Parameterized Prompt for Incremental Object Detection

Zijia An[1,2] Boyu Diao[1,2*] Ruiqi Liu[1,2] Libo Huang[1,2] Chuanguang Yang[1,2] Fei Wang[1,2]
Zhulin An[1,2] Yongjun Xu[1,2]
[1]State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{anzijia23p, diaoboyu2012, yangchuanguang, wangfei, anzhulin, xyj}@ict.ac.cn
huanglibo@gmail.com liuruiqi23@mails.ucas.ac.cn

Abstract

Recent studies have demonstrated that incorporating trainable prompts into pretrained models enables effective incremental learning. However, the application of prompts in incremental object detection (IOD) remains underexplored. Our study reveals that existing prompt-pool-based approaches assume disjoint class sets across incremental tasks, which are unsuitable for IOD as they overlook the inherent co-occurrence phenomenon in detection. In co-occurring scenarios, unlabeled objects from previous tasks may appear in current task images, leading to confusion in prompts pool. In this paper, we hold that prompt structures should exhibit adaptive consolidation properties across tasks, with constrained updates to prevent confusion and catastrophic forgetting. Motivated by this, we introduce Parameterized Prompts for Incremental Object Detection (P^2IOD). Leveraging neural networks global evolution properties, P^2IOD employs networks as the parameterized prompts to adaptively consolidate knowledge across tasks. To constrain prompts structure updates, P^2IOD further engages a parameterized prompts fusion strategy. Extensive experiments on PASCAL VOC2007 and MS COCO datasets demonstrate that P^2IOD's effectiveness in IOD and achieves the state-of-the-art performance among existing baselines. Code is available at https://github.com/EMLS-ICTCAS/P2IOD.

1. Introduction

In response to external changes, humans possess strong adaptability, allowing them to incrementally accumulate knowledge [11, 12]. Similarly, we expect object detection algorithms to learn in an incremental manner. Existing detection methods suffer from catastrophic forgetting [26] during incremental learning. This issue arises because current detection frameworks rely on predefined labeled datasets [50], implicitly assuming static data distributions. When learning from dynamic data distributions, these frameworks tend to forget previously learned knowledge [35], resulting in severe performance degradation.

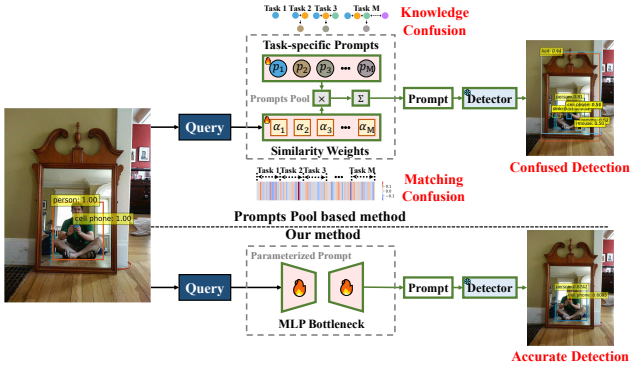

Figure 1. Existing prompts-pool-based methods store task-specific prompts learned from different tasks and match objects to their most relevant prompts based on similarity weights during inference. However, preserving knowledge in a task-isolated manner causes confusion in IOD. In contrast, our method redesigns the prompts pool as parameterized prompts to holistically preserve and update knowledge across tasks, thus mitigating confusion.

To address this challenge, many incremental object detection (IOD) methods [7, 17, 25, 27, 33] exploit the inherent co-occurrence phenomenon, where detection images typically contain both labeled objects from the current task and unlabeled objects from previous tasks. In such co-occurring scenarios, object distribution remains relatively static [1], providing latent knowledge to supplement previous tasks. A key problem in IOD thus lies in effectively leveraging the static object distribution present in co-occurring scenarios. Recently, with the rise of pre-trained models [4], prompting has emerged as a promising direction for incremental learning [14, 19]. Yet, its suitability for IOD's co-occurring scenarios remains unexplored. Gaurav et al. [2] first introduce prompting into IOD, adopting

*Corresponding Author.

a well-established prompts pool from incremental classification. We observe that the prompts pool exhibits confusion when incorporating the knowledge of the static object distribution in co-occurring scenarios, leading to a negative impact on performance.

An ideal prompts pool stores task-specific prompts learned from different tasks and matches objects to its most relevant prompts based on similarity weight during inference [38]. However, when leveraging the static distribution of objects in co-occurring scenarios, the prompts pool encounters severe confusion, specifically manifesting as matching and task confusion. As shown in Fig. 1, the former matching confusion refers to an object that cannot match the most relevant prompt. We visualize the similarity weight between an object and different prompts. It can be observed that since the object appears across all tasks, they exhibit high similarity with all task-specific prompts, making it impractical to match the most relevant prompt. On the other hand, the task confusion refers to task-specific prompts learning knowledge outside of its tasks. The unlabeled previous objects in co-occurring scenarios provide latent knowledge, causing the prompts learned for the current task to incorporate knowledge from all previous tasks, which undermines the clarity of the prompt's representation. We refer to the matching and task confusion introduced by the prompts pool in IOD as prompts pool confusion, which negatively affects IOD's performance.

To tackle the above problems, this paper proposes **P**arameterized **P**rompt for **I**ncremental **O**bject **D**etection (P^2IOD). We argue that preserving knowledge in a task-isolated manner leads to confusion when handling co-occurring scenarios in IOD. To overcome the confusion, we advocate that the structure for preserving prompt knowledge should exhibit an adaptive consolidation property, ensuring that knowledge is preserved holistically across tasks, while previous task knowledge can be dynamically updated in co-occurring scenarios. Building upon this, we further mitigate catastrophic forgetting by constraining updates to critical parameters within the prompt structure. Based on this insight, P^2IOD redesigns the prompts pool as a parameterized multi-layer perceptron (MLP), so as to leveraging the adaptive consolidation property inherent in neural networks, which naturally update learned knowledge in response to losses from co-occurring objects. We interpret the constraint on parameterized prompts as a form of model fusion, where the parameters of previous and current prompts are preserved or merged based on their importance and consistency, ensuring that the knowledge from each task is retained. In addition, we introduce pseudo-labeling to mine latent knowledge from co-occurring objects.

