Non-exemplar Domain Incremental Object Detection via Learning Domain Bias

Xiang Song[1], **Yuhang He**[2*], **Songlin Dong**[2], **Yihong Gong**[2]
[1]School of Software Engineering, Xi'an Jiaotong University
[2]College of Artificial Intelligence, Xi'an Jiaotong University
songxiang@stu.xjtu.edu.cn, heyuhang@xjtu.edu.cn, dsl972731417@stu.xjtu.edu.cn, ygong@mail.xjtu.edu.cn

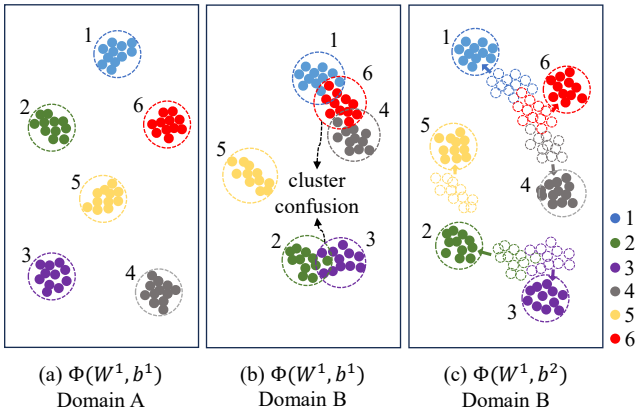
Abstract

Domain incremental object detection (DIOD) aims to gradually learn a unified object detection model from a dataset stream composed of different domains, achieving good performance in all encountered domains. The most critical obstacle to this goal is the catastrophic forgetting problem, where the performance of the model improves rapidly in new domains but deteriorates sharply in old ones after a few sessions. To address this problem, we propose a non-exemplar DIOD method named **learning domain bias** (LDB), which learns domain bias independently at each new session, avoiding saving examples from old domains. Concretely, a base model is first obtained through training during session 1. Then, LDB freezes the weights of the base model and trains individual domain bias for each new incoming domain, adapting the base model to the distribution of new domains. At test time, since the domain ID is unknown, we propose a domain selector based on nearest mean classifier (NMC), which selects the most appropriate domain bias for a test image. Extensive experimental evaluations on two series of datasets demonstrate the effectiveness of the proposed LDB method in achieving high accuracy on new and old domain datasets. The code is available at https://github.com/SONGX1997/LDB.

Introduction

With the rapid development of artificial intelligence, object detection plays an important role in many application fields (Arnold et al. 2019; Li et al. 2021; Zou et al. 2023). However, current object detection methods (Redmon et al. 2016; He et al. 2017; Carion et al. 2020; Li et al. 2022) are quite weak in the face of domain changes. These changes refer to variations in data distribution, attributable to background factors such as picture style, lighting, weather, etc., and object factors such as shape, appearance, and color, etc. All of them hinder the algorithm's capacity to detect objects accurately. Given the continuous emergence of new domains in real-world scenarios, learning new domains while not forgetting knowledge of old ones presents a formidable challenge to existing object detection methods. Therefore, the domain incremental object detection (DIOD) problem (Ding et al. 2023) is a vital but challenging problem in real-world scenarios.

*Corresponding Author



(a) $\Phi(W^1, b^1)$ Domain A (b) $\Phi(W^1, b^1)$ Domain B (c) $\Phi(W^1, b^2)$ Domain B

Figure 1: The illustration of *domain bias*. $\Phi(W^1, b^1)$ is a ViT model pre-trained on domain A, where W^1 and b^1 are its weight and bias, respectively. $\Phi(W^1, b^2)$ is a model obtained by fixing the weight while training the bias on domain B. (a) Testing $\Phi(W^1, b^1)$ on domain A. The model shows good performance. (b) Testing $\Phi(W^1, b^1)$ on domain B. Though the samples of the same class are clustered together, the clusters of different classes are mixed (*i.e.*, the clusters of (1, 4, 6) and (2, 3)), leading to performance drops. (c) Testing $\Phi(W^1, b^2)$ on domain B. Freezing the model weight W^1 and learning a new bias b^2 for the new domain B, the model clusters the samples of the same class as well as separates clusters of different classes.

Compared with the widely studied class incremental learning (CIL) paradigm (Li and Hoiem 2017; Rebuffi et al. 2017; Zhu et al. 2021), the DIOD problem faces two primary challenges: 1) **How to maintain the performances of classification and localization at the same time.** In contrast to CIL methods which focus on the image classification task, the DIOD problem requires the model to simultaneously prevent the forgetting of classification and location of the objects during the incremental process. 2) **How to incrementally learn a unified model handling various data distributions with large domain gaps**. In CIL, the model only needs to adjust the feature space slightly to learn new classes, while DIOD is more challenging, which requires the model to change the entire feature space to adapt to new domains with large gaps in the data distribution.

A plausible approach to address the DIOD problem involves adapting the class incremental object detection (CIOD) methods to the DIOD task. Most existing CIOD methods (Hao et al. 2019; Feng, Wang, and Yuan 2022) leverage knowledge distillation to assimilate new knowledge within a single feature space, thereby mitigating catastrophic forgetting. However, when applied to the DIOD task, these methods result in multiple domains sharing a single feature space, leading to suboptimal results for each individual domain. Another alternative strategy involves introducing state-of-the-art domain incremental classification methods such as L2P (Wang et al. 2022c) and S-Prompt (Wang, Huang, and Hong 2022) to the DIOD task. These methods freeze powerful pre-trained models, appending learnable prompts to input embeddings to learn knowledge from new domains. However, when applied to the DIOD task, due to the frozen backbone, these methods cannot modify the feature space by only fine-tuning prompts. Therefore, it is difficult for them to adapt the feature space to the data distribution of new domains. Furthermore, Ding *et al.* (Ding et al. 2023) have explored the DIOD task and presented an exemplar-based method, saving examples from old domains to mitigate the catastrophic forgetting issue. However, retaining examples from previous datasets may not be advisable due to data security and privacy, and storing such samples inevitably increases storage space.

