-specific prototypes for each class.

Figure 1. A comparison of the MSDA architectures using the mean-teacher method in the case with two source domains. While state-of-the-art methods require domain-specific parameters (typically a detection head for each source domain) for preserving domain information, our method stores domain-specific information using prototypes for each class and domain.

representation, while the latter part learns a domain-specific representation. This allows the model to align the features from multiple domains while preserving domain-specific information. As illustrated in Fig. 1b, DMSN relies on a mean-teacher training framework, with domain-specific student subnets for each source domain. In contrast, as shown in Fig. 1c, TRKP proposed an adversarial disentanglement method and preserved domain-specific knowledge using a separate detection head for each source domain. The detection results from teacher detection heads are combined to generate pseudo-labels for the student network trained on the target domain. To fully exploit the potential of mul-

tiple sources, both methods advocate for learning domain-specific weights associated with each source domain.

Learning domain-specific weights for each source domain leads to MSDA architectures that are, however, difficult to train for OD since the number of parameters increases rapidly with the number of sources. It has also been observed that the weighted combination of the source domains based on the heuristics of domain similarities is not optimal, where the similarity of each pair of source-target domains is quantified by a weight value [30]. Additionally, DMSN and TRKP focus on finding a common representation space between domains through adversarial train-

ing [33] and adversarial disentanglement [27], yet ignore the class-wise alignment of object categories. The appearance, scale, orientation, and other visual characteristics of objects may vary across domains. If objects in the source and target domains exhibit significant differences, it may be challenging to align the features extracted by the OD model to accurately detect objects in the target domain. An increase in the number of source domains in MSDA exacerbates this problem. In this paper, we propose learning domain-specific class prototype vectors to address these issues. We argue that MSDA can be simplified without learning domain-specific parameters for each source domain. Instead, learning domain-specific class prototype vectors is sufficient to encode domain-specific information for each source domain. Additionally, given the use of prototypes, the number of parameters of our method does not increase significantly with the number of source domains, thereby reducing memory issues and possible overfitting on the source data.

In this paper, a new MSDA method is introduced for OD that is based on mean-teacher [17] and adversarial training (see Fig. 1d). Our Prototype Mean Teacher (PMT) method relies on a cost-effective architecture that stores class and domain-specific information using prototypes. To learn domain-invariant features, a multi-class domain discriminator is integrated into the features generated from the OD backbone. To preserve the domain-specific information of all domains, PMT learns class prototypes using the prototype network for all the domains. Additionally, the prototypes are trained using a contrastive loss to perform a class-conditional and domain-specific adaptation of the detector. The resulting PMT method allows adapting detectors to achieve a higher OD accuracy while being conceptually more straightforward than state-of-the-art methods.

Our main contributions are summarized as follows.
(1) A novel Prototype-based Mean Teacher (PMT) method is introduced for the MSDA of ODs. Our approach shows that prototype vectors can elegantly and efficiently encode domain-specific information.
(2) Using a multi-domain discriminator and a contrastive loss on the prototypes for each domain and class, the proposed PMT can perform class- and domain-conditional adaptation, improving a detector's performance.
(3) Our experimental results and ablations show that the PMT method scales well to the number of source domains and can outperform state-of-the-art MSDA methods for OD on benchmark domain adaptation datasets.

2. Related Works

(a) Unsupervised Domain Adaptation. UDA methods seek to alleviate the problem of domain shift by adapting the detector trained on a labeled source dataset to a target domain using an unlabeled target dataset. Minimiz-

ing the domain discrepancy and adversarial learning are the most popular approaches for UDA. In [3], the authors integrated a domain discriminator into Faster-RCNN to learn domain-invariant feature representations. Later, [22] proposed strong and weak alignment loss. [10] proposed a two-step domain alignment using cycle GAN to mitigate the impact of domain shift between the source and target domains. [32] proposed a graph-based prototype alignment, using contrastive loss. [15] utilized a combination of adversarial training and the mean teacher paradigm. All these methods were designed for UDA from a single source domain. When the source data itself is coming from multiple domains, modeling the domain discrepancy improves the accuracy and robustness of the target OD model [27, 33]. In our work, we focus on MSDA methods for OD that explicitly consider domain discrepancies among source datasets.