Our contributions can be summarized as follows.

(i) This is the first work to investigate the prompts pool confusion caused by the co-occurrence phenomenon. We further advocate that prompt structures should exhibit an adaptive consolidation property and adopt constrained updates to prevent confusion and forgetting.

(ii) We propose P^2IOD, which redesigns the prompts pool as parameterized prompts and employs parameterized prompt fusion to constrain parameter updates.

(iii) Extensive experiments on PASCAL VOC2007 and MS COCO datasets demonstrate the effectiveness of the proposed method in IOD, achieving state-of-the-art performance in existing baselines.

2. Related Work

2.1. Incremental Learning

In recent years, the strong generalization ability of pre-trained models (PTM) injects new vitality into incremental learning [46]. A promising approach is to freeze the PTM's parameters and add learnable lightweight prompts to adjust the PTM [15, 32, 37, 38]. However, the learnable prompts also face the challenging issue of catastrophic forgetting. L2P [38] and DualPrompt [37] design a prompts pool to store task-specific prompts trained under different tasks. During inference, the top-K most relevant prompts are selected through an instance-wise query mechanism, thereby alleviating the catastrophic forgetting caused by updating prompts. CodaPrompt [32] replaces the top-K selection criterion with a more natural selection mechanism, using a learnable linear combination to determine the contribution of the prompts. DAP [15] uses an MLP to generate finer-grained prompts for each instance and utilizes a prompts pool to store conditional input embeddings that supplement task-specific information. The above prompt-based methods are discussed in the context of incremental image classification tasks and show remarkable results, but their applicability in more complex incremental object detection tasks is still not fully established.

2.2. Incremental Object Detection

The distinction between incremental object detection and other incremental tasks lies in the co-occurrence phenomenon inherent in detection scenarios. Co-occurring scenarios contain numerous unlabeled objects from previous tasks that can supplement previous task knowledge. Knowledge distillation [10, 13, 23, 39, 44] provides a flexible way to mine previous task knowledge. Such approaches [3, 7, 31, 41, 42] employ the original detector to regularize the outputs and intermediate features of the incremental detector, thereby facilitating the transfer of knowledge in training data from the original to the incremental detector. As this knowledge inherently contains information about unlabeled objects, it enables the implicit mining of unlabeled objects within the co-occurring scenarios. Furthermore, some methods [2, 17, 43] explicitly mine unla-

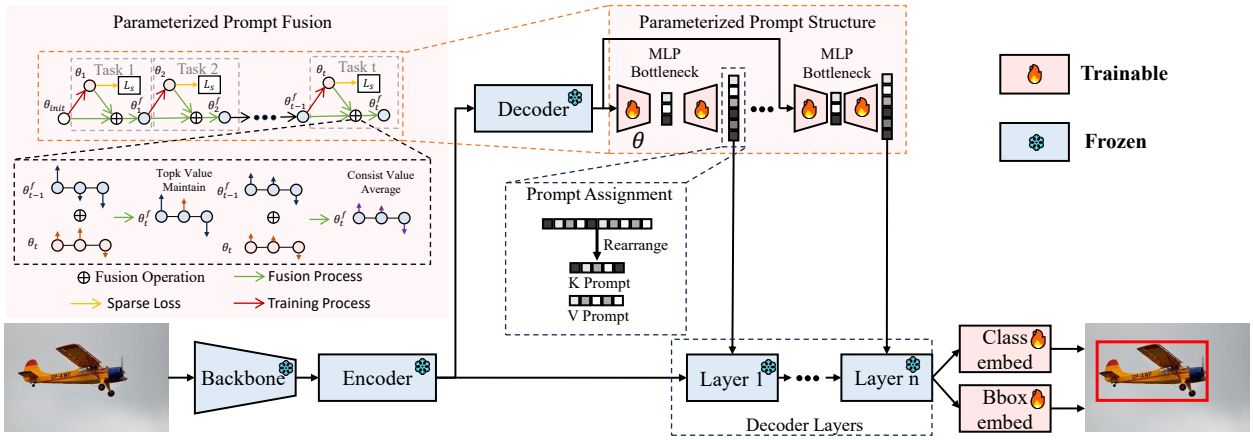

Figure 2. The overall framework of P^{2}IOD. To address the issue of prompts pool confusion, P^{2}IOD redesigns the prompts pool as a parameterized prompt structure consisting of multi-layer perceptron (MLP) bottlenecks. To further alleviate catastrophic forgetting, P^{2}IOD proposes a parameterized prompt fusion strategy, which adds an additional fusion process after each incremental training process.

beled objects through pseudo-labeling. These methods use the original detector to label the objects in the training data and subsequently filter out incorrect labels based on specific criteria. The static distribution labels obtained through the pseudo-labeling method allow the detectors to be immune to catastrophic forgetting. The above methods exhibit strong performance in co-occurring scenarios.

With the rise of PTM, the prompting of the PTM has become a promising direction for incremental learning. Prompting of the PTM stores prompts as an additional memory module, allowing the PTM to learn and retain relevant information from each incremental task. Gaurav [2] constructs a prompts pool to store the prompts learned from different tasks and matches the most relevant prompt during inference. However, we discover that introducing a prompts pool faces severe prompts pool confusion in co-occurring scenarios. Therefore, effectively incorporating prompts into PTM still requires further research in IOD.