To address the above challenges, we propose a non-exemplar-based DIOD method by learning *domain bias*, which does not preserve previous old samples. As shown in Figure 1(a), $\Phi(W^1, b^1)$ is a vision transformer (ViT) (Dosovitskiy et al. 2020) model pre-trained on domain A, where W^1 and b^1 are its weight and bias, respectively. When testing it on domain B (Figure 1(b)), although samples of the same class can be clustered together, clusters of different classes are confused (*i.e.*, the clusters of (1,4,6) and (2,3)), leading to performance drops. In Figure 1(c), we freeze the model weight and only learn a new bias b^2 for new domain B. The obtained model $\Phi(W^1, b^2)$ can cluster samples of the same class and separate different clusters in the new domain B. Therefore, we observe that given a pre-trained model, its weights can cluster samples of the same class together and are robust to domain changes, while biases can separate different clusters and are sensitive to domain changes. Inspired by this observation, we can freeze the model's weights and learn independent biases for each domain (i.e., *domain bias*) to adapt the model to the distribution of new domains.

On this basis, we propose a learning domain bias (LDB) method to address the DIOD problem. Concretely, based on state-of-the-art transformer-based object detection method ViTDet (Li et al. 2022), our LDB method trains the base model at the first session without retaining extra parameters. Starting from the second session, LDB freezes the base model and adds individual domain bias for each new domain to adapt the new domain's feature distribution. The domain bias includes two parts: multi-layer perceptron (MLP) bias and multi-head attention (MHA) bias, which are integrated into the base model's backbone. At test time, a domain selector based on nearest mean classifier (NMC) (Mensink et al. 2013; Rebuffi et al. 2017) is devised. We adopt the ImageNet-1K pre-trained model to extract the mean features of each domain training set as the classification head. For an input test image, the Euclidean distance of its features to the classification head is computed to select the most appropriate domain bias. In summary, the principal contributions of this work are as follows:

1) We propose LDB, a novel non-exemplar DIOD method, adapting the base model to the data distribution of continuous new domains by learning individual domain bias for each domain.
2) We design a domain selector via nearest mean classifier, which infers the most appropriate domain bias for an input image at test time.
3) The experimental results on two datasets demonstrate that our LDB method significantly outperforms the state-of-the-art incremental learning methods and domain adaptation methods on the non-exemplar DIOD problem.

Related Work

Incremental Learning

In the field of incremental learning, many works have been devoted to addressing the catastrophic forgetting in class incremental learning (CIL), which has led to its recent surge of attention (Rebuffi et al. 2017; Wu et al. 2019; Zhao et al. 2020; Tao et al. 2020a; Wang et al. 2022a, 2023). A major solution is the rehearsal-based method (Tao et al. 2020a; Yan, Xie, and He 2021; Wang et al. 2022b; Douillard et al. 2022), which utilizes a memory buffer to store some representative samples from old data for replay to prevent old knowledge from being forgotten. However, this method may have privacy and data leakage risks. Therefore, some existing methods focus on the more valuable but challenging non-exemplar CIL problem. LwF (Li and Hoiem 2017) is one of the classic non-exemplar CIL methods, which assigns previous models as teachers and leverages knowledge distillation to alleviate forgetting. PASS (Zhu et al. 2021) memorizes a class representative prototype for each old class to maintain the decision boundary of previous tasks and employs self-supervised learning to learn more general and transferable features.

In addition to CIL, several works (Tao et al. 2020b; Tang et al. 2021; Volpi, Larlus, and Rogez 2021; Xie, Yan, and He 2022) have focused on another representative incremental learning problem—domain incremental learning (DIL), where classes are kept constant but the domains involved often vary a lot in sequence. L2P (Wang et al. 2022c) firstly proposes learning a set of prompts that dynamically inform a pre-trained model to solve corresponding tasks. Subsequently, based on the contrastive language-image pre-training model (Radford et al. 2021), S-Prompt (Wang, Huang, and Hong 2022) learns prompts for each domain independently to solve the catastrophic forgetting problem in DIL.

Recently, some works adapts the CIL methods to class incremental object detection (CIOD), such that achieving class incremental learning in the object detection problem. Similar to CIL, most solutions (Liu et al. 2020; Joseph et al. 2021;

Yang et al. 2023; Liu et al. 2023) are rehearsal-based methods, which preserve old samples and exploit knowledge distillation to mitigate catastrophic forgetting. A few works address the non-exemplar CIOD problem. CIFRCN (Hao et al. 2019) extends the region proposal network and only uses new class images for knowledge distillation. ERD (Feng, Wang, and Yuan 2022) proposes a response-based non-exemplar CIOD method that learns classification head and regression head information, thereby transferring category knowledge.

Source-Free Domain Adaptation

A similar problem setting to our DIOD paradigm is source-free domain adaptation (SFDA). Researches (Chidlovskii, Clinchant, and Csurka 2016; Liang et al. 2019; Kundu et al. 2020; Liang et al. 2021; Yang et al. 2022) utilize a model trained in the source domain to transfer knowledge to an unlabeled target domain. Data from the source domain is not available during this process. MCC (Jin et al. 2020) proposes a loss function for minimal class confusion that does not require source domain data for domain alignment. IRG (VS, Oza, and Patel 2023) designs a contrast loss to represent the object relation of the input in the target domain. These object relations are modeled using an instance-relation graph network and then used to guide contrastive learning.

SFDA aims to leverage knowledge from old domains to improve the discriminative ability of models on new domains, however, the knowledge of the source domain is prone to catastrophic forgetting. In contrast, the goal of our method is to gradually build a unified model that learns knowledge from new domains without forgetting knowledge from old domains, eventually achieving satisfactory object detection performance on both old and new domains.