(b) Multi-Source Domain Adaptation. In the MSDA setting, the labeled training data belongs to multiple source domains, and the MSDA method aims to distill the knowledge from these source domains to the target domain. Only two methods have been proposed for this setting called DMSN [33] and TRKP [27]. Fundamentally, both methods preserve domain-specific information by utilizing specific subnets for each source domain. Thus the parameters of these methods increase linearly as the number of source domains increases. In our method, we preserve domain-specific information using prototypes for each domain which results in a constant size for the MSDA model regardless of the number of source domains.

(c) Prototype-based Learning. This learning paradigm is used in different forms for open-world OD [11], semi-supervised OD [14], domain-adaptive OD [35], and few-shot OD [26]. In open-world OD, prototypes are used to achieve class separation in the feature space and help unknown class identification [11]. In semi-supervised OD, prototypes are used for class distribution alignment between pseudo-labels and their highly overlapping proposals [14]. In the detection of objects across domains, prototypes are used to align the foreground and background regions between the source and target domains [35]. Few-shot OD used universal prototypes to learn the invariant object characteristics across seen and novel classes [26]. In contrast, we use prototypes in MSDA settings to simplify domain-specific feature learning.

3. Proposed Method

In the MSDA setting, we assume that there are N source domains $S_1, S_2, \ldots, S_N$ and one target domain T. Each source domain can be represented as $S_j = \{(x_i^j, y_i^j)\}_{i=1}^{M_j}$, where $j = 1, 2, \ldots, N$ indicates the source domain, and where M_j indicates the number of images in the source domain j. Here, x_i^j represents the input image i of domain

j, and y_i^j represents its corresponding annotations (bounding boxes and object categories). The target domain T can be represented as $T = \{\{x_i\}_{i=1}^{M_T}\}$, where M_T indicates the number of images in the target domain. Here, x_i represents the input image i in the target domain. In the work, we assume that all domains have the same K object categories.

The detailed architectural diagram of our PMT method is illustrated in Fig. 2. We use the mean teacher framework with unbiased teacher [17] for semi-supervised OD as our base model. Training of our model is performed in two stages. First, a supervised-only training called the burn-in stage is performed to help achieve reliable pseudo-labels on the target domain because the teacher network is initialized using these weights. Second, the model is adapted with labeled data from the source domains and unlabeled data from the target domain. With the source data, we continue using the supervised training (as in the burn-in stage). Target domain pseudo-labels are obtained from the teacher to train the model on the target domain. To produce domain invariant features, a discriminator is introduced in the network. Additionally, the prototype network is integrated to preserve domain-specific information, and class-conditioned adaptation is performed using a contrastive loss. The rest of this section details the individual components of our proposed PMT.

3.1. Supervised Learning

In our problem setting, the annotations are available for the source domains, so we use them to train the model in a supervised way. For this, we compute the standard supervised detection loss of the object detector:

$$\mathcal{L}_{\text{sup}} = \sum_{j=1}^{N} \sum_{i=1}^{M_j} \mathcal{L}_{\text{cls}}(x_i^j, y_i^j) + \mathcal{L}_{\text{reg}}(x_i^j, y_i^j) \qquad (1)$$

where $\mathcal{L}_{\text{cls}}$ is the cross-entropy loss used for bounding box classification, and $\mathcal{L}_{\text{reg}}$ is the smooth-L1 loss used for the bounding box regression as in FRCNN.