3. Preliminaries

3.1. Object Detection Baseline

We introduce parameterized prompts on the transformer-based Deformable-DETR [47] and Co-DETR [49] to validate our motivation. In transformer-based object detection frameworks, there exist two types of attention mechanisms. The first is the multi-head attention mechanism [34] in Transformers. Given a query element and a set of key elements, the multi-head attention module obtains attention weights based on the similarity between the query-key pairs and adaptively aggregates important features according to the attention weights. To enable the model to focus on content from different representational subspaces and different positions, the multi-head attention mechanism combines the outputs of multiple attention heads with different learnable weights. Let $q \in \Omega_q$ indexes a query element with representation feature $z_q \in \mathbb{R}^C$, and $k \in \Omega_k$ indexes a key element with representation feature $x_k \in \mathbb{R}^C$, where C is the feature dimension, Ω_q and Ω_k specify the query and key elements, respectively. The attention weights A_{mqk} are calculated by

$$A_{mqk} = \text{softmax}\left(\frac{z_q^T U_m^T V_m x_k}{\sqrt{C_v}}\right), \tag{1}$$

where $U_m \in \mathbb{R}^{C_v \times C}$ and $V_m \in \mathbb{R}^{C_v \times C}$ are learnable weights of q and k. The process of calculating the aggregated features in the multi-head attention mechanism can be represented by

$$\text{MHA}\left(z_q, x\right) = \sum_{m=1}^{M} W_m \left[\sum_{k \in \Omega_k} A_{mqk} \cdot {W'}_m x_k\right], \tag{2}$$

where m indexes the attention heads, ${W'}_m \in \mathbb{R}^{C_v \times C}$ and $W_m \in \mathbb{R}^{C \times C_v}$ are also learnable weights ($C_v = C/M$). Moreover, to disambiguate different spatial positions, the representation features z_q and x_k are usually introduced with positional embeddings.

The second attention mechanism is the deformable attention mechanism [47]. It not only preserves the spatial structure of the feature map but also helps the detector accelerate convergence and reduce computational complexity. Given an input feature map $x \in \mathbb{R}^{C \times H \times W}$, let q index a query element with content feature z_q and a 2-d reference point p_q, the deformable attention feature is calculated by

$$\text{DA}\left(z_q, p_q, x\right) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot {W'}_m x\left(p_q + \Delta p_{mqk}\right)\right], \tag{3}$$

where m indexes the attention head, k indexes the sampled keys, and K is the total sampled key number ($K \ll HW$).

Δp_{mqk} and A_{mqk} denote the sampling offset and attention weight of k^{th} sampling point in the m^{th} attention head, respectively. Both Δp_{mqk} and A_{mqk} are obtained via linear projection over the query feature z_q.

To the best of our knowledge, there is currently no research integrating prompts into deformable attention mechanisms. The difficulty lies in the fact that deformable attention is a spatially local attention structure, while prompt interaction requires a global attention structure, making their integration challenging. Therefore, we only introduce prompts in the multi-head attention mechanism, which is used for object query interaction in the decoder.

3.2. Pseudo Labeling for Mining Potential Objects

In IOD, unlabeled previous task objects may appear in the background of current task images. Properly mining these previous task objects can significantly reduce forgetting, while treating these objects as background can lead to more severe forgetting. To make a fair comparison, we introduce the same pseudo-labeling method as MD-DETR [2] to mine the knowledge of unlabeled objects in the background.

In T_t, we employ the detector trained on T_{t-1} to infer on each training sample, obtaining predictions: $\hat{y}_i = \left\{\hat{s}_i, \hat{b}_i\right\}$. Here, $\hat{s}_i$ represents the score for the highest-scoring category, and $\hat{b}_i$ provides the bounding box coordinates for this prediction. Pseudo-labeling mechanism [5, 9, 25] sets a threshold τ to filter predictions with $\hat{s}_i$ higher than this threshold as pseudo label $\tilde{y}_i = \left\{\tilde{c}_i, \tilde{b}_i\right\}$. The threshold τ ensures that only the most reliable predictions are used when generating pseudo labels. Here, $\tilde{c}_i$ represents the pseudo label's category name, $\tilde{b}_i$ is the bounding box coordinates for this pseudo label. Pseudo labels incorporate the knowledge from previous tasks $\{T_1 \dots T_{t-1}\}$, effectively alleviating the detector's forgetting.

4. METHODS

4.1. Overview

We propose P^{2}IOD to alleviate the confusion in prompt-pool-based IOD methods when learning co-occurring object knowledge. Fig. 2 illustrates the complete framework. We hold that prompt structures should exhibit the ability to adaptively consolidate knowledge across tasks while constraining updates to prevent catastrophic forgetting. Therefore, P^{2}IOD redesigns the prompts pool into parameterized prompts, leveraging neural networks' inherent adaptive consolidation to naturally update learned knowledge in response to losses from co-occurring objects. The parameterized prompts are implemented as multi-layer perceptron (MLP) bottlenecks composed of feedforward networks and are integrated into different decoder layers to enhance prompt diversity. Considering that IOD algorithms are often deployed on resource-constrained scenarios such as edge devices [21, 22], P^{2}IOD proposes a parameterized prompt fusion strategy to constrain updates to the prompt structure with low computational overhead. During incremental training, only the parameters of class embeddings, bounding box embeddings, and the parametrized prompt structure (θ) are trainable, while all other parameters (θ^*) remain frozen to prevent knowledge forgetting. Fig. 2 illustrates the complete framework.

4.2. Parameterized Prompt Structure

We hold that the prompt structure should adaptively consolidate the potential knowledge that emerges in the co-occurring scenarios. To achieve this, we design the prompt structure as an MLP bottleneck composed of FNN layers rather than the prompts pool. This parameterized prompt structure encodes prompt-related knowledge into the neural network weight space and generates instance-specific prompts.

We follow the method in [2] by employing the frozen pre-trained detector as a query function to extract queries, which are then utilized as inputs to the parameterized prompt. Given an input instance x, a set of proposals $P \in \mathbb{R}^{N \times D}$ is generated through a single pass of $P = \theta^*(x)$, where N is the number of proposals and D is the embedding dimension of each proposal. P contains both object and background information related to the instance x, which is preliminarily extracted by the frozen pre-trained detector. However, the number of proposals in P is too large to be directly used as query features, requiring compression. Unlike the approach in [2], where only object-related proposals are compressed, we find that jointly compressing both object and background proposals in P^{2}IOD is more effective. We believe this is because including background knowledge in the prompts helps the detector better distinguish between foreground and background. In contrast, [2] shows that adding background query features degrades its performance. This degradation stems from the knowledge retention and matching mechanism of the prompts pool, which limits the acquisition of background knowledge, indicating that the prompts pool structure is not suitable for IOD. Detailed analysis and experimental comparisons can be found in Appendix B.4. The entire query function Q can be represented as follows:

$$Q(x, \theta^*) = \frac{1}{N} \sum_{n=1}^{N} \{\theta^*(x)\}_n, \tag{4}$$

where $\{\theta^*(x)\}_n$ is the n^{th} proposal.