Bias Tuning

Recently, a series of efficient fine-tuning methods for large language models have been proposed, *e.g.*, Prompt (Lester, Al-Rfou, and Constant 2021), Adapter (Pfeiffer et al. 2021) and LoRA (Hu et al. 2022), etc. Bitfit (Ben Zaken, Goldberg, and Ravfogel 2022) is a competitive method among them, where only the bias term of the model is modified during fine-tuning and other parameters are frozen. Based on this, AdapterBias (Fu et al. 2022) is proposed, which assigns different representation shifts to task-related tokens according to the importance of tokens, so as to obtain better fine-tuning effects. Inspired by this advance, we investigate the value of the bias term for domain incremental learning, and introduce the concept of domain bias in our framework to achieve state-of-the-art performance.

Method

Problem Definition

Assuming T disparate domain datasets $\{\mathbf{D}^1, \mathbf{D}^2, \ldots, \mathbf{D}^T\}$, where each dataset $\mathbf{D}^t = \{\mathbf{X}^t, \mathbf{Z}^t\}$ consists of a training set $\mathbf{X}^t$ and a test set $\mathbf{Z}^t$. All domain datasets consist of objects belonging to the same class. Each training set $\mathbf{X}^t$ can be defined as $\mathbf{X}^t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^{N_x}$, where $\mathbf{x}_i^t$ represents the i-th training example, and $\mathbf{y}_i^t$ refers to a collection of categories and bounding-boxes of objects labeled in $\mathbf{x}_i^t$. Within the non-exemplar DIOD setting, a unified object detection model is designed to learn incrementally from the sequential training sets, without saving any samples from previous domains. At each session t, $\mathbf{X}^t$ constitutes the solely accessible training set, and the model thus acquired will be assessed utilizing the aggregated test sets $\mathbf{Z}^{1\sim t} = \mathbf{Z}^1 \cup \cdots \cup \mathbf{Z}^t$.

Overall Framework

The object detector ViTDet can be considered a combination of a transformer-based feature extractor $f(;\theta)$ and a Mask R-CNN detection head $g(;\vartheta, \phi_c, \phi_b)$. ϕ_c and ϕ_b are class score predictor and bounding box predictor, respectively, both of which are fully connected (FC) layers. ϑ is the rest parameters of Mask R-CNN detection head. We use $\Theta = \{\theta, \vartheta, \phi_c, \phi_b\}$ to represent the total parameters of ViTDet. Initially, loading ImageNet-1K pre-trained model Θ^0, we train a base model Θ^1 using $\mathbf{X}^1$ by classification loss, bounding-box loss and average binary cross-entropy loss (He et al. 2017; Li et al. 2022). Then, we use the same loss functions to incrementally fine-tune the base model on $\mathbf{X}^2, \mathbf{X}^3, \cdots$, and get $\Theta^2, \Theta^3, \cdots$.

Figure 2 illustrates the framework of LDB. The key idea of our method is to freeze the base model and train independent domain bias for each domain, adapting the base model to the data distribution of new domains. During the training process, the base model Θ^1 is obtained at session 1, and no additional parameters need to be learned. Starting from session t ($t > 1$), new predictors ϕ_c^t, ϕ_b^t and domain bias $\mathbf{b}^t$ are added for training, while the base model, as well as predictors and domain biases that are not relevant to the current session, are frozen. When testing, since we do not know which domain the input image belongs to (*i.e.*, the domain ID is unknown), we design a domain selector based on NMC. We adopt the feature extractor of the pre-trained model Θ^0, calculating the mean feature for each domain as the classification head. For a test image, the distance of its feature to the classification head is calculated to determine the domain ID. Thus, the values of $\alpha^2, \alpha^3, \cdots, \alpha^t$ are obtained to select appropriate domain bias and predictors for inference.

Learning Domain Bias

As shown in Figure 2, for each transformer block, the domain bias $\mathbf{b}^t$ appended to the base model consists of MLP bias and MHA bias. Both of them share the same modules: learnable token $v \in \mathbb{R}^C$ and $\gamma \in \mathbb{R}^{H\times W}$, where H, W, and C are the length, width, and number of channels of the feature map, respectively. Similar to the ordinary bias term of neural networks, v is the token added to the output of MLP/MHA to learn domain-specific bias information. γ is generated by a linear layer and is used to weight v. Compared with the ordinary bias term (adding the same bias to each input token), γ can adaptively weight v to add different biases for different input tokens, thereby helping the model to better learn feature distribution of new domains.