3.2. Learning with Pseudo-labels

Since annotations are unavailable for the target domain data, we cannot directly proceed with supervised training as with the source domain. We obtain pseudo-labels for images for the target domain using the mean-teacher method [23]. For that, two augmented versions of the target-domain images are created, called weak and strong augmentations. The weak augmentation is simply the image rescaling and horizontal flip transformation. The strong augmentation includes color jittering, grayscale, Gaussian blur, and cutout patches, which perform only pixel-level transformations. We followed the scale ranges provided in [17] for the strong augmentation. Then, the weak and strong augmented versions are processed by the teacher and student networks,

respectively, as shown in Fig. 2. The predictions made by the teacher model are used as pseudo-labels for the training of the student model. To avoid noisy pseudo-labels from the teacher model, confidence thresholding is applied to the teacher network prediction while computing the pseudo-labels. For this, we followed the same settings as in [17]. The used unsupervised training loss for the target domain using labels from the teacher model is:

$$\mathcal{L}_{\text{unsup}} = \sum_{i=1}^{M_T} \mathcal{L}_{\text{cls}}(x_i, \widetilde{y}_i) + \mathcal{L}_{\text{reg}}(x_i, \widetilde{y}_i) \qquad (2)$$

where $\widetilde{y}_i$ represents the filtered pseudo-labels generated by the teacher model.

3.3. Domain-invariant Features with Discriminator

The feature representation of a model consists of two parts: domain-invariant and domain-specific. In a domain adaptation problem, we try to promote the domain-invariant feature space. To achieve this, UDA methods add a binary domain discriminator to the output of the backbone [3, 15]. This can be extended to MSDA by having multiple such discriminators. But this doesn't consider the domain discrepancy between the source domain and increases the computational cost [18, 36]. Instead of using multiple binary discriminators, we overcome these issues by introducing a multi-class discriminator. This discriminator is connected to the network using a gradient reversal layer (GRL) [7]. It seeks to classify images according to a domain. When this classification loss is back-propagated, the GRL layer reverses the gradients, so the backbone network will try to increase the classification loss, challenging the ability of the discriminator to distinguish between domains. Gradually, the domain of the image-level features extracted by the backbone network will be indistinguishable from the discriminator. This simple adversarial game between the discriminator and the backbone allows us to learn domain-invariant features [3]. The discriminator is trained with the cross-entropy loss between the $\hat{y}^j$, the i-th one-hot vector for the domain j, and the output of the domain discriminator on image x_i^j:

$$\mathcal{L}_{\text{dis}} = -\sum_{j=1}^{N+1} \sum_{i=1}^{M_i} \hat{y}^j \log(D(G(F(x_i^j)))) \qquad (3)$$

where F is the CNN feature extractor, G is the GRL, and D is the multi-domain discriminator. The summation goes from $j = 1, 2, ... N + 1$ as we consider the target domain.

3.4. Domain-specific Features with Prototypes

In our PMT method, prototypes are used to preserve domain-specific information. Preserving domain-specific information helps for distilling with confident pseudo labels
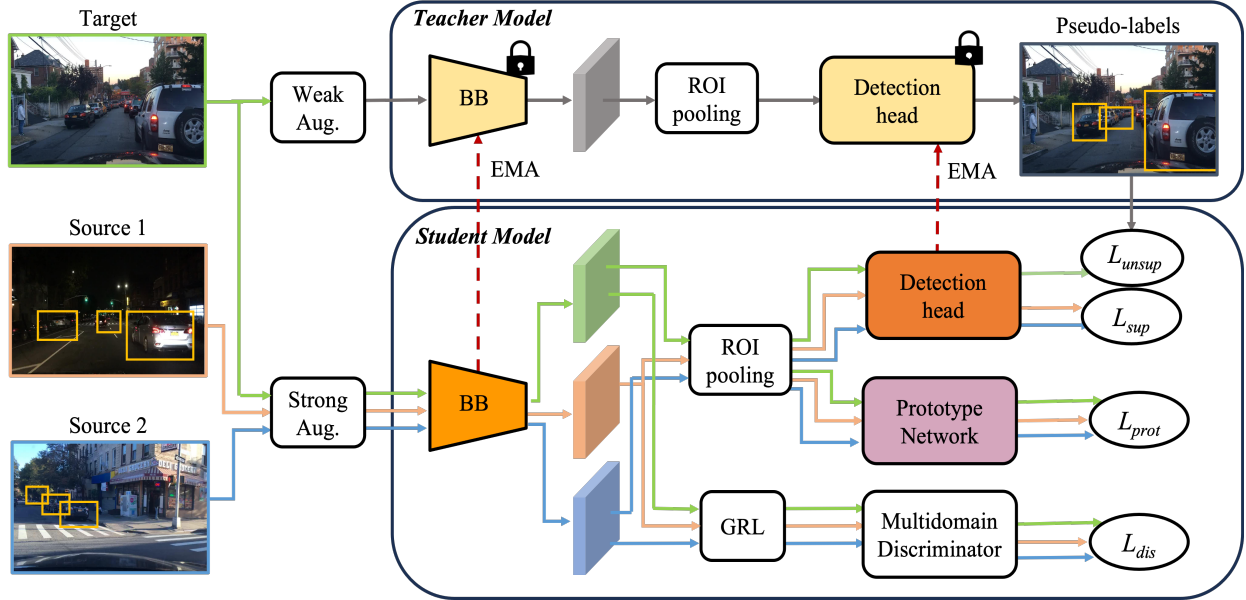
Figure 2. Architectural diagram of the proposed Prototype-based Mean Teacher (PMT) for MSDA. Following the mean-teacher framework, the student is trained with backpropagation, while the teacher is an exponential moving average of the student. The student is trained with images from all domains, and feature alignment is performed at both the image and instance levels using a discriminator and a prototype, respectively. During inference, the teacher model is only employed.