We take the $Q(x, \theta^*)$ as the input to the parameterized prompt, which outputs the prompts $p \in \mathbb{R}^{L_p \times D}$. L_p represents the length of prompts. The parameterized prompt is an MLP bottleneck composed of two FNN layers, which

can effectively remove redundant information in the query through linear dimensionality reduction. The entire process can be represented as:

$$p = \text{ReLU}\left(Q\left(x, \theta^*\right) \cdot W^{(1)}\right) \cdot W^{(2)}, \tag{5}$$

where $W^{(1)} \in \mathbb{R}^{D \times d}$ represents an FNN layer for dimensionality reduction, in which d is the bottleneck dimension; $W^{(2)} \in \mathbb{R}^{d \times \hat{D}}$ is an FNN layer with upper-projection parameters, where $\hat{D} = D \times L_p$; RELU is non-linear activation in between.

The $p \in \mathbb{R}^{L_p \times D}$ are integrated into the decoder's multi-head self-attention layers. The process can be expressed as follows:

$$\text{MHA}_p\left(q_o, p\right) = \sum_{m=1}^{M} W_m \left[\sum_k A_{mqk} \cdot [{W'}_m q_o : p_v]\right], \tag{6}$$

$$A_{mqk} = \text{softmax}\left(\frac{q_o^T U_m^T [V_m q_o : p_k]}{\sqrt{C_v}}\right), \tag{7}$$

where q_o is the objects queries [47], $[x : y]$ represents the concatenate operation. Following [37], we assign $p \in \mathbb{R}^{L_p \times D}$ into $p_k \in \mathbb{R}^{\frac{L_p}{2} \times D}$ and $p_v \in \mathbb{R}^{\frac{L_p}{2} \times D}$, and concatenate them to ${W'}_m q_o$ and $V_m q_o$ respectively, while keeping $q_o^T U_m^T$ unchanged. This manner ensures that the input and output sequence lengths remain the same before and after integrating prompts. To increase prompt diversity, we introduce independent parameterized prompts into each decoder layer of the frozen pre-trained detector.

4.3. Parameterized Prompt Fusion for Incremental Learning

The parameterized prompt also faces catastrophic forgetting during incremental learning. To address the forgetting, we introduce model fusion after each incremental training process. During the fusion process, we aim to retain the important parameters of each task and average the consistent parameters across tasks. We also introduce a sparse loss to concentrate the knowledge of each task in different parameter subsets to facilitate the model fusion.

Model fusion. For a sequence of incremental tasks $\{T_1 \dots T_t\}$, we add a fusion process after the training process in $\{T_2 \dots T_t\}$ to fuse the parameterized prompt of the current task with those of the previous task. We denote the parameterized prompt obtained from training as θ_t and those obtained from fusion as θ_t^f. For T_t ($t \geqslant 2$), the parameterized prompt used for testing is θ_t^f, which is obtained by fusing θ_t and θ_{t-1}^f (when $t = 2$, we fuse θ_2 and θ_1).

We fuse θ_t and θ_{t-1}^f based on the degree of parameter variation. To describe the variation of parameterized prompt between current and previous tasks (θ_t and θ_{t-1}^f), we compute the task vector $\boldsymbol{v}_t = \theta_t - \theta_{t-1}^f$. The task vector $\boldsymbol{v}_t$ simultaneously conveys the parameter variation's magnitude and direction. Inspired by [40], we decompose the task vector $\boldsymbol{v}_t$ into a magnitude vector μ_t ($\mu_t = |\boldsymbol{v}_t|$) and a sign vector γ_t ($\gamma_t = \text{sgn}\left(\boldsymbol{v}_t\right)$, taking values in ± 1) as $\boldsymbol{v}_t = \gamma_t \odot \mu_t$, where $\odot$ is the element-wise product. We also describe the overall variation of parameterized prompt in previous tasks by computing the task vector $\boldsymbol{v}_{t-1}^f = \theta_{t-1}^f - \theta_{init}$, where θ_{init} denotes the initialized parameterized prompt.

During the T_t fusion process, we preserve critical parameters guided by μ_t and μ_{t-1}^f, and average consistent parameters based on γ_t and γ_{t-1}^f. To preserve critical parameters, we first sort the values in μ_{t-1}^f and select the top-k% indices, denoted as $\mathcal{I}_{t-1}^f$. The corresponding parameter in θ_{t-1}^f are preserved with priority at the indices in $\mathcal{I}_{t-1}^f$. Next, we sort μ_t and identify the top-l% indices, denoted as $\mathcal{I}_t$. The parameter in θ_t are preserved at the indices in $\mathcal{I}_t$, excluding any overlap with the indices in $\mathcal{I}_{t-1}^f$. To average consistent parameters, we locate positions where $\gamma_t = \gamma_{t-1}^f$, indicating directional consistency. At these positions, excluding those already reserved for preservation (i.e., $\mathcal{I}_{t-1}^f \cup \mathcal{I}_t$), we take the average of θ_t and θ_{t-1}^f. Finally, all remaining undecided parameters are assigned the corresponding values from θ_{t-1}^f. The overall fusion process can be formally expressed as follows:

$$\theta_t^f[i] = \begin{cases} \theta_{t-1}^f[i], & i \in \mathcal{I}_{t-1}^f \\ \theta_t[i], & i \in \mathcal{I}_t \setminus \mathcal{I}_{t-1}^f \\ \frac{1}{2}(\theta_t[i] + \theta_{t-1}^f[i]), & \gamma_t[i] = \gamma_{t-1}^f[i],\ i \notin \mathcal{I}_{t-1}^f \cup \mathcal{I}_t \\ \theta_{t-1}^f[i], & \text{otherwise} \end{cases} \tag{8}$$

where i denotes the i-th parameter in the parameterized prompts. The pseudo-code for parameterized prompt fusion is outlined in Appendix C.1.