At session t, $t > 1$, let the MLP bias be $\mathbf{b}_p^t$, the MHA bias

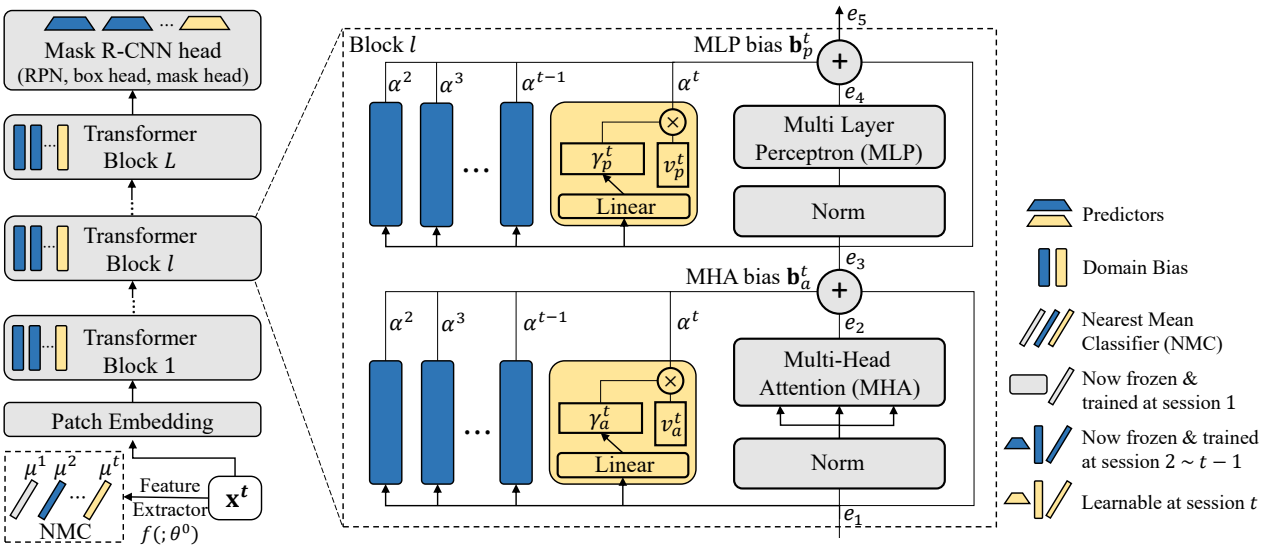


Figure 2: The framework of our proposed LDB. During training, a ViTDet model is first trained at session 1, obtaining the base model (gray parts). Then, at session t ($t >= 2$), additional domain bias and predictor (gold parts) are added for training and α^t is set to 1, while the base model, previously trained domain biases, and predictors (blue parts) are frozen and $\alpha^2, \alpha^3, \cdots, \alpha^{t-1}$ are set to 0. The classification head $\mu^1, \mu^2, \cdots, \mu^t$ of NMC is initialized by $f(;\theta^0)$ at each session. During testing, the feature of a test image is extracted by $f(;\theta^0)$. Then, the distance of the feature to $\mu^1, \mu^2, \cdots, \mu^t$ is calculated, determining the value of $\alpha^2, \alpha^3, \cdots, \alpha^t$ (see Eq.(6) and (7)) to select the most appropriate domain bias and predictor for inference.

be $\mathbf{b}_a^t$. We can calculate $\mathbf{b}_a^t$ as follows:

$$\mathbf{b}_a^t = \alpha^2 b_a^2 + \alpha^3 b_a^3 + \cdots + \alpha^t b_a^t, \tag{1}$$

where $\alpha^2, \alpha^3, \cdots, \alpha^t \in \{0, 1\}$ is used to determine the value of MHA bias for training or inference. When training at session t, b_a^t is trainable and α^t is set to 1, while $b_a^2, b_a^3, \cdots, b_a^{t-1}$ are frozen, and their corresponding $\alpha^2, \alpha^3, \cdots, \alpha^t$ are set to 0, so

$$\mathbf{b}_a^t = b_a^t, \tag{2}$$

where b_a^t can be defined as follows:

$$\begin{aligned} b_a^t &= Linear(e_1) \circ v_a^t = \gamma_a^t \circ v_a^t \\ &= \begin{bmatrix} \gamma_{a,11}^t v_a^t & \gamma_{a,12}^t v_a^t & \cdots & \gamma_{a,1W}^t v_a^t \\ \gamma_{a,21}^t v_a^t & \gamma_{a,22}^t v_a^t & \cdots & \gamma_{a,2W}^t v_a^t \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{a,H1}^t v_a^t & \gamma_{a,H2}^t v_a^t & \cdots & \gamma_{a,HW}^t v_a^t \end{bmatrix}, \end{aligned} \tag{3}$$

where $\gamma_a^t \in \mathbb{R}^{H \times W}$ is the weight used to adjust MHA bias, v_a^t is the MHA bias token, $\circ$ is hadamard product, and $e_1 \in \mathbb{R}^{H \times W \times C}$ is the input embedding of block l. At this point, the output of the MHA becomes $e_3 = e_2 + \mathbf{b}_a^t$, where e_2 is the original output of MHA. Similarly, we can also get the MLP bias:

$$\begin{aligned} \mathbf{b}_p^t &= Linear(e_3) \circ v_p^t = \gamma_p^t \circ v_p^t \\ &= \begin{bmatrix} \gamma_{p,11}^t v_p^t & \gamma_{p,12}^t v_p^t & \cdots & \gamma_{p,1W}^t v_p^t \\ \gamma_{p,21}^t v_a^t & \gamma_{p,22}^t v_p^t & \cdots & \gamma_{p,2W}^t v_p^t \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p,H1}^t v_a^t & \gamma_{p,H2}^t v_p^t & \cdots & \gamma_{p,HW}^t v_p^t \end{bmatrix}, \end{aligned} \tag{4}$$

where γ_p^t is the weight used to adjust MLP bias and v_p^t is the MLP bias token. The output of transformer block l becomes $e_5 = e_4 + \mathbf{b}_p^t$, where e_4 is the original output.

Domain Selector Based on Nearest Mean Classifier (NMC)

In domain incremental learning, the domain ID is not available at test time. Hence, it is essential to infer the domain ID using the input image to select the most appropriate domain bias. To this end, we propose an NMC-based (Mensink et al. 2013; Rebuffi et al. 2017) domain selector that can efficiently judge which domain an input image belongs to. Concretely, in the training stage of session t, we use the feature extractor $f(;\theta^0)$ of the ImageNet-1K pre-trained model to initialize the domain selector:

$$\mu^t = \frac{1}{\|\mathbf{X}^t\|} \sum_{\mathbf{x}^t \in \mathbf{X}^t} Avg(f(\mathbf{x}^t; \theta^0)), \tag{5}$$

where μ^t is the mean feature of domain t, $\mathbf{x}^t$ is a train sample, and Avg performs average pooling on the output features: $\mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^C$. At test time of session t, the feature of a test sample $\mathbf{z} \in \mathbf{Z}^{1 \sim t}$ is extracted by $f(;\theta^0)$. We then select the closest μ to get domain ID:

$$k = \arg\min_i L_2(\mu^i, Avg(f(\mathbf{z}; \theta^0))), i = 1, \cdots, t \tag{6}$$

where k is the domain ID of test sample $\mathbf{z}$, and L_2 is the Euclidean distance. Finally, we can calculate α^j ($j = 2, \cdots, t$) to select the most suitable domain bias:

$$\alpha^j = \begin{cases} 1, & k = j \\ 0. & else \end{cases} \tag{7}$$

Eq.(7) means that if the input $\mathbf{z}$ belongs to the domain $k, k > 1$, the corresponding b_a^k and b_p^k will be appended to model Θ^1 for inference. If $\mathbf{z}$ belongs to the first domain, all α^j ($j = 2, \cdots, t$) are set to 0, and model Θ^1 is used directly for inference.