from individual source domains. To obtain the prototypes, we added a prototype network to the output of the ROI pool features. This network maps the output of the ROI pool to a d dimensional feature space, where class-conditioned feature alignment is performed using a contrastive loss. This aligns representations of the same object categories across different domains and separates representations of different object categories. Fig. 3 illustrates the prototypes working for domain-specific feature alignment. We produce a local prototype for each class (e.g., car, truck, bus in Fig. 3) in each domain, and these prototypes preserve the domain-specific information. The prototype for each class "across all domains", called global prototypes, is obtained as the mean of the local prototypes of that class. The contrastive loss on global prototypes pushes them apart in the feature space, resulting in better separation of the classes. It also aligns the local prototypes from all domains for each class, reducing confusion due to appearance variation. Note that the prototype network shares parameters across domains, so its parameter size won't change regardless of the number of domains used. Let p_k^j denote the local prototype of the kth class in domain j. The global prototype of class k is denoted as P_k. Let the number of occurrences of class k from domain j in a minibatch be c_k^j. The prototype update value for class k from domain j in a minibatch is computed as

$$q_k^j = \frac{1}{|c_k^j|} \sum_{r=1}^{c_k^j} P(R(F(x^j), r)) \quad (4)$$

where P is the prototype network (a 2-layer MLP), R is the faster R-CNN ROI-pooling and F is the CNN feature extractor. Thus, $R(F(x^j), r)$ pools features from the RoI r corresponding to the ground-truth box for class k from domain j in a minibatch. For the target domain without ground-truth annotations, the pseudo-labels obtained from the teacher network are used to compute the update value q_k^{N+1}. When $c_k^j = 0$ in a minibatch, no updates are performed for the corresponding prototype. The local prototype vector p_k^j for each class k and domain j is stored in memory along with a count value ρ_k^j that tells how many
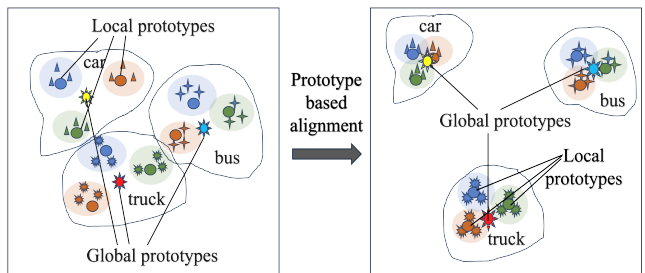

Figure 3. Prototype-based feature alignment with multiple source domains. There are three domains and three classes. Each domain has a prototype for each class. Initially, there is confusion between classes and the intra-class distance to global prototypes from multiple domains is also large. After alignment, class confusion and intra-class distance to global prototypes are reduced.