Sparse Loss. In model fusion, we retain the important parameters of both current and previous tasks to maintain the learned knowledge. However, in practice, the learned parameters exhibit redundancy, making it difficult to identify parameter importance. We expect the model to learn sparse parameters to concentrate critical knowledge in a small subset of parameters. Therefore, we introduce an additional L_1 loss as a sparse loss L_s, defined as:

$$L_s = \lambda \sum_j |\theta_j|, \tag{9}$$

where λ controlls the sparsity level, and θ_j refers to the parameterized prompts in the j^{th} decoder layer.

Table 1. Average precision (AP_{50}, %) is compared on the PASCAL VOC2007 dataset under single-step settings of 19+1, 15+1, 10+10, and 5+15. We add the superscript * to the accuracy that may be overestimated. The reasons for the overestimation are detailed in 5.1.

Method	19+1			15+5			10+10		
	1-19	20	1-20	1-15	16-20	1-20	1-10	11-20	1-20
OW-DETR [9]	70.2	62.0	69.8	72.2	59.8	69.1	63.5	67.9	65.7
ABR [24]	71.0	69.7	70.9	73.0	65.1	71.0	71.2	72.8	72.0
Faster ILOD [28]	68.9	61.1	68.5	71.6	56.9	67.9	69.8	54.5	62.1
PROB [48]	73.9	48.5	72.6	73.5	60.8	70.1	66.0	67.2	66.5
PseudoRM [43]	72.9	67.3	72.6	73.4	60.9	70.3	69.1	68.6	68.9
BPF [27]	74.5	65.3	74.1	75.9	63.0	72.7	71.7	74.0	72.9
VLM-PL [17]	73.7*	89.3*	73.6	73.9*	82.4*	72.4	80.3*	76.3*	78.3
MD-DETR (MS COCO) [2]	76.8*	67.2*	76.1	77.4*	69.4*	76.7	73.1*	77.5*	73.2
P^{2}IOD (MS COCO)	78.5	62.6	**77.7**	83.3	66.9	**79.2**	82.0	80.4	**81.2**
MD-DETR (Objects365) [2]	89.4	68.7	88.3	86.1	84.7	85.8	81.8	87.3	84.6
P^{2}IOD (Objects365)	89.7	77.5	**89.1**	91.2	85.2	**89.7**	88.4	91.1	**89.8**

5. EXPERIMENTS

5.1. Experimental Settings

Datasets. We evaluate our proposed method on PASCAL VOC2007 [6] and MS COCO [20]. The PASCAL VOC2007 contains 20 diverse object classes, including 9,963 images, split into 5,011 for training and 4,952 for testing. The MS COCO, with its 80 object classes spread across 118,000 training images and 5,000 evaluation images, makes it a more challenging benchmark.

Eval metrics. Followed by [2], we use the mean average precision at an IOU threshold of 0.5 (AP_{50}, %) as the metric. For PASCAL VOC2007, following previous works [17], we provide the AP_{50} of the current task classes and the previous task classes to better reflect the method's stability and plasticity. There are two evaluation methods for obtaining task precision: validating on the entire test set versus using task-specific test subsets. The second method yields higher precision for the same detector. We adopt the first method and add superscript * to results from the second method to ensure fair comparison. For MS COCO, following previous works [16], we provide the AP_{50} of all learned classes after learning at each task.

Implementation details. We implement our proposed method based on Deformable-DETR [47] pre-trained on the MS COCO dataset and Co-DETR [49] pre-trained on the Objects365 dataset [30], both obtained from HuggingFace. The large-scale Objects365 dataset contains 365 categories and over 600,000 images, making it a suitable pre-training source for incremental learning experiments on MS COCO dataset. Furthermore, since the official MD-DETR implementation is only available on Deformable-DETR, we port MD-DETR [2] on Co-DETR for a fair comparison.

5.2. Comparison

Single-step setting. We compare three single-step scenarios on the PASCAL VOC2007 dataset, where the co-occurrence levels gradually increase in the 19+1, 15+5, and 10+10 settings. As shown in Tab. 1, P^{2}IOD with MS COCO and Objects365 pretrained detectors achieve excellent performance across all experimental settings. Compared to the prompt-pool-based MD-DETR, P^{2}IOD achieves accuracy improvements of 1.6% / 0.8%, 2.5% / 3.9%, and 8.0% / 5.2% in the respective scenarios, indicating that the performance advantage of P^{2}IOD becomes increasingly significant as the co-occurrence level rises. This trend further demonstrates that P^{2}IOD mitigates the interference caused by prompts pool confusion in co-occurring scenarios. We also compare single-step scenarios on the MS COCO dataset in the Appendix B.1, and the results further demonstrate the effectiveness of P^{2}IOD.

Multi-step setting. We compare the multi-step settings on PASCAL VOC2007 and MS COCO datasets. Tab. 4 shows that on PASCAL VOC2007, the performance degradation of MD-DETR becomes increasingly severe as the number of incremental steps grows. The degradation stems from prompts pool confusion which intensifies as the number of tasks increases. In contrast, our method effectively mitigates such confusion, consistently achieving superior performance across all settings. For the MS COCO dataset, as shown in Tab. 2, P^{2}IOD also exhibits the aforementioned advantages. Moreover, P^{2}IOD consistently outperforms other existing approaches across different settings on both datasets, demonstrating its effectiveness and the strong potential of prompt-based techniques in IOD.

5.3. Analysis

Ablation Study. In Tab. 3, categories 1-5 reflect the stability, while categories 6-20 mainly reflect plasticity. After in-

Table 2. Average precision (AP_{50}, %) is compared on the MS COCO dataset under multi-step settings of 40+20+20 and 40+10+10+10+10.