Predictor Design

As described in Overall Framework, the score predictor ϕ_c^t and bounding box predictor ϕ_b^t are all FC layers, so we can parameterize them as $[W_c^t, b_c^t]$ and $[W_b^t, b_b^t]$, respectively, where $W_c^t \in \mathbb{R}^{n+1}$, $W_b^t \in \mathbb{R}^{4n}$, n is the number of class. Formally, these predictor outputs are calculated by

$$\hat{y} = W_c^t g(f(\mathbf{x}^t; \theta); \vartheta) + b_c^t, \tag{8}$$

$$\hat{o} = W_b^t g(f(\mathbf{x}^t; \theta); \vartheta) + b_b^t, \tag{9}$$

Since each session t has three predictors independent of other sessions, we could construct a predictor pool $P = \{[W_c^t, b_c^t], [W_b^t, b_b^t]\}_{t=1}^{t=T}$ to store them. During inference, when the domain ID is obtained, the corresponding predictors are also selected for class score and bounding box prediction.

Experiments

Benchmarks and Implementation

Datasets. We adopt the Pascal VOC series and BDD100K series dataset to evaluate the effectiveness of our LDB on the DIOD task. The Pascal VOC series consists of datasets from four different domains: Pascal VOC 2007 (Everingham et al. 2010), Clipart, Watercolor, and Comic (Inoue et al. 2018). We choose the same 6 categories of images for the four datasets. Thus, Pascal VOC 2007 contains 3,551 images for training and 3,527 images for testing. Clipart contains 372 images for training and 352 images for testing. Watercolor and Comic contain 1,000 images for training and 1,000 images for testing, respectively.

The BDD100K series consists of autonomous driving datasets from three different domains: BDD100k (Yu et al. 2020), Cityscape (Cordts et al. 2016) and Rainy Cityscape (Hu et al. 2019), each containing 8 types of objects. BDD100K is one of the largest and most diverse publicly available driving datasets. We use 70,000 images from the training set for training, and 10,000 images from the validation set for testing. Cityscape contains 2,975 images for training and 500 for testing. Rainy Cityscape is synthesized from Cityscape using the digital synthesis method to simulate rainy weather, of which 9,432 images are used for training and 1,188 for testing.

Evaluation Metrics. Following (Ding et al. 2023; Yang et al. 2023), we use the average precision (AP) with the IoU threshold=0.5 as a performance metric and report the mean average precision (mAP), *i.e.* the mean AP of all learned sessions. At each session t, after training on X^t, the obtained model Θ^t will be evaluated on the combined test sets $Z^{1\sim t}$. The evaluation results reflect the ability of the model to resist catastrophic forgetting.

Implementation Details. We adopt ViTDet (Li et al. 2022) with 12 transformer blocks and 768 channel dimensions as the backbone. The model is pre-trained by ImageNet-1K and MAE (He et al. 2022). We train the model for 20 epochs (5 warm up epochs) using AdamW (Loshchilov and Hutter 2018) optimizer with a weight decay of 0.1. The learning rate is set to 2e-4, training batch size is set to 2, and input size is set to 1,024. More implementation details are provided in the **Appendix**.

Methods	Buffer Size	Session 2	Session 3	Session 4
Upper-bound	-	72.6	69.4	67.6
TP-DIOD-B	150/domain	65.8	62.1	57.5
FT-seq	0/domain	57.5	52.6	49.5 (↓ 7.3)
FT-FC		59.1	54.4	44.2 (↓ 12.6)
MCC		47.6	34.4	23.7 (↓ 33.1)
IRG		51.5	43.7	33.2 (↓ 23.6)
LwF		60.4	53.6	53.2 (↓ 3.6)
PASS		61.7	51.4	49.8 (↓ 7.0)
L2P		59.9	55.2	45.5 (↓ 11.3)
S-Prompt		59.4	54.3	45.0 (↓ 11.8)
CIFRCN		65.3	57.7	53.5 (↓ 3.3)
ERD		58.9	50.7	48.7 (↓ 8.1)
LDB(Ours)	0/domain	**68.1**	**64.2**	**56.8** (↓ 0.0)

Table 1: Experimental results (mAP) on Pascal VOC series.

Methods	VOC	Clipart	Watercolor	Comic	mAP	Param
L2P	78.2	32.5	43.3	27.8	45.5	0.33%
S-Prompt	80.8	33.9	45.2	20.1	45.0	0.24%
LDB(Ours)	**82.4**	**50.1**	**57.5**	**37.0**	**56.8**	**0.19%**

Table 2: Comparison of our LDB with prompt-based methods at session 4 on Pascal VOC series.

Comparison Methods

For comparison with the proposed LDB, we select several non-exemplar incremental learning methods, including: the CIL methods LwF (Li and Hoiem 2017) and PASS (Zhu et al. 2021), the DIL methods L2P (Wang et al. 2022c) and S-Prompt (Wang, Huang, and Hong 2022), the CIOD methods CIFRCN (Hao et al. 2019) and ERD (Feng, Wang, and Yuan 2022), where PASS, S-Prompt and ERD are best-performing methods in their fields. We also choose the exemplar-based DIOD method TP-DIOD (Ding et al. 2023) for comparison. Moreover, we implement the classic SFDA method MCC (Jin et al. 2020) and the state-of-the-art SFDA method IRG (VS, Oza, and Patel 2023) in the DIOD problem. We add labels to these SFDA methods in the new domains for a fair comparison. In addition, to prove the relative effectiveness of all methods, FT-seq (fine-tuning sequence training), FT-FC (fine-tuning predictors) and Upper-bound (fully supervised training) are conducted. In the ablation experiments, we compare our NMC-based domain selector with the K-means and K-NN (K&K in short) selector adopted in S-Prompt, where $k = 5$. All these methods use the same ViTDet as the backbone, and the specific implementation details are introduced in the **Appendix**.