times p_k^j has been updated thus far. Given the current update value q_k^j, local prototype p_k^j is updated as:

$$p_k^j = \frac{\rho_k^j p_k^j + q_k^j}{\rho_k^j + 1} \qquad (5)$$

The count of updates ρ_k^j is incremented after this operation. After updating the prototype vectors, the contrastive loss is computed in two parts. The first part aligns the prototypes of the same class from different domains by maximizing their similarity. The second part pushes apart the prototypes of different classes to reduce class confusion. To align the prototypes of the same class from different domains, we need to consider pairwise combinations of the domains: $C = Comb(N + 1, 2)$ (including source and target domains). Let $\mathcal{C}$ be the cosine similarity between two vectors. The alignment of category k across different domains is achieved by maximizing the following similarity:

$$\frac{1}{CK} \sum_{k=1}^{K} \sum_{j=1}^{N+1} \sum_{l \neq j}^{N+1} \mathcal{C}(p_k^j, p_k^l) \qquad (6)$$

normalized by the total number of pairwise similarities CK. To push the prototypes of the different classes apart, we minimize their cosine similarity. For this, we first compute the global prototypes from local prototypes of each domain as their weighted mean:

$$P_k = \frac{\sum_{j=1}^{N} \rho_k^j p_k^j}{\sum_{j=1}^{N} \rho_k^j} \qquad (7)$$

where the weight $\frac{\rho_k^j}{\sum_{j=1}^{N} \rho_k^j}$ assigns importance to the prototype of class k from domain j according to its number of occurrence ρ_k^j. Once P_k for each class is computed, we minimize the following similarity function:

$$\sum_{k=1}^{K} \sum_{l \neq k}^{K} \mathcal{C}(P_k, P_l) \qquad (8)$$

Note that prototypes from the target domain are not considered here because of the noise they induce due to the use of pseudo-labels. The final contrastive loss on prototypes is:

$$\mathcal{L}_{\text{prot}} = \sum_{k=1}^{K} \sum_{l \neq k}^{K} \mathcal{C}(P_k, P_l) - \frac{1}{CK} \sum_{k=1}^{K} \sum_{j=1}^{N+1} \sum_{l \neq j}^{N+1} \mathcal{C}(p_k^j, p_k^l) \qquad (9)$$

3.5. Training Summary

The FRCNN model is initialized with Imagenet pretrained weights. The weights of the student model are updated using back-propagation. In contrast, the weights of the teacher model are obtained as the EMA of the student model's weights over time. Training starts with a burn-in training stage, where only the Eq. 1 is used to train the student model. Then, the teacher model is initialized as a copy of the student model, and then the weights of the student model are adapted using the back-propagation, while the teacher weights are the EMA of the student model's weights. In the unsupervised MSDA training step, all the above losses are used jointly to adapt the student model. The total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \alpha \mathcal{L}_{\text{unsup}} + \beta \mathcal{L}_{\text{dis}} + \gamma \mathcal{L}_{\text{prot}} \qquad (10)$$

where hyperparameters α, β, and γ weight the contribution of each loss. For inference, only the teacher model is used to predict output detections.

4. Results and Discussion

4.1. Experimental Methodology

Implementation Details. Our experiments follow the same procedure as in [27, 33]. The FRCNN [21] VGG16 backbone pre-trained with ImageNet was used as the detection framework. Similar to [9], ROI-alignment was used, and the shorter side of the input image was reduced to 600 pixels. For our mean-teacher training, the same setting was used as in [17]. For filtering the pseudo-labels, we set the confidence threshold of 0.7. The model is trained in the burn-in setting for 15 epochs. Then, the model was trained in the unsupervised DA setting for another 15 epochs. For all settings, the value α and β were set to 1 and 0.1, respectively, while the value of γ was set to 1.2, 0.6, and 0.1 for Sections 4.2, 4.3 and 4.4. The weight smoothing coefficient of EMA is set to 0.9996. All the experiments were conducted on 4 A100 GPUs, with batch size 4 and a learning rate of 0.2. Our approach was implemented using Pytorch [19] and Detectron2 [28].