Method	$\mathcal{T}_1$ (1-40)	**40+20+20**		**40+10+10+10+10**			
		$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
ERD [7]	63.7	54.5	48.6	53.9	46.7	39.9	31.8
CL-DETR [25]	63.7	58.3	54.1	54.4	50.2	45.6	38.2
DyQ-DETR [45]	63.7	57.0	55.7	55.9	53.8	50.8	49.8
SDDGR [16]	68.6	62.6	59.5	62.8	60.2	59.0	54.7
LEA [33]	75.3	62.0	57.5	66.3	61.7	59.7	56.5
GCD [36]	-	-	60.4	-	-	-	55.1
MD-DETR (Objects365) [2]	79.0	69.4	60.3	68.1	61.7	53.7	49.4
P^{2}IOD (Objects365)	79.6	71.3	**68.8**	74.1	70.9	69.3	**64.8**

Table 3. Ablation study results (AP_{50}, %) for component's contribution evaluated on PASCAL VOC2007 in 5+5+5+5 setting.

Methods	Pseudo Labeling	Parameterized Prompt Structure	Model Fusion	Sparse Loss	**5+5+5+5**		
					1-5	**6-20**	**1-20**
(a)					73.3	65.4	67.4
(b)	✓				73.3	64.6	66.8
(c)	✓	✓			70.7	76.6	75.1
(d)	✓	✓	✓		73.1	76.0	75.3
(e)		✓	✓	✓	67.0	73.0	71.5
(f)	✓	✓	✓	✓	**74.0**	**77.2**	**76.4**

Table 4. Average precision (AP_{50}, %) is compared on the PASCAL VOC2007 dataset under multi-step settings of 10+5+5 and 5+5+5+5. We add the superscript * to the accuracy may be overestimated. The reasons for the overestimation are detailed in 5.1.

Method	**10+5+5**			**5+5+5+5**		
	1-10	**10-20**	**1-20**	**1-5**	**6-20**	**1-20**
ABR [24]	68.7	67.1	67.9	64.7	56.4	58.4
Faster ILOD [28]	68.3	57.9	63.1	55.7	16.0	25.9
MMA [3]	67.4	60.5	64.0	62.3	31.2	38.9
BPF [27]	69.1	68.2	68.7	60.6	63.1	62.5
VLM-PL [17]	67.9*	67.9*	67.9	64.5*	68.4*	65.5
MD-DETR (MS COCO) [2]	68.5	60.3	60.7	55.2	63.6	61.5
P^{2}IOD (MS COCO)	81.3	74.2	**77.8**	74.0	77.2	**76.4**
MD-DETR (Objects365) [2]	80.1	87.5	83.8	60.9	80.7	75.8
P^{2}IOD (Objects365)	89.1	89.1	**89.1**	86.4	87.4	**87.1**

troducing the pseudo-labeling method (b), due to the lack of learnable parameters, pseudo-labeling not only fails to enhance stability but also interferes with current task learning, reducing plasticity. The parameterized prompt structure (c) increases the accuracy of categories 6-20 by 9.5%, significantly enhancing plasticity, but the accuracy of categories 1-5 drops by 2.6%, indicating that forgetting still exists. The model fusion (d) alleviates the forgetting problem and balances stability and plasticity to some extent. The sparse loss (f) reduces parameter conflicts between tasks, improving overall accuracy by 1.1%. When combined with all methods, it increases the overall accuracy by 9.0% compared to the baseline. Furthermore, removing the pseudo-labeling (e) results in a significant decrease in stability. The observation demonstrates that our method, by alleviating the prompts pool confusion, allows the pseudo labeling mechanism to effectively mine old-class objects in the background without introducing adverse effects.

Impact of Hidden Layer Dimension. We analyze the impact of the hidden layer dimension in the parameterized prompt structure. The dimension of the hidden layer influences the degree of dimensionality reduction applied to the proposals. We conduct experiments in the PASCAL VOC2007 under the 5+5+5+5 setting. In Fig. 4, as the hidden layer dimension increases, the model's accuracy initially improves and then declines, suggesting that a moderate increase in the hidden dimension helps retain critical information, while an overly large dimension introduces redundant information that interferes with prompt generation. As the hidden dimension increases, the number of parameters in the parameterized prompt increases accordingly. Our method achieves a favorable trade-off between performance and parameter efficiency (76.3%, 1.1M) when the hidden dimension is set to 64. In contrast, MD-DETR [2] requires more parameters, while simultaneously achieving lower accuracy (61.5%, 1.8M).

Prompt Comparison. We compare the distribution similarity of prompts across tasks between P^{2}IOD and MD-DETR [2]. We conduct this experiment on PASCAL VOC2007 and use Maximum Mean Discrepancy (MMD) [8] to evaluate the distribution similarity of prompts, with the average MMD (A-MMD) across all tasks as the evaluation metric. A larger A-MMD value indicates a more significant prompt diversity. As shown in Fig. 5, the diversity of prompts in our P^{2}IOD increases with the depth of decoder layers, and the diversity at the final layer is significantly higher