Comparison Results

Results on Pascal VOC Series. Table 1 reports the detailed comparison results on Pascal VOC series, which are learned in the order of Pascal VOC 2007 → Clipart → Watercolor → Comic. This series of datasets includes 4 different styles of images, so we can evaluate the anti-forgetting ability of the model for different domains. Figure 3(a) shows the comparison curves of the 12 non-exemplar-based methods. It is

Methods	Buffer Size	Session 2	Session 3
Upper-bound	-	57.0	58.5
TP-DIOD-B	200/domain	53.4	51.5
FT-seq	0/domain	51.6	43.6 (↓ 7.5)
FT-FC		51.3	48.0 (↓ 3.1)
MCC		44.3	36.1 (↓ 15.0)
IRG		49.3	38.7 (↓ 12.4)
LwF		52.1	44.1 (↓ 7.0)
PASS		51.7	43.3 (↓ 7.8)
L2P		51.5	47.7 (↓ 3.4)
S-Prompt		51.6	49.4 (↓ 1.7)
CIFRCN		51.8	48.9 (↓ 2.2)
ERD		51.1	48.1 (↓ 3.0)
LDB(Ours)	0/domain	**52.3**	**51.1** (↓ 0.0)

Table 3: Experimental results (mAP) on BDD100K series.

Methods	BDD100K	Cityscape	Rainy	mAP	Param
L2P	49.7	48.8	44.8	47.7	0.30%
S-Prompt	**51.6**	52.0	44.7	49.4	0.16%
LDB(Ours)	50.3	**52.7**	**50.2**	**51.1**	**0.13%**

Table 4: Comparison of our LDB with prompt-based methods at session 3 on BDD100K series.

observed from the table and figure that our LDB outperforms all other state-of-the-art non-exemplar-based methods at each encountered session, and even surpasses exemplar-based method TP-DIOD-B at session 2 and 3. Concretely, LDB achieves **68.1%**, **64.2%** and **56.8%** mAP at session 2, 3, and 4, exceeding the second best non-exemplar-based method CIFRCN by **2.8%**, **6.5%** and **3.3%**, respectively.

Table 1 also compares the LDB and SFDA methods. We observe that the SFDA methods perform poorly on the DIOD task even with labels of new domains. Our LDB outperforms both MCC and IRG by a large margin.

Table 2 illustrates the comparison of LDB with recent prompt-based methods L2P and S-Prompt at session 4. Accuracies tested on Clipart, Watercolor, and Comic reflect the ability of these methods to adapt the base model to new domains. Compared to the second best method L2P, LDB improves the accuracy by **4.2%** (**78.2%**→**82.4%**), **17.6%** (**32.5%**→**50.1%**), **14.2%** (**43.3%**→**57.5%**) and **9.2%** (**27.8%**→**37.0%**) in the four domains, respectively. Furthermore, our method uses minimal extra parameters (**0.19%**). This proves that compared to the prompts, fine-tuning domain bias effectively changes the feature space of the model to adapt to the data distribution of new domains.

Results on BDD100K series. Table 3 summarizes the comparison results of 13 methods on the BDD100K series, which are learned in the order of BDD100K → Cityscape → Rainy Cityscape. These three datasets contain different autonomous driving scenarios, which can verify the anti-forgetting ability of our method for the actual environment. From this table, we observe similar conclusions as the Pascal VOC series. LDB surpasses the second best non-exemplar method S-Prompt by **0.7%** (**51.6%**→**52.3%**) and **1.7%** (**49.4%** →**51.1%**) at session 2 and 3, respectively.

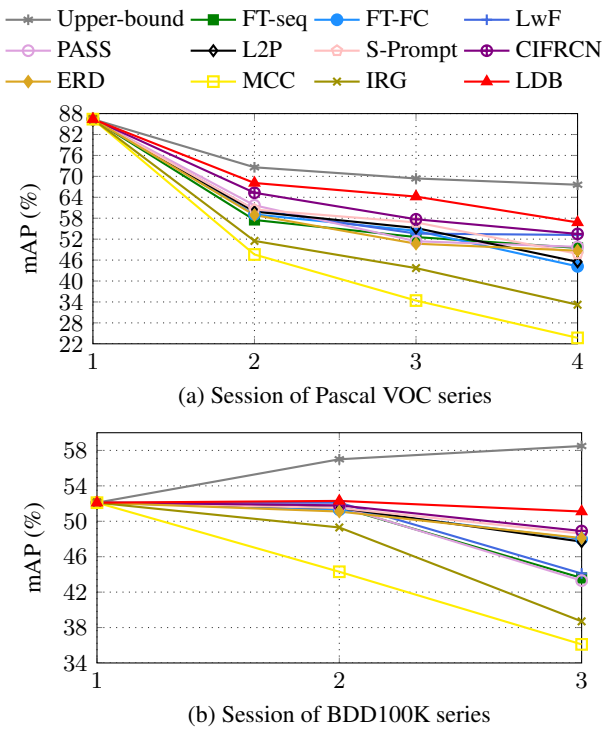


Figure 3: Performance curves of the 12 non-exemplar-based methods with respect to session $1 \sim T$. (a) Pascal VOC series; (b) BDD100K series.