Quantitative comparison. We compared our approach with the following baseline: (1) Source-Only: FRCNN [21] is trained in a supervised manner on source domain data, and no adaptation is performed. (2) UDA Blending: all the source domain data is combined, and UDA methods [15, 17, 22, 29, 32] are used to adapt FRCNN. (3) MSDA: adaptation of FRCNN from independent source datasets. It includes MSDA methods developed for classification [20, 36] and upgraded for OD, as well as MSDA methods developed for OD [33] [27]. (4) Oracle: Target-Only and All-Combined. In the Target-Only case, we performed supervised training of FRCNN only on labeled target data. In the All-Combined case, supervised training is performed on all source and target data combined. Note that some of the baseline results are taken from [33] and [27]. Class-wise AP is reported in the supplementary material.

Setting	Method	D	N	D+N
Source Only	FRCNN [21]	30.4	25.0	28.9
UDA Blending	Strong-Weak [22]	31.4	26.9	29.9
	Graph Prototype [1]	31.8	27.6	30.6
	Cat. Regularization [29]	31.2	28.4	30.2
	UMT [5]	33.8	21.6	33.5
	Adaptive Teacher [15]	34.9	27.8	34.6
MSDA	MDAN [36]	-	-	27.6
	M^{3}SDA [20]	-	-	26.5
	DMSN [33]	-	-	35.0
	TRKP [27]	-	-	39.8
	PMT(ours)	-	-	**45.3**
Oracle	Target-Only	-	-	26.6
	All-Combined	-	-	45.6

Table 1. Detection AP of PMT compared against the baseline, UDA, MSDA, and oracle methods on BDD100K. Source domains are daytime (D) and night (N) subsets and the target is always Dusk/Dawn of BDD100K.

Setting	Method	C	K	C+K
Source Only	FRCNN [21]	44.6	28.6	43.2
UDA Blending	Strong-Weak [22]	45.5	29.6	41.9
	Cat. Regularization [29]	46.5	30.8	43.6
	UMT [5]	47.5	35.4	47.0
	Adaptive Teacher [15]	49.8	40.1	48.4
MSDA	MDAN [36]	-	-	43.2
	M^{3}SDA [20]	-	-	44.1
	DMSN [33]	-	-	49.2
	TRKP [27]	-	-	58.4
	PMT(ours)	-	-	**58.7**
Oracle	Target-Only	-	-	60.2
	All-Combined	-	-	69.7

Table 2. AP for the car class. Our proposed method PMT is compared against the baseline, UDA, MSDA, and oracle methods on the *Daytime* domain of BDD100K dataset. C and K refer to Cityscapes and Kitty datasets.

4.2. Cross Time Adaptation

In this setting, images are collected at different times of the day, so domain shift is caused by changes in illumination. The performance of our method in this setting is evaluated on the BDD100K [34] dataset. The dataset consists of data collected at three different times - Daytime, Dusk/Dawn, and Night Time. In our experiments, Daytime and Night are used as the source domains, and Dusk/Dawn as the target domain. The source domain consists of labeled 64,699 (36,728 daytime + 27,971 night) images. In the target domain, we have 5,027 unlabeled images, that are used for training. The evaluation is done on 778 validation set images of Dusk/Dawn. The mAP on 10 classes is reported.

The mAP performance of our method is compared against the other approaches in Table 1. It can be observed that the UDA methods that blend source datasets are able to increase the performance of the detector compared to the Source-Only baseline. However, this increase in performance is not large, as they disregard the inter-source domain shift. It is also interesting to note that two state-of-the-art MSDA classification methods [20, 36], perform worse than the Source-Only baseline. This degradation in performance shows the fundamental difference in MSDA methods for classification and detection and a strong motivation for MSDA for OD. Our method improves the performance of the best-performing MSDA method [27] by 5.5%. It can also be seen that the performance of our PMT method is much better than the Oracle Target-Only case (because of the small size of the target dataset), and it approached the mAP of the Oracle All-Combined case.