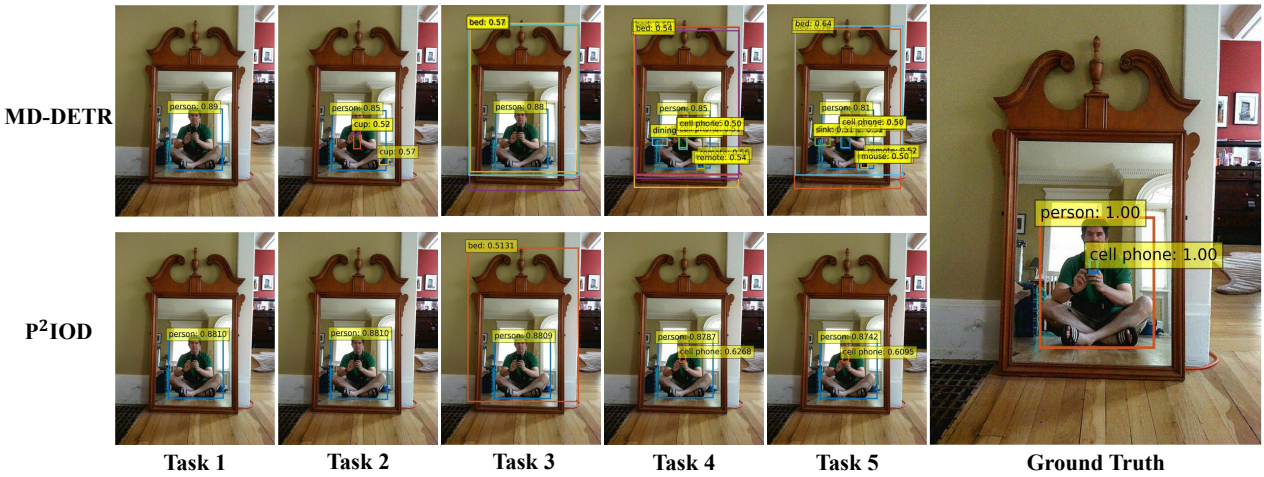

Figure 3. Visualized comparison between P^{2}IOD and MD-DETR. MD-DETR exhibits more false positives and a faster decline in the positive target's confidence than P^{2}IOD, indicating the impact of prompts pool confusion.

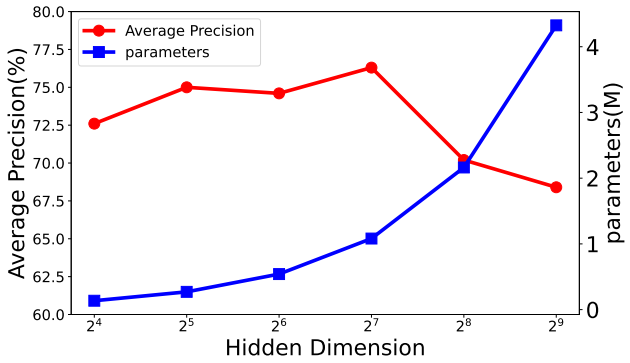

Figure 4. Average Precision (AP_{50}, %) and parameters (M) on different hidden layer dimensions in the parameterized prompt structure on PASCAL VOC2007 under the 5+5+5+5 setting.

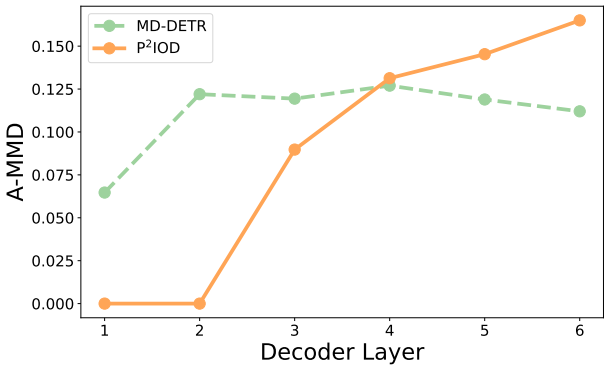

Figure 5. Distribution similarity of prompts across different decoder layers in MD-DETR and P^{2}IOD. A larger A-MMD value indicates a more significant prompt diversity.

than that of MD-DETR. Our method generates category-independent prompts in shallow layers and category-related prompts in deep layers, aligning well with the multi-layer decoder architecture, while MD-DETR is constrained by its pool structure and struggles to match this characteristic. Furthermore, our method can generate more diverse prompts at the final layer used for object prediction. The variations in prompt distributions highlight the effectiveness of our approach.

5.4. Visualized Comparison

We analyze the visualized comparison between P^{2}IOD and MD-DETR to illustrate that the confusion is being addressed. In Fig. 3, the visualizations of MD-DETR exhibit numerous false positives, indicating that the confused prompts pool introduces incorrect prompts into the detector, thereby increasing scores for irrelevant categories. In contrast, P^{2}IOD reduces such false positives, demonstrating the effectiveness of our approach in mitigating confusion. We also observe that although P^{2}IOD and MD-DETR have nearly identical confidence in detecting people in the first task, MD-DETR's confidence rapidly declines as the number of tasks increases, suggesting that confusion in MD-DETR undermines confidence in positive objects. In contrast, the confidence in P^{2}IOD remains almost unchanged, proving that our method is unaffected by confusion.

6. Conclusion

In this study, we identify a confusion issue within the prompt-pool-based IOD methods. To address this issue, we argue that prompts in IOD should not be exclusively assigned to individual tasks but should instead exhibit adaptive consolidation properties across tasks, with constrained updates. We propose a parameterized prompt structure and parameterized prompt fusion to validate our hypothesis. Experiments on multiple datasets demonstrate that our framework exhibits superior performance. To our knowledge, this is the first work addressing prompts pool confusion in IOD, laying a foundation for broader prompt-based IOD applications.

Acknowledgments

This work is partially supported by the Chinese Academy of Sciences Project for Young Scientists in Basic Research (YSBR-107).

References

[1] Zijia An, Boyu Diao, Libo Huang, Ruiqi Liu, Zhulin An, and Yongjun Xu. Ior: Inversed objects replay for incremental object detection. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 1

[2] Gaurav Bhatt, James Ross, and Leonid Sigal. Preventing catastrophic forgetting through memory networks in continuous detection. In *European Conference on Computer Vision*, pages 442–458. Springer, 2024. 1, 2, 3, 4, 6, 7

[3] Fabio Cermelli, Antonino Geraci, Dario Fontanel, and Barbara Caputo. Modeling missing annotations for incremental learning in object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3700–3710, 2022. 2, 7

[4] Fang Cheng, Hui Liu, and Xinwei Lv. Metagnsdformer: Meta-learning enhanced gated non-stationary informer with frequency-aware attention for point-interval remaining useful life prediction of lithium-ion batteries. *Advanced Engineering Informatics*, 69:103798, 2026. 1

[5] Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Open world detr: Transformer based open world object detection. *arXiv preprint arXiv:2212.02969*, 2022. 4

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 6, 1

[7] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9427–9436, 2022. 1, 2, 7

[8] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 7

[9] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9235–9244, 2022. 4, 6

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[11] Jincai Huang, Yongjun Xu, Qi Wang, Qi Cheems Wang, Xingxing Liang, Fei Wang, Zhao Zhang, Wei Wei, Boxuan Zhang, Libo Huang, et al. Foundation models and intelligent decision-making: Progress, challenges, and perspectives. *The Innovation*, 2025. 1