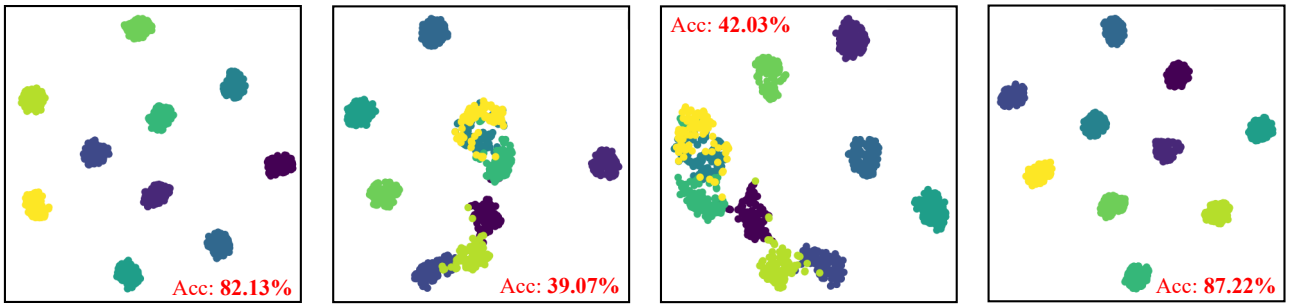
Table 4 presents the comparison results of LDB with L2P and S-Prompt at the last session of BDD100K series. Similar to Table 2, our LDB has the highest accuracies of **52.7%** and **50.2%** on Cityscape and Rainy Cityscape, respectively. Using the least extra parameters (**0.13%**), it is **0.7%** and **5.5%** higher than S-Prompt, respectively.

Figure 3(b) shows the comparison curves of 12 non-exemplar-based methods on the BDD100K series. As incremental learning proceeds, the superiority of LDB becomes more pronounced, demonstrating the anti-forgetting ability of old domain knowledge.

Ablation Study

The Effect of Each Component. As shown in Table 5, to demonstrate the impact of each proposed component, we conduct the ablation studies by building the following five models on Pascal VOC series: (1) Using the FT-FC model and NMC domain selector as a baseline, in which only the predictors are trainable; (2) Adding the MLP bias to the baseline model; (3) Adding the MHA bias to the baseline model; (4) Adding both the MLP bias and MHA bias to FT-FC model and adopting K&K to select domain ID (Wang, Huang, and Hong 2022); (5) Adding both the MLP bias and MHA bias to the baseline model, which is equivalent to the proposed LDB method.

As expected, in row 1, the baseline model produces the lowest mAP of **37.6%** at session 4. In rows 2 and 3, adding



(a) Model Θ^1, SVHN (b) Model Θ^1, MNIST (c) Model Θ^1 with 0 bias, MNIST (d) Model Θ^2, MNIST

Figure 4: Validation of domain bias using ViT on SVHN and MNIST dataset.

MLP bias	MHA bias	K&K	NMC	Session 2	3	4
×	×	×	√	59.1	54.4	44.2 (↑ 0.0)
√	×	×	√	67.3	62.9	55.3 (↑ 11.1)
×	√	×	√	67.6	63.2	55.6 (↑ 11.4)
√	√	√	×	67.4	62.9	55.9 (↑ 11.7)
√	√	×	√	**68.1**	**64.2**	**56.8** (↑ 12.6)

Table 5: Ablation study of the contribution of each component on Pascal VOC series.

Blocks	VOC	Clipart	Watercolor	Comic	mAP
10 ∼ 12	**84.7**	42.6	51.1	27.3	51.4 (↑ 0.0)
7 ∼ 12	83.6	50.2	54.0	32.8	55.2 (↑ 3.8)
4 ∼ 12	82.4	**51.4**	56.3	36.2	56.6 (↑ 5.2)
1 ∼ 12	82.4	50.1	**57.5**	**37.0**	**56.8** (↑ 5.4)

Table 6: Comparison results of adding domain bias to different transformer blocks at session 4 on Pascal VOC series.

MLP bias and MHA bias to the baseline contributes **11.1%** and **11.4%** relative improvement, respectively. In row 5, it can be observed that adding two biases leads to the highest relative improvement of **12.6%**. In addition, comparing row 4 and row 5, we find that under the same model, our NMC-based domain selector achieves a relative improvement of **0.9%** compared with the K&K used in S-Prompt. These experimental results strongly demonstrate that MLP bias, MHA bias, and NMC-based domain selector are both very effective for improving the performance of LDB.

The Effect of Adding Domain Bias to Different Blocks. In order to explore the optimal domain bias addition method, we gradually increase the number of transformer blocks with domain bias, changing from the last three blocks (10∼12) to all blocks. The performance of session 4 on Pascal VOC series is reported in Table 6. As expected, adding domain bias for blocks 10∼12 yields the lowest mAP (51.4%). When adding domain bias to the last six and nine blocks, the performance is improved by **3.8%** and **5.2%**, respectively. Adding domain bias to all blocks leads to the best performance (**56.8%**). This proves that more domain biases can better help the base model adapt to the data distribution of the new domain, achieving better performance.

Validation of Domain Bias. To verify the role of domain bias, we use ViT to conduct domain incremental classification experiments and visualize the output features by TSNE (Van der Maaten and Hinton 2008). First, as shown in Figure 4(a), we train a base model Θ^1 on the SVHN (Netzer et al. 2011) dataset with an accuracy of 82.13%. Then, we test the model Θ^1 on the MNIST dataset, only getting 39.07% accuracy. Figure 4(b) shows that samples of the same class are still clustered, but some clusters are mixed together, leading to a drop in accuracy. Next, we set the bias of Θ^1 to 0 and test it on the MNIST, getting an accuracy of 42.03%. Figure 4(c) proves that clustering of samples can be achieved only by weights regardless of domains, *i.e.* weights are robust to domain changes. Finally, we only fine-tune the bias of the model Θ^1 on the MNIST dataset, getting the model Θ^2 which achieves an accuracy of **87.22%** on MNIST (Figure 4(d)), an improvement of more than **45%** (**39.07%/42.03%** $\rightarrow$ **87.22%**). The above visualization results demonstrate that the weights of ViT can cluster the objects independent of the domains, while the biases can separate different clusters and are sensitive to domain changes. Therefore, we can only fine-tune the bias to adapt the model to the data distribution of new domains.