4.3. Cross Camera Adaptation

In this setting, the data are collected using different cameras, and domain shift is caused by changes in the camera's

resolution and viewpoint. In this setting BDD100K [34], Cityscape [4], and Kitty [8] datasets are used. For our experiments, we used Cityscape and Kitty as the source domains, while the Daytime domain of BDD100k was used as the target domain. For both training and evaluation, we only considered the images with car objects. This provides a source domain that consists of 9,515 (2,831 Cityscapes + 6,684 Kitty) labeled images. In the target domain, we have 36,728 unlabeled images used for training. The evaluation is done on 5,258 validation set images of Daytime. The AP is reported only on the car object category in Table 2.

Note that in this experiment, there is only one object category (car), so the prototype separation part was removed for our contrastive loss. It can be observed that the variation in the performance is similar to Section 4.2. Our PMT can outperform state-of-the-art methods, but the increase in AP is lower compared to TRKP. We suppose this is due to the removal of the prototype separation part of our contrastive loss. In ablation studies, we showed that every loss component contributed to obtaining the best PMT performance.

4.4. Extension to Mixed Domain Adaptation

In the MSDA, the domain shift among source data is not always limited to one factor. So, a setting with domain shift with multiple factors is considered for validation. In the source domain, we considered the MS COCO [16], Cityscapes [4], and Synscapes [25] datasets, and the Daytime domain of the BDD100K dataset are the target domain. Among the source domains, the domain shift is mixed and there are more classes, making this study a challenging scenario. Also, the number of source domains is increased from two to three. The source domain has 99,724 (2,975 Cityscapes + 71,749 MS COCO + 25,000 Synscapes) labeled images and, the target domain has 36,728 unlabeled images. The evaluation is performed on 5,258 validation set images of Daytime. For training and evaluation, we

Setting	Method	C	C + M	C + M + S
Source Only	FRCNN [21]	23.4	29.7	30.9
UDA Blending	Unbiased Teach. [17]	-	18.5	25.1
	Adaptive Teach. [15]	-	22.9	29.6
MSDA	TRKP [27]	-	35.3	37.1
	PMT(ours)	-	**38.7**	**39.7**
Oracle	Target-Only	-	-	38.6
	All-Combined	-	47.1	48.2

Table 3. mAP on 7 object categories of our PMT compared against baselines on the *Daytime* domain of BDD100K. C, M, and S refer to Cityscapes, MS COCO, and Synscapes datasets.

only employ the images having 7 common object categories between the datasets. From the results reported in Table 3, we can observe that the Source-Only setting performs better than the UDA Blending setting. This is because of the complex domain shift among the source data. When only Cityscapes and MS COCO are used as the source domain, our method outperforms the previous best-performing method by 3.4%. After adding the Synscapes dataset our method still outperforms other methods, including the Oracle Target-Only results.

4.5. Ablation Study

The cross-time adaptation setting is used in the ablation studies. The two main components of our method are the prototype network and multi-class discriminator. To clearly disentangle the effect of terms in the contrastive loss for prototypes, we considered them as two separate parts: "prototype separation" and "prototype alignment" part. The prototype separation loss increases the distance between different object categories across domains, whereas the prototype alignment loss aligns the same object category across various domains. The experimental results shown in Table 4 show that the results without using the discriminator and prototype are better than the Source-Only result of Table 1. This is due to the effectiveness of the mean-teacher framework in MSDA. By adding the discriminator to the network, performance is increased by 10.2%. This result is already better than the previous state of art MSDA methods for OD. By adding both of our prototype loss terms individually, the performance of the model further increases. Note that using the alignment loss performs better than the separation loss because the separation loss only discriminates the categories, and the vanilla FRCNN is already doing it. Finally, when both the prototype loss terms are combined, the performance is further improved.

4.6. Parameter Growth with Number of Domains

One of the important advantages of not using domain-specific weights is the reduction in the number of parameters to learn. While the existing MSDA detectors have a significant linear growth in parameters with the number

Multidomain Discriminator	Prototype Separation	Prototype Alignment	AP$_{50}$
			31.7
✓			41.9
✓	✓		43.0
✓		✓	43.4
✓	✓	✓	44.6

Table 4. Ablation study on the components of our PMT method.

Method	Number of source domains				
	1	2	3	4	5
DMSN [33]	45.994	75.426	104.858	134.290	163.722
TRKP [27]	45.994	59.942	73.890	87.838	101.786
PMT (ours)	46.586	46.587	46.588	46.589	46.590

Table 5. The growth of m4odel parameter (in millions) as the number of source domains increases. While the parameters of DMSN [33] and TRKP [27] grow quickly, the parameters of our method PMT remain almost constant because the increase is only due to the prototype vectors.

of domains, the parameter growth of our method is negligible as the number of source domains increases. Tab. 5 illustrates this. DMSN [33] makes some part of the feature extractor also domain-specific, thus showing a significantly higher rate of growth compared to other methods. TRKP [27] makes only the detection heads domain-specific, so the growth rate is less compared to DMSN. Our method has only domain-specific prototypes which require fewer parameters per domain than the other approaches.

5. Conclusion

State-of-the-art approaches for MSDA learn domain-invariant and domain-specific parameters for each source domain to effectively adapt DL models from multiple source domains. In this paper, a new MSDA method is proposed, where domain-specific information is encoded using class prototypes. This avoids the need to train a domain-specific subnet for each source domain, simplifying the MSDA architecture significantly. The number of parameters required for our method remains almost constant regardless of the number of source domains, and it is comparable to adaptation with single-source UDA methods, since only prototype vectors are needed for each source domain. Experimental results indicate that our PMT method can effectively exploit multiple domains and outperform state-of-the-art methods on challenging OD baselines.

References

[1] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection, 2019. 1, 7

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[3] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild, 2018. 1, 3, 4

[4] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[5] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection, 2021. 1, 7

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015. 1, 4

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 6

[10] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection, 2019. 3

[11] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, 2021. 3

[12] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G. Macready. A robust learning approach to domain adaptive object detection, 2019. 1

[13] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6091–6100, 2019. 1

[14] Aoxue Li, Peng Yuan, and Zhenguo Li. Semi-supervised object detection via multi-instance alignment with global class prototypes. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[15] 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 *CVPR*, 2022. 1, 2, 3, 4, 6, 7, 8

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 7

[17] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection, 2021. 3, 4, 6, 8

[18] G. Park and S. Wan Lee. Information-theoretic regularization for multi-source domain adaptation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9194–9203, 2021. 4

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[20] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation, 2019. 6, 7

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 6, 7, 8

[22] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection, 2019. 1, 3, 6, 7

[23] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2018. 1, 4

[24] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection, 2019. 1

[25] Magnus Wrenninge and Jonas Unger. Synscapes: A photo-realistic synthetic dataset for street scene parsing, 2018. 7

[26] Aming Wu, Yahong Han, Linchao Zhu, Yi Yang, and Cheng Deng. Universal-prototype augmentation for few-shot object detection. 2021. 3

[27] Jiaxi Wu, Jiaxin Chen, Mengzhe He, Yiru Wang, Bo Li, Bingqi Ma, Weihao Gan, Wei Wu, Yali Wang, and Di Huang. Target-relevant knowledge preservation for multi-source domain adaptive object detection, 2022. 1, 2, 3, 6, 7, 8

[28] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[29] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection, 2020. 6, 7

[30] Minghao Xu, Hang Wang, and Bingbing Ni. Graphical modeling for multi-source domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[31] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *CVPR*, pages 12355–12364, 2020. 1

[32] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment, 2020. 3, 6

[33] Xingxu Yao, Sicheng Zhao, Pengfei Xu, and Jufeng Yang. Multi-source domain adaptation for object detection, 2021. 1, 2, 3, 6, 7, 8

[34] 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, 2020. 7

[35] Yixin Zhang, Zilei Wang, and Yushi Mao. Rpn prototype alignment for domain adaptive object detector. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[36] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 1, 4, 6, 7

[37] Y. Zheng, D. Huang, S. Liu, and Y. Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *CVPR*, 2020. 1

[38] X. Zhou, R. Girdhar, A. Joulin, and P. Krähenbühl andd I. Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 1

[39] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 687–696, 2019. 1