[12] Libo Huang, Zhulin An, Yan Zeng, Yongjun Xu, et al. Kfc: Knowledge reconstruction and feedback consolidation enable efficient and effective continual generative learning. In *The Second Tiny Papers Track at ICLR 2024*, 2024. 1

[13] Libo Huang, Yan Zeng, Chuanguang Yang, Zhulin An, Boyu Diao, and Yongjun Xu. etag: Class-incremental learning via embedding distillation and task-oriented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12591–12599, 2024. 2

[14] Libo Huang, Zhulin An, Chuanguang Yang, Boyu Diao, Fei Wang, Yan Zeng, Zhifeng Hao, and Yongjun Xu. Preprompt: Predictive prompting for class incremental learning. *arXiv preprint arXiv:2505.08586*, 2025. 1

[15] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11847–11857, 2023. 2

[16] Junsu Kim, Hoseong Cho, Jihyeon Kim, Yihalem Yimolal Tiruneh, and Seungryul Baek. Sddgr: Stable diffusion-based deep generative replay for class incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28772–28781, 2024. 6, 7, 2

[17] Junsu Kim, Yunhoe Ku, Jihyeon Kim, Junuk Cha, and Seungryul Baek. Vlm-pl: Advanced pseudo labeling approach for class incremental object detection via vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4170–4181, 2024. 1, 2, 6, 7

[18] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. Rilod: Near real-time incremental learning for object detection at the edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 113–126, 2019. 2

[19] Xiangqi Li, Libo Huang, Zhulin An, Weilun Feng, Chuanguang Yang, Boyu Diao, Fei Wang, and Yongjun Xu. Geometric feature embedding for effective 3d few-shot class incremental learning. In *Forty-second International Conference on Machine Learning*, 2025. 1

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 6

[21] Hangda Liu, Boyu Diao, Yu Yang, Wenxin Chen, Xiaohui Peng, and Yongjun Xu. Gensor: A graph-based construction tensor compilation method for deep learning. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 558–569. IEEE, 2025. 4

[22] Ruiqi Liu, Boyu Diao, Libo Huang, Zijia An, Zhulin An, and Yongjun Xu. Continual learning in the frequency domain. *Advances in Neural Information Processing Systems*, 37:85389–85411, 2024. 4

[23] Ruiqi Liu, Boyu Diao, Libo Huang, Zijia An, Hangda Liu, Zhulin An, and Yongjun Xu. Low-redundancy distillation for continual learning. *Pattern Recognition*, page 111712, 2025. 2

[24] Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost Van De Weijer. Augmented box replay: Overcoming foreground shift for incremental object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11367–11377, 2023. 6, 7

[25] Yaoyao Liu, Bernt Schiele, Andrea Vedaldi, and Christian Rupprecht. Continual detection transformer for incremental object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23799–23808, 2023. 1, 4, 7, 2

[26] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 1

[27] Qijie Mo, Yipeng Gao, Shenghao Fu, Junkai Yan, Ancong Wu, and Wei-Shi Zheng. Bridge past and future: Overcoming information asymmetry in incremental object detection. In *European Conference on Computer Vision*, pages 463–480. Springer, 2024. 1, 6, 7

[28] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern recognition letters*, 140:109–115, 2020. 6, 7

[29] Can Peng, Kun Zhao, Sam Maksoud, Meng Li, and Brian C Lovell. Sid: Incremental learning for anchor-free object detection via selective and inter-related distillation. *Computer vision and image understanding*, 210:103229, 2021. 2

[30] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 6, 1

[31] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017. 2

[32] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11909–11919, 2023. 2

[33] Xiang Song, Yuhang He, Jingyuan Li, Qiang Wang, and Yihong Gong. Learning endogenous attention for incremental object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30354–30364, 2025. 1, 7, 2

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[35] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[36] Xu Wang, Zilei Wang, and Zihan Lin. Gcd: Advancing vision-language models for incremental object detection via global alignment and correspondence distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8015–8023, 2025. 7, 2

[37] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pages 631–648. Springer, 2022. 2, 5

[38] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022. 2

[39] Fangwen Wu, Lechao Cheng, Shengeng Tang, Xiaofeng Zhu, Chaowei Fang, Dingwen Zhang, and Meng Wang. Navigating semantic drift in task-agnostic class-incremental learning. *arXiv preprint arXiv:2502.07560*, 2025. 2

[40] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023. 5

[41] Dongbao Yang, Yu Zhou, Wei Shi, Dayan Wu, and Weiping Wang. Rd-iod: Two-level residual-distillation-based triple-network for incremental object detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(1):1–23, 2022. 2

[42] Dongbao Yang, Yu Zhou, Aoting Zhang, Xurui Sun, Dayan Wu, Weiping Wang, and Qixiang Ye. Multi-view correlation distillation for incremental object detection. *Pattern Recognition*, 131:108863, 2022. 2

[43] Dongbao Yang, Yu Zhou, Xiaopeng Hong, Aoting Zhang, Xin Wei, Linchengxi Zeng, Zhi Qiao, and Weipinng Wang. Pseudo object replay and mining for incremental object detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 153–162, 2023. 2, 6

[44] Chengqing Yu, Fei Wang, Chuanguang Yang, Zezhi Shao, Tao Sun, Tangwen Qian, Wei Wei, Zhulin An, and Yongjun Xu. Merlin: Multi-view representation learning for robust multivariate time series forecasting with unfixed missing rates. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3633–3644, 2025. 2

[45] Jichuan Zhang, Wei Li, Shuang Cheng, Yali Li, and Shengjin Wang. Dynamic object queries for transformer-based incremental object detection. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 7

[46] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*, 2024. 2

[47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3, 5, 6, 1

[48] Orr Zohar, Kuan-Chieh Wang, and Serena Yeung. Prob: Probabilistic objectness for open world object detection. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11444–11453, 2023. 6

[49] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 3, 6, 1

[50] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023. 1