Conclusion

This paper focuses on an important yet challenging problem termed domain incremental object detection (DIOD), where the categories of objects remain constant but the involved domains vary greatly in order. To alleviate catastrophic forgetting in DIOD, we propose a non-exemplar method namely learning domain bias (LDB). A ViTDet model is first trained normally at session 1. Since the second session, we train individual domain bias for each new domain, adapting the base model to the distribution of these domains. At test time, we propose an NMC-based domain selector to choose the most appropriate domain bias for a test image. Extensive experimental results on PASCAL VOC series and BDD100K series datasets show that our LDB significantly outperforms existing state-of-the-art methods.

Acknowledgments

This work was funded by the National Key Research and Development Project of China under Grant No. 2020AAA0105600, and by the National Natural Science Foundation of China under Grant No. U21B2048 and No. 6230070870.

References

Arnold, E.; Al-Jarrah, O. Y.; Dianati, M.; Fallah, S.; Oxtoby, D.; and Mouzakitis, A. 2019. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10): 3782–3795.

Ben Zaken, E.; Goldberg, Y.; and Ravfogel, S. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 1–9. Dublin, Ireland: Association for Computational Linguistics.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chidlovskii, B.; Clinchant, S.; and Csurka, G. 2016. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 451–460.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Ding, L.; Song, X.; He, Y.; Wang, C.; Dong, S.; Wei, X.; and Gong, Y. 2023. Domain Incremental Object Detection Based on Feature Space Topology Preserving Strategy. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Douillard, A.; Ramé, A.; Couairon, G.; and Cord, M. 2022. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9285–9295.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

Feng, T.; Wang, M.; and Yuan, H. 2022. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9427–9436.

Fu, C.-L.; Chen, Z.-C.; Lee, Y.-R.; and Lee, H.-y. 2022. AdapterBias: Parameter-efficient Token-dependent Representation Shift for Adapters in NLP Tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 2608–2621. Seattle, United States: Association for Computational Linguistics.

Hao, Y.; Fu, Y.; Jiang, Y.-G.; and Tian, Q. 2019. An End-to-End Architecture for Class-Incremental Object Detection with Knowledge Distillation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, X.; Fu, C.-W.; Zhu, L.; and Heng, P.-A. 2019. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8022–8031.

Inoue, N.; Furuta, R.; Yamasaki, T.; and Aizawa, K. 2018. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5001–5009.

Jin, Y.; Wang, X.; Long, M.; and Wang, J. 2020. Minimum class confusion for versatile domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 464–480. Springer.

Joseph, K.; Khan, S.; Khan, F. S.; and Balasubramanian, V. N. 2021. Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5830–5840.

Kundu, J. N.; Venkat, N.; Babu, R. V.; et al. 2020. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4544–4553.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, 280–296. Springer.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Li, Z.; Liu, F.; Yang, W.; Peng, S.; and Zhou, J. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.

Liang, J.; He, R.; Sun, Z.; and Tan, T. 2019. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2975–2984.

Liang, J.; Hu, D.; Wang, Y.; He, R.; and Feng, J. 2021. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, X.; Yang, H.; Ravichandran, A.; Bhotika, R.; and Soatto, S. 2020. Multi-task incremental learning for object detection. *arXiv preprint arXiv:2002.05347*.

Liu, Y.; Schiele, B.; Vedaldi, A.; and Rupprecht, C. 2023. Continual detection transformer for incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23799–23808.

Loshchilov, I.; and Hutter, F. 2018. Fixing weight decay regularization in adam.

Mensink, T.; Verbeek, J.; Perronnin, F.; and Csurka, G. 2013. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2624–2637.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Pfeiffer, J.; Kamath, A.; Rücklé, A.; Cho, K.; and Gurevych, I. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 487–503.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Tang, S.; Su, P.; Chen, D.; and Ouyang, W. 2021. Gradient regularized contrastive learning for continual domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2665–2673.

Tao, X.; Chang, X.; Hong, X.; Wei, X.; and Gong, Y. 2020a. Topology-preserving class-incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, 254–270. Springer.

Tao, X.; Hong, X.; Chang, X.; and Gong, Y. 2020b. Bi-objective continual learning: Learning 'new'while consolidating 'known'. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5989–5996.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Volpi, R.; Larlus, D.; and Rogez, G. 2021. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4443–4453.

VS, V.; Oza, P.; and Patel, V. M. 2023. Instance relation graph guided source-free domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3520–3530.

Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022a. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, 398–414. Springer.

Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022b. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, 398–414. Springer.

Wang, S.; Shi, W.; Dong, S.; Gao, X.; Song, X.; and Gong, Y. 2023. Semantic Knowledge Guided Class-Incremental Learning. *IEEE Transactions on Circuits and Systems for Video Technology*.

Wang, Y.; Huang, Z.; and Hong, X. 2022. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35: 5682–5695.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022c. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 139–149.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 374–382.

Xie, J.; Yan, S.; and He, X. 2022. General incremental learning with domain-aware categorical representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14351–14360.

Yan, S.; Xie, J.; and He, X. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3014–3023.

Yang, D.; Zhou, Y.; Hong, X.; Zhang, A.; and Wang, W. 2023. One-Shot Replay: Boosting Incremental Object Detection via Retrospecting One Object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 3127–3135.

Yang, S.; Wang, Y.; Wang, K.; Jui, S.; et al. 2022. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems*.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13208–13217.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5871–5880.

Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; and Ye, J. 2023. Object detection in 20 years: A survey. *Proceedings of the IEEE